NANYANG TECHNOLOGICAL UNIVERSITY

EVENT-DRIVEN DYNAMIC SCHEDULING OF DISCRETE MANUFACTURING

CHONG CHIN SOON

SCHOOL OF ELECTRICAL & ELECTRONIC ENGINEERING

2007

Event-driven Dynamic Scheduling Of Discrete Manufacturing

Chong Chin Soon

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

Acknowledgement

I would like to express my sincere and deepest thanks to my supervisors, Professor Robert Gay and Associate Professor Appa Iyer Sivakumar for providing me the guidance, enthusiastic encouragement and invaluable advice throughout the course of the research. At many stages of my research, I benefited from their advice, particularly so when exploring new ideas. Their positive outlook and confidence in my research always inspire me to continue more in-depth research. Their indispensable help contributed enormously to the production of this thesis.

I would like to acknowledge the research opportunity given by Singapore Institute of Manufacturing Technology (SIMTech). I also wish to thank the Deputy Executive Director, directors and other colleagues from SIMTech for providing me various facilities to carry out this research. Their help are gratefully acknowledged.

I would like also to thank my family for helping me keep the whole thing in perspective.

Summary

The increasingly business globalization has set new challenges to manufacturing systems, and across their distributed supply chains, primarily due to shift from a vendor's to a customer's market. In this new environment, product changes are frequent, time to market is short, product variants are increased, demands are fluctuating and last but not least, production costs are to be reduced significantly. These market changes translate into new production requirements and thus demand significant operational improvement for manufacturing enterprises facing the global competition. The most important of all is for the enterprises to maintain high performance, amid all process changes and disruptions, over the whole manufacturing period. Most manufacturing systems today have excellent system performance under normal conditions, but many may collapse under substantial process disruptions.

Scheduling can play a significant role in coping with changes and disruptions in manufacturing system. However, traditional scheduling and optimization techniques, which deal with static and deterministic scheduling problems, are ineffective to deal with dynamic and stochastic changes in production environments. Scheduling today must take real-time events into consideration so as to achieve better response and maintain good performance in a continuous way. This scheduling problem is not trivial because the scheduling performance can be sensitive to disruptions, scheduling problems are NP-hard, and manufacturing environments are dynamic, uncertain and often unpredictable.

This research seeks to address the new scheduling challenges through event-driven dynamic scheduling particularly for discrete manufacturing environment. The research investigates on how the performance of production scheduling can be maintained or even improved under production disruptions, using scheduling techniques. The work thus encompasses research into strategies and heuristics to make scheduling approaches more effective in solving dynamic and stochastic scheduling problems, with primary emphasis on long-term system performance. Our work adopts a three-phase sequential approach, which involves a survey of existing scheduling approaches, experimentation and evaluation of scheduling approaches, and development of an effective strategy and/or algorithm.

The survey of past literature reveals that manufacturing environments, solution features, and scheduling techniques (strategies and heuristics) are the three major factors that can impact on the effectiveness of scheduling. Although there have been efforts in consolidating research in scheduling, a unified approach that aims at identifying the various factors and resolving the deficiencies in the existing approaches is still lacking. Our approach enables us to distinguish the most significant scheduling factors from the less important ones, and guides us to address the critical scheduling problems. Based on the survey, we decide to focus on the long-term steady state performance of manufacturing systems rather than the performance of specific schedules. With this criterion as our primary goal, we are able to compare the relative performance of the commonly used scheduling techniques in discrete manufacturing environments. These techniques are periodic scheduling and dispatching rules.

Our single-machine and two-machine analytical models show that event-driven scheduling can achieve better performance than periodic scheduling in terms of mean cycle time of jobs. To verify the analytical results and compare the performance of different scheduling techniques, we develop an object-oriented simulation test bed and a set of scheduling algorithms and dispatching rules for the purposes of simulation experimentation. Through simulation study using dynamic and stochastic manufacturing models, we confirm the analytical results.

Subsequent simulation study on the performance of event-driven scheduling algorithms and dispatching rules indicates that the performance of the various heuristics varies with the shop floor conditions. Event-driven scheduling algorithms only outperform dispatching rules under certain job shop conditions, but not under other conditions. Event-driven scheduling algorithms are preferred in not too highly utilized and more stable manufacturing environments. Further, no single scheduling heuristic can perform well in all different conditions of dynamic and stochastic manufacturing environments.

Based on the simulation study, we finally conclude that under highly dynamic and stochastic manufacturing environments, it is important to ensure that all machines are equally loaded with jobs in order to achieve better mean cycle time performance. Applying this principle, we derive an improved dynamic scheduling approach. Further experiments show that the proposed scheduling approach outperforms existing scheduling approaches on the stated shop floor conditions. In summary, an effective scheduling strategy for dynamic and stochastic manufacturing environments is to use an efficient event-driven scheduling algorithm for not too highly utilized and more stable manufacturing environments, and a dynamic scheduling approach that considers balanced loading on all machines for highly dynamic and stochastic manufacturing environments.

Table of Contents

ACKNOWLEDGEMENT .... **I**

SUMMARY .... **II**

TABLE OF CONTENTS .... **V**

LIST OF FIGURES .... **X**

LIST OF TABLES .... **XIV**

LIST OF ABBREVIATIONS .... **XVII**

NOMENCLATURE .... **XX**

AUTHOR'S PUBLICATIONS .... **XXI**

1 INTRODUCTION **1**

1.1 Background and Motivation 1

1.2 Objectives 5

1.3 Contribution 7

1.4 Thesis Organization 9

2 THEORETICAL BACKROUND AND RELATED WORK **11**

2.1 Introduction 11

2.2 A Reactive Scheduling Framework 15

2.2.1 Scheduling and Rescheduling Environments 15

2.2.2 Rescheduling Strategies 17

2.2.3 Rescheduling Methods ………………………………….. 20

2.3 Rescheduling Performance Measures ………………………………. 22

2.4 Scheduling and Rescheduling Approaches ……………………….. 23

2.4.1 Scheduling Approaches ………………………………… 28

2.4.2 Rescheduling Approaches ……………………………… 35

2.4.2.1 Dynamic Approaches …………………………… 35

2.4.2.2 Reactive and Proactive Scheduling Approaches 36

2.4.2.2.1 Generating Robust Schedules ……….. 37

2.4.2.2.2 Updating Schedules ………………… 41

2.5 Impact of Rescheduling ……………………………………………... 52

2.6 Summary ………………………………………………………… 61

3 RESCHEDULING FACTORS AND PROBLEM FORMULATION ….. 64

3.1 Introduction ……………………………………………………. 64

3.2 Production Disturbances in Manufacturing Environments ………… 67

3.2.1 Diversity and Classification of Production Disturbances 68

3.2.2 Characteristics of Production Disturbances …………….. 70

3.3 Characteristics of Manufacturing Environments …………………… 72

3.4 Characteristics of Rescheduling …………………………………….. 73

3.4.1 Solution Features ………………………………………... 73

3.4.2 Rescheduling Features …………………………………... 77

3.4.2.1 Scheduling Horizon …………………………….. 77

3.4.2.2 Scheduling Frequency …………………………... 79

3.5 Performance Criteria and Metrics …………………………………... 80

3.5.1 Performance Measures: Scheduling …………………….. 83

3.5.2 Performance Measures: Rescheduling ………………….. 84

3.5.2.1 Solution Stability …………………… 84

3.5.2.1.1 Starting Time Deviation …….. 84

3.5.2.1.2 Sequence Deviation ………… 85

3.5.2.2 Solution Adequacy ………………….. 86

3.6 Disruption Modeling and Analysis ……………………………….. 87

3.6.1 Steady State Model ……………………………………… 88

3.6.2 Perturbation Model ……………………………………… 90

3.7 Scope of Rescheduling Problems …………………………………… 93

3.8 Rescheduling Factors Not Considered in the Scope ………………… 99

3.9 Summary ………………………………………………………… 102

4 ANALYSIS OF EVENT-DRIVEN AND PERIODIC RESCHEDULING 105

4.1 Introduction ……………………………………………………… 105

4.2 Concepts of Periodic and Event-Driven Rescheduling ……………... 106

4.3 A Single-Machine Queuing Model ………………………………… 107

4.3.1 A Single-Arrival Queuing Model *M/G/1* ………………. 110

4.3.2 A Group-Arrival Queuing Model $M^b/G/1$……………….. 113

4.3.3 Comparison of Single-Arrival and Group-Arrival …....... 116

4.4 A Two-Machine Queuing Model …………………………………… 120

4.4.1 A Single-Arrival Queuing Model ……………………….. 121

4.4.2 A Group-Arrival Queuing Model ……………………….. 129

4.4.3 Analysis of Periodic and Event-Driven Rescheduling ….. 130

4.5 Summary ………………………………………………………… 132

5 AN OBJECT-ORIENTED SIMULATION TEST BED ………………… 134

5.1 Introduction ………………………………………………………. 134

5.2 Characteristics of a Typical Discrete Manufacturing System ………. 136

5.3 The Overall Framework …………………………………………….. 137

5.4 Application of Object-Oriented Programming ………………………. 143

5.5 Performance Measures ………………………………………………. 146

5.6 Dispatching Rules and Scheduling Algorithms …………………….. 147

5.6.1 Dispatching Rules ……………………………………….. 148

5.6.2 Scheduling Algorithms …………………………………….. 150

5.6.2.1 Recursive Simulation …………………………… 150

5.6.2.2 The Concept of Recursive Simulation based Algorithm …………………………………….. 151

5.6.2.3 Recursive Simulation based Scheduling Algorithms and Tabu Search ……………………. 154

5.7 Summary ………………………………………………………… 156

6 EXPERIMENTS, RESULTS AND ANALYSIS ………………………… 158

6.1 Introduction ………………………………………………………. 158

6.2 Scheduling Algorithms and Dispatching Rules …………………….. 159

6.2.1 Evaluation of Scheduling Heuristics Using Static Job Shop Problems ………………………………………... 160

6.2.2 Discussions on Experimental Results …………………… 164

6.3 The Job Shop Model and Assumptions …………………………...... 165

6.4 Experimental Design ………………………………………………... 168

6.4.1 Design of Experiment (DOE) on Frequency of 171

Rescheduling ……………………………………………....

6.4.2 Design of Experiment (DOE) on Scheduling Heuristics ... 173

6.5 Experimental Results ……………………………………………... 173

6.5.1 Experimental Results on Frequency of Rescheduling ….. 174

6.5.2 Experimental Results on Scheduling Heuristics ………... 176

6.6 Discussions and Analysis ………………………………………… 182

6.6.1 Discussions on Frequency of Rescheduling ……………. 182

6.6.2 Discussions on Scheduling Heuristics …………………... 186

6.7 Summary …………………………………………………………. 195

7 CONCLUSION AND RECOMMENDATIONS ………………………… 199

7.1 Conclusion ……………………………………………………... 199

7.2 Limitations of the Research ………………………………………. 203

7.3 Recommendations for Further Research ……………………………. 204

REFERENCES ……………………………………………………………… 206

APPENDICES …………………………………………………………… 234

Appendix A: Little's law …………………………………………… 235

Appendix B: PASTA property ………………………………………. 239

Appendix C: Mean value approach to derive the mean waiting time for

M/G/1 ………………………………………………………… 240

Appendix D: Benchmark Problems …………………………………. 244

List of Figures

FIGURE 2.1 USING RIGHT-SHIFT RESCHEDULING TO UPDATE A SCHEDULE ………………………………………………… 44

FIGURE 2.2 USING AFFECTED OPERATION RESCHEDULING TO UPDATE A SCHEDULE ………………………………….. 45

FIGURE 3.1 THREE MAJOR SCHEDULING FACTORS THAT CAN AFFECT THE PERFORMANCE OF RESCHEDULING …… 66

FIGURE 3.2 SOLUTION STABILITY VERSUS SOLUTION ROBUSTNESS ……………………………………………... 74

FIGURE 3.3 FINITE TIME INVOLVED IN PERFORMING SCHEDULING AND ITS EFFECT ON SOLUTION ADEQUACY ………………………………………………….. 77

FIGURE 3.4 COMPARISON OF SOLUTION QUALITY BETWEEN DIFFERENT SCHEDULING HORIZONS …………………... 79

FIGURE 3.5 WORST CASE SCENARIO OF RESCHEDULING INVOCATION ……………………………………………... 80

FIGURE 3.6 SIMULATION-BASED EXPERIMENTAL SETUP TO ASSESS SCHEDULING PERFORMANCE …………………. 81

FIGURE 3.7 A FEASIBLE SCHEDULE …………………………………… 89

FIGURE 3.8 A PERTURBED SCHEDULE ………………………………... 91

FIGURE 4.1 PERIODIC RESCHEDULING ……………………………….. 106

FIGURE 4.2 A SINGLE-ARRIVAL AND A GROUP-ARRIVAL QUEUING SYSTEMS ………………...................................... 109

FIGURE 4.3 NUMBER OF ENTITIES IN THE FIRST QUEUE OF GROUP-ARRIVAL QUEUING MODEL …………………..... 114

FIGURE 4.4 A TWO-MACHINE QUEUING MODEL ……………………. 120

FIGURE 4.5 UTILIZATION VERSUS *K* PLOT FOR DATA SET 1 ……. 126

FIGURE 4.6 UTILIZATION VERSUS *K* PLOT FOR DATA SET 2 ……. 126

FIGURE 4.7 UTILIZATION VERSUS *K* PLOT FOR DATA SET 3 ……. 127

FIGURE 4.8 UTILIZATION VERSUS *K* PLOT FOR DATA SET 4 ……. 127

FIGURE 5.1 ARCHITECTURE OF PRODUCTION PLANNING AND CONTROL FOR A TYPICAL DISCRETE MANUFACTURING SYSTEM ……………………………… 137

FIGURE 5.2 THE FIVE MODULES OF THE SIMULATION TEST BED 137

FIGURE 5.3 THE INTERACTIONS BETWEEN THE FIVE MODULES OF THE SIMULATION TEST BED …………………………. 139

FIGURE 5.4 THE STRUCTURAL VIEW OF THE SCHEDULING MODULE AND ITS INTERACTION WITH OTHER MODULES ……………………………………………………. 140

FIGURE 5.5 CORE CLASSES AND INTERACTIONS BETWEEN THEM 144

FIGURE 5.6 MAJOR CLASS INHERITANCE …………………………… 145

FIGURE 5.7 SUBCLASSES OF DISPATCHER CLASS ………………….. 148

FIGURE 5.8 RECURSIVE SIMULATION ………………………………… 152

FIGURE 5.9 SUBCLASSES OF SCHEDULER CLASS …………………... 154

FIGURE 6.1 TOTAL EXECUTION TIMES OF VARIOUS HEURISTICS AND PROCEDURES FOR THE 82 JOB SHOP PROBLEMS 164

FIGURE 6.2 PERCENTAGE IMPROVEMENT VERSUS EXECUTION TIME OF THE HEURISTICS AND PROCEDURES ……….. 165

FIGURE 6.3 MEAN CYCLE TIME VERSUS RESCHEDULING PERIOD FOR BREAKDOWN DURATION OF 30 ……....................... 175

FIGURE 6.4 MEAN CYCLE TIME VERSUS RESCHEDULING PERIOD FOR BREAKDOWN DURATION OF 600 ………………….. 176

FIGURE 6.5 MEAN CYCLE TIME VERSUS FACTORS FOR BREAKDOWN DURATION OF 30 …………………………. 177

FIGURE 6.6 MEAN CYCLE TIME VERSUS FACTORS FOR BREAKDOWN DURATION OF 600 ……………………….... 178

FIGURE 6.7 CYCLE TIME VARIANCE VERSUS FACTORS FOR BREAKDOWN DURATION OF 30 …………………………. 179

FIGURE 6.8 CYCLE TIME VARIANCE VERSUS FACTORS FOR BREAKDOWN DURATION OF 600 ……………………….... 179

FIGURE 6.9 MEAN CYCLE TIME VERSUS FACTORS FOR BREAKDOWN DURATION OF 30 …………………………. 181

FIGURE 6.10 TIME VARIANCE VERSUS FACTORS FOR BREAKDOWN DURATION OF 30 …………………………. 182

FIGURE 6.11 A SINGLE-MACHINE QUEUING MODEL ………………… 183

FIGURE 6.12 A SINGLE-MACHINE QUEUING MODEL WITH JOBS WAITING ……………………………………………………... 188

FIGURE 6.13 A NETWORK OF QUEUES ………………………………….. 190

FIGURE 6.14 A SIMPLIFIED NETWORK OF QUEUES …………………... 191

FIGURE A.1 ARRIVALS AND SERVICE COMPLETIONS AT A QUEUING SYSTEM …………………………………………. 236

FIGURE C.1 EVOLUTION OF UNFINISHED WORK IN THE SERVER 241

List of Tables

TABLE 2.1	SCHEDULING AND RESCHEDULING ENVIRONMENTS	17
TABLE 2.2	SCHEDULING AND RESCHEDULING STRATEGIES ……	19
TABLE 2.3	SUMMARY OF REACTIVE SCHEDULING STRATEGIES	20
TABLE 2.4	SCHEDULING AND RESCHEDULING METHODS ……….	22
TABLE 2.5	DIFFERENT APPROACHES TO SCHEDULING PROBLEMS …………………………………………………...	25
TABLE 2.6	CLASSIFICATIONS AND DESCRIPTIONS OF SELECTED PAPERS USING REACTIVE SCHEDULING FRAMEWORK ………………………………………………...	26
TABLE 2.7	SUMMARY OF SOME ROBUST SCHEDULE GENERATION METHODS ………………………………..	38
TABLE 2.8	SUMMARY OF SOME SCHEDULE UPDATE APPROACHES ………………………………………………..	41
TABLE 2.9	SUMMARY OF SELECTED PAPERS ON THE IMPACT OF VARIOUS SCHEDULING ASPECTS ON SYSTEM PERFORMANCE ……………………………………………...	53
TABLE 3.1	THE CONSOLIDATED CLASSIFICATION SCHEME OF DISRUPTIONS ………………………………………………..	68
TABLE 3.2	COMMON CHANGES AND DISTURBANCES …………….	69
TABLE 3.3	COMPARISON OF COMMON CHANGES AND	71

	DISTURBANCES IN INDUSTRY AND LITERATURE ……	
TABLE 3.4	MANAGEMENT STRATEGIES AND RELATED PERFORMANCE MEASURES FOR SCHEDULING ……….	83
TABLE 3.5	SCOPE OF RESCHEDULING PROBLEMS …………………	99
TABLE 4.1	DATA SETS FOR PLOTS …………………………………….	125
TABLE 6.1	RELATIVE PERFORMANCE OF MEAN CYCLE TIME FOR DISPATCHING RULES AND SCHEDULING HEURISTICS AS COMPARED TO FIFO ……………………	159
TABLE 6.2	RELATIVE PERFORMANCE OF MEAN CYCLE TIME FOR RECURSIVE HEURISTICS COMPARED TO THE CORRESPONDING DISPATCHING RULES ……………….	160
TABLE 6.3	RELATIVE PERFORMANCE OF MEAN CYCLE TIME FOR TOP RECURSIVE HEURISTICS AND TABU SEARCH AS COMPARED TO FIFO ………………………..	161
TABLE 6.4	THE EXPERIMENTAL PARAMETERS AND VALUES …..	168
TABLE 6.5	MEAN CYCLE TIME OF DIFFERENT SCHEDULING HEURISTICS FOR BREAKDOWN DURATION OF 30 …….	178
TABLE 6.6	CYCLE TIME VARIANCE OF DIFFERENT SCHEDULING HEURISTICS FOR BREAKDOWN DURATION OF 30 ……	179
TABLE 6.7	MEAN CYCLE TIME OF SELECTED SCHEDULING HEURISTICS FOR BREAKDOWN DURATION OF 30 ……	192
TABLE 6.8	MEAN CYCLE TIME OF SELECTED DISPATCHING RULES FOR BREAKDOWN DURATION OF 600 ………….	192
TABLE 6.9	CYCLE TIME VARIANCE OF SELECTED SCHEDULING HEURISTICS FOR BREAKDOWN DURATION OF 30 ……	193

TABLE 6.10 CYCLE TIME VARIANCE OF SELECTED DISPATCHING RULES FOR BREAKDOWN DURATION OF 600 …………. 193

List of Abbreviations

Abbreviations of System and Performance

Abbreviation	Full Name	Abbreviation	Full Name
CT	Cycle Time	**OTD**	On-time Delivery
DD	Due Date	**WIP**	Work In Process
OEE	Overall Equipment Effectiveness	**WL**	Work Load

Abbreviations of Scheduling and Optimization

Abbreviation	Full Name	Abbreviation	Full Name
AI	Artificial Intelligence	**MILP**	Mixed Integer Linear Programming
AOR	Affected Operation Rescheduling	**MODD**	Modified Operation Due Date
BB	Branch and Bound	**MOR**	Most Operation Remaining
CAF	Clear-a-Fraction	**MUR**	Match Up Rescheduling
CBR	Case Based Reasoning	**MWR**	Most Work Remaining
CBS	Constraint Based Scheduling	**NN**	Neural Network

CLB	Clear-the-Largest-Buffer-Level	**NS**	Nominal Schedule
CR	Critical Ratio	**OPT**	Optimized Production Technology
CTB	Control Theoretical Based	**RHSA**	Reactive Hierarchical Scheduling Approach
DAI	Distributed Artificial Intelligence	**RS**	Robust Schedule
DCA	Delay or Deferred Commitment Approach	**RSR**	Right Shift Rescheduling
EDD	Earliest Due Date	**SA**	Simulated Annealing
FIFO	First In First Out	**SBP**	Shifting Bottleneck Procedure
GA	Genetic Algorithm	**SPT**	Shortest Processing Time
KB	Knowledge based	**SRPT**	Shortest Remaining Processing Time
LP	Linear Programming	**TIS**	Time in System
LPRPT	Least Percent Remaining Process Time	**TS**	Tabu Search
LPT	Longest Processing Time	**WINQ**	Work in Next Queue

Other Abbreviations

Abbreviation	**Full Name**	**Abbreviation**	**Full Name**
FMS	Flexible Manufacturing System	**OOM**	Object-Oriented Modeling
FS	Flow Shop	**OOP**	Object-Oriented

			Programming
JS	Job Shop	**TBF**	Time Between Failure
MES	Manufacturing Execution System	**TTR**	Time To Repair
MFS	Modified Flow Shop		

Nomenclature

Author's Publications

Journal Articles

1. Chong, C. S., Lendermann, P., Gan, B. P., Duarte, B. M., Fowler, J. W., Callarman, T. E., "Development and Analysis of a Customer-Demand-Driven Semiconductor Supply Chain Model using the High Level Architecture", IJSPM special issue on Supply Chain Modeling and Simulation, 2005. To appear.

2. Sivakumar, A.I., Chong, C.S., "A Simulation Based Analysis of Cycle Time Distribution, and Throughput in Semiconductor Manufacturing", International Journal of Computers in Industry, Vol.45, pp. 59-78, Elsevier, 2001.

Conference Papers

3. Chong, C. S., Low, Y. H., Sivakumar, A. I., Robert Gay, "A Bee Colony Optimization Algorithm to Job Shop Scheduling", Winter Simulation Conference 2006, Accepted, 2006.

4. Chong, C. S., Sivakumar, A. I., Robert Gay, Low, Y. H., "Using Simulation Based Approach to Improve on the Mean Cycle Time Performance of Dispatching Rules", Winter Simulation Conference 2005, pp. 2194-2202, 2005.

5. Chong, C. S., Sivakumar, A. I., and Robert Gay, "Simulation based Scheduling Using a Two-pass Approach", Winter Simulation Conference 2004, pp. 1433-1439, 2004.

6. Chong, C. S., Sivakumar, A. I., and Robert Gay, 2002, "Simulation-based Scheduling for Dynamic Discrete Manufacturing", Winter Simulation Conference 2004, pp. 1465-1473, 2004.

7. Chong, C. S., Sivakumar, A. I., and Robert Gay, 2002, "Design, Development and Application of an Object Oriented Simulation Toolkit for Real-time Semiconductor Manufacturing Scheduling," Winter Simulation Conference 2002, pp. 1984-1956, 2002.

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Manufacturing environments can be broadly classified into discrete, repetitive or process manufacturing. Combinations of these environments often coexist in companies. Discrete and repetitive manufacturing produces distinct parts or products such as wafer, integrated-circuit packages, etc., whereas process manufacturing produces products that are fluid in nature such as liquid and gas. The characteristic that distinguishes discrete manufacturing is the focus on the individual requirements of customers. Advanced discrete manufacturing, such as wafer fabrication, is complex due to the large number of job-steps and machines involved and may require many hundreds of process steps performed by several hundred unique tool groups. The complexity is exacerbated by re-entrant processing and random disturbances such as machine breakdowns and random yield [Yan, 2000]. Modern discrete manufacturing systems are characterized by high levels of automation and integration among their various components [Bodner, 1994].

Discrete manufacturing today is facing more challenges than ever before due to increasingly competitive global economic market [Ferranti, 2002]. This globalization trend is forcing business to compete by a different set of rules. The ability to succeed in this demanding market environment is determined not only by the quality and cost of the products, but also the responsiveness to customer's changing requirements, with shorter product cycles, more rapid product and process development requirements [Bodner, 1994]. These criteria could only be satisfied by a lean manufacturing strategy emphasizing on high production quality, rapid adaptability to customized and small lot manufacturing with increasing variety of products while continually decreasing costs. Attaining these capabilities requires not only nimble and highly flexible organizations [Jackson, 1997], ongoing improvement of manufacturing processes, responsiveness to requirements and changes [Mittler, 1999], but also the development of effective planning and control systems that can plan, schedule, and execute operations strategy effectively [Corney, 2002]. Effective planning and control systems must quickly manage and compensate the manufacturing systems, caused by external and internal disturbances [Jackson, 1997].

Practical manufacturing environments are dynamic and stochastic in nature. New jobs constantly arrive into the systems, and unanticipated events such as machine breakdowns can occur. The information available to the decision-maker is both incomplete and uncertain. Thus, scheduling usually has to be done without a perfect knowledge of the near future. The unanticipated events or “disturbances” may cause the performance of production units to deviate [Stoop, 1996]. Often it is unclear how one decision will influence the satisfaction of conflicting organizational goals.

Shop floor scheduling tools have so far achieved limited penetration within the manufacturing industry, although scheduling problems have been studied for many decades [Little, 1995]. Considering scheduling problems as dynamic and stochastic is important, as it has been identified that one of the major causes that scheduling techniques do not work in practice is due to the differences between the generated schedules and the actual status of the shop floor [Stoop, 1996]. These differences are often caused by disturbances. This issue is not well addressed by commonly used periodic scheduling approaches. In these approaches, new schedules are only constructed periodically. Since the schedule generation process does not match the dynamics of the production facility, the human scheduler may have to update the schedule manually. In event-driven scheduling methods, schedules are updated on real-time events and new schedules can be generated to account for the new conditions [Wan, 1995]. Events are defined as any state changes or occurrences in the system such as the arrival of a job.

The current and emerging challenges of discrete manufacturing lead to the scheduling related issues that need to be addressed:

- The effects of uncertainties and unexpected events on manufacturing performance are unclear. The manufacturing performance is evaluated based on a set of desired performance measures such as mean cycle time, tardiness and so on. The different characteristics and conditions of manufacturing environments that can impact the performance are also not known.

- Shop floor scheduling problems may no longer be treated as static problems. They may become dynamic and stochastic problems, with deviations of actual execution from planned actions requiring calls to the scheduling function to resolve the differences and maintain solution quality.

- New requirements, such as solution stability in the face of uncertainties and unexpected events, will appear. The solution stability measures the number of changes that a solution undergoes during its execution. These requirements may need to be handled together with the requirement of producing schedules of the best possible intrinsic quality.

These issues necessitate us to look at the functions of scheduling in discrete manufacturing environments. The challenges provide us the motivation to study various scheduling techniques particularly event-driven scheduling techniques. These scheduling techniques can be further supported by the increasing availability of time sensitive data in advanced and modern manufacturing environments. Further, in the past literature [Farn, 1979], it was claimed that an ideal solution to scheduling problem of dynamic shop floors would be to reschedule as each new job arrived. However, so far very few comparative studies have been found on comparing event-driven scheduling to other rescheduling techniques such as periodic scheduling and dispatching. It is thus hoped that this research can subsequently contribute to effective scheduling techniques, which can then play a key-enabling role in improving the efficiency of modern production enterprises facing the international competition.

Our research work seeks to address the scheduling related issues in dynamic and stochastic discrete manufacturing environments. The work is motivated by the complex, dynamic and stochastic environments of discrete manufacturing particularly in semiconductor manufacturing.

1.2 Objectives

Manufacturing decisions vary with the length of time over which their consequences persist. Different planning horizons are therefore essential in the decision-making process. The planning horizons can generally be divided into long, intermediate or short term [Hopp, 1996]. In discrete manufacturing, production planning is usually considered as intermediate term while scheduling and dispatching are short term. Planning is the process of balancing materials and plant resources to best meet customer demand while achieving business goals. Scheduling is the process of precise sequencing of all material and plant resources, at operation level, over the near term time horizon to best meet customer demand. Dispatching is a localized decision about which job to process next every time a job is completed on each machine or machine center. Typically, the output of planning consists of material requirements in time, and these requirements are passed to scheduling system [Stoop, 1996]. We will focus primarily on short term detailed scheduling and dispatching.

Our main research objective is to explore and develop an effective scheduling approach, encompassing both scheduling strategy and heuristic, for dynamic and stochastic discrete manufacturing environments, and subsequently test and validate the approach using a simulation test bed. Throughout our research work we will take discrete manufacturing as the model and therefore use the concepts of this area. However, our conviction is that the results can be further extended to other application domains. To achieve the research objective, our work is divided into three distinct phases:

1) Study of the current research in event-driven scheduling for discrete manufacturing.

   This study should cover issues and approaches in scheduling and rescheduling for static, deterministic, dynamic or stochastic manufacturing environments. Although our focus is on the techniques in event-driven scheduling, it is equally important for us to review other scheduling techniques such as static scheduling as most of these techniques are applicable to event-driven scheduling.

2) Experimentation, evaluation and/or analysis of different rescheduling strategies and methods on a representative discrete manufacturing model that is subjected to different levels of disruptions.

   The difficulties faced in the process of scheduling within actual manufacturing environments can be eased by approaching the problem with a better understanding, and preserving solution options for producing quality results. Experimentation and evaluation of rescheduling is important to better understand the issues relating to the design and development of an effective rescheduling approach. Some of the aspects pertaining to event-driven scheduling that need to be taken into consideration in the evaluation and analysis are as followed:

   - The analysis of the arising new requirements of rescheduling such as robustness and stability.

   - The definition and optimization of criteria in rescheduling: solution quality, stability and adequacy.

- The strategies and methods which are able to deal with scheduling as a dynamic and stochastic problem.

- The characteristics of disruptions that may affect the rescheduling approaches such as the type of disruptions, the frequency disruptions, the duration of disruptions, and so on..

3) Derivation of an effective technique to address event-driven scheduling in discrete manufacturing environments.

   An effective event-driven scheduling technique is important in discrete manufacturing environments, which are dynamic and stochastic in nature. Improving scheduling effectiveness under uncertainty will improve cost performance of the system. An effective scheduling technique should be able to address dynamic and stochastic scheduling problems and adjust its focus of attention quickly and properly in response to the changing state of problem solving.

In summary, our primary research objective is to formulate an effective event-driven scheduling approach, encompassing both scheduling strategy and heuristic, for dynamic and stochastic discrete manufacturing environments.

1.3 Contribution

The main contributions of this research include:

1) Classification schemes for both the scheduling and rescheduling environments, strategies and methods (refer to Chapter 2). The classifications are an important

part to better understand the conditions or factors that may affect the manufacturing performance under production scheduling and rescheduling.

2) Identification and analysis of rescheduling factors and the tradeoffs between them (refer to Chapter 3). These factors are classified into three major categories: disturbance related, shop floor related and scheduling related factors. The analysis is important to determine and select the critical factors in rescheduling for further research.

3) One-machine and two-machine analytical models to demonstrate event-driven scheduling can achieve better performance than periodic scheduling in terms of mean waiting time of jobs (refer to Chapter 4). The models show that rescheduling strategy (or policy) can impact the performance of manufacturing systems.

4) An object-oriented simulation test bed to assess the performance of scheduling algorithms and dispatching rules for discrete manufacturing systems (refer to Chapter 5). The test bench is important to evaluate the various factors in rescheduling and identify their relationships.

5) A new scheduling algorithm based on the concept of recursive simulation (refer to Chapter 5). In this recursive simulation technique, simulated decision makers themselves use simulation to make decision. The nested simulation runs within a simulation are used to evaluate the possible outcomes given that a decision was made on one way or the other. Benchmarking results using static job shop problems indicate that the recursive algorithm is comparable in performance to other efficient procedures such as Tabu search. The algorithm is used in event-driven scheduling approaches for the purpose of experimentation.

6) An improved dispatching rule that outperforms other dispatching rules published in the past research in terms of mean cycle time performance across all different shop floor conditions in the simulation experiments (refer to Chapter 6).

7) Last but not least, an effective event-driven scheduling approach, which encompasses both the scheduling strategy and scheduling heuristic, for dynamic and stochastic discrete manufacturing environments (refer to Chapter 6).

1.4 Thesis Organization

This thesis consists of 7 chapters.

Chapter 2 presents an overall literature review in the area of event-driven scheduling. A classification scheme of event-driven scheduling is proposed.

Chapter 3 formulates and mathematically analyzes event-driven scheduling problems in detail.

Chapter 4 presents analytical models that compare the performance of periodic and event-driven scheduling strategies for discrete manufacturing systems, where different job types arrive dynamically for processing.

Chapter 5 discusses the design and development of a simulation test bed that facilitates the performance assessment of job dispatchers and schedulers for discrete manufacturing systems.

Chapter 6 presents a comparative study on the performance of scheduling algorithms and dispatching rules in dynamic job shop environments of discrete manufacturing systems using the designed test bed.

Chapter 7 concludes the research and proposes further works related to event-driven scheduling.

CHAPTER 2

THEORETICAL BACKGROUND AND RELATED WORK

2.1 Introduction

In the past few decades, a lot of effort in theoretical research has been observed in developing techniques or methods to generate optimal or near-optimal solutions for static, non-changing shop floor scheduling problems [Vieira, 2002; Raheja, 2002a]. This class of scheduling is termed under many different names: static, deterministic, off-line, generative, predictive scheduling, or simply the word "scheduling" on its own [Suresh, 1993; Matsuura, 1993; Pinedo, 1995]. In general, such effort formulates shop floor scheduling problems as combinatorial optimization problems, and the schedules are computed over a specific time frame assuming all problem characteristics are known in advance. Such scheduling effort focuses mainly on the construction of efficient schedules, without consideration of reconciling any discrepancies with actual progress on the shop floor. Even though these scheduling algorithms may be able to compute optimal schedules, in practice they rarely achieve optimality due to the dynamically changing environments in real-life problems [Wang, 1999].

Static shop floor scheduling problems are widely surveyed in the literature [Dorn, 1995a; Jones, 1998; Suh, 1998; Vieira, 2002]. Many techniques have also been proposed and implemented, and these are reviewed in several papers [Liu, 1996; Jones, 1998; Jain, 1999]. These techniques range from simple dispatching rules [Barman, 1997] to more sophisticated parallel Branch and Bound algorithms [Perregaard, 1998], and bottleneck based heuristics [Adams, 1988]. More recent techniques such as artificial intelligence and local search methods have also been proposed. There are also hybrid approaches, and some of these appear to perform better than pure approaches [Jain, 1999]. In general, algorithms exist to find the optimal solutions for some simple shop floor problems [French, 1982; Bauer, 1991]. However for more complex problems, the optimal algorithms tend to require excessive computation time. Where solutions cannot be found in reasonable time, heuristic techniques such as dispatching rules are often used.

Traditionally, shop floor scheduling problems are considered to be static and deterministic. Schedules are generated periodically and to be followed as closely as possible. However in reality, the scheduling problems are subject to many sources of uncertainty that cannot be neglected, and are difficult to model or taken into consideration when creating the schedules. Among the sources of uncertainty are machine breakdowns, quality problems, etc. These uncertainties can cause considerable differences between the schedules and its actual realization on the shop floor. These differences tend to degrade the quality of schedules as time passes [Parunak, 1991; Smith, 1995]. In practice, major deviations from the initial schedules are often “repaired” by the shop floor operators when they react to the disruptions by delaying the affected tasks or performing the tasks out of order. Such reactions are usually quick but myopic, and often fall prey to the complexity of interacting constraints.

The inefficiency to react to production disturbances makes static schedules difficult to change, and is considered as a limitation of the approaches. To resolve this issue, “reactive” scheduling approaches can be considered [Smith, 1994; Wan, 1995]. Reactive scheduling can react to disruptions in a timely manner, and recover from disruptions with minimal change effects while maintaining system performance of shop floor. In reactive scheduling, scheduling and rescheduling processes need to be addressed. Scheduling generates a production schedule for a given set of jobs on the resources, while rescheduling updates the production schedule in response to disruptions or changes [Vieira, 2002].

Reactive scheduling enables businesses to make faster and better decisions about how to commit resources. With business environment becoming increasingly challenging and competitive, achieving good production performance under the dynamic and stochastic scheduling environments can become the key to turning these challenges into competitive advantages. This is particularly true in most discrete manufacturing such as semiconductor manufacturing, which constantly faces the problems of costly assets, tradeoffs among decision factors, and narrow margins for error.

Basic types of research pertaining to reactive scheduling can be identified in the literature for examples, proposition of new approaches and study of reactive scheduling. In the former, new approaches (i.e. strategies and methods) for reactive scheduling are proposed. In the latter, the study, analysis and evaluation of the reactive scheduling problems and existing approaches are carried out.

Section 2.2 presents a framework for understanding the research in reactive scheduling. The research scope of reactive scheduling varies greatly, and thus appropriate classification schemes can help to understand not only the approaches but also the control strategy that can impact the system performance under various manufacturing environments.

Section 2.3 briefly reviews the performance criteria and measures for reactive scheduling. It also highlights the important performance measures for reactive scheduling, and other alternative measures.

Section 2.4 surveys various approaches in scheduling and rescheduling. A classification scheme that encompasses both scheduling and rescheduling is proposed. Methods for robust schedule generation and schedule update are also studied and compared in this section.

Section 2.5 reviews rescheduling strategy and shows how the rescheduling can impact the performance of manufacturing systems.

Section 2.6 summarizes the work done in this chapter, and consolidates the research work by the research community. It subsequently highlights some of the important issues that are yet to be addressed in rescheduling.

2.2 A Reactive Scheduling Framework

The scope of research in reactive scheduling ranges from rescheduling strategies to methods for resolving constraint violations under disruptions. In order to have a common ground to understand and compare the reactive scheduling research, a framework for both scheduling and rescheduling is proposed. This framework is adapted from previous work in the literature [Sabuncuoglu, 2000; Vieira, 2002]. The framework encompasses the environments, strategies, and methods relating to scheduling and rescheduling. This framework is discussed in the following subsections.

2.2.1 Scheduling and Rescheduling Environments

The scheduling and rescheduling environments are related to the various conditions of shop floors that may impact the performance of reactive scheduling approaches. The classification is extended from the previous work [Vieira, 2002], which divides the rescheduling environments into a single factor: static or dynamic. We classify the environments by three factors: shop floor layout (or flow pattern) and its workload conditions, job arrivals, and stochastic conditions (refer to Table 2.1). It is believed that all these factors can have an impact on the performance of reactive scheduling approaches.

Shop floor layout defines the way parts or products flow through manufacturing systems. The complexity of the scheduling problems depends on the flow pattern. The main layouts include single machine, parallel machines, flow shop, job shop and flexible manufacturing system (FMS). The single machine model [French, 1982;

Bauer, 1991; Pinedo, 1995] is the simplest layout. It consists of one machine that performs all operations, and jobs (or orders) usually contain single operations. The parallel machine model [French, 1982; Pinedo, 1995] has multiple, identical machines. A flow shop [French, 1982; Pinedo, 1995] consists of different machines and jobs with multiple operations, each of which is to be executed on a specific machine. All jobs have to visit the machines in the same sequence. A job shop is the similar to a flow shop, except that each job can visit the machines in a different sequence. Flexible manufacturing systems are extended job shops [Bongaerts, 1998a; 1998b], where the operations of jobs follow a sequence which is defined by a precedence graph instead of a fixed operation sequence, different workstations can perform the same operation, and transport and setup have to be modeled. There are many other variations to these layouts, and the details are described in scheduling literature [Pinedo, 1995].

The workload level describes the utilization level of a shop floor. The shop floor can be at high or low utilization level. The workload distribution specifies whether a system is being uniformly or non-uniformly loaded. In a non-uniformly loaded shop floor, some resources may be overloaded with jobs while others are underutilized. A recent study [Sabuncuoglu, 2000] shows the important effect of workload distribution on the system performance of shop floor.

Job arrival can be either static or dynamic [Vieira, 2002; Sabuncuoglu, 2000; Raheja, 2002a]. In a static shop floor environment, there is a finite set of jobs which are available for processing at time zero. In a dynamic environment, there is an infinite stream of jobs, which continue to arrive over an infinite time horizon [Church, 1992]

Practical shop floors are also subject to uncertainties due to disruptions or variations. Disruptions are unanticipated events that occur on the shop floor environments such as machine breakdowns, urgent jobs, and so on. Variations are deviations of the process parameters due to the fluctuation of process characteristics. The exact values of these parameters may not be known in advance, but their distributions are known. Examples include processing and repair time variation, and yield uncertainty. Depending on the absence and presence of these forms of randomness, the shop floor environments can be classified into either deterministic or stochastic environments respectively [Pinedo, 1995].

Table 2.1: Scheduling and rescheduling environments

Scheduling and rescheduling environments					
Shop floor		**Job arrival**		**Stochastic condition**	
Layout (or flow pattern)	**Workload condition**	**Static**	**Dynamic**	**Deterministic**	**Stochastic**
Single machine, parallel machine, flow shop, job shop, flexible manufacturing system	Workload level (or utilization level); workload distribution (bottleneck or uniform)	Finite set of jobs	Infinite set of jobs	All parameters are known in advance, and with no disruptions	Variability and uncertainty in parameters, and with disruptions

2.2.2 Rescheduling Strategies

There are two major strategies for controlling production in dynamic and stochastic manufacturing environments. These strategies are dynamic scheduling and predictive-reactive (or simply reactive) scheduling (refer to Table 2.2). For convenience, we will use the collective name of “scheduling heuristics” to mean the dispatching rules or scheduling algorithms used in both scheduling strategies.

In dynamic scheduling, decentralized production control methods dispatch the jobs whenever necessary. This dispatching normally uses localized information and makes use of heuristics such as dispatching rules to prioritize jobs that are waiting for processing at a resource [Church, 1992]. The computational effort of dispatching rules is usually low. In control theoretic or closed loop approach, rules can be developed for deciding which and what action to take in response to random disruptions [Gershwin, 1994; Qi, 2002]. The method will build a decision model based on existing shop floor conditions and select the most appropriate job to be processed next. The job selection is often based on the effectiveness of candidate dispatching rules. Since no firm schedule is generated in advance in dynamic scheduling, the effect of disruptions on the shop floor is minimal. The limitation of this dynamic approach is that it fails to provide any advance plan for support activities. Further, it is hard to predict system performance as decisions are made locally in real time.

In predictive-reactive scheduling strategy, there are two stages [Jain, 1997; Mehta, 1998]. The first stage generates initial schedules. The second stage updates the schedules in response to events. Some researchers propose approaches for generating robust schedules that may perform well even if disruptions occur.

A rescheduling policy is necessary to implement a predictive-reactive scheduling strategy. Three types of rescheduling policies can be identified: periodic, event-driven and hybrid [Vieira, 2002]. A periodic policy regenerates schedule periodically and implements the schedule on a rolling time horizon basis. This approach normally yields more schedule stability and less schedule nervousness than constant rescheduling. Unfortunately, following a predetermined schedule in the face of significant changes in the system conditions may compromise system performance of shop floor. Determining the optimal rescheduling period can also be a difficult task when using this type of policy [Vieira, 2002].

In an event-driven rescheduling policy, rescheduling may occur in response to disruptions or variations. This policy can be based on certain criteria to trigger rescheduling. For example, rescheduling can be triggered by the number of unscheduled jobs in a queue exceeding a specific quantity [Vieira, 2000a], or triggered by the difference between the planned and actual system performance [Kim, 1994]. In the extreme of this policy, a new schedule can be created (or revised) on each event. In this scenario, the time spent doing rescheduling can become excessive. Consider the case of events occurring in rapid succession, event-driven scheduler may be in a permanent state of rescheduling, with high nervousness (low stability) and excessive computational requirements.

A hybrid rescheduling policy reschedules the system periodically and also when major events take place. Major events are usually machine breakdowns, but can also be arrival of urgent jobs, job cancellation, etc. The hybrid policy attempts to combine the advantages of periodic and event-driven rescheduling approaches. In this policy, the decision to reschedule is based on the conditions of the system at the time of disruptions.

Table 2.2: Scheduling and rescheduling strategies

Scheduling and Rescheduling Strategies				
Dynamic		**Predictive-reactive (Generate and update)**		
		Scheduling and Rescheduling Policies		
Dispatching rules	Control theoretic approach	Periodic	Event-driven	Hybrid

A summary of reactive scheduling strategies, in terms of advantages, disadvantages and shop floor prerequisites for effective application, is presented in Table 2.3.

Table 2.3: Summary of reactive scheduling strategies

Strategy	Methods/ Policies	Advantages	Disadvantages	Shop floor prerequisites for effective application	References
Dynamic	Dispatching rules	• Simplest method • Easy to Implement • Low computational burden • Use localized information • Effect of disruption is minimal	• Poor quality of schedule • Highly problem dependent • No production schedule	• Job attributes (process time, due dates, etc.) • Current state of the machine	• Church, 1992 • Li, 1993 • Panwalkar, 1977 • Chen, 1995
	Control theoretic approach	• Better quality than dispatching rules • Can handle uncertainties, robust to unexpected events	• Response time is usually greater • No production schedule	• Job attributes (process time, due dates, etc.) • Current state of the machine	• Gershwin, 1994 • Qi, 2002
Predictive-reactive (generate and update)	Periodic	• Can optimize each run • Better schedule stability, less nervousness	• Can not respond to unexpected events • Computationally intensive	• Shop floor status	• Church, 1992 • Sivakumar, 2000a; 2000b • Vieira, 2002
	Event-driven	• Can continuously optimize • Handle unexpected events • Potential for better performance	• Computationally intensive	• Real-time shop floor status • Real-time events	• Jain, 1997
	Hybrid	• Can optimize • Handle major unexpected events	• Computationally intensive	• Real-time shop floor status • Major real-time events	• Vieira, 2002

2.2.3 Rescheduling Methods

In predictive-reactive scheduling, initial schedules are generated and subsequently updated in response to events. The initial schedule can be either nominal or robust (refer to Table 2.4). The methods that create nominal scheduling are those relating to static, deterministic scheduling. Generating robust schedules is an attempt to maintain good schedule performance with simple adjustments at the subsequent update stage, so as to minimize the impact of disruptions on the schedules. It differs from the nominal scheduling approach as it tries to ensure that the initial schedules and realized schedules do not differ drastically, while maintaining a high level of schedule performance. In certain cases, a robust but partial schedule can be generated first, and the incomplete portions are resolved at later time [Wu, 1999]. This technique gives the shop floor some flexibility to handle disruptions. For example, an incomplete schedule may contain only the sequence of jobs to process on each machine, but not the start and end times of the jobs on the machine.

There are two major methods to update a schedule that is no longer feasible due to a disruption: complete and partial rescheduling. Complete rescheduling regenerates the schedule for the entire set of operations (of jobs) that are not yet processed before the rescheduling point, including those not affected by the disruption [Vieira, 2000a; 2000b; Yamamoto, 1985]. A rescheduling point is the point in time when a scheduling decision is made [Vieira, 2002]. The main disadvantage of this approach is the excessive computational effort and possibly unsatisfactory response time [Wu, 1995].

Partial rescheduling can often be used to improve schedule stability and reduce schedule nervousness. A good partial rescheduling method is one that leads to minimum deviation of the performance measures while incorporating the necessary modifications and maintaining schedule stability. Since rules and constraints are involved when schedules are adjusted, schedule repair can often be viewed as a constrained scheduling problem. The objective of the problem is to minimize the deviation in the performance during the repair while satisfying the constraints.

Table 2.4: Scheduling and rescheduling methods

Scheduling and Rescheduling Methods

Schedule Generation		Schedule Update	
Nominal schedule	Robust schedule	Partial rescheduling	Complete rescheduling

2.3 Rescheduling Performance Measures

Performance metrics are important to benchmark different rescheduling approaches, and to guide rescheduling process. The performance measures for rescheduling can be classified into two categories: schedule quality and schedule stability [Vieira, 2002; Jain, 1997]. Schedule quality measures how efficient a schedule achieves a desired objective function such as makespan and tardiness. Stability measures the degree of deviation between the revised schedule and the initial schedule.

Measures of schedule quality are applicable to both scheduling and rescheduling. These are generally time-based measures [Pinedo, 1995; Shafaei, 1999a; 1999b]. Some of the common measures are: makespan [Yamamoto, 1985; Wu, 1993; Mehta, 1998], mean tardiness [Kim, 1994; Jain, 1997], mean cycle time or flow-time [Vieira, 2000a; 2000b], average resource utilization [Jain, 1997], and maximum lateness [Church, 1992].

Schedule stability is only applicable to rescheduling. It can measure changes in a schedule in two ways [Wu, 1993; Abumaizar, 1997]. The first method measures the deviations of operation starting times between the initial schedule and revised schedule. The second way measures the difference in the operation sequences on the resources between the two schedules. Some variations of the measures exist. For example, Cowling and Johansson [2001] measures the schedule stability by considering the average absolute change of the start and completion times of operations of the initial and revised schedules.

In some cases, time-based performance measures may not completely reflect the economic performance of manufacturing systems. To solve this issue, cost-based performance measures can be used. Examples are job processing cost, work-in-process cost, and penalty cost of missed due dates, etc. A total cost function can then be derived, and the objective then is to minimize the total cost. In rescheduling, the costs can occur in three categories: computational costs, setup costs and transportation costs [Vieira, 2002]. Computational costs are related to the costs of running scheduling systems. Setup costs are due to changes in the schedule, resulting in the reallocation of tooling and fixtures. Transportation costs are related to delivering materials earlier than required with additional cost incurred when transporting jobs from one scheduled machine to other points in the shop floor. An example of cost-based performance measure is used by Bean et al., [1991]. In their paper, the solution cost, which is based on the number of jobs reassigned, is required to be balanced against tardiness costs and computational effort.

2.4 Scheduling and Rescheduling Approaches

Different perspectives of scheduling environments often necessitate different solution approaches. The deterministic, static perspective requires static scheduling techniques. Scheduling environments that are dynamic and stochastic require dynamic, reactive or proactive scheduling techniques. The dynamic scheduling approaches are by far the most commonly used techniques. The reactive scheduling methods react to disturbances and subsequently recover from the disruptions. The proactive scheduling approaches take into account the risk of disturbances in its schedule, and then react to the disruptions when they actually occur.

Majority of dynamic and stochastic scheduling problems in the literature are single machine [Farn, 1979; Church, 1992; Ovacik, 1994a], flow shop [Vieira, 2000a; 2000b], job shop [Yamamoto, 1985; Wu, 1995; Sadeh, 1994; Herrmann, 1995] or flexible manufacturing system [Wu, 1989; Dutta, 1990; Kim, 1994]. Most of these problems consider new job arrivals as dynamic environments [Farn, 1979; Wu, 1989; Church, 1994a; Ovacik, 1994b] and machine breakdowns or tool unavailability as stochastic conditions [Yamamoto, 1985; Dutta, 1990; Nof, 1991; Bean, 1991].

A summary of the techniques under different aspects of scheduling environments is shown in Table 2.5. Some of the selected work from the past research is classified using the scheduling and rescheduling framework is shown in Table 2.6. This research is discussed in the following subsections. Section 2.4.1 examines scheduling approaches while Section 2.4.2 describes rescheduling approaches. It is important for us to study scheduling approaches as most rescheduling approaches are adapted from their static scheduling counterparts. Further, static scheduling methods can also be used as the first stage of the predictive-reactive scheduling to generate initial schedules, or even the revised schedules in industry. Hence, to better understand the rescheduling approaches, it is thus necessary for us to selectively study the static scheduling approaches. However, the emphasis will be on the principles of the approaches.

Table 2.5: Different approaches to scheduling problems

<table>
<tr><th>Scheduling Environments (job arrival, stochastic condition)</th><th>Approaches</th><th>Methods or Algorithms</th><th>References</th></tr>
<tr><td>(static, deterministic)</td><td>Static, deterministic scheduling</td><td>• Optimization
♦ Enumerative methods
▪ Mathematical Formulation (Lagrangian Relaxation, Decomposition Methods, Integer Linear Programming, Mixed Integer Linear Programming, Surrogate Duality)
▪ Branch & Bound
• Approximation
♦ General Algorithms (Iterative methods)
▪ Artificial Intelligence (Neural Network, Expert Systems, Constraint Satisfaction, Ant Colonization, Fuzzy Logic, Distributed Artificial Intelligence)
▪ Local Search (Evolutionary Computation, Genetic Algorithms, Reinsertion Algorithms, Tabu Search, Threshold Algorithms such as Simulated Annealing and Iterative Improvement, Large Step Optimization, etc.)
♦ Tailored Algorithms (Constructive methods)
▪ Bottleneck Based Heuristics (Shifting Bottleneck Procedure, Macro and Micro Opportunistic)
▪ Insertion Algorithms (Beam Search, Filtered Beam Search)
▪ Priority Dispatch Rules (SPT, MWR, EDD, etc.)</td><td>• Pinedo, 1995
• Jones, 1998
• Jain, 1998
• Jain, 1999</td></tr>
<tr><td>(static, stochastic), (dynamic, deterministic) or (dynamic, stochastic)</td><td>Dynamic, reactive or proactive scheduling</td><td>Dynamic (i.e. no schedule generation)
• Dispatching rules
• Control theoretic approach

Predictive-reactive scheduling
1. Schedule generation
▪ Nominal Schedule (Same as static or deterministic scheduling approaches)
▪ Robust Schedule (Genetic Algorithm, Branch & Bound, Shifting Bottleneck, Fuzzy Logic, etc.)
2. Schedule update (or repair)
▪ Tailored Algorithms (Right Shift Rescheduling, Affected Operation Rescheduling, Match-up Scheduling, Micro opportunistic, Dispatching rules, etc.)
▪ General Algorithms (Case-based reasoning, Constraint-based, Fuzzy Logic, Neural Network, etc.)</td><td>• Panwalkar, 1977
• Gershwin, 1994
• Hoogeveen, 2000
• Naroska, 2002
• Jones, 1998
• Raheja, 2002a
• Vieira, 2002</td></tr>
</table>

Table 2.6: Classifications and descriptions of selected papers using reactive scheduling framework

References	**Environments** 1. Shop floor 2. Job arrival 3. Stochastic condition	**Strategies** 1. Strategy 2. Policy	**Methods** 1. Schedule generation 2. Schedule update	**Objective functions**
Farn, 1979	1. Single machine 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Periodic	1.Nominal schedule (dispatching rules & heuristics based on traveling salesman problem) 2. Complete rescheduling	Changeover time
Yamamoto, 1985	1. Job shop 2. Static 3. Machine breakdowns	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Branch & Bound) 2. Complete rescheduling	Makespan
Wu, 1989	1. Flexible manufacturing system 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Periodic	1. Nominal schedule (simulation based dispatching rules) 2. Partial rescheduling (partial simulation window)	Mean tardiness, mean flow time
Perkins, 1989	1. Job shop 2. Semi-dynamic[1] 3. Process time variations	1. Dynamic 2. Event-driven	1. Clear-a-Fraction (CAF) policies 2. Dispatching	Buffer levels
Dutta, 1990	1. Flexible manufacturing system 2. Static 3. Machine breakdowns, new jobs, job priority changes	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (knowledge-based) 2. Partial rescheduling (rerouting, preemption, etc.)	Mean completion time, mean machine utilization
Bean, 1991	1. Parallel machine 2. Static 3. Machine breakdowns, unavailable tools	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (predetermined schedule is assumed) 2. Partial rescheduling (match-up scheduling approach)	Weighted total tardiness
Church, 1992	1. Single machine 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Hybrid	1. Nominal schedule (earliest due date) 2. Complete rescheduling	Maximum lateness
Matsuura, 1993	1. Job shop 2. Semi-dynamic[1] 3. Machine breakdowns, specification changes, rush jobs	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Branch & Bound) 2. First come first serve, shortest processing time for job selection after first disruption	Makespan
Ovacik, 1994a	1. Single machine 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (rolling horizon algorithm based on Branch & Bound) 2. Partial rescheduling (rolling horizon, after scheduling *n* jobs)	Maximum lateness
Kim, 1994	1. Flexible manufacturing system 2. Semi-dynamic[1] 3. Machine breakdowns, urgent jobs	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (best of the dispatching rules in simulation) 2. Complete rescheduling	Mean Flow Time, mean tardiness, combination
Leon, 1994	1. Job shop 2. Static 3. Process time variations	1. Predictive-reactive 2. Event-driven	1. Robust schedule (Genetic algorithms) 2. Partial rescheduling (Right-shift)	Makespan
Sadeh, 1994	1. Job shop 2. Static 3. Machine breakdowns, new job arrivals, process time variations	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Micro-opportunistic) 2. Partial rescheduling (Micro-opportunistic)	Maximum tardiness cost, inventory cost

References	Environments 1. Shop floor 2. Job arrival 3. Stochastic condition	Strategies 1. Strategy 2. Policy	Methods 1. Schedule generation 2. Schedule update	Objective functions
Wu, 1995	1. Job shop 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (predetermined schedule is assumed) 2. Partial rescheduling (scheduling graph is constructed from nominal schedule, then algorithms for time and relationship effects can be used for rescheduling)	Makespan
Dorn, 1995b	1. Job shop 2. Static 3. Process time variations	1. Predictive-reactive 2. Event-driven	1. Robust schedule (constructive heuristics using fuzzy reasoning) 2. Partial rescheduling (Tabu search by progressively interchange jobs)	Job criticality, robustness
Herrmann, 1995	1. Job shop 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Genetic Algorithm) 2. Complete rescheduling	Number of tardy jobs
Sabuncuoglu, 1997; 1999	1. Flexible manufacturing system 2. Static 3. Machine breakdown, process time variations	1. Predictive-reactive 2. Periodic	1. Nominal schedule (beam search & dispatching rules) 2. Compete rescheduling	Mean tardiness, makespan
Byeon, 1998	1. Job shop 2. Static 3. Process time variations	1. Predictive-reactive 2. Event-driven	1. Robust schedule (decomposition method with Branch & Bound) 2. Partial rescheduling (dynamic dispatching method)	Weighted tardiness, robustness
Lawrence, 1997	1. Job shop 2. Static 3. Processing time variations	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (optimum using Branch & Bound, non-optimum using shifting bottleneck and heuristic methods) 2. Complete rescheduling (shifting bottleneck algorithm or dispatch heuristics)	Makespan
Jain, 1997	1. Flexible manufacturing system 2. Static 3. Machine breakdown, increased order priority, rush orders arrival, order cancellation	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Genetic Algorithm) 2. Partial rescheduling (specific heuristic rules to handle each of the unexpected events)	Predictive (mean flow time, mean tardiness, average resource utilization), reactive (stability & response time)
Abumaizar, 1997	1. Job shop 2. Static 3. Machine breakdown	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (adaptive heuristic algorithm) 2. Complete and partial rescheduling (affected operation rescheduling based on binary branching algorithm)	Predictive (makespan), reactive (stability)
Mehta, 1998	1. Job shop 2. Dynamic 3. Machine breakdown	1. Predictive-reactive 2. Periodic	1. Robust schedule (shifting bottleneck to form operation sequence and then idle time insertion using constructive heuristics) 2. Partial rescheduling (right-shift rescheduling)	Maximum lateness

References	Environments 1. Shop floor 2. Job arrival 3. Stochastic condition	Strategies 1. Strategy 2. Policy	Methods 1. Schedule generation 2. Schedule update	Objective functions
Akturk, 1999	1. Modified flow shop[2] 2. Static 3. Machine breakdown	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (predetermine schedule is assumed) 2. Partial rescheduling (match-up point with reactive hierarchical scheduling approach)	Tardiness, earliness, stability
Bierwirth, 1999	1. Job shop 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Event-driven	1. Nominal schedule (Genetic Algorithm) 2. Complete rescheduling (population modified from previous Genetic algorithm run)	Mean flow time
Sabuncuoglu, 2000	1. Job shop with workload distribution 2. Static and dynamic 3. Machine breakdown	1. Predictive-reactive & dynamic 2. Periodic with variable interval	1. Nominal schedule (filtered beam search) 2. Partial rescheduling (heuristics)	Mean tardiness, makespan
Vieira, 2000a	1. Single machine 2. Dynamic 3. Deterministic	1. Predictive-reactive 2. Periodic & event-driven	1. First in first out and group jobs with same setup 2. Right-shift	Mean flow time, machine utilization

Note:
[1]Semi-dynamic – dynamic jobs arrival but with a priori known ready times
[2]Modified flow shop – simplified job shop with all jobs follow the same routing

2.4.1 Scheduling Approaches

Two broad classes of methods can be identified to find optimal (or near-optimal) solution for scheduling problems: optimization and approximation methods. Optimization methods, which are directed at finding optimal solutions by the application of enumerative algorithms, are computationally inefficient, and are therefore of limited practical use for rescheduling. One of the enumerative techniques is Branch and Bound (BB) [Perregaard, 1998]. In this approach, a dynamically constructed tree representing the solution space of all feasible schedules is implicitly searched. This technique formulates procedures and rules to allow large portions of the tree to be removed from the search based on lower and upper bounds. However, its excessive computing requirement prohibits its application to large shop floor problems.

An integer linear programming is basically a linear programming (LP) with the additional requirement that the variables have to be integers [Pinedo, 1995]. If only a subset of the variables is required to be integers and the remaining ones are allowed to be real, the method is referred to as a mixed integer programming. Lagrangian relaxation solves integer programming problems by omitting specific integer valued constraints and adding the corresponding costs (due to these omissions and/or relaxations) to the objective function. Lagrangian relaxation is computationally expensive for large shop floor scheduling problems.

Due to the limitation of the exact enumerative techniques, approximation methods become viable alternatives. In these approaches, optimal solutions are not guaranteed but the gain in speed enables larger problems to be solved. The earliest approximation algorithms are priority dispatch rules [Panwalkar, 1977]. These rules assign priorities to all the operations that are available to be sequenced and then choose the operation with the highest priority. Over the last 30 years, the performance of a large number of the priority dispatch rules has been studied extensively using simulation techniques [Jones, 1998]. These rules are normally easy to implement and have a low computation burden. However, the rules are myopic, highly problem dependent.

Beam search is a search algorithm based on the idea of Branch and Bound (BB) [Pinedo, 1995]. Beam search is a breadth first search, since nodes are evaluated at each level before going deeper in the search tree. The success of a beam search depends on the evaluation function that is employed at each node. The evaluation function is used to evaluate all nodes at a tree level and then select the most promising nodes to explore further. In filtered beam search, not all nodes at a given level are evaluated. Only the most promising nodes at level k are selected as nodes to branch from. The remaining nodes at that level are discarded permanently. The evaluation process that determines which nodes are the promising ones is a crucial component of this approach. There is a computation time compromise between crude prediction and thorough evaluation. Normally, to obtain the best result, for all the nodes generated at level *k*, a crude prediction is done. Based on the outcome of these crude predictions, a number of nodes are selected for a thorough evaluation, and the remaining nodes are discarded permanently. Based on the outcome of the careful evaluation, again a subset of these nodes is selected, from which further branches will be generated.

An example of bottleneck based scheduling techniques is optimized production technology (OPT). In this approach, bottlenecks are scheduled with operations first to optimize the throughput of the plant. The rest of the operations are then scheduled so as to reduce inventory. In more sophisticated bottleneck procedures such as OPIS [Smith, 1995], new bottlenecks are recognized during the construction of a schedule. In this approach, two scheduling perspectives are used: resource-centered and job-centered perspectives. The resource-centered perspective is for scheduling bottleneck resources whereas the job-centered perspective is to schedule non-bottleneck operations on a job-by-job basis. Instead of relying on initial bottleneck analysis, OPIS repeats the bottleneck analysis each time a resource or a job has been scheduled. The author termed this technique of detecting new bottlenecks and revising scheduling strategy during the scheduling process as “opportunistic scheduling”. However, the opportunism in this approach remains limited as it typically requires scheduling an entire bottleneck (or at least a large chunk of it) before switching to another bottleneck.

The Shifting Bottleneck Procedure (SBP) is a constructive algorithm that is characterized by the following tasks: sub-problem identification, bottlenecks selection, sub-problem solution, and re-optimization [Demirkol, 1997]. SBP develops a schedule for one machine at a time. The actual strategy involves relaxing the problem into *m* one-machine problems and solving each sub-problem one at a time. Each one-machine solution is compared with all the others and the machines are ranked on the basis of their solutions. The machine having the largest lower bound is identified as the bottleneck machine. SBP sequences the bottleneck machine first, with the remaining not sequenced machines ignored and the machines already scheduled held fixed. Every time the bottleneck machine is scheduled each previously sequenced machine susceptible to improvement is locally re-optimized by solving the one machine problem again. The fundamental problem with SBP is the difficulty in performing re-optimization and the generation of infeasible solutions.

Neural Network (NN) [Satake, 1994] is an attempt to mirror the learning and prediction abilities of human beings. NN is a network of many simple processors or "units", each possibly having a small amount of local memory. The units are connected by communication channels which usually carry numeric data. The units operate only on their local data and on the inputs they receive via the connections. Neural network models are distinguished by interconnected group of units, unit characteristics, and training or learning rules. In supervised learning neural networks for example, neural networks attempt to capture the desired relationships between inputs and the outputs through exposure to learning patterns. Back-propagation is widely used in supervised training procedure. Back-propagation applies the gradient-descent technique in the feed-forward network to change a collection of weights so that some cost function can be minimized. The cost function is dependent on weights and training patterns. After the network propagates the input values to the output layer the error between the desired output and actual output will be “back-propagated” to the previous layer.

Constraint satisfaction techniques [Nuijten, 1998] are iterative methods. The techniques aim at reducing the effective size of the search space by applying constraints that restrict the order in which variables are selected and the sequence in which possible values are assigned to each variables. An important concept in constraint satisfaction techniques is constraint propagation [Pinedo, 1995]. Constraint propagation works by systematically reducing all variable domains. Each constraint has an associated set of variables that are affected by the constraint. Each variable belongs to the associated set of at least one constraint. When the domain of that variable is altered, the associated set of variables for each of these constraints is affected. The change to their domains in turn affects other variables via other constraints. The effect of a change to one variable is propagated via constraints throughout all the decision variables. Although these techniques are concerned with trying to achieve a feasible schedule such that all the constraints can be met, many of these methods have difficulty representing constraints and require excessive backtracking.

Genetic Algorithm [Davis, 1985; Kobayashi, 1995; Jain, 1997; Khoo, 2000] are search techniques based on an abstract model of natural evolution such that the quality of individuals builds to the highest level of compatible with the environments. In Genetic algorithm (GA), a population of solutions exists at each cycle (or iteration) of the algorithm. This population is said to evolve into a new population at the next iteration. The evolution is achieved by using specific operators: reproduction, crossover and mutation. Reproduction copies a solution from the old population to the new population with a probability depending on the fitness of the solution. The fitness of a solution is the value of the objective function for that solution. In general, the better the value of objective function, the higher the probability that solution will survive to the next iteration. Crossover swaps a section of two solutions. Each new solution contains part of the two old solutions. This operator attempts to create solutions that have the best properties of the original solutions. Mutation varies a solution by randomly changing a small part of it. This step is necessary to avoid GA from being stuck in a local optimum, and allows the algorithms to jump to new solutions. Despite the fact that many elaborate schemes have been proposed, GA is unable to successfully represent complex shop floor problems and crossover operators cannot generate feasible schedules without losing their efficiency.

Threshold algorithm is an iterative method that chooses a new configuration if the difference in cost between the current solution and a neighbor is below a given threshold. A popular threshold algorithm is simulated annealing [Vakharia, 1990; Jeffcoat, 1993; Satake, 1999]. Simulated annealing (SA) is a random oriented local search technique that was introduced as an analogy from statistical physics of the computer simulation of the annealing process of a hot metal until its minimum energy state is reached. By initially allowing a high level of movement throughout the solution space, SA attempts to search the entire space, providing global optimization. Later in the process, the cooling allows only small movements in the solution space, and the process converges on a final solution. As SA is a generic technique it is seldom able to achieve good solutions, and it may require excessive computational effort.

Tabu search [Laguna, 1991; Armentano, 2000] is a local search meta-heuristic that relies on specialized memory structures to avoid entrapment in local minima and achieve an effective balance of intensification and diversification. It guides a search process away from solutions that appear to duplicate or resemble previously achieved solutions. A move from one schedule to another schedule is made by evaluating all candidates and choosing the best available. Some moves are classified as tabu (i.e. forbidden) because they either trap the search at local optimums, or they lead to cycling (repeating part of the search). These moves are put onto the Tabu list, which is built up from the history of moves used during the search. These tabu moves force exploration of the search space until the old solution area is left behind. More elaborate schemes can be applied to intensify the search in areas that have historically been good or diversify the search to unexplored regions of the solution space.

Distributed Artificial Intelligence (DAI) [Jain, 1999] is based on “divide and conquer” principle. This approach requires a problem decomposition technique, and the development of different expert or knowledge-based systems that can cooperate to solve the overall scheduling problem. DAI is usually performed by using agent paradigm. An agent is a unique software process operating asynchronously with other agents. Agents are complete knowledge-based system by themselves. The set of agents in a system may be heterogeneous with respect to long-term knowledge, solution-evaluation criteria, or goals, as well as languages, algorithms and hardware requirements.

The literature on static, deterministic scheduling indicates that the better approaches to shop floor scheduling problems are of hybrid construction by integrating problem specific local heuristics with general meta-solvers [Jain, 1999]. Further, the selection of a scheduling method often depends on tradeoffs between computation time and solution quality.

2.4.2 Rescheduling Approaches

We review the research in the dynamic scheduling approaches in Section 2.4.2.1 and the details on reactive and proactive scheduling approaches in Section 2.4.2.2.

2.4.2.1 Dynamic Approaches

The dynamic scheduling approaches include both the dispatching rules and control theoretic approach. These approaches are usually studied off-line based on the past historical characteristics of shop floors using either analytical (e.g. queuing network) or simulation models (e.g. discrete event simulation). Simulation is the most widely used technique to study the approaches [Suresh, 1993; Sivakumar, 2000a; 2000b]. The dynamic and stochastic shop floor environments can be modeled by using distribution functions (such as Poisson distribution, Exponential distribution, etc). Through simulation evaluation and analysis, the methods can be evaluated before deploying onto the actual shop floors.

An extensive list of dispatching rules can be found in the past literature. These rules range from simple dispatching rules to combinations of rules [Panwalkar, 1977; Blackstone, 1982]. Muhlemann et al. [1982] studied the performance of a number of dispatching rules under different scheduling frequencies (e.g. daily). The results showed that it is difficult to come up with a consistent rule that performs well under a variety of shop floor configurations and operating conditions. Kumar and others

[Perkins, 1989; Kumar, 1993; 1994] studied the control of dynamic manufacturing systems. They described classes of dispatching rules that decide which job a resource should process next. In the paper by Perkins and Kumar [1989] for example, they use Clear-a-Fraction (CAF) scheduling policies that dispatch jobs at the machines of an FMS. One of the policies is the Clear-the-Largest-Buffer-Level (CLB) policy that chooses part, which has the largest current buffer level.

Hoogeveen et al. [2000] studied the scheduling issue of maximizing the number of early jobs on a single machine with preemption-restart model. The shortest remaining processing time (SRPT) algorithm is used to construct a schedule of early jobs only, since any tardy jobs can be appended to this schedule in an arbitrary order. A job is preempted only if a newly released job can be completed earlier. Chen et al. [1995] investigated the problem of on-line scheduling of jobs on *n* identical machines where preemption is allowed, for makespan performance using an approximate algorithm. A recent paper by Naroska and Schwiegelshohn [2002] addressed the non-preemptive on-line scheduling of parallel jobs using list scheduling algorithm. In the analytical approaches, the mathematical models developed can be highly coupled, non-linear, and simplifying assumptions are often required to make the equations more tractable.

2.4.2.2 Reactive and Proactive Scheduling Approaches

Both the reactive and proactive [Szelke, 1995] scheduling approaches can be classified under predictive-reactive scheduling. The reactive scheduling approaches respond to events, whereas proactive approaches aim at preventing foreseeable disturbances, by considering them in the schedule as soon as they are known, and before they can invalidate the schedule. In this approach, the manufacturing performance trends are monitored continuously and statistically processed in real time to better predict the future evolution of the system.

Predictive-reactive scheduling is a commonly researched strategy in rescheduling for dynamic and stochastic manufacturing systems [Leon, 1994; Jain, 1997; Mehta, 1998; Suh, 1998; Sun, 2001; Raheja 2002a]. The key stages in this approach are schedule generation and update [Sabuncuoglu, 2000; Church, 1992]. Wu and Li [1995] described predictive-reactive scheduling as an iterative process of three steps: evaluation, generation and revision. The evaluation step evaluates the impact that a disruption causes. No further action is required if the impact is acceptably small. The generation step determines the rescheduling solutions that can enhance the performance of the initial schedule. The revision step updates the initial schedule or generates a new one. If the result is unacceptable, the generation step is revisited.

Schedule generation methods include most of the literature in the area of static, deterministic scheduling and the commonly used methods are discussed in Section 2.4.1. This section concentrates on methods that generate robust schedules and methods to update schedules in response to events.

2.4.2.2.1 Generating Robust Schedules

Generating robust schedules is an attempt to maintain scheduling performance with simple schedule adjustments [Byeon, 1998; Wu, 1999]. Simple adjustments ensure the rescheduling process can react to changes in a timely manner. In general, robust scheduling techniques reduce the risk of disturbances and preserves flexibility. A summary of robust schedule generation methods is presented in Table 2.7.

Table 2.7: Summary of some robust schedule generation methods

Categories	Techniques	Approaches	Advantages	Disadvantages	Performance measures	Ref.
Partial schedule	Deferred commitment	Decomposition method and solve a variant of generalized assignment problem. Then Branch and Bound on disjunctive graph to form partial schedule	Flexibility to handle disruptions. Global view of the system. Good robustness for medium levels of disturbances	Computational intensive. Incomplete schedule for other support activities. Degraded schedule quality	Weighted tardiness, robustness	Byeon, 1998; Wu, 1999
Complete schedule	Idle time insertion	Shifting bottleneck to form operation sequence and then idle time insertion using constructive heuristics	Flexibility to handle disruptions. Global view of the system. Not much degradation of system performance	Computational intensive. Degraded schedule quality due to time slack. Wastage of machine capacity if disturbances do not occur	Maximum lateness	Mehta, 1998
	Avoid early idle times	Rolling time horizon using evolutionary algorithm (i.e. GA) that penalizes early idle times in the active schedules	Flexibility to handle disruptions. Avoid early idle times. Global view of the system. Better solution quality than priority dispatch rules	Very computational intensive. Idle times must exist in the schedule. Degraded schedule quality due to time slack	Tardiness	Branke, 2002
	Fuzzy set theory	Constructive heuristics to generate an initial schedule. The crisp processing times replaced by fuzzy possibility distribution	Flexibility to handle disruptions. Allow incorporation of robustness in the schedule evaluation. Global view of the system	Computational intensive. Degraded schedule quality due to time slack	Job criticality, robustness	Dorn, 1995a

There are a few approaches to robust schedule generation. One approach is delayed (or deferred) commitment scheduling which is based on the idea that decisions are taken only if it can not be avoided any further [Parunak, 1991; Berry, 1993a]. A number of papers have proposed methods for creating such partial schedules that are robust with respect to disruptions. Byeon et al. [1998] and Wu et al. [1999] presented approaches to create robust partial schedules for a job shop that is subject to disturbances. Byeon et al. [1998] decomposed a job shop scheduling problem and solved a variant of the generalized assignment problem. Wu et al. [1999] used a Branch and Bound algorithm to process the corresponding disjunctive graph and formed a partial schedule. The incomplete portions of the schedule are resolved using dispatching methods at the later appropriate time, giving the shop some flexibility to handle disruptions. Their results showed that, in a range of situations, such a schedule leads to better shop performance than dispatching rules alone. However, as the amount of processing time variability increases, dispatching rules achieve better performance.

Another approach is temporal abstractions, which are based on the idea that reasoning behavior vary with the temporal interval [Berry, 1993a]. This idea is based on the fact that disruptions to a schedule should be handled differently according to the temporal interval they influence. A practical scenario to partial schedules is the distribution of scheduling decisions throughout the control hierarchy of a factory. The detailed decisions are only made as refinements of higher level decisions during the operation of the system [Fox, 1985].

An alternative method to robust schedule is to generate complete schedules with the insertion of idle times. Mehta and Uzsoy [1998] presented an approach to create schedules that include intentionally inserted idle times as a means to reduce the impact of disruptions. The method uses the shifting bottleneck algorithm to form operation sequences and then inserts idle time using a scheduling heuristic. Their studies indicated that schedules that are robust to stochastic disturbances such as machine breakdowns could be generated without much degradation of system performance. However, this technique can allocate some slack times for activity that may not occur [Akturk, 1999].

O'Donovan et al. [1999] described approaches which generate schedules that are robust with respect to machine breakdowns. The scheduling objective is to minimize the expected deviation in completion times between the initial and the revised schedules as well as to minimize expected tardiness on a one-machine scheduling problem. The approach, similar to Mehta and Uzsoy [1998], first uses a dispatching rule to generate a schedule and then uses a simple policy to insert idle time between jobs based on the expected downtimes. They also described a slightly different version of the method. In this version, the impaired conditions that the repaired machines will have after failures are considered while generating robust schedules. The results showed that these approaches improve schedule robustness with little impact on other performance measures. In a recent paper, Branke and Mattfeld [2002] introduced a secondary goal of penalizing early idle times into the objective function of their rolling time horizon scheduling approach using evolutionary algorithm. This approach, which they call anticipatory scheduling, is shown to preserve machine capacity by avoiding early idle times.

A variant of idle time insertion technique is the fuzzy set theory [Jones, 1998]. The uncertainties in shop floor are represented by fuzzy numbers using an interval of confidence. In the paper by Dorn et al. [1995a] for example, an application of fuzzy temporal reasoning to represent and propagate schedule uncertainty was discussed. The system uses constructive heuristics to generate an initial schedule. Some constraints may be violated in the schedule since jobs are scheduled in order of their priority. When the schedule is evaluated for such constraint violations, the processing times are replaced by their fuzzy possibility distributions, and temporal events are propagated using rules for the addition of fuzzy numbers. Temporal constraint violations are expressed in a degree of satisfaction based on the amount of overlap of the temporal estimates of events whose relationships are constrained. The schedule is evaluated in terms of the degree of satisfaction of the applicable constraints. The degree of robustness of the schedule is measured simply in terms of degrees of overlap of certain critical events. If the degree of satisfaction of a constraint is unacceptable, then the schedule is repaired by rescheduling. An iterative improvement method based on Tabu Search is used to improve the schedule score by progressive interchange of jobs.

2.4.2.2.2 Updating Schedules

A number of schedule update approaches are reported in the literature [Bean 1991; Leon, 1994; Sadeh, 1994; Abumaizar, 1997]. A summary of schedule update methods is presented in Table 2.8.

Table 2.8: Summary of some schedule update approaches

Categories	Schedule recovery method	Advantages	Disadvantages	Performance measures	Prerequisites for effective application	References
Tailored Algorithms (Heuristics)	Right Shift Re-scheduling (RSR)	Simplest method	Poor schedule quality. Constraints and rules are not resolved	Makespan and schedule stability	Shop floor should be stable with minor and infrequent disruptions	Leon, 1994; Efstathiou, 1996; Brandimarte, 1997; 2000

Categories	**Schedule recovery method**	**Advantages**	**Disadvantages**	**Performance measures**	**Prerequisites for effective application**	**References**
	Affected operation rescheduling (AOR)	Simple to implement. Better schedule quality than RSR	Computation time is greater. Limited disruptions can be handled	Makespan and schedule stability	Shop floor should be stable with minor and infrequent disruptions	Abumaizar, 1997; Leon, 1994; Hasle, 1994 Li, 1993
	Match-up rescheduling (MUR)	Can be simple to implement. Better schedule stability than RSR	Need to combine with other methods. Limited disruptions can be handled	Makespan and schedule stability	Shop floor should be stable with minor and infrequent disruptions. Enough idle time must be present in the initial schedule	Bean, 1991; Abumaizar, 1997; Akturk, 1999
	Micro-opportunistic rescheduling	Better quality schedule that RSR and AOR	Computation time can be greater than RSR and AOR	Tardiness and inventory cost	Shop floor should be stable with minor and infrequent disruptions	Sadeh, 1994
General Algorithms (Artificial Intelligence)	Case-based reasoning (CBR)	Well-suited to domain specific problems. Continuous learning from past cases. Multiple disruptions can be modeled and addressed	Extensive search through the database is time-consuming. An extensive experience database is essential	Schedule quality. Reactive efficiency in terms of deviation from the initial schedule	Job shops with specific rule sets and multiple disruptions where scheduling experience is available as expert's advice or case database	Sycara, 1994; Ovacik, 1994a; Miyashita, 1994; Szelke, 1997; Sushil, 1996
	Constraint-based scheduling	Timely response is possible in stipulated time frame. Performs better than CBR as it often includes both knowledge base and constraint satisfaction modules	Real-time approach requires further refinement	Computation time. Schedule quality and repaired weighted tardiness	Dynamic job shops with multiple disruptions and events	Sycara, 1994; Miyashita, 1995a; Spargg, 1997

Categories	Schedule recovery method	Advantages	Disadvantages	Performance measures	Prerequisites for effective application	References
	Fuzzy logic	Complete scan of the schedule for constraint violation after every repair. Random processing times can be used for disruptions	The knowledge of the domain has to be built into the algorithm	Computation time. Schedule quality and repaired weighted tardiness	Job shops with variability in processing time and large number of constraints	Dorn, 1995b; Dorn, 1994a; 1994b; 1994c; Schmidt, 1994; Dorn, 1995a; Slany, 1996
	Neural network	Response time is very fast for trained neural net. Predictions are extrapolated from past experience	Carefully prepared training sets are required for accurate prediction	Computation time. Weighted tardiness	More applicable in job shops with a continuous flow and repetitiveness in the type of disturbances	Garner, 1994; Rovithakis, 2001
	Multi-agents (in distributed AI)	Completely automated approach. Responsive. Multithread possible	Coordination between the agents is difficult to achieve	Computation time and quality measures	Dynamic job shop with uncertainties and random disruptions	Henseler, 1994; Szelke, 1994a; 1994b; Rabelo, 1994

The Right Shift Rescheduling (RSR) approach [Leon, 1994; Efstathiou, 1996; Brandimarte, 1997; 2000] is the simplest update approach reported in the literature. RSR postpones each remaining operation globally to accommodate the disruptions so as to make the schedule feasible [Abumaizar, 1997; Raheja, 2002b]. Figure 2.1 illustrates RSR operation. In this example, machine *M2* fails while processing job *J1* and the estimated repair time required is *r* unit time. By applying RSR, the completion time of Job *J1* on machine *M2* is delayed by *r* unit time, and the completion times of the remaining tasks on *M2*, *M3* and *M4* are also delayed by *r* unit time. The schedule obtained by this approach is generally of poor quality due to the resulting gaps in the schedule. This method can be used only if the shop floor is stable and only minor deviations occur from the initial schedule.

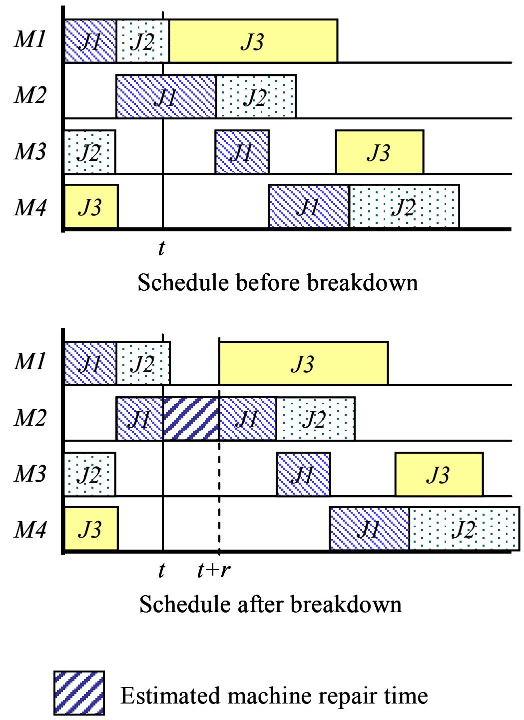


Figure 2.1: Using right-shift rescheduling to update a schedule

Affected Operation Rescheduling (AOR) [Li, 1993; Jain, 1997; Abumaizar, 1997] only reschedules the operations affected directly or indirectly by disruptions. This method claims to preserve the initial schedule as much as possible, tending to maintain schedule stability. The basic concept of AOR is to accommodate any disruption by pushing the starting times of some operations forward by the minimum amount possible so as to keep all the constraints satisfied and to preserve the initial sequence of the operations on each machine. Figure 2.2 illustrates the concept. In this example, job *J3* is not affected by AOR as compared to RSR.

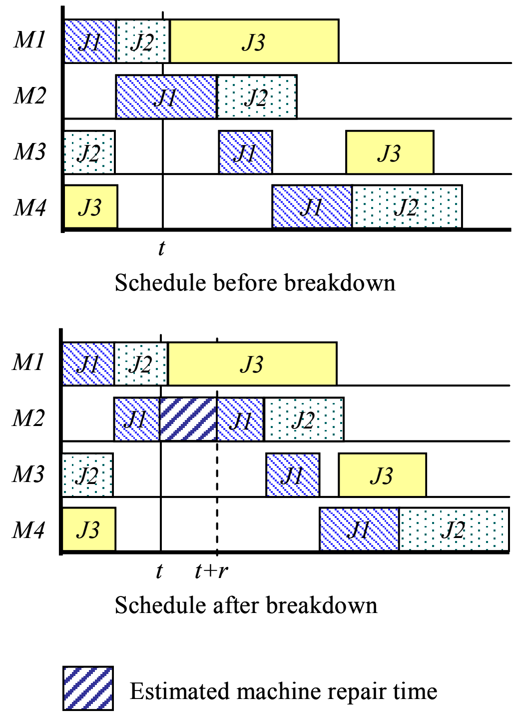


Figure 2.2: Using affected operation rescheduling to update a schedule

The AOR algorithm is often based on the construction of a binary tree for the schedule. The binary tree provides information on identification of affected operations and the function of net change. The schedule generated by AOR is claimed to perform better than RSR for objective function such as makespan but lack the optimization and quality improvement features in other repair approaches. These approaches are not suitable for highly stochastic shop floors as they handle the schedule disruptions independently of each other and only one disruption is considered at any one time.

Some AOR approaches can be problem specific. Jain and Elmaraghy [1997] described an iterative repair process of two steps. In the first step, the existing schedule is evaluated based on the change of the conditions, demands or constraints. In the second step, improved solution is determined. These algorithms assume some of the jobs in the system have alternative process plans and can provide immediate solutions to problems resulting from four types of disturbances: machine breakdowns, increased order priority, rush orders arrival and order cancellations. For example, a task that is unable to continue as scheduled, due to machine breakdown, will be reassigned to alternative free machines, failing which preemption is attempted. Most of the heuristic-based AOR algorithms are easy to use, implement and can handle disruptions effectively [Raheja, 2002b]. However, they do not perform a search for optimum repair and constraint satisfaction.

Match-up scheduling is another type of partial rescheduling [Bean, 1991; Abumaizar, 1997; Akturk, 1999]. When a machine failure requires revision of the current schedule, the revision is carried out but subject to temporal constraint that the revised schedule "matches up" to the initial schedule as soon as possible after the machine breakdown. This allows some consideration of the stability of the shop floor. In the match-up heuristic by Bean et al. [1991] for example, the time horizon is searched with equal increments until a unique match-up point is identified. Re-sequencing is then performed from the disruption to the match-up time using the best of six priority ordering rules. If the best solution results in large tardiness costs, the match-up point is increased. If the match-up point becomes excessive, multi-machine lot assignment rules based on an integer programming or priority rules are used to reassign jobs to different machines. Results indicated that the match-up approach is effective in terms of schedule quality, computation times, and schedule stability [Akturk, 1999]. However in this approach, it is claimed that enough idle time must be present in the initial schedule and disruptions sufficiently spaced over time to allow the system to get back on schedule before the next disruption.

Another schedule repair heuristic is micro-opportunistic [Sadeh, 1994]. In this approach, a set of operations that need to be rescheduled is identified and unscheduled. The scheduling problem is then consists of the unscheduled operations and the constraints imposed on these operations by operations that have been executed or have not been unscheduled. A look-ahead analysis will identify the most critical conflict to solve first and other conflicts will be solved subsequently. Resource contention is continuously monitored during the construction of the schedule, and the problem solving effort constantly redirected towards the most serious bottleneck resource. In the event that a feasible schedule could not be built, a larger number of operations are unscheduled. This approach may be very computation intensive.

Simulation-based techniques are also used as heuristic rescheduling. In the work by Kim and Kim [1994], a simulation-based real-time scheduling methodology for an FMS was developed. In their method, various dispatching rules are evaluated and the best one selected for a given criterion according to the results of simulation. The best dispatching rule is then used and the performance of shop floor is periodically monitored. When the difference between the actual performance and the value estimated by simulation exceeds a given limit, a new simulation is performed with remaining operations in the simulation, and a new rule is selected accordingly. Sivakumar et al. [2000] described a simulation-based near real-time scheduling for semiconductor test manufacturing.

Case based reasoning [Dorn, 1994a; 1995a; Sycara, 1994; Szelke, 1997] has been applied in reactive scheduling to find a case that best suits the disrupted schedule. Case-based reasoning (CBR) system uses past experience, stored in databases as cases, to help solve a current problem. The idea is first to collect and index a large collection of examples; then, when presented with a new situation, to fetch and adapt the stored example that most closely resembles the new situation and solve the problem. The case database, which is used to maintain the appropriate reaction to schedule disruptions, needs to be created by domain experts. Schedules are often complex, and to model all details of the schedule or the production environment in a case is not feasible. Instead, some characteristic surface knowledge is captured in a case. This abstraction of the problem also supports the application of the case for a new problem. Gradually, the cases are refined and a better knowledge base is created. The implementation of case-based reasoning in an experimental setup for schedule correction has been reported [Dorn, 1994] in which case-based reasoning is combined with an iterative improvement method using fuzzy constraints and a Tabu search for repair. Sycara and Miyashita [1994] described another case-based approach for selecting a repair tactic within a constraint-based schedule repair procedure. The repair tactics include adjusting start times, swapping operations, and switching to alternative resources. Case-based reasoning combines the advantages of both heuristics and experience modeling, as the case contains explicit heuristics for a given problem and it uses the solutions to old cases to solve new problems. However this approach is not very responsive, as it has to search through an extensive case base [Dorn, 1994].

The concept of constraint based scheduling is founded on the reuse of knowledge about the past in order to solve the current problem more quickly [Givry, 1997]. In constraint based approaches, problem changes are viewed as a set of constraint additions and removals. Smith et al. [1990] used a constraint-based schedule repair procedure to modify the operations in a partial schedule whose length is determined by the conflict duration. Suresh and Chaudhari [1993] have surveyed several other knowledge-based systems in this area. One of them is CABINS [Miyashita, 1995b], which is a hybrid intelligent scheduling system. The incremental accumulation and reuse of past experience is achieved through case-based reasoning, whereas constraint-based scheduling was used for the propagation and resolution of the effect of repair. The dominant constraints satisfied during the schedule recovery are activity precedence and resource capacity.

In fuzzy logic approaches [Dorn, 1994; Schmidt, 1994], threshold values can be defined for acceptable or unacceptable degrees of constraints. Uncertain events or disruptions will lead to constraint violations, and these are then evaluated by progressive shuffling of jobs. In this manner, the fuzzy repair strategy is applied until the violated constraints are satisfied [Dorn, 1994]. In this approach, high variability in the processing times can be accommodated. Another benefit is that the same process can be used for both the generation of robust predictive schedules and the subsequent schedule recovery [Schmidt, 1994]. On the other hand, since fuzzy logic is a hill-climbing search methodology, the repair could frequently be localized, leading to poor quality in terms of overlaps in processing times or unsatisfied constraints.

In neural network approaches [Garner, 1994], a trained neural network is used to predict the repair strategy by extrapolating the acquired knowledge, which is contained in an extensive knowledge database. For accurate predictions, appropriate training sets are required for each combination of circumstances. The training of a neural network is usually performed in a single pass and in some cases excessive re-training time can prove disadvantageous [Rovithakis, 2001]. The response time of the trained neural network is fast and predictions are usually reliable. Neural networks are also advantageous as they can learn and grow with the growth in the knowledge database and accumulation of experience.

Genetic Algorithm (GA) can also be used in rescheduling [Sushil, 1996; Bierwirth, 1999]. In these approaches, a GA finds and saves solutions to a scheduling problem, when the problem is changed slightly, and the old solutions are injected into the initial population of the genetic algorithm that is trying to solve the new problem. In a hybrid approach of GA and case-based reasoning [Sushil, 1996], appropriate solutions to open shop scheduling problems are injected into the genetic algorithm's population to speed up the augment genetic search on a related open shop rescheduling problem. Bierwirth and Mattfield [1999] presented a GA that reuses the previous solution to solve a job shop scheduling problem every time a new job arrives. Compared to a GA that starts from scratch, the GA with injected solutions quickly finds good solutions to new problem and that the quality of solutions after convergence is usually better.

Other AI based reactive scheduling approaches include knowledge-based scheduling methods that are used to generate feasible schedules and interactive methods to revise the initial schedules. Wysk et al. [1986] developed an integrated expert system/simulation scheduler called MPECS. The system uses both forward and backward chaining to select a small set of potentially good rules from a predefined set of dispatching rules and other heuristics in the knowledge base. These rules optimize a single performance measure, which could change from one scheduling period to the next. The selected rules are then evaluated one at a time using a simulation. The best rules are then implemented on a laboratory system. Dutta [1990] developed a knowledge based (KB) methodology to perform real-time production control in FMS environments. The proposed mechanism monitors the system and takes a corrective action whenever disruptions occur. The results showed that the KB mechanism with such corrective actions renders effective and robust production control.

The approaches discussed so far are all algorithmic solutions to the rescheduling problems. An alternative to solving the problems is structural techniques. One example is to decompose a scheduling problem into several sub-problems (medium-term and short-term planning, machine allocation problem, sequencing problem, etc.). Such decomposition is a structural solution for the computational complexity of the scheduling problem. Another approach is to decompose the problem in a heterarchical way, which is highly distributed form of control without centralized or explicit direct control [Lin, 1994]. The concept for heterarchical decomposition is derived from biological metaphors and analogies to the free market economy. In biological systems or market economies, reactivity to disturbances emerges almost automatically from simple mechanisms in the behavior of individual entity and the interactions between these behaviors. An implementation of this approach is OPIS, which has a blackboard-based control architecture where the current schedule is stored, together with all unscheduled tasks and requests for schedule repair. A number of expert modules are capable of solving sub-problems for schedule generation and revision. A "Top Level Manager" (TLM) opportunistically guides the search by calling the right expert modules for the right sub-problems in the right sequence.

Multi-agent paradigm can be used in structural solutions to solve rescheduling problems in a distributed way [Suresh, 1993]. The agents are processors with local procedures and rules, which can be applied to the problem solving. They need to cooperate, as none of the agents possess information necessary to solve the global problem. This approach solves sub-problems locally with an agent, and finds a global solution by interactions between the different agents, which maintain the local schedules. They operate through the cooperation of many interacting subsystems, each with its own objective, and modes of operation. Two main techniques can be observed: first, splitting the problem into several agents that locally solve the problem and globally negotiate the integrated solution; second, assigning a resource to an agent, which is then in charge of providing the service to other agents [Liu, 1994; Wooldridge, 1996].

2.5 Impact of Rescheduling

Apart from the effort in rescheduling methods, another body of work is on the impact that other aspects of the rescheduling have on manufacturing system performance. These other aspects include the decision as to whether to completely or partially reschedule, tradeoffs between schedule quality and stability, types of disruptions, and rescheduling frequencies.

The practical importance of the decision whether to completely or partially reschedule was discussed by Dorn and Kerr [1994] and Lee et al. [1996]. One of their core focuses is on the robustness and stability of the repaired schedules. Schedule robustness was discussed by Leon et al. [1994] and Daniels and Kouvelis [1995]. The impact of the chosen schedule repair strategy in response to machine failure and to design a robust initial schedule to minimize this impact was discussed. The stability of a repaired schedule was studied by Cowling and Johansson [2001].

The tradeoffs between schedule quality and stability were studied by Wu et al. [1993], who considered these tradeoffs for events taking place in real time, in order to compare the performance of different schedule repair strategies. Cowling and Johansson [2001] developed a general framework for using real-time information to improve on scheduling decisions, which allows them to trade off the quality of the revised schedule against the production disturbance that results from changing the initial schedule.

A summary of work on the impact of various aspects on manufacturing system performance is shown in Table 2.9.

Table 2.9: Summary of selected papers on the impact of various scheduling aspects on system performance

Reference	**Scheduling / Rescheduling Policy**	**Environments**	**Performance Measures**	**Remarks and Results**
Holloway, 1974	Periodic rescheduling using multi-pass heuristic procedure	Job shop, deterministic versus variable processing times	Number of tardy jobs, mean tardiness, variance of tardiness, and maximum tardiness	Multi-pass procedure is compared to single-pass dispatching rules. Results are problem dependent
Farn, 1979	Periodic rescheduling using optimum seeking algorithms	Single machine, static and dynamic	Changeover time	Optimum seeking algorithms is compared to dispatching rules with sequence dependent setup. Rescheduling leads to lower setup costs. Best heuristic for static job shop is not the best for the corresponding dynamic job shop
Yamamoto, 1985	Rescheduling is triggered by machine breakdown	Job shop, machine breakdown	Makespan	The rescheduling approach is compared to static scheduling and dispatching rules. The approach outperforms the sequencing policy and dispatching rules
Bean, 1991	Event-driven rescheduling using match-up algorithms	Multiple resource, machine breakdown and unavailable tool	Weighted total tardiness	The approach is compared with the no response policy (RSR) and several dispatching rules. Match-up algorithm leads to better performance
Church, 1992	Periodic rescheduling using earliest due date (EDD) rule, and reschedule on exceptional events (hybrid)	Single machine, dynamic job arrivals	Maximum lateness	Periodic rescheduling policies lead to near optimal when order release is periodic. More frequent rescheduling does not improve system performance significantly. However less frequent rescheduling can deteriorate performance

Reference	**Scheduling / Rescheduling Policy**	**Environments**	**Performance Measures**	**Remarks and Results**
Matsuura, 1993	Event-driven rescheduling using method to switch from sequencing to dispatching when unexpected event occurs	Job shop, semi-dynamic, machine breakdown, specification changes, rush jobs	Makespan	Switching method is compared to sequencing (BB) and dispatching rules (FCFS & SPT). Sequencing is preferred when machine breakdown. Dispatching is favored for rush job. The switching method performs well for makespan performance
Kim, 1994	A scheduling approach using simulation mechanism to evaluates dispatching rules and selects the best rule for a given criterion. Rescheduling occur when the difference between the actual performance value the estimated value exceeds a given limit	FMS, semi-dynamic, machine breakdown, urgent jobs	Mean flow time, mean tardiness and combination	The results show that too long monitoring periods resulted in worse performance of the systems and that too frequent monitoring could also negatively affect performance
Ovacik, 1994a	A family of rolling horizon heuristics, rescheduling after scheduling *n* jobs	Single machine, dynamic job arrivals, sequence-dependent setup	Maximum lateness	The procedures are compared with earliest due date (EDD) dispatching rule alone and EDD with local search procedure. The procedures outperform dispatching rules both on average and in the worst case
Sabuncuoglu, 1997; 1999	Periodic rescheduling using beam search and dispatching rules	FMS, machine breakdown, process time variation	Mean tardiness and makespan	Frequency of rescheduling is studied. It is not always beneficial to reschedule in response to every unexpected event. Periodic response with an appropriate period length is effective to disruptions
Lawrence, 1997	Static optimal schedule (BB with shifting bottleneck) and dynamic dispatching methods	Job shop, process time variations	Makespan	Optimal and heuristic methods are compared with varying degrees of uncertainty. Dynamic dispatching rules perform better than the static optimal schedules for even moderate amount of processing time uncertainty. Static optimal schedule quickly deteriorates with the introduction of processing time uncertainty when compared with dynamically updated heuristic schedules

Reference	Scheduling / Rescheduling Policy	Environments	Performance Measures	Remarks and Results
Shafaei, 1999a; 1999b	Periodic rolling horizon time procedure using non-delay algorithms with seven dispatching rules	Job shop, dynamic, machine breakdown, processing time variations, workload condition, system size	Cost-based performance measure, average throughput, mean tardiness, robustness	Non-delay algorithm and dispatching rules are compared with different frequency of rescheduling, workload condition and system size. Frequent rescheduling becomes more effective as the level of uncertainty increases. Performance becomes more sensitive as the system size increases. Bottleneck is a major factor that influences the robustness of a schedule when there are uncertainties
Akturk, 1999	Event-driven rescheduling using reactive hierarchical scheduling approach (RHSA) on machine breakdown	Modified flow shop, machine breakdown	Tardiness, stability	The approach is compared with right-shift rescheduling and several dispatching rules. RHSA is effective in terms of schedule quality, stability and computation times
Sabuncuoglu , 2000	Periodic heuristic algorithm that is based on the filtered beam search (optimization-based)	Job shop, machine breakdown, workload condition, system size	Mean tardiness, makespan	The algorithm is compared to dispatching rules. Optimization based scheduling method performs better than dispatching rules when the workload across the machine is not uniform. Periodic rescheduling method is affected more than the dispatching mechanism when there is stochastic disturbances such as machine breakdowns
Vieira, 2000a; 2000b	Periodic and event-driven rescheduling using FIFO dispatching rule to sequence job and group jobs with similar types to save setup time	Single machine, parallel machine	Mean flow time	Rescheduling frequency is studied using dispatching rules with same setup consideration. A lower rescheduling frequency lowers the number of setups by grouping similar jobs but increases manufacturing cycle time and work-in-progress. A higher rescheduling frequency allows the system to react more quickly to disruptions but may increases the number of setups

The earliest study in this area attributes to Holloway and Nelson [1974] who implemented a multi-pass heuristic procedure in a job shop by generating schedules periodically. Simulation was employed to compare the performance of the schedule generated by the heuristic procedure, a non-delay transformation of that schedule, and the non-delay schedules obtained with single-pass dispatching rules. The delay schedule produced by the heuristic procedure was found to be superior under certain conditions. Under other conditions, the relative performance of the scheduling rules appears highly problem dependent. Later, Farn and Muhlemann [1979] used simulation to compare dispatching rules and optimum seeking algorithms for the static and dynamic single machine scheduling problems with sequence-dependent setup times. Arriving jobs are included in the schedule at the next rescheduling point, and the schedule is created periodically using a priority rule such as first-in-first-out or shortest processing time. They concluded that rescheduling often leads to lower setup costs, and the best heuristic for a static problem is not necessarily the best for the corresponding dynamic problem. Subsequently, Muhlemann et al. [1982] studied the dynamic job shop scheduling problem and compared different scheduling heuristics across a range of scenarios, including length of rescheduling period, the number of jobs in the backlog, and the amount of uncertainty in processing times and machine failures. They suggested that the rescheduling period can affect system performance more when there is greater uncertainty and that the tradeoffs between the cost of scheduling and the benefits of more frequent scheduling needs to be explored. Yamamoto and Nof [1985] studied a rescheduling policy in a static scheduling environment with random machine breakdowns. Rescheduling was triggered whenever a machine breakdown occurs. The results indicated that the proposed approach outperforms the fixed sequencing policy and dispatching rules.

Bean et al. [1991] considered event-driven rescheduling of shop floors with multiple resources on unanticipated events. They proposed a match-up algorithm to reschedule so as to match-up with the initial schedule at some point in the future whenever a machine breakdown occurs. The match-up approach was compared with the no response policy and several dispatching rules. The results showed that the match-up algorithm leads to better performance than a simple pushback strategy (RSR). Later, Akturk and Gorgulu [1998] applied a new rescheduling strategy and a match-up point determination procedure to a modified flow shop. They pointed out that the initial schedule and the duration of breakdown can affect the efficiency of the proposed approach. The results indicated that their approach can improve the schedule quality, schedule stability and computation times in comparison with several match-up alternatives and the static pushback strategy (RSR).

Church and Uzsoy [1992] developed a hybrid (i.e. periodic and event-driven) rescheduling policy for single machine model with dynamic job arrivals. Their system reschedules the facility periodically. Regular events occurring between routine rescheduling are ignored until the next rescheduling point. However, when an event is classified as an exception, immediate action is taken, with the entire facility being rescheduled and resulting schedule implemented until the next schedule generation point. To create a schedule, the system uses earliest due date (EDD) rule to minimize maximum lateness. The paper claimed that periodic rescheduling policy leads to near optimal performance when order release is periodic. In addition, rescheduling at the arrival of a “rush” job is useful, but more frequent rescheduling does not improve system performance significantly. The results also indicated that the performance of periodic scheduling deteriorates as the length of rescheduling period increases. Later, Ovacik and Uzsoy [1994b] proposed several rolling time horizon procedures in a single machine environment with sequence dependent setup. In these procedures, the dynamic scheduling problem is decomposed into a series of smaller sub-problems. The forecast window length and the maximum size of the sub-problems can be determined by algorithm parameters. These parameters enable tradeoffs between solution time and schedule quality to be described and selected. The experimental results indicated that these procedures outperform dispatching rules both on average and in the worst case, in reasonable computation times.

Matsuura et al. [1993] studied the problem of selection between sequencing and dispatching as a rescheduling approach in a job shop environment involving machine breakdowns, specification changes, and rush jobs. Their results indicated that sequencing approach is preferred to dispatching rules where small machine breakdowns occur. However in the rush job case, dispatching approach becomes more efficient. They concluded the combined switching approach perform better compared to other approaches particularly when specifications changes occur and on rush job arrival. In another simulation study, Kim and Kim [1994] considered minor and major disturbances in their scheduling system. The simulation mechanism evaluates various dispatching rules and selects the best one. This heuristic will be called periodically, according to a monitoring period that is a multiple of the mean operation processing time, and at major disturbances. Several values for the monitoring periods were studied. They concluded that there was an advantage to checking the system performance periodically and that long monitoring periods resulted in worse performance of the systems and that too frequent monitoring could also negatively affect performance.

Sabuncuoglu and Karabuk [1997; 1999] studied the frequency of rescheduling in an FMS environment. The authors proposed several reactive scheduling policies to cope with machine breakdowns and processing time variations. Their results indicated that it is not always beneficial to reschedule the operations in response to every unexpected event and the periodic response with an appropriate period length can be quite effective in dealing with the interruptions. A moderate level of rescheduling frequency was also proposed to alleviate the negative effects of machine breakdowns.

Lawrence and Sewell [1997] compared optimal (Branch and Bound with shifting bottleneck) and heuristic methods (i.e. dispatching rules) in a job shop environment with makespan objective when processing times are known with varying degrees of uncertainty. The results indicated that dynamic scheduling heuristics perform better than the static optimum schedules for even moderate amount of processing time uncertainty. The results hold for the degree of uncertainty commonly observed for practical scheduling problems. It was shown that fixed optimal sequences derived from deterministic assumptions quickly deteriorate with the introduction of processing time uncertainty when compared with dynamically updated heuristic schedules. Similar results were obtained by Wu et al. [1999]. Their studies showed that as processing time variability increases, dispatching rules lead to better performance than robust partial schedule. These results suggested that for practical scheduling problems, dynamic heuristics provide results comparable or superior to optimal fixed schedules found using sophisticated deterministic algorithms. The principal benefit of simple heuristics is that they can be dynamically applied, whereas the use of sophisticated optimal and algorithmic techniques exerts a large and often prohibitive computational toll. However, sequence-dependent setup was not considered in the job shop environments. In the studies by Ovacik and Uzsoy [1994a], their results showed that the dispatching rule, earliest due date (EDD) often performs poorly when sequence-dependent setup is considered.

Shafaei and Brunn [1999a; 1999b] investigated a number of dispatching rules in a dynamic and stochastic job shop. The objectives of the studies include the effects of rescheduling interval on the system performance, and the effects of workload condition and system size on the performance. Their results indicated that under loose due date conditions, the performance is not particularly sensitive to changes in rescheduling interval. However, at tight due date conditions, the rescheduling interval had a much more significant effect on performance. Further, the performance becomes more sensitive as the size of the system increases, and bottleneck is a major factor that influences the robustness of a schedule when there are uncertainties. They have also shown that frequent rescheduling becomes more effective as the level of uncertainty increases.

Sabuncuoglu and Bayiz [2000] studied the reactive scheduling problems in a stochastic manufacturing environment. Periodic response policy and partial rescheduling were analyzed under machine breakdowns in a classical job shop system with the effect of system size and type of workload allocation (uniform and bottleneck). The schedule generation method used in the study is a heuristic algorithm that is based on the filtered beam search. The dispatching rule for the mean tardiness criterion is a modified operation due date (MODD) priority rule and a global evaluation function of the shortest processing time (SPT). For the makespan criterion, most work remaining (MWR) priority rule were used. Their results showed that the optimization based schedule generation method performs better than dispatching rules when the load across the machine are not uniform in both static and stochastic environments. Further, periodic schedule generation method is affected more than the dispatching mechanism when there are stochastic disturbances such as machine breakdowns. This result is consistent with the previous studies by Lawrence and Sewell [1997] and Sabuncuoglu and Karabuk [1997].

Vieira et al. [2000a] studied a single-machine system and developed analytical models to estimate the system performance. The work considered two rescheduling policies: periodic and event driven based on queue size. The scheduling algorithm uses a first-in-first-out (FIFO) dispatching rule to sequence jobs and it also groups jobs with similar types to save setup time. Their results showed that the analytical models can predict the performance of a single-machine system operating under those rescheduling strategies. Vieira et al. [2000b] extended the study by investigating parallel machine systems, which have more complex rescheduling strategies. These papers have shown that rescheduling frequency can significantly affect the system performance (average flow time). A lower rescheduling frequency lowers the number of setups (reducing unproductive time wasted on setups) by grouping similar jobs but increases manufacturing cycle time and work-in-progress. A higher rescheduling frequency allows the system to react more quickly to disruptions but may increase the number of setups.

2.6 Summary

In this chapter, we review rescheduling research and propose classification schemes for the scheduling and rescheduling environments, strategies and methods. We classify the past literature under the new framework and compare the advantages and disadvantages of the rescheduling strategies and policies. Further studies reveal that different manufacturing environments require different solution approaches. Static, deterministic scheduling problems demand static scheduling approaches whereas dynamic or stochastic scheduling environments call for dynamic, reactive or proactive scheduling approaches.

We also review and summarize some static scheduling approaches, which are often used in rescheduling. We adopt a classification methodology for the scheduling techniques from the previous work [Jain, 1998], and extend it to rescheduling techniques. Two major techniques to rescheduling are identified: dynamic and reactive scheduling. The former uses dispatching rules to select jobs on the shop floors whereas the latter constructs schedules on a rolling horizon basis.

Another area of our work is on the impact of rescheduling strategies on the manufacturing performance. The general consensus is that the performance of rescheduling strategies is dependent on the various aspects of scheduling environments and scheduling techniques. These aspects include optimal scheduling algorithm versus dynamic heuristic methods, robust schedule versus nominal schedule generation, schedule quality versus stability, frequencies of rescheduling versus event-driven rescheduling, and so on. In general, the performance of optimization methods and dispatching rules tends to converge when there is increasing uncertainty and variability in the system.

The past literature indicates that the problems of disruptions on shop floors are attracting more rescheduling research. The key focus of the research is on maintaining solution stability under disruptions. Different rescheduling strategies and various rescheduling methods have been proposed and studied. No general and consistent conclusions can be drawn from the literature as rescheduling problems seem to be complex problems involving a variety of conditions. A comprehensive study seems necessary to yield more insights into the various aspects, and the advantages and disadvantages of rescheduling in different problem settings. The research issues include the tradeoffs or relationships between schedule quality and stability under different scheduling strategies for different types of disruptions. Detailed study and analysis of various aspects of rescheduling and the effects on dynamic and stochastic shop floor environments is thus crucial for better understanding of the emerging rescheduling challenges.

Although there have been some effort in consolidating the research work in rescheduling, a unified method, which aims at resolving the deficiencies in the existing approaches, is still lacking. The key focus is to find effective and yet simple rescheduling approaches that can perform well under constant disruptions without compromising much on the solution quality. Such techniques should consider the scheduling needs of the dynamic and stochastic shop floor environments on a continuous basis, and provide quality solution at all times.

CHAPTER 3

RESCHEDULING FACTORS AND PROBLEM FORMULATION

3.1 Introduction

Discrete manufacturing environment today is characterized by increasing levels of variability and uncertainty. Both variability and uncertainty can lead to poor and unreliable manufacturing performance. To deal with these issues, an appropriate planning and scheduling function can play an important role. The goal is not to prevent or eliminate [Harlin, 2002; Ylipää, 2002] the variability or uncertainty, but to reduce its effects. Thus, it is important to assess the impact that the dynamic and stochastic environments have on the scheduling and rescheduling. Only with the understanding of the effects that variability and uncertainty have on the scheduling function, can we seek to develop a better technique for solving the rescheduling problem.

Variability and uncertainty arise due to unanticipated events in manufacturing systems. These unforeseen events are known under the collective name of "disturbances" [Schumacher, 2003]. A production disturbance is considered as a deviation from earlier assumptions. Disturbances can result from changes in customer's demand, breakdowns, materials supplies, human errors, deficient information flow, and so on. Studies in industry have shown that 20% to 30% of work is performed on other equipment than was originally planned [Lampkemeyer, 1991], and manufacturing systems are vulnerable to various types of production disturbances [Harlin, 2002]. Recent studies have also shown that production disturbances can, on average, reach 80% of the total loss of the overall equipment effectiveness (OEE) of even 50% mean value [Ylipää, 2002]. Hence, the production disturbances not only warrant a better understanding in their influence on scheduling function, but also must be considered as an integral part of the scheduling function in manufacturing systems.

However, the scheduling problem is not trivial as local disturbances can affect the global performance of a manufacturing system in a non-linear way due to NP-hard nature of scheduling problem [Bongaerts, 1997]. Non-linearity implies that for small deviations of the input parameters, the value and the structure of the optimal solution may change considerably. In his overview on stochastic scheduling models, Pinedo [1995] even concluded that the greater the number of uncertainties in the system, the easier it is to use simple scheduling rules.

While the effect of disturbances on scheduling is often neglected [Pinedo, 1995], disturbance is only one of the three major factors (or dimensions) that can affect the performance of rescheduling (Figure 3.1). The other two factors are the characteristics of shop floors and the rescheduling techniques. We will discuss these factors prior to exploring any rescheduling solution.

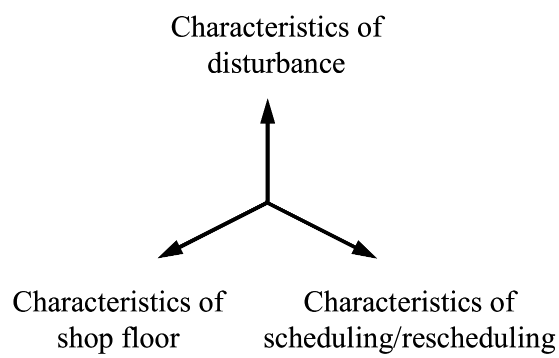


Figure 3.1: Three major scheduling factors that can affect the performance of rescheduling

In Section 3.2, production disturbances are discussed in detail. In Section 3.3, characteristics of manufacturing environments that can affect the performance of rescheduling are described. In Section 3.4, some of the important rescheduling concepts and tradeoffs in finding near-optimal or robust solutions are studied. Section 3.5 discusses the performance criteria and metrics that are commonly used to assess the performance of rescheduling. Section 3.6 seeks to study the impact of production disturbances on manufacturing systems through analytical models. Section 3.7 consolidates the issues in rescheduling for dynamic and stochastic shop floor environments, and attempts to identify the actual scope of the research problems. Section 3.8 studies factors and settings other researchers have considered for the dynamic and stochastic scheduling problems, but are not considered in the scope of our research. Finally, Section 3.9 summarizes the chapter.

3.2 Production Disturbances in Manufacturing Environments

A production disturbance can be defined as an unanticipated change to production conditions with a negative effect on the process performance [Brueckner, 1998]. A production disturbance should be differentiated from a production change, which is an intentional alteration to the production conditions. We term both disturbances and changes under a common name of "disruptions". Research on disturbances usually centers on their impact from the perspective of a scheduler. In this respect, a disturbance can be considered as an event which upsets the contents of the schedule. Machine breakdowns or an overrun processing time would therefore be regarded as disturbances. If the schedule is based on erroneous assumptions and the production process deviates (as a result), then this would also be referred to as a disturbance in the scheduling.

Two major kinds of disturbances can be observed: Boolean and parametric disturbances [Schumacher, 2003]. Examples of Boolean disturbances are technical malfunctions, which may or may not cause a break in operations. An example of parametric disturbances is overrunning of a process, whereby the determining factor is the actual level of the disturbance. Only cases in which a deviation from the standard duration of the process exceeds a predefined value would be considered as a disturbance.

3.2.1 Diversity and Classification of Production Disturbances

Some classification schemes for disturbances exist in the literature. Suresh and Chaudhari [1993] classified disturbances into categories of job-related, machine-related, operation-related or others. In another classification [Stoop, 1996], disturbances were divided into three categories, relating to capacity, orders or measurement of data. There are similarities and differences between the classification schemes. In this research, we propose a consolidated classification scheme that seeks to encompass the most commonly found changes and disturbances in the discrete manufacturing environments, as shown in Table 3.1. In this scheme, disruptions can be divided into two categories:

1) **Disruptions relating to capacity** – These disruptions result in changes in production capacity and thus affect the operation of the production units e.g., machine breakdowns, unavailability of specific tools, variations in processing times, etc.

2) **Disruptions relating to jobs (or orders)** – These disruptions delay (or advance) the progress of jobs e.g., unavailability of specific materials, due date changes, etc.

Table 3.1: The consolidated classification scheme of disruptions

Classification	**Type**
Disruptions (Changes or disturbances)	• Capacity • Jobs

The new classification scheme, together with some common disruptions [Vieira, 2002; The Mascada Consortium, 2003] and the related references, is shown in Table 3.2.

Table 3.2: Common changes and disturbances

Disruptions	**Type**	**Examples of changes or disturbances**	**References**
Changes	Capacity	Change in the work force	The Mascada Consortium, 2003
		New production technology	The Mascada Consortium, 2003
		Change in production capacity	The Mascada Consortium, 2003
	Jobs	New products or variants	The Mascada Consortium, 2003
		New operations or flow	The Mascada Consortium, 2003
		Specification changes (i.e. insertion or deletion of operations in existing job)	Matsuura, 1993
Disturbances	Capacity	Machine failures	Yamamoto, 1985; Church, 1992; Li, 1993; Kim, 1994; Jain, 1997; Abumaizar, 1997; Sabuncuoglu, 1999; Shafaei, 1999a
		Unavailability of tools	Suresh, 1993; Stoop, 1996
		Operator absenteeism	Church, 1992
		Differences between estimated and actual processing time, setup time, etc. (i.e., over- or under-estimating)	Li, 1993; Stoop, 1996
		Differences between estimated and actual yield (quality miss), etc.	Suresh, 1993
	Jobs	Delayed, shortage or defective of materials or supplies	Li, 1993; Abumaizar, 1997
		Demand changes (fluctuating)	The Mascada Consortium, 2003
		Rework or quality problems	Church, 1992; Li, 1993; The Mascada Consortium, 2003
		Urgent (or rush, or hot) orders	Li, 1993; Kim, 1994; Jain, 1997; Abumaizar, 1997
		Change in job priority	Jain, 1997
		Job (or order) cancellation	Li, 1993; Jain, 1997; Abumaizar, 1997
		Due date change (i.e., delay or advance)	Li, 1993

Some of these disruptions may lead to further disruptions (i.e. propagation of disruptions). Unavailability of tools or operator absenteeism for example, may lead to non-functioning of the related machine. Rework or quality problems may result in process time variation, job cancellation and so on. Due date change of orders may induce a change in job priority.

3.2.2 Characteristics of Production Disturbances

Table 3.3 compares the different types of disruptions occurring in industry [The Mascada Consortium, 2003] and being studied by the research community. The second column of the table shows the types of disruptions, and the third and fourth columns indicate survey results pertaining to the industry and academic respectively. For industry case, the data related to the number of factories involved in the survey and the frequencies of disruptions is presented. For academic case, the number of studies associated with a specific type of disruptions is exhibited. The papers in the survey are those papers deal with disruptions and are given in the reference section.

The industrial study shows that the most frequent intentional changes are the introduction of new products or new variants, and changes in work force. However, our study on academic papers shows that very little work has been done on the intentional changes related to their impact on the schedule by the research community. The primary reason is that these changes can be planned ahead of time.

The data in Table 3.3 indicates that there is a close mapping on disturbances that occur frequently in the industries and that are often being studied by the research community. These disturbances, in order of their occurrences, are machine failures, fluctuating demand and process time variations.

Table 3.3: Comparison of common changes and disturbances in industry and literature

Disruptions	**Type of disruptions**	**Industry**		**Academic**
		Number of factories involved in the industrial survey	**Frequencies of disruptions**	**Number of studies in the surveyed literature**
Changes	New products or variants	12	63	0
	Change in workforce	5	26	0
	New production technology	3	16	0
	Change of capacity	2	11	0
Disturbances	Machine failures	13	**68**	**9**
	Demand changes	10	**53**	**4**
	Process time variations (including rework or quality problems)	10	**53**	**3**
	Bad delivery (i.e. incorrect material)	8	42	0
	Late delivery	7	37	2
	Work force unavailability (operator absenteeism)	2	11	2

The types of disruptions shown in Table 3.3 are also known as rescheduling factors by Vieira et al. [2002]. Obviously, the type of disruptions can affect the generated schedules. Further, all these disruptions contain a common set of properties that can also affect the schedules:

- **Size** – This is related to the magnitude of disturbances. Examples are machine failures can be short or long period, and orders can be of small or large lot size.

- **Frequency** – This is related to the frequency of disturbances, or the rate of occurrences. An example is the machine failures: consider two scenarios with the same total failure durations over a time period, the failures can happen frequently in many short periods, or infrequently but in a few long periods.

- **Time of occurrence** – This refers to the time of occurrence of the disruptions, which can occur either early or late in a schedule.

3.3 Characteristics of Manufacturing Environments

Apart from the disturbances, the characteristics of shop floors may impact the performance of rescheduling. These characteristics are related to the shop floor conditions, and can be categorized into three factors:

- **Type** – This is related to the different types (or layout) of shop floors. The two most common types are flow shop and job shop. Different shop floor models may lead to different level of difficulty in solving the scheduling problems. It is commonly perceived that a job shop model is more complex than a flow shop model.

- **Size** – This refers to the size of a scheduling problem, and can be expressed as the number of machines and job operations present in the scheduling problem.

- **Workload** – This is associated with the shop floor loading and load distribution. Workload can be expressed in terms of utilization levels on the shop floor, whereas workload distribution is related to uniformity of loading. A shop floor can be uniformly or non-uniformly loaded. In a uniformly loaded shop floor, the average utilization levels of machines are alike. However, in a non-uniformly loaded shop floor, some machines are overloaded with jobs compared to other machines. The overloaded machines are commonly known as bottlenecks in the system. Depending on the demand fluctuations of different products over time, bottlenecks may shift from production period to production period.

3.4 Characteristics of Rescheduling

The aspects of rescheduling that are important in determining the effectiveness of rescheduling are the solution features and rescheduling features. The former is related to the features of the scheduling solutions and the tradeoffs between them, while the latter is concerned with the effects of using different features in rescheduling.

3.4.1 Solution Features

In most situations, the information available to the decision-maker is both incomplete and uncertain. Scheduling is often considered as a problem of decision-making under uncertainty [Berry, 1993b]. Uncertainty refers to the uncertainty at a given time about the dynamic stream of events in the future. Under this uncertain environment, some performance criteria for rescheduling can be identified to describe the solutions. These criteria are:

- **Solution quality** – A solution is considered to have a sufficient level of quality if it meets shop performance measures or commonly known as soft constraints, in addition to all the hard (or non-relaxable) constraints. A solution is termed as a feasible solution if it meets all the hard constraints. Hence, a quality solution must also be a feasible solution, but not otherwise. The difference between hard and soft constraints is the constraints to satisfy and criteria to optimize respectively. The latter is very much depending on the way users express their objectives. The exact level of quality of a solution depends on the way it satisfies the soft constraints. Examples of soft constraints are throughput rate, and utilization levels of machines. Examples of hard constraints include the operation precedence of jobs, machine requirement of operations, and so on. This criterion is equally applicable in static, deterministic scheduling.

- **Solution stability** – Stability measures the number of revisions or changes that a schedule (or solution) undergoes during its execution. Methods for measuring solution stability include calculating the average absolute change of start times of operations [Raheja, 2002a], or both the start and completion times [Cowling, 2001] of operations for the initial and the revised schedules. This criterion is important in applications where changes in the solution are costly or difficult to execute.

  Solution stability is related to the robustness of a solution. A solution can be made robust by making it resistant to changes. For instance, keeping some capacity unused in case of machine failures can improve robustness at the expense of machine utilization. Robustness indicates how much disruption would degrade the performance of the system as it executes the schedule [Vieira, 2002]. The difference between solution stability and solution robustness is in the time. Stability is a measurement for the past solutions whereas robustness is an indicator for future solutions (see Figure 3.2).

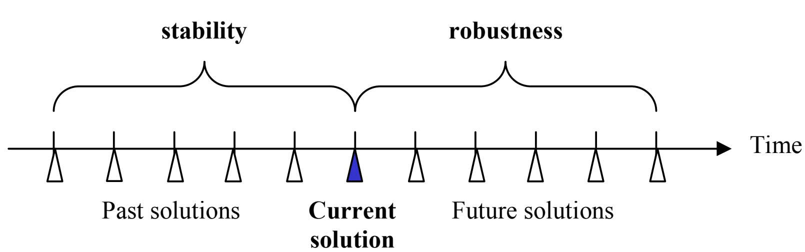


Figure 3.2: Solution stability versus solution robustness

- **Solution adequacy (or solution reliability)** – Adequacy refers to the validity of a solution after a certain elapsed time. The adequacy problem arises because any scheduling task will take a certain minimum time (Δt). For instance, a feasible solution produced on the basis of a given context at time t is of no value to the system (i.e. its adequacy is null), if the context have completely changed from time t to time $t + \Delta t$ in the system.

The adequacy of solution can be taken as how fast a scheduler reacts to disruptions. One of the most simplistic measures is the time taken to reschedule. This measure can be used to indicate the computational burden and the response time to disruptions. In certain cases, a scheduler is required to complete its decision within a limited time (i.e. temporal deadline). This is an operational constraint, and can be distinguished into two kinds:

- **Deadline is known** – The system must deliver a solution before the end of a given time contract.

- **Deadline is unknown** – The system can be interrupted and asked for a solution any time, or the system can choose the right moment to deliver its solution. In the former, the algorithm should perform well on average during a specified temporal interval. In the latter, meta-reasoning approach can be based on the notion of utility, which monitors the progress of problem-solving algorithm [Hansen, 1996].

Temporal constraints impose a distinction between a mandatory portion to obtain a first feasible solution (i.e. at minimum computation time), and an optional portion to obtain an optimal solution (i.e. within temporal deadline).

Conflicting relationships exist between some of these solution features. These relationships require further elaboration:

- **Solution quality versus solution stability** – Conflicts exist between solution quality and stability [Verfaillie, 1999]. If solution stability is systematically favored, solution quality may degrade as changes occur and the differences between the current problem and the initial solution grow under frequent disruptions.

  Generating robust solutions is an attempt to maintain schedule quality (or solution quality) with simple adjustments when updating. Common ways to generate robust solutions are by directly (e.g., idle time insertion) or indirectly (e.g., fuzzy logic) inserting slack time into schedule, or by generating least commitment schedule [Dorn, 1995b; Mehta, 1998]. Due to the introduction of slacks, these methods may have a degrading effect on the solution quality especially if less disruption occurs than anticipated. In other words, robust schedules can be sub-optimal.

- **Solution quality versus solution adequacy** – Generally speaking, an optimum seeking approach searches in a larger solution space and can thus generate better solution quality but at a cost of greater computation times [Berry, 1993a]. On the contrary, a simple rescheduling approach with good response time often leads to poor solution quality. If finding quality solutions can take a large amount of time, there are conflicts between solution quality and solution adequacy in dynamic and stochastic shop floor environments. Under a specific dynamic context, amount of computing resources and problem complexity, the higher the quality of the solutions, the longer the time to produce them, and thus the lower their adequacy. Figure 3.3 shows finite time involved in scheduling.

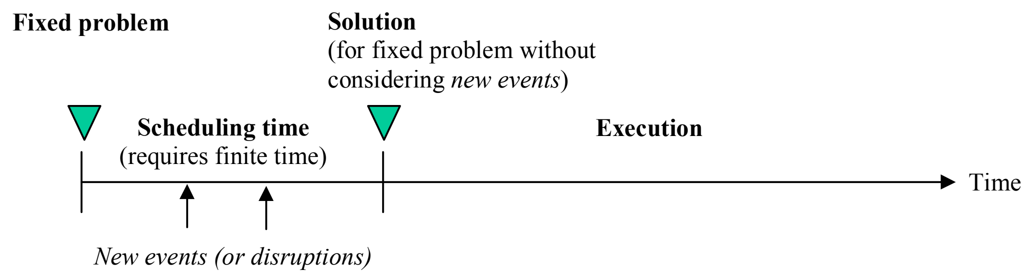


Figure 3.3: Finite time involved in performing scheduling and its effect on solution adequacy

Given a finite time required to perform a scheduling task, the solution adequacy can be very low under highly dynamic and uncertain production environments [Verfaillie, 1999]. In this case, performing scheduling to look for quality solution may no longer be meaningful. Under this circumstance, dynamic scheduling approaches using simple dispatching rules may be a better choice. However, the solution quality may be sacrificed due to the myopic nature of these rules.

3.4.2 Rescheduling Features

There are a number of features in rescheduling that can affect the performance of rescheduling. Careful consideration of these features is important to ensure effective rescheduling under disruptions.

3.4.2.1 Scheduling Horizon

In most scheduling approaches, temporal (or scheduling) horizon needs to be established. This horizon can be fixed either statically or dynamically, and can be either single or multiple. In dynamically fixed horizon, its length may be determined based on the level of environmental and scheduling uncertainty. A multiple horizon is a set of nested horizons, with a scheduling granularity associated with each of them and increasing with size. For example, within a one-hour horizon, dates may be specified to within one minute. And, within a one-day horizon, they are specified to within one hour.

Temporal horizon is limited by the level of uncertainty in the environments. Beyond a certain limit, the schedule may not be reliable. When determining the scheduling horizon, factors that can affect the horizon are to be considered:

- **Computation time –** Scheduling horizon has a direct impact on the computation time. A longer horizon may take more time to compute the schedule.

- **Solution quality –** Scheduling horizon has an influence on the solution quality. A longer horizon may potentially produce better quality solution. This is because longer horizon results in larger search space and hence more solution alternatives, which may lead to better solutions.

  To illustrate, consider a simple example of 4 jobs and 2 machines in a shop floor model, with two different scheduling horizons, as shown in Figure 3.4. One schedule is with a single long horizon (Figure 3.4 (a)) whereas the other consists of two smaller segments of equal horizons (Figure 3.4 (b)), which are half the former. Assume that the primary objective of scheduling is to minimize makespan and scheduling is performed for each horizon. In the case of half-time horizon, jobs J_1, J_2 and J_3 are scheduled first based on the job priority (Figure 3.4 (b)). This will result in longer makespan ($C_{\max}^{b} > C_{\max}^{a}$) as compare to the single long horizon case.

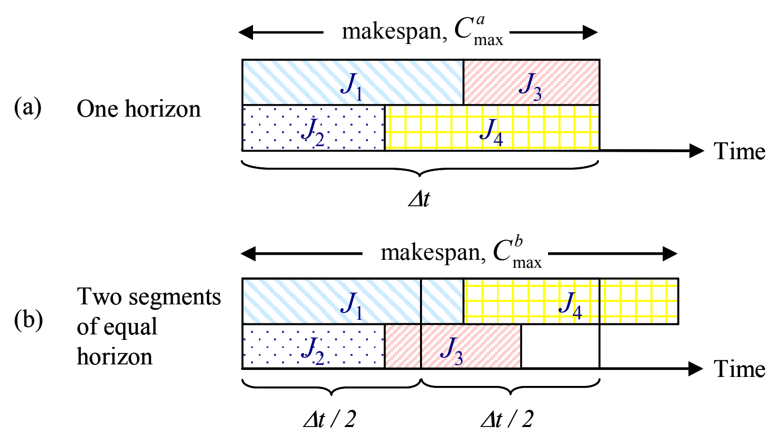


Figure 3.4: Comparison of solution quality between different scheduling horizons

- **Coordinated behavior** – There can be a commit horizon where the schedule is fixed ahead of time for coordination among tasks. For example, when execution cannot follow decision immediately, either decision needs time to be sent to the execution system (e.g. factory oversea), or execution needs time to be prepared (e.g., materials have to be prepared). Operational constraints therefore imply that the scheduling horizon must be longer than the commit horizon for coordinated behavior in the environments.

3.4.2.2 Scheduling Frequency

Scheduling frequency states how often rescheduling is performed. It is the inverse of the rescheduling period, which is the time between two consecutive scheduling points. A scheduling point is a moment in time when a rescheduling process is started. Rescheduling (refer to Figure 3.5) can be invoked anytime from time t to just before the end of time $t + h$ of the current horizon (h). Considering a maximum time, d for decision making, scheduling task can be invoked at worst at $t + h - d$ and scheduling ready at $t + h$. Note that time t and $t + h - d$ are the start of scheduling process and $h - d$ is the rescheduling period.

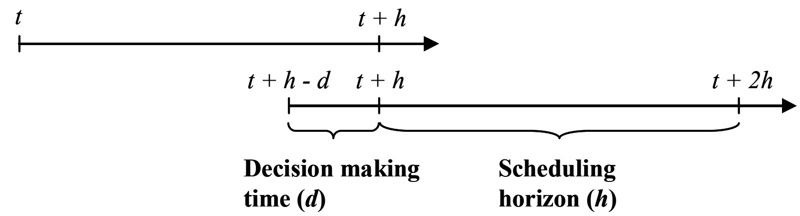


Figure 3.5: Worst case scenario of rescheduling invocation

Rescheduling may also be performed when new conditions invalidate current schedule, or when it is near the end of the current horizon. In the worst case scenario, rescheduling is required just before the end of the current horizon. At best, rescheduling can be performed on every disruption. In practice, rescheduling can be event-driven, periodic, or hybrid. The advantages of being more reactive (e.g. by event-driven policy) are better quality of service (since disruptions and requests are quickly taken into account), more reactive to uncertainty and ability to constantly update the schedule [Verfaillie, 1999]. The disadvantage of being more reactive is a shorter time for reasoning, and thus may cause degradation in solution quality. One way to reduce rescheduling frequency is to be more phlegmatic about external events by filtering or aggregating the events.

3.5 Performance Criteria and Metrics

Rescheduling heuristics can range from optimum seeking approaches such as Branch and Bound algorithm to fast procedures such as dispatching rules. To determine which approach is better under different environmental conditions, it is necessary define a set of performance measures which can quantify the performance of different rescheduling approaches.

The quality of solutions generated by different rescheduling approaches is often assessed off-line through simulation experimentations. A functional view of the experimental setup is shown in Figure 3.6. In this setup, a manufacturing model is used to mimic the characteristics of the actual manufacturing system. Events that are associated with disruptions such as the arrivals of new jobs, machines breakdowns, and so on are generated by the event generation module. The scheduling module reacts to the events and performs rescheduling accordingly. The scheduling module also interacts with the manufacturing model to obtain current shop floor status, and to provide job dispatching lists for the machines in the model. The performance of the scheduling procedure is assessed by the performance measure module, which tracks jobs and machines status in the manufacturing model.

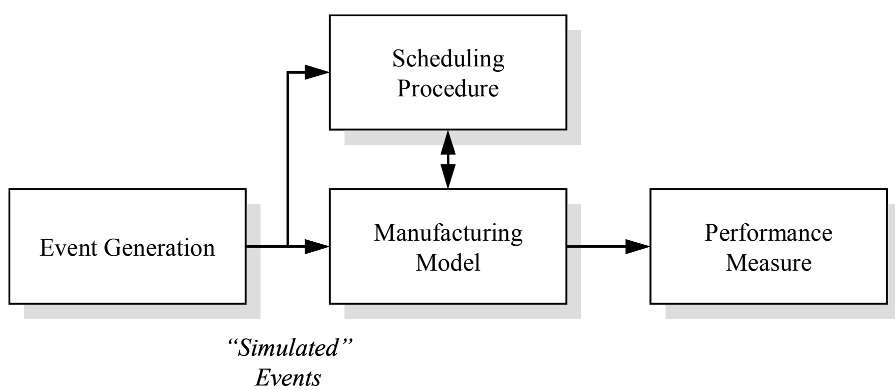


Figure 3.6: Simulation-based experimental setup to assess scheduling performance

Two categories of performance criteria can be identified relating to rescheduling. The first is the solution quality, which measures the effectiveness of a scheduling procedure in meeting soft constraints. The second is the performance criteria pertaining to rescheduling. These criteria are solution stability and solution adequacy.

Similarly, two groups of performance measures can be identified corresponding to the performance criteria: scheduling and rescheduling performance measures. Scheduling performance measures are for assessing the performance of scheduling system in achieving production related objectives, while rescheduling performance metrics are for evaluating the performance of the scheduling system under disruptions. In the following subsections, the commonly employed performance measures relating to scheduling and rescheduling are discussed.

3.5.1 Performance Measures: Scheduling

Table 3.4 shows the results of an industrial survey [The Mascada Consortium, 2003] conducted on the major performance criteria and the related performance measures. Some of these criteria such as high delivery reliability, short delivery time, and so on are related to scheduling and may be improved through effective scheduling approaches. The corresponding performance measures are commonly used in research. These measures include makespan [Yamamoto, 1985; Wu, 1993; Mehta, 1998; Sabuncuoglu, 1999], mean tardiness [Kim, 1994; Jain, 1997; Sabuncuoglu, 1999], and mean flow time or cycle time [Kim, 1994; Jain, 1997; Vieira, 2000a; 2000b].

Table 3.4: Management Strategies and related performance measures for scheduling

Rank	Criteria	Related performance measures
1	Consistent product quality	None
2	**High delivery reliability**	Mean tardiness, total tardiness, maximum lateness and number of late jobs.
3	Higher product quality than competitor	None
4	First to market	None
5	**Short delivery time**	Mean cycle time and makespan.
6	Customization	None
7	Better after sales service	None
8	**Lower price than competition**	Cost (or economic) based measures such as total cost function.
9	Broader product range	None
10	More product features	None

Most realistic scheduling problems require the simultaneous optimization of more than one objective function. To consider multiple performance criteria in scheduling, an often used measure is the weighted mean [Sivakumar, 2000a; 2000b]. Suppose there are n numerically defined performance criteria $(P_1, P_2, \ldots P_n)$ and for these criteria, the corresponding weights are $(W_1, W_2, \ldots W_n)$. We have:

$$\text{Weighted mean, } \omega = \frac{1}{\sum_{i=1}^{n} W_i} \cdot \sum_{i=1}^{n} W_i \cdot P_i$$

Another way to incorporate multiple criteria is to evaluate scheduling decisions by using economic (cost-based) performance measure instead of the time-based performance measures. This approach assigns certain cost to each criterion and the objective is to minimize the total cost.

3.5.2 Performance Measures: Rescheduling

One of the common objectives of rescheduling is to minimize the deviations from the initial schedule when responding to disruptions. The schedule recovery task can be constrained by the precedence order of job operations, resource constraints, the initial schedule and the key scheduling objectives. For such scheduling problems, the performance measures can often be linked to the stability or adequacy of the solutions.

3.5.2.1 Solution Stability

To measure stability criteria, a distance or metric of changes has to be established in order to compare two solutions [Verfaillie, 1999]. Commonly used metrics are: (1) a weighted sum of time shifts, (2) a count of activities shifted in time, (3) a sum of changes of positions of activities on a resource. In cases (1) and (2), the starting and ending times of activities are considered. Two stability measures that are commonly used in literature are: starting time deviation and sequence deviation. The former is based on the temporal movements of job operations while the latter is based on the change in operation sequence of the schedule.

3.5.2.1.1 Starting Time Deviation

The starting time deviation measure is represented by the difference in the starting times of activities between initial and revised schedules. It can be computed as the absolute sum of difference in the starting times of the job operations between the initial and the revised schedules divided by the total number of operations [Abumaizar, 1997]:

$$\text{Start time deviation measure, } \eta = \frac{\sum_{i=1}^{n}\sum_{j=1}^{o(i)}\left|s_{ij}^{r} - s_{ij}^{o}\right|}{\sum_{i=1}^{n} o(i)}$$

Where,

n = total number of jobs

$o(i)$ = total number of operations on job i

s_{ij}^{r} = starting time for operation j of job i in the revised schedule

s_{ij}^{o} = starting time for operation j of job i in the initial schedule

A schedule is said to be stable if it deviates minimally from the initial schedule. Thus, solution stability is considered good if the starting time deviation is minimal. This is a useful measure of effectiveness for rescheduling heuristics, especially in shop environments where tools or materials are delivered to the machine based on the initial schedule. Note that the measure considers both the carrying costs and rush order costs. The carrying costs are incurred if the tools or materials are delivered earlier than requested whereas the rush order costs are incurred if the tools or materials are requested earlier than planned.

3.5.2.1.2 Sequence Deviation

This measure is important if setups are prepared in advance based on the initial operation sequence on the machines. For instances, jobs may wait on pallets in a queue, and tools may be planned in advance according to the initial sequence. Thus, a sequence change will incur costs in re-sequencing the queue, reallocating the pallets, and re-planning the tools or setups. To measure the deviation in sequence, the following function [Abumaizar, 1997] is used:

Sequence deviation measure, $\vartheta = \sum_{i=1}^{m} \sum_{j=1}^{o(i)} n\left(P_j^o \cap S_j^r\right)$

Where,

m = total number of machines

$o(i)$ = total number of operations on machine i

P_j^o = set of operations processed before operation (preceding operations) j in the initial schedule

S_j^r = set of operations processed after operation (succeeding operations) j of job i in the revised schedule

$n\left(P_j^o \cap S_j^r\right)$ = cardinality of $P_j^o \cap S_j^r$

3.5.2.2 Solution Adequacy

One way to measure rescheduling adequacy is the response time of rescheduling approaches to disruptions. This measure can be used to indicate the burden on computing resource running the scheduling system [Vieira, 2002]. It can be measured as the mean time taken to perform rescheduling in response to events. We propose another measure, which is the mean response time relative to mean time between disruptions:

$$\text{Adequacy, } A = \frac{\dfrac{1}{n(E)} \sum_{i=1}^{n} (c_i - \tau_i)}{\dfrac{1}{n(E)-1} \cdot \sum_{i=1}^{n-1} (\tau_{i+1} - \tau_i)}$$

Where,

$n(E)$ = total number of events or disruptions

τ_i = start time of event i

c_i = completion time of rescheduling in response to event i

A higher value of A for a rescheduling approach simply means that the approach, in general, takes longer time to respond to events. On the other hand, a lower value for a rescheduling approach denotes that the approach is faster in responding to events.

We propose three measures for the cases where temporal constraints with known deadlines are imposed on the reasoning. These are lateness, tardiness and number of missed deadlines relative to the total number of disruptions. The first two are shown mathematically as follows:

$$\text{Lateness, } L = \frac{1}{n(E)} \sum_{i=1}^{n(E)} (d_i - c_i)$$

$$\text{Tardiness, } T = \frac{1}{n(E)} \sum_{i=1}^{n(E)} \max(0, d_i - c_i)$$

Where,

d_i = temporal deadline of event i

3.6 Disruption Modeling and Analysis

In this section, we look at how production disturbances affect discrete manufacturing operations by using mathematical models. Analysis of the models can lead us to identify important strategies for rescheduling. This model is adapted from work done by Frizelle et al. [1998]. We first discuss a mathematical model for a shop floor model. This is then followed by the derivation of a disturbance propagation model.

In the shop floor model, machines are linked to one another by the passage of jobs between them. The pattern of work is determined by a schedule. Each job follows a path through the machines visiting some but not others. Each operation on a machine is characterized by a duration, which may include setup and process times. There will also be some slack, which may be zero, between the operation and the next operation on the machine. If all these slack is zero then the machine is considered as a bottleneck. If a job visits a second machine, there will also be some slack between the completion of the job on the first machine and the start of the same job on the next machine. A feasible schedule is one where the slacks are all zero or positive. We will first derive a steady state model and then perturb the model. By perturbing the model, we can observe the effect of a disturbance on the model.

3.6.1 Steady State Model

Consider a manufacturing model with N jobs, $i=1,2,\ldots N$, where each job i consists of a number of operations, on $k=1,2,\ldots M$ machines. For each machine k, there will be N_k $(\leq N)$ operations assigned to it. For each job i, there will be N_i $(\leq M)$ machines in its path. The operation in position j of machine k is referred as operation j, with a duration $p_{i,k}$, The slack between operation j and $j+1$ on machine k is denoted as $\delta_{j,k}$ as shown in Figure 3.7. In the figure, $J^{i}_{j,k}$ denotes job i, with its operation located at j^{th} position on machine k. There will also be a slack $\sigma_{i,k}$ between job i on machine k, and job i on machine $k+1$ (i.e. next machine that job i visits). A schedule is feasible if

$\delta_{j,k}, \sigma_{i,k} \geq 0$ for all $\delta_{j,k}, \sigma_{i,k}$. Otherwise it will be infeasible. We shall only deal with feasible schedules in our models.

Assume that $t_{j,k}$ is the start time of the operation in position j on machine k, and let $\tau_{i,k}$ be the start time of job i on machine k. We can derive the following recursive equations:

The capacity equation, $t_{j,k} = p_{j-1,k} + \delta_{j-1,k} + t_{j-1,k}$ **(3.1)**

The sequence equation, $\tau_{i,k} = p_{i,k-1} + \sigma_{i,k-1} + \tau_{i,k-1}$ **(3.2)**

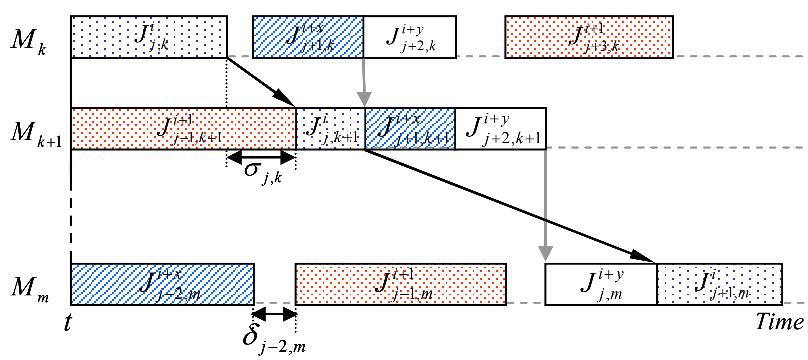


Figure 3.7: A feasible schedule

Solving (3.1) recursively gives:

$$t_{N_k,k} = \sum_{j=1}^{N_k-1} \left(p_{j,k} + \delta_{j,k}\right) + t_{1,k} \tag{3.3}$$

Solving (3.2) recursively gives:

$$\tau_{i,N_i} = \sum_{k=1}^{N_i-1} \left(p_{i,k} + \sigma_{i,k}\right) + \tau_{i,1} \tag{3.4}$$

By using equations (3.3) and (3.4), it is possible to mathematically link two operations in the schedule. For example, consider operations $J^i_{j,k}$ and $J^i_{j+1,m}$ in Figure 3.7. We can trace from $J_{j,k}^{i}$ to $J_{j+1,m}^{i}$ by many different paths, two of them are $\{J_{j,k}^{i}, J_{j,k+1}^{i}, J_{j+1,m}^{i}\}$ by using (3.3) and $\{J_{j,k}^{i}, J_{j+1,k}^{i+x}, J_{j+1,k+1}^{i+x}, J_{j+2,k+1}^{i+y}, J_{j,m}^{i+y}, J_{j+1,m}^{i}\}$ by both (3.3) and (3.4). Obviously, all the paths are of equal length. Hence, this property of equal path length can be used to define a steady state schedule. Generally speaking, we can derive a general expression for any path from first operation on machine k to another operation by combining both (3.3) and (3.4):

$$t_{N_k,N_i} = \sum_{k=c}^{N_i-1}\left(p_{i,k} + \sigma_{i,k}\right) + \ldots + \sum_{j=a}^{N_k-b}\left(p_{j,k} + \delta_{j,k}\right) + \ldots + t_{1,k} \tag{3.5}$$

Where,

t_{N_k,N_i} = start time of operation J_{N_k,N_i}^{x}, which can generally be used to denote any operation

3.6.2 Perturbation Model

In this section, a mathematical model is used to illustrate the effects of production disturbances on discrete manufacturing operations. By using a mathematical model, one is able to better understand (as compared to words) how a perturbation impacts manufacturing operations, and how the perturbation is eventually absorbed by slacks between operations in the model. The effects of perturbations or disturbances on manufacturing operations are important in the study of event-driven dynamic scheduling as it can affect the strategies as well as performance of rescheduling.

Using the steady state model, we can consider the situation where a disturbance occurs and model its impact on the operations. Suppose that a disturbance of size

$\Delta(>0)$ is applied at operation $J^{i}_{j,k}$ (refer to Figure 3.7) and we want to analyze its impact on operation $J^{i}_{j+1,m}$. If $\Delta \le \delta_{j,k}$, the disturbance is absorbed by the slack. However, if $\Delta > \delta_{j,k}$, the disturbance is propagated along machine k. Operation $J^{i}_{j+1,m}$ will be moved forward by $\Delta - \delta_{j,k}$ (refer to Figure 3.8). Obviously, the larger the value $\delta_{j,k}$, the smaller will be the propagated disturbance Δ and the sooner it will be absorbed.

Rewriting (3.1) by introducing disturbance $\Delta_{j-1,k}$, we have:

$$t_{j,k} + \Delta_{j-1,k} = p_{j-1,k} + \delta_{j-1,k} + t_{j-1,k} + \Delta_{j-1,k} \qquad \textbf{(3.6)}$$

Rearranging (3.6), we have:

$$t_{j,k} + \Delta_{j-1,k} - \delta_{j-1,k} = p_{j-1,k} + t_{j-1,k} + \Delta_{j-1,k} \qquad \textbf{(3.7)}$$

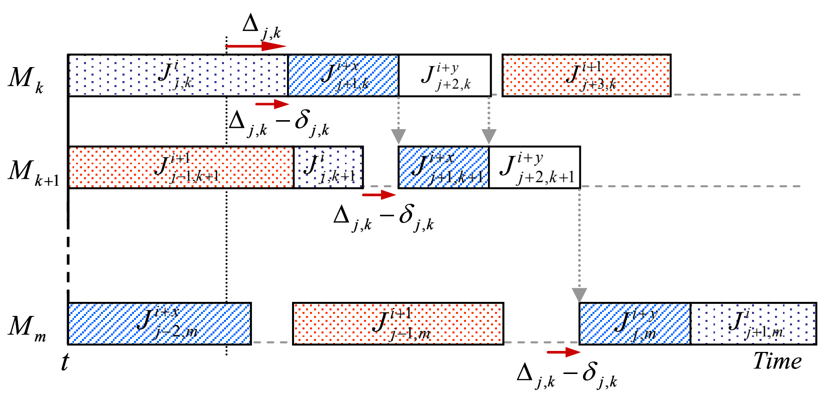


Figure 3.8: A perturbed schedule

Let $\Delta_{j,k} = \Delta_{j-1,k} - \delta_{j-1,k}$, we have from (3.7)

$$t_{j,k} + \Delta_{j,k} = p_{j-1,k} + t_{j-1,k} + \Delta_{j-1,k} \qquad \textbf{(3.8)}$$

But $\Delta_{j,k}(\geq 0)$ can be considered as disturbance at the succeeding operation on machine *k*. This means that disturbance is propagated along the machine by pushing each operation against its successor and absorbing the slack in the process but also attenuating the disturbance. If $\Delta_{j,k} = 0$, then there is no propagation. All the subsequent start times will be unaltered. Hence, there is a unique absorption point.

A similar reasoning holds for sequence equation. For example, if $\Delta_{i,k} > \sigma_{i,k}$ in Figure 3.7, then the disturbance propagates along machine *k* as well as operation sequence of job J^i. Clearly, among all the possible paths from the source of a disturbance to a specific operation, the path with the minimum total slack will eventually determine the impact of the disturbance on the operation.

Based on the perturbation model, we can infer that in order to have a robust schedule, slack (e.g. spare capacity on machines) is needed. If there is no slack, disturbance will not be absorbed or attenuated, and hence may significantly affect the succeeding operations on machines or on job operation sequences. If more operations are affected, the stability measure tends to be higher (or worse). Thus, it is important to leave spare capacity on critical machines in order to make the schedule more robust to disturbances. However, allowing spare capacity on critical machines may result in poor solution quality due to underutilized machines. Thus, an effective rescheduling approach must be able to strike a balance between solution quality and solution stability.

3.7 Scope of Rescheduling Problems

We have examined the characteristics of three major factors that can affect the performance of rescheduling: disturbances, shop floor conditions and rescheduling. Under each of the factors, there are various sub-factors. Further, each of these factors may interact with each others in a complex way. To consider all of them in the rescheduling research is definitely a major task to undertake. Therefore, we have to focus our effort on the most important ones by ways of consolidation and elimination. We will analyze each of these factors and provide our rationale to either include or exclude the individual factor in our research.

We identify the various properties of disruptions: type, size, frequency and time of occurrence. Many different types of disturbances can be found in industries and literature. Some of these disturbances may lead to the same end results as others, and thus will not be included in our study. Unavailability of tools or operator absenteeism for example, may lead to unavailability of the machine that requires the tools or operators to function. Bad or late delivery will subsequently result in fluctuation in the arrival of jobs. We choose to center on the three most commonly found and studied disturbances: machine breakdowns, fluctuating demand and process time variations.

From the list of properties pertaining to disturbances, some of them, such as time of occurrence, are related to a particular schedule rather than to a shop floor. In dynamic and stochastic shop floor environments, the satisfactory performance of a schedule doesn't imply the same for all subsequent schedules. We believe that it is the long-term steady state performance of the underlying scheduling function that is more important. Considering this aspect in the actual manufacturing environments, disturbances can and will occur at any time with respect to the pre-computed schedules over the stipulated manufacturing period. Clearly, the time of occurrence is beyond control in the actual manufacturing environments and thus is of little value to study further. Moreover, the dynamic scheduling approaches such as dispatching rules do not generate schedules, but only dispatch jobs whenever machines become available. Hence, to have a fair comparison between different scheduling approaches, it is logical for us to pursue only the properties of disturbances that are applicable to both the long-term performance and different scheduling approaches. The properties in this category are type, size and frequency of disturbances.

The three characteristics of shop floors are: type, size and workload. The results from the past research showed that the type of shop floors may or may not affect the relative performance of scheduling heuristics. In a study of a simplified flow shop, Barrett and Barman [1986] investigated the performance of 5 dispatching rules and their combinations. Their results showed no difference in the relative effectives of the rules when compared to the same rules in a job shop. In a subsequent simulation study on 9 dispatching rules for a dynamic flow shop [Barrett, 1990], it was observed that the performance of the dispatching rules evaluated in a flow shop is similar to the performance of the rules in a job shop. In another study [Rajendran, 1999], a comparative study of 13 dispatching rules in dynamic flow shops and job shops was presented. In this simulation study, it was observed that the relative performance of dispatching rules can be influenced by the routing of jobs and shop floor configurations. Apparently, the past experimental studies exhibited conflicting results on the relative performance of dispatching rules for job shops and flow shops. However, a detailed study shows that the conflicts can be attributed to the dispatching rules with look-ahead component. This is because look-ahead component in the rules is more meaningful in job shops than in flow shops due to non-unidirectional routing of jobs in job shops. Our work will focus on job shop environments, which is more representative of the discrete manufacturing environments.

A variety of scheduling heuristics particularly dispatching rules have been tested in shop floors of various sizes and it was found that the size of the shops does not affect the relative performance of these rules [Baker, 1960; Nanot, 1963]. Buffa [1968a; 1968b] also observed that shop size never appears to be a major variable, and it is possible to experiment with small shop floors and generalize the resulting conclusions. This finding is valuable as it is always cheaper to simulate smaller shop floors.

Past studies have shown that the performance of scheduling heuristics can be affected by the utilization rate or loading level of the shop floor. Rajendran and Holthaus [1999] found that the best dispatching rule for mean cycle time at low utilization levels (80%) becomes less effective at high utilization levels (95%) in a job shop. However, it was also observed that the performances of some rules are consistent and good in the same order for different utilization levels in job shop environments (although they are not the best). In another study of a dynamic flow shop [Barrett, 1990], it was observed that shortest processing time (SPT) is the best performing rule for mean cycle time at different utilization levels (75% and 95%). Another aspect of workload is the workload distribution. Since we are more concerned with the long term steady-state performance of shop floors, we prefer to focus on the utilization levels instead of workload distribution of the shop floor. Further, long term bottlenecks in the system often demand dedicated scheduling heuristics to handle the conditions in order to achieve better solutions. However, we assume in this work that temporary bottlenecks will naturally occur and shift from period to period due to demand fluctuations.

In terms of scheduling solution, we have identified three solution features: solution quality, solution stability and solution adequacy. We have discussed about the conflicting nature between solution quality and stability. The existence of the conflict is evidenced by using the steady state and perturbation models. In general, in order to have a robust schedule, idle times in the schedule (or spare capacity on the machines) are needed. However, inserting idle times directly or indirectly in the schedules is not a common practice in the discrete manufacturing industry. Hence, we will not focus on scheduling techniques that insert idle times. This leads us to disregard solution stability in our work. The decision is also supported by the following reasons:

- It is assumed that disruptions are uncertain and unpredictable, and hence inserting idle times may result in unexpectedly negative effects. Furthermore, idle time insertion may only be suitable to certain classes of manufacturing environments which are highly dynamic and stochastic.

- Some scheduling strategies or techniques such as delayed commitment scheduling and dynamic dispatching rules can not be compared on the performance measure of solution stability. This is because these scheduling techniques generate either partial schedules or no schedule. This causes difficulty in applying the solution stability measures such as the start time deviation measure to the scheduling approaches.

- For rescheduling, solution stability criteria can always be treated as soft constraints in the performance measures and can thus be considered as part of solution quality. One way to contain multiple criteria is to use the weighted mean measure. By adjusting the weights of the different performance criteria, emphasis can be shifted from one criterion to another. By using this method, the rescheduling problems become multi-objective optimizing problems. For the revised schedule generation, the goal is then to optimize production objectives and to minimize changes relative to the initial schedules. Since our focus is more on event-driven scheduling and not on optimizing multiple objectives, we prefer to concentrate on one primary objective, mean cycle time of jobs.

Solution adequacy measure can be used to evaluate the validity of a solution. However, the issue of solution adequacy is not normally addressed in rescheduling research for discrete manufacturing environments because:

- Time of rescheduling is generally much shorter than the time between disruptions. Hence, rescheduling task in the environments is often considered non-time critical.

- Temporal constraints are uncommon in production scheduling for discrete manufacturing. Further, the ways to deal with temporal constraints are relatively well established and some general guidelines have been discussed in Section 3.4.1. With the standard methods, it becomes more an implementation issue rather than a research issue to handle temporal constraint in rescheduling.

Therefore, we chose to deal with the issue of solution adequacy only with the simple measure of execution time.

Two factors in rescheduling have been discussed: scheduling horizon and scheduling frequency. There is very little research on the scheduling horizon. We have demonstrated earlier by an example (Section 3.4.2.1) that the scheduling horizon can have impact on solution quality. The scheduling horizon and frequency are both related to periodic rescheduling. The less frequent the rescheduling, the longer the scheduling horizon needs to be. Past research indicates that the frequency of rescheduling can impact the performance of rescheduling. The earlier study in this subject was by Muhlemann et al. [1982]. In their work, the performance of a number of heuristics under different scheduling conditions, which are determined by rescheduling frequency, level of uncertainty in process times and machine breakdowns, was investigated. It was concluded that the performance of scheduling algorithms deteriorates as the rescheduling period increases (frequency of rescheduling decreases).

A more recent study on periodic rescheduling strategies is by Vieira et al. [2000a]. An analytical model was used to predict the performance of one-machine systems in an environment where different job types arrive dynamically for processing and setup would be incurred when production changes from one job type to another. The scheduling algorithm was a first-in-first-out (FIFO) dispatching rule and similar types of jobs were grouped to save setup time. It was found that longer rescheduling periods reduce the number of setups, and thus improve machine utilization, but increase average cycle time of jobs. The result is to be expected as there is a tradeoff between reducing the number of setups and reducing the amount of inventory (or average cycle time of jobs).

Table 3.5 summarizes the scope of work in rescheduling that we will focus on. The table shows the major factors, the corresponding sub-factors and remarks.

Table 3.5: Scope of Rescheduling Problems

Major Factors	Sub-factors	Remarks
Disturbance	Type	Machine breakdowns, fluctuating demands and process time variations
	Size	Different durations of breakdowns
	Frequency	Different frequencies of job arrivals and breakdowns
	Time of occurrence	- Not considered
Shop floor	Type	Job shop model
	Size	Shop model with fixed number of machines
	Workload	Different utilization levels
rescheduling	Solution quality	One primary performance criterion, mean cycle time
	Solution stability	- Not considered
	Solution adequacy	Execution time of rescheduling
	Scheduling horizon	Different horizons in periodic rescheduling
	Scheduling frequency	Different frequencies in periodic rescheduling
	Temporal constraint	- Not considered

3.8 Rescheduling Factors Not Considered in the Scope

In dynamic and stochastic job shop environments where orders of work continue to arrive randomly and disruptions occur frequently, many factors can influence the performance of scheduling heuristics. We have discussed the important factors to be considered in rescheduling. However, there are some other factors, which past researchers have considered, and are important for us to understand their effects on the scheduling performance. These factors include the performance measures, methods of assigning job attributes, job arrival rate distribution, order release procedure, and uncertainty in estimating process parameters. We shall discuss these factors and settings in conjunction with the past research.

The performance measures that are most frequently used are cycle time (or flow time), lateness and tardiness. Cycle time is defined as the amount of time a job spends in the system. Lateness is the amount of time by which the completion time of a job exceeds its due date, whereas tardiness is the positive lateness of a job. Many researchers have used functions of cycle time, lateness, and tardiness as the measures of effectiveness of scheduling heuristics. Past results have shown that the effectiveness of a scheduling heuristic varies with the measurement criteria [Rajendran, 1999; Blackstone, 1982]. For example, minimization of cycle time does not minimize tardiness even for a single machine model. In general, no single rule performs well for all important performance criteria.

The computation of some performance measures such as lateness and tardiness makes use of the assigned due date of jobs. However, there are many ways to assign due date to newly arrived jobs. For example, the assignment methods can be based on total work content, or the number of operations of jobs [Blackstone, 1982]. It was found that the performance of scheduling heuristics for measurement criteria such as lateness and number of tardy jobs is dependent on the method of due-date establishment [Conway, 1965; Elvers, 1973]. In consideration of the impact of due-date establishment on tardiness measure, there are combined scheduling approaches proposed in the literature. For example, Miyazaki [1981] combined due-date assignment and sequencing procedure to reduce job tardiness in a job shop. However for cycle time criteria, no due-date assignment is required.

The effect of various arrival rate distributions on the relative success of job shop dispatching rules was studied by Elvers [1973; 1974]. In the study, 16 different distributions were tested on 10 dispatching rules. It was concluded that the distribution with respect to shape and range of the arrival rate for incoming jobs is not a significant factor in evaluating the relative effectiveness of dispatching rules [Blackstone, 1982].

For some manufacturing systems, release of jobs can be moderated. In this case, when jobs arrived at the shop they are not immediately released, but queue in a pre-shop pool instead. The job release is controlled, allowing a limit to the workload in the shop floor. For this class of manufacturing systems, there are primarily two ways in which control can be exercised over the shop in terms of shop scheduling and control. First, new jobs to be released into the shop can be specified through a release policy. Second, for jobs already in the shop, which job to be processed next at each machine can be determined through a sequencing policy. One of the earlier studies in the release or input control is by Deane and Moodie [1972]. They proposed a dispatching rule that includes workload balance as a consideration. Later, Irastorza and Deane [1974] extended the procedure by incorporating an algorithm to balance the shop workload through control of the release of jobs to the shop floor. In a subsequent simulation study by Wein [1988], it was found that the combined scheduling approach has a significant impact on the mean cycle time performance of semiconductor wafer fabrication, with larger improvements coming from discretionary input control than from lot scheduling policy [Qi, 2002].

In scheduling, it is often necessary to have estimates, for example processing times of operations. The schedule or sequence produced may be dependent on these estimates. When processing actually takes place these estimates may not be achieved. In a study by Dannenbring [1997] on 11 flow shop sequencing heuristics, it was observed that performances are not greatly affected by the data errors in estimated versus true processing times. Even at 20% standard errors, none of the heuristics has as much as 5% increase in relative error. The results also indicated that those procedures that perform best with error-free data are the most sensitive to errors in the data. Conway and Maxwell [1962] explored the performance of shortest process time (SPT) rule in an m-machine environment. It was found that imperfect information about processing times had little effect on the operation of the SPT rule.

From these simulation studies, it is clear that the only other factor that can influence the mean cycle time performance of scheduling heuristics is the order release procedure. However for our study, we are more concerned with the relative performance of scheduling heuristics under the same order release policy, and thus the release policy could be disregarded in our experiments.

3.9 Summary

We have examined and discussed in detail the various issues relating to rescheduling under dynamic and stochastic shop floor environments. We identify the three dimensions associated with the factors in rescheduling, which must be carefully considered in the study of rescheduling. The importance of some of these factors, and the tradeoffs between some of them is also highlighted. We explore several performance measures that can be used to guide the search in solution space, or to benchmark the rescheduling approaches. We study the impact of a disturbance on a schedule using analytical models, and show the conflicts between solution quality and stability. Without losing the key focus on the objectives of our research, we subsequently select the important factors among the factors of rescheduling to be considered in our study.

Our study on issues relating to rescheduling for dynamic and stochastic shop floor environments shows that rescheduling problems are more complex than scheduling problems. Apart from standard scheduling problems pertaining to solution quality, two other aspects which are solution stability and solution adequacy need to be considered. Solution quality and stability are conflicting objectives to achieve. On one hand, solution quality usually requires production objectives to be optimized. On the other hand, solution stability demands minimum changes in the revised schedules as compared to the initial schedules. To balance the two solution features, we propose to treat solution stability as soft constraints. Standard methods that consider multiple performance criteria such as weighted mean can then be used to favor either solution quality or stability depending on shop floor conditions. In the case of solution adequacy, we choose to compare the performance of different scheduling heuristics by measuring the time taken to generate schedules.

Past research in rescheduling centers on the performance of schedules, and the main focus is on the solution stability. The objective of the research is to reduce the impact of disruptions on the schedules. The two widely adopted rescheduling techniques are either to create robust schedules or to minimize changes in the revised schedules as compared to the initial schedules. We divert from this focus as we think that it is the long-term steady state performance of a shop floor rather than the performance of specific schedules, is more important to discrete manufacturing environments. There are few points to support our decision:

- Disturbances on a shop floor are uncertain and unpredictable. In an environment that is uncertain and unpredictable, it is difficult to predict the disruptions in advance and thus may cause the performance of the scheduling methods to vary from period to period.

- Comparing the performance of scheduling techniques by using schedules is only applicable to scheduling methods that are based on rolling horizon. Other common scheduling strategies in the industry such as dispatching rules, which do not generate schedules, can not be evaluated. This results in very little research value in comparing the relative performance of scheduling strategies across different classes. By using the long-term steady state performance as our benchmarking criteria, we are able to compare the relative performance of a wide spectrum of scheduling strategies and methods, ranging from optimizing-based approaches to dispatching rules.

The in-depth study on issues relating to rescheduling gives us a better understanding on the factors in rescheduling. This leads us to identify the work that the research community has done and what the gaps are. The study has also enabled us to set apart what are important in our event-driven rescheduling research from commonly found research problems in rescheduling.

CHAPTER 4

ANALYSIS OF EVENT-DRIVEN AND PERIODIC RESCHEDULING

4.1 Introduction

This chapter presents analytical models that compare the performance of periodic and event-driven scheduling strategies for discrete manufacturing systems, where different job (or entity) types arrive dynamically for processing. The former performs rescheduling periodically, whereas the latter reschedules on events. The environment is subject to random variations of processing time. The scheduling algorithm is based on job priority, where one or more types of jobs have priority over other classes. The models can be used to estimate performance measures such as mean waiting time of jobs in queue. The analytical models enable system performance to be estimated quickly without requiring too much effort in constructing simulation models and running experiments. Although this work considers simple queuing networks, the results can provide some important insights into more complex production systems.

The chapter is organized as follows: the next section describes the concepts of periodic and event-driven rescheduling. Section 4.3 describes a single-machine queuing model. Section 4.4 extends the single-machine queuing model to consider two-machine queuing model. Finally, Section 4.5 summarizes the chapter.

4.2 Concepts of Periodic and Event-Driven Rescheduling

For rolling horizon based rescheduling, a rescheduling policy is required. Three basic types of policies are: periodic, event-driven, and hybrid. A periodic policy reschedules production facility periodically (see Figure 4.1). For the event-driven policy, rescheduling occurs in reacting to production events. A hybrid rescheduling policy reschedules the system on major events in addition to the regular periodic rescheduling. In our analysis, we only consider periodic and event-driven rescheduling. The performance of hybrid rescheduling policy is assumed to be between periodic and event-driven scheduling policies.

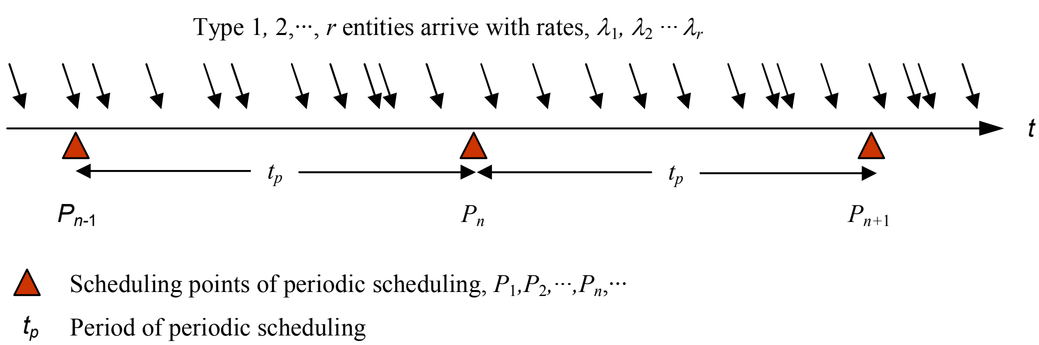


Figure 4.1: Periodic rescheduling

Consider a queuing system where there are r types of entities (jobs, orders, parts, users or customers), and type i entities arrive according to a Poisson stream with arrival rate $\lambda_i, i = 1,\cdots,r$. The service time (or process time) and residual service time of a type i entity are denoted by B_i and R_i respectively. The dispatching algorithm of the queue is based on priority, where type 1 entities have the highest priority; type 2 entities the second highest and so on. Entities of the same priority are serviced in order of arrival. We consider non-preemptive rule where higher priority entities may not interrupt the service time of lower priority entities, but have to wait till the service time of the low priority entity has completed.

For event-driven scheduling, we assume that whenever there is an arrival of a new entity, the entity will be prioritized with the existing entities that are already in the queue (if the queue is not empty). For periodic policy, rescheduling is triggered only at specific time points as depicted in Figure 4.1 ($P_{1,\ldots}P_{n\text{-}1}$, P_n, …). Entities that arrive between the scheduling points will not be prioritized until the next rescheduling point. Based on these assumptions, we can intuitively study the event-driven policy using single-arrival queuing model while the periodic scheduling policy using the group-arrival queuing model. In our study, we consider $M/G/1$ queuing models, which are a good approximation for our analysis since inter-arrival times of entities in discrete manufacturing system can often be considered as independent or "memoryless".

4.3 A Single-Machine Queuing Model

We shall assume from now on that, unless otherwise stated, the random variables for inter-arrival times, $X(i)$ are independent and identically distributed (i.i.d), with rate λ. Similarly, the random variables for service times, $B(i)$ are independent and identically distributed, with rate μ. The expected values of random variables X and B are:

$$E[X] = 1/\lambda$$
$$E[B] = 1/\mu$$

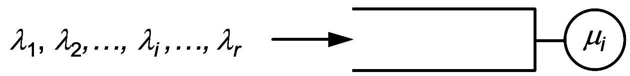
We shall also assume that if the queuing system is allowed to operate for a long time, it can reach equilibrium (or steady state). We define the quantity:

$$\rho = \text{"utilization ratio"}$$
$$= \frac{\text{effective rate of entity arrivals at the queuing system}}{\text{total available rate of service by the service facility}}$$

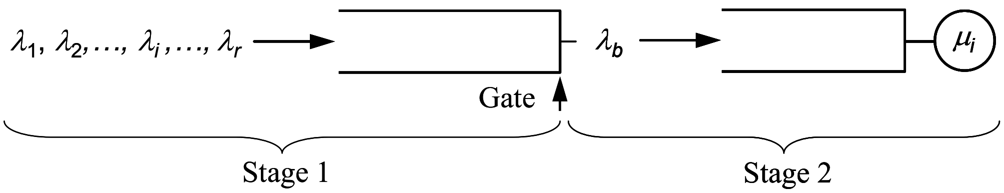
From the definition, it is clear that for a single-machine queuing system:

$$\rho = \frac{\lambda}{\mu} = \lambda \cdot E[B] \tag{4.1}$$

Figure 4.2 shows the single-machine queuing models for both single arrival (Figure 4.2a) and group arrival (Figure 4.2b). For the single-arrival queuing system, there are r types of entities, with type i entities arrive according to a Poisson stream with arrival rate $\lambda_i, i = 1,\cdots,r$. The entities will be serviced one by one according to priority, with type i entities being serviced with rate $\mu_i = 1,\cdots,r$. For the group-arrival queuing model, there are two stages involved. In the first stage, the arrivals of entities follow exactly the same as that of the single-arrival queuing system. However, the entities in the queue are not serviced, but will be released in a single group (or batch) to the second stage periodically. The queue in the first stage can be imagined as a gate that opens periodically. The gate closes immediately after all the entities which are waiting in the queue are transferred to the second queue. The transferring time from the first queue to the second queue is assumed to be 0. In the second stage, the entities are considered to arrive in groups with arrival rate λ_b, and the entities will be serviced one by one according to priority, with type i entities being serviced with rate $\mu_i = 1,\cdots,r$. We shall derive and compare the mean waiting time of a type i entity for event-driven and periodic scheduling policies by considering single-arrival and group-arrival queuing models respectively.



(a) A single-arrival queuing model *M/G/*1



(b) A group-arrival queuing model *M^b/G/*1

Figure 4.2: A single-arrival and a group-arrival queuing systems

The following notations are defined and are applicable to both queuing models unless specified otherwise:

$E_S[W_i]$ = the mean waiting time of priority i entities in the queue for single-arrival queuing model

$E_{G1}[W_i]$ = the mean waiting time of priority i entities in the queue at stage 1 for group-arrival queuing model

$E_{G2}[W_i]$ = the mean waiting time of priority i entities in the queue at stage 2 for group-arrival queuing model

$E_G[W_i]$ = the total mean waiting time of priority i entities for group-arrival queuing model

$E_S[L_i]$ = the mean number of priority i entities in the queue for single-arrival queuing model

$E_{G2}[L_i]$ = the mean number of priority i entities in the queue at stage 2 for group-arrival queuing model

r = the number of types of entities

b_i = the number of type i entities in the group for group-arrival queuing model

λ_i = the arrival rate of priority i entities

λ_b = the arrival rate of group of entities at stage 2 for group-arrival queuing model

t_p = the inter-arrival time of group of entities at stage 2 for group-arrival queuing model

μ_i = the service rate of priority i entities

$E[B_i]$ = the mean service time of priority i entities, and thus $E[B_i] = 1/\mu_i$

$E[R_i]$ = the mean residual service time of priority i entities

ρ_i = the utilization ratio of priority i entities

In the models, we shall only consider steady-state condition and hence we have:

$$\sum_{j=1}^{r} \rho_j = \sum_{j=1}^{r} \frac{\lambda_j}{\mu_j} \leq 1 \tag{4.2}$$

In the following subsections, we shall first derive an analytical model for a single-arrival queuing model, followed by a group-arrival queuing model. Subsequently, we shall compare the models.

4.3.1 A Single-Arrival Queuing Model *M*/*G*/1

Consider a single-arrival queuing system (refer to Figure 4.2a) with *r* types of entities, we have:

Lemma 4.1:

For $\sum_{j=1}^{r} \rho_j \leq 1$, the mean waiting time $E_S[W_i]$ of a type i entity is given by the following expression:

$$E_S[W_i] = \frac{\sum_{j=1}^{i-1} E_S[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i} \rho_j} \qquad \textbf{(4.3)}$$

Proof 4.1:

For the highest priority (i.e. priority 1) entities, it holds that (refer to Appendix C for derivation of the expression):

$$E_S[W_1] = E_S[L_1] \cdot E[B_1] + \sum_{j=1}^{r} \rho_j E[R_j] \qquad \textbf{(4.4)}$$

According to Little's law [Little, 1961], we have:

$$E_S[L_1] = \lambda_1 E_S[W_1] \qquad \textbf{(4.5)}$$

Combining (4.4) and (4.5), and noting that $\rho_1 = \lambda_1 E[B_1]$ yields:

$$E_S[W_1] = \frac{\sum_{j=1}^{r} \rho_j E[R_j]}{1 - \rho_1} \qquad \textbf{(4.6)}$$

To determine the mean waiting time for the lower priority entities, consider a type i entity with $i > 1$. The waiting time of the entity can be divided into a number of portions. The first portion X_1, is the amount of work associated with the entity in service and all entities with the same or higher priority present in the queue upon its arrival. The second portion X_2, is the amount of higher priority work arriving during X_1. Subsequently the third portion X_3 is the amount of higher priority work arriving during X_2, and so on. Hence the mean waiting time of a type i entity ($i > 1$) is given by:

$$E_S[W_i] = E[X_1 + X_2 + X_3 + \cdots] = \sum_{k=1}^{\infty} E[X_k] \tag{4.7}$$

Clearly, X_{k+1} depends on X_k. During X_k period, there are approximately $\sum_{j=1}^{i-1} \lambda_j X_k$ new entities of higher priority than i arrive, with the sum of service times of $\sum_{j=1}^{i-1} \lambda_j X_k E[B_j]$. Hence we have:

$$X_{k+1} = \left(\sum_{j=1}^{i-1} \lambda_j E[B_j] \right) \cdot X_k = \left(\sum_{j=1}^{i-1} \rho_j \right) \cdot X_k \tag{4.8}$$

Taking expectation of (4.8), we have:

$$E[X_{k+1}] = \left(\sum_{j=1}^{i-1} \rho_j \right) \cdot E[X_k] \tag{4.9}$$

Repeated application of the relation (4.9) yields:

$$E[X_{k+1}] = \left(\sum_{j=1}^{i-1} \rho_j \right)^k \cdot E[X_1], \qquad k = 0, 1, 2, \cdots \tag{4.10}$$

Substitute (4.10) into (4.7), we have:

$$E_S[W_i] = \sum_{k=1}^{\infty} \left(\sum_{j=1}^{i-1} \rho_j \right)^{k-1} \cdot E[X_1] \tag{4.11}$$

But the term $\sum_{k=1}^{\infty} \left(\sum_{j=1}^{i-1} \rho_j \right)^{k-1}$ in (4.11) is an infinite arithmetic series which can be further simplified to $1 \Big/ \left(1 - \sum_{j=1}^{i-1} \rho_j \right)$. Hence we have:

$$E_S[W_i] = \frac{E[X_1]}{1 - \sum_{j=1}^{i-1} \rho_j} \tag{4.12}$$

As mentioned earlier, the first portion of work an arriving type i entity has to wait for is the sum of the service times of all entities with the same or higher priority present in the queue plus the remaining service time of the entity in service. Therefore, we have:

$$E[X_1] = \sum_{j=1}^{i} E_S[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j] \tag{4.13}$$

Substitute (4.13) into (4.12), we have:

$$E_S[W_i] = \frac{\sum_{j=1}^{i} E_S[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i-1} \rho_j} \tag{4.14}$$

Applying Little's law $E_S[L_i] = \lambda_i E_S[W_i]$, noting that $\rho_i = \lambda_i E_S[B_i]$ and rearranging (4.14), we have:

$$\begin{aligned} E_S[W_i] &= \frac{\lambda_i E_S[W_i] \cdot E[B_j] + \sum_{j=1}^{i-1} E_S[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i-1} \rho_j} \\ &= \frac{\sum_{j=1}^{i-1} E_S[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i} \rho_j} \end{aligned}$$

Thus we arrive at expression (4.3), the mean waiting time of type i entity in the queue for single-arrival queuing model.

4.3.2 A Group-Arrival Queuing Model $M^b/G/1$

We now consider *M/G/1* priority queue where entities do not arrive one by one, but in groups with *r* types of entities. Similar to single-arrival queuing model, type 1 entities have the highest priority. The groups arrive according to a Poisson process with rate

λ_b. The number of type i entities in the group is b_i. We shall derive the mean waiting time of type i entities in the group. Further, we define $\rho_i = \lambda_b b_i E[B_i]$.

Lemma 4.2:

For $\sum_{j=1}^{r} \rho_j \leq 1$, the mean waiting time of a type i entity for group-arrival queuing model is given by the following expression:

$$E_G[W_i] = E_{G1}[W_i] + E_{G2}[W_i] \quad \textbf{(4.15)}$$

Where,

$$E_{G1}[W_i] = \frac{t_p}{2} \quad \textbf{(4.16)}$$

$$E_{G2}[W_i] = \frac{\sum_{j=1}^{i-1} E_{G2}[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j] + \sum_{j=1}^{i-1} b_j E[B_j] + \frac{1}{2}(b_i - 1)E[B_i]}{1 - \sum_{j=1}^{i} \rho_j} \quad \textbf{(4.17)}$$

Proof 4.2:

For the group-arrival queuing model, type i entities need to wait at the first queue before moving to the second queue. The number of type i entities in the first queue at time t is depicted in Figure 4.3.

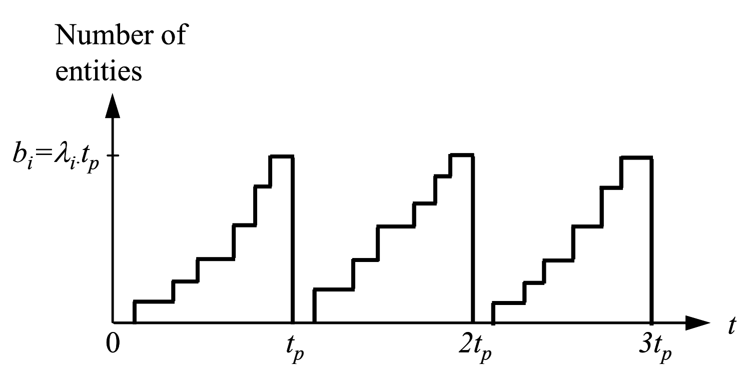


Figure 4.3: Number of entities in the first queue of group-arrival queuing model

The total waiting time for type i entities from time $(n-1)\cdot t_p$ to $n\cdot t_p$, where $n=1,2,\cdots\infty$, is given by the area under the curve. The mean waiting time for type i entities is given by the total waiting time divided by the number of type i entities during the period ($=b_i$). Since the total area from time $(n-1)\cdot t_p$ to $n\cdot t_p$ is given by $(t_p b_i)/2$ and thus, we have:

$$E_{G1}[W_i]=\frac{\frac{1}{2}t_p b_i}{b_i}=\frac{t_p}{2}$$

Hence we have derived (4.16) for expression (4.15), the mean waiting time of type i entity in the first queue for group-arrival queuing model. The procedure to determine the mean waiting time for a type i entity in the second queue, where $i>1$, is similar to the approach used in the single-arrival queuing model. Using expression (4.12) and considering $\rho_i=\lambda_b b_i E[B_i]$, we have:

$$E_{G2}[W_i]=\frac{E_{G2}[X_1]}{1-\sum_{j=1}^{i-1}\rho_j} \qquad \textbf{(4.18)}$$

Where,

$$E_{G2}[X_1]=\sum_{j=1}^{i}E_{G2}[L_j]\cdot E[B_j]+\sum_{j=1}^{r}\rho_j E[R_j]+\sum_{j=1}^{i-1}b_j E[B_j]+\sum_{k=1}^{b_i}p_k\cdot(k-1)E[B_i] \qquad \textbf{(4.19)}$$

The first two terms at the right-hand side of (4.19) correspond to the mean waiting time of the whole group in the queue. The second last term is the mean waiting time due to servicing of higher priority members in the group and the last term represents the mean waiting time of servicing members in its own group. p_k is the probability that a type i entity under consideration is in position k in its own group. Since there are b_i number of type i entities in the group, we have:

$$p_k = \frac{1}{b_i}$$

Taking expectation on the last term in (4.19), we have:

$$\begin{aligned}\sum_{k=1}^{b_i} p_k \cdot (k-1) \cdot E[B_i] &= \frac{1}{b_i} E[B_i] \cdot \sum_{k=1}^{b_i} (k-1) \\ &= \frac{1}{b_i} E[B_i] \cdot \frac{1}{2} b_i (b_i - 1) \\ &= \frac{1}{2}(b_i - 1) E[B_i]\end{aligned} \qquad \textbf{(4.20)}$$

Substitute (4.20) into (4.19) and then (4.19) into (4.18), we have:

$$E_{G2}[W_i] = \frac{\sum_{j=1}^{i} E_{G2}[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j] + \sum_{j=1}^{i-1} b_j E[B_j] + \frac{1}{2}(b_i - 1) E[B_i]}{1 - \sum_{j=1}^{i-1} \rho_j} \qquad \textbf{(4.21)}$$

Applying Little's law $E_{G2}[L_i] = \lambda_b b_i E_{G2}[W_i]$ and rearranging (4.21), we have:

$$E_{G2}[W_i] = \frac{\sum_{j=1}^{i-1} E_{G2}[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j] + \sum_{j=1}^{i-1} b_j E[B_j] + \frac{1}{2}(b_i - 1) E[B_i]}{1 - \sum_{j=1}^{i} \rho_j}$$

Hence we arrive at (4.17) for expression (4.15), the mean waiting time of a type i entity in the second queue for group-arrival queuing model.

4.3.3 Comparison of Single-Arrival and Group-Arrival

To compare single-arrival (i.e. event-driven rescheduling policy) and group-arrival (i.e. periodic rescheduling policy) queuing models, we must have the same arrival rates for the entities in both queuing models. We assume mean inter-arrival time for groups at stage 2 of group-arrival queuing model to be t_p, which corresponds to the periodic rescheduling period. Thus, we have:

$$\lambda_b = \frac{1}{t_P} \tag{4.22}$$

Applying (4.22) and $b_i = \lambda_i t_P$, we have:

$$\rho_i = \lambda_b b_i E[B_i] = \frac{1}{t_P} \cdot \lambda_i t_P \cdot E[B_i] = \lambda_i E[B_i]$$

Which means that ρ_i is the same for both single-arrival and group-arrival queuing models under steady-state condition. Substituting $b_i = \lambda_i t_P$ into (4.17) and rearranging, we have:

$$\begin{aligned} E_{G2}[W_i] &= \frac{\sum_{j=1}^{i-1} E_{G2}[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i} \rho_j} + \\ &\quad \frac{\left(\sum_{j=1}^{i-1} \lambda_j E[B_j]\right) \cdot t_P + \frac{1}{2}(\lambda_i t_P - 1) E[B_i]}{1 - \sum_{j=1}^{i} \rho_j} \\ &= \frac{\sum_{j=1}^{i-1} E_{G2}[L_j] \cdot E[B_j] + \sum_{j=1}^{r} \rho_j E[R_j]}{1 - \sum_{j=1}^{i} \rho_j} + \frac{\left(2 \cdot \sum_{j=1}^{i} \rho_j - \rho_i\right) \cdot t_P - E[B_i]}{2 \cdot \left(1 - \sum_{j=1}^{i} \rho_j\right)} \end{aligned} \tag{4.23}$$

To compare single-arrival and group-arrival queuing models, we shall prove the following lemma:

Lemma 4.3:

For $\sum_{j=1}^{r} \rho_j \leq 1$, the mean waiting time of a type i entity for group-arrival queuing model is given by the following expression:

$$E_{G2}[W_i] \geq E_S[W_i] + \frac{\left(2 \cdot \sum_{j=1}^{i} \rho_j - \rho_i\right) \cdot t_P - E[B_i]}{2 \cdot \left(1 - \sum_{j=1}^{i} \rho_j\right)} \qquad \textbf{(4.24)}$$

Where $E_S[W_i]$ is the mean waiting time for a type i entity for single-arrival queuing model.

Proof 4.3:

To derive (4.24), we need to prove that:

$$\sum_{j=1}^{i-1} E_{G2}[L_j] \cdot E[B_j] \geq \sum_{j=1}^{i-1} E_S[L_j] \cdot E[B_j] \qquad \textbf{(4.25)}$$

Where $E_{G2}[L_i]$ and $E_S[L_i]$ are the mean number of type i entities in the queue under steady-state condition for group-arrival and single-arrival queuing models respectively. The terms $\sum_{j=1}^{r} \rho_j E[R_j]$ and $1 - \sum_{j=1}^{i} \rho_j$ of (4.3) and (4.23) are effectively the same for both single-arrival and group-arrival queuing models under the stated condition. This is because the mean residual service time of priority i entities, $E[R_i]$ is given by (refer to Appendix C for derivation):

$$E[R_i] = \frac{E[B_i^{\,2}]}{2E[B_i]}$$

Hence, the terms $\sum_{j=1}^{r} \rho_j E[R_j]$ and $1-\sum_{j=1}^{i} \rho_j$ are only dependent on ρ_i, $E[B_i]$ and $E[B_i^{\,2}]$, which are considered to be the same for both single-arrival and group-arrival queuing models.

Consider the mean waiting time of priority 1 entities for group-arrival queuing model from expression (4.17) and replacing i with 1, we have:

$$E_{G2}[W_1] = \frac{\sum_{j=1}^{r} \rho_j E[R_j]}{1-\rho_1} + \frac{\frac{1}{2}(b_1 - 1)E[B_1]}{1-\rho_1} \tag{4.26}$$

Comparing (4.26) to (4.6), which is the mean waiting time of priority 1 entities for single-arrival queuing model, we have $E_{G2}[W_1] \geq E_S[W_1]$ since the second term of (4.26) is greater than or equal to 0. Applying Little's law, $E_{G2}[L_1] = \lambda_1 E_{G2}[W_1]$ and $E_S[L_1] = \lambda_1 E_S[W_1]$, it can thus be shown that $E_{G2}[L_1] \geq E_S[L_1]$. Applying the same principle on (4.3) and (4.17), but considering priority 2 entities with $i = 2$, we can prove that $E_{G2}[W_2] \geq E_S[W_2]$ and thus $E_{G2}[L_2] \geq E_S[L_2]$ (by Little's law) since $E_{G2}[L_1] \geq E_S[L_1]$. By repeated application of the principle but for lower priority entities, we can thus show that in general, $E_{G2}[L_i] \geq E_S[L_i]$. This implies that expression (4.25) holds, and thus lemma 4.3 must be true. This further leads to $E_G[L_i] \geq E_S[L_i]$.

The second term at the right-hand side of expression (4.24) increases linearly with t_P, as ρ_i and $E[B_i]$ are constants under steady-state condition. This implies that event-driven rescheduling outperforms (i.e. with lower mean waiting time) periodic rescheduling when $t_P > 0$. This observation can be reasoned from the fact that in periodic rescheduling, entities that arrive before the rescheduling points need to wait until the next rescheduling point, and also wait for the higher priority entities in the group to be processed first.

Further, if the rescheduling period t_P becomes very small (i.e. approaches 0 but not 0), $E_{G1}[W_i]$ (refer to expression (4.16)) and the second term at the right-hand side of the expression (4.24) approaches 0 since $b_i(=\lambda_i t_P)$ approaches 1. This effectively means that the performance of periodic rescheduling moves towards to that of event-driven rescheduling when rescheduling period t_P approaches 0. It should be noted that when rescheduling period t_P approaches 0, b_i can also be 0. However, $b_i = 0$ implies that no entity arrives into the queuing system, and thus periodic rescheduling will generate the same schedule as the previous schedule.

4.4 A Two-Machine Queuing Model

We consider two-machine queuing model where entities arriving into one queue can be redirected to the second queue in the model as depicted in Figure 4.4. This model can be used to represent event-driven rescheduling scenario where available capacity of a second machine (or server) can be utilized to process entities arriving to the first queue. For periodic rescheduling policy, the opportunity of using the available capacity of the second server may be lost due to periodic nature of the policy.

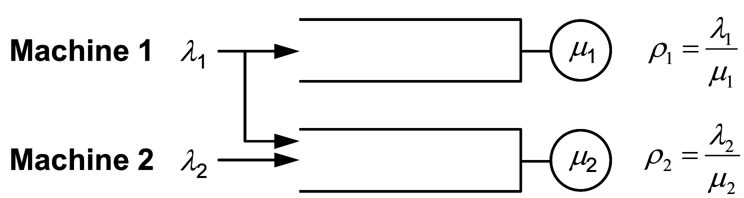


Figure 4.4: A two-machine queuing model

The combined queuing model can be used to represent a typical combination of a bottleneck and a non-bottleneck machine in practice. To reduce the workload of the bottleneck, the entities that arrive at the bottleneck can be rescheduled onto the non-bottleneck whenever the non-bottleneck machine is in idle status. This rescheduling strategy will not therefore affect the second machine significantly since only its idle time is being utilized (i.e. provided that the service time of each additional entity does not exceed the idle period of machine 2 significantly). We will first study the single-arrival (i.e. event-driven rescheduling) queuing model in Section 4.4.1 before proceeding to group-arrival queuing model (i.e. periodic rescheduling) in Section 4.4.2. We then analyze both queuing models in Section 4.4.3. In order not to complicate the models, we shall focus on one type of entities.

4.4.1 A Single-Arrival Queuing Model

Suppose that entities arrive at machines 1 and 2 according to Poisson streams with rates λ_1 and λ_2 respectively. The service time of machine 1 is denoted by $E[B_1]$. Similarly, $E[B_2]$ is the service time for machine 2. We shall show the following lemma:

Lemma 4.4:

For $\rho_1 \le 1$ and $\rho_2 \le 1$,

$$\rho_1^{k+1} = \rho_1^k - \min\left\{ \begin{array}{l} \left[\dfrac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k \right) \right], \\ \left(\rho_1^k \right)^2 \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \dfrac{E[B_2]}{E[B_1]} \right) \right] \end{array} \right\}, \ k = 0,1,2,\cdots \quad \textbf{(4.27)}$$

$$\rho_2^{k+1} = \rho_2^k + \min\left\{ \left(1-\rho_2^k\right), \left(C \cdot \frac{E[B_2]^2}{E[B_1]} - \rho_2^k \frac{E[B_2]}{E[B_1]} \right)^2 \left(1-\rho_2^k\right) \right\},\ k = 0,1,2,\cdots \tag{4.28}$$

Where,

C = constant

ρ_1 = utilization of machine 1

ρ_2 = utilization of machine 2

Proof 4.4:

To derive an expression for the mean waiting time of entities in queue for machine 1, we evaluate what is seen by an arriving entity at the queue of machine 1. There are four possible states relating to the machines, based on the combination of busy and idle status of the two machines (i.e. 2^2 = 4), on the arrival of the entity. The probability that the machine is busy on entity arrival is equal to the fraction of time that the machine is busy (i.e. by PASTA property, refer to Appendix B). This is simply given by the utilization of the machine, ρ, and the fraction of time that the machine is idle is thus $1-\rho$. The 4 combined states of probabilities for machine status are thus given by:

$\rho_1\rho_2$ = the probability that both machines are busy

$\rho_1(1-\rho_2)$ = the probability that machine 1 is busy and machine 2 is idle

$(1-\rho_1)\rho_2$ = the probability that machine 1 is idle and machine 2 is busy

$(1-\rho_1)(1-\rho_2)$ = the probability that both machines are idle

For the proposed rescheduling policy, the entities on arrivals will be rescheduled from machine 1 to machine 2 under the conditions that machine 1 is busy and machine 2 is idle. However when entities arriving at queue 1 are being rescheduled to machine 2, the utilizations of both machines will change (i.e. utilization of machine 1 decreases and utilization of machine 2 increases). Therefore, new rates may need to be re-computed (since the probability model only considers $\mu_1 = \mu_2$). But these changes are recursive in nature, and if these go on, the utilization rates will reach a steady-state condition. Alternatively, we can set a limit on the number of entities that can be rescheduled per unit time so as not to overload machine 2.

To derive the new rates, we need to know the number of entities that will be rescheduled from queuing system 1 to queuing system 2 per unit time, λ_N. The probability that an arriving entity is in the period of machine 1 busy and machine 2 idle is given by $\rho_1(1-\rho_2)$. Hence, the maximum number of arriving entities that can be rescheduled per unit time is:

$$\lambda_N = \lambda_1 \rho_1 (1-\rho_2) \tag{4.29}$$

Thus, the new effective rates of arrivals for queue 1 and 2 become:

$$\lambda_1^* = \lambda_1 - \lambda_N = \lambda_1 - \lambda_1 \rho_1 (1-\rho_2) \tag{4.30}$$
$$\lambda_2^* = \lambda_2 + \lambda_N = \lambda_2 + \lambda_1 \rho_1 (1-\rho_2) \tag{4.31}$$

Suppose we keep the number of entities rescheduled per unit time fixed at λ_N for a very long time, say T. We can compute the new utilization rates, ρ_1^* and ρ_2^* by considering the fraction of time the machine is busy. During the long period of time T, we would expect there have been $\lambda_1^* T$ and $\lambda_2^* T$ entities arriving to the queuing systems 1 and 2 respectively, each of which takes on average $E[B_1]$ and $E[B_2]$ units of time to be served respectively. This means:

$$\rho_1^* = \frac{\textit{units of time machine is busy}}{T} = \frac{\lambda_1^* T \cdot E[B_1]}{T} = \lambda_1^* E[B_1] \tag{4.32}$$

$$\rho_2^* = \frac{\lambda_2^* T \cdot E[B_2]}{T} = \lambda_2^* E[B_2] \tag{4.33}$$

Substituting (4.30) into (4.32) and (4.31) into (4.33), we have:

$$\rho_2^{\ *} = [\lambda_1 - \lambda_1\rho_1(1-\rho_2)]\cdot E[B_1] \tag{4.34}$$
$$\rho_2^{\ *} = [\lambda_2 + \lambda_1\rho_1(1-\rho_2)]\cdot E[B_2] \tag{4.35}$$

When both queuing systems reach equilibrium, we apply the rescheduling policy again. Repeated application of the procedure, a recursive relation can be derived by inspecting equations (4.30) and (4.31):

$$\lambda_1^{k+1} = \lambda_1^k - \lambda_1^k\rho_1^k\left(1-\rho_2^k\right) \tag{4.36}$$
$$\lambda_2^{k+1} = \lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right) \tag{4.37}$$

Similarly by inspecting (4.34) and (4.35), we get:

$$\rho_1^{k+1} = \left[\lambda_1^k - \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\cdot E[B_1] \tag{4.38}$$
$$\rho_2^{k+1} = \left[\lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\cdot E[B_2] \tag{4.39}$$

Clearly the limit $\rho_2^{k+1} < 1$ must be applied. Substituting inequality $\rho_2^{k+1} < 1$ into (4.39), we obtain $\left[\lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\cdot E[B_2] < 1$. However from (4.31), we have $\lambda_2 + \lambda_N = \lambda_2 + \lambda_1\rho_1(1-\rho_2)$, which also means that $\lambda_2^k + \lambda_N^k = \lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)$. Replacing the term $\left[\lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]$ in $\left[\lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\cdot E[B_2] < 1$ by term $\lambda_2^k + \lambda_N^k$ using expression $\lambda_2^k + \lambda_N^k = \lambda_2^k + \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)$, we therefore obtain the expression $\left(\lambda_2^k + \lambda_N^k\right)\cdot E[B_2] < 1$ or $\lambda_N^k < \mu_2 - \lambda_2^k$, where $\mu_2 = 1/E[B_2]$. Substitute the condition into (4.38) and (4.39), we have:

$$\rho_1^{k+1} = \left\{\lambda_1^k - \min\left[\mu_2 - \lambda_2^k, \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\right\}\cdot E[B_1] \tag{4.40}$$
$$\rho_2^{k+1} = \left\{\lambda_2^k + \min\left[\mu_2 - \lambda_2^k, \lambda_1^k\rho_1^k\left(1-\rho_2^k\right)\right]\right\}\cdot E[B_2] \tag{4.41}$$

Adding equations (4.36) and (4.37) up, we have:

$$\lambda_1^{k+1} + \lambda_2^{k+1} = \lambda_1^k + \lambda_2^k = \cdots = \lambda_1 + \lambda_2 = C \tag{4.42}$$

Where C is a constant. This is expected as the sum of arrivals per unit time for the two queuing systems are not affected by the rescheduling policy (i.e. conservation law of material).

Expressing (4.40) and (4.41) in terms of ρ by using (4.42) and $\rho = \lambda E[B]$, we thus have:

$$\rho_1^{k+1} = \rho_1^k - \min\left\{ \begin{array}{l} \left[\dfrac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k\right) \right], \\ \left(\rho_1^k\right)^2 \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \dfrac{E[B_2]}{E[B_1]} \right) \right] \end{array} \right\}, \; k = 0,1,2,\cdots$$

$$\rho_2^{k+1} = \rho_2^k + \min\left\{ \left(1 - \rho_2^k\right), \left(C \cdot \frac{E[B_2]^2}{E[B_1]} - \rho_2^k \frac{E[B_2]}{E[B_1]} \right)^2 \left(1 - \rho_2^k\right) \right\}, \; k = 0,1,2,\cdots$$

We thus arrive at recursive expressions (4.27) and (4.28) that can be used to compute utilization rates of both machines under event-driven rescheduling policy for the two-machine queuing model. The expressions are only dependent on the initial utilization rates of the machines while other parameters are constant.

Plotting expressions (4.27) and (4.28) on graphs for different input values (refer to Table 4.1) shows that when the utilization of machine 1 is high (data sets 1 and 2), the system reaches steady-state quickly (refer to Figures 4.5 and 4.6). However, when the utilization of machine 1 is lower (data sets 3 and 4), the system requires longer time to reach the steady state (refer to Figures 4.7 and 4.8). From the plots, it is clear that the analytical model always makes full use of available capacity of machine 2.

Table 4.1: Data sets for plots

Inputs	**Data set 1**	**Data set 2**	**Data set 3**	**Data set 4**
λ_2/λ_1	0.5	0.8	0.75	0.67
$E[B_2]/E[B_1]$	1.0	1.0	1.0	1.0
ρ_1	1.0	1.0	0.8	0.3
ρ_2	0.5	0.8	0.6	0.2

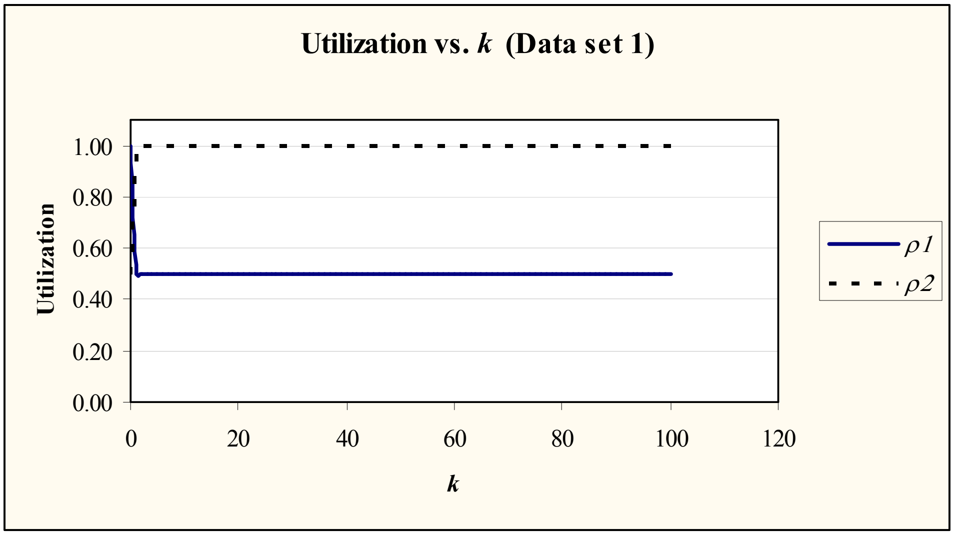


Figure 4.5: Utilization versus *k* plot for data set 1

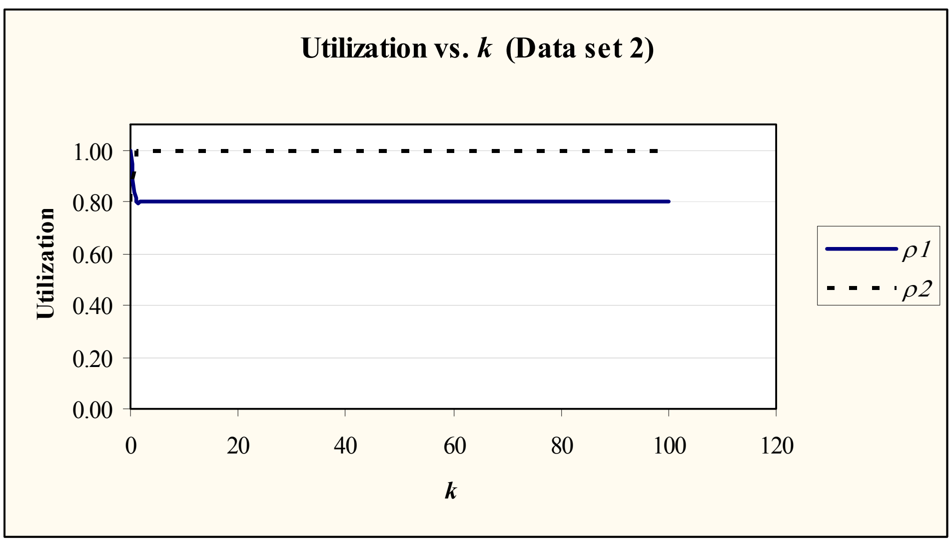


Figure 4.6: Utilization versus *k* plot for data set 2

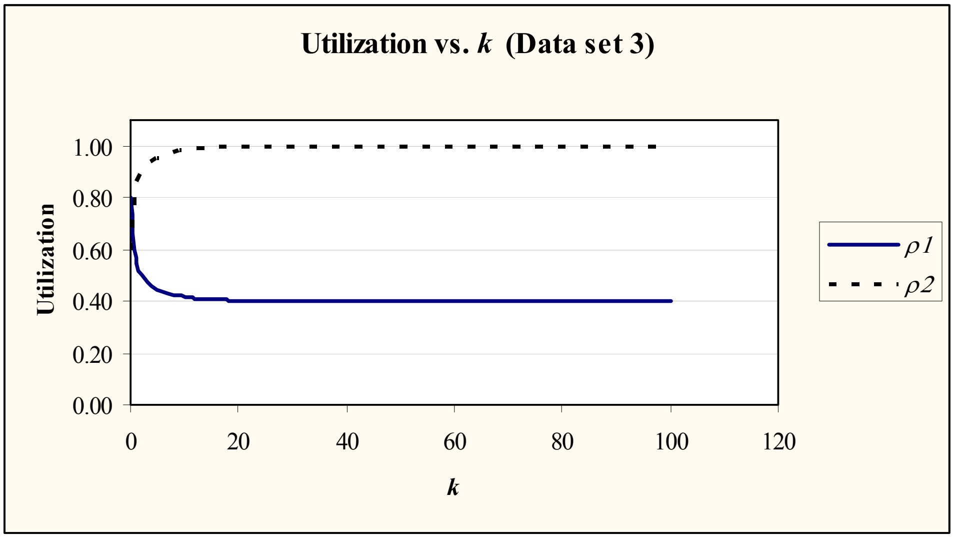


Figure 4.7: Utilization versus *k* plot for data set 3

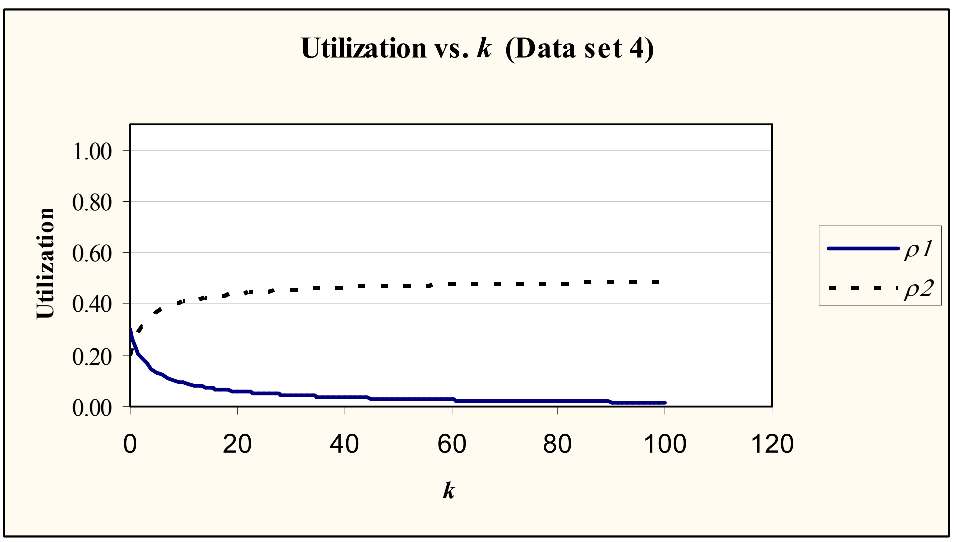


Figure 4.8: Utilization versus *k* plot for data set 4

The mean waiting times in queue for the entity that arrives at machines 1 and 2:

$$E^k[W_1] = E^k[L_1]\cdot E[B_1] + \rho_1^k E[R_1] \quad \textbf{(4.43)}$$
$$E^k[W_2] = E^k[L_2]\cdot E[B_2] + \rho_2^k E[R_2] \quad \textbf{(4.44)}$$

Where,

$E[L]$ = mean number of entities for queue 1 or 2

$E[R]$ = mean residual processing time for machine 1 or 2

Substitute Little's law, $E[L] = \lambda \cdot E[W]$ into (4.43) and (4.44) and rearranging, we have:

$$E^k[W_1] = \frac{\rho_1^k E[R_1]}{1-\rho_1^k} \tag{4.45}$$

$$E^k[W_2] = \frac{\rho_2^k E[R_2]}{1-\rho_2^k} \tag{4.46}$$

Based on the derivation in Appendix C, we obtain the following expressions for (4.45) and (4.46):

$$E^k[W_1] = \frac{\left(1 + C_{B_1}{}^2\right)}{2} \cdot \frac{\rho_1^k}{1-\rho_1^k} \cdot E[B_1] \tag{4.47}$$

$$E^k[W_2] = \frac{\left(1 + C_{B_2}{}^2\right)}{2} \cdot \frac{\rho_2^k}{1-\rho_2^k} \cdot E[B_2] \tag{4.48}$$

Where,

C_B = coefficient of variation of processing times for machine 1 or 2

In order for the rescheduling policy to have the least effect on the performance of the queuing system 2, only its idle period should be used for processing additional entities from queuing system 1. The length of an idle period of queuing system 2 is, on average, $1/\lambda_2$ since an idle period occurs when the machine is waiting for an entity to arrive after the queue becomes empty. Since the arrival process is Poisson and therefore memoryless, the machine will wait a negative exponentially distributed amount of time until the next entity arrives. Thus when an entity arrives at queuing system 1 while machine 2 is idle, the mean remaining idle time should be $1/(2\lambda_2)$. The additional entity to be scheduled onto the idle period of machine 2 should thus satisfy:

$$E[B] \le \frac{1}{2\lambda_2}$$

Therefore, the number of entities that can be rescheduled from queue 1 to queue 2 is determined by the stated inequality.

4.4.2 A Group-Arrival Queuing Model

Consider group arrivals with group size b (i.e. periodic rescheduling) instead of single arrivals (i.e. event-driven rescheduling) on the queuing system 1, we shall show the following lemma:

Lemma 4.5:

For $\rho_1 \le 1$ and $\rho_2 \le 1$,

$$\rho_1^{k+1} = \rho_1^k - \min\left\{ \begin{aligned} &\left[\frac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k\right)\right], \\ &n^k \lambda_b \rho_1^k E[B_1] \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \frac{E[B_2]}{E[B_1]}\right)\right] \end{aligned} \right\} \quad \textbf{(4.49)}$$

$$\rho_2^{k+1} = \rho_2^k + \min\left\{\left(1 - \rho_2^k\right), n^k \lambda_b E[B_1] \cdot \left(C \cdot E[B_2] - \rho_2^k\right)\left(1 - \rho_2^k\right)\right\} \quad \textbf{(4.50)}$$

Where,

C = constant

n = the number of entities in the group arrivals that can be rescheduled onto queuing system 2

λ_b = group arrival rate

ρ_1 = utilization of machine 1

ρ_2 = utilization of machine 2

Proof 4.5:

From (4.29), we have:

$$\lambda_{N_b} = n\lambda_b \rho_1 (1-\rho_2) \tag{4.51}$$

As n ($n \le b$) is the number of entities in the group arrivals that can be rescheduled onto queuing system 2, it is reasonable to assume that n can not be more than b in order not to interfere the second queuing system significantly. Using similar derivation procedure as single-arrival case, we thus can show that:

$$\rho_1^{k+1} = \rho_1^k - \min\left\{ \begin{array}{l} \left[\dfrac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k \right) \right], \\ n^k \lambda_b \rho_1^k E[B_1] \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \dfrac{E[B_2]}{E[B_1]} \right) \right] \end{array} \right\}$$

$$\rho_2^{k+1} = \rho_2^k + \min\left\{ \left(1-\rho_2^k\right), n^k \lambda_b E[B_1] \cdot \left(C \cdot E[B_2] - \rho_2^k \right)\left(1-\rho_2^k\right) \right\}$$

4.4.3 Analysis of Periodic and Event-Driven Rescheduling

For comparison purposes, we are interested in the mean waiting time of queuing system of machine 1. We shall show the following:

Lemma 4.6:

For $\rho_1 \le 1$ and $\rho_2 \le 1$,

$$E_G^k[W] \ge E_S^k[W] \tag{4.52}$$

Where,

$E_G^k[W]$ = the mean waiting time of entities in group-arrival queuing model of machine 1

$E_S^k[W]$ = the mean waiting time of entities in single-arrival queuing model of machine 1

Proof 4.6

Re-arranging (4.27) and let ρ_S^{k+1} be the utilization of machine 1 for single-arrival queuing system, we have:

$$\rho_S^{k+1} = \rho_1^k - \min\left\{ \begin{array}{l} \left[\dfrac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k \right) \right], \\ \lambda_1^k \rho_1^k E[B_1] \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \dfrac{E[B_2]}{E[B_1]} \right) \right] \end{array} \right\} \quad \textbf{(4.53)}$$

Similarly, let ρ_G^{k+1} be the utilization of machine 1 for group-arrival queuing system, we have:

$$\rho_G^{k+1} = \rho_1^k - \min\left\{ \begin{array}{l} \left[\dfrac{E[B_1]}{E[B_2]} - \left(C \cdot E[B_1] - \rho_1^k \right) \right], \\ n^k \lambda_b \rho_1^k E[B_1] \cdot \left[1 - \left(C \cdot E[B_2] - \rho_1^k \dfrac{E[B_2]}{E[B_1]} \right) \right] \end{array} \right\} \quad \textbf{(4.54)}$$

Comparing (4.53) and (4.54), we deduce that $\rho_G^{k+1} \geq \rho_S^{k+1}$ since $\lambda_1^k \geq n^k \lambda_b$. From (4.47), we can conclude that $E_G^k[W] \geq E_S^k[W]$ since the term $\rho^k / \left(1 - \rho^k\right)$ increases (i.e. exponentially) with increasing ρ^k. This implies that event-driven rescheduling outperforms periodic rescheduling in the proposed rescheduling technique.

Further from (4.51), increasing the rescheduling period, t_P results in decreasing value of λ_b (since $\lambda_b = 1/t_P$). However, the number of entities, n that can be rescheduled for group arrivals is almost constant (i.e. n depends on the length of idle period in queuing system 2). Thus, the number of entities that can be rescheduled per unit time, λ_{N_b} decreases with increasing t_P. This results in increase of utilization (4.49) and subsequently the mean waiting time (4.47). In addition, from (4.47), an increase in variability of the queuing system (i.e. greater C_{B_1}) will also result in higher mean waiting time in the queuing system with greater rescheduling period, t_P.

4.5 Summary

We successfully compare the performance of periodic and event-driven rescheduling policies using queuing models. The performance measure is the mean waiting time of an entity arriving into queue. We derive two queuing models for comparison purposes. In these models, event-driven rescheduling policy performs rescheduling whenever a new entity arrives into the system, whereas periodic rescheduling only reschedules after every specific period. We model this characteristics by considering event-driven rescheduling policy as single-arrival queuing model while periodic rescheduling policy as group-arrival queuing model. The first model is an *M/G/1* with single-arrival queuing model and *r* types of entities. The second model is a two-machine queuing model, where entities arrive to the first queue can be rescheduled to the second queue for processing.

We show that event-driven rescheduling can achieve a better performance as compared to periodic rescheduling in terms of mean waiting time of entities (or jobs) in the models. Shorter waiting time implies shorter cycle time (or flow time) and queue length as waiting time is related to cycle time and queue length in a linear manner. The scheduling algorithm based on priority of entities can be considered as a general scheduling algorithm, whereby the priority of entities can be replaced with more specific attributes such as the processing times or remaining processing times of the entities. This study shows that rescheduling strategy can impact system performance, and hence it is important to consider appropriate planning and control strategies in manufacturing systems.

It should be noted that the objectives of mean cycle time and cycle time variance are conflicting in nature. The event-driven strategy will schedule higher priority entities in front of lower priority entities whenever higher priority entities arrive. This leads to longer cycle times for lower priority entities as the entities may not be processed for a very long time. On the contrary, the periodic strategy will only schedule on periodic basis, and may thus allow lower priority entities to be processed before the next rescheduling occurs. Therefore, periodic strategy may lead to lower variability of the cycle time as compared to event-driven strategy.

CHAPTER 5

AN OBJECT-ORIENTED SIMULATION TEST BED

5.1 Introduction

This chapter discusses the design and development of an object-oriented simulation test bed that facilitates the performance assessment of schedulers and job dispatchers for discrete manufacturing systems. The performance is related to dynamic and stochastic scheduling problems where jobs appear from time to time and machine breakdowns occur randomly.

The test bed emulates the dynamic and stochastic shop floor of discrete manufacturing systems using discrete-event simulation techniques, and enables different shop floor conditions, scheduling strategies and techniques to be incorporated and evaluated. Object-oriented modeling (OOM) and programming (OOP) techniques are used to develop the test bed.

Methodologies based on discrete-event simulation are attractive when dealing with discrete manufacturing systems [Enns, 2003]. A discrete-event simulation is one in which the state of a model changes at only discrete, and possibly random set of simulated time points. They allow both material and information to be modeled, as well as sophisticated decision logic for planning and control. Simulation can handle stochastic environments, dynamic demand patterns, capacity constrained resources, and so on [Hunter, 2002]. In fact, any degree of detail can be modeled, eliminating the need for simplifying assumptions and constraints [Mönch, 2003], and without actually manipulating, disrupting or building the model physically [Chen, 2002; Brown, 1997].

Object-oriented modeling (OOM) allows decomposition of a problem into a number of entities called objects and then builds data and functions around these objects. OOM techniques allow for an easy integration of association and aggregation relations between the real-world objects in a manufacturing system into the simulation test bed. The major advantages of OOM for manufacturing systems derive from its reusability, multiple levels of abstraction and common framework. Reusability of OOM comes directly from the object-oriented programming (OOP) advantages of inheritance, encapsulation, message passing and dynamic (or late) binding [Mize, 1992]. Multiple levels of abstraction allow quantity of information in a model to vary with the levels of abstraction [Benjamin, 1998]. A "low level abstraction" model contains more information than a "high level abstraction" model. Common framework enables a base representation of a system in OOP to be easily adapted and modified for a variety of designated purposes.

This chapter begins by describing the characteristics of a typical manufacturing system and the design criteria used for the simulation test bed [Sivakumar, 2000a;

Chong, 2002]. It then discusses the overall framework of the test bed, and then presents the application of OOP for the development of the test bed. Subsequently, the performance criteria, scheduling algorithms and dispatching rules that are incorporated into the test bed are described. This chapter finally ends with a summary.

5.2 Characteristics of a Typical Discrete Manufacturing System

In a typical discrete manufacturing system (see Figure 5.1), a production planning and control tool obtains production related information from a manufacturing execution system (MES), computes manufacturing control commands, and then sends the instructions to the shop floor, where the control actions are executed. The instructions consist of dispatching lists for machine centers (or machines). This information is used to determine which job to process next for each machine center. There are two approaches to generate the lists: dispatching or scheduling. In dispatching, a set of rules is applied to obtain the appropriate sequence of jobs for a specific machine center. An example is the first-in-first-out (FIFO) rule where the jobs are sequenced in the order of their arrival at the machine center. In scheduling, scheduling algorithms are applied to sequence available jobs on the shop floor so as to achieve desired production goals. The end results are also dispatching lists for the machine centers on the shop floor. The key difference between dispatching and scheduling is that dispatching rules normally use only local information relating to the particular machine center for sequencing jobs, whereas scheduling often considers all machine centers on the entire shop floor.

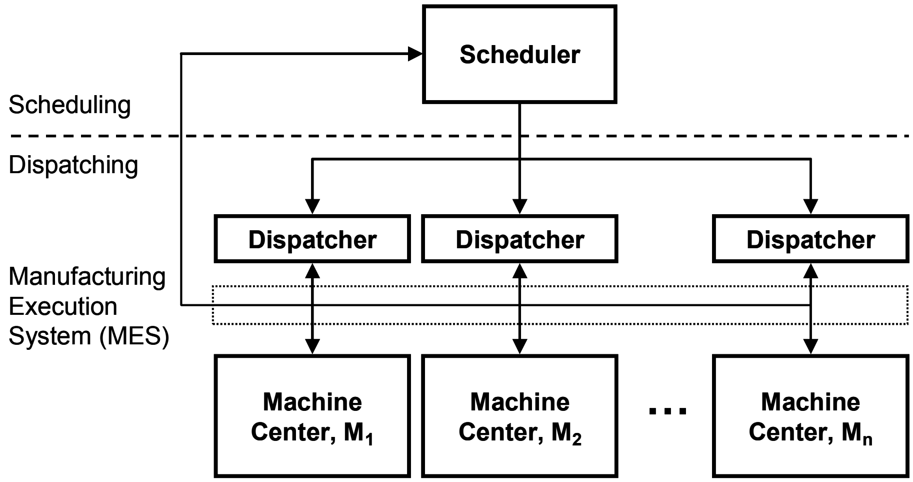


Figure 5.1: Architecture of production planning and control for a typical discrete manufacturing system

In a manufacturing system, jobs flow through a set of machine centers and the processes involved follow a "seize-hold-release" behavioral pattern. Machines can be down for repair when processing jobs. The jobs resume its processing for the remaining processing time when the machines are up again. The production system is considered to be dynamic and stochastic, with uncertainty in job arrivals, uncertainty in the reliability of the machines, and so on.

In order to assess the performance of a diverse set of scheduling algorithms and dispatching rules, the simulation test bed needs a common framework to support two major tasks: mimics the behavior of a dynamic and stochastic shop floor, and provides a common interface that allows incorporation of different scheduling algorithms and dispatching rules with minimum effort.

5.3 The Overall Framework

For our test bed, we propose a component-based framework which takes a "divide-and-conquer" approach to simulation modeling. The component-based approach sees a simulation as being composed of a set of components. The whole simulation is partitioned into a number of smaller simulation tasks, which are modeled by each component individually. The benefit of doing so is that the complexity is reduced. Each component is a set of smaller tasks whose internal logic is much simpler than that the whole simulation.

In the framework, we mimic the functional structure found in a manufacturing system. Manufacturing events are generated to imitate the shop floor events. An event is a happening that changes the state of a model (or system). For example, the arrival of new jobs and machine breakdowns are events that can be generated and simulated. Each job contains a number of operations to go through, and the operations are to be performed on the specific machine centers according to the step specification and requirement of the job. As in the real manufacturing systems, monitoring of job progress and tracking of machine status are performed by a tracking module. The information pertaining to jobs and machines is stored in a database for later scheduling and reporting purposes. The framework allows an event-driven and a time-driven triggering of the planning and control system. In the case of event-driven triggering, events of the shop floor trigger the generation of control information. These triggers can include new job arrivals, machine breakdowns or ups, etc. The planning and control system will then be called to compute new schedules. In the case of time-driven triggering, a timer starts the planning and control system periodically (e.g. at the beginning of a shift).

The requirements of the test bed are met through the development of five modules as depicted in Figure 5.2. These modules are scheduling, dispatching, event generation, manufacturing and tracking modules. Figure 5.3 shows how the five modules interact with each others.

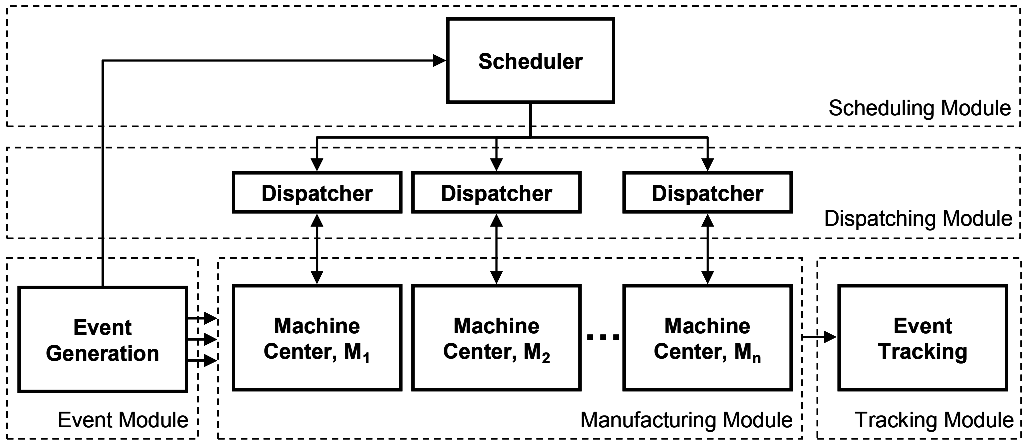


Figure 5.2: The five modules of the simulation test bed

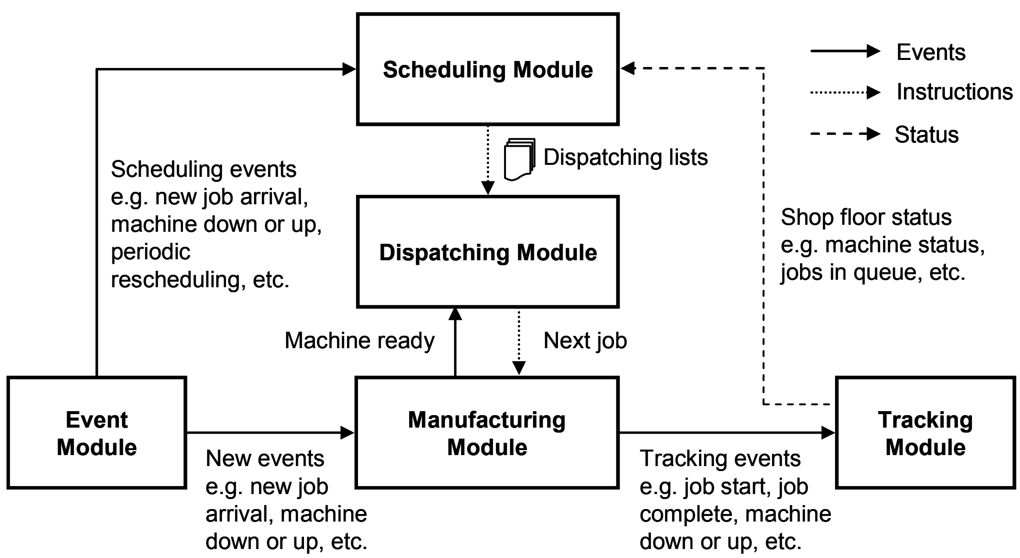


Figure 5.3: The interactions between the five modules of the simulation test bed

The scheduling module functions as a production scheduler in the test bed. The module generates schedules on what jobs to be processed at what time on which machine in the shop floor. The scheduling algorithms in the module require the process flow information (sequence, processing time, etc.) and shop floor status information such as current machine status. This information is provided by the tracking module. Scheduling is performed on regular basis, and can be periodically driven (e.g. every shift), event driven (e.g. new job arrivals), or a combination of both approaches. The generated schedules are transformed into dispatching lists for dispatchers and executed until the shop floor is rescheduled (see Figure 5.4). Thus scheduling occurs on a rolling horizon basis. A rolling schedule is formed by solving a multi-period problem and implementing only the first period's decisions; one period later the multi-period model is updated and the process repeated. For some rescheduling algorithms, previous schedules are required to generate revised schedules.

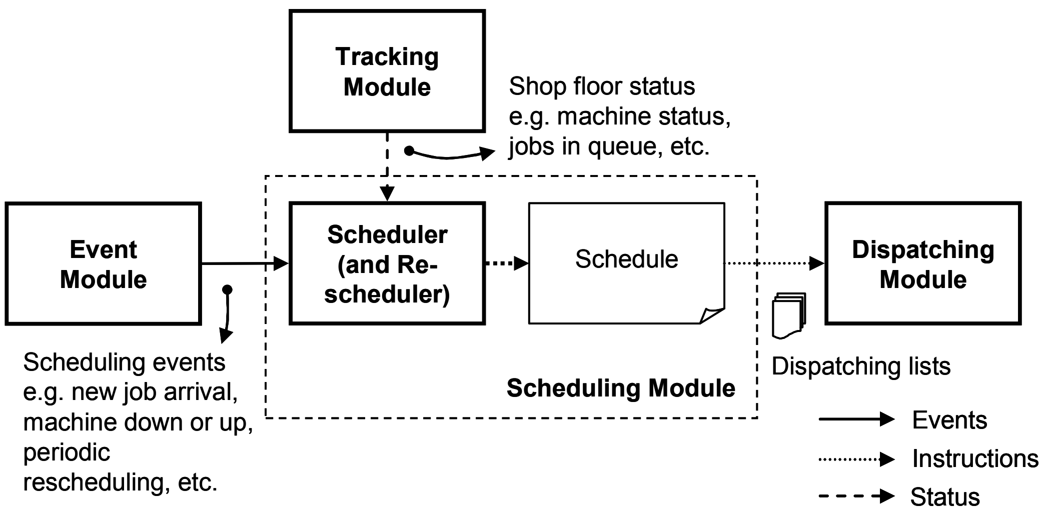


Figure 5.4: The structural view of the scheduling module and its interaction with other modules

The dispatching module operates as dispatchers for the machine centers. The dispatchers can either apply a set of rules or make use of the pre-computed dispatching lists of a scheduler to sequence jobs for a specific machine center. In the former, the dispatcher calculates a ranking for jobs waiting to be processed at the machine centers. The dispatcher can consider job attributes, machine status, and queuing information in its rules. Whenever a machine of a machine center becomes available, the next job is selected based on the rules. In the latter, the dispatcher follows the sequence of jobs as defined in the dispatching lists that are generated by a scheduler.

The event generation module mimics the dynamic and stochastic shop floor of manufacturing systems by generating events which can either be deterministic or stochastic. Deterministic events are mainly periodic events (e.g. every shift) for the scheduler to perform periodic scheduling, or scheduled events such as preventive maintenance. Stochastic events include new job arrivals, machine breakdowns and ups, etc. These events are used to drive both the scheduling and manufacturing modules. Upon receiving a new event, the scheduling module may perform rescheduling to compute new schedules for the shop floor. The manufacturing module responds to events and performs the appropriate tasks as in the actual manufacturing systems. For instance, in the case of an event related to a new job arrival, the job will be channeled to the relevant machine center based on the job's process routing.

The manufacturing module simulates material flow within the production facility. This module emulates the specific characteristics of a manufacturing system for the performance assessment of shop floor. The building blocks of manufacturing module consist of general elements found in a discrete manufacturing environment, such as machines, jobs, routings, etc. The inheritance feature of OOP can be utilized to create subclasses of these basic objects that more completely model the behavior of specific items. For example, the modeling of a general machine can be extended to such specific machine types as a batch machines such as ovens. These enhancements are readily achievable in the object-oriented paradigm because the more specific subclasses automatically inherit the capabilities of their more general parents.

The tracking module serves to collect data by tracking events that have taken place in the shop floor. These events include job arrivals, jobs joined machine queues, jobs started or finished processing on machines, machines broken down, and so on. This information is stored in a database and is later used to track shop floor status for rescheduling purposes, and to generate performance report.

The amount of variability in the manufacturing environments can be specified by changing the coefficient of variation in model configuration. For instance, uncertainty can be introduced with respect to the reliability of the machines. Machine reliability is specified in terms of time-to-failure and time-to-repair inputs for each capacity constrained resource.

The test bed is able to monitor both transient and long-term average behavior through the tracking module. This information can be useful for three types of applications. First, the performance of manufacturing systems using any planning and control logic can be better understood. This means that the characteristics of manufacturing systems such as demand patterns, scheduling algorithms or dispatching rules, processing features and so on can be treated as parameters which can then be experimentally controlled. Second, different planning and control scenarios can be compared. Comparisons under different scenarios allow interaction effects to be determined so that circumstances, under which each planning and control system performs, can be studied. Third, the test bed facilitates developing and comparing new algorithms for planning and control.

5.4 Application of Object-Oriented Programming

Object-oriented programming (OOP) techniques have been used to develop the test bed. The principal idea behind OOP is that all items in the system can be treated as objects. An object is an instance of a class, which is a software module that provides a complete definition of the capabilities of members of the class. These capabilities are provided by the methods (or procedures) and attributes (or data storage) contained within the immediate class definition or inherited from other class definitions to which this class is related. OOP embodies the four key concepts (inheritance, encapsulation, message passing and dynamic binding) which result in making software systems more understandable, modifiable, and reusable [Mize, 1992].

The core classes of the test bed and the interactions between them are shown in Figure 5.5. These classes form the foundation from which more specialized subclasses can be derived through inheritance. The OOP concept of dynamic binding is used throughout to implement the classes. Dynamic binding allows a variable to take different types dependent on the content at a particular time. This ability of a variable is called polymorphism. Polymorphism allows the core classes to be written based on the abstract interfaces of the objects which will be manipulated. Future extension in the form of new types of objects can then be done with ease by sharing the same common interface. These new objects are implemented as subclasses of the core classes.

The event manager contains simulation clock and event list. The simulation clock tracks the passage of simulated time. The clock advances in discrete steps during simulation run. The event list is responsible for determining which events to occur next (the events are sorted in ascending temporal order of simulation time.) and the appropriate state transitions to be executed. The occurrence of an event may trigger the occurrence of other events at later times. These future occurrences of events are implemented in a simulation model by placing the appropriate events on the event list. Event manager selects the next event to be fired from the event list and pushes forward the simulation clock to the time of that event. The selected event is then invoked by triggering the event object referenced by the event. The triggering is executed by calling "On Event" method of the event object. Upon triggering, the event object updates its state and can generate future tasks by adding new events into the event manager. The event manager repeatedly fires the next event until either a pre-assigned simulation time is reached or the event list becomes empty. The event object can also monitor a specific set of events through "Add Monitor" method in the event manager. The event manager will invoke the event object directly on firing the next event. This monitor feature is used by the tracking module to track events in the simulation model.

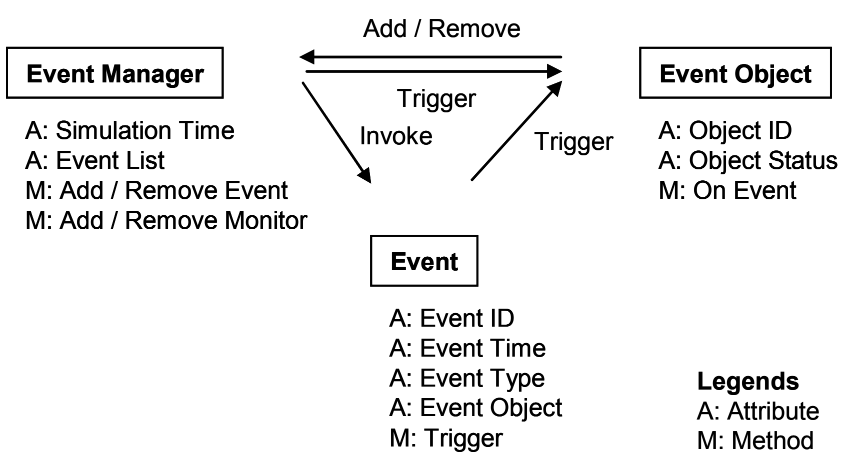


Figure 5.5: Core classes and interactions between them

The major classes and subclasses in the test bed and their inheritance relationship are provided in Figure 5.6. Event can be either job or machine type. The former deals with job related events such as job arrivals, job starts processing, etc. The latter handles machine related events such as machine downs or ups, etc. Events that are related to neither job nor machine are handled by event class itself. An example is the periodic rescheduling event.

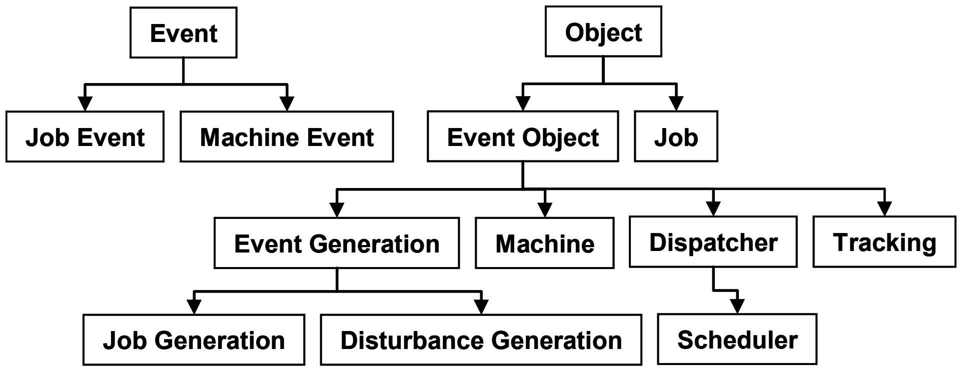


Figure 5.6: Major class inheritance

"Object" class has dynamic data feature, which allows data storage to be added dynamically to objects. This feature is inherited to all subclasses of Object class. Both "Job" and "Event Object" classes are subclasses of the Object class. Job class is used to represent jobs on the actual shop floor and contains process routing and operation related information. Event Object class and all its subclasses have function that can be triggered by the event manager on invocation of an event. This enables the classes to respond to events, and forms the underlying structure of the simulation test bed. To schedule future tasks as well as to interact with each others, the five modules in the test bed (event generation, dispatching, scheduling, manufacturing and tracking modules) make use of this functionality. For examples, the event module can schedule the next job arrival event, while the dispatching, scheduling and tracking modules can opt to monitor a set of specific events.

Under "Event Generation" class, two subclasses are implemented: "Job Generation" and "Disturbance Generation" classes. Job Generation class creates new job arrivals whereas Disturbance Generation class produces machine breakdown events. "Scheduler" class is defined under "Dispatcher" class because both classes respond to events, and Scheduler class requires a more specialized implementation than Dispatcher class. Dispatcher class responds to machine ready event so as to select the next job to load onto the specific machine whereas Scheduler class responds to job arrival, machine breakdown or up, and periodic rescheduling events.

The test bed is implemented using C++ programming language. We choose C++ as the implementation language for two reasons. First, general programming languages have very good compiler support, and hence their execution speed is generally faster. Second, language level reusability is an important factor to be considered, and C++ is one of the languages that support code reuse. With Standard Template Library (STL), C++ programs can achieve high efficiency while maintaining a high level of code reuse.

5.5 Performance Measures

We have implemented to report on mean cycle time (or flow time), cycle time variance, machine utilization, and throughput. However for our study, the main focus is on mean cycle time and its variance. Mean cycle time is defined as average amount of time jobs spend in the system. Mean cycle time and its variance are considered as key performance criteria in discrete manufacturing system such as semiconductor manufacturing [Sivakumar, 1999]. A shorter cycle time improves response to customer requirements, and results in a reduced work-in-process (WIP). The smaller WIP and inventory buffer can reduce the risk of product obsolescence due to technological changes. Further, in certain types of discrete manufacturing such as wafer fabrication, reducing work-in-progress shortens the period that wafers are exposed to contaminants. Shorter exposure period can lead to improved yield.

Reducing the variance of the cycle time is also important as it allows a more accurate prediction of production completion time, which can then facilitate improved downstream coordination of further operations [Lu, 1994]. Reduced variance can also result in better quality of service, in terms of promised delivery dates, for customers. The mean cycle time can be defined mathematically as:

$$\text{Mean cycle time, } \overline{F} = \frac{1}{n}\sum_{i=1}^{n}\left(C_i - T_i\right)$$

Where,

n = total number of jobs completed

C_i = completion time of job i

T_i = time of arrival of job i

Cycle time variance is defined mathematically as:

$$\text{Cycle time variance, } \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}\left(C_i - T_i - \overline{F}\right)^2$$

5.6 Dispatching Rules and Scheduling Algorithms

The decision as to which job to load onto a machine when it becomes available can be made by using scheduling heuristics [Conway, 1962]. Scheduling heuristics can range from simple to complex. Simple heuristics use the information on the jobs to be processed at a particular machine, and more complex heuristics utilize information on jobs waiting at other machines or on shop characteristics. Dispatching rules such as shortest processing time (SPT) and shortest remaining processing time (SRPT) are local heuristics while rules such as process time plus work-in-next-queue (PT+WINQ) is a more complex heuristic since it makes use of information on job waiting at other machines.

A set of dispatching rules and scheduling algorithms has been developed in the test bed. These rules and algorithms are implemented as subclasses of dispatcher and scheduler classes respectively. Further details of the heuristics are presented in the following subsections.

5.6.1 Dispatching Rules

Before we present the dispatching rules, we introduce the terminology used in this section:

d_{ij} = process time for operation j of job i. Operation j of job i is performed on the machine that becomes available at the current time and this machine requires a job to be off-loaded from the queue

$o(i)$ = total number of operations on job i

W_i = total work content of jobs in the queue of the next operation of job i. If operation j is the last operation for job i, then W_i is zero.

Z_i = priority index assigned to job i at the time of decision of dispatching. The job with the least Z_i is chosen for loading onto machine.

The following dispatching rules (refer to Figure 5.7) have been implemented:

- **FIFO (first-in-first-out):** This rule chooses the job that has entered the queue at the earliest for loading. FIFO is an effective rule for minimizing the maximum cycle time [Rajendran, 1999]. This rule is used as a benchmark in the study.

- **SPT (shortest process time):** This rule selects the job with operation on the machine having the shortest process time. This rule is the most commonly used for job shop scheduling and is found to be very effective in minimizing mean cycle time [Conway, 1962; Blackstone, 1982]

  $$Z_i = d_{ij}$$

- **SRPT (shortest remaining process time):** This rule chooses the job with the least remaining process time, which excludes the current operation under consideration. It is considered in the study since it is found that the rule is effective in minimizing mean cycle time. This rule is optimal with respect to minimizing steady-state mean cycle time (or flow time) for the queuing model *M/G/1*. The rule is proven optimal analytically using priority queuing model in which the processing times of jobs are known upon arrival and preemption without loss of time or processing [Schrage, 1966; 1968; Smith, 1978]. Preemption will occur whenever the processing time of a newly arriving job is less than the remaining processing time of the job in service.

  $$Z_i = \sum_{q=j}^{o(i)} d_{iq}$$

- **PT+WINQ (process time plus work-in-next-queue):** PT is the operation process time of the job on the machine, and WINQ is total work content of jobs in the queue of next operation of the job. This is a look-ahead rule which is found to be effective in minimizing mean cycle time [Holthaus, 1997].

  $$Z_i = d_{ij} + W_i$$

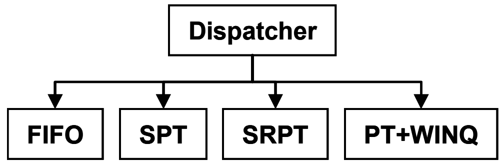


Figure 5.7: Subclasses of dispatcher class

5.6.2 Scheduling Algorithms

In our research, we apply simulation technique to dispatching rules to create a new kind of scheduling algorithm for production scheduling. We term this algorithm as a recursive simulation based algorithm.

5.6.2.1 Recursive Simulation

The concept of recursive simulation is similar to the popular operating construct in computer science known as "recursion", which is a common technique to split a complex problem into its single simplest case. This concept can similarly be applied in simulation to allow for nested simulation runs within a simulation run, which is termed as "recursive simulation" [Gilmer, 2000]. In this simulation technique, simulated decision makers themselves use simulation to make decision. The recursive simulation runs within a simulation are used to evaluate the possible outcomes given that a decision was made on one way or the other. The number of levels of recursion, the types (deterministic or stochastic) and numbers of runs can be varied.

Very little is reported in literature relating to recursive simulation. The only known publication to date is the use of recursive simulation in military decision-making [Gilmer, 2000; 2003; 2004]. The use of recursive simulation has been demonstrated to improve on the quality of decisions. However, the use of recursive simulation runs can incur high computation costs, especially in the cases of multiple levels of recursion and multiple replications. Nonetheless, with the technological trend of rapid multiplication of computation power and advances in distributed simulation cloning which allows multiple possibilities at a decision point to be tested concurrently [Chen, 2004], the execution speed of recursive simulation can eventually be improved.

In our research of recursive simulation for shop floor scheduling, we have limited the scope to a single level recursion and deterministic simulation run. Multiple levels of recursive simulation runs tend to improve quality of schedules, but the amount of improvement is usually not justified for the excessive computation costs incurred.

5.6.2.2 The Concept of Recursive Simulation based Algorithm

Figure 5.8 shows the concept of recursive simulation. The base simulation run is the run invoked by a simulation practitioner. Suppose at a simulation time t_1, a simulated entity needs to make a decision. The entity may use simulation (i.e. first level recursive simulation run) to evaluate various alternatives, and subsequently decides the best alternative to take based on the simulated results. Within the simulation run pertaining to each alternative, the simulated entity can again invoke simulation (second level recursive simulation run) to study different scenarios at decision points.

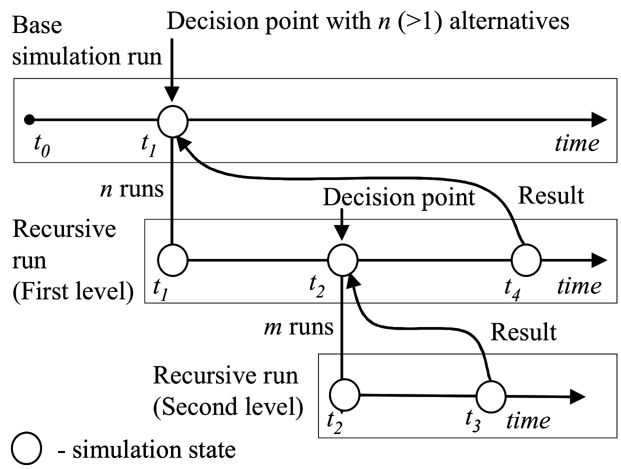


Figure 5.8: Recursive simulation

In recursive simulation based scheduling, a decision point in simulated time can be the time when a machine is ready to process next job from the queue. In this case, the machine is the entity and the jobs in the queue are the alternatives. Recursive simulation run is used to evaluate each choice (or job), and the choice that gives the best measure of performance, mean cycle time for instance, will be selected at the decision point. For deterministic simulation, the number of recursive simulation runs needed at a decision point is equal to the number of jobs (>1) in the queue. If only a single job is in the queue, the job will be picked for processing without requiring any recursive simulation run. The run length of recursive simulation can be either fixed or based on certain terminating criteria such as a fixed number of jobs complete all its operations.

To better illustrate the proposed scheduling approach which is currently limited to a single level of recursive simulation run, consider an example of heuristic being used to dispatch jobs to machines in the base simulation run and the performance criteria is mean cycle time. Suppose at a decision point in the base simulation run where a machine is ready to load the next job, and there are n jobs in the queue. To determine the best job to load onto the machine, a recursive simulation run (i.e. first level recursive run) is invoked to evaluate the performance for each job in the queue (jobs 1 to n) assuming the simulated scenario that the specific job was already loaded onto the machine in the base simulation. Dispatching rules such as FIFO, SPT, etc. can be used in the recursive simulation to dispatch jobs to machines. The mean cycle time of all jobs in the shop floor problems are determined at the end of the recursive simulation run and the result retained for comparison to other recursive runs (i.e. $n - 1$ runs) of other jobs in the queue assuming the job loaded onto the machine in the base simulation run. The job that gives the best mean cycle time among the recursive runs will then be loaded onto the machine at the decision point in the base simulation. This process is repeated for each decision point in the base simulation.

The execution speed of the scheduling approach is apparently dependent on the number of recursive runs and the execution speed of each recursive simulation. The number of recursive runs is proportional to the product of the number of decision points in the base simulation and the average number of jobs in the queue at each decision point. Both the number of decision points and the total number of operations is proportional to the number of jobs in the job shop. Therefore the execution speed is directly proportional to the square of the total number of jobs in the system. The execution speed of each recursive simulation run is affected directly by the underlying dispatching rule and the run length of the simulation. It is noted that if a static job shop with a finite number of jobs in the system is considered, as simulation time progresses in the base simulation, more jobs will complete its operations and therefore fewer jobs are available for selection in the queue. This will result in fewer recursive simulation runs required as simulation time advances.

5.6.2.3 Recursive Simulation based Scheduling Algorithms and Tabu Search

For our research, we have implemented 8 scheduling algorithms (see Figure 5.9) as subclasses of Scheduler class. 6 of these algorithms are recursive simulation based algorithms using dispatching rules in the first level recursive simulation run (refer to Section 5.6.2). These algorithms are rFIFO, rSPT, rSPRT, rPT+WINQ, rDR1 and rDR2. The other two scheduling algorithms, TB1 and TB2, are Tabu search algorithms. The details of these algorithms are described as follows:

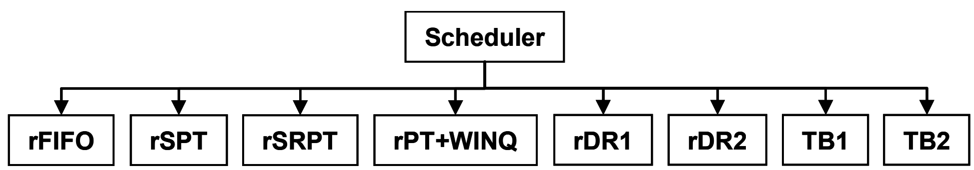


Figure 5.9: Subclasses of scheduler class

- **rFIFO, rSPT, rSRPT and rPT+WINQ:** These heuristics are the recursive simulation based implementation of the corresponding dispatching rules. In these heuristics, each job in the queue of a ready machine in the base simulation is enumerated through recursive simulation to determine the best job to load onto the machine. The specified dispatching rules are used in the recursive simulation runs (first level) to dispatch jobs onto machines.

- **rDR1:** This heuristic evaluates the performance of each job in the queue of a specific machine in the base simulation by using a combination of dispatching rules FIFO, SPT, SRPT, PT+WINQ, LPRPT (Least percent remaining process time which is equal to the remaining process time over total process time) and (PT+WINQ)/TIS (where TIS is the time in system of the job) in the recursive runs (number of recursive runs = number of jobs in the queue x number of dispatching rules) and selects the job with the recursive run that gives the best performance.

- **rDR2:** This heuristic combines rFIFO, rSPT, rSRPT, rPT+WINQ and rDR1, and always returns the best schedule among all the recursive simulation based heuristics, but incurs additional computational burden.

- **TB1:** This is a modified implementation of Tabu search approach described by Nowicki and Smutnicki [1996]. Their work is considered as one of the most efficient Tabu search implementations for the performance criteria of makespan [Blazewicz, 1996]. Tabu search is a meta-heuristic based on a local search technique which attempts to exploit the solution space beyond local optimality through forbidden (or tabu) move. Instead of considering makespan, the search approach has been modified to consider mean cycle time for shop floor problems. To cater for the mean cycle time criterion, the critical path for each job in the schedule is calculated [Armentano, 2000]. This path determines the completion time of the job, and consequently the cycle time of the job. In comparison with the original work which only determines the critical path of the schedule, the size of neighborhood in the modified work has increased substantially. The size of neighborhood plays an important role in the efficiency of the Tabu search procedure. TB1 procedure uses SPT to obtain an initial solution and searches for new solutions in the neighborhood based on the critical paths of the jobs. The terminating criterion is set to be a specific number of iterations without any improvement such that the run time is in similar range to rDR2 heuristic.

- **TB2:** This procedure is the same as TB1 except for the stopping criterion, which is set to be 5 times the number of iterations of TB1. Hence TB2 should achieve equal or better performance than TB1 but with much higher computation costs.

5.7 Summary

In this chapter, we present a framework that allows the performance assessment of dispatching rules or scheduling algorithms for discrete manufacturing systems. Based on the framework, we develop a simulation based test bed using object-oriented programming techniques. Discrete event simulation is used to emulate shop floors in the test bed. The test bed enables experimentation to be performed on different shop floor conditions, dispatching rules and scheduling algorithms.

We have also proposed and implemented a recursive simulation based scheduling algorithm to improve on the mean cycle time performance of dispatching rules. Dispatching rules are often known to be highly problem dependent. By applying recursive simulation technique, we are able to increase the search space and subsequently generate better solutions.

The work on recursive simulation technique indicates that the approach is intuitive and easily customizable for different dispatching rules and performance criteria. Multiple levels of recursive simulation runs can also be applied to enhance performance further if computation costs are not an issue. The recursive simulation based algorithm will be used in the research on dynamic and stochastic job shop scheduling problems. This is primarily because of the lower computation costs, in addition to the comparable performance to other efficient scheduling approaches in the literature.

The merits of recursive simulation based scheduling approach are that it is a constructive scheduling approach, and it requires no fine tuning of parameters as compared to popular iterative local search methods such as tabu search, ant colony and simulated annealing algorithms. The performance of the local search approaches for job shop scheduling can be affected significantly by the choice of neighborhood structures. Further, these local search approaches usually come with a set of parameters that need to be experimented and fine tuned for better performance. These parameters include length of tabu list and size of elite solutions and so on for tabu search [Armentano, 2000], pheromone trail intensity, impact of trail and attractiveness, etc. for ant colony [Blum, 2004], initial temperature, number of iterations at each temperature, how and when temperature decreases, etc. for simulated annealing [Jeffcoat, 1993], population size, crossover and mutation rates for genetic algorithm [Bierwirth, 1999].

It should be noted that the number of recursive calls will grow exponentially with increasing number of jobs. However, there are ways to improve the computation time of recursive simulation for job shop scheduling. Initial heuristic search algorithm such as beam search, can be used to estimate the promise of each job in a machine queue. Only the first m most promising jobs will be considered for recursive simulations. Shorter recursive simulation run length can also be applied to improve the computation time. In this case, the run length of recursive simulation can be fixed to a shorter duration or a fewer numbers of operations. Another approach is to perform initial computationally intensive off-line study on recursive simulation to identify the most effective set of dispatching rules, and subsequently use the best set of rules in the actual on-line scheduling problems.

CHAPTER 6

EXPERIMENTS, RESULTS AND ANALYSIS

6.1 Introduction

This chapter primarily presents a comparative study on the performance of scheduling algorithms and dispatching rules in the dynamic and stochastic shop floor environments of discrete manufacturing systems. A total of 4 scheduling algorithms and 5 dispatching rules are considered in the study with respect to the objectives of mean cycle time and cycle time variance. To facilitate the study, a simulation model of a hypothetical job shop with 10 machines is used in the test bed. The shop model is with random routing of jobs and machine breakdowns under different set of conditions, which include utilization level, processing time variation, frequency and duration of breakdown. The study is conducted in two stages. In the first stage, the relative performance of a scheduling heuristic under both event-driven and periodic rescheduling strategies is evaluated. In the second stage, the scheduling algorithms under event-driven strategy are compared to the dispatching rules.

In the past research, queuing theory techniques have also been used in the study of shop floors. However, the analytical methods can become very complex to solve even for small shops. Therefore, simplifying assumptions are common in the analytical models. For example, a landmark analytical study by Jackson [1957a; 1957b] based on decomposition principle must meet some simplifying assumptions for the analytical model to be valid. One of the assumptions is that all jobs must be served in first-in-first-out (FIFO) manner. In real-life situations, jobs pass through the network of queue in a large variety of patterns. Hence, in attempting to study job dispatching methods other than FIFO, one can not escape the problem of having interrelated queues. For this reason, virtually all studies of job shop dispatching rules have employed simulation rather than analytical techniques [Muhlemann, 1982].

This chapter is organized as follows: Section 6.2 discusses scheduling algorithms and dispatching rules for the shop floors. Section 6.3 describes the shop floor scheduling problems in detail. Section 6.4 discusses the experimental design. This is then followed by experimental results in Section 6.5. The discussion and analysis of the results are presented in Section 6.6. Finally, Section 6.7 summarizes the chapter.

6.2 Scheduling Algorithms and Dispatching Rules

Sets of scheduling heuristics are developed in our simulation study. These scheduling heuristics, which consist of scheduling algorithms and dispatching rules, are discussed in Section 5.6 (Chapter 5). To evaluate the effectiveness of the developed scheduling heuristics, we have conducted an evaluation using a set of benchmarking static job shop problems. The purpose of this evaluation is to identify the most appropriate scheduling heuristics to be used in the experimentation of dynamic and stochastic job shop problems. Experimental studies using discrete-event simulation often demand high computation costs, and therefore it is orderly to perform a preliminary filtering on the scheduling heuristics before the actual simulation studies.

6.2.1 Evaluation of Scheduling Heuristics Using Static Job Shop Problems

The relative performance of the implemented scheduling heuristics is studied by evaluating them on a set of static job shop problems. These job shop problems are standard problems with a finite set of jobs and machines. Each job is with a number of operations that require the machines for processing. No machine breakdown is involved. The job shop problems are obtained from past literature, and are as follows (refer to Appendix D for the details of the job shop problems) [Ganesan, 2004]:

- 3 problems from Fisher & Thompson [1963], referred as ftp06, ftp10 and ftp20
- 40 problems from Lawrence [1984], referred as la01 – la40
- 20 problems due to Storer et al. [1992], referred as swv01 – swv20
- 10 problems used by Applegate and Cook [1991], referred as orb1 – orb10
- 4 problems used by Yamada and Nakano [1992], referred as yn1 – yn4
- 5 problems formulated by Adams et al. [1988], referred as abz5 – abz9

The sizes of these problems range from 6 to 50 jobs and 5 to 20 machines. Larger sizes of shop problems are not considered in the study as past results have concluded that the size of the shop does not affect the relative performance of the dispatching rules, and valid conclusions could be drawn from experiment with relatively small shops [Nanot, 1963; Buffa, 1968].

Table 6.1 summarizes the results on the relative performance of mean cycle time in terms of percentage for the dispatching rules and recursive simulation based heuristics as compared to FIFO. The results show the average, minimum and maximum improvement for the mean cycle time of jobs as well as the number of solutions that is better than the 82 solutions generated by FIFO.

Table 6.1: Relative performance of mean cycle time for dispatching rules and scheduling heuristics as compared to FIFO

Relative improvement to FIFO	Mean improvement (%)	Minimum improvement (%)	Maximum improvement (%)	Number of solutions better than FIFO (%)
SPT	16.10	2.30	38.85	100.00
SRPT	17.74	-2.28	46.07	98.78
PT+WINQ	14.69	1.09	39.04	100.00
rFIFO	13.62	3.57	39.63	100.00
rSPT	20.55	6.11	47.51	100.00
rSRPT	22.41	8.85	49.82	100.00
rPT+WINQ	19.91	6.63	44.21	100.00
rDR1	22.84	6.31	50.74	100.00
rDR2	23.20	6.69	50.74	100.00

Dispatching rules SPT, SRPT and PT+WINQ outperform FIFO from 15 to 18%. This result agrees with the past studies [Holthaus, 1997; Rajendran, 1999]. SRPT is slightly better than SPT and PT+WINQ. However, for one of the job shop problem, SRPT underperforms FIFO rule. The results also indicate that recursive simulation based heuristics outperform FIFO rule from 14% to 23%, and the recursive simulation based heuristics record better results than FIFO for all the shop problems. As expected, rDR1 and rDR2 give the best results since the heuristics use multi-pass scheduling techniques in addition to the recursive heuristics. However the performance gain of the multi-pass scheduling procedures is not significant as compared to the best result of the simpler recursive simulation heuristics, and the amount of improvement may not warrant the additional computation costs incurred.

Table 6.2 shows the performance of recursive heuristics relative to the corresponding dispatching rules. The results for the lower end improvement are encouraging as by simply applying recursive heuristic to simple dispatching rules such as FIFO, which is an average performer for mean cycle time, an improvement of 14% can be achieved. Even with the top performers of mean cycle time (SPT, SRPT and PT+WINQ), an improvement of 5 - 6% can still be attained relative to the corresponding dispatching rules.

Table 6.2: Relative performance of mean cycle time for recursive heuristics compared to the corresponding dispatching rules

Relative improvement to the corresponding dispatching rule	Mean improvement (%)	Minimum improvement (%)	Maximum improvement (%)	Number of solutons better than the corresponding dispatching rule (%)
rFIFO	13.62	3.57	39.63	100.00
rSPT	5.35	0.80	14.03	100.00
rSRPT	5.62	0.73	16.30	100.00
rPT+WINQ	6.24	0.50	12.38	100.00

Table 6.3 compares the top performers of recursive heuristics with Tabu search procedures TB1 and TB2. The results show that the recursive heuristics are

comparable to Tabu search procedures. The improvement of TB2 over TB1 is relatively small, considering the much higher computation costs.

Table 6.3: Relative performance of mean cycle time for top recursive heuristics and Tabu search as compared to FIFO

Relative improvement to FIFO	Mean improvement (%)	Minimum improvement (%)	Maximum improvement (%)	Number of solutions better than FIFO (%)
rSRPT	22.41	8.85	49.82	100.00
rDR1	22.84	6.31	50.74	100.00
rDR2	23.20	6.69	50.74	100.00
TB1	20.99	5.60	46.93	100.00
TB2	22.24	5.79	47.85	100.00

Figure 6.1 presents the total computation costs incurred for the 82 job shop problems for the dispatching rules and the recursive simulation based heuristics. The execution speed of recursive simulation based heuristics rFIFO, rSPT, rSRPT and rPT+WINQ is about 15 times the corresponding dispatching rules. Further, rDR1 and rDR2 record much higher computation costs, about 80 and 190 times respectively compared to the dispatching rules.

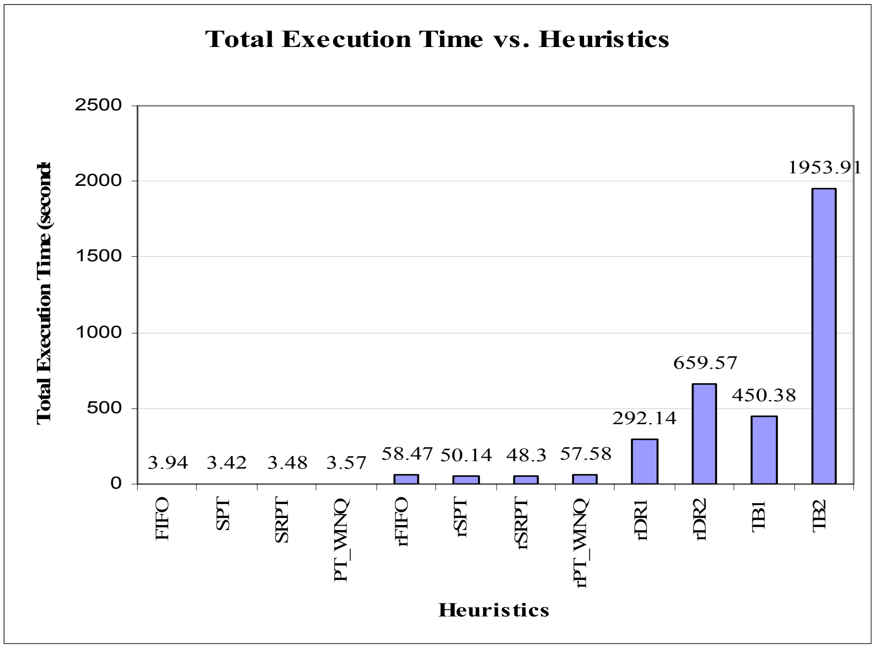


Figure 6.1: Total execution times of various heuristics and procedures for the 82 job shop problems

6.2.2 Discussions on Experimental Results

Depending on the performance requirement and computing constraints, we can determine the most appropriate dispatching rules or scheduling algorithms based on the results (see Figure 6.2). For example, if we are interested in very fast approach to job shop scheduling, dispatching rule SPT is a good choice with worst performance guarantee. However, if we can compromise execution speed for the mean cycle time performance, rSRPT or even rDR1 can be better alternatives. Based on this result, we have selected the most suitable dispatching rules and scheduling algorithms for the experimental study of dynamic and stochastic job shop problems.

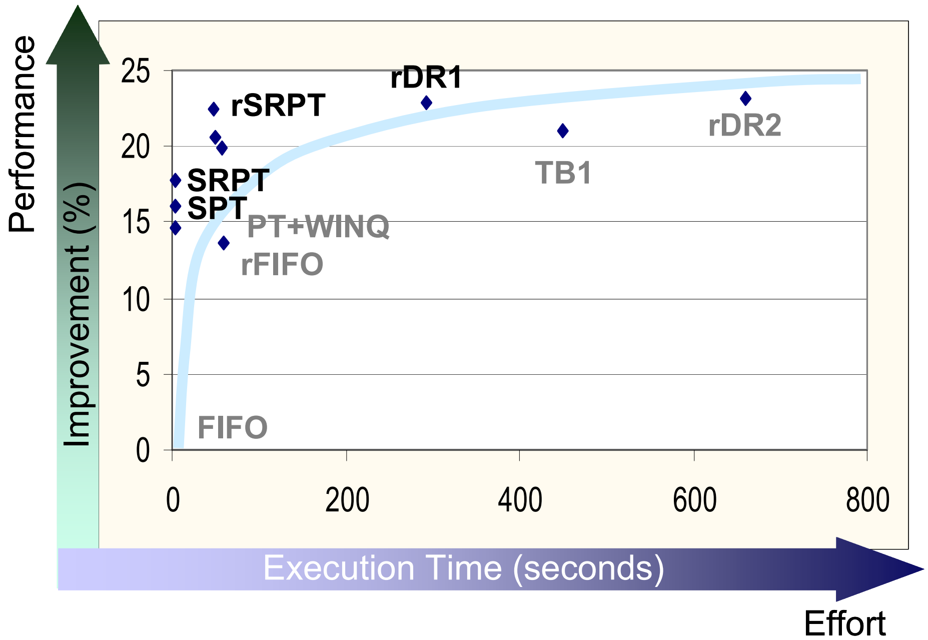


Figure 6.2: Percentage improvement versus execution time of the heuristics and procedures

The selected dispatching rules are: first-in-first-out (FIFO), shortest processing time (SPT), shortest remaining processing time (SRPT) and process time plus work-in-next-queue (PT+WINQ). FIFO is used for benchmarking purposes while other dispatching rules are found to perform well in terms of cycle time measures. The scheduling algorithms in the set are recursive simulation-based heuristics established on the corresponding dispatching rules. These include rFIFO, rSPT, rSRPT and rPT+WINQ. Another heuristic, rDR1, which extends the earlier recursive simulation based heuristics by evaluating a set of dispatching rules, is also included in the study.

6.3 The Job Shop Model and Assumptions

To analyze the dynamic and stochastic characteristics of discrete manufacturing systems, a job shop model has been developed using the simulation test bed (Section

5.6, Chapter 5) to facilitate the comparative study. Jobs are scheduled to arrive at the shop according to a Poisson distribution. The inter-arrival times of jobs follow an exponential distribution. Each job is assigned with operations, the processing time for each operation and the corresponding machines for the operations. The number of operations of each job is set to be 10, which is equal to the number of machines in the shop. The routing for each job is generated randomly, with each machine having an equal probability of being chosen. Every job is made to go through each machine, with machines being assigned randomly. No machine will be revisited since each machine is visited by each job. The jobs are routed in the shop according to open shop manner, whereby the number of routings is not limited and each job could have a different routing sequence. The process times are drawn from uniform distributions. By generating jobs in this way, the shop is assumed to have equal work load condition, and hence preventing long-term bottleneck situations in the model. By adjusting the time interval between the releases of jobs, the utilization level of the shop can be controlled.

After being assigned with operations, a job is transported to the specific machine according to its routing sequence. The job joins the waiting line (or queue) of the machine. Jobs in the queue will be dispatched to the machine for processing once the machine becomes available. The job dispatching follows assigned priority index according to dispatching rules, or job sequences of pre-computed schedules. The pre-determined schedules for all machines can be created by either event-driven or periodic scheduling strategy. For event-driven scheduling strategy, new schedules are generated whenever a new job arrived, machine downs and ups. For periodic scheduling policy, new jobs arriving will only be incorporated into the next scheduling. The new jobs will be added to the jobs already on the shop floor and another set of schedules will be generated. This process repeats until the end of simulation. A job leaves the production system when all the operations are completed. Performance measures such as cycle time, lateness and tardiness are collected at the time of job departure. At the end of the simulation run, total throughput, number of tardy jobs, etc are computed. The production system is operated for a designated length of time sufficient to obtain samples of all performance measures.

To model the stochastic characteristics of the manufacturing systems, machines are made inactive according to a Poisson distribution. The Time-Between-Failure follows an exponential distribution, while the Time-To-Repair is pulled from a uniform distribution. Machines can break down while processing or idling. If a job is interrupted by a breakdown event, the job resumes its operation after the machine returns to normal from the down state.

The assumptions pertaining to the job shop model are as follows:

1) Each machine can perform only one operation at a time on a job,
2) A job once taken up for processing should be completed before another job is taken, i.e. job preemption is not allowed,
3) An operation cannot be taken up for processing until its previous operations are complete,
4) There are no limiting resources other than the machines,
5) There are no interruptions in the shop floor except for machine breakdowns,
6) There are no alternate routes,
7) The jobs are independent of each other, i.e. no assembly is involved,
8) No sequence dependent setups,
9) There are no parallel machines.

The job shop model does not consider setup times or transit times as separate variables. Nor does it represent such practices as order splitting, which are shop practices for dividing one job into two or more jobs in the course of manufacturing.

6.4 Experimental Design

The dynamic and stochastic job shop problems are generated with dynamic arrival of jobs and machine breakdowns by using discrete-event simulation technique. To obtain valid results for dynamic and stochastic job shop problems, it is required to ensure that the simulation model is in a statistically stable state. This is necessary since at the start of the simulation run the shop floor is completely idle and empty. When the first few jobs are released, their operations can start immediately. However, when the subsequent jobs are released, they may not start as promptly as the first few jobs, since the shop floor is still working on the earlier jobs. Eventually however, these delays stabilize, at least in a statistical sense. A statistical analysis of the shop parameters, such as the average of the jobs' total cycle times would show that after certain number of jobs has been released to the system, a statistical equilibrium is established. To make sure the experiment is run under steady-state conditions, pilot runs of the model are observed and verified for the conditions.

Each simulation experiment consists of 30 different runs (or replications). In each run, the shop floor is continuously loaded with jobs that are numbered on arrival. In order to ascertain when the system reaches a steady state, we continuously observe the shop parameters, mean cycle time of jobs and total work-in-progress in the system, until the shop reaches a steady state. A graphical technique was used to determine when the shop reaches a steady-state condition [Law, 1984]. It is found that the shop reaches a steady state after arrival of about 600 jobs, which is considered as warm up period. We have fixed the run length for each replication as 5000 completed jobs after reaching steady state.

Analysis of variance is performed to test the significance of the factors. As for the computation of statistics from a given replication, we collect data from jobs numbering from 601 to 5600. The shop will be loaded with jobs, until the completion of the 5000 numbered jobs. This helps in overcoming the problem of "censored" data [Conway, 1965]. The problem of censored data may arise if the collection of statistics is based on jobs that were completed for a specified length of simulation time. Since a number of jobs remain uncompleted at the end of each simulation run, each run really represents a sample from which certain elements have disappeared. All other things being equal, a job that is running late is less likely to be completed by the time the simulation terminates than one that is on time. Thus, the effect of censored data is to bias on the low side all estimates of the mean and variance of cycle time. This bias is likely to be unequal for different dispatching rules [Blackstone, 1982].

The parameters and values that were used in the experiments are shown in Table 6.4. There are two different utilization levels, two levels of process time variations, two different breakdown durations, and two different percentages (or frequencies) of machine breakdowns, making a total number of 16 simulation experiment sets for each scheduling strategy or scheduling heuristic.

Shop utilization is established at two levels of the jobs. Low utilization is approximately 70% and high shop utilization is about 90%. The two levels of shop utilization are compatible with the experiments in the past literature [Kutanoglu, 1999; Lin, 1997; Litchfield, 2000; Metan, 2005]. The levels of utilization range from

60% to 95%, with 60% to 75% commonly being considered as low utilization and 90% to 95% regarded as high utilization. The job arrival rate is adjusted to achieve the levels of utilization required on the bottleneck machine, which determines the system load level. Small variations are present in these utilization levels due to the stochastic nature of the job arrivals [Barrett, 1990]. Processing time variability is set at two levels, 10% and 90%, with values pulling from uniform distributions. The 10% variation of processing time ranges from 9 to 11 unit time, while the 90% variation has processing time range from 1 to 19 unit time. These variations are introduced to reflect the differences in work rates that occur in real production processes. Two breakdown durations are set to be 30 and 600 unit time to denote breakdowns with different levels of severity. Two levels of percentage of breakdowns are set to be 5% and 25% to represent the more and less reliable shop floor respectively.

Table 6.4: The experimental parameters and values

Parameter	**Values**	**Remarks**
Utilization levels	70%, 90%	Utilization levels are controlled by adjusting the inter-arrival times of jobs through exponential distributions. Utilization levels of 70% and 90% are selected to represent a shop with a low and high congestion respectively.
Processing time variations	±10%, ±90%	Process time variations are controlled by adjusting the minimum and maximum parameters of uniform distributions. The mean processing time is set at 10 unit time
Duration of breakdowns	30, 600	Two different breakdown durations are tested to represent low and high breakdown levels, which are 30 and 600 unit time respectively. The durations are pulled from uniform

Parameter	Values	Remarks
		distributions with ±10% variations
Percentage of breakdowns	5%, 25%	The percentage of breakdowns is defined by the percentage of machine capacity. This percentage is controlled by adjusting the inter-arrival times of breakdown events through exponential distributions. Breakdowns are considered as busy states of the machines since the machines are not available to process jobs. The percentage of breakdowns is adapted accordingly to accommodate the duration of breakdowns

The experiments on dynamic and stochastic job shop problems are in two stages, and are described in Section 6.4.1 and Section 6.4.2 respectively. The first stage is pertaining to the frequency of rescheduling and the second stage is on the study of scheduling algorithms and dispatching rules. These experiments are designed to evaluate the performance of the scheduling heuristics under different job shop conditions. Common random number streams are used to send the same set of jobs through each configuration. Average of the performance criteria over 30 replications are compared and are used to evaluate the performance of the scheduling strategies and scheduling heuristics.

6.4.1 Design of Experiment (DOE) on Frequency of Rescheduling

Our first concern is on the frequency with which rescheduling occurs. The objective of these experiments is to study the effect of frequency of rescheduling on the mean cycle time (and cycle time variance) performance of shop floor under dynamic and stochastic manufacturing environments. The study is to confirm the results of the analytical models we derive in Chapter 4. 8 different rescheduling durations of practical significance are considered:

1) event-driven (considered as 0 unit time)
2) 120 unit time
3) 240 unit time
4) 500 unit time
5) 600 unit time
6) 700 unit time
7) 1440 unit time
8) 2880 unit time

The event-driven rescheduling policy triggers rescheduling on new job arrivals, machine downs and ups. We initially run experiments for a 2^5 factorial design with the 4 factors relating to the job shop conditions (see Table 6.4) and another factor relating to the duration of periodic rescheduling, which is set to a low of 0 unit time and a high of 2880 unit time. The experiments enable us to identify the major factors that affect the performance. We then proceed to run the various frequencies of rescheduling on the 16 combinations of job shop conditions as mentioned earlier. In order not to be overloaded with the number of experiments to run and not to repeat the past study by other researchers [Muhlemann, 1982], we limit the experiments to a single scheduling algorithm based on the recursive simulation based scheduling heuristic using FIFO as the underlying dispatching rule.

6.4.2 Design of Experiment (DOE) on Scheduling Heuristics

The objective of the experiments is to compare the mean cycle time and cycle time variance performance of a set of scheduling heuristics for shop floor under dynamic and stochastic manufacturing environments. With the experimental results, we can subsequently derive an effective event-driven scheduling approach for dynamic and stochastic manufacturing environments. In the experiments on scheduling heuristics, we first perform experiments on the dispatching rules, FIFO, SPT, SRPT and PT+WINQ on the 16 combinations of the job shop conditions to compare their performance. We then proceed with recursive simulation based scheduling heuristics, rFIFO, rSPT, rPT+WINQ and rDR1 but on the selected job shop factors as follows:

- Utilization level, 70% and 90%
- Process time variation, 10%
- Duration of breakdown, 30
- Percentage level of breakdown, 5%

We carefully pick the job shop conditions so as not to incur excessive computation costs. This is because recursive simulation based scheduling algorithms require longer execution time compared to the dispatching rules.

6.5 Experimental Results

For dynamic and stochastic job shop problems, the results are obtained by taking the average of the mean values of the 30 replications. A significant difference (alpha = 0.05) between the performance of the scheduling heuristics is found for most allowance factors and performance measures. The results of this study are evaluated more in terms of the ranked order of the heuristics than statistical significance, while statistically significant findings are reported where appropriate. The experimental results will be presented in the following subsections according to the stages of experiments mentioned earlier.

6.5.1 Experimental Results on Frequency of Rescheduling

Analysis of variance (ANOVA) using alpha value of 0.05 shows that processing time variations has no significant effect on the performance criteria of mean cycle time and cycle time variance. Other parameters are found to have impact on the performance measures. Based on the experimental settings, the effect of the parameters on the performance criteria in the order of the most to the least significant is: breakdown durations, percentage levels of breakdowns, utilization levels and rescheduling periods.

The main effect plots and interaction plots among the 5 factors using Minitab clearly show that increasing the length of rescheduling period increases the mean cycle time of jobs in the job shop. The plots of mean cycle time versus rescheduling period for breakdown durations of 30 and 600 unit times are shown in Figure 6.3 and Figure 6.4 respectively. The cycle time variance versus duration of rescheduling plots are not shown since no consistent trend can be identified from the experimental results.

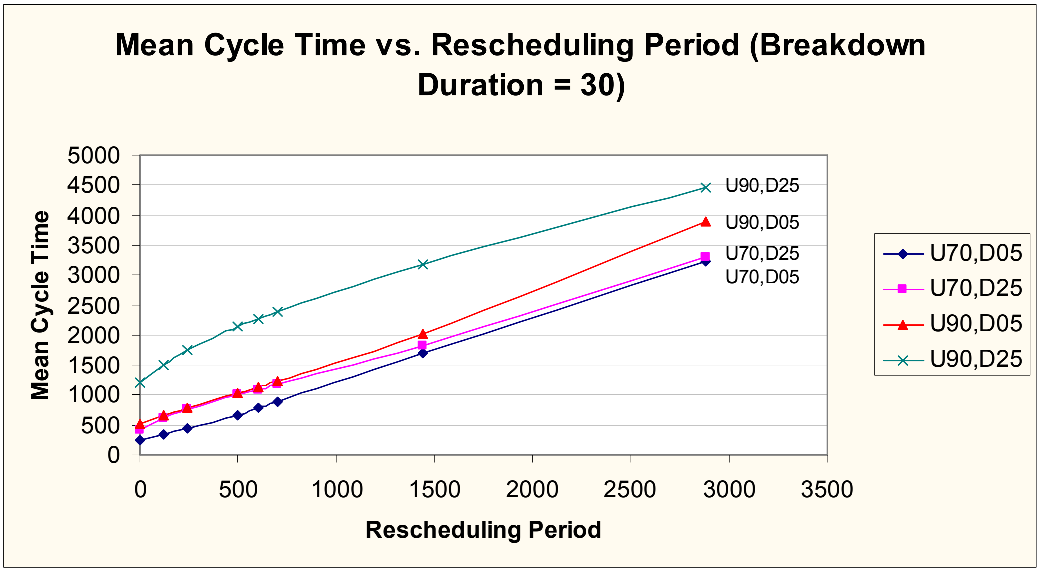


Figure 6.3: Mean cycle time versus rescheduling period for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

From the plots, it can be observed that mean cycle time varies with the length of rescheduling period in a linear manner under a specific combination of shop floor conditions. The effect of breakdown durations on mean cycle time can be noted by comparing Figure 6.3 and Figure 6.4. Long duration of machine breakdowns has a significant effect on the mean cycle time. For example, comparing the job shop conditions of high utilization level and high percentage level of breakdowns (U90, D25), the mean cycle time for job shop conditions of long breakdown duration ranges from 5.5 to 12.5 times to that of short breakdown duration.

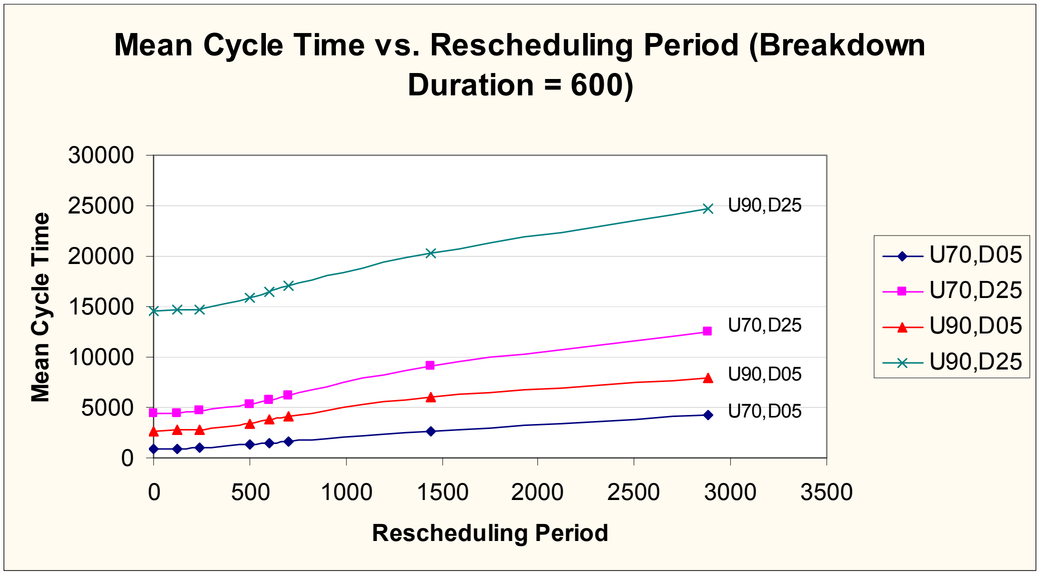


Figure 6.4: Mean cycle time versus rescheduling period for breakdown duration of 600 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

It could also be observed from Figure 6.4 that long breakdown duration has a more significant impact on the mean cycle time performance for rescheduling periods which are shorter than breakdown duration. This can be seen as initial horizontal lines on the plots for rescheduling durations of 0, 120 and 240 unit time.

6.5.2 Experimental Results on Scheduling Heuristics

Analysis of variance (ANOVA) using alpha of 0.05 shows that processing time variations has no significant effect on the performance criteria of mean cycle time and cycle time variance for the 4 factors relating to the job shop conditions and for any two of the dispatching rules. Based on the experimental settings, the effect of the parameters on the performance criteria in the order of the most to the least significant is: breakdown durations, percentage of breakdowns, utilization levels and dispatching rules.

The main effect plots and interaction plots of experiments based on 2^5 factorial design with any 2 of the 4 dispatching rules show that the relative performance of dispatching rules is similar under the 4 factors related to the job shop conditions. This means that a dispatching rule, which performs better than another dispatching rule under a specific set of job shop conditions, still outperforms the other rule on another set of job shop conditions.

The histograms of mean cycle time versus the factors of job shop conditions for the dispatching rules are shown in Figure 6.5 and Figure 6.6. Figure 6.5 presents histograms for breakdown duration of 30 unit time, whereas Figure 6.6 exhibits histograms for breakdown duration of 600 unit time. From the results, it is clear that the relative performance of the dispatching rules for the mean cycle time is in the order of PT+WINQ, SPT, FIFO and SRPT.

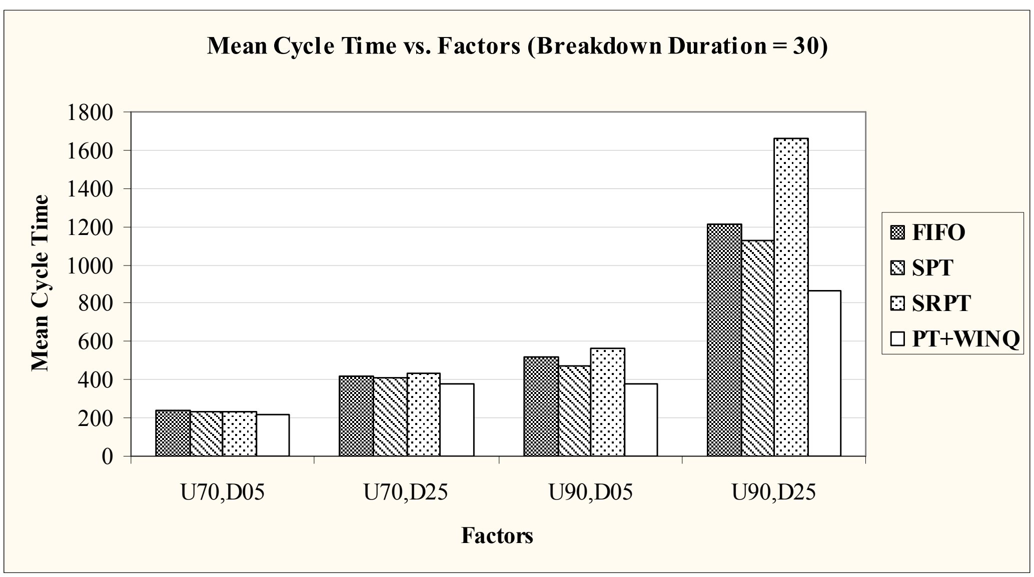


Figure 6.5: Mean cycle time versus factors for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

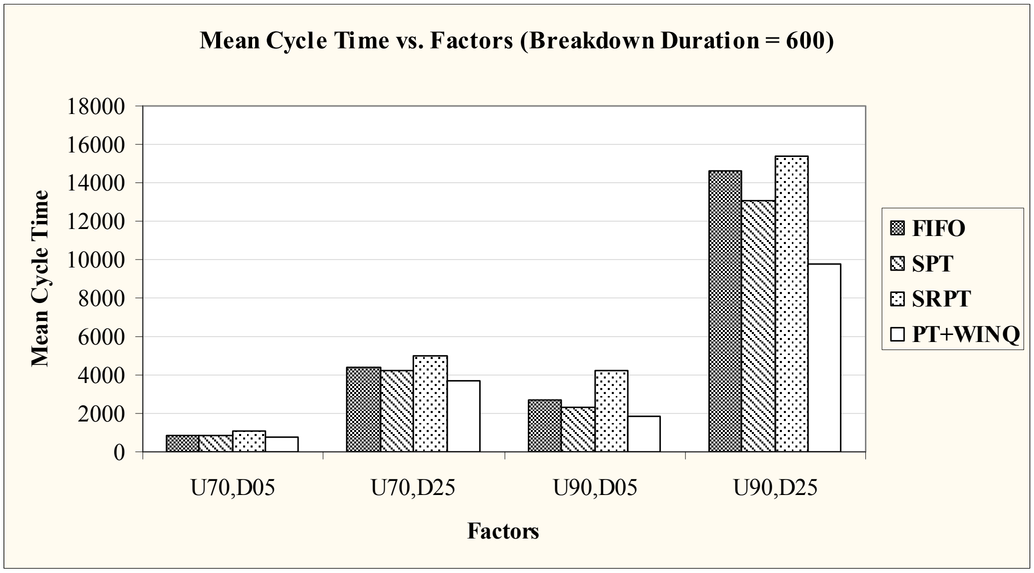


Figure 6.6: Mean cycle time versus factors for breakdown duration of 600 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

The histograms of cycle time variance versus factors of job shop conditions for the dispatching rules are shown in Figure 6.7 and Figure 6.8. Figure 6.7 presents histograms for breakdown duration of 30 unit time, whereas Figure 6.8 shows histogram for breakdown duration of 600 unit time. From the results, FIFO marginally outperforms other dispatching rules. PT+WINQ is slightly better than SPT while SRPT is the worst among the rules.

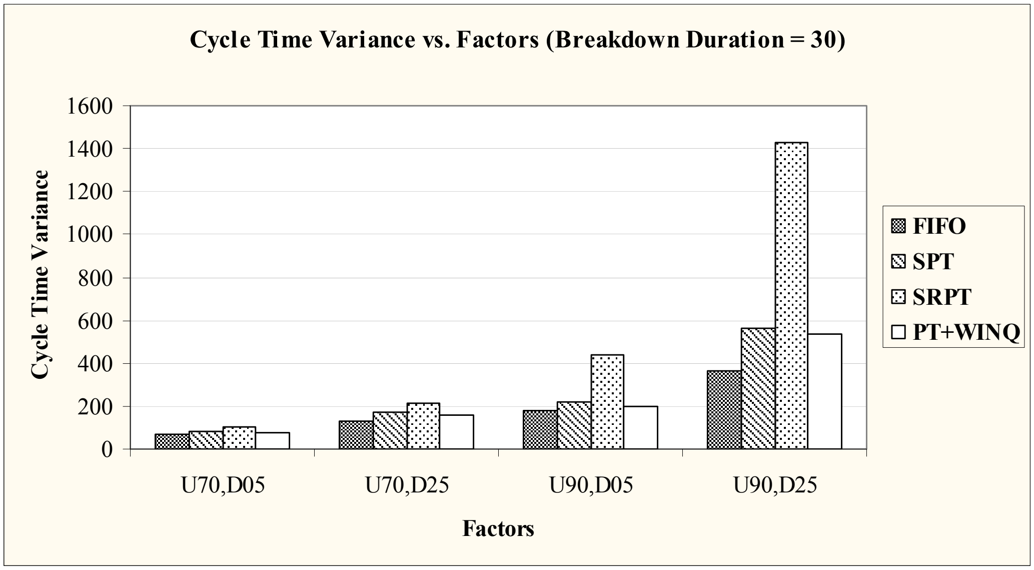


Figure 6.7: Cycle time variance versus factors for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

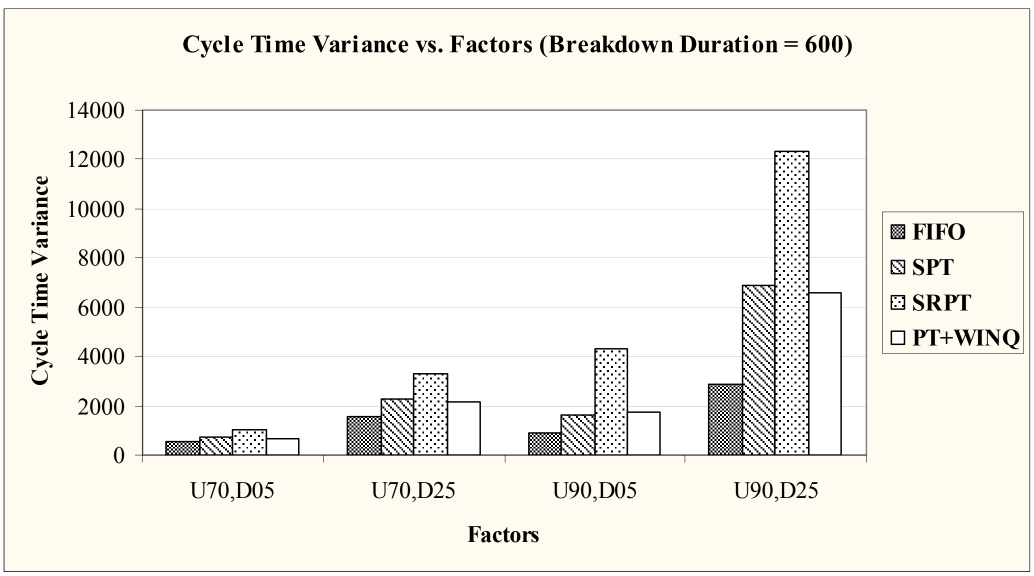


Figure 6.8: Cycle time variance versus factors for breakdown duration of 600 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Tables 6.5 and 6.6 show the mean cycle time and the cycle time variance for the dispatching rules and recursive simulation based scheduling algorithms. The corresponding histograms are shown in Figures 6.9 and 6.10. The main effect plots and interaction plots of the experiments based on the 2^5 factorial design with any two of the heuristics show that the relative performance of heuristics can vary under the 4 factors of job shop conditions. This implies that a scheduling heuristic, which performs better than another heuristic under a specific combination of job shop conditions, may under perform the other heuristic on another set of job shop conditions.

The results relating to the mean cycle time performance indicate that for utilization level of 70% and percentage level of breakdown of 5% (U70, D05), recursive simulation based heuristics rPT+WINQ and rDR1 outperform all other dispatching rules. However, for other job shop conditions, PT+WINQ performs better than all other scheduling heuristics.

Table 6.5: Mean cycle time of different scheduling heuristics for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
FIFO	236.80	419.54	515.05	1214.89
SPT	228.38	407.78	469.47	1128.32
PT+WINQ	215.22	**381.26**	**376.52**	**864.10**
rFIFO	211.15	392.14	421.07	1126.47
rSPT	209.12	**387.56**	408.42	**1038.77**
rPT+WINQ	**208.04**	388.61	**395.47**	1054.38
rDR1	**207.67**	387.66	430.55	1104.46

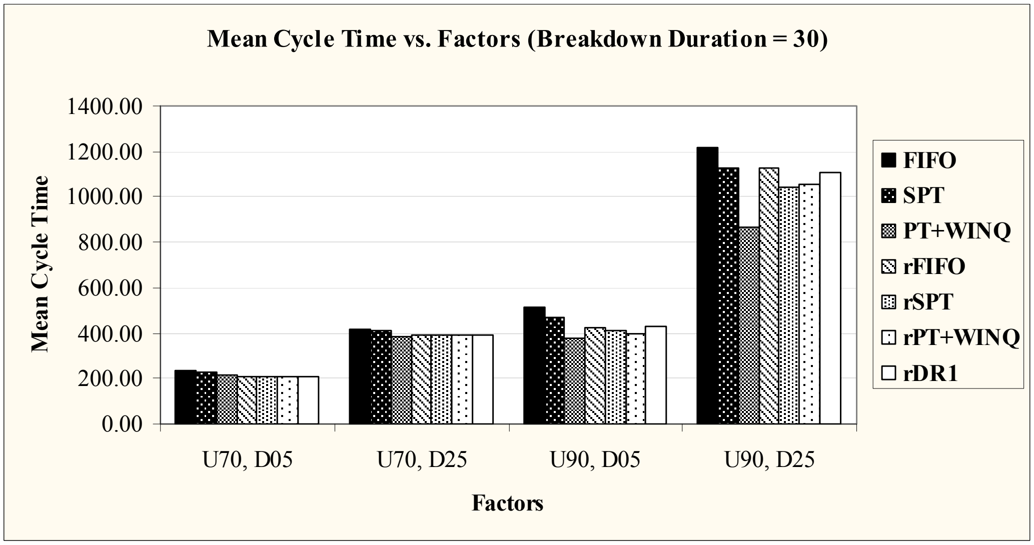


Figure 6.9: Mean cycle time versus factors for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

In the case of cycle time variance, FIFO outperforms all other heuristics. This is followed by PT+WINQ for most of the shop floor conditions. Recursive simulation based heuristics such as rFIFO, rPT+WINQ and rDR1 perform better than PT+WINQ under the conditions of low utilization level and low percentage level of breakdowns (U70, D05).

Table 6.6: Cycle time variance of different dispatching rules for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
FIFO	**71.48**	**131.74**	**180.94**	**364.40**
SPT	83.44	168.26	218.56	563.28
PT+WINQ	75.89	**156.45**	**200.05**	**537.37**
rFIFO	**73.50**	166.38	219.02	733.28
rSPT	78.38	167.54	248.91	689.16
rPT+WINQ	75.23	165.91	215.75	648.41
rDR1	75.54	164.99	247.73	673.18

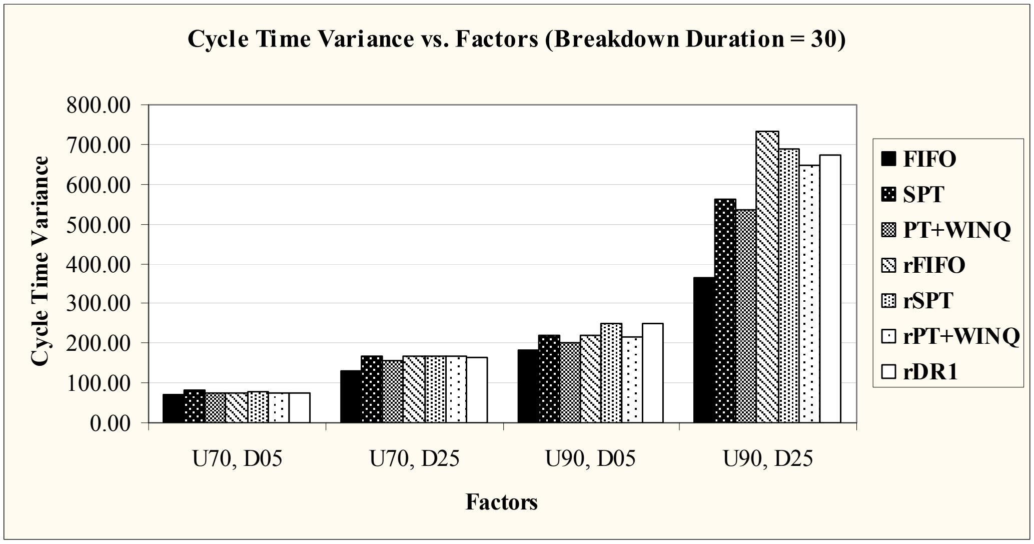


Figure 6.10: Cycle time variance versus factors for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

6.6 Discussions and Analysis

We discuss and analyze the experimental results according to the stages of the conducted experiments. We first discuss on the results relating to the experiments on frequency of rescheduling. This is followed by the discussion on the experimental results pertaining to scheduling heuristics.

6.6.2 Discussions on Frequency of Rescheduling

The experimental results on the frequency of rescheduling tally with our analytical models. Mean cycle time increases linearly with the length of rescheduling period under different set of shop floor conditions. The results also agree well with the past studies [Muhlemann, 1982]. In general, the performance of job shop deteriorates as the rescheduling period increases.

Our studies on the breakdown duration also reveal that long breakdown duration has a significant impact on the mean cycle time performance for shorter rescheduling periods. This effect is observed as initial horizontal lines in the plot of Figure 6.4. It is thus concluded that long breakdown duration can delay jobs in the shop floor by long duration regardless of rescheduling period. This implies that for shop floor with possible long breakdown duration, it is of little benefit to have rescheduling periods that are much shorter than breakdown duration as jobs will be delayed by the breakdowns.

We can approximate and analyze the effect of machine breakdowns on the job's waiting time by considering a single-machine queuing model as depicted in Figure 6.11 [Adon, 2006]. Assume there is only one type of entities representing jobs arriving to the system according to a Poisson stream with arrival rate λ and the processing times are exponentially distributed with rate μ. The server denoting machine is successively up and down, subject to time-dependent breakdowns (the machine breaks down at random instants, whether it is processing or not). The up and down times of the machine are exponentially distributed with means $1/\eta$ and $1/\theta$ respectively. As soon as the machine has been repaired, processing resumes at the point where it was interrupted. Let ρ_U and ρ_D denote the fraction of time the machine is up and down respectively. Hence we have:

$$\rho_U = \frac{1/\eta}{1/\eta + 1/\theta} = \frac{1}{1+\eta/\theta} \quad \textbf{(6.1)}$$

$$\rho_D = 1 - \rho_U = \frac{1}{1+\theta/\eta} \quad \textbf{(6.2)}$$

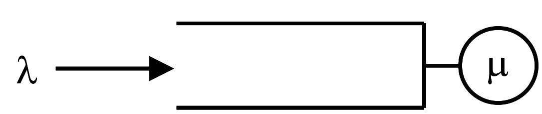


Figure 6.11: A single-machine queuing model

For steady state condition, we require:

$$\frac{\lambda}{\mu} < \rho_U$$

Consider an arriving job into the queuing system. The job will find an average of $E[L]$ jobs in the queue and each of them has an exponential processing time with mean $1/\mu$. Hence the time in system for the job is equal to $(E[L]+1)/\mu$ for a machine without breakdown. When breakdowns do occur according to a Poisson process with rate η, the mean number of breakdowns experienced by the job is $\eta(E[L]+1)/\mu$, with the mean duration of each breakdown as $1/\theta$. When a job arrives, the probability that the machine is already down is ρ_D while the probability that the machine is up is $(1-\rho_D)$. When the machine is down, the arriving job has an extra mean delay of $1/\theta$. Summarizing, we have the mean cycle time for jobs as:

$$\begin{aligned} E[S] &= (1-\rho_D)\left[(E[L]+1)\frac{1}{\mu} + \eta(E[L]+1)\frac{1}{\mu}\cdot\frac{1}{\theta}\right] + \\ &\quad \rho_D\left[(E[L]+1)\frac{1}{\mu} + \eta(E[L]+1)\frac{1}{\mu}\cdot\frac{1}{\theta} + \frac{1}{\theta}\right] \\ &= \left(1+\frac{\eta}{\theta}\right)(E[L]+1)\frac{1}{\mu} + \frac{\rho_D}{\theta} \\ &= (E[L]+1)\frac{1}{\mu\rho_U} + \frac{\rho_D}{\theta} \end{aligned} \tag{6.3}$$

But Little's law states:

$$E[L] = \lambda E[S] \tag{6.4}$$

Substituting (6.4) into (6.3), we obtain:

$$E[S] = \frac{1/(\mu\rho_U) + \rho_D/\theta}{1-\lambda/(\mu\rho_U)} \tag{6.5}$$

From expression (6.5), it is clear that mean cycle time for jobs is directly proportional to the duration of down times with mean $1/\theta$, if λ, μ, ρ_U (or ρ_D) are kept constant (compare Figures 6.3 and 6.4). Consider the case where we increase the utilization level of the machine by increasing λ while keeping other factors constant, the mean cycle time for jobs will increase accordingly (compare Figure 6.3 or 6.4, between (U90, D25) and (U70, D25), or (U90, D05) and (U70, D05)). Similarly, if we only increase ρ_D, the mean cycle time for jobs will also increase accordingly (compare Figure 6.3 or 6.4, between (U90, D25) and (U90, D05), or (U70, D25) and (U70, D05)). Using expression (6.5), we can thus evaluate the relative mean cycle time performance of dynamic and stochastic shop floor under different conditions.

Extending the analysis to group-arrival instead of single-arrival queuing model, we can evaluate the impact of machine breakdowns on periodic scheduling. Assume again there is only one type of jobs arriving to the system but in batches of size b according to a Poisson stream with arrival rate λ_b and the processing times are exponentially distributed with rate μ. The machine is successively up and down, the up and down times of which are exponentially distributed with means $1/\eta$ and $1/\theta$ respectively. Let ρ_U and ρ_D denote the fraction of time the machine is up and down respectively. First consider the case of no machine breakdown, we obtain:

$$E[S] = E[L]\frac{1}{\mu} + \frac{1}{2}(b-1)\frac{1}{\mu} + \frac{1}{\mu} \qquad \textbf{(6.6)}$$

The first term at the right-hand side of (6.6) corresponds to the mean waiting time for jobs already in the queue. The second term is the mean waiting time of servicing members in its own group. The last term is the service time of the job under consideration. Consider next the queuing model that incorporates machine breakdown, we obtain:

$$\begin{aligned} E[S] &= (E[L]+1)\frac{1}{\mu} + \frac{1}{2}(b-1)\frac{1}{\mu} + \eta\left(E[L]+1+\frac{1}{2}(b-1)\right)\frac{1}{\mu}\cdot\frac{1}{\theta} + \rho_D\frac{1}{\theta} \\ &= \left(1+\frac{\eta}{\theta}\right)\left(E[L]+\frac{1}{2}(b+1)\right)\frac{1}{\mu} + \frac{\rho_D}{\theta} \\ &= \left(E[L]+\frac{1}{2}(b+1)\right)\frac{1}{\mu\rho_U} + \frac{\rho_D}{\theta} \end{aligned} \quad \textbf{(6.7)}$$

Considering Little's law $E[L] = \lambda_b bE[S]$ and rearranging (6.7), we have:

$$E[S] = \frac{(b+1)/(2\mu\rho_U) + \rho_D/\theta}{1-\lambda_b b/(\mu\rho_U)} \quad \textbf{(6.8)}$$

However $\lambda_b = 1/t_P$, t_P being the rescheduling period and let $b = \lambda t_P$, we have:

$$E[S] = \frac{(\lambda t_P + 1)/(2\mu\rho_U) + \rho_D/\theta}{1-\lambda/(\mu\rho_U)} \quad \textbf{(6.9)}$$

From (6.9), it can be observed that when rescheduling period t_P is small or near zero, ρ_D/θ dominates. However when t_P is large, $(\lambda t_P + 1)/(2\mu\rho_U)$ can dominate the mean cycle time for jobs. The observation can be seen in Figure 6.4. When rescheduling period is small, the mean cycle time for jobs is dominated by breakdown duration. However, when rescheduling period is large, the mean cycle time increases proportionally with the rescheduling period. Further, when the rescheduling period t_P becomes very small (i.e. approaches 0 but not 0), (6.9) becomes (6.5) since $b(= \lambda t_P)$ approaches 1.

6.6.3 Discussions on Scheduling Heuristics

The experimental results show that dynamic and stochastic shop floor environments can significantly influence the performance of scheduling heuristics. Under static shop floor conditions, SRPT outperforms other dispatching rules for mean cycle time. However, under dynamic and stochastic shop floor conditions, SRPT performs poorly.

In dynamic and stochastic job shop environments, random events which can disrupt the shop floor constantly occur. These events, which are unanticipated and thus unpredictable, are difficult to be taken into consideration in scheduling. Further, any optimal or near-optimal schedules that are pre-computed may no longer optimal due to the disruptions. Clearly, the higher the utilization level and the percentage level of breakdown lead to more disruptions and hence higher uncertainty to the shop floors. This probably explains why recursive simulation based scheduling algorithms outperform dispatching rules only under low utilization level and low percentage level of breakdown (U70, D05), but not in other shop floor conditions. The experimental results thus raise an important issue of whether it is worthwhile to compute schedules using optimization algorithms, which are often more computationally intensive as compared to dispatching rules, in dynamic and stochastic shop floor environments with high utilization level (90%), high percentage level of breakdown or combination. This issue is also discussed in the past research [Pinedo, 1995; Wang, 1999].

To understand what conditions can contribute to a better scheduling heuristic for mean cycle time performance in dynamic and stochastic shop floor environments, we use simple queuing models for our analysis. Consider a single-machine non-preemptive queuing model with jobs (each with single operations) waiting as depicted in Figure 6.12. Jobs $J_i, J_{i+1},..., J_{i+n},...$ with processing time $B_i, B_{i+1},..., B_{i+n},...$ respectively, are waiting to be processed by the machine. To achieve minimum mean cycle time, the job with minimum processing time among the waiting jobs must be selected for processing when machine becomes available. The conjecture can be proved by comparing two different scenarios where jobs of two different processing times are selected for processing [Pinedo, 1995].

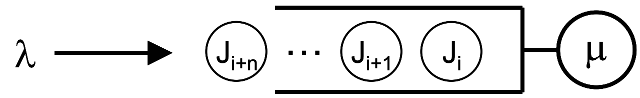


Figure 6.12: A single-machine queuing model with jobs waiting

Consider two jobs J_A and J_B among the waiting jobs with processing times B_A and B_B respectively, and let B_A<B_B. If job J_A is processed first, each job in the queue will incur an additional time of B_A in their cycle times:

$$T_i + B_A, T_{i+1} + B_A, ..., T_{i+n} + B_A, ...$$

Where T_i is the total accumulated cycle time of job J_i up to the time of the job selection process. Consider there are n jobs in the queue including job J_A, and hence the additional cycle time accumulated for all jobs in the queue is $n \cdot B_A$. If next job to be selected for processing is J_B, then the additional incurred cycle time is $(n-1) \cdot B_B$. Similarly, we can show that when job J_B is processed first followed by J_A, the additional accumulated cycle time for all the jobs are $n \cdot B_B$ and $(n-1) \cdot B_A$. Summing up the additional cycle times, we have:

Additional cycle time incurred by processing J_A first and then J_B =

$$n \cdot B_A + (n-1) \cdot B_B = n \cdot (B_A + B_B) - B_B$$

Additional cycle time incurred by processing J_B first and then J_A =

$$n \cdot B_B + (n-1) \cdot B_A = n \cdot (B_A + B_B) - B_A$$

Since B_A<B_B, hence $n \cdot (B_A + B_B) - B_B < n \cdot (B_A + B_B) - B_A$. Since mean cycle time is computed using the total cycle time accumulated for all jobs, hence processing job J_A first and then job J_B gives a better mean cycle time for the single-machine queuing model. Extending this reasoning to all the jobs in the queue, it can be shown that to achieve minimum mean cycle time, jobs need to be processed in ascending order of processing time.

In our reasoning, we can only show that always selecting job with the shortest processing time gives a better mean cycle time performance compared to selecting other jobs in the queue. However, it is not guaranteed that the shortest processing time (SPT) dispatching rule will lead to optimal performance in terms of mean cycle time for a dynamic shop floor. In this job shop, new jobs constantly arrive. A new job with even shorter processing time than the job selected for processing may arrive just after the selected job starts its processing on the machine. The best result can be guaranteed only when preemption is allowed whenever the processing time of a newly arriving job is less than the remaining processing time of the job being processed. In fact, it was proved in the past research [Scharge, 1966; 1968; Smith, 1978] that shortest remaining processing time (SRPT) minimizes mean cycle time, in a single-machine model with job preemption.

Analyzing network of queues is more complicated than single-machine queuing model due to complex interaction of jobs between queues. The interaction arises because each job consists of a number of steps, and each step has an operation that requires specific machine for processing. For such a network of queues, it would be a challenging task to determine the best strategy to dispatching jobs to machines. Consider a network of queues as depicted in Figure 6.13. In a typical queue in the network (i.e. rectangle with dotted lines, Figure 6.13), newly arriving jobs or jobs from other queues can join the queue. Since arrivals of new jobs are random in nature, it is hard to predict and analyze its effect on the performance of shop floor. Hence, we shall focus on queues without considering the effect of arrivals of new jobs into the queues.

Suppose a machine is ready to select the next job to process (refer to Figure 6.13), our goal is to choose the most preferred job so as to achieve lower mean cycle time based on known information. Selecting the job with the shortest processing time ensures that the remaining jobs in the queue incur the minimum total accumulated waiting time. However, the job may cause more waiting time in other queues. Hence, in selecting a specific job in a queue for processing, we should also consider its impact on other queues in the job's succeeding operation.

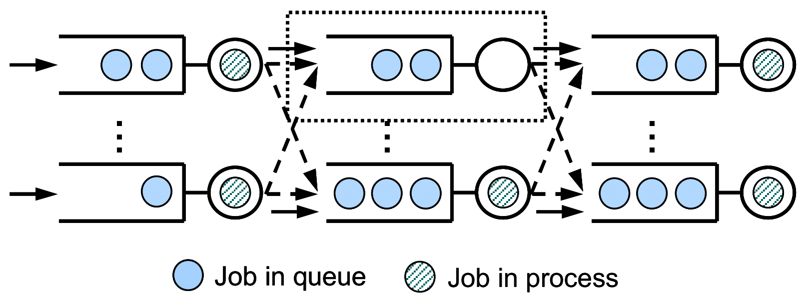


Figure 6.13: A network of queues

To better illustrate the idea, consider a simplified network of queues as shown in Figure 6.14. At time t, machine M_1 is ready to select the next job to process, and there are two jobs J_1^1 and J_2^1 in the queue. The succeeding operation of job J_1^1 requires machine M_2, whereas the succeeding operation of job J_2^1 needs machine M_3. There are jobs waiting in the queues of machines M_2 and M_3, and jobs are being processed on machines M_2 and M_3.

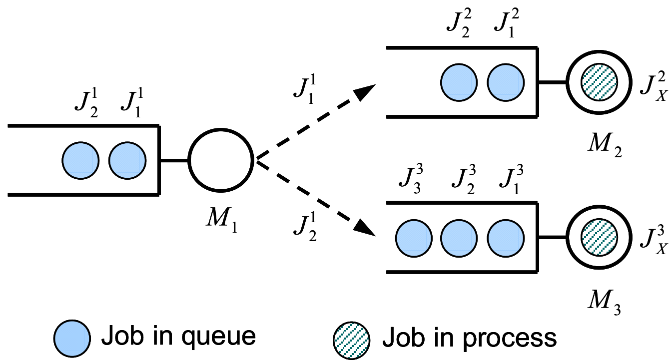


Figure 6.14: A simplified network of queues

Clearly, one of the important criteria in selecting which job to process next on machine M_1 is to evaluate the condition of the queue to which the job's succeeding operation will join. If the queue has a large number of jobs waiting to be processed, it may be better to select another job that will join another queue that has a smaller number of jobs waiting. Large number of jobs waiting in a queue usually means that the machine is a bottleneck machine, and the job joining the queue may be required to wait for a long duration before being processed. To be more precise, we can compute the total waiting time required in the queue and use the measure to determine the queue length. We can derive the following formula:

Total waiting time on machine k, $W_k = \sum_{j=1}^{n_k} B_{j,k} + R_k$

Where,

$B_{j,k}$ = processing time of job j on machine k

n_k = number of jobs in the queue of machine k

R_k = remaining process time of the job on machine k or remaining repair time of machine k

However, the job to be selected for processing on machine M_1 will only join its next operation's queue after being processed on machine M_1. Hence, we have a revised formula:

Net total waiting time on machine k, $W_k^* = \sum_{j=1}^{n_k} B_{j,k} + R_k - B_x$

Where,

B_x = processing time of the job to be selected for processing on machine M_1

But during the period B_x, the following events may happen:

1) New jobs may arrive to the shop floor and join the queue of machine k,
2) Jobs being processed on other machines join the queue of machine k,
3) Jobs in other queues join the queue of machine k.

It is not possible to predict events pertaining to (1) as arrivals are uncertain and unpredictable. For (2), the related waiting time can be computed. For (3), it is hard to estimate the waiting time involved as the job priorities and ranking in other queues are dependent on dynamic conditions. Formulate (2) into mathematical form, we have:

Waiting time due to jobs from other machines, $W_O = \sum_{i \in M} B_{j,k}^*$

Where,

$B_{j,k}^*$ = processing time of job j on machine k, for a job that will join machine k next but currently being processed on other machines. The remaining processing time of the job on the other machine must be less than or equal B_x

Hence, we have:

Revised net total waiting time, $W_k^{**} = \sum_{j=1}^{n} B_{j,k} + R_k - B_x + W_O$ **(6.10)**

It has been shown earlier that for single-machine queuing model, SPT rule gives a better mean cycle time performance. To incorporate this portion into dispatching for a network of queue, we consider the additional waiting time incurred if a job in the queue other than the job with the shortest processing time is selected for processing on machine M_1 next. Hence, we have:

Additional waiting time, $\Delta W = n \cdot \left(B_x - \min\{B_1, B_2, ..., B_n\} \right)$ **(6.11)**

Where,

n = number of jobs in queue of machine M_1

B_x = processing time of the job to be selected for processing on machine M_1

$\{B_1, B_2, ..., B_n\}$ = processing times of jobs in the queue of machine M_1

Combining (6.10) and (6.11), we have the priority index for a job J_i on machine M_1 as:

Priority index, $Z_i = W_k^{**} + \Delta W$ **(6.12)**

This priority index is computed for each job in the queue of machine M_1, and the job with the lowest value of the index will be selected for processing next. In case of tie, the job with the shortest processing time will be selected first, and if two jobs have the same processing time, first-in-first out (FIFO) rule applies.

To evaluate the performance of the proposed dispatching rule (mPT+WINQ), we have conducted simulation experiments and compared it to other scheduling heuristics as shown in Tables 6.7, 6.8, 6.9 and 6.10. Table 6.7 shows the results for breakdown duration of 30 unit time whereas Table 6.8 presents the results for breakdown duration of 600 unit time. Tables 6.9 and 6.10 display the corresponding results for cycle time variance. Note that only selected scheduling heuristics are shown. The new dispatching rule outperforms PT+WINQ by 1 to 2.7% in terms of mean cycle time for various shop floor conditions, and it is comparable to PT+WINQ in terms of cycle time variance.

Table 6.7: Mean cycle time of selected scheduling heuristics for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
PT+WINQ	215.22	**381.26**	**376.52**	**864.10**
mPT+WINQ	213.01	**374.46**	**370.26**	**847.17**
rPT+WINQ	**208.04**	388.61	395.47	1054.38
rDR1	**207.67**	387.66	430.55	1104.46

Table 6.8: Mean cycle time of selected dispatching rules for breakdown duration of 600 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdowns 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
FIFO	869.37	4387.18	2661.38	14622.60
SPT	813.55	4266.13	2303.67	13072.70
PT+WINQ	**762.08**	**3717.22**	**1872.30**	**9737.34**
mPT+WINQ	**748.69**	**3652.41**	**1821.14**	**9511.73**

Table 6.9: Cycle time variance of selected scheduling heuristics for breakdown duration of 30 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
FIFO	**71.48**	**131.74**	**180.94**	**364.40**
PT+WINQ	75.89	156.45	200.05	**537.37**
mPT+WINQ	74.58	**156.31**	**199.60**	543.90
rFIFO	**73.50**	166.38	219.02	733.28

Table 6.10: Cycle time variance of selected dispatching rules for breakdown duration of 600 (U70, U90 – Utilization 70%, 90%; D05, D25 – Breakdown 5%, 25%)

Heuristics	U70, D05	U70, D25	U90, D05	U90, D25
FIFO	**532.32**	**1577.86**	**897.47**	**2861.62**
SPT	691.11	2266.26	**1608.73**	6863.88
PT+WINQ	**685.69**	**2143.88**	1706.89	**6573.86**
mPT+WINQ	695.78	2151.92	1704.28	6586.52

Based on the experimental results for mean cycle time performance criteria, we propose to use the recursive simulation based algorithm (rDR1) for shop floor conditions of low utilization level, low percentage of breakdown and short breakdown duration, and deploy the proposed mPT+WINQ for other shop floor conditions.

6.7 Summary

In this chapter, we evaluate the performance of a set of dispatching rules and scheduling algorithms under different set of shop floor conditions. We first examine the impact of frequency of rescheduling on scheduling heuristics. We then investigate the performance of the scheduling heuristics.

The evaluation results on static job shop problems show that the recursive simulation heuristics are comparable in performance to other efficient procedures such as Tabu search, and generally achieve lower computation costs particularly for simpler recursive simulation based algorithms.

The experimental results on frequency of rescheduling show that increasing rescheduling periods increases the mean cycle time of jobs in a linear manner. The results match with our analytical models. The simulation studies further indicate that prolonged breakdown duration has a significant impact on the mean cycle time of jobs. With long breakdown duration, event-driven rescheduling or high frequency of rescheduling has no substantial advantages over lower frequency of rescheduling.

We derive a single-machine queuing model that considers machine breakdowns to analyze the characteristics of event-driven and periodic rescheduling under different set of the factors. The model indicates that mean cycle time of jobs is proportional to the duration of machine down times while other factors are kept constant. The analytical model can also be used to explain the effect of changes of other factors such as utilization levels and frequency of breakdowns, on a shop floor.

The experiments on the scheduling heuristics indicate that the recursive simulation based scheduling algorithms perform well for mean cycle time measure under the condition of low utilization level, low percentage of breakdown and short breakdown duration. Dispatching rule, PT+WINQ outperforms other scheduling heuristics for mean cycle time performance under other different shop floor conditions. FIFO rule, which is a poor performer in terms of mean cycle time, stands out in the criterion of cycle time variance. The results on mean cycle time are in contrast to our benchmarking results on static job shops. The study shows that a scheduling heuristic that perform well in static job shop conditions may perform poorly in dynamic and stochastic job shop environments. The results indicate that good schedules can be vulnerable to changes happening in the shop floors. This could be due to the unpredictable nature of dynamic and stochastic shop floors: new jobs constantly arrive, breakdowns can occur anytime on any machine, and so on.

The latter part of our study focuses on deriving a better heuristic for mean cycle time performance under dynamic and stochastic shop floors. We analyze a single-machine queuing model and show that shortest processing time (SPT) dispatching rule gives a better mean cycle time performance. However, SPT rule underperforms other rules (such as PT+WINQ) in a network of queues, as shown in our experimental results. This is due to the fact that for multi-machine job shops, complex interactions exist between jobs and machines. We further study on a network of queues, and observe one of the important criteria in selecting which job to process next on a machine should also depend on the conditions of the queue to which the job's succeeding operation will be processed. Large number of jobs waiting in a queue usually implies that the machine is a bottleneck machine. It is therefore hypothesized that selecting a job that later joins a less congested queue is an important criterion in job dispatching.

Based on the hypothesis, we derive a formula to compute the priority index for job dispatching. The formula incorporates both the processing time and the next operation's possible waiting time of a job. The experimental results on the proposed dispatching rule indicate that an improvement of about 1 to 2.7% is recorded for mean cycle time performance as compared to PT+WINQ across all shop floor conditions. In terms of cycle time variance criteria, the new rule is comparable to PT+WINQ.

Based on the experimental results, we conclude that an effective event-driven scheduling approach is to use the recursive simulation based algorithm (rDR1) for shop floor conditions of low utilization level, low percentage of breakdown and short breakdown duration, and to deploy the proposed mPT+WINQ for other shop floor conditions.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

We have discussed the background pertaining to event-driven scheduling for discrete manufacturing. This covers the challenges, the dynamic and stochastic scheduling problems and the issues in the environments. It is generally perceived that the traditional static, deterministic scheduling approaches can no longer satisfy the scheduling requirement of discrete manufacturing facing competitive global market. Scheduling function today needs to achieve good performance while reacting to unanticipated changes. This requirement poses unprecedented technical challenges on the scheduling problems, which give rise to new research opportunity and thus demand research effort to better understand and resolve the issues.

We have reviewed on scheduling and rescheduling ranging from the environments, strategies and methods to performance measures. It can generally be concluded that different manufacturing environments require different scheduling approaches. Static manufacturing environments favor static, deterministic scheduling approaches, less stable environments require reactive or proactive scheduling approaches and highly dynamic and stochastic environments demand dynamic scheduling approaches. Much of the work in the research community has been in the area of rescheduling approaches to ensure solution stability under process disturbances. There is also work in the area relating to the impact of rescheduling on manufacturing performance. The general understanding is that the performance of rescheduling is dependent on the various aspects of manufacturing environments and scheduling techniques. In general, the performance of most optimization methods and dispatching rules tends to converge when there is increasing uncertainty and variability in the systems.

We have presented the problems and analyzed the issues in rescheduling for discrete manufacturing environments. Firstly, we discuss on the characteristics of disturbances that can impact the rescheduling performance. Secondly, we describe various solution features and their inter-relationships under dynamic and stochastic manufacturing environments. These features are solution quality, stability and adequacy. The solution quality is related to the level of satisfying shop performance measures. The solution stability is pertaining to the measure of schedules in resisting changes under shop disruptions. The solution adequacy refers to the validity of a solution after certain elapsed time. Thirdly, we highlight the rescheduling factors that are controllable, and their effects on solution features and the tradeoffs. Fourthly, we review the performance metrics for assessing the performance of rescheduling systems. Fifthly, we formulate steady state and disturbance propagation models to study the effect of a disturbance on the production process. Finally, without losing the key focus on the objectives of our research, we select the important factors in rescheduling to be considered further in our research.

By using analytical models, we understand that the disturbances affect production process, cause deviation from the steady operating conditions, and often have the greatest effect on the critical resources (or bottlenecks) in production systems. The analytical models also show that solution quality and solution stability are conflicting objectives to achieve. Solution quality requires production objectives to be optimized while solution stability demands minimum changes in the revised schedules. We propose to treat solution stability as soft constraints and incorporate into solution quality as one of the performance criteria. This can be performed using standard multi-objective methods such as weighted mean measure. In this way, rescheduling problems become multi-objective optimization problems.

Past research in rescheduling centers on the performance of schedules, and the main emphasis is on the stability of the solutions. We divert from this focus because it is observed that the long-term steady state performance of a shop floor rather than the performance of specific schedules, is more important to discrete manufacturing. Using the long-term performance as the benchmarking criteria, we are able to compare the performance of the popular scheduling techniques in the industry, periodic rescheduling and dynamic dispatching.

We have derived analytical models to evaluate the relative performance of event-driven and periodic rescheduling policies. The models consider event-driven rescheduling policy as single-arrival queuing model while periodic rescheduling policy as group-arrival queuing model. It is shown that event-driven rescheduling can achieve a better performance as compared to periodic rescheduling in terms of mean waiting time of jobs. This implies that rescheduling policy can impact system performance, and hence it is important to consider appropriate planning and control policies in a manufacturing system.

We have designed and developed a simulation test bed to assess the performance of scheduling algorithms and dispatching rules for discrete manufacturing systems. Discrete event simulation is used to emulate shop floors and enables experimentation to be performed on different shop floor models, scheduling algorithms and dispatching rules. We propose and implement a recursive simulation based scheduling algorithm to improve on the performance of dispatching rules. The recursive simulation technique enables different dispatching rules and performance criteria to be incorporated without much effort. Further, multiple levels of recursive simulation runs can be employed if necessary to enhance performance if computation costs are not an issue. The results on static job shop problems show that the recursive simulation heuristic is comparable in performance to other efficient procedures such as Tabu search.

We evaluate the performance of a set of scheduling algorithms and dispatching rules under different sets of job shop conditions. The results on the frequency of rescheduling indicate that mean cycle time of jobs increase linearly with the rescheduling periods. These results match with our analytical models. The results on scheduling heuristics indicate that the recursive simulation based scheduling algorithms only perform well in mean cycle time under the conditions of low utilization level and low percentage of breakdown. The results clearly show that schedules generated by optimization algorithms can be vulnerable to changes happening in the shop floors.

Based on the experimental results, we subsequently focus on deriving a better heuristic for mean cycle time performance under dynamic and stochastic shop floor conditions. We analyze a single-machine queuing model and show that shortest processing time (SPT) dispatching rule gives a better mean cycle time performance. We subsequently study a network of queues, and observe that one of the important criteria in selecting which job to process next on a machine should also depend on the conditions of the queue of the machine to which the job's succeeding operation will be processed. Based on this principle, we derive a formula to compute the priority index for job dispatching. The experimental results on the proposed dispatching rule show an improvement of about 1 to 2.7% in the case of mean cycle time performance as compared to PT+WINQ across all different shop floor conditions. In terms of cycle time variance criteria, the new rule is comparable to PT+WINQ.

Last but not least, based on the experimental results, we propose an effective event-driven scheduling approach, which encompasses both rescheduling strategy and heuristic, for dynamic and stochastic discrete manufacturing environments. For mean cycle time performance criteria, we propose to use an effective scheduling algorithm such as the recursive simulation based algorithm for shop floor conditions of low utilization level, low percentage of breakdown and short breakdown duration, and deploy an effective dispatching rule such as the modified PT+WINQ for other shop floor conditions.

7.2 Limitations of the Research

Our research primarily focuses on single-objective scheduling algorithms for representative dynamic and stochastic job shop problems. Real-life manufacturing environments exhibit a variety of more complicated issues, with multiple production objectives, multiple constraints, and complex process routings. To apply our research findings to the actual manufacturing environments, further research needs to be performed to accommodate the inherent complexity in the environments.

Majority of our research concentrates on the performance measure of mean cycle time for jobs. This includes the analysis of event-driven and periodic rescheduling, and the experiments as well as the corresponding analysis. The results on cycle time measure are not directly applicable to other performance measures such as job tardiness, machine utilization, and so on.

Past research has shown that the mean cycle time performance of manufacturing environments is affected by both input control and scheduling policy. Our research mainly focuses on scheduling policy, and thus to obtain more effective mean cycle time performance, it is recommended to use a combined scheduling approach of input control and scheduling policy.

7.3 Recommendations for Further Research

Our work on recursive simulation based scheduling is in its preliminary stage. Much more research can be carried out to improve on the recursive simulation technique. One area is in the use of stochastic simulation based scheduling techniques for the dynamic and stochastic scheduling problems. Another area is to reduce the computation costs. To improve the execution speed, the recursive heuristic can be restricted to important decisions, and the time frame of each recursive run can also be reduced. Different scope and resolution for different decisions can also be considered.

Further, fast methods can be derived to estimate the lower bound of each choice at a decision point prior to the recursive runs to identify the most promising choices. Distributed and parallel simulation technique can also be applied to run multiple recursive runs concurrently. Grid computing, which has become an increasingly popular solution to optimize resource allocation, is a powerful platform that can be used to run multiple recursive runs in parallel. The technology can be used to accelerate the simulation and scheduling if there are huge resources coupled with ultra high bandwidth available for computation.

So far, we have centered our research on algorithmic solution to the event-driven rescheduling problem. Another area of work is on techniques pertaining to architectural configurations, which may provide improvements on the performance of manufacturing systems. Different decomposition techniques, such as highly distributed form of control, may be used to solve the dynamic and stochastic scheduling problems. One of the commonly used distributed approaches is the multi-agent paradigm.

References

[Abumaizar, 1997] R. J. Abumaizar and J. A. Svestka, “Rescheduling job shops under random disruptions”, International Journal of Production Research, Vol. 35, No. 7, pp. 2065-2082, 1997.

[Adams, 1988] J. Adams, E. Balas and D. Zawack, “The shifting bottleneck procedure for job shop scheduling”, Management Science, Vol. 3, pp. 391-401, 1988.

[Adon, 2006] I. Adon, “Algorithmic methods in queueing theory”, Course notes [On-line]. Available: www.win.tue.nl/~iadan/algoritme/notes5.pdf, pp. 6-9, May 2007.

[Akturk, 1999] M. S. Akturk and E. Gorgulu, “Match-up scheduling under a machine breakdown”, European Journal of Operational Research, Vol. 112, No. 1, pp. 81-97, 1999.

[Applegate, 1991] D. Applegate and W. Cook, “A computational study of the job shop scheduling problem”, ORSA Journal on Computing, Vol. 3, No. 2, pp. 149-156, 1991.

[Armentano, 2000] V. A. Armentano and C. R. Scrich, “Tabu search for minimizing total tardiness in a job-shop”, International Journal of Production Economics, Vol. 63, pp. 131-140, 2000.

[Baker, 1960] C. T. Baker and B. P, Dzielinski, “Simulation of a simplified job shop”, Management Science, Vol. 6, No. 1, pp. 311-323, 1960.

[Baker, 1977] K. R. Baker, “An experimental study of the effectiveness of rolling schedules in production planning”, Decision Science, Vol. 8, pp. 19-27, 1977.

[Barman, 1997] S. Barman, “Simple priority rule combinations: An approach to improve both flow time and tardiness”, International Journal of Production Research, Vol. 35, No. 10, pp. 2857-2870, 1997.

[Barrett, 1986] R. T. Barrrett and S. Barman, “A SLAM II simulation study of a simplified flow shop”, Simulation, Vol. 47, No. 5, pp. 181–189, 1986.

[Barrett, 1990] R. T. Barrett and S. N. Kadipasaoglu, “Dispatching rules for a dynamic flow shop”, Production and Inventory Management Journal, First Quarter, Vol. 31, No. 1, pp. 54–58, 1990.

[Bauer, 1991] A. Bauer, R. Bowden, J. Browne, J. Duggan and G. Lyons, Shop floor control systems, from design to implementation, Chapman & Hall, London, 1991.

[Bean, 1991] J. C. Bean, J. R. Birge, J. Mittenthal and C. E. Noon, “Matchup scheduling with multiple resources, release dates and disruptions”, Operations Research, Vol. 39 No. 3, pp. 470-483, 1991.

[Benjamin, 1998] P. Benjamin, M. Erraguntla, D. Delen and R. Mayer, “Simulation modeling at multiple levels of abstraction”, In Proceedings of the 1998 Winter Simulation Conference, ed. D. J. Medeiros, E. F. Watson, J. S. Carson and M. S. Manivannan, pp. 391-398, 1998.

[Berry, 1993a] P. M. Berry, “Current trends in scheduling”, AIRING, No.17; Artificial Intelligence Applications Institute, University of Edinburgh, 1993.

[Berry, 1993b] P. M. Berry, “Uncertainty in scheduling: Probability, problem reduction, abstractions and the user”, IEE Computing and Control Division Colloquium on Advanced Software Technologies for Scheduling, Digest No: 1993/163, April 1993.

[Bierwirth, 1999] C. Bierwirth and D. C. Mattfeld, “Production scheduling and rescheduling with genetic algorithms”, Evolutionary Computation, Vol. 7, No. 1, pp. 1-17, 1999.

[Blackstone, 1982] J. H. Blackstone, D. T. Philips and G. L. Hogg, “A state-of-the-art survey of dispatching rules for manufacturing job shop operations”, International Journal of Production Research, Vol. 20, No. 1, pp. 27-45, 1982.

[Blazewicz, 1996] J. Blazewicz, W. Domschke and E. Pesch, “The job shop scheduling problem: Conventional and new solution techniques”, European Journal of Operational Research, Vol. 93, No. 1, pp. 1-33, 1996.

[Blum, 2004] C. Blum and M. Sampels, “An ant colony optimization algorithm for shop scheduling problems”. Journal of Mathematical Modeling and Algorithms, Vol. 3, No. 3, pp. 285 - 308.

[Bodner, 1994] D. A. Bodner, S. Narayanan, U. Sreekanth, T. Govindaraj, L. F. McGinnis and C. M. Mitchell, "Analysis of discrete manufacturing systems for developing object-oriented simulation models", In Proceedings of 3rd Industrial Engineering Research Conference Proceedings, pp. 154-159, Atlanta, GA, 1994.

[Bongaerts, 1997] L. Bongaerts, H. V. Brussel and P. Valckenaers, “Schedule execution using perturbation analysis”, International Symposium on Non-Linear dynamics in production processes and systems, Hannover, Germany, September 17-18, 1997.

[Bongaerts, 1998a] L. Bongaerts, H. V. Brussel and P. Valckenaers, P, "Schedule execution using perturbation analysis", In Proceedings of IEEE International Conference on Robotics and Automation, pp. 2747-2752, Leuven, Belgium, May 1998.

[Bongaerts, 1998b] L. Bongaerts, L. Monostori, D. McFarlane and B. Kádár, "Hierarchy in distributed shop floor control", In Proceedings of the 1st International Workshop on Intelligent Manufacturing Systems, IMS Europe 1988, pp. 97-113, April 1998, Lausanne, Switzerland, and in: Computers in Industry, Elsevier.

[Brandimarte, 1997] P. Brandimarte, M. Rigodanza and L. Roero, L, "Conceptual modeling of an object oriented scheduling architecture based on the shifting bottleneck procedure", International Journal of Production Research, Vol. 35, No. 7, pp. 2065–2082, 1997.

[Brandimarte, 2000] P. Brandimarte, M. Rigodanza and L. Roero, "Conceptual modeling of an object oriented scheduling architecture based on the shifting bottleneck procedure", IIE Transactions, Vol. 32, No. 10, pp. 921-929, 2000.

[Branke, 2002] J. Branke and D. C. Mattfeld, "Anticipatory scheduling for dynamic job shop problems", Workshop on On-line Planning and Scheduling, In 6th International Conference on Artificial Intelligence Planning and Scheduling, pp. 3-10, Toulouse, France, April 2002.

[Brown, 1997] S. Brown, F. Chance, J. W. Fowler and J. Robinson. "A centralized approach to factory simulation", Future Fab International, Vol. 3, pp. 83-86, 1997.

[Brueckner, 1998] S. Brueckner, "MASCADA: Manufacturing control systems capable of managing production change and disturbances", Newsletter of AgentLink (The European Network of Excellence for Agent-Based Computing), ed. Woodridge, Vol. 1, No. 1, pp. 16-19, November 1998.

[Buffa, 1968a] E. S. Buffa, Production-Inventory Systems: Planning and Control, Homewood Illinois: Richard D. Irwin, Inc., 1968.

[Buffa, 1968b] E. S. Buffa, Operations Management: Problems and Models, John Wiley & Sons, New York, pp. 454-460, 1968.

[Byeon, 1998] E. S. Byeon, S. D. Wu and R. H. Storer, "Decomposition heuristics for robust job shop scheduling", IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, pp. 303-313, 1998.

[Chen, 1995] B. Chen, A. V. Vliet and G. J. Woeginger, "An optimal algorithm for preemptive on-line scheduling", Operations Research Letters, Vol. 18, pp. 127-131, 1995.

[Chen, 2004] D. Chen, S. J. Turner, W. T. Cai, B. P. Gan and Y. H. Low, "Incremental HLA-based distributed simulation cloning", In Proceedings of the 2004 Winter Simulation Conference, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, pp. 386-394, Dec. 2004.

[Chen, 2002] G. Chen and B. K. Szymanski, "A component-oriented discrete event simulation", In Proceedings of the 2002 Winter Simulation Conference, ed. E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, 2002. pp. 776-782, 2002.

[Chen, 1992] P. Chen, "Heuristic sampling: A method for predicting the performance of tree searching programs", SIAM Journal on Computing, Vol. 21, No. 2, pp. 295-315, 1992.

[Chiu, 1995] C. Chiu and Y. Yih, "A learning based methodology for dynamic scheduling in a distributed manufacturing system", International Journal of Production Research, Vol.33, No.11, pp. 3217-3232, 1995.

[Chong, 2002] C. S. Chong, A. I. Sivakumar and Robert Gay, "Design, development and application of an object oriented simulation toolkit for real-time semiconductor manufacturing scheduling," Winter Simulation Conference 2002, pp. 1984-1956.

[Church, 1992] L. K. Church and R. Uzsoy, "Analysis of periodic and event-driven rescheduling policies in dynamic shops", International Journal of Computer Integrated Manufacturing, Vol. 5, No. 3, pp. 153-163, 1992.

[Conway, 1965] R. W. Conway, "Priority dispatching and job lateness in a job shop", Journal of Industrial Engineering, Vol. 16, pp. 228-237, 1965.

[Conway, 1960] R. W. Conway, B. M. Johnson and W. L. Maxwell, "An experimental investigation of priority dispatching", Journal of Industrial Engineering, Vol. 11, pp. 221-230, 1960.

[Conway, 1962] R. W. Conway and W. L. Maxwell, "Network dispatching by shortest operation discipline", Operations Research, Vol. 10, No. 1, pp. 51-73, 1962.

[Corne, 1999] D. Corne, M. Dorigo and F. Glover, New ideas in optimization, McGraw-Hill, ISBN 0-07-709506-5, 1999.

[Corney, 2002] W. Corney, "Global manufacturing", APICS, The performance advantage, pp. 33-34, Feb 2002.

[Cowling, 2001] P. Cowling and M. Johansson, "Using real time information for effective dynamic scheduling", European Journal of Operational Research, Vol. 139, No. 2, pp. 230-244, 2001.

[Daniels, 1995] R. L. Daniels and P. Kouvelis, "Robust scheduling to hedge against processing time uncertainty in single-stage production", Management Science, Vol. 41, No. 2, pp. 363-376, 1995.

[Dannenbring, 1977] D. G. Dannenbring, "An evaluation of flow shop scheduling heuristics", Management Science, Vol. 23, pp. 1174-1182, 1977.

[Davis, 1985] L. Davis, "Job shop scheduling with genetic algorithms", In Proceedings of the 1st International Conference on Genetic Algorithms and their Applications, pp. 136-140, Pittsburgh, PA, USA, Lawrence Erlbaum Associates, 1985.

[Deane, 1972] R. H. Deane and C. L. Moodie, "A dispatching methodology for balancing workload assignments in a job shop production facility", AIIE Transactions, Vol. 4, pp. 277-285, 1972.

[Demirkol, 1997] E. Demirkol, S, Mehta and R. Uzsoy, "A computational study of shifting bottleneck procedures for shop scheduling problems", Journal of Heuristics, Vol. 3, No. 2, pp. 111-137, 1997.

[Dorn, 1994a] J. Dorn, "Case based reactive scheduling", Chapter 4, Artificial Intelligence in Reactive Scheduling, pp. 32-50, Chapman and Hall (UK), 1994.

[Dorn, 1994b] J. Dorn and R. M. Kerr, “Co-operating scheduling systems communication through fuzzy sets”, In Proceedings of the 2nd IFAC/IFIP/IFORS Workshop on Intelligent Manufacturing Systems, pp. 687-704, 1994.

[Dorn, 1994c] J. Dorn, R. Kerr and G. Thalhammer, “Reactive scheduling in a fuzzy-temporal framework”, Knowledge Based Reactive Scheduling, IFIP Transactions B (Applications in Technology), B-15, pp. 39-54, 1994.

[Dorn, 1995a] J. Dorn, “Case-based reactive scheduling”, Artificial Intelligence in Reactive Scheduling, ed. R. Kerr and E. Szelke, pp. 32-50, London: Chapman & Hall, 1995.

[Dorn, 1995b] J. Dorn, R. Kerr and G. Thalhammer, “Reactive scheduling: Improving the robustness of schedules and restricting the effects of shop floor disturbances by fuzzy reasoning”, International Journal on Human-Computer Studies, Vol. 42, No. 6, pp. 687-704, 1995.

[Dutta, 1990] A. Dutta, “Reacting to scheduling exceptions in FMS environments”, IIE Transactions, Vol. 22, No. 4, pp. 300-314, 1990.

[Efstathiou, 1996] J. Efstathiou, “Anytime heuristic schedule repair in manufacturing industry”, IEE Proceedings -Control Theory and Applications, Vol. 142 No. 2, pp. 114-124, 1996.

[Elvers, 1973] D. A. Elvers, “Job shop dispatching rules using various delivery setting criteria”, Produciton Inventory Management, Vol. 14, pp. 62-73, 1973.

[Elvers, 1974] D. E. Elvers, "The sensitivity of the relative effectiveness of job shop dispatching rules with various arrival distributions", AIIE Transactions, Vol. 6, pp. 41-48, 1974.

[Enns, 2003] S. T. Enns and P. Suwanruji, "A simulation test bed for production and supply chain modeling", In Proceedings of the 2003 Winter Simulation Conference, ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, pp. 1174-1182, 2003.

[Farn, 1979] C. K. Farn and A. P. Muhlemann, "The dynamic aspects of a production scheduling problem", International Journal of Production Research, Vol. 17, No. 1, pp. 15-21, 1979.

[Ferranti, 2000] M. Ferranti, "The new global economy: bracing for the wave", InfoWorld [On-line]. Available: http://www.itworld.com/Tech/2418/IW001127hnglobal/, Nov 2000.

[Fisher, 1963] H. Fisher and G. L. Thompson, "Probabilistic learning combination of local job shop scheduling rules", Industrial Scheduling, pp. 225-251, 1963.

[Fox, 1985] M. S. Fox and K. G. Kempf, "Opportunistic scheduling for robotic assembly", In Proceedings of the IEEE International Conference on Automation and Robotics, pp. 880-889, St.Louis Missouri, 1985.

[French, 1982] S. French, Sequencing and Scheduling: An introduction to the mathematics of the job-shop, Wiley, 1982.

[Frizelle, 1998] G. Frizelle, D. McFarlane and L. Bongaerts, "Disturbance measurement in manufacturing production systems", In Proceedings of the ASI-98 of NOE on ICIMS (Advanced Summer Institute '98 of the Esprit Network of Excellence in Intelligent Control and Integrated Manufacturing Systems), pp. 159-162, Bremen, June 1998.

[Ganesan, 2004] V. K. Ganesan, A. I. Sivakumar and G. Srinivasan, "Hierarchical minimization of completion time variance and makespan in jobshops", Computers & Operations Research, (In press: date of accept: 03-Aug-2004).

[Garner, 1994] B. J. Garner and G. J. Ridley, "Application of neural network process models in reactive scheduling", Knowledge based Reactive Scheduling, IFIP Transactions B (Applications in Technology), B-15, pp.19-28, 1994.

[Gershwin, 1994] S. B. Gershwin, Manufacturing Systems Engineering, PTR Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[Gilmer, 2000] J. B. Gilmer and F. J. Sullivan, "Recursive simulation to aid models of decisionmaking", In Proceedings of the 2000 Winter Simulation Conference, In Proceedings of the 2000 Winter Simulation Conference, ed. J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick. pp. 958-963. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 2000.

[Gilmer, 2003] J. B. Gilmer and F. J. Sullivan, "The use of recursive simulation to support decisionmaking", In Proceedings of the 2003 Winter Simulation Conference, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice. pp. 1116-1121. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 2003.

[Gilmer, 2004] J. B. Gilmer, "An urban terrain abstraction to support decision making using recursive simulation", In Proceedings of the 2004 Winter Simulation Conference, ed. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, pp. 982-988. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, 2004.

[Givry, 1997] S. D. Givry, G. Verfaillie and T. Schiex, "Bounding the optimum of constraint optimization problems", In Proceedings of CP-97, pp. 405-419, Schloss Hagenberg, Austria, 29 October – 1 November, 1997.

[Hansen, 1996] E. A. Hansen and S. Zilberstein, "Monitoring the progress of anytime problem-solving", In Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96), pp. 1229-1234, Portland, Oregon, 1996.

[Harlin, 2002] U. Harlin, T. Ylipää and S. Fjällström, "Production disturbance handling from an operative perspective in advanced manufacturing systems", Humans in complex environment. Innovate – Integrate – Implement. Nordic Ergonomics Society 34th Annual Congress Norrköping, Sweden, 2002.

[Hasle, 1994] G. Hasle and S. F. Smith, "Directing an opportunist scheduler: An empirical investigation on reactive scenario", Chapter 1, Artificial Intelligence in Reactive Scheduling, pp. 1-11, Chapman and Hall (UK), 1994.

[Henseler, 1994] H. Henseler, From reactive to active scheduling by using multi agents, Chapter 2, Artificial Intelligence in Reactive scheduling, pp. 12-18, Chapman and Hall (UK), 1994.

[Herrmann, 1993] J. W. Herrmann, C. Y. Lee and J. L. Snowdon, "A classification of static scheduling problems", Complexity in Numerical Optimization, ed. P. Pardalos, World Scientific, pp. 203-253, Singapore, 1993.

[Herrmann, 1995] J. W. Herrmann, C. Y. Lee and J. Hinchman, "Global job shop scheduling with a genetic algorithm", Production and Operations Management, Vol. 4, No. 1, pp. 30-45, 1995.

[Holloway, 1974] C. A. Holloway and R. T. Nelson, “Job shop scheduling with due dates and variable processing times”, Management Science, Vol. 20, No. 9, 1974.

[Holthaus, 1997] O. Holthaus and C. Rajendran, “Efficient dispatching rules for scheduling in a job shop”, International Journal of Production Economics, Vol. 48, No. 1, pp. 87-105, 1997.

[Hoogeveen, 2000] J. A. Hoogeveen, C. N. Potts and G. J. Woeginger, “On-line scheduling on a single machine: Maximizing the number of early jobs”, Operations Research Letters, Vol. 27, pp. 193-198, 2000.

[Hopp, 1996] W. J. Hopp and M. L. Spearman, Factory Physics: Foundations of Manufacturing Management, Irwin/McGraw-Hill, ISBN 0-256-15464-3, 1996.

[Hunter, 2002] J. Hunter, D. Delp, D. Collins and J. Si, “Understanding a semiconductor process using a full-scale model”, IEEE Transactions on Semiconductor Manufacturing, Vol. 15, No. 2, pp. 285-289, May 2002.

[Irastorza, 1974] J. C. Irastorza and E. H. Deane, “A loading and balancing methodology for job shop control”, AIIE Transactions, Vol. 6, pp. 302-310, 1974.

[Jackson, 1957a] J. R. Jackson, “Networks of waiting lines”, Operations Research., Vol. 5, pp. 518-521, 1957.

[Jackson, 1957b] J. R. Jackson, “Simulation research on job shop production”, Naval Research Logistics Quarterly, Vol. 4, No. 4, pp. 287-295, 1957.

[Jackson, 1997] M. Jackson and C. Johansson, “Real time discrete event simulation of a PCB production system for operational support”, In Proceedings of the 1997 Winter

Simulation Conference, ed. D.J. Medeiros, E.F. Watson, J.S. Carson, and M.S. Manivannan, pp. 832-837, 1997.

[Jain, 1997] A. K. Jain and H. A. Elmaraghy, “Production scheduling/rescheduling in flexible manufacturing”, International Journal of Production Research, Vol. 35, No. 1, pp. 281-309, 1997.

[Jain, 1998] A. S. Jain and S. Meeran, “A state-of-the-art review of job-shop scheduling techniques”, Technical report, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.

[Jain, 1999] A. S. Jain and S. Meeran, “Deterministic job-shop scheduling: past, present and future”, European Journal of Operational Research, Vol. 113, No. 2, pp. 390-434, 1999.

[Jeffcoat, 1993] D. E. Jeffcoat and R. L. Bulfin, “Simulated annealing for resource-constrained scheduling”, European Journal of Operations Research, Vol. 70, pp. 43-51, 1993.

[Jones, 1998] A. Jones and L. C. Rabelo, “Survey of job shop scheduling techniques”, NISTIR, National Institute of Standards and Technology, Gaithersburg, MD, 1998.

[Khoo, 2000] L. P. Khoo, S. G. Lee and X. F. Yin, “A prototype genetic algorithm enhanced multi-objective scheduler for manufacturing systems”, International Journal of Advance Manufacturing Technology, Vol. 16, No. 2, pp. 131-138, 2000.

[Kim, 1994] M. H. Kim and Y. Kim, “Simulation based real time scheduling in a flexible manufacturing systems”, Journal of Manufacturing Systems, Vol. 13, No. 2, pp. 85-93, 1994.

[Kleinrock, 1975] L. Kleinrock, Queuing Systems, Volume L Theory, Wiley, New York, 1975.

[Kobayashi, 1995] S. Kobayashi, I. Ono, and M. Yamamura, "An efficient genetic algorithm for job shop scheduling problems", In Proceedings of the 6th International Conference on Genetic Algorithms, pp. 506-511, Morgan Kaufmann Publishers, San Francisco, CA, 1995.

[Kumar, 1993] P. Kumar, "Re-entrant lines", Queuing Systems: Theory and Applications: Special Issue on Queuing Networks, Vol. 13, pp. 87-110, 1993.

[Kumar, 1994] P. R. Kumar, "Scheduling manufacturing systems of re-entrant lines", Stochastic Modeling and Analysis of Manufacturing Systems, ed. David D. Yao, pp. 325-360, Springer-Verlag, New York, 1994.

[Kutanoglu, 1999] E. Kutanoglu and I. Sabuncuoglu, "An analysis of heuristics in a dynamic job shop with weighted tardiness objectives", International Journal of Production Research, Vol. 37, No. 1, pp. 165-187, 1999.

[Laguna, 1991] A. M. Laguna, A. J. W. Barnes and A. F. Glover, "Tabu search methodology for a single machine scheduling problem", Journal of International Manufacturing, Vol. 2, pp. 63-74, 1991.

[Lampkemeyer, 1991] U. Lampkemeyer, B. C. Schmidt and J. Detand, "FLEXPLAN: Knowledge based planning and control in manufacturing environments", In Proceedings of CIM-Europe 7th Annual Conference, Torino, Italy, 1991.

[Law, 1984] A. M. Law and W. D. Kelton, "Confidence intervals for steady state simulation: A survey of fixed sample size procedures", Operations Research, Vol. 32, pp. 1221-1239, 1984.

[Lawrence, 1997] S. R. Lawrence and E. C. Sewell, "Heuristic, optimal, static, and dynamic schedules when processing times are uncertain", Journal of Operations Management, Vol. 15, pp. 71-82, 1997.

[Lee, 1996] H. S. Lee, S. S. Murthy, S. W. Haider and D. V. Morse, "Primary production scheduling at steelmaking industries", IBM Journal of Research Development, Vol. 40 No. 2, pp. 231-252, 1996.

[Leon, 1994] V. J. Leon, D. Wu, and R. H. Storer, "Robustness measures and robust scheduling for job shops", IIE Transactions, Vol. 26, No. 5, pp. 32-43, 1994.

[Li, 1993] R. K. Li, Y. T. Shyu and S. Adiga, "A heuristic rescheduling algorithm for computer-based production scheduling systems", International Journal of Production Research, Vol. 31, pp. 1815-1826, 1993.

[Lin, 1994] G. Y. Lin and J. J. Solberg, "Autonomous control for open manufacturing systems", Computer Control of Flexible Manufacturing Systems, pp. 169-206, eds. Sanjay B. Joshi and Jeffrey S. Smith, Chapman & Hall, 1994.

[Lin, 1997] S. C. Lin, E. D. Goodman and W. F. Punch, "A genetic algorithm approach to dynamic job-shop scheduling problems", In Proceedings of Seventh International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, San Francisco, pp. 481-488, 1997.

[Litchfield, 2000] J. Litchfield and R. Narasimhan, "Improving job shop performance through process queue management under transfer batching", Production and Operations Management, Vol. 9, No. 4, pp. 336-348, 2000.

[Little, 1995] D. Little, J. Kenworthy, P. Jarvis and K. Porter, "Scheduling across the supply chain", Logistics Information Management, Vol. 8, No. 1 pp. 42-48, 1995.

[Little, 1961] J. D. C. Little, "A proof of the queuing formula L=λW", Operations Research, Vol. 9, pp. 383-387, 1961.

[Liu, 1994] J. Liu, "Collective problem solving through coordination in a society of reactive agents", Technical Report, The Robotics Institute, Carnegie Mellon University, 1994.

[Liu, 1996] J. Liu and B. L. MacCarthy, "Classification of FMS scheduling problems", International Journal of Production Research, Vol. 34, No. 3, pp. 647-656, 1996.

[Lu, 1994] S. C. H. Lu, D. Ramaswamy and P. R. Kumar, "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants", IEEE Transactions Semiconductor Manufacturing, Vol. 7, No. 3, pp. 374-388, 1994.

[Matsuura, 1993] H. Matsuura, H. Tsubone and M. Kanezashi, 1993, "Sequencing, dispatching, and switching in a dynamic manufacturing environment", International Journal of Production Research, Vol. 31, No. 7, pp. 1671-1688, 1993.

[Mehta, 1998] S. V. Mehta and R. M. Uzsoy, "Predictable scheduling of a job shop subject to breakdowns", IEEE Transactions on Robotics and Automation, Vol. 14, pp. 365-378, 1998.

[Metan, 2005] G. Metan and I. Sabuncuoglu, "A simulation based learning machenism for scheduling systems with continuous control and update structure", In Proceedings of the 2005 Winter Simulation Conference, e.d. M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, pp. 2148-2156, 2005.

[Mittler, 1999] M. Mittler and A. K. Schömig, "Comparison of dispatching rules for semiconductor manufacturing using large facility models", In Proceedings of the 30th Winter Simulation Conference pp. 709-713, 1999.

[Miyashita, 1994] K. Miyashita and K. P. Sycara, "Exploiting failure information for the case based schedule repair", Journal of Japanese Society of Artificial Intelligence, Vol. 9, No. 4, pp. 559-568, 1994.

[Miyashita, 1995a] K. Miyashita, "Case based knowledge acquisition for schedule optimization", Artificial Intelligence in Engineering, Vol. 9, No. 4, pp. 277-287, 1995.

[Miyashita, 1995b] K. Miyashita, "CABINS: A framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair", Artificial Intelligence, Vol.76, pp. 377-426, 1995.

[Miyazaki, 1981] S. Miyazaki, "Combined scheduling system for reducing job tardiness in a job shop", International Journal of Production Research, Vol. 19, No. 2, pp. 201-211, 1981.

[Mize, 1992] J. H. Mize, H. C. Bhuskute, D. B. Pratt and M. Kamath, “Modeling of integrated manufacturing systems using an object-oriented Approach”, IIE Transactions, Vol. 24, No. 1, pp. 14-26, July 1992.

[Mönch, 2003] L. Mönch, O. Rose and R. Sturm, “A simulation framework for the performance assessment of shop-floor control systems”, SIMULATION, Vol. 79, No. 3, pp. 163-170, 2003.

[Muhlemann, 1982] A. P. Muhlemann, A. G. Lockett and C. K. Faro, “Job shop scheduling heuristics and frequency of scheduling”, International Journal of Production Research, Vol. 20, No. 2, pp. 227-241, 1982.

[Nanot, 1963] Y. R. Nanot, “An experimental investigation and comparative evaluation of priority disciplines in job-shop-line queuing networks”, Management Sciences Research Project, UCLA, Research Report, No. 87, 1963.

[Naroska, 2002] E. Naroska and U. Schwiegelshohn, “On an on-line scheduling problem for parallel jobs”, Information Processing Letters, Vol. 81, No. 6, pp. 297-304, 2002.

[Nof, 1991] S. Y. Nof and F. H. Grant, “Adaptive/predictive scheduling: Review and a general framework”, Production Planning and Control, Vol. 2, No. 4, pp. 298-312, 1991.

[Nowicki, 1996] E. Nowicki and C. Smutnicki, “A fast taboo search algorithm for the job shop problem”, Management Science, Vol. 42, No. 6, pp. 797-813, 1996.

[Nuijten, 1998] W. P. M. Nuijten and C. L. Pape, "Constraint-based job-shop scheduling with ILOG scheduler", Journal of Heuristics, Vol. 3, No. 4, pp. 271-286, 1998.

[O'Donovan, 1999] R. O'Donovan, R. Uzsoy and K. N. McKay, "Predictable scheduling of a single machine with breakdowns and sensitive jobs", International Journal of Production Research, Vol. 37, No. 18, pp. 4217-4233, 1999.

[Ovacik, 1994a] I. M. Ovacik and R. Uzsoy, "Exploiting shop floor status information to schedule complex job shops", Journal of Manufacturing Systems, Vol.13, No. 2, pp. 73-84, 1994.

[Ovacik, 1994b] I. M. Ovacik and R. Uzsoy, "Rolling horizon algorithms for a single machine dynamic scheduling problem with sequence dependent setup times", International Journal of Production Research, Vol. 32, No. 6, pp. 1243-1263, 1994.

[Panwalkar, 1977] S. S. Panwalkar and W. Iskander, "A survey of scheduling rules", Operations Research, Vol. 25, No.1, pp. 45-61, 1977.

[Parunak, 1991] H. V. D. Parunak, "Characterizing the manufacturing scheduling problem", Journal of Manufacturing Systems, Vol. 10, No. 3, pp. 241-258, 1991.

[Perkins, 1989] J. R. Perkins and P. R. Kumar, "Stable, distributed, real-time scheduling of flexible manufacturing/assembly/disassembly systems", IEEE Transactions on Automatic Control, Vol. 34, No. 2, pp. 139-148, 1989.

[Perregaard, 1998] M. Perregaard and J. Clausen, "Parallel branch-and-bound methods for the job shop scheduling problem", Annals of Operations Research, Vol. 83, pp. 137-160, 1998.

[Pinedo, 1995] M. Pinedo, Scheduling: Theory, algorithms, and systems, Prentice Hall International Series in Industrial and Systems Engineering, Prentice Hall, Inc., New Jersey, USA, 1995.

[Qi, 2002] C. Qi, and A. I. Sivakumar, "Review of job release control methodologies in semiconductor wafer fabrication", In 4th Asia-Pacific Conference on Simulation Evolution and Learning, 2002.

[Rabelo, 1994] R. J. Rabelo and L. M. Camarinha-Matos, A holistic control architecture infrastructure for dynamic scheduling, Chapter 7, Artificial Intelligence in Reactive scheduling, pp. 78-94, Chapman and Hall (UK), 1994.

[Raheja, 2002a] A. S. Raheja and V. Subramaniam, "Reactive schedule repair of job shops", In Proceedings of the Singapore-MIT Alliance's 2nd Annual Symposium: IMST Parallel Session Program, Singapore, January 2002.

[Raheja, 2002b] A. S. Raheja and V. Subramaniam, "Reactive recovery of job shop schedules - a review", International Journal of Advanced Manufacturing Technology, Vol. 19, No. 10, pp. 756-763, 2002.

[Rajendran, 1999] C. Rajendran and O. Holthaus, "A comparative study of dispatching rules in dynamic flowshops and jobshops", European Journal of Operational Research, Vol. 116, pp. 156–170, 1999.

[Rickel, 1988] J. Rickel, "Issues in the design of scheduling systems," In Proceedings of Conference on Expert Systems and Intelligent Manufacture, pp. 70-89, 1988.

[Rovithakis, 2001] G. A. Rovithakis, S. E. Perrakis and M. A. Christodoulou, “Application of a neural network scheduler on a real manufacturing system”, IEEE Transactions on Control Systems Technology, Vol. 9, No. 2, pp. 261-270, 2001.

[Sabuncuoglu, 1997] I. Sabuncuoglu and S. Karabuk, “Analysis of scheduling-rescheduling problems in a stochastic manufacturing environment”, Technical Report IEOR-9704, Department of Industrial Engineering, Bilkent University, Ankara, 1997.

[Sabuncuoglu, 1999] I. Sabuncuoglu and S.Karabuk, “Rescheduling frequency in an FMS with uncertain processing times and unreliable machines”, Journal of Manufacturing Systems, Vol. 18, No. 4, pp. 268-283, 1999.

[Sabuncuoglu, 2000] I. Sabuncuoglu and M. Bayiz, “Analysis of reactive scheduling problems in a job shop environment”, European Journal of Operations Research, Vol. 126, No. 3, pp. 567-586, November 2000.

[Sadeh, 1994] N. Sadeh, Micro-opportunistic scheduling: The Micro-Boss factory scheduler, Chapter 4 in Intelligent Scheduling, eds. M. Zweben and M. Fox, Morgan Kaufmann Publishers, 1994.

[Satake, 1994] T. Satake, K. Morikawa and N. Nakamura, “Neural network approach for minimizing the makespan of the general job-shop”, International Journal of Production Economics, Vol. 33, pp. 67-74, 1994.

[Satake, 1999] T. Satake, K. Morikawa, K. Takaahashi and N. Nakamura, “Simulated annealing approach for the minimizing the make span of the general job shop”, International Journal of Production Economics, Vol. 61, pp. 515-522, 1999.

[Schmidt, 1994] G. Schmidt, “How to apply fuzzy logic to reactive production scheduling”, Knowledge based Reactive Scheduling, IFIP Transactions B (Applications in Technology), B-15, pp. 57–66, 1994.

[Schmidt, 1996] G. Schmidt, “Case-based reasoning for production scheduling”, Preprints of the 9th International Working Seminar on Production Economics, Vol. 1, pp. 169-175, Innsbruck, Austria, 1996.

[Schrage, 1966] L. E. Schrage and L. W. Miller, “The queue M/G/1 with the shortest remaining processing time discipline”, Operations Research, Vol. 14, No. 4, pp. 670-684, Jul.-Aug. 1966.

[Schrage, 1968] L. Schrage, “A proof of the optimality of the shortest remaining processing time discipline”, Operations Research, Vol. 16, No. 3, pp. 687-690, May-Jun. 1968.

[Schumacher, 2003] J. Schumacher, “A disturbance management approach to improving the performance of batch process operations”, Ph.D. Thesis, Faculty Technology, Policy and Management, ISBN 90-901-6925-3, June 2003.

[Shafaei, 1999a] R. Shafaei and P. Brunn, “Workshop scheduling using practical (inaccurate) data - Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach”, International Journal of Production Research, Vol. 37, No. 17, pp. 3913-3925, 1999.

[Shafaei, 1999b] R. Shafaei, and P. Brunn, “Workshop scheduling using practical (inaccurate) data - Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment”, International Journal of Production Research, Vol. 37, No. 18, pp. 4105-4117, 1999.

[Sivakumar, 1999] A. I. Sivakumar, “Optimization of cycle time and utilization in semiconductor test manufacturing using simulation based, on-line near-real-time scheduling system”, In Proceedings of the 1999 Winter Simulation conference, ed. P.A. Farington, H.B. Nembhard, D.T. Sturrok, and G.W. Evans, pp. 727-735. IEEE, Piscataway, New Jersey, 1999.

[Sivakumar, 2000a] A. I. Sivakumar and C. S. Chong, “Development of an object oriented simulation engine for on-line simulation and optimization”, In Proceedings of the 33rd Annual Simulation Symposium, pp. 291-298, 2000.

[Sivakumar, 2000b] A. I. Sivakumar C. S. Chong and C. C. Wei, “Multi-objective scheduling in semiconductor test manufacturing using simulation based on-line near-real-time system”, International Conference on Modeling and Analysis of Semiconductor Manufacturing, Arizona USA, pp. 200-205, 2000.

[Slany, 1996] W. Slany, “Scheduling as a fuzzy multiple criteria optimization problem”, Fuzzy Sets and Systems, Vol. 78, No. 2, pp. 197–222, 1996.

[Smith, 1978] D. R. Smith, “A new proof of the optimality of the shortest remaining time discipline”, Operations Research, Scheduling, Vol. 26, No. 1, pp. 197-199, Jan.-Feb. 1978.

[Smith, 1994] S. F. Smith, OPIS: A methodology and architecture for reactive scheduling, Intelligent Scheduling, Chapter 2, pp. 29-66, eds. M. Zweben and M. Fox, Morgan Kaufmann Publishers, San Francisco, 1994.

[Smith, 1990] S. F. Smith, P. S. Ow, J. Y. Potvin, N. Muscotella and D. Matthys, “An integrated framework for generating and revising factory schedules”, Journal of Operational Research Society, Vol. 41, No. 6, pp. 539-552, 1990.

[Smith, 1995] S. F. Smith, Reactive scheduling systems, Intelligent Scheduling Systems, eds. D. Brown and W. Scherer, Kluwer Academic Publishing, Boston (Mass.), ISBN/ISSN: 0-7923-9515-8, 1995.

[Smith, 2002] S. F. Smith, "Technologies for dynamic scheduling", Workshop on On-line Planning and Scheduling, In 6th International Conference on Artificial Intelligence Planning and Scheduling, pp. 1-2, Toulouse, France, April 2002.

[Spargg, 1997] J. E. Spargg, G. Fozzard and D. J. Tyler, "Constraint based reactive rescheduling in a stochastic environment", Recent Advances in AI Planning, In Proceedings of 4th European Conference on Planning (ECP'97), pp. 403-413, Springer-Verlag, 1997.

[Stoop, 1996] P. P. M. Stoop and V. C. S. Wiers, "The complexity of scheduling in practice", International Journal of Operations and Production Management, Vol. 16, No. 10, pp. 37-53, 1996.

[Storer, 1992] R. H. Storer, S. D. Wu and R. Vaccari, "New search spaces for sequencing problems with application to job shop scheduling", Management Science, Vol. 38, No. 1, pp 1495-1509, 1992.

[Suh, 1998] M. S. Suh, A. Lee, Y. J. Lee and Y. K. Ko, "Evaluation of ordering strategies for constraint satisfaction reactive scheduling", Decision Support Systems, Vol. 22, pp. 187-197, 1998.

[Sun, 2001] J. Sun and D. Xue, "A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources", Computers in Industry, Vol. 46, No. 2, pp. 189-207, September 2001.

[Suresh, 1993] V. Suresh and D. Chaudhari, "Dynamic scheduling – a survey of research.", International Journal of Production Economics, Vol. 32, No. 1, pp. 53–63, 1993.

[Sushil, 1996] J. L. Sushil and Z. J. Xu, "Genetic algorithms for open shop scheduling and re-scheduling", In Proceedings of ISCA 11th Conference on Computers and Their Applications, pp. 99-102, 1996

[Sycara, 1994] K. P. Sycara and K. Miyashita, "Adaptive schedule repair", IFIP TC5/WG5.7 International Workshop on Knowledge-Based Reactive Scheduling, IFIP Transactions B (Applications in Technology), B-15, pp. 107–123, 1994.

[Szelke, 1994a] E. Szelke and R. M. Kerr, "Knowledge based reactive scheduling - state of the art", International Journal of Production Planning and Control, Vol. 5, pp. 124-145, 1994.

[Szelke, 1994b] E. Szelke and G. Markus, A black board based perspective of reactive scheduling, Chapter 6, Artificial Intelligence in Reactive scheduling, pp. 60-77, Chapman and Hall (UK), 1994.

[Szelke, 1995] E. Szelke and L. Monostori, "Reactive and proactive scheduling with learning in reactive operation management", In Proceedings of IFIP International Working Conference on Managing Concurrent Manufacturing to Improve Industrial Performance, pp. 456-483, Seattle, Washington, USA, 1995.

[Szelke, 1997] E. Szelke and G. Markus, "A learning reactive scheduler using CBR/L", Computer in Industry, Vol. 33, No. 1, pp. 31-46, 1997.

[The Mascada Consortium, 2003] The Mascada Consortium, "Analysis and evaluation of change and disturbances in industrial plants", WP1 Dissemination Report, [On-line] Available: http://www.mech.kuleuven.ac.be/pma/project/mascada/welcome.html, January 2003.

[Vakharia, 1990] A. Vakharia and Y. Chang, "A simulated annealing approach to scheduling a manufacturing cell", Naval Research Logistics, Vol. 37, pp. 559-577, 1990.

[Verfaillie, 1999] G. Verfaillie, S. D. Givry and D. Lesaint, "What is new in on-line scheduling ?" [On-line] Available: http://www.lcr.thomson-csf.fr/projects/planet/ols-characteristics.html, 1999.

[Vieira, 2000a] G. E. Vieira, J. W. Herrmann and E. Lin, "Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies", International Journal of Production Research, Vol. 38, No. 8, pp. 1899-1915, 2000.

[Vieira, 2000b] G. E. Vieira, J. W. Herrmann and E. Lin, "Predicting the performance of rescheduling strategies for parallel machine systems", Journal of Manufacturing Systems, Vol. 19, No. 4, pp. 256-266, 2000.

[Vieira, 2002] G. E. Vieira, J. W. Herrmann and E. Lin, "Rescheduling manufacturing systems: a framework of strategies, policies, and methods", International Journal of Scheduling, Vol. 6, No. 1, pp. 39-62, March 2002.

[Wan, 1995] Y. W. Wan, "Which is better, off-line or real-time scheduling", International Journal of Production Research, Vol. 33, No. 7, pp. 2053-2059, 1995.

[Wang, 1999] C. H. Wang and C. J. J. Paredis, "On-line planning of flexible assembly systems: An agent-based approach", In Proceedings of the 15th ISPE/IEE International Conference on CAD/CAM, Robotics and Factories of the Future, Águas de Lindóia, SP, Brazil, August 1999.

[Wein, 1988] L. M. Wein, "Scheduling semiconductor wafer fabrication", IEEE Transactions on Semiconductor Manufacturing, Vol. 1, No. 3, pp. 115-130, August 1988.

[Wolff, 1982] R. W. Wolff, "Poisson arrivals see time averages", Operations Research., Vol. 30, pp. 223-231, 1982.

[Wolff, 1989] R. W. Wolff, Stochastic modeling and the theory of queues, Prentice-Hall, London, 1989.

[Wooldridge, 1996] M. Wooldridge, S. Bussmann, and M. Klosterberg, "Production sequencing as negotiation", In Proceedings of the 1st International Conference on the Practical Application of Agents and Multi-Agent Systems Technology (PAAM-96), pp. 709-726, London, UK, April 1996.

[Wu, 1995] H. H. Wu and R. K. Li, "A new rescheduling method for computer based scheduling systems", International Journal of Production Research, Vol. 33, No. 8, pp. 2097-2110, 1995.

[Wu, 1989] S. D. Wu and R. A. Wysk, "An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing", International Journal of Production Research, Vol. 27, No. 9, pp. 1603-1623, 1989.

[Wu, 1993] S. D. Wu, R. H. Storer and P. C. Chang, "One-machine rescheduling heuristics with efficiency and stability as criteria", Computers and Operations Research, Vol. 20, No. 1, pp. 1-14, 1993.

[Wu, 1999] S. D. Wu, E. S. Byeon and R. H. Storer, "A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness", Operations Research, Vol. 47, No. 1, pp. 113-124, 1999.

[Wysk, 1986] R. Wysk, D. Wu and R. Yang, "A multi-pass expert control system (MPECS) for flexible manufacturing systems", NBS Special Publication, Vol. 724, pp. 251-278, 1986.

[Yamada, 1992] T. Yamada and R. Nakano, "A genetic algorithm applicable to large scale job shop problems", Parallel Problem Solving from Nature 2, ed. Manner and Manderick, Elsevier, Amsterdam, pp. 281–90, 1992.

[Yamamoto, 1985] M. Yamamoto and S. Y. Nof, "Scheduling/rescheduling in the manufacturing operation system environment", International Journal of Production Research, Vol. 23, No. 4, pp. 705-722, 1985.

[Yan, 2000] H. Yan, S. Lou and S. P. Sethi, "Robustness of various production control policies in semiconductor manufacturing", Production and Operations Management, Vol. 9, No. 2, pp. 171-183, 2000.

[Ylipää, 2002] T. Ylipää, "Correction, prevention and elimination of production disturbances", PROPER project description March 2002, Department of Product and Production Development (PPD), Chalmers University of Technology, Gothenburg, 2002.

APPENDICES

Appendix A: Little's law

Little's law states that:

$$E[L] = \lambda E[W]$$

Where,

$E[L]$ = the mean number of entities in the system

$E[W]$ = the mean waiting time in the system (or the mean sojourn time)

λ = the average number of entities entering the system per unit time

Note that it is assumed that the capacity of the system is sufficient to deal with the arriving entities (i.e. the number of entities in the system does not grow to infinity).

Little's law can be proved using an intuitive argument [Kleinrock, 1975]. Consider the positions at the "entrance" and "exit" of a queuing system where the number of entities that arrive and leave during an interval of arbitrary length, τ is counted respectively, starting from an empty system at time $t = 0$ (see Figure A.1). We let:

$a(\tau)$ = number of arrivals at the queuing system in $[0, \tau]$

$c(\tau)$ = number of service completions in $[0, \tau]$

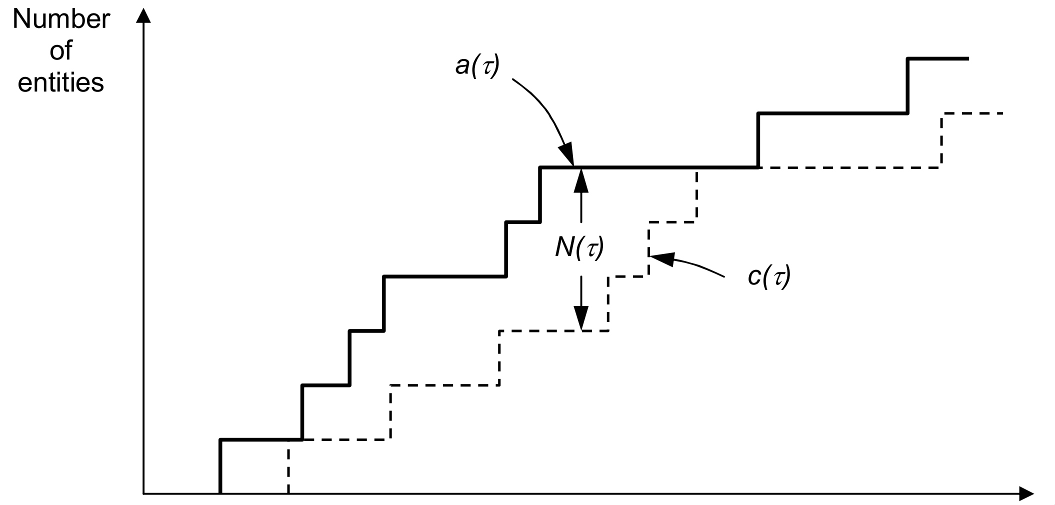


Figure A.1: Arrivals and service completions at a queuing system

If the system is empty at $t = 0$, the number of entities in the system at the time $t = \tau$ is given by:

$$N(\tau) = a(\tau) - c(\tau) \tag{A.1}$$

We can use (A.1) to express the total amount of time $l(\tau)$, spent by all entities in the queuing system during the interval $[0, \tau]$:

$$l(\tau) = \int_0^\tau N(t)\,dt = \int_0^\tau [a(t) - c(t)]\,dt \tag{A.2}$$

Clearly, $l(\tau)$ represents the area between the functions $a(\tau)$ and $c(\tau)$, as illustrated in Figure A.1. The average number of entities $\overline{N}(\tau)$ in the queuing system during the interval $[0, \tau]$ can thus be obtained by dividing the total amount of time spent by all entities in the queuing system $l(\tau)$, by the time τ :

$$\overline{N}(\tau) = \frac{l(\tau)}{\tau} = \frac{l(\tau)}{a(\tau)}\frac{a(\tau)}{\tau} \tag{A.3}$$

The ratio $a(\tau)/\tau$ is the average number of arrivals per unit of time (i.e. the arrival rate) during the interval τ and can be denoted as $\overline{\lambda}_\tau$. Similarly, $l(\tau)/a(\tau)$ is the average time spent by an entity in the queuing system during the interval $\left[0,\tau\right]$ and can be denoted as $\overline{W}_{\tau}$. We can then write:

$$\overline{N}(\tau) = \overline{\lambda}_{\tau}\overline{W}_{\tau} \quad \textbf{(A.4)}$$

If we now let the length of the interval τ, tend to infinity, and if the limits of the quantities exist and from (A.4) we have the relationship:

$$\overline{L} = \lambda\overline{W} \quad \textbf{(A.5)}$$

A few remarks can be noted:

1) In deriving (A.5), the entities are counted at the “entrance” and “exit” of the queuing system. The same analysis could also be performed by counting at the entries to the queuing system (as before) and exits from the queue (or, in other words, entries to the service facility). In that case we could have derived the result.

   $$\overline{L}_q = \lambda\overline{W}_q \quad \textbf{(A.6)}$$

   Where $\overline{L}_q$ and $\overline{W}_q$ are the average number and average stay in the queue respectively.

   In a similar way, but by focusing now on the service facility itself (i.e. by counting entries and exits from the service facility), we could show that

   $$\overline{L}_s = \lambda E\left[B\right] = \frac{\lambda}{\mu} \quad \textbf{(A.7)}$$

   Where $\overline{L}_s$, is the (steady-state) average number of entities in the service facility and $E\left[B\right]$, is the mean service time of the service facility (Note that this result is independent of the number of servers, m. What matters on the right-hand side of (A.7) is the average amount of time an entity spends in the facility, $E\left[B\right]$).

2) (A.5) and (A.6) hold irrespective of the method used to determine the order of entry to the service facility. They also hold for the case where entities belong to a number of distinct classes to which different levels of priority are assigned. Within each of the classes, (A.5) and (A.6) are valid.

3) The only condition that was placed on the arrival process at the queuing system is that the quantity $\lim_{\tau \to \infty} \frac{a(\tau)}{\tau}$, the long-term arrival rate, be finite.

Appendix B: PASTA property

For queuing systems with Poisson arrivals, that is $M/\cdot/\cdot$ systems, a special property holds that arriving entities find on average the same situation in the queuing system as an outside observer looking at the system at an arbitrary point in time. More precisely, the fraction of entities finding on arrival the system in some state A is exactly the same as the fraction of time the system is in state A. This property is only true for Poisson arrivals.

In general this property is not true. For instance, in a $D/D/1$ system which is empty at time 0, and with arrivals at 1, 3, 5 … and service times 1, every arriving customer finds an empty system, whereas the fraction of time the system is empty is ½.

This property of Poisson arrivals is called PASTA property, which is the acronym for Poisson Arrivals See Time Averages. Intuitively, this property can be explained by the fact that Poisson arrivals occur completely random in time. A rigorous proof of the PASTA property can be found in the paper by Wolff [1982; 1989]

Appendix C: Mean value approach to derive the mean waiting time for *M/G/1*

To derive the mean waiting time of entities in the queue, consider a new arriving entity into the queue. The entity first has to wait for the residual service time of the entity in service (if there is one) and then continues to wait for the servicing of all entities which were already waiting in the queue on arrival. Let the random variable R denote the residual service time and let L denote the number of entities waiting in the queue. Hence:

$$E[W] = E[L] \cdot E[B] + E[R]$$

Where,

$E[W]$ = the mean waiting time in the queue

$E[L]$ = the mean number of entities waiting in the queue

$E[B]$ = the mean service time

$E[R]$ = the mean residual service time

And by Little's law

$$E[L] = \lambda E[W]$$

We find:

$$E[W] = \frac{E[R]}{1-\rho} \qquad \textbf{(C.1)}$$

Where,

$$\rho = \lambda E[B]$$

To derive the mean residual service time, consider a long interval of time t (refer to the graph in Figure C.1). The average value of the sawtooth curve can be calculated by dividing the sum of the areas of the triangles by the length of the interval. The triangles may be separated by idle periods (queue empty), and the number of triangles *n*, is determined by the arrival rate λ; mean number is λt.

$$E[R] = \frac{1}{t}\int_0^t R(t)dt = \frac{1}{t}\sum_{i=1}^{n}\frac{1}{2}B_i^2 = \frac{n}{t}\cdot\frac{1}{n}\cdot\sum_{i=1}^{n}\frac{1}{2}B_i^2 = \lambda\cdot\frac{1}{2}E[B^2] \qquad \textbf{(C.2)}$$

Where,

$$\lambda = \frac{n}{t}$$

$$\frac{1}{2}E[B^2] = \frac{1}{2}\cdot\frac{1}{n}\cdot\sum_{i=1}^{n}B_i^2$$

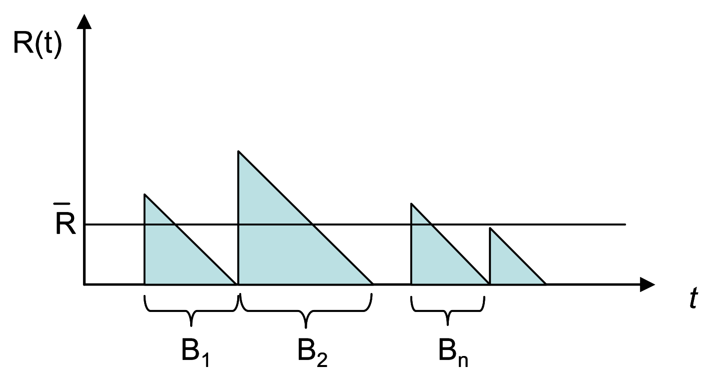


Figure C.1: Evolution of unfinished work in the server

Note that we can rewrite (C.2) as follows:

$$E[R] = \frac{\lambda\cdot E[B]\cdot E[B^2]}{2E[B]} = \frac{\rho E[B^2]}{2E[B]} = \rho E[R']$$

The expression $\rho E[R']$ is more applicable for multiple job type arrivals into the queuing systems.

Substitute (C.2) into (C.1), we have:

$$E[W] = \frac{\lambda E[B^2]}{2(1-\rho)} = \frac{\rho}{1-\rho}\cdot\frac{E[B^2]}{2E[B]} \qquad \textbf{(C.3)}$$

For a random variable *X*, we have the variance, ${\sigma_X}^2$ and coefficient, C_X:

$$Var(X) = \sigma_X{}^2 = E\left[X^2\right] - E\left[X\right]^2 \quad \textbf{(C.4)}$$

$$C_X{}^2 = \frac{Var(X)}{E[X]^2} = \frac{\sigma_X{}^2}{E[X]^2} \quad \textbf{(C.5)}$$

Rearranging (C.4) and substitute (C.5) into (C.4), we have:

$$E\left[X^2\right] = E[X]^2 + \sigma_X^2 = \left(1 + C_X{}^2\right) \cdot E[X]^2 \quad \textbf{(C.6)}$$

Substitute (C.6) into (C.3) and replace X with B, we have:

$$E[W] = \frac{\left(1 + C_B{}^2\right) \cdot \lambda E[B] \cdot E[B]}{2(1-\rho)} = \frac{\left(1 + C_B{}^2\right)}{2} \cdot \frac{\rho}{1-\rho} \cdot E[B]$$

Mean waiting time in the system is given by:

$$\begin{aligned} E[W_s] &= E[B] + E[W] \\ &= E[B] + \frac{\left(1 + C_B{}^2\right)}{2} \cdot \frac{\rho}{1-\rho} \cdot E[B] = \left(1 + \frac{1 + C_B{}^2}{2} \cdot \frac{\rho}{1-\rho}\right) \cdot E[B] \end{aligned}$$

By applying Little's law, we obtain the corresponding expressions for the numbers:

$$E[L] = \lambda E[W] = \frac{\lambda^2 E\left[B^2\right]}{2(1-\rho)} = \frac{1 + C_B{}^2}{2} \cdot \frac{\rho^2}{1-\rho}$$

$$E[L_s] = \lambda E[W_s] = \lambda E[B] + \frac{\lambda E\left[B^2\right]}{2(1-\rho)} = \rho + \frac{1 + C_B{}^2}{2} \cdot \frac{\rho^2}{1-\rho}$$

An important observation is that, clearly, the mean waiting time (and thus the mean number of entities) only depends upon the first two moments of service time (and not upon its distribution or higher moments), i.e. the expectation $E[B]$ and variance σ_B^2. So in practice it is sufficient to know the mean and standard deviation of the service time in order to estimate the mean waiting time (and mean number of entities). Further, mean values increase linearly with the variance.

In the case of the exponential distribution (*M/M/1*) one has:

$$\sigma_B{}^2 = E[B]^2 \Rightarrow C_B{}^2 = 1$$

In the case of constant service time (*M/D/1*) one has:

$$\sigma_B{}^2 = 0 \Rightarrow C_B{}^2 = 0$$

Appendix D: Benchmark Problems

These benchmark problems are job shop problems with standard data representation. Each row of the table represents one job, with odd columns of the table denoting machine index and even columns of the table referring to the processing time of the job on the machine. The machine index starts from 0. A total of 82 standard job shop problems are given as follows:

- 3 problems from Fisher & Thompson [1963], referred as ftp06, ftp10 and ftp20

ftp06 (6 x 6)

2	1	0	3	1	6	3	7	5	3	4	6
1	8	2	5	4	10	5	10	0	10	3	4
2	5	3	4	5	8	0	9	1	1	4	7
1	5	0	5	2	5	3	3	4	8	5	9
2	9	1	3	4	5	5	4	0	3	3	1
1	3	3	3	5	9	0	10	4	4	2	1

ftp10 (10 x 10)

0	29	1	78	2	9	3	36	4	49	5	11	6	62	7	56	8	44	9	21
0	43	2	90	4	75	9	11	3	69	1	28	6	46	5	46	7	72	8	30
1	91	0	85	3	39	2	74	8	90	5	10	7	12	6	89	9	45	4	33
1	81	2	95	0	71	4	99	6	9	8	52	7	85	3	98	9	22	5	43
2	14	0	6	1	22	5	61	3	26	4	69	8	21	7	49	9	72	6	53
2	84	1	2	5	52	3	95	8	48	9	72	0	47	6	65	4	6	7	25
1	46	0	37	3	61	2	13	6	32	5	21	9	32	8	89	7	30	4	55
2	31	0	86	1	46	5	74	4	32	6	88	8	19	9	48	7	36	3	79
0	76	1	69	3	76	5	51	2	85	9	11	6	40	7	89	4	26	8	74
1	85	0	13	2	61	6	7	8	64	9	76	5	47	3	52	4	90	7	45

ftp20 (20 x 5)

0	29	1	9	2	49	3	62	4	44
0	43	1	75	3	69	2	46	4	72
1	91	0	39	2	90	4	12	3	45
1	81	0	71	4	9	2	85	3	22
2	14	1	22	0	26	3	21	4	72
2	84	1	52	4	48	0	47	3	6
1	46	0	61	2	32	3	32	4	30
2	31	1	46	0	32	3	19	4	36
0	76	3	76	2	85	1	40	4	26
1	85	2	61	0	64	3	47	4	90
1	78	3	36	0	11	4	56	2	21
2	90	0	11	1	28	3	46	4	30
0	85	2	74	1	10	3	89	4	33
2	95	0	99	1	52	3	98	4	43

```
 0  6 1 61 4 69 2 49 3 53
 1  2 0 95 3 72 4 65 2 25
 0 37 2 13 1 21 3 89 4 55
 0 86 1 74 4 88 2 48 3 79
 1 69 2 51 0 11 3 89 4 74
 0 13 1  7 2 76 3 52 4 45
```

- 40 problems from Lawrence [1984], referred as la01 – la40

la01 (10 x 5)

```
 1 21 0 53 4 95 3 55 2 34
 0 21 3 52 4 16 2 26 1 71
 3 39 4 98 1 42 2 31 0 12
 1 77 0 55 4 79 2 66 3 77
 0 83 3 34 2 64 1 19 4 37
 1 54 2 43 4 79 0 92 3 62
 3 69 4 77 1 87 2 87 0 93
 2 38 0 60 1 41 3 24 4 83
 3 17 1 49 4 25 0 44 2 98
 4 77 3 79 2 43 1 75 0 96
```

la02 (10 x 5)

```
 0 20 3 87 1 31 4 76 2 17
 4 25 2 32 0 24 1 18 3 81
 1 72 2 23 4 28 0 58 3 99
 2 86 1 76 4 97 0 45 3 90
 4 27 0 42 3 48 2 17 1 46
 1 67 0 98 4 48 3 27 2 62
 4 28 1 12 3 19 0 80 2 50
 1 63 0 94 2 98 3 50 4 80
 4 14 0 75 2 50 1 41 3 55
 4 72 2 18 1 37 3 79 0 61
```

la03 (10 x 5)

```
 1 23 2 45 0 82 4 84 3 38
 2 21 1 29 0 18 4 41 3 50
 2 38 3 54 4 16 0 52 1 52
 4 37 0 54 2 74 1 62 3 57
 4 57 0 81 1 61 3 68 2 30
 4 81 0 79 1 89 2 89 3 11
 3 33 2 20 0 91 4 20 1 66
 4 24 1 84 0 32 2 55 3  8
 4 56 0  7 3 54 2 64 1 39
 4 40 1 83 0 19 2  8 3  7
```

la04 (10 x 5)

```
 0 12 2 94 3 92 4 91 1  7
 1 19 3 11 4 66 2 21 0 87
```

```
1 14 0 75 3 13 4 16 2 20
2 95 4 66 0  7 3  7 1 77
1 45 3  6 4 89 0 15 2 34
3 77 2 20 0 76 4 88 1 53
2 74 1 88 0 52 3 27 4  9
1 88 3 69 0 62 4 98 2 52
2 61 4  9 0 62 1 52 3 90
2 54 4  5 3 59 1 15 0 88
```

la05 (10 x 5)

```
1 72 0 87 4 95 2 66 3 60
4  5 3 35 0 48 2 39 1 54
1 46 3 20 2 21 0 97 4 55
0 59 3 19 4 46 1 34 2 37
4 23 2 73 3 25 1 24 0 28
3 28 0 45 4  5 1 78 2 83
0 53 3 71 1 37 4 29 2 12
4 12 2 87 3 33 1 55 0 38
2 49 3 83 1 40 0 48 4  7
2 65 3 17 0 90 4 27 1 23
```

la06 (15 x 5)

```
1 21 2 34 4 95 0 53 3 55
3 52 4 16 1 71 2 26 0 21
2 31 0 12 1 42 3 39 4 98
3 77 1 77 4 79 0 55 2 66
4 37 3 34 2 64 1 19 0 83
2 43 1 54 0 92 3 62 4 79
0 93 3 69 1 87 4 77 2 87
0 60 1 41 2 38 4 83 3 24
2 98 3 17 4 25 0 44 1 49
0 96 4 77 3 79 1 75 2 43
4 28 2 35 0 95 3 76 1  7
0 61 4 10 2 95 1  9 3 35
4 59 3 16 1 91 2 59 0 46
4 43 1 52 0 28 2 27 3 50
0 87 1 45 2 39 4  9 3 41
```

la07 (15 x 5)

```
0 47 4 57 1 71 3 96 2 14
0 75 1 60 4 22 3 79 2 65
3 32 0 33 2 69 1 31 4 58
0 44 1 34 4 51 3 58 2 47
3 29 1 44 0 62 2 17 4  8
1 15 2 40 0 97 4 38 3 66
2 58 1 39 0 57 4 20 3 50
2 57 3 32 4 87 0 63 1 21
4 56 0 84 2 90 1 85 3 61
4 15 0 20 1 67 3 30 2 70
4 84 0 82 1 23 2 45 3 38
3 50 2 21 0 18 4 41 1 29
4 16 1 52 0 52 2 38 3 54
4 37 0 54 3 57 2 74 1 62
```

```
 4 57 1 61 0 81 2 30 3 68
```

la08 (15 x 5)

```
 3 92 2 94 0 12 4 91 1  7
 2 21 1 19 0 87 3 11 4 66
 1 14 3 13 0 75 4 16 2 20
 2 95 4 66 0  7 1 77 3  7
 2 34 4 89 3  6 1 45 0 15
 4 88 3 77 2 20 1 53 0 76
 4  9 3 27 0 52 1 88 2 74
 3 69 2 52 0 62 1 88 4 98
 3 90 0 62 4  9 2 61 1 52
 4  5 2 54 3 59 0 88 1 15
 0 41 1 50 4 78 3 53 2 23
 0 38 4 72 2 91 3 68 1 71
 0 45 3 95 4 52 2 25 1  6
 3 30 1 66 0 23 4 36 2 17
 2 95 0 71 3 76 1  8 4 88
```

la09 (15 x 5)

```
 1 66 3 85 2 84 0 62 4 19
 3 59 1 64 2 46 4 13 0 25
 4 88 3 80 1 73 2 53 0 41
 0 14 1 67 2 57 3 74 4 47
 0 84 4 64 2 41 3 84 1 78
 0 63 3 28 1 46 2 26 4 52
 3 10 2 17 4 73 1 11 0 64
 2 67 1 97 3 95 4 38 0 85
 2 95 4 46 0 59 1 65 3 93
 2 43 4 85 3 32 1 85 0 60
 4 49 3 41 2 61 0 66 1 90
 1 17 0 23 3 70 4 99 2 49
 4 40 3 73 0 73 1 98 2 68
 3 57 1  9 2  7 0 13 4 98
 0 37 1 85 2 17 4 79 3 41
```

la10 (15 x 5)

```
 1 58 2 44 3  5 0  9 4 58
 1 89 0 97 4 96 3 77 2 84
 0 77 1 87 2 81 4 39 3 85
 3 57 1 21 2 31 0 15 4 73
 2 48 0 40 1 49 3 70 4 71
 3 34 4 82 2 80 0 10 1 22
 1 91 4 75 0 55 2 17 3  7
 2 62 3 47 1 72 4 35 0 11
 0 64 3 75 4 50 1 90 2 94
 2 67 4 20 3 15 0 12 1 71
 0 52 4 93 3 68 2 29 1 57
 2 70 0 58 1 93 4  7 3 77
 3 27 2 82 1 63 4  6 0 95
 1 87 2 56 4 36 0 26 3 48
 3 76 2 36 0 36 4 15 1  8
```

la11 (20 x 5)

2	34	1	21	0	53	3	55	4	95
0	21	3	52	1	71	4	16	2	26
0	12	1	42	2	31	4	98	3	39
2	66	3	77	4	79	0	55	1	77
0	83	4	37	3	34	1	19	2	64
4	79	2	43	0	92	3	62	1	54
0	93	4	77	2	87	1	87	3	69
4	83	3	24	1	41	2	38	0	60
4	25	1	49	0	44	2	98	3	17
0	96	1	75	2	43	4	77	3	79
0	95	3	76	1	7	4	28	2	35
4	10	2	95	0	61	1	9	3	35
1	91	2	59	4	59	0	46	3	16
2	27	1	52	4	43	0	28	3	50
4	9	0	87	3	41	2	39	1	45
1	54	0	20	4	43	3	14	2	71
4	33	1	28	3	26	0	78	2	37
1	89	0	33	2	8	3	66	4	42
4	84	0	69	2	94	1	74	3	27
4	81	2	45	1	78	3	69	0	96

la12 (20 x 5)

1	23	0	82	4	84	2	45	3	38
3	50	4	41	1	29	0	18	2	21
4	16	3	54	1	52	2	38	0	52
1	62	3	57	4	37	2	74	0	54
3	68	1	61	2	30	0	81	4	57
1	89	2	89	3	11	0	79	4	81
1	66	0	91	3	33	4	20	2	20
3	8	4	24	2	55	0	32	1	84
0	7	2	64	1	39	4	56	3	54
0	19	4	40	3	7	2	8	1	83
0	63	2	64	3	91	4	40	1	6
1	42	3	61	4	15	2	98	0	74
1	80	0	26	3	75	4	6	2	87
2	39	4	22	0	75	3	24	1	44
1	15	3	79	4	8	0	12	2	20
3	26	2	43	0	80	4	22	1	61
2	62	1	36	0	63	3	96	4	40
1	33	3	18	0	22	4	5	2	10
2	64	4	64	0	89	1	96	3	95
2	18	4	23	3	15	1	38	0	8

la13 (20 x 5)

3	60	0	87	1	72	4	95	2	66
1	54	0	48	2	39	3	35	4	5
3	20	1	46	0	97	2	21	4	55
2	37	0	59	3	19	1	34	4	46
2	73	3	25	1	24	0	28	4	23
1	78	3	28	2	83	0	45	4	5
3	71	1	37	2	12	4	29	0	53
4	12	3	33	1	55	2	87	0	38

```
 0 48 1 40 2 49 3 83 4  7
 0 90 4 27 2 65 3 17 1 23
 0 62 3 85 1 66 2 84 4 19
 3 59 2 46 4 13 1 64 0 25
 2 53 1 73 3 80 4 88 0 41
 2 57 4 47 0 14 1 67 3 74
 2 41 4 64 3 84 1 78 0 84
 4 52 3 28 2 26 0 63 1 46
 1 11 0 64 3 10 4 73 2 17
 4 38 3 95 0 85 1 97 2 67
 3 93 1 65 2 95 0 59 4 46
 0 60 1 85 2 43 4 85 3 32
```

la14 (20 x 5)

```
 3  5 4 58 2 44 0  9 1 58
 1 89 4 96 0 97 2 84 3 77
 2 81 3 85 1 87 4 39 0 77
 0 15 3 57 4 73 1 21 2 31
 2 48 4 71 3 70 0 40 1 49
 0 10 4 82 3 34 2 80 1 22
 2 17 0 55 1 91 4 75 3  7
 3 47 2 62 1 72 4 35 0 11
 1 90 2 94 4 50 0 64 3 75
 3 15 2 67 0 12 4 20 1 71
 4 93 2 29 0 52 1 57 3 68
 3 77 1 93 0 58 2 70 4  7
 1 63 3 27 0 95 4  6 2 82
 4 36 0 26 3 48 2 56 1 87
 2 36 1  8 4 15 3 76 0 36
 4 78 1 84 3 41 0 30 2 76
 1 78 0 75 4 88 3 13 2 81
 0 54 4 40 2 13 1 82 3 29
 1 26 4 82 0 52 3  6 2  6
 3 54 1 64 0 54 2 32 4 88
```

la15 (20 x 5)

```
 0  6 2 40 1 81 3 37 4 19
 2 40 3 32 0 55 4 81 1  9
 1 46 4 65 2 70 3 55 0 77
 2 21 4 65 0 64 3 25 1 15
 2 85 0 40 1 44 3 24 4 37
 0 89 4 29 1 83 3 31 2 84
 4 59 3 38 1 80 2 30 0  8
 0 80 2 56 1 77 4 41 3 97
 4 56 0 91 3 50 2 71 1 17
 1 40 0 88 4 59 2  7 3 80
 0 45 1 29 2  8 4 77 3 58
 2 36 0 54 3 96 1  9 4 10
 0 28 2 73 1 98 3 92 4 87
 0 70 3 86 2 27 1 99 4 96
 1 95 0 59 4 56 3 85 2 41
 1 81 2 92 4 32 0 52 3 39
 1  7 4 22 2 12 0 88 3 60
 3 45 0 93 2 69 4 49 1 27
 0 21 1 84 2 61 3 68 4 26
```

```
 1 82  2 33  4 71  0 99  3 44
```

la16 (10 x 10)

```
 1 21  6 71  9 16  8 52  7 26  2 34  0 53  4 21  3 55  5 95
 4 55  2 31  5 98  9 79  0 12  7 66  1 42  8 77  6 77  3 39
 3 34  2 64  8 62  1 19  4 92  9 79  7 43  6 54  0 83  5 37
 1 87  3 69  2 87  7 38  8 24  9 83  6 41  0 93  5 77  4 60
 2 98  0 44  5 25  6 75  7 43  1 49  4 96  9 77  3 17  8 79
 2 35  3 76  5 28  9 10  4 61  6  9  0 95  8 35  1  7  7 95
 3 16  2 59  0 46  1 91  9 43  8 50  6 52  5 59  4 28  7 27
 1 45  0 87  3 41  4 20  6 54  9 43  8 14  5  9  2 39  7 71
 4 33  2 37  8 66  5 33  3 26  7  8  1 28  6 89  9 42  0 78
 8 69  9 81  2 94  4 96  3 27  0 69  7 45  6 78  1 74  5 84
```

la17 (10 x 10)

```
 4 18  7 21  9 41  2 45  3 38  8 50  5 84  6 29  1 23  0 82
 8 57  5 16  1 52  7 74  2 38  3 54  6 62  9 37  4 54  0 52
 2 30  4 79  3 68  1 61  8 11  6 89  7 89  0 81  9 81  5 57
 0 91  8  8  3 33  7 55  5 20  2 20  4 32  6 84  1 66  9 24
 9 40  0  7  4 19  8  7  6 83  2 64  5 56  3 54  7  8  1 39
 3 91  2 64  5 40  0 63  7 98  4 74  8 61  1  6  6 42  9 15
 1 80  7 39  8 24  3 75  4 75  5  6  6 44  0 26  2 87  9 22
 1 15  7 43  2 20  0 12  8 26  6 61  3 79  9 22  5  8  4 80
 2 62  3 96  4 22  9  5  0 63  6 33  7 10  8 18  1 36  5 40
 1 96  0 89  5 64  3 95  9 23  7 18  8 15  2 64  6 38  4  8
```

la18 (10 x 10)

```
 6 54  0 87  4 48  3 60  7 39  8 35  1 72  5 95  2 66  9  5
 3 20  9 46  6 34  5 55  0 97  8 19  4 59  2 21  7 37  1 46
 4 45  1 24  8 28  0 28  7 83  6 78  5 23  3 25  9  5  2 73
 9 12  1 37  4 38  3 71  8 33  2 12  6 55  0 53  7 87  5 29
 3 83  2 49  6 23  9 27  7 65  0 48  4 90  5  7  1 40  8 17
 1 66  4 25  0 62  2 84  9 13  6 64  7 46  8 59  5 19  3 85
 1 73  3 80  0 41  2 53  9 47  7 57  8 74  4 14  6 67  5 88
 5 64  3 84  6 46  1 78  0 84  7 26  8 28  9 52  2 41  4 63
 1 11  0 64  7 67  4 85  3 10  5 73  9 38  8 95  6 97  2 17
 4 60  8 32  2 95  3 93  1 65  6 85  7 43  9 85  5 46  0 59
```

la19 (10 x 10)

```
 2 44  3  5  5 58  4 97  0  9  7 84  8 77  9 96  1 58  6 89
 4 15  7 31  1 87  8 57  0 77  3 85  2 81  5 39  9 73  6 21
 9 82  6 22  4 10  3 70  1 49  0 40  8 34  2 48  7 80  5 71
 1 91  2 17  7 62  5 75  8 47  4 11  3  7  6 72  9 35  0 55
 6 71  1 90  3 75  0 64  2 94  8 15  4 12  7 67  9 20  5 50
 7 70  5 93  8 77  2 29  4 58  6 93  3 68  1 57  9  7  0 52
 6 87  1 63  4 26  5  6  2 82  3 27  7 56  8 48  9 36  0 95
 0 36  5 15  8 41  9 78  3 76  6 84  4 30  7 76  2 36  1  8
 5 88  2 81  3 13  6 82  4 54  7 13  8 29  9 40  1 78  0 75
 9 88  4 54  6 64  7 32  0 52  2  6  8 54  5 82  3  6  1 26
```

la20 (10 x 10)

```
 6  9 1 81 4 55 2 40 8 32 3 37 0  6 5 19 9 81 7 40
 7 21 2 70 9 65 4 64 1 46 5 65 8 25 0 77 3 55 6 15
 2 85 5 37 0 40 3 24 1 44 6 83 4 89 8 31 7 84 9 29
 4 80 6 77 7 56 0  8 2 30 5 59 3 38 1 80 9 41 8 97
 0 91 6 40 4 88 1 17 2 71 3 50 9 59 8 80 5 56 7  7
 2  8 6  9 3 58 5 77 1 29 8 96 0 45 9 10 4 54 7 36
 4 70 3 92 1 98 5 87 6 99 7 27 8 86 9 96 0 28 2 73
 1 95 7 92 3 85 4 52 6 81 9 32 8 39 0 59 2 41 5 56
 3 60 8 45 0 88 2 12 1  7 5 22 4 93 9 49 7 69 6 27
 0 21 2 61 3 68 5 26 6 82 9 71 8 44 4 99 7 33 1 84
```

la21 (15 x 10)

```
 2 34 3 55 5 95 9 16 4 21 6 71 0 53 8 52 1 21 7 26
 3 39 2 31 0 12 1 42 9 79 8 77 6 77 5 98 4 55 7 66
 1 19 0 83 3 34 4 92 6 54 9 79 8 62 5 37 2 64 7 43
 4 60 2 87 8 24 5 77 3 69 7 38 1 87 6 41 9 83 0 93
 8 79 9 77 2 98 4 96 3 17 0 44 7 43 6 75 1 49 5 25
 8 35 7 95 6  9 9 10 2 35 1  7 5 28 4 61 0 95 3 76
 4 28 5 59 3 16 9 43 0 46 8 50 6 52 7 27 2 59 1 91
 5  9 4 20 2 39 6 54 1 45 7 71 0 87 3 41 9 43 8 14
 1 28 5 33 0 78 3 26 2 37 7  8 8 66 6 89 9 42 4 33
 2 94 5 84 6 78 9 81 1 74 3 27 8 69 0 69 7 45 4 96
 1 31 4 24 0 20 2 17 9 25 8 81 5 76 3 87 7 32 6 18
 5 28 9 97 0 58 4 45 6 76 3 99 2 23 1 72 8 90 7 86
 5 27 9 48 8 27 7 62 4 98 6 67 3 48 0 42 1 46 2 17
 1 12 8 50 0 80 2 50 9 80 3 19 5 28 6 63 4 94 7 98
 4 61 3 55 6 37 5 14 2 50 8 79 1 41 9 72 7 18 0 75
```

la22 (15 x 10)

```
 9 66 5 91 4 87 2 94 7 21 3 92 1  7 0 12 8 11 6 19
 3 13 2 20 4  7 1 14 9 66 0 75 6 77 5 16 7 95 8  7
 8 77 7 20 2 34 0 15 9 88 5 89 6 53 3  6 1 45 4 76
 3 27 2 74 6 88 4 62 7 52 8 69 5  9 9 98 0 52 1 88
 4 88 6 15 1 52 2 61 7 54 0 62 8 59 5  9 3 90 9  5
 6 71 0 41 4 38 3 53 7 91 8 68 1 50 5 78 2 23 9 72
 3 95 9 36 6 66 5 52 0 45 8 30 4 23 2 25 7 17 1  6
 4 65 1  8 8 85 0 71 7 65 6 28 5 88 3 76 9 27 2 95
 9 37 1 37 4 28 3 51 8 86 2  9 6 55 0 73 7 51 5 90
 3 39 2 15 6 83 9 44 7 53 0 16 4 46 5 24 1 25 8 82
 1 72 4 48 0 87 2 66 9  5 6 54 7 39 8 35 5 95 3 60
 1 46 3 20 0 97 2 21 9 46 7 37 8 19 4 59 6 34 5 55
 5 23 3 25 6 78 1 24 0 28 7 83 8 28 9  5 2 73 4 45
 1 37 0 53 7 87 4 38 3 71 5 29 9 12 8 33 6 55 2 12
 4 90 8 17 2 49 3 83 1 40 6 23 7 65 9 27 5  7 0 48
```

la23 (15 x 10)

```
 7 84 5 58 8 77 2 44 4 97 6 89 3  5 1 58 9 96 0  9
 6 21 1 87 4 15 5 39 2 81 3 85 7 31 8 57 9 73 0 77
 0 40 5 71 8 34 9 82 3 70 6 22 4 10 7 80 2 48 1 49
```

5	75	2	17	3	7	6	72	4	11	7	62	8	47	9	35	1	91	0	55
9	20	4	12	6	71	7	67	0	64	2	94	8	15	5	50	3	75	1	90
6	93	5	93	1	57	7	70	8	77	4	58	0	52	2	29	9	7	3	68
7	56	0	95	8	48	4	26	2	82	1	63	9	36	3	27	6	87	5	6
3	76	5	15	9	78	1	8	8	41	2	36	4	30	6	84	0	36	7	76
0	75	7	13	2	81	8	29	4	54	6	82	5	88	1	78	9	40	3	13
2	6	1	26	7	32	6	64	4	54	0	52	5	82	3	6	9	88	8	54
8	62	2	67	5	32	0	62	7	69	3	61	1	35	4	72	9	5	6	93
2	78	9	90	0	85	1	72	8	64	6	63	3	11	7	82	5	88	4	7
4	28	9	11	7	50	6	88	0	44	5	31	2	27	1	66	8	49	3	35
2	14	5	39	6	56	4	62	3	97	9	66	7	69	1	7	8	47	0	76
1	18	8	93	7	58	6	47	3	69	9	57	2	41	5	53	4	79	0	64

la24 (15 x 10)

7	8	9	75	0	72	6	74	4	30	8	43	2	38	5	98	1	26	3	19
6	19	8	73	3	43	0	23	1	85	4	39	5	13	9	26	2	67	7	9
1	50	3	93	5	80	4	7	0	55	2	61	6	57	8	72	9	42	7	46
1	68	7	43	4	99	6	60	5	68	0	91	8	11	3	96	9	11	2	72
7	84	2	34	8	40	5	7	1	70	6	74	3	12	0	43	9	69	4	30
8	60	0	49	4	59	5	72	9	63	1	69	7	99	6	45	3	27	2	9
6	71	2	91	8	65	1	90	9	98	4	8	7	50	0	75	5	37	3	17
8	62	7	90	5	98	3	31	2	91	4	38	9	72	1	9	0	72	6	49
4	35	0	39	9	74	5	25	7	47	3	52	2	63	8	21	6	35	1	80
9	58	0	5	3	50	8	52	1	88	6	20	2	68	5	24	4	53	7	57
7	99	3	91	4	33	5	19	2	18	6	38	0	24	9	35	1	49	8	9
0	68	3	60	2	77	7	10	8	60	5	15	9	72	1	18	6	90	4	18
9	79	1	60	3	56	6	91	2	40	8	86	7	72	0	80	5	89	4	51
4	10	2	92	5	23	6	46	8	40	7	72	3	6	1	23	0	95	9	34
2	24	5	29	9	49	8	55	0	47	6	77	3	77	7	8	1	28	4	48

la25 (15 x 10)

8	14	4	75	3	12	2	38	0	76	5	97	9	12	1	29	7	44	6	66
5	38	3	82	2	85	4	58	6	87	9	89	0	43	1	80	7	69	8	92
9	5	1	84	0	43	6	48	4	8	7	7	3	41	5	61	8	66	2	14
2	42	1	8	0	96	5	19	4	59	7	97	9	73	8	43	3	74	6	41
6	55	2	70	3	75	8	42	4	37	7	23	1	48	5	5	9	38	0	7
8	9	2	72	7	31	0	79	5	73	3	95	4	25	6	43	9	60	1	56
0	97	2	64	3	78	5	21	4	94	9	31	8	53	6	16	7	86	1	7
3	86	7	85	9	63	0	61	2	65	4	30	5	32	1	33	8	44	6	59
2	44	3	16	4	11	6	45	1	30	9	84	8	93	0	60	5	61	7	90
7	36	8	31	4	47	6	52	0	32	5	11	2	28	9	35	3	20	1	49
8	20	6	49	7	74	4	10	5	17	3	34	0	85	2	77	9	68	1	84
1	85	5	7	8	71	6	59	4	76	0	17	3	29	2	17	7	48	9	13
2	15	6	87	7	11	1	39	4	39	8	43	0	19	3	32	9	16	5	64
6	32	2	92	5	33	8	82	1	83	7	57	9	99	4	91	3	99	0	8
4	88	7	7	8	27	1	38	3	91	2	69	6	21	9	62	5	39	0	48

la26 (20 x 10)

8	52	7	26	6	71	9	16	2	34	1	21	5	95	4	21	0	53	3	55
4	55	5	98	3	39	9	79	0	12	8	77	6	77	7	66	2	31	1	42
5	37	4	92	2	64	6	54	1	19	7	43	0	83	3	34	9	79	8	62
1	87	5	77	0	93	3	69	2	87	7	38	8	24	6	41	9	83	4	60
2	98	5	25	6	75	9	77	1	49	3	17	8	79	0	44	7	43	4	96

1	7	4	61	0	95	2	35	9	10	8	35	5	28	3	76	7	95	6	9
5	59	9	43	0	46	4	28	6	52	3	16	2	59	1	91	8	50	7	27
5	9	9	43	8	14	7	71	4	20	6	54	3	41	0	87	1	45	2	39
1	28	8	66	0	78	2	37	9	42	3	26	5	33	6	89	4	33	7	8
4	96	3	27	6	78	5	84	2	94	8	69	1	74	9	81	7	45	0	69
4	24	7	32	9	25	2	17	3	87	8	81	5	76	6	18	1	31	0	20
8	90	5	28	1	72	7	86	2	23	3	99	6	76	9	97	4	45	0	58
2	17	4	98	3	48	1	46	8	27	6	67	7	62	0	42	9	48	5	27
0	80	8	50	3	19	7	98	5	28	2	50	4	94	6	63	1	12	9	80
9	72	0	75	4	61	8	79	6	37	2	50	5	14	3	55	7	18	1	41
3	96	2	14	5	57	0	47	7	65	4	75	8	79	1	71	6	60	9	22
1	31	7	47	8	58	3	32	4	44	5	58	6	34	0	33	2	69	9	51
1	44	7	40	2	17	0	62	8	66	6	15	3	29	9	38	5	8	4	97
2	58	3	50	4	63	9	87	0	57	6	21	7	57	8	32	1	39	5	20
1	85	0	84	5	56	3	61	9	15	7	70	8	30	2	90	6	67	4	20

la27 (20 x 10)

3	60	4	48	5	95	0	87	1	72	9	5	8	35	7	39	6	54	2	66
7	37	6	34	0	97	5	55	2	21	3	20	4	59	9	46	8	19	1	46
4	45	2	73	1	24	8	28	0	28	3	25	5	23	7	83	9	5	6	78
0	53	2	12	9	12	1	37	8	33	3	71	6	55	5	29	7	87	4	38
4	90	2	49	9	27	7	65	5	7	6	23	0	48	3	83	8	17	1	40
3	85	4	25	2	84	6	64	9	13	1	66	7	46	8	59	0	62	5	19
5	88	6	67	4	14	0	41	1	73	7	57	2	53	3	80	9	47	8	74
1	78	5	64	4	63	6	46	3	84	0	84	8	28	9	52	7	26	2	41
1	11	0	64	6	97	9	38	2	17	4	85	5	73	3	10	8	95	7	67
3	93	2	95	7	43	1	65	8	32	0	59	6	85	5	46	9	85	4	60
2	61	3	41	5	49	4	23	0	66	7	49	8	70	9	99	1	90	6	17
4	13	7	7	1	98	8	57	0	73	3	73	2	68	5	40	9	98	6	9
9	86	6	76	4	14	3	41	1	85	0	37	8	19	2	17	7	54	5	79
1	40	2	53	7	97	5	87	8	96	4	84	3	16	6	66	9	52	0	95
6	33	1	33	3	87	0	18	2	55	8	13	4	77	7	60	9	42	5	74
7	92	5	91	8	79	2	54	4	69	6	79	3	33	1	61	9	39	0	16
6	82	1	41	4	28	5	64	2	78	3	76	7	6	8	49	9	47	0	58
0	52	5	42	8	24	9	91	3	47	6	88	4	91	7	52	2	28	1	35
5	82	2	76	3	86	6	93	4	84	7	38	8	95	9	37	1	21	0	23
9	77	4	8	6	42	7	64	0	70	2	45	8	45	5	28	3	67	1	86

la28 (20 x 10)

8	32	1	81	4	55	7	40	0	6	5	19	9	81	3	37	2	40	6	9
2	70	3	55	7	21	4	64	1	46	8	25	9	65	0	77	5	65	6	15
7	84	4	89	3	24	1	44	2	85	8	31	9	29	6	83	5	37	0	40
4	80	5	59	0	8	2	30	6	77	3	38	1	80	7	56	9	41	8	97
6	40	2	71	0	91	7	7	9	59	8	80	3	50	5	56	1	17	4	88
7	36	9	10	0	45	6	9	4	54	8	96	2	8	5	77	1	29	3	58
6	99	8	86	3	92	0	28	1	98	4	70	5	87	9	96	2	73	7	27
1	95	3	85	5	56	4	52	0	59	2	41	6	81	8	39	9	32	7	92
1	7	7	69	4	93	6	27	5	22	0	88	8	45	3	60	9	49	2	12
7	33	2	61	8	44	5	26	1	84	6	82	3	68	0	21	9	71	4	99
8	43	0	72	4	30	5	98	9	75	1	26	7	8	6	74	3	19	2	38
6	19	2	67	8	73	1	85	9	26	4	39	7	9	0	23	5	13	3	43
8	72	7	46	5	80	3	93	2	61	4	7	9	42	1	50	0	55	6	57
4	99	0	91	9	11	5	68	7	43	3	96	2	72	8	11	6	60	1	68
9	69	0	43	3	12	8	40	1	70	6	74	2	34	5	7	4	30	7	84
7	99	3	27	4	59	5	72	2	9	6	45	0	49	9	63	1	69	8	60

0	75	3	17	2	91	7	50	8	65	5	37	9	98	1	90	6	71	4	8
9	72	1	9	3	31	6	49	2	91	8	62	7	90	0	72	5	98	4	38
4	35	2	63	5	25	6	35	8	21	7	47	3	52	1	80	0	39	9	74
2	68	5	24	9	58	8	52	0	5	6	20	3	50	7	57	1	88	4	53

la29 (20 x 10)

8	14	2	38	7	44	0	76	5	97	3	12	4	75	6	66	9	12	1	29
0	43	2	85	3	82	5	38	4	58	9	89	8	92	6	87	7	69	1	80
3	41	7	7	9	5	0	43	2	14	4	8	5	61	1	84	8	66	6	48
2	42	3	74	4	59	6	41	1	8	9	73	8	43	0	96	5	19	7	97
7	23	8	42	4	37	6	55	0	7	5	5	2	70	9	38	3	75	1	48
8	9	6	43	7	31	4	25	5	73	3	95	0	79	2	72	9	60	1	56
1	7	5	21	8	53	6	16	4	94	0	97	3	78	2	64	7	86	9	31
2	65	6	59	7	85	1	33	4	30	8	44	0	61	3	86	9	63	5	32
6	45	2	44	5	61	8	93	1	30	7	90	9	84	4	11	3	16	0	60
4	47	7	36	8	31	1	49	3	20	2	28	6	52	9	35	5	11	0	32
2	77	4	10	9	68	5	17	0	85	1	84	8	20	6	49	7	74	3	34
0	17	5	7	1	85	3	29	2	17	4	76	6	59	8	71	9	13	7	48
6	87	4	39	8	43	7	11	2	15	3	32	5	64	0	19	1	39	9	16
5	33	3	99	6	32	4	91	8	82	2	92	9	99	7	57	1	83	0	8
3	91	5	39	2	69	8	27	7	7	6	21	1	38	9	62	4	88	0	48
2	67	7	80	3	24	0	88	4	18	1	44	8	45	9	64	5	80	6	38
9	59	3	72	6	47	4	40	7	21	5	43	0	51	8	52	1	24	2	15
3	70	2	31	6	20	8	76	1	40	7	43	0	32	5	88	9	5	4	77
4	47	5	64	9	85	3	49	7	58	1	26	0	32	8	80	2	14	6	94
5	59	2	96	0	5	7	79	8	34	4	75	3	26	6	9	9	23	1	11

la30 (20 x 10)

6	32	3	16	1	33	8	12	7	70	4	10	9	75	0	82	5	88	2	20
8	39	4	81	3	91	5	56	9	69	1	45	6	59	0	86	2	36	7	68
3	84	2	57	7	41	5	73	4	81	0	88	8	38	9	17	6	83	1	5
4	20	5	6	2	15	8	19	1	30	0	94	6	45	7	17	3	18	9	88
9	24	6	49	5	16	4	11	3	60	7	5	8	63	1	25	2	15	0	45
1	86	8	50	2	77	6	54	9	48	0	93	3	32	7	92	5	45	4	71
5	86	6	90	3	78	9	88	2	57	0	32	7	57	8	86	4	71	1	39
2	59	3	18	9	31	4	41	7	20	5	83	8	65	0	54	6	94	1	69
3	47	4	79	6	76	0	59	1	72	2	8	9	30	5	73	7	57	8	84
0	59	2	89	4	10	7	45	3	8	5	54	6	88	8	20	9	7	1	62
5	63	6	9	4	77	3	37	2	5	8	13	9	79	1	24	7	10	0	82
0	74	1	32	2	61	7	53	4	92	9	20	8	10	3	5	6	45	5	23
2	85	9	51	0	61	5	99	4	37	6	94	1	98	8	65	3	33	7	75
0	51	3	24	5	8	6	30	7	12	8	23	2	7	4	17	9	35	1	81
1	71	5	42	8	68	2	31	6	29	3	63	4	65	9	70	7	27	0	93
1	28	5	38	4	51	7	70	2	33	8	78	9	45	3	90	6	54	0	72
0	18	2	90	4	25	6	92	8	85	5	35	7	29	1	81	9	80	3	59
5	67	2	96	1	38	4	86	0	97	3	94	7	86	6	35	9	82	8	45
2	92	8	51	4	59	6	52	5	8	9	70	1	75	3	54	7	60	0	33
3	98	7	80	5	78	0	82	2	7	9	89	1	69	4	51	8	79	6	62

la31 (30 x 10)

4	21	7	26	9	16	2	34	3	55	8	52	5	95	6	71	1	21	0	53
8	77	5	98	1	42	7	66	2	31	3	39	6	77	9	79	4	55	0	12
2	64	4	92	3	34	1	19	8	62	6	54	7	43	0	83	9	79	5	37

0	93	8	24	3	69	7	38	5	77	2	87	4	60	6	41	1	87	9	83
9	77	0	44	4	96	8	79	6	75	2	98	5	25	3	17	7	43	1	49
3	76	2	35	5	28	0	95	7	95	4	61	8	35	1	7	6	9	9	10
1	91	7	27	8	50	3	16	4	28	5	59	6	52	0	46	2	59	9	43
1	45	7	71	2	39	0	87	8	14	6	54	3	41	9	43	5	9	4	20
2	37	3	26	4	33	9	42	0	78	6	89	7	8	8	66	1	28	5	33
1	74	0	69	5	84	3	27	9	81	7	45	8	69	2	94	6	78	4	96
5	76	7	32	6	18	0	20	3	87	2	17	9	25	4	24	1	31	8	81
9	97	8	90	5	28	7	86	0	58	1	72	2	23	6	76	3	99	4	45
9	48	5	27	6	67	7	62	4	98	0	42	1	46	8	27	3	48	2	17
9	80	3	19	5	28	1	12	4	94	6	63	7	98	8	50	0	80	2	50
2	50	1	41	4	61	8	79	5	14	9	72	7	18	3	55	6	37	0	75
9	22	5	57	4	75	2	14	7	65	3	96	1	71	0	47	8	79	6	60
3	32	2	69	4	44	1	31	9	51	0	33	6	34	5	58	7	47	8	58
8	66	7	40	2	17	0	62	9	38	5	8	6	15	3	29	1	44	4	97
3	50	2	58	6	21	4	63	7	57	8	32	5	20	9	87	0	57	1	39
4	20	6	67	1	85	2	90	7	70	0	84	8	30	5	56	3	61	9	15
6	29	0	82	4	18	3	38	7	21	8	50	1	23	5	84	2	45	9	41
3	54	9	37	6	62	5	16	0	52	8	57	4	54	2	38	7	74	1	52
4	79	1	61	8	11	0	81	7	89	6	89	5	57	3	68	9	81	2	30
9	24	1	66	4	32	3	33	8	8	2	20	6	84	0	91	7	55	5	20
3	54	2	64	6	83	9	40	7	8	0	7	4	19	5	56	1	39	8	7
1	6	4	74	0	63	2	64	9	15	6	42	7	98	8	61	5	40	3	91
1	80	3	75	0	26	2	87	9	22	7	39	8	24	4	75	6	44	5	6
5	8	3	79	6	61	1	15	0	12	7	43	8	26	9	22	2	20	4	80
1	36	0	63	7	10	4	22	3	96	5	40	9	5	8	18	6	33	2	62
4	8	8	15	2	64	3	95	1	96	6	38	7	18	9	23	5	64	0	89

la32 (30 x 10)

6	89	1	58	4	97	2	44	8	77	3	5	0	9	5	58	9	96	7	84
7	31	2	81	9	73	4	15	1	87	5	39	8	57	0	77	3	85	6	21
2	48	5	71	0	40	3	70	1	49	6	22	4	10	8	34	7	80	9	82
4	11	6	72	7	62	0	55	2	17	5	75	3	7	1	91	9	35	8	47
0	64	6	71	4	12	1	90	2	94	3	75	9	20	8	15	5	50	7	67
2	29	6	93	3	68	5	93	1	57	8	77	0	52	9	7	4	58	7	70
4	26	3	27	1	63	5	6	6	87	7	56	8	48	9	36	0	95	2	82
1	8	7	76	3	76	4	30	6	84	9	78	8	41	0	36	2	36	5	15
3	13	8	29	0	75	2	81	1	78	5	88	4	54	9	40	7	13	6	82
0	52	2	6	3	6	5	82	6	64	9	88	8	54	4	54	7	32	1	26
8	62	1	35	4	72	7	69	0	62	5	32	9	5	3	61	2	67	6	93
2	78	3	11	7	82	4	7	1	72	8	64	9	90	0	85	5	88	6	63
7	50	4	28	3	35	1	66	2	27	8	49	9	11	6	88	5	31	0	44
4	62	5	39	0	76	2	14	6	56	3	97	1	7	7	69	9	66	8	47
6	47	2	41	0	64	7	58	9	57	8	93	3	69	5	53	1	18	4	79
7	76	9	81	0	76	6	61	4	77	8	26	2	74	5	22	1	58	3	78
6	30	8	72	3	43	0	65	1	16	4	92	5	95	9	29	2	99	7	64
1	35	3	74	5	16	4	85	0	7	2	81	6	86	8	61	9	35	7	34
1	97	7	43	4	72	6	88	5	17	0	43	8	94	3	64	9	22	2	42
7	99	2	84	8	99	5	98	1	20	6	31	3	74	0	92	9	23	4	89
8	32	0	6	4	55	5	19	9	81	1	81	7	40	6	9	3	37	2	40
6	15	2	70	8	25	1	46	9	65	4	64	7	21	0	77	5	65	3	55
8	31	7	84	5	37	3	24	2	85	4	89	9	29	1	44	0	40	6	83
4	80	0	8	9	41	5	59	7	56	3	38	2	30	8	97	6	77	1	80
9	59	0	91	3	50	8	80	1	17	6	40	2	71	5	56	4	88	7	7
7	36	3	58	4	54	5	77	2	8	6	9	0	45	9	10	1	29	8	96
0	28	3	92	2	73	7	27	8	86	5	87	9	96	1	98	6	99	4	70
9	32	1	95	3	85	6	81	2	41	8	39	7	92	0	59	5	56	4	52
4	93	2	12	5	22	6	27	8	45	7	69	3	60	1	7	0	88	9	49

```
 2 61  5 26  9 71  8 44  0 21  6 82  3 68  7 33  1 84  4 99
```

la33 (30 x 10)

```
 2 38  4 75  9 12  5 97  0 76  1 29  8 14  6 66  7 44  3 12
 0 43  5 38  1 80  3 82  2 85  4 58  6 87  8 92  9 89  7 69
 6 48  4  8  8 66  7  7  2 14  3 41  5 61  0 43  1 84  9  5
 5 19  3 74  6 41  4 59  8 43  2 42  9 73  7 97  1  8  0 96
 3 75  5  5  2 70  8 42  7 23  6 55  1 48  9 38  4 37  0  7
 2 72  7 31  3 95  0 79  4 25  1 56  8  9  9 60  5 73  6 43
 9 31  3 78  6 16  4 94  7 86  5 21  0 97  8 53  1  7  2 64
 3 86  2 65  6 59  8 44  1 33  7 85  0 61  5 32  9 63  4 30
 4 11  5 61  9 84  3 16  7 90  1 30  0 60  8 93  2 44  6 45
 5 11  2 28  0 32  7 36  8 31  4 47  3 20  6 52  9 35  1 49
 5 17  3 34  6 49  1 84  0 85  8 20  7 74  9 68  4 10  2 77
 8 71  5  7  3 29  1 85  4 76  6 59  2 17  0 17  9 13  7 48
 1 39  9 16  4 39  6 87  7 11  3 32  2 15  0 19  5 64  8 43
 5 33  8 82  2 92  1 83  6 32  3 99  9 99  4 91  0  8  7 57
 7  7  0 48  9 62  4 88  6 21  5 39  8 27  3 91  1 38  2 69
 9 64  8 45  3 24  7 80  2 67  4 18  6 38  0 88  5 80  1 44
 2 15  3 72  4 40  7 21  8 52  0 51  9 59  1 24  6 47  5 43
 4 77  7 43  1 40  2 31  8 76  6 20  5 88  3 70  9  5  0 32
 2 14  7 58  9 85  5 64  1 26  6 94  0 32  3 49  8 80  4 47
 9 23  1 11  8 34  4 75  7 79  3 26  2 96  0  5  6  9  5 59
 0 75  2 20  8 10  3 66  6 43  7 37  1  9  9 83  4 68  5 52
 8 54  1 26  4 79  7 88  6 84  0  6  2 54  9 59  3 28  5 42
 4 56  9 29  3 36  0 40  6 86  8 68  2 69  7 23  5 62  1 16
 7 53  1  5  6 17  9 59  2 59  8 78  3 64  0 82  4 13  5 12
 9  7  6 62  7 90  5 83  1 85  3 69  0 16  4 81  2 58  8 66
 7 24  2 65  1 69  5 42  9 82  6 82  0 83  3 46  8 72  4 33
 1 10  8 27  7 43  5 20  4 71  9 65  2 73  6 99  0 24  3 64
 9 35  3 92  0 38  5 35  7 30  8 45  2  8  4 82  1 34  6 21
 5 23  7 84  9  7  4 85  8 60  1 15  2 52  6 94  3 83  0  6
 2 70  6 29  8 27  9 80  4  6  7 39  1 79  0 28  3 66  5 66
```

la34 (30 x 10)

```
 2 51  7 59  1 35  5 73  9 65  0 27  6 13  3 81  8 32  4 74
 4 64  7 33  5 75  2 33  8 10  0 28  3 38  6 53  9 49  1 55
 6 83  1 23  2 72  3  7  9 72  0  6  4 39  5 52  8 90  7 21
 3 82  1 23  2 93  4 78  6 88  7 53  9 28  8 65  5 21  0 61
 4 41  6 12  9 12  3 77  1 70  7 24  0 81  5 73  2 62  8  6
 4 98  3 28  6 42  9 72  0 15  8 15  5 94  2 33  1 51  7 99
 0 32  8 22  9 96  4 15  6 78  3 31  5  7  1 94  2 23  7 86
 7 93  2 97  3 43  5 73  0 24  8 68  9 88  1 42  4 35  6 72
 2 14  0 44  8 13  5 67  1 63  3 49  7  5  4 17  6 85  9 66
 7 82  9 15  3 72  4 26  0  8  1 68  6 21  8 45  2 99  5 27
 4 93  6 23  0 51  8 54  3 49  1 96  2 56  9 36  5 53  7 52
 8 60  0 14  4 70  9 55  1 23  5 83  3 38  2 24  7 37  6 48
 0 62  7 15  8 69  9 23  1 82  6 26  4 45  5 33  3 12  2 37
 6 72  1  9  7 15  5 28  8 92  9 12  0 59  3 64  4 87  2 73
 0 50  1 14  7 90  5 46  3 71  4 48  2 80  9 61  8 24  6 44
 0 22  9 94  5 16  3 73  2 54  8 54  4 46  1 97  6 61  7 75
 9 55  3 67  6 77  4 30  7  6  1 32  8 47  5 93  2  6  0 40
 1 30  3 98  7 79  0 22  6 79  2  7  8 36  9 36  5  9  4 92
 8 37  7 72  2 52  4 31  1 82  9 54  5  7  6 82  3 73  0 49
 1 73  3 83  7 45  2 76  4 43  9 29  0 35  5 92  8 39  6 28
```

```
 2 58 0 26 1 48 8 52 7 34 6 96 5 70 4 98 3 80 9 94
 1 70 8 23 5 26 4 14 6 90 2 93 3 21 0 42 7 18 9 36
 4 28 6 76 7 25 0 17 1 84 2 67 8 87 3 43 9 88 5 84
 7 30 3 91 8 52 4 80 0 21 5  8 9 37 2 15 6 12 1 92
 1 28 4  7 7 46 6 92 2 77 3 15 9 69 8 54 0 47 5 39
 9 50 5 44 2 64 8 38 4 93 6 33 7 75 0 41 1 24 3  5
 7 94 0 17 6 87 2 21 8 92 9 28 1 61 4 63 3 34 5 77
 3 72 8 98 9  5 4 28 2  9 5 95 6 64 1 43 0 50 7 96
 0 85 2 85 8 39 1 98 7 24 3 71 5 60 4 55 9 22 6 35
 3 78 6 49 2 46 1 11 0 90 5 20 9 34 7  6 4 70 8 74
```

la35 (30 x 10)

```
 0 66 2 84 3 26 7 29 9 94 6 98 8  7 5 98 1 45 4 43
 3 32 0 97 6 55 2 88 8 93 9 88 1 20 4 50 7 17 5  5
 4 43 3 68 8 47 9 68 1 57 6 20 5 81 2 60 7 94 0 62
 1 57 5 40 0 78 6  9 2 49 9 17 3 32 4 30 8 87 7 77
 0 52 4 30 3 48 5 48 1 26 9 17 6 93 8 97 7 49 2 89
 7 95 0 33 1  5 6 17 5 70 3 57 4 34 2 61 8 62 9 39
 7 97 5 92 1 31 8  5 2 79 4  5 3 67 0  5 9 78 6 60
 2 79 4  6 7 20 8 45 6 34 3 24 9 26 5 68 1 16 0 46
 7 58 9 50 2 19 8 93 6 49 3 25 5 85 4 50 0 93 1 26
 9 81 6 71 5  7 1 39 2 16 8 42 0 71 4 84 3 56 7 99
 8  9 0 86 9  6 3 71 6 97 5 85 4 16 2 42 7 81 1 81
 4 72 3 24 0 30 8 56 2 43 1 61 7 82 6 40 5 59 9 43
 9 43 1 13 6 70 7 93 0 95 8 12 4 15 2 78 5 97 3 14
 0 14 6 26 1 71 3 46 8 80 5 31 4 37 9 27 7 92 2 67
 2 12 0 43 5 96 6  7 3 45 7 20 1 13 9 29 4 60 8 33
 1 78 5 50 6 84 0 42 8 84 4 30 9 76 2 57 7 87 3 59
 4 49 7 50 1 15 8 13 0 93 6 50 9 32 5 59 3 10 2 35
 1 25 0 47 7 60 8 33 4 53 5 37 9 73 2 22 3 87 6 79
 0 84 6 83 1 71 5 68 9 89 8 11 3 60 4 50 2 33 7 97
 1 14 0 38 6 88 5  5 4 77 7 92 8 24 2 73 9 52 3 71
 7 62 9 19 6 38 3 15 8 64 2 64 4  8 1 61 0 19 5 33
 2 33 5 46 4 74 0 56 6 84 9 83 8 19 7  8 3 32 1 97
 4 50 3 71 6 50 2 97 9  8 0 17 7 19 8 92 5 54 1 52
 8 32 1 79 3 97 5 38 9 49 4 76 6 76 0 56 2 78 7 54
 5 13 3  5 2 25 0 86 1 95 9 28 6 78 8 24 7 10 4 39
 7 48 2 59 0 20 9  7 5 31 6 97 1 89 4 32 3 25 8 41
 5 87 0 18 9 48 2 43 1 30 6 97 7 47 8 65 3 69 4 27
 6 71 5 20 8 20 1 78 3 39 0 17 7 50 2 44 9 42 4 38
 0 50 9 42 3 72 5  7 1 77 7 58 4 78 2 89 6 70 8 36
 3 32 9 95 2 13 0 73 6 97 8 24 4 49 5 57 1 68 7 94
```

la36 (15 x 15)

```
  4 21   3 55   6 71 14 98 10 12   2 34   9 16   1 21   0 53   7 26   8 52   5
95 12 31 11 42 13 39
 11 54   4 83   1 77   7 64   8 34 14 79 12 43   0 55   3 77   6 19   9 37   5
79 10 92 13 62   2 66
  9 83   5 77   2 87   7 38   4 60 12 98   0 93 13 17   6 41 10 44   3 69 11
49   8 24   1 87 14 25
  5 77   0 96   9 28   6   7   4 95 13 35   7 35   8 76 11   9 12 95   2 43   1
75 10 61 14 10   3 79
 10 87   4 28   8 50   2 59   0 46 11 45 14   9   9 43   6 52   7 27   1 91 13
41   3 16   5 59 12 39
  0 20   2 71   4 78 13 66   3 14 12   8 14 42   6 28   1 54   9 33 11 89   8
```

```
26  7 37 10 33  5 43
  8 69  4 96 12 17  0 69  7 45 11 31  6 78 10 20  3 27 13 87  1 74  5
84 14 76  2 94  9 81
  4 58 13 90 11 76  3 81  7 23  9 28  1 18  2 32 12 86  8 99 14 97  0
24 10 45  6 72  5 25
  5 27  1 46  6 67  8 27 13 19 10 80  2 17  3 48  7 62 11 12 14 28  4
98  0 42  9 48 12 50
 11 37  5 80  4 75  8 55  7 50  0 94  9 14  6 41 14 72  3 50 10 61 13
79  2 98 12 18  1 63
  7 65  3 96  0 47  4 75 12 69 14 58 10 33  1 71  9 22 13 32  5 57  8
79  2 14 11 31  6 60
  1 34  2 47  3 58  5 51  4 62  6 44  9  8  7 17 10 97  8 29 11 15 13
66 12 40  0 44 14 38
  3 50  7 57 13 61  5 20 11 85 12 90  2 58  4 63 10 84  1 39  9 87  6
21 14 56  8 32  0 57
  9 84  7 45  5 15 14 41 10 18  4 82 11 29  2 70  1 67  3 30 13 50  6
23  0 20 12 21  8 38
  9 37 10 81 11 61 14 57  8 57  0 52  7 74  6 62 12 30  1 52  2 38 13
68  4 54  3 54  5 16
```

la37 (15 x 15)

```
  5 19  6 64 11 73  9 13  2 84 14 88  3 85 10 41 12 53 13 80  1 66  7
46  8 59  4 25  0 62
  1 67  3 74  7 41  2 57 14 52  0 14  9 64  8 84  6 78  5 47 13 28  4
84 10 63 12 26 11 46
  6 97  8 95  0 64  9 38 10 59 12 95  2 17 11 65 13 93  3 10  5 73  1
11  4 85 14 46  7 67
 10 23 12 49  3 32  4 66  2 43  0 60  8 41  7 61 13 70  9 49 11 17  6
90  1 85 14 99  5 85
  9 98  8 57  3 73  6  9  0 73  7  7  1 98  4 13 13 41  5 40 11 85 10
37  2 68 14 79 12 17
 11 66  7 53  5 86  6 40  0 14  3 19 13 96  4 95  2 54 10 84 12 97  8
16 14 52  1 76  9 87
  4 77  2 55  9 42  5 74 14 91 13 33 10 16 12 54  0 18  3 87  7 60  8
13  6 33  1 33 11 61
  6 41  5 39 11 82  9 64 14 47 10 28  7 78 13 49  1 79  4 58  2 92  3
79 12  6  0 69  8 76
 11 21  5 42  9 91  2 28  0 52  6 88 12 76 13 86 10 23  1 35  7 52  4
91  3 47 14 82  8 24
 11 42  1 93  3 95 13 45  9 28 14 77  0 84 10  8  7 45  4 70  5 37  6
86 12 64  8 67  2 38
  4 97 12 81  1 58  7 84  5 58  0  9 11 87  3  5  2 44 13 85  6 89 10
77  9 96 14 39  8 77
 12 80  1 21 10 10  5 73  8 70  6 49  2 31 13 34  4 40 11 22  0 15 14
82  3 57  9 71  7 48
  2 17  7 62  5 75  9 35  1 91 14 50  3  7 10 64 13 75 12 94  0 55  6
72  8 47  4 11 11 90
 11 93  6 57  1 71 12 70  9 93  5 20  3 15 13 77 10 58  0 12  2 67  8
68 14  7  7 29  4 52
 13 76  3 27  4 26  9 36 11  8 10 36  0 95  8 48  2 82  6 87  5  6  1
63  7 56 12 36 14 15
```

la38 (15 x 15)

```
  1 26 12 67  0 72  6 74 14 13  8 43  4 30  3 19 10 23 11 85  5 98 13
43  2 38  7  8  9 75
```

```
 14 42  0 39  4 55 12 46  1 19  8 93  9 80  5 26 10  7  6 50 11 57  3
73  2  9  7 61 13 72
  3 96  4 99 12 34  6 60  7 43 14  7 13 12  8 11 11 70 10 43  0 91  1
68  9 11  5 68  2 72
 14 63 11 45  4 49  1 74  8 27  0 30  9 72  7  9 12 99 13 60  5 69  6
69  2 84  3 40 10 59
  2 91  0 75  9 98  3 17 10 72 13 31 11  9 14 98  7 50  5 37  4  8  8
65  1 90 12 91  6 71
 11 35  6 80  4 39  3 62 14 74  5 72 10 35  9 25  1 49  8 52  7 63  2
90 13 21 12 47  0 38
 14 19  7 57 10 24 13 91  3 50  0  5 11 49 12 18  9 58  5 24  8 52  1
88  2 68  6 20  4 53
  7 77 14 72  5 35 11 90  4 68  6 18  3  9  0 33  8 60 10 18 12 10 13
60  1 38  2 99  9 15
 13  6  8 86  2 40  9 79 12 92 11 23  5 89 10 95  6 91  7 72  0 80  1
60  3 56  4 51 14 23
  1 46  6 28  5 34 11 77  4 47  0 10 14 49  8 77 10 48  7 24 12  8  2
72 13 55  9 29  3 40
 10 22  4 89 12 79  0  7  9 15  1  6 11 30  6 38  5 11  8 52  3 20  7
5 14  9  2 20 13 28
  5 73 14 56  2 37  3 22 13 25  6 58  1  8  7 93  4 88  8 17 12  9 11
69 10 71  9 85  0 55
  9 85 14 58  3 46  8 64  2 49  6 37  1 33  4 30  5 26  0 20 13 74 10
77 12 99 11 56  7 21
 10 17  3 24  4 89  5 15 11 60  1 42  8 98  2 64 13 92  0 63  7 52 12
54  6 75 14 23  9 38
  3  8  5 17 11 56  7 93 14 26  9 62  6  7 10 88  0 97  1  7  2 43  8
29 13 35 12 87  4 57
```

la39 (15 x 15)

```
10 51 14 43  7 80  4 18  6 38  3 24  2 67 12 15 11 24 13 72  8 45  5
80  9 64  1 44  0 88
  6 40  9 88 10 77  5 59 11 20  3 52  8 70  0 40  4 32 13 76 12 43  7
31  2 21 14  5  1 47
  0 32  3 49 10  5  5 64  7 58  8 80  6 94 11 11  1 26 13 26 14 59  9
85  4 47 12 96  2 14
  5 23  6  9  0 75 12 37 11 43  2 79  4 75  3 34  7 20 13 10 14 83 10
68  9 52  8 66  1  9
 12 69  9 59  3 28 14 62 13 36  1 26  6 84 11 16  8 54  5 42  2 54  0
6 10 40  7 88  4 79
 13 78 12 53 11 17  5 29  4 82  2 23  9 12  8 64  1 86  7 59  6  5  3
68 14 59 10 13  0 56
 10 83 13 46  9  7 12 65 11 69  6 62  0 16  2 58  8 66  5 83  7 90 14
42  4 81  3 69  1 85
  7 73 10 71  8 64  6 10  9 20 11 99  4 24 14 65  5 82  3 72 12 43  1
82 13 27  2 24  0 33
  4 82  1 34  3 92  2  8  0 38  8 45  6 21  5 35 12 52  9 35 11 15 14
23 10  6 13 83  7 30
  2 84  5  7  9 66 10  6  4 28 13 27  6 79  7 70  0 85  1 94  3 60 14
80 12 39  8 66 11 29
  3 44  6 58 13 14  8 65  1 72  5 14 12 52  4 21  9 25  0  5 11 51  7
61 14 55 10 42  2 36
 14 43 10 72  5 78 11 12 12 17  0 46  9 27  6 51  2 63  1 79  8 79  7
91  4 49 13 26  3 93
  7 49  0 49  4 71  5 78  9 44 10 41 12 91 13 84  8 91  6 21 11 47 14
28  3 61  2 70  1 93
  3 25  4 85  0 66  2 45 10 95 12 21  8 84  5 24  9 53  7 67  6 91 11
11 13 32  1 30 14 89
```

```
  3 92  7 93  0 99  1 40 10 37 12 69  5 66  6 57 14 22  9 44  8 73 13
97 11 18  2 69  4 41
```

la40 (15 x 15)

```
 9 65 10 28  4 74 12 33  2 51 14 75  5 73  8 32  6 13  3 81  1 35  7
59 13 38 11 55  0 27
 0 64  1 53 11 83  2 33  4  6  9 52 14 72  8  7 13 90 12 21  6 23  3
10 10 39  5 49  7 72
14 73  3 82  1 23 12 62  6 88  5 21  8 65 11 70  7 53 10 81  2 93 13
77  0 61  9 28  4 78
 1 12  6 51  7 33  4 15 14 72 10 98  9 94  5 12 11 42  2 24 13 15  8
28  3  6 12 99  0 41
12 97  5  7  9 96  4 15 14 73 13 43  0 32  8 22 11 42  1 94  2 23  7
86  6 78 10 24  3 31
 1 72  5 88  2 93 13 13  4 44 14 66  6 63  7 14  9 67 10 17 11 85  0
35  3 68 12  5  8 49
 9 15  7 82  6 21 14 53  3 72 13 49  2 99  4 26 12 56  8 45  1 68 10
51  0  8  5 27 11 96
 3 54  7 24  4 14  8 38  5 36  2 52 14 55 12 37 11 48  0 93 13 60 10
70  1 23  6 23  9 83
 3 12  8 69  6 26  9 23 14 28  1 82  5 33  4 45 13 64  7 15 11  9 12
73 10 59  2 37  0 62
 0 87  5 12  7 80  4 50 10 48 12 90  1 72 13 24  6 14  8 71 11 44  9
46  2 15 14 61  3 92
 2 54  0 22  6 61  4 46  3 73  5 16 12  6  9 94 14 93 13 67  8 54  7
75 11 32 10 40  1 97
10 92 14 36  4 22  9  9  3 47  1 77 12 79 13 36  6 30  8 98 11 79  7
7  5 55  2  6  0 30
 0 49 13 83  3 73  6 82  1 82 14 92 11 73  4 31 10 35  9 54  5  7  8
37  7 72  2 52 12 76
10 98 12 34 13 52  4 26  1 28  3 39  8 80  5 29  9 70  0 43  6 48  7
58  2 45 14 94 11 96
 1 70 10 17  6 90 12 67  4 14  8 23  3 21  7 18 13 43 11 84  5 26  9
36  2 93 14 84  0 42
```

- 20 problems due to Storer et al. [1992], referred as swv01 – swv20

swv01 (20 x 10)

```
 3  19  2  27  1  39  4  13  0  25  8  37  9  40  5  54  7  74  6  93
 2  69  0  30  4   1  3   4  1  64  7  71  5   2  9  84  6  31  8   8
 4  79  3  80  0  86  2  55  1  54  8  81  6  72  7  86  5  59  9  75
 2  76  3  15  1  26  0  17  4  30  8  44  7  91  6  83  5  52  9  68
 4  73  3  87  1  74  0  39  2  98  9 100  5  43  8  17  7   7  6  77
 1  63  0  49  2  16  3  55  4   9  9  73  5  61  8  34  6  82  7  46
 0  87  1  71  4  43  3  80  2  39  7  70  8  18  6  41  9  79  5  44
 4  70  2  22  0  73  3  62  1  64  5  25  8  19  6  69  9  41  7  28
 3  16  0  84  1  58  4   7  2   9  5   8  6  10  7  17  8  42  9  65
 3   8  0  10  1   3  4  41  2   3  7  40  8  56  5  53  9  96  6  13
 4  62  1  60  3  64  2  12  0  39  5   2  7  64  6  87  9  21  8  60
 2  66  1  71  3  23  4  75  0  78  7  74  6  35  9  24  8  23  5  50
 1   5  3  92  4   6  0  69  2  80  7  13  5  17  9  89  6  80  8  47
 0  82  3  84  1  24  2  47  4  93  7  85  5  34  6  73  8  28  9  91
 4  55  0  57  3  63  2  24  1  40  7  30  6  37  5  99  8  88  9  41
 1  75  2  47  3  68  0   7  4  78  7  80  6   2  9  23  8  49  5  50
```

0	91	4	25	2	10	1	21	3	94	8	6	7	59	5	84	9	75	6	7
2	85	1	31	0	94	4	94	3	11	5	21	9	7	6	61	8	50	7	93
1	27	0	77	4	13	2	30	3	2	5	88	7	4	9	39	6	53	8	54
1	34	2	12	3	31	0	24	4	24	7	16	5	6	9	88	8	81	6	11

swv02 (20 x 10)

2	16	1	58	0	22	4	24	3	53	8	9	9	57	7	63	5	92	6	43
3	6	1	48	4	14	0	66	2	24	7	2	9	85	6	73	8	19	5	99
4	100	2	90	0	63	1	14	3	31	5	27	9	15	8	1	6	51	7	33
2	98	3	84	4	52	0	12	1	96	9	60	6	74	8	93	5	45	7	49
4	39	0	54	2	28	3	8	1	30	8	57	6	75	5	9	7	41	9	19
3	94	0	8	2	89	1	13	4	37	8	36	6	63	9	24	5	71	7	97
3	90	2	69	1	25	4	15	0	65	7	52	6	56	9	91	8	83	5	86
3	59	1	99	4	41	0	68	2	14	7	4	9	55	6	48	8	13	5	15
4	36	2	17	1	51	0	16	3	54	8	45	5	50	7	98	6	68	9	82
1	75	0	11	4	55	2	93	3	51	6	61	9	40	7	19	8	24	5	55
4	56	0	73	3	59	2	38	1	51	6	99	8	29	9	53	5	7	7	72
3	68	4	50	1	88	2	88	0	33	5	47	8	52	6	26	9	74	7	68
2	3	3	42	0	45	1	57	4	28	5	14	8	22	9	31	6	44	7	38
3	89	0	73	4	12	1	9	2	49	5	11	8	15	7	41	9	37	6	10
3	76	2	97	4	100	1	92	0	25	5	8	9	92	7	51	6	58	8	65
4	50	0	54	3	85	1	47	2	45	6	99	9	39	5	32	8	87	7	56
0	70	2	58	3	33	1	85	4	25	8	5	7	65	9	20	6	52	5	44
1	22	3	45	4	60	0	66	2	5	7	61	6	73	9	60	5	14	8	44
4	64	0	97	2	31	1	4	3	43	9	47	7	93	6	100	5	10	8	51
3	9	4	87	2	34	0	62	1	56	5	66	8	95	7	56	9	42	6	86

swv03 (20 x 10)

2	19	0	30	1	68	4	55	3	24	8	34	7	72	5	32	9	62	6	45
2	63	1	11	4	65	3	16	0	67	9	95	8	23	7	82	6	52	5	53
2	19	4	17	1	79	3	49	0	12	7	41	9	67	8	40	6	25	5	42
0	42	2	71	3	27	4	95	1	19	5	48	8	100	6	31	7	25	9	38
3	1	1	100	4	68	0	94	2	89	5	86	7	35	9	29	8	56	6	55
4	93	1	53	2	4	3	48	0	57	8	99	7	67	5	86	6	80	9	60
4	82	1	95	2	12	0	60	3	80	8	88	7	5	6	81	9	52	5	69
3	79	1	31	4	63	0	28	2	64	8	63	5	29	7	75	9	18	6	33
4	9	1	64	2	31	0	13	3	33	9	82	6	79	5	30	7	84	8	20
2	14	0	56	1	95	4	34	3	13	6	16	5	44	7	45	8	62	9	86
4	66	3	9	2	66	1	46	0	12	5	10	7	58	6	6	8	62	9	17
4	89	1	52	2	37	3	74	0	7	8	43	5	96	7	89	6	21	9	66
1	73	3	68	2	5	4	49	0	67	9	23	7	7	5	44	8	30	6	29
2	21	0	68	1	88	4	75	3	64	6	6	8	72	7	66	9	66	5	56
1	24	4	25	2	69	0	27	3	51	9	60	8	26	6	45	5	77	7	93
2	19	3	17	1	82	4	75	0	34	5	67	9	89	6	91	7	13	8	35
4	2	0	21	3	83	1	19	2	65	6	65	8	8	9	68	7	60	5	7
1	63	3	49	2	4	4	2	0	50	9	99	5	27	6	68	8	46	7	89
0	48	4	45	3	100	2	66	1	30	6	58	7	73	9	94	5	36	8	5
2	36	0	53	4	56	3	57	1	77	9	7	6	59	8	8	5	15	7	23

swv04 (20 x 10)

2	16	0	59	4	10	3	95	1	64	8	92	9	56	7	3	5	73	6	17
1	5	4	64	3	30	2	14	0	96	9	11	8	73	7	35	6	93	5	12
3	35	4	75	0	54	1	30	2	83	9	20	8	29	7	38	6	90	5	39

```
 4  29   3  21   0  52   2  93   1  20   5   5   7  11   8  53   9  56   6  98
 0  17   3  16   4  41   1  78   2 100   5  55   8  27   6   2   7  87   9  55
 3  97   1  32   4  84   2  71   0  38   9  64   7  16   5   5   6  41   8  41
 3  41   1  57   4  37   0  64   2  92   6  19   9  47   7  94   8  79   5  21
 0  23   3  67   1  39   4  98   2  63   8  83   5  45   6  89   9  81   7  44
 1  88   0  59   3  39   2  63   4  91   8  36   5  44   6  45   9  43   7  12
 2  29   1  17   0   6   3  74   4  51   9  14   6   2   5  56   7  49   8  14
 3  75   2  10   4   1   0  35   1  99   7  56   5  95   9  78   6  53   8  82
 0  75   2  96   1  21   3  90   4  55   6  23   7  40   9  76   8  55   5  45
 3  90   4  64   0  72   2  33   1  59   7  51   6  74   5  85   9  76   8  38
 3  57   1  84   2  87   4   2   0  68   8   4   5  77   6  37   7  37   9  94
 1  16   3  46   4  34   2  23   0  77   7  68   8  14   9  54   5  37   6  99
 4  24   1  73   2  92   0  43   3  42   5  81   7  99   8  88   9  80   6   5
 1  56   2  51   0   3   4  87   3  25   5  62   7  11   8  88   6  68   9  29
 2  85   3   3   4  21   0  49   1  79   8  38   5  37   9  72   7  18   6  18
 0   2   3  55   1  31   2  29   4  98   5  92   6  43   8  99   7  67   9  41
 4  69   3  64   0  61   1  13   2  31   5   6   8  84   9  94   7  32   6  54
```

swv05 (20 x 10)

```
 2  19   1  30   3  80   0  84   4  14   8  51   5  73   6  91   7  81   9  71
 2  74   4  79   1  39   0   7   3  66   9   6   5  93   8  76   6  21   7  76
 4  90   3  33   1  38   2  73   0  61   8  61   7  76   5  86   9  28   6  35
 4   1   3  22   2   1   0  77   1  33   6  98   5   4   9  27   8   8   7  68
 2  63   4   5   1  95   0   7   3  50   8  46   9  28   6  70   5  60   7  34
 0  98   1  73   4  15   3  21   2  32   7  24   9   9   8  24   5   7   6  34
 3  51   4  47   2  30   1  16   0  51   5  41   6  79   7  79   9   3   8  72
 4   3   1  59   0  53   3  20   2  19   6  20   9  16   7  90   5  96   8  18
 1  34   2  55   3  97   0  93   4  90   7  81   5  63   8  41   6   1   9  51
 4  77   3  87   1  92   2  83   0  45   7  75   9  60   6  75   5  93   8  33
 0  31   2  66   1  58   4  17   3  94   5  63   7  80   9  61   6  78   8  52
 4  70   1  25   2  75   0  89   3  41   7 100   5  73   6  28   8  94   9  88
 1  67   4  62   3  12   2  55   0  62   5  58   8  66   7  73   6  55   9   1
 4  81   0  37   1   2   3  39   2  17   7  74   6  71   8  61   5  42   9   5
 3  62   0  31   4  63   2  31   1   5   9   7   7  77   8  34   6  34   5   3
 0   5   2  55   3  62   1  82   4  80   6   6   8   7   7  29   5  80   9  89
 3  26   1  50   2  58   0  22   4  68   7  12   6   9   9  34   5  90   8  87
 0  50   2  28   1  64   4  34   3  63   7   9   9  48   6  63   8  61   5   2
 0  47   2  23   1  23   4  82   3  98   7  66   6  78   8 100   9  79   5  32
 1  13   4  14   0  90   2  77   3  80   9  30   7  31   5  36   6  51   8  69
```

swv06 (20 x 15)

```
  1  16   6  58   2  22   4  24   5  53   3   9   0  57  10  63   8  92  12
43   7  41  13  26  14  20   9  44  11  93
  2  89   1  94   0  86   3  13   6  54   4  41   5  55   7  98  13  38  14
80   9   1  11 100  12  90  10  63   8  14
  1  26   6  96   3  32   4  75   5   9   0  57   2  39  12  54  14  28  10
8  11  30  13  57   9  75   7   9   8  41
  3  37   2  36   5  63   0  24   6  71   1  97   4  74  14  19  12  45   8
24  11  71  13  53  10  61   9   6   7  32
  3  57   0  55   1  21   5  84   2  23   6  79   4  90  11   8  14  59  10
99   9  41  12  68   8  14  13   4   7  55
  4  10   2  81   1  13   3  78   0  78   5  10   6  48   9  37  11  21   7
88  12  75  14  11  13  55  10  93   8  51
  6 100   2  52   3  54   1  37   5  26   4  74   0  87   8  13  12  88  10
94  14  73   7  55  11  68   9  50  13  88
```

```
  4  47   5  70   6   7   2  72   0  62   3  30   1  95  10  18   9  65   7
69 13  89   8  89 14  64 12  81 11  25
  6   1   1  10   0  72   3  59   4  92   5  53   2  89  14  52   7  48   8
8 13  69 10  49   9  26 12  76 11  97
  6  85   2  47   4  45   1  99   0  39   5  32   3  87  10  56   8  98  11
13   7  96 12  71 14  95   9  11 13  78
  0  17   2  21   3  87   6  41   5  41   4  31   1  96   8  17  11  95  13
29 14   3 10  71   7  64   9  97 12  31
  6   9   0  87   4  34   1  62   3  56   5  66   2  95   9  56  14  42   8
86   7  68 12  82 10  82 13  52 11  97
  3  86   1  37   2  49   0   2   6  30   5  63   4   4  14  47   8  84  10
5 13  13   9  39 12  18   7  76 11  63
  0  29   6  34   1  53   3   7   5  19   4  26   2  63  12  22  10  98  13
77 14  11   7  87   9   5 11  44   8  42
  6  44   4  91   1  91   2  58   0  77   3  51   5  14  13   1   9  17   7
55 12  40   8  95 14  31 11  54 10  37
  5  59   4  47   1  56   6  39   2   7   0  43   3  39  13  75  10  43  12
32   9   6 11  93   7  69   8  47 14  93
  4  24   1  30   3  97   6  17   0   7   2  55   5   8   7  70  10  87   8
29 12  20 13  29 11  51   9  14 14  32
  2  29   4  99   3  17   0  96   1  50   5  67   6  91  10  91  13  14  12
14   7  19   8  36 11  11 14  83   9   6
  0   7   6  60   3  31   5  76   1  23   2  83   4  30   8  73  14  76  11
17 10  53 13   9 12  72   7  89   9  24
  3  63   0  89   2   2   1  46   6  86   5  74   4   1   7  34   9  30  12
19 13  48 11  75   8  72 14  47 10  58
```

swv07 (20 x 15)

```
  3  92   1  49   2  93   6  48   0   1   4  52   5  57   8  16  12   6  13
6 11  19   9  96   7  27 14  76 10  60
  5   4   3  96   6  52   1  87   2  94   4  83   0   9  11  85  10  47   8
63   9  31 13  26 12  46   7  49 14  48
  1  34   6  34   4  37   2  82   0  25   5  43   3  11   9  71  14  55   7
34 11  77 12  20   8  89 10  23 13  32
  3  49   5  12   6  52   2  76   0  64   1  51   4  84  10  42  12   5   7
45   8  20 11  93 14  48 13  75   9 100
  2  35   1   1   3  15   6  49   5  78   4  80   0  99   9  88   7  24  11
20 10 100   8  28 14  71 13   1 12   7
  3  69   6  24   5  21   4   3   1  28   2   8   0  42  10  33  11  40   9
50   8   8 13   5 12  13   7  42 14  73
  0  83   4  15   2  62   6  27   5   5   1  65   3 100  14  65  10  82   7
89 13  81   9  92   8  38 11  47 12  96
  6  98   4  24   2  75   0  57   1  93   3  74   5  10   7  44  13  59  11
51 12  82 14  65 10   8   8  12   9  24
  4  55   0  44   3  47   5  75   2  81   6  30   1  42  10 100   8  81   7
29 13  31   9  47 11  34 12  77 14  92
  2  18   5  42   0  37   4   1   3  67   6  20   1  91   8  21  14  57  12
100 10 100 11  59 13  77   9  21   7  98
  3  42   1  16   4  19   6  70   2   7   0  74   5   7  12  50   9  74   8
46 14  88 13  71 10  42   7  34 11  60
  6  12   4  45   2   7   0  15   1  22   3  31   5  70  13  88   9  46   8
44 14  45 12  87 11   5   7  99 10  70
  4  51   5  39   0  50   2   9   3  23   6  28   1  49  13   5  12  17  14
40 10  30 11  62   8  65   7  84   9  12
  6  92   0  67   5  85   1  88   3  18   4  13   2  70   7  69  14  10  13
52   8  42 11  82 10  19 12  21   9   5
  4  34   0  60   1  52   5  70   2  51   6   2   3  43  10  75  11  45   8
53 12  96 13   1 14  44   7  66   9  19
```

```
  6  31   1  44   0  84   3  16   4  10   2   4   5  48  13  67  14  11  12
21   8  78   7  42  11  44   9  37  10  35
  1  20   4  40   3  37   2  68   6  42   0  11   5   6  10  44  11  43  12
17  14   3   7  77  13 100   9  82   8   5
  5  14   0   5   3  40   1  70   4  63   2  59   6  42   9  74  13  32   7
50  10  21  14  29  12  83  11  64   8  45
  6  70   0  28   3  79   4  25   5  98   2  24   1  54  12  65  13  93  10
74   7  22   9  73  11  75   8  69  14   9
  5 100   2  46   4  69   3  41   1   3   6  18   0  41   8  94  11  97  12
30  14  96   7   7   9  86  13  83  10  90
```

swv08 (20 x 15)

```
  3   8   4  73   2  49   5  24   6  81   1  68   0  23  12  69   8  74  10
45  11   4  14  59   9  25   7  70  13  68
  3  34   2  33   5   7   1  69   4  54   6  18   0  38   8  28  12  12  14
50  10  66   7  81   9  81  13  91  11  66
  0   8   6  20   3  52   4  83   5  18   2  82   1  68   7  50  14  54  11
6  10  73  13  48   9  20   8  93  12  99
  2  41   0  72   1  91   4  52   5  30   3   1   6  92  13  52   8  41   9
45  14  43  12  97  10  64  11  71   7  76
  0  48   1  44   5  49   6  92   3  29   2  29   4  88  14  14  10  99   8
22  13  79   9  93  12  69  11  63   7  68
  0  56   6  42   2  42   3  93   1  80   4  54   5  94  12  80  14  69  11
39   8  85  10  95  13  12   9  28   7  64
  0  90   4  75   6   9   1  46   2  91   3  93   5  93  14  77   9  63  11
50  12  82  13  74   8  67   7  72  10  76
  0  55   2  90   6  11   3  60   4  75   1  23   5  74  11  54   7  97  12
32  13  67  10  15  14  48   8 100   9  55
  6  71   5  64   2  40   0  32   3  92   1  59   4  69  13  68  14  34  12
71   8  28   9  94   7  82  10   1  11  58
  6  36   4  46   1  50   5  87   3  33   2  94   0   3  14  60  11  45  13
84   9   1   8  38  10  22  12  39   7  50
  1  53   0  34   5  56   6  97   3  95   4  32   2  28  14  48   7  54  12
98   8  84   9  77  10  46  13  65  11  94
  2   1   5  97   0  77   4  82   6  14   1  18   3  74  14  52  11  14  12
93   9  35   8  34  13  84  10   6   7  81
  1  62   0  86   2  57   6  80   5  37   3  94   4  77   7  72   9  26  11
41  10   7   8  56  13  98  14  67  12  47
  5  45   3  30   0  57   6  68   1  61   2  34   4   2   7  57  13  96   9
10  12  85  14  42  10  93   8  89  11  43
  6  49   4  53   1  51   2   4   0  17   5  21   3  31  10  45  13  45   9
63  11  21   8   4   7  23  14  90  12   1
  6  68   5  18   0  87   3   6   4  13   2   9   1  40   8  83   7  95  12
27  10  94  14  68  11  22  13  28   9  66
  2  80   6  14   0  67   5  15   1  14   3  97   4  23   8  45  10   1  11
5  14  87   7  34  12  12   9  98  13  35
  4  33   2  20   3  74   6  20   5   3   0  90   1  37  13  56  12  38   8
7  14  84   9 100  11  41  10   6   7  97
  6  47   4  63   3   1   0  28   2  99   1  41   5  45  14  60  13   2   7
25   8  59   9  39  10  76  11  89  12   5
  6  67   2  46   3  25   1   2   5  22   4   8   0  22  13  64   7  82  12
99  11  79  10  87   8  71   9  24  14  19
```

swv09 (20 x 15)

```
  5   8   3  73   0  69   2  38   6   6   4  62   1  78   9  79   8  59  13
```

```
77 11  22 10  80 12  58 14  49  7  48
  3  34  4  29  2  69  0   5  5  63  1  82  6  94 14  17 11  94  9
29 10   5 13  75  7  15  8  61 12  61
  1  52  2  30  0  25  6  17  3  46  4  86  5   3 14  70 11  34  9
23 10  68 13  76  8  53 12  71  7   9
  2  50  4  20  3  24  0  53  1  97  5  79  6  92 14   3 12  52 10
75  9  74  8  59  7  75 13  84 11  99
  2  15  0  61  3  47  4  38  6  49  5  21  1   6 11   8  8  71 14
83 13  24 12  18  9  33  7  70 10 100
  4  48  5  50  2  66  0  92  6   2  3  58  1  23  9  84  8  66 10
12  7  36 14   4 12  88 13  64 11  12
  3  29  0  25  6  44  5  87  2  42  1  44  4  86  8  28 10  86  9
74 14  77 13  59 12  94  7  58 11  16
  4  31  3  58  0  94  5  69  2  44  1  93  6  92  9  80  8  63 12
47 13   3  7  79 11  39 14  80 10  75
  1  69  2  27  0  76  5  19  6  86  3  16  4  31 12  33  9  69 13
19 10  43 14   9 11  37  7  35  8  24
  2  75  3  78  6  41  4  60  5  59  0  42  1  60 12  18  8  31 10
15  7  54 14  60  9  20 11  61 13  69
  4  89  6  20  1  27  5  78  3   2  2  21  0  55 13  79 11  77 10
99  9  70 12  30  7  97  8  41 14  98
  6   1  2  10  4  84  5  72  0  14  1   9  3  51  7  22 14  65 10
100 13  65 11  43  8  10 12  14  9  19
  5  50  2  13  3  49  6  75  1  42  0  81  4  89  9 100 14  54 13
37 10   7 11  38  8  25 12  78  7  79
  2  44  3  77  5  26  1  42  4   9  6  73  0  60  9  61 10  85 12
14 11  92  7 100 14  49  8  46 13  12
  2  72  0  53  1  43  5  65  6  59  4  87  3  13  8  71 12  25  9
71 10  89 11   2  7  76 14  21 13  12
  2  60  6  28  5  33  1  36  0   6  3  96  4  48  9  40 11  79 10
60  8  39 13  34  7  54 12  20 14  52
  5  82  2  12  3  11  4  61  1  21  0  21  6  34 12  86 14  53  8
7  9   4  7  95 10  62 13  54 11  82
  5  72  0  13  3  46  6  97  1  87  4  87  2  11  7  45 14  85 11
66  8  43  9  39 13  34 10  30 12  55
  1  39  5  19  0  19  4  73  6  63  3  30  2  69  9  36  7  13 10
96 12  27 13  59 14  76 11  62  8  14
  1   7  4  14  3  79  2  27  6  43  0  96  5  24 11  30  7  27 12
2  8  69 14  75 13  34 10  79  9  96
```

swv10 (20 x 15)

```
  3   8  2  73  1  79  0  95  6  69  4   9  5   5  8  85  9  52 11
43 14  32  7  91 10  24 13  89 12  38
  6  45  1  70  4  84  3  24  5  18  0  20  2  71  8  21  7  60  9
98 10  70 13  52 12  34 11  23 14  52
  6  16  4  68  1  85  0  39  5  40  2  98  3  61 10  77  7  60 11
73  9  66 14  84  8  16 13  43 12  88
  0  72  1  17  3  68  4  89  2  94  6  98  5  56 10  88 13  27  9
60 12  61  8   8  7  88 11  48 14  65
  6  78  2  24  5  28  0  73  4  21  1  69  3  52 14  32  8  83 11
48 10  29 13  48 12  92  9  43  7  82
  4  54  6  31  5  14  3  47  0  82  1  75  2   4  8  31 12  72  7
58  9  45 13  91 14  31 11  61 10  27
  4  88  1  28  5  92  6  62  3  93  0  14  2  65  7  33  9  44  8
31 14  32 11  72 13  47 12  61 10  34
  0  52  1  59  5  98  3   6  2  19  6  53  4  39  8  74 12  48 10
33 13  49 11  92  7  22 14  41  9  37
  0   2  6  85  3  34  2  51  4  97  5  95  1  73 14  61  9  28 12
```

```
73  8  21 11  85  7  75 13  42 10   7
 5  94  1  28  0  77  2  56  6  79  4   2  3  82  9  88 10  93 12
44 14   5  8  96  7  34 13  56 11  41
 2  15  5  88  6  18  3  14  1  82  0  58  4  33 13  19 10  42  9
36 14  57 12  85  7   3 11  62  8  36
 3  30  6  33  0  13  4   4  2  74  1  37  5  78 14   2 13  56  9
21 10  61 11  81  7  18  8  59 12  62
 5  40  1  75  6  45  0  41  3  97  2  65  4  92  7  11 12  44  8
40  9 100 11  91 14  66 13  53 10  27
 1  83  2  52  0  84  3  66  5   3  6   5  4  71 13  41 10  42 11
63 12  50 14  43  8   3  9  35  7  18
 4  44  0  26  1  59  6  81  2  84  5  81  3  91 13  41  7  42 11
53  8  63 14  89  9  15 10  64 12  40
 1  46  0  97  5  67  4  97  3  71  6  88  2  69 14  44 12  20 11
52 13  34 10  74  8  79  7  10  9  87
 3  71  6  13  4 100  2  67  1  57  5  24  0  36  7  88 14  79  8
21  9  86 12  60 11  28 10  14 13   3
 0  97  6  24  2  41  4  40  1  51  5  73  3  19  9  27 12  70 13
98 10  11 11  83  7  76  8  60 14  12
 5  88  3  48  1  33  4  96  6  10  0  49  2  52 10  38 13  49  7
31 12  94 14  23  9   7 11   5  8   4
 2  85  0 100  5  51  6  91  1  21  3  83  4  30 12  23  9  48  8
19 11  47 10  95  7  23 14  78 13  22
```

swv11 (50 x 10)

0	92	4	47	3	56	2	91	1	49	5	39	9	63	7	12	6	1	8	37
0	86	2	100	1	75	3	92	4	90	5	11	7	85	8	54	9	100	6	38
1	4	4	94	3	44	2	40	0	92	8	53	6	40	9	5	5	68	7	27
4	87	0	48	1	59	2	92	3	35	6	99	7	46	9	27	8	83	5	91
0	83	1	78	4	76	3	64	2	44	8	12	9	91	6	31	7	98	5	63
3	49	0	15	1	100	4	18	2	24	6	92	9	65	5	26	7	29	8	24
0	28	3	53	4	84	2	47	1	85	7	100	5	34	6	35	8	90	9	88
2	61	4	71	3	54	1	34	0	13	9	47	8	2	6	97	7	27	5	97
0	85	2	75	1	33	4	72	3	49	7	23	5	12	8	90	6	87	9	42
2	24	3	20	1	65	4	33	0	75	9	47	6	84	8	44	7	74	5	29
2	48	3	27	4	1	0	23	1	66	6	35	7	46	9	29	5	63	8	44
2	79	0	4	4	61	3	46	1	69	7	10	8	88	9	19	6	50	5	34
0	16	4	31	3	77	2	3	1	25	8	88	7	97	9	49	6	79	5	22
1	40	0	39	4	15	2	93	3	48	6	63	9	74	8	46	7	91	5	51
4	48	0	93	2	8	3	50	1	5	6	48	7	46	9	35	5	88	8	97
3	70	1	8	2	65	0	32	4	84	8	9	6	43	7	10	5	72	9	60
0	21	2	28	1	26	3	91	4	58	9	90	6	43	8	64	5	39	7	93
1	50	2	60	0	51	4	90	3	93	7	20	9	33	8	27	6	12	5	89
1	21	3	3	2	47	4	34	0	53	9	67	8	8	5	68	7	1	6	71
3	57	4	26	2	36	0	48	1	11	9	44	7	25	5	30	8	92	6	57
1	20	0	20	4	6	3	74	2	48	9	77	8	15	5	80	7	27	6	10
3	71	1	40	0	86	2	23	4	29	7	99	8	56	6	100	9	77	5	28
4	83	0	61	3	27	1	86	2	99	7	31	5	60	8	40	9	84	6	26
4	68	1	94	3	46	2	60	0	33	7	46	5	86	9	63	6	70	8	89
4	33	1	13	2	91	3	27	0	38	8	82	7	31	6	23	9	27	5	87
4	58	3	30	0	24	2	12	1	38	8	2	9	37	5	59	6	37	7	36
2	62	1	47	4	5	3	39	0	75	7	60	9	65	8	61	6	77	5	31
4	100	0	21	1	53	3	74	2	3	8	34	6	6	7	91	9	80	5	28
1	8	0	3	2	88	3	54	4	18	9	4	6	34	5	54	8	59	7	42
3	33	4	72	0	83	2	17	1	23	6	24	8	60	9	96	7	78	5	70
4	63	2	36	3	70	0	97	1	99	6	71	9	92	5	41	8	73	7	97
2	28	1	37	4	24	0	30	3	55	8	38	5	9	9	77	7	17	6	51
3	15	0	46	2	14	4	18	1	99	9	48	6	41	5	10	7	47	8	80

4	89	3	78	2	51	1	63	0	29	7	70	9	7	5	14	8	84	6	32
4	26	1	69	2	92	3	15	0	23	8	42	6	95	5	47	9	83	7	56
1	38	2	44	3	47	4	23	0	10	9	63	7	65	6	21	5	70	8	56
3	42	4	85	1	29	0	35	2	66	9	46	8	25	5	90	7	85	6	75
3	99	0	46	4	74	2	96	1	48	5	52	6	13	7	88	8	4	9	30
1	15	3	80	4	47	2	25	0	8	9	61	7	70	8	23	6	93	5	5
0	90	2	51	3	66	4	5	1	86	5	59	6	97	9	28	7	85	8	9
0	59	1	50	4	40	3	23	2	93	7	61	9	96	8	63	6	34	5	14
1	62	2	72	4	30	0	21	3	15	5	77	6	13	7	2	8	22	9	22
2	20	4	14	3	85	1	4	0	2	9	33	7	90	5	48	8	90	6	62
0	49	3	49	4	46	1	89	2	64	9	72	8	6	5	83	6	13	7	66
4	74	1	55	2	73	0	25	3	16	7	19	9	38	6	22	5	26	8	63
3	13	2	96	1	8	0	15	4	97	6	95	7	2	5	66	8	57	9	46
4	73	1	97	3	39	0	22	2	90	9	64	6	65	8	31	5	98	7	85
3	43	2	67	0	38	1	77	4	11	7	61	5	7	9	95	8	97	6	69
0	35	2	68	1	5	3	46	4	4	7	51	6	44	5	58	9	69	8	98
2	68	1	81	0	2	3	4	4	59	9	53	8	69	5	69	6	14	7	21

swv12 (50 x 10)

0	92	4	49	1	93	3	48	2	1	7	52	6	57	9	16	5	6	8	6
4	82	3	25	2	69	1	86	0	54	6	15	5	31	9	5	7	6	8	18
0	31	1	26	3	46	2	49	4	48	8	74	7	82	5	47	9	93	6	91
0	34	4	37	1	82	3	25	2	43	6	11	9	71	5	55	7	34	8	77
4	22	0	91	3	54	2	49	1	97	9	2	7	46	5	98	6	27	8	89
2	46	3	70	1	3	0	44	4	24	9	65	6	60	5	94	8	58	7	22
3	53	0	99	1	80	2	74	4	29	6	72	7	54	5	98	8	60	9	69
3	96	1	87	0	36	2	57	4	7	8	36	9	26	5	94	6	47	7	70
3	5	2	47	1	59	0	57	4	28	9	24	8	79	6	19	5	44	7	35
0	96	1	4	3	60	2	43	4	39	7	97	5	2	9	81	6	89	8	91
2	23	4	74	3	98	0	24	1	75	9	57	8	93	6	74	5	10	7	44
3	36	4	5	2	36	0	49	1	90	8	62	5	74	9	4	6	85	7	53
2	44	1	47	3	75	4	81	0	30	7	42	8	100	9	81	6	29	5	31
1	2	0	18	3	88	2	27	4	5	5	36	7	30	6	51	8	51	9	31
1	21	0	57	3	100	2	100	4	59	8	77	7	21	5	98	6	38	9	84
4	97	2	72	1	70	3	99	0	42	6	94	5	59	9	90	8	78	7	13
3	16	2	19	1	70	0	7	4	74	6	7	5	50	9	74	8	46	7	88
3	45	4	91	2	28	0	52	1	12	5	45	6	7	7	15	9	22	8	31
3	56	2	3	1	8	4	25	0	90	8	99	6	22	9	65	7	51	5	31
0	23	3	28	1	49	2	5	4	17	7	40	9	30	5	62	8	65	6	84
2	88	0	86	4	8	1	41	3	12	6	67	9	77	5	94	7	80	8	11
4	81	3	42	0	19	2	100	1	10	5	23	9	71	8	18	6	93	7	36
4	74	2	73	3	63	1	9	0	51	8	39	7	7	6	96	5	81	9	22
1	1	3	44	0	66	4	19	2	65	7	10	6	23	8	26	9	76	5	77
1	54	2	18	4	99	0	79	3	22	5	2	6	42	8	54	7	90	9	28
3	16	4	1	1	28	0	54	2	97	5	71	6	53	8	32	7	26	9	28
0	82	3	5	2	18	4	71	1	50	5	41	7	62	9	89	6	93	8	54
2	63	3	59	0	42	1	74	4	32	5	50	6	21	7	29	8	83	9	64
4	29	2	76	1	6	3	44	0	4	9	81	5	29	7	95	8	66	6	89
3	55	4	84	1	36	0	42	2	64	5	81	8	85	6	76	7	4	9	16
4	100	0	46	1	69	3	41	2	3	6	18	5	41	7	94	8	97	9	30
3	34	4	35	2	18	1	58	0	98	9	78	8	17	5	53	6	85	7	86
4	68	2	89	1	99	0	3	3	92	5	10	6	52	7	30	8	66	9	69
0	21	3	65	4	19	2	14	1	76	9	84	6	45	5	24	8	54	7	73
4	47	0	68	2	87	3	92	1	96	6	29	5	90	8	29	7	39	9	100
2	35	0	60	4	61	1	61	3	72	9	57	8	94	5	77	7	1	6	53
3	85	2	38	0	79	4	43	1	71	6	44	5	87	8	61	7	51	9	37
1	100	2	33	3	94	0	59	4	25	5	88	9	50	6	19	8	4	7	66
2	8	0	85	1	80	4	75	3	1	7	17	9	32	6	60	5	30	8	57

4	25	2	98	1	94	3	49	0	34	9	37	7	80	6	50	8	25	5	72
3	51	4	49	1	53	2	7	0	73	6	96	7	19	9	41	5	55	8	42
0	57	1	86	2	1	4	61	3	66	6	28	5	56	7	68	8	21	9	65
2	98	1	100	0	47	4	28	3	4	7	34	9	55	5	32	6	72	8	66
4	2	0	74	2	20	1	39	3	63	5	88	9	3	7	22	6	8	8	73
2	44	0	1	3	52	1	43	4	4	6	36	9	75	8	58	5	61	7	38
2	21	4	6	3	32	1	74	0	57	5	72	8	10	9	34	6	91	7	94
4	26	0	59	3	53	1	45	2	23	5	55	8	12	7	34	6	98	9	43
2	4	1	53	4	57	3	95	0	6	6	30	8	1	7	92	9	20	5	86
1	98	2	77	3	65	4	51	0	85	7	23	6	79	5	30	8	41	9	17
4	58	2	43	3	14	0	74	1	64	7	37	8	78	6	33	9	42	5	80

swv13 (50 x 10)

4	68	1	39	2	79	0	72	3	65	5	82	7	33	6	82	8	66	9	55
2	14	3	45	0	18	4	72	1	27	7	57	6	90	8	19	9	19	5	50
4	25	1	77	0	64	3	18	2	19	8	27	6	97	9	81	7	65	5	11
3	70	0	29	2	31	1	39	4	62	8	12	9	2	5	91	7	98	6	91
2	90	4	51	3	38	1	27	0	29	6	67	8	95	9	60	7	86	5	64
4	90	0	55	3	69	1	76	2	97	7	94	5	57	8	65	9	80	6	24
1	23	4	13	0	90	3	24	2	41	8	69	7	8	5	81	6	94	9	76
3	19	1	37	0	16	4	4	2	68	6	45	8	79	9	4	7	30	5	33
2	36	0	76	3	97	4	71	1	19	9	87	6	97	8	64	5	84	7	43
2	20	1	77	0	71	3	73	4	47	7	88	5	100	9	16	8	69	6	77
3	55	4	96	0	8	2	61	1	40	8	46	7	29	9	71	5	89	6	59
2	21	0	18	3	37	4	97	1	59	7	79	6	2	5	80	8	85	9	59
4	19	1	83	2	1	0	95	3	48	9	37	7	59	5	56	8	57	6	81
0	8	1	60	4	91	3	85	2	27	9	39	5	31	6	62	7	94	8	12
4	2	3	10	0	17	1	38	2	96	6	21	9	81	8	64	5	76	7	46
2	46	1	4	4	25	3	41	0	11	5	96	9	56	6	10	7	25	8	32
0	21	1	77	4	22	2	72	3	53	9	28	7	23	5	2	8	52	6	83
3	9	4	37	0	2	2	74	1	15	8	26	5	83	6	90	7	51	9	80
3	6	1	7	0	57	2	4	4	56	7	11	5	57	8	12	6	94	9	29
1	40	2	93	3	65	4	66	0	96	9	5	7	32	8	85	5	93	6	94
1	38	2	19	4	22	0	73	3	7	5	63	8	28	6	23	9	11	7	84
1	96	4	10	0	29	3	59	2	94	5	26	7	22	8	52	6	37	9	50
1	38	3	31	2	76	0	8	4	8	6	50	5	95	8	5	9	25	7	62
0	15	2	84	4	100	3	76	1	66	7	56	5	95	8	94	6	56	9	85
3	73	2	38	1	84	0	42	4	37	5	16	7	24	9	59	6	60	8	23
3	43	1	79	0	80	2	44	4	65	5	81	7	7	8	93	6	55	9	34
2	8	4	2	0	12	3	55	1	60	9	91	6	6	5	83	8	31	7	91
0	8	4	46	3	47	2	57	1	47	9	55	8	74	7	98	6	54	5	51
2	56	4	90	1	41	0	35	3	62	7	4	5	15	9	89	6	73	8	66
0	2	4	39	3	44	1	68	2	54	7	7	8	76	9	29	5	90	6	53
2	34	0	94	3	1	1	23	4	45	8	83	7	84	5	49	6	67	9	49
4	4	2	70	1	19	0	19	3	92	5	70	7	33	9	50	8	82	6	48
4	64	2	76	0	70	3	83	1	91	7	98	8	37	5	3	9	75	6	92
3	96	1	17	0	20	4	13	2	28	7	21	9	65	5	87	6	54	8	98
0	68	4	40	3	98	2	90	1	38	7	45	8	21	5	9	9	3	6	47
0	58	4	19	2	16	3	74	1	32	9	32	5	58	6	93	7	1	8	80
0	32	2	99	1	95	3	2	4	8	9	55	6	32	8	26	5	6	7	68
3	7	4	45	2	19	0	97	1	56	7	22	9	72	8	98	5	59	6	20
2	97	4	98	3	43	0	28	1	23	5	3	8	75	9	43	7	58	6	71
3	31	0	88	2	88	1	82	4	65	5	53	9	15	7	68	6	60	8	99
4	4	0	100	2	95	1	11	3	28	5	80	7	25	9	87	6	25	8	9
0	75	3	10	4	59	2	80	1	60	5	75	8	87	6	33	9	10	7	31
0	54	3	6	4	7	1	72	2	49	7	72	8	64	6	32	9	86	5	69
4	15	3	19	1	18	0	84	2	96	9	71	8	64	6	38	5	58	7	62
1	32	4	80	2	83	3	83	0	50	5	81	7	82	9	33	8	10	6	55

0	65	4	95	3	84	2	64	1	18	9	27	6	70	7	74	5	87	8	68
1	50	2	49	0	96	3	1	4	89	8	42	5	88	9	91	6	64	7	3
3	44	0	91	1	5	2	100	4	77	6	20	5	13	7	25	9	71	8	71
0	86	4	91	1	19	2	69	3	71	5	13	8	87	6	98	9	43	7	13
4	8	0	60	3	31	2	93	1	8	9	1	7	19	6	8	5	85	8	24

swv14 (50 x 10)

4	69	0	37	3	64	1	1	2	65	9	34	5	67	8	43	7	72	6	79
1	11	0	7	3	68	4	43	2	52	6	29	9	71	7	81	8	12	5	36
4	90	3	29	1	1	2	1	0	14	8	38	5	13	9	21	7	41	6	97
1	46	0	26	4	83	2	36	3	20	9	4	8	23	7	65	5	56	6	42
4	46	0	39	2	92	3	53	1	62	9	68	7	65	8	74	6	87	5	46
4	13	1	44	3	43	2	67	0	75	6	5	9	94	5	95	7	28	8	85
1	1	2	99	4	36	3	86	0	65	8	32	5	17	7	71	6	15	9	61
2	18	4	63	3	15	0	59	1	33	7	95	5	63	6	85	8	34	9	3
4	13	2	25	0	82	3	23	1	26	7	22	9	35	8	16	6	24	5	41
3	1	1	7	0	21	2	73	4	39	6	32	7	77	5	29	8	89	9	21
1	53	3	27	4	55	0	16	2	64	5	78	9	32	8	60	7	20	6	20
1	71	2	54	3	21	0	20	4	23	9	40	5	99	7	61	6	94	8	71
2	76	4	72	3	91	0	75	1	7	6	53	8	32	7	71	5	63	9	53
2	12	1	3	4	35	0	64	3	30	5	94	8	67	7	31	6	79	9	14
4	63	1	28	3	87	0	89	2	52	8	2	9	21	7	92	6	44	5	37
0	79	1	65	4	35	3	78	2	17	8	90	5	54	9	91	7	57	6	23
3	20	1	93	4	61	0	76	2	23	5	10	8	34	7	20	9	87	6	77
0	37	2	17	1	92	4	30	3	59	5	47	8	7	7	45	6	13	9	60
4	90	3	74	0	46	2	36	1	2	6	9	5	83	8	90	7	88	9	39
3	83	0	85	2	20	4	88	1	94	6	14	5	16	7	62	9	53	8	9
0	4	4	16	2	64	1	60	3	79	5	37	6	49	7	67	9	95	8	5
3	32	0	86	1	5	4	66	2	77	7	15	5	68	9	40	8	1	6	4
0	2	1	48	4	23	3	25	2	58	9	55	7	14	8	21	6	85	5	27
1	71	4	92	3	99	2	56	0	81	7	79	6	66	9	42	8	47	5	43
1	77	4	85	3	72	2	19	0	71	5	34	7	9	9	14	6	62	8	58
4	38	0	3	2	61	3	98	1	76	5	14	9	56	8	26	7	43	6	44
1	68	4	54	0	62	2	93	3	22	6	57	7	79	9	19	5	77	8	45
2	62	1	96	4	56	0	68	3	24	5	41	6	19	7	2	8	73	9	50
2	86	0	53	3	3	1	89	4	37	7	100	5	59	9	23	6	19	8	35
3	90	4	94	0	21	2	78	1	85	5	94	6	90	8	28	9	92	7	56
4	85	2	97	0	8	3	27	1	86	9	26	7	5	8	96	5	68	6	57
0	58	3	4	4	49	2	1	1	79	8	10	6	44	9	87	5	16	7	13
3	85	0	24	4	23	1	41	2	59	8	20	6	52	5	58	9	75	7	77
0	47	1	89	2	68	4	88	3	17	6	48	8	84	9	100	5	92	7	47
1	30	0	1	3	61	4	20	2	73	8	78	7	41	9	52	5	43	6	74
0	11	4	58	3	66	2	67	1	18	8	42	7	88	9	49	5	62	6	71
4	5	2	51	3	67	1	20	0	11	7	37	6	42	8	25	9	57	5	1
0	58	4	83	2	9	3	68	1	21	6	28	9	77	5	19	7	32	8	66
3	85	2	58	0	65	1	80	4	50	7	79	5	43	8	29	9	9	6	18
3	74	2	29	0	11	1	23	4	34	7	84	8	57	5	77	6	83	9	82
2	6	4	67	0	97	3	66	1	21	8	90	9	46	6	12	5	17	7	96
4	34	1	5	2	13	0	100	3	12	8	63	7	59	5	75	6	91	9	89
1	30	2	66	0	33	3	70	4	16	6	80	5	58	8	8	7	86	9	66
3	55	0	46	2	1	1	77	4	19	7	85	9	32	6	59	5	37	8	69
2	3	0	16	1	48	4	8	3	51	7	72	6	19	8	58	9	59	5	94
3	30	4	23	1	92	0	18	2	19	9	32	6	57	5	50	7	64	8	27
2	18	0	72	4	92	1	6	3	67	8	100	6	32	9	14	5	51	7	55
4	48	0	87	1	96	2	58	3	83	8	77	5	26	7	77	9	72	6	86
1	80	4	5	0	50	3	65	2	85	7	88	5	47	6	33	8	50	9	75
1	78	0	96	4	80	3	5	2	99	9	58	5	38	7	29	8	69	6	44

swv15 (50 x 10)

2	93	4	40	0	1	3	77	1	77	5	16	9	74	8	11	6	51	7	92
0	92	4	80	1	76	3	59	2	70	5	86	9	17	6	78	7	30	8	93
1	44	2	92	3	96	4	77	0	53	9	10	7	49	5	84	8	59	6	14
1	60	2	19	3	76	0	73	4	85	7	13	8	93	5	68	9	50	6	78
2	20	0	24	3	41	1	2	4	4	9	44	7	79	8	81	5	16	6	39
3	41	2	35	1	32	4	18	0	15	8	98	6	29	5	19	7	14	9	26
1	59	0	45	4	53	3	44	2	98	5	84	6	23	7	45	8	39	9	89
1	30	4	51	3	25	0	51	2	84	6	60	5	45	7	89	8	25	9	97
0	47	3	18	2	40	4	62	1	58	5	36	7	93	8	77	9	90	6	15
3	33	1	68	0	41	4	72	2	20	6	69	7	47	5	22	9	47	8	22
2	28	1	100	4	20	0	35	3	26	5	24	9	41	6	42	7	100	8	32
0	65	2	12	4	53	3	93	1	40	8	18	7	23	5	60	6	89	9	53
0	58	1	60	4	97	3	31	2	50	9	85	5	64	7	38	6	85	8	35
3	64	0	58	1	49	2	45	4	9	8	49	6	22	5	99	9	15	7	7
0	10	4	85	3	72	2	37	1	77	5	70	7	45	9	8	6	83	8	57
4	93	0	87	1	87	2	18	3	4	8	78	5	67	9	20	6	17	7	35
4	72	0	56	3	57	2	15	1	45	6	41	5	40	9	85	8	32	7	81
0	36	3	63	4	79	2	32	1	5	6	25	7	86	9	91	5	21	8	35
2	83	4	29	0	9	1	38	3	73	7	50	9	99	5	18	8	29	6	41
0	100	3	29	2	60	4	63	1	64	8	71	6	35	5	26	9	9	7	22
1	81	0	60	3	62	4	48	2	68	7	28	5	69	8	92	6	79	9	10
0	40	4	80	1	41	2	10	3	68	8	28	9	51	7	33	6	82	5	25
4	30	2	12	0	35	3	17	1	70	9	29	7	18	8	93	6	94	5	37
1	36	2	41	3	27	4	36	0	78	7	64	6	88	5	25	9	92	8	66
2	65	3	27	4	74	0	32	1	40	5	88	8	73	6	92	7	83	9	42
0	48	1	85	2	92	4	95	3	61	8	72	9	76	5	58	7	11	6	89
3	84	2	50	0	70	4	24	1	42	9	55	5	100	6	70	7	4	8	68
0	95	4	41	2	11	3	98	1	85	5	64	6	8	7	26	8	6	9	6
0	84	2	49	1	17	3	69	4	55	8	75	6	45	9	38	7	59	5	28
2	48	0	29	4	1	1	64	3	41	5	23	7	64	9	31	6	56	8	12
2	81	4	25	3	33	0	22	1	50	5	74	9	56	8	33	7	85	6	83
1	62	4	25	0	21	2	20	3	8	6	36	9	9	5	91	8	90	7	49
1	43	0	16	2	91	3	96	4	24	5	11	9	91	7	41	8	35	6	66
1	91	2	20	4	44	0	42	3	87	9	57	6	15	5	38	8	42	7	89
0	33	3	95	4	68	2	22	1	80	7	53	8	13	9	70	5	22	6	69
0	15	3	47	1	24	2	31	4	41	8	14	9	28	7	59	5	52	6	39
2	95	0	42	4	5	1	57	3	67	6	30	9	21	8	70	5	9	7	20
2	54	0	15	1	20	3	64	4	83	9	40	7	6	5	89	6	91	8	48
0	22	4	27	1	77	3	25	2	16	8	72	9	61	6	75	7	4	5	19
3	68	1	82	2	16	0	83	4	2	7	10	8	88	5	41	9	21	6	66
1	64	0	76	2	85	3	71	4	97	5	97	7	8	6	40	8	70	9	35
0	94	1	45	2	94	4	84	3	44	8	41	5	30	7	47	6	19	9	22
2	23	1	10	0	82	3	93	4	90	8	67	7	9	9	18	5	22	6	87
0	75	2	27	4	97	3	9	1	57	9	14	5	50	7	31	8	62	6	23
1	42	3	41	2	35	0	75	4	18	9	65	7	38	6	38	8	51	5	56
4	72	1	63	0	33	2	27	3	41	5	52	7	42	9	10	6	14	8	71
2	91	1	89	0	44	4	91	3	26	6	49	5	22	8	31	9	69	7	5
3	42	1	34	0	4	4	34	2	16	6	86	7	25	8	99	5	67	9	25
4	34	1	93	0	26	3	81	2	9	7	96	8	79	9	68	5	76	6	10
3	19	1	47	4	13	2	98	0	32	7	12	9	45	6	52	8	49	5	34

swv16 (50 x 10)

1	55	3	46	5	71	8	29	0	47	2	12	7	57	4	79	6	91	9	30
2	96	6	94	8	98	0	55	3	10	1	95	5	95	7	37	9	82	4	2
6	43	3	93	8	30	2	41	0	23	1	60	7	14	4	15	5	42	9	56

0	45	6	85	2	59	7	76	1	93	9	62	4	33	8	46	5	33	3	35
2	45	3	36	8	11	6	96	7	96	1	8	0	75	5	6	4	13	9	2
9	51	7	75	0	4	3	13	5	12	1	4	2	38	6	30	4	42	8	28
9	58	4	33	6	77	2	11	3	37	8	64	5	94	7	89	1	96	0	93
6	37	3	67	0	88	9	92	8	19	4	27	7	46	1	58	2	60	5	55
4	60	2	88	0	23	5	69	8	60	1	32	7	4	6	56	9	25	3	14
2	98	5	56	1	68	6	63	7	61	3	78	8	45	0	62	4	31	9	70
7	66	8	80	0	18	3	97	9	47	5	38	1	26	2	8	6	90	4	90
0	16	7	6	4	53	6	86	5	81	8	49	3	90	2	57	1	34	9	56
2	69	8	65	5	20	4	15	1	61	3	71	6	71	9	58	0	24	7	71
4	84	5	20	9	58	0	55	8	98	2	75	7	46	3	81	1	71	6	46
5	6	6	58	7	90	1	54	9	73	0	92	4	39	3	23	2	100	8	18
2	32	5	58	6	97	1	49	3	61	0	69	8	2	4	3	9	32	7	46
0	78	7	14	4	98	3	26	8	25	9	45	6	12	2	98	1	99	5	69
2	50	1	95	4	82	9	25	0	68	8	83	5	36	7	78	3	35	6	27
6	29	7	20	8	55	4	14	2	66	5	52	0	75	9	63	1	93	3	64
1	11	0	18	9	42	4	81	7	2	2	39	3	83	6	11	5	38	8	52
4	11	8	99	9	2	7	10	3	91	5	83	6	61	0	21	2	69	1	8
9	11	7	65	1	14	2	85	3	5	8	5	5	11	4	47	6	67	0	41
9	60	7	9	8	16	2	4	5	34	6	2	4	30	1	32	0	51	3	51
9	31	2	41	1	13	6	28	5	97	3	8	7	42	4	95	8	46	0	93
4	1	6	91	8	49	3	75	1	19	7	100	0	58	2	14	5	34	9	82
3	28	5	68	9	30	7	68	1	10	6	20	8	47	4	51	0	44	2	32
9	86	3	9	1	80	0	89	5	93	4	12	8	13	7	10	6	18	2	4
0	22	5	12	8	95	4	24	3	30	1	81	2	21	7	28	9	100	6	27
1	87	0	68	2	64	3	33	7	59	5	95	6	1	9	14	8	82	4	43
2	14	6	98	0	86	1	85	8	85	5	12	4	99	7	8	3	21	9	7
5	47	9	90	0	88	1	52	8	43	4	62	7	33	3	51	6	97	2	22
2	59	7	26	4	76	0	26	3	71	8	59	1	73	9	70	5	57	6	10
6	92	2	10	9	45	0	11	1	53	3	35	8	76	4	83	7	55	5	79
9	96	4	3	3	92	7	67	6	60	8	35	5	70	0	52	2	39	1	94
4	65	0	17	9	26	7	46	5	81	1	42	2	64	6	46	3	96	8	59
9	6	3	21	8	46	0	82	2	74	5	56	7	94	6	83	4	63	1	21
6	89	5	23	8	78	2	33	9	4	7	97	3	60	1	29	0	79	4	93
0	46	1	46	4	20	7	91	2	76	9	83	3	14	6	61	5	84	8	76
7	82	8	43	6	76	1	36	0	27	9	93	5	71	4	81	2	45	3	62
7	51	9	27	5	12	6	52	4	85	8	66	0	100	3	44	2	82	1	36
3	75	7	13	6	63	1	78	4	1	8	60	2	24	5	10	9	56	0	3
5	48	4	32	2	82	0	1	1	2	7	35	3	16	9	67	8	74	6	39
7	24	0	8	8	96	3	59	2	41	4	23	1	37	9	4	5	69	6	27
1	23	9	3	2	85	6	93	5	18	7	47	0	96	8	6	4	60	3	3
6	99	2	14	9	16	3	81	8	89	1	53	7	86	4	39	5	3	0	87
5	67	8	53	0	77	4	69	2	55	3	78	6	95	1	76	7	2	9	71
1	5	6	89	0	37	3	88	7	20	9	4	4	77	8	27	5	31	2	47
1	66	2	55	4	15	7	35	3	76	9	91	6	35	5	37	8	54	0	33
3	79	5	2	6	17	1	65	7	27	8	53	4	52	9	35	0	23	2	59
9	100	0	55	5	14	2	86	4	69	3	87	8	46	1	3	6	89	7	100

swv17 (50 x 10)

7	9	2	57	9	62	5	34	6	83	0	33	1	80	4	46	3	21	8	89
9	82	1	35	8	37	5	26	6	21	3	78	7	64	4	33	2	40	0	21
7	14	5	49	3	48	9	34	4	52	1	16	2	78	0	24	8	58	6	43
2	94	3	86	8	41	5	27	7	29	6	53	9	5	0	36	4	98	1	37
7	55	1	87	8	51	5	29	9	93	3	51	0	54	6	85	2	20	4	29
2	88	1	98	3	67	8	41	6	23	9	70	7	26	4	28	5	17	0	87
2	78	0	18	4	43	3	86	9	78	6	43	7	62	8	42	1	44	5	9
9	37	4	89	3	26	6	59	0	89	5	90	1	91	8	28	7	37	2	51
3	82	2	31	1	98	5	25	0	16	7	23	9	92	4	89	6	32	8	12

6	66	1	58	5	14	3	42	0	62	8	66	4	46	7	88	2	89	9	97
8	94	9	11	6	3	1	86	2	4	5	19	7	93	4	43	0	78	3	11
5	22	1	87	9	61	2	2	3	15	6	37	7	81	0	17	8	31	4	73
6	28	0	86	3	54	2	68	4	63	1	33	8	22	5	35	9	84	7	15
6	18	1	2	2	23	8	49	7	82	9	8	4	73	5	31	3	20	0	1
7	49	5	8	2	36	8	31	6	47	3	90	0	7	9	6	1	44	4	51
4	43	1	95	0	18	9	99	7	98	3	26	8	99	5	90	2	24	6	91
1	49	6	69	3	73	9	52	0	10	7	41	8	42	5	96	4	85	2	76
0	5	1	69	3	38	7	35	5	23	2	40	8	17	4	33	6	99	9	82
3	42	1	93	4	90	6	88	2	70	8	11	9	54	7	76	5	40	0	94
5	88	9	44	0	63	7	92	1	4	4	91	6	92	8	53	3	52	2	38
5	83	3	75	1	44	2	79	7	63	6	32	0	10	4	2	9	6	8	56
7	71	0	23	5	93	3	44	6	36	4	27	2	96	1	23	9	35	8	21
5	42	2	43	6	37	9	98	0	55	3	35	4	45	1	8	8	5	7	100
0	40	8	34	2	7	9	17	5	60	4	98	7	34	6	23	1	37	3	58
9	87	2	39	3	23	8	48	6	83	7	50	5	9	1	49	0	37	4	42
6	60	5	3	2	60	7	40	0	54	1	68	4	49	8	50	9	22	3	34
5	22	1	55	2	32	0	83	8	38	4	22	6	29	7	23	9	59	3	90
9	51	2	27	6	81	8	87	0	79	7	1	3	14	5	73	4	25	1	14
6	88	1	46	5	16	2	62	9	95	7	63	4	78	0	9	3	68	8	37
4	77	2	13	8	96	3	61	0	21	7	39	5	12	6	49	9	73	1	86
7	91	5	14	3	37	0	17	9	49	4	27	1	68	2	60	6	42	8	15
9	13	4	25	6	62	0	4	1	31	8	76	5	3	7	8	3	26	2	95
7	45	5	50	1	14	0	69	9	43	4	1	6	73	8	35	3	1	2	61
4	57	1	1	0	74	8	1	6	96	2	92	7	85	5	42	3	12	9	38
7	49	5	31	8	79	6	83	1	40	4	65	3	34	2	32	9	97	0	25
9	24	5	40	4	81	3	10	6	59	8	83	2	66	1	28	7	33	0	31
5	33	4	39	3	50	1	96	7	62	2	72	8	42	6	86	9	66	0	80
3	88	7	47	0	35	4	69	1	79	9	61	2	25	8	56	5	68	6	96
9	23	6	95	0	42	1	84	8	57	4	42	2	2	5	79	3	29	7	90
9	96	8	21	4	17	7	12	1	25	2	9	6	7	5	26	0	81	3	51
1	63	7	16	6	40	2	22	9	48	5	87	0	15	8	24	3	37	4	55
7	95	0	60	3	62	2	7	9	2	8	81	5	83	4	64	1	68	6	66
3	24	7	60	6	35	2	77	1	85	8	57	9	29	5	59	4	53	0	14
1	24	6	30	0	9	3	89	8	72	4	77	2	7	5	23	9	73	7	35
0	66	8	12	1	9	5	50	2	14	9	76	4	90	3	43	7	48	6	63
3	97	1	29	0	59	4	64	9	17	2	77	5	60	7	16	6	61	8	40
9	5	4	22	2	3	8	63	5	1	7	23	0	1	3	61	1	92	6	19
6	91	8	74	1	88	5	2	7	61	4	39	0	35	2	23	9	84	3	27
8	87	5	58	7	44	1	6	6	22	3	57	9	78	4	19	2	74	0	6
4	6	1	94	0	45	2	54	9	67	7	90	5	19	8	72	6	70	3	58

swv18 (50 x 10)

7	35	6	23	2	92	4	5	5	40	1	90	3	30	9	35	8	8	0	86
2	60	3	97	8	21	9	70	7	82	0	12	4	3	5	45	1	75	6	69
7	96	2	38	0	61	1	55	4	31	5	48	9	79	3	4	6	12	8	29
4	83	7	82	8	97	1	43	0	95	6	92	2	18	3	29	5	4	9	67
3	46	9	80	8	66	2	38	4	95	1	40	7	89	0	32	6	64	5	1
6	57	4	80	8	68	7	27	0	90	5	45	3	98	9	59	1	6	2	94
5	50	0	91	2	97	9	63	7	52	3	48	4	4	8	96	1	18	6	100
7	23	6	43	3	25	8	83	2	76	9	41	1	88	0	31	5	44	4	13
2	20	3	90	9	20	4	42	8	72	5	46	1	27	0	81	6	40	7	34
7	80	5	97	0	42	2	49	9	10	1	10	3	71	4	71	6	14	8	98
2	79	3	29	0	96	7	66	1	58	8	31	4	47	5	76	6	59	9	88
8	93	6	3	1	7	3	27	5	66	7	23	0	60	4	97	2	66	9	55
9	12	8	39	4	77	5	79	0	26	7	58	2	98	6	38	3	31	1	28
6	8	9	48	4	4	1	87	3	38	2	28	8	10	0	19	7	82	5	83
5	6	9	13	2	86	6	19	3	26	7	79	0	55	1	85	8	33	4	30

3	37	8	26	7	29	6	74	9	43	5	17	0	45	2	28	1	58	4	15
7	15	3	37	6	21	5	47	2	90	0	37	9	33	1	42	4	7	8	62
8	49	4	46	1	28	7	18	6	41	2	57	0	75	3	21	9	3	5	32
6	98	1	30	8	24	4	91	9	73	7	25	5	49	0	40	2	9	3	4
6	33	3	94	1	21	2	90	9	86	7	85	5	29	0	17	4	94	8	90
6	3	4	85	1	66	7	61	8	57	3	84	2	5	9	40	0	54	5	70
7	81	1	98	2	45	0	18	6	65	9	1	4	98	3	30	8	84	5	82
6	40	7	77	3	72	1	97	5	39	4	21	0	59	8	42	9	90	2	26
5	57	3	63	1	14	4	64	6	23	8	78	2	54	0	51	9	100	7	96
5	61	1	55	6	73	2	87	4	35	3	41	7	96	0	32	8	91	9	60
9	19	5	90	8	91	0	45	3	66	2	84	1	61	7	3	6	84	4	100
2	33	9	72	6	27	8	14	3	59	0	39	7	20	5	29	4	54	1	88
4	45	0	18	3	73	2	26	8	55	6	22	7	27	1	46	9	43	5	77
2	57	9	16	1	71	8	25	7	50	3	41	6	58	5	71	4	9	0	32
8	48	9	32	0	42	3	73	1	56	7	53	6	3	5	66	4	15	2	44
6	69	7	14	1	2	8	40	4	70	9	90	3	38	2	31	5	55	0	50
9	100	8	14	0	55	2	5	5	12	4	79	1	68	3	83	6	89	7	78
4	26	5	44	8	39	1	84	7	64	9	98	3	38	2	2	6	27	0	18
3	98	2	10	9	99	8	50	0	20	6	12	4	7	1	57	7	87	5	89
0	64	8	63	7	98	5	31	1	30	6	62	3	11	4	89	9	31	2	34
3	26	6	43	4	69	7	27	8	92	2	51	1	10	5	29	9	21	0	37
8	21	5	98	0	64	6	38	2	23	1	13	7	89	9	89	4	21	3	27
4	39	7	32	1	67	0	33	5	16	2	43	6	62	3	42	9	70	8	90
7	73	9	45	3	37	0	45	2	61	6	25	5	15	4	5	8	58	1	98
7	94	0	17	6	15	5	81	9	64	3	62	1	2	8	16	2	35	4	40
5	32	6	37	9	11	0	25	1	37	8	21	2	76	7	52	4	56	3	87
3	23	2	40	1	6	7	31	6	25	9	98	8	29	4	4	5	25	0	33
8	96	9	30	1	95	3	2	6	3	2	22	0	62	4	30	7	1	5	99
9	54	5	3	0	78	2	43	6	90	7	88	4	1	8	97	1	30	3	96
5	29	6	60	3	80	1	94	2	67	0	42	8	17	9	27	7	75	4	86
1	17	5	62	2	25	7	80	6	62	9	19	8	81	3	73	0	57	4	90
9	31	3	54	5	28	1	19	4	4	2	34	8	64	6	46	7	60	0	27
9	95	7	1	2	43	3	6	4	7	8	66	1	45	5	13	0	80	6	1
3	20	7	82	0	87	1	65	6	64	8	61	2	21	5	32	9	16	4	37
0	49	3	54	2	31	8	69	1	21	5	2	6	73	9	35	4	66	7	82

swv19 (50 x 10)

7	74	1	27	5	66	3	89	6	58	0	11	8	77	9	17	2	70	4	97
5	10	0	11	2	38	3	60	1	50	7	35	6	94	9	52	4	2	8	20
7	17	0	65	6	93	8	62	9	91	5	2	1	51	2	4	3	19	4	10
4	87	3	3	9	81	0	17	6	44	2	82	7	16	5	13	8	100	1	85
9	18	6	33	7	35	0	78	2	68	3	68	8	3	5	2	4	53	1	25
2	36	8	41	6	60	9	43	0	66	5	34	3	24	7	11	1	5	4	55
9	52	4	99	6	62	0	50	1	24	8	73	7	19	3	23	2	15	5	2
4	85	9	21	3	27	7	53	0	86	1	36	6	35	5	99	8	30	2	43
6	43	5	31	9	99	2	12	0	6	7	79	3	81	1	18	8	73	4	55
4	90	6	100	1	15	0	40	7	96	9	25	5	43	8	23	2	31	3	7
5	61	4	88	6	10	3	48	0	100	2	62	1	83	8	20	7	42	9	19
9	35	7	41	6	16	3	58	0	86	2	69	5	58	1	93	4	47	8	77
2	61	0	40	4	99	1	51	7	46	6	39	3	43	9	37	8	88	5	9
4	15	8	38	2	84	5	98	6	17	1	91	7	91	9	23	3	48	0	98
3	26	2	42	8	55	4	24	0	43	1	83	9	27	7	38	6	37	5	58
5	21	8	78	6	97	0	77	9	82	4	26	3	22	1	90	7	57	2	31
4	3	9	44	3	90	1	64	5	52	8	35	7	18	2	45	0	4	6	14
8	60	6	59	3	67	2	85	0	43	7	93	5	44	4	22	1	68	9	38
4	77	8	41	2	74	6	99	0	100	1	45	9	14	3	26	7	98	5	77
8	38	9	57	7	42	5	64	1	80	6	81	4	70	3	13	2	41	0	65
9	36	4	22	8	39	0	76	1	78	2	27	5	55	3	10	6	5	7	71

7	70	9	81	1	60	5	85	3	63	6	97	2	61	8	44	0	5	4	35
9	38	0	94	2	46	5	20	8	87	1	41	4	41	3	40	7	99	6	48
7	30	6	9	5	13	2	79	8	81	0	25	9	93	4	85	3	78	1	76
4	6	8	58	6	51	7	48	2	68	3	34	5	78	9	59	1	98	0	36
4	90	6	56	7	97	9	37	0	38	1	47	2	56	3	8	5	37	8	7
0	66	8	15	1	39	5	89	7	3	9	54	3	24	2	14	6	99	4	73
3	12	9	37	4	79	8	95	0	50	1	74	6	1	5	55	7	98	2	49
8	99	9	79	3	99	2	87	0	80	4	13	5	99	6	13	1	54	7	61
1	51	9	21	3	32	6	20	0	80	7	58	2	91	5	84	8	62	4	91
1	11	8	38	2	14	9	12	3	39	5	34	0	37	6	94	4	10	7	2
6	76	9	86	3	40	4	30	2	97	0	59	8	100	7	9	5	55	1	86
3	33	1	49	0	94	2	17	6	17	8	70	5	17	7	42	4	26	9	24
4	75	1	20	9	93	2	58	3	51	0	94	6	24	7	70	8	51	5	82
8	59	1	9	3	59	5	62	9	79	7	53	6	48	4	98	2	76	0	71
6	90	2	35	5	89	0	59	9	28	7	51	4	69	3	36	1	32	8	27
5	10	6	85	4	97	1	3	0	79	9	86	3	10	7	80	2	37	8	39
7	60	0	27	5	69	8	58	6	67	2	36	9	31	3	69	1	16	4	22
2	27	5	16	6	15	4	40	8	16	1	92	9	60	7	43	3	2	0	7
1	79	7	99	0	27	9	56	5	29	6	17	8	67	4	34	3	86	2	61
6	57	7	100	4	73	9	17	8	3	3	64	2	99	0	71	5	27	1	90
2	80	5	23	4	54	6	39	9	77	3	65	7	59	0	7	1	63	8	32
4	98	6	17	8	44	5	1	3	10	7	56	2	95	9	80	0	99	1	64
8	60	7	74	3	60	6	30	0	81	5	25	4	89	9	19	2	59	1	21
1	67	0	42	8	93	2	47	5	34	7	11	6	100	9	15	4	99	3	2
9	35	3	61	5	93	8	83	7	87	4	66	0	96	2	55	1	41	6	61
8	22	5	25	7	29	3	70	6	93	1	19	0	49	9	62	2	19	4	73
8	11	4	93	5	97	1	28	2	14	0	75	7	41	3	40	9	62	6	66
7	76	6	61	8	64	3	90	0	20	2	43	9	50	1	13	5	4	4	47
3	38	4	11	0	30	5	37	7	57	9	64	1	68	8	42	2	19	6	79

swv20 (50 x 10)

8	100	7	30	4	42	9	11	2	31	1	71	5	41	0	1	3	55	6	94
4	81	6	20	3	96	7	39	8	29	0	90	9	61	2	64	1	86	5	47
5	80	0	56	1	88	7	19	2	68	8	95	3	44	4	22	9	60	6	80
4	86	6	70	0	88	2	15	7	50	1	54	9	88	3	25	8	89	5	33
0	48	1	57	4	86	8	60	3	78	5	4	9	60	7	40	2	11	6	25
6	23	7	9	1	90	0	51	2	52	9	14	5	30	4	1	8	25	3	83
1	30	4	75	5	76	9	100	7	54	2	41	6	50	8	75	0	1	3	28
2	46	3	78	1	37	7	12	6	56	4	50	8	66	5	39	0	8	9	72
1	24	6	90	0	32	3	6	2	99	9	22	8	12	4	63	7	81	5	52
6	62	3	9	8	59	0	66	4	41	1	32	5	29	7	79	9	84	2	4
9	57	5	99	6	2	3	17	0	51	7	10	4	14	1	64	2	99	8	27
7	81	0	67	9	83	2	30	5	25	6	87	1	29	3	7	8	93	4	1
5	65	8	53	9	48	4	28	7	74	0	60	6	77	2	22	1	5	3	98
1	97	5	37	0	71	7	49	6	51	3	17	4	38	9	67	8	28	2	31
0	20	8	94	3	39	6	73	9	63	4	8	2	57	1	27	7	26	5	42
8	77	1	68	9	20	7	100	4	1	5	77	6	17	3	35	2	65	0	86
8	68	6	62	4	79	7	84	1	60	3	56	0	10	9	86	5	60	2	30
4	71	2	74	6	6	1	56	3	69	0	8	8	50	9	78	5	4	7	89
8	29	5	5	1	59	3	96	0	46	4	91	2	48	7	53	6	21	9	82
2	19	9	96	0	73	1	39	5	54	8	50	7	60	3	50	4	65	6	78
7	68	4	15	2	26	3	26	0	13	9	13	5	96	8	70	6	27	1	93
6	41	8	18	4	66	7	9	1	31	2	92	0	3	3	78	5	41	9	53
5	9	0	64	2	15	6	73	4	12	1	43	8	89	7	69	3	32	9	22
5	93	6	19	3	74	8	81	0	72	2	94	9	19	1	26	4	53	7	7
3	48	2	29	5	51	8	72	7	35	6	32	1	38	0	98	4	58	9	54
0	94	9	23	4	41	6	53	2	53	7	27	1	62	3	68	8	84	5	49
4	4	1	4	0	66	7	90	9	78	2	29	5	2	6	86	3	23	8	46

3	78	5	61	2	97	7	68	8	92	0	15	4	12	6	77	1	12	9	22
0	100	7	89	6	71	2	70	8	89	4	72	5	78	3	23	9	37	1	2
0	91	3	74	2	36	4	72	6	62	1	80	9	20	7	77	5	47	8	80
1	44	0	67	4	66	8	99	6	59	5	5	7	15	2	38	3	40	9	19
1	69	9	35	3	86	0	7	2	35	5	32	6	66	4	89	8	63	7	52
3	3	4	68	1	66	7	27	6	41	5	2	9	77	0	45	2	40	8	39
4	66	3	42	7	79	0	55	6	98	9	44	5	6	8	73	1	55	2	1
3	80	8	18	9	94	2	27	5	42	4	17	7	74	0	65	6	6	1	27
2	73	4	70	5	51	0	84	8	29	9	95	1	97	7	28	3	68	6	89
9	85	6	56	5	54	3	76	2	50	0	43	1	8	7	93	4	17	8	65
1	1	3	17	2	61	5	38	4	71	7	18	0	40	9	94	6	41	8	74
3	30	8	22	6	39	9	56	5	3	7	64	4	74	2	21	0	93	1	1
0	17	8	8	9	20	5	38	3	85	7	5	2	63	1	18	4	89	6	88
8	87	5	44	0	42	1	34	9	11	7	13	3	71	4	88	6	32	2	12
2	39	1	73	6	43	0	48	9	77	8	48	5	23	7	66	3	94	4	68
1	98	7	19	3	69	6	5	8	85	9	19	0	30	2	43	5	87	4	70
2	45	1	60	4	30	9	71	5	35	0	75	3	75	6	41	8	67	7	37
3	63	7	39	2	16	9	69	1	46	5	20	6	57	4	51	0	66	8	40
2	7	7	73	6	17	1	21	0	24	8	2	5	68	4	22	9	36	3	60
1	20	4	17	8	12	9	29	5	28	0	7	3	38	6	57	7	22	2	75
5	53	4	7	7	5	8	27	9	38	2	100	6	48	0	53	1	11	3	18
1	49	7	47	4	81	8	9	0	20	2	63	3	15	6	1	9	10	5	5
4	49	6	27	7	17	5	64	2	30	8	56	0	42	3	97	9	82	1	34

- 10 problems used by Applegate and Cook [1991], referred as orb01 – orb10

orb01 (10 x 10)

0	72	1	64	2	55	3	31	4	53	5	95	6	11	7	52	8	6	9	84
0	61	3	27	4	88	2	78	1	49	5	83	8	91	6	74	7	29	9	87
0	86	3	32	1	35	2	37	5	18	4	48	6	91	7	52	9	60	8	30
0	8	1	82	4	27	3	99	6	74	5	9	2	33	9	20	7	59	8	98
1	50	0	94	5	43	3	62	4	55	7	48	2	5	8	36	9	47	6	36
0	53	6	30	2	7	3	12	1	68	8	87	4	28	9	70	7	45	5	7
2	29	3	96	0	99	1	14	4	34	7	14	5	7	6	76	8	57	9	76
2	90	0	19	3	87	4	51	1	84	5	45	9	84	6	58	7	81	8	96
2	97	1	99	4	93	0	38	7	13	5	96	3	40	9	64	6	32	8	45
2	44	0	60	8	29	3	5	6	74	1	85	4	34	7	95	9	51	5	47

orb02 (10 x 10)

0	72	1	54	2	33	3	86	4	75	5	16	6	96	7	7	8	99	9	76
0	16	3	88	4	48	8	52	9	60	6	29	7	18	5	89	2	80	1	76
0	47	7	11	3	14	2	56	6	16	4	83	1	10	5	61	8	24	9	58
0	49	1	31	3	17	8	50	5	63	2	35	4	65	7	23	6	50	9	29
0	55	6	6	1	28	3	96	5	86	2	99	9	14	7	70	8	64	4	24
4	46	0	23	6	70	8	19	2	54	3	22	9	85	7	87	5	79	1	93
4	76	3	60	0	76	9	98	2	76	1	50	8	86	7	14	6	27	5	57
4	93	6	27	9	57	3	87	8	86	2	54	7	24	5	49	0	20	1	47
2	28	6	11	8	78	7	85	4	63	9	81	3	10	1	9	5	46	0	32
2	22	9	76	5	89	8	13	6	88	3	10	7	75	4	98	1	78	0	17

orb03 (10 x 10)

```
 0 96 1 69 2 25 3  5 4 55 5 15 6 88 7 11 8 17 9 82
 0 11 1 48 2 67 3 38 4 18 7 24 6 62 5 92 9 96 8 81
 2 67 1 63 0 93 4 85 3 25 5 72 6 51 7 81 8 58 9 15
 2 30 1 35 0 27 4 82 3 44 7 92 6 25 5 49 9 28 8 77
 1 53 0 83 4 73 3 26 2 77 6 33 5 92 9 99 8 38 7 38
 1 20 0 44 4 81 3 88 2 66 6 70 5 91 9 37 8 55 7 96
 1 21 2 93 4 22 0 56 3 34 6 40 7 53 9 46 5 29 8 63
 1 32 2 63 4 36 0 26 3 17 5 85 7 15 8 55 9 16 6 82
 0 73 2 46 3 89 4 24 1 99 6 92 7  7 9 51 5 19 8 14
 0 52 2 20 3 70 4 98 1 23 5 15 7 81 8 71 9 24 6 81
```

orb04 (10 x 10)

```
 0  8 1 10 2 35 3 44 4 15 5 92 6 70 7 89 8 50 9 12
 0 63 8 39 3 80 5 22 2 88 1 39 9 85 6 27 7 74 4 69
 0 52 6 22 1 33 3 68 8 27 2 68 5 25 4 34 7 24 9 84
 0 31 1 85 4 55 8 80 5 58 7 11 6 69 9 56 3 73 2 25
 0 97 5 98 9 87 8 47 7 77 4 90 3 98 2 80 1 39 6 40
 1 97 5 68 0 44 9 67 2 44 8 85 3 78 6 90 7 33 4 81
 0 34 3 76 8 48 7 61 9 11 2 36 4 33 6 98 1  7 5 44
 0 44 9  5 4 85 1 51 5 58 7 79 2 95 6 48 3 86 8 73
 0 24 1 63 9 48 7 77 8 73 6 74 4 63 5 17 2 93 3 84
 0 51 2  5 4 40 9 60 1 46 5 58 8 54 3 72 6 29 7 94
```

orb05 (10 x 10)

```
 9 11 8 93 0 48 7 76 6 13 5 71 3 59 2 90 4 10 1 65
 8 52 9 76 0 84 7 73 5 56 4 10 6 26 2 43 3 39 1 49
 9 28 8 44 7 26 6 66 4 68 5 74 3 27 2 14 1  6 0 21
 0 18 1 58 3 62 2 46 6 25 4  6 5 60 7 28 8 80 9 30
 0 78 1 47 7 29 5 16 4 29 6 57 3 78 2 87 8 39 9 73
 9 66 8 51 3 12 7 64 5 67 4 15 6 66 2 26 1 20 0 98
 8 23 9 76 6 45 7 75 5 24 3 18 4 83 2 15 1 88 0 17
 9 56 8 83 7 80 6 16 4 31 5 93 3 30 2 29 1 66 0 28
 9 79 8 69 2 82 4 16 5 62 3 41 6 91 7 35 0 34 1 75
 0  5 1 19 2 20 3 12 4 94 5 60 6 99 7 31 8 96 9 63
```

orb06 (10 x 10)

```
 0 99 1 74 2 49 3 67 4 17 5  7 6  9 7 39 8 35 9 49
 0 49 3 67 4 82 2 92 1 62 5 84 8 45 6 30 7 42 9 71
 0 26 3 33 1 82 2 98 5 83 4 16 6 64 7 65 9 36 8 77
 0 41 1 62 4 73 3 94 6 51 5 46 2 55 9 31 7 64 8 46
 1 68 0 26 5 50 3 46 4 25 7 88 2  6 8 13 9 98 6 84
 0 24 6 80 2 91 3 55 1 48 8 99 4 72 9 91 7 84 5 12
 2 16 3 13 0  9 1 58 4 23 7 85 5 36 6 89 8 71 9 41
 2 54 0 41 3 38 4 53 1 11 5 74 9 88 6 46 7 41 8 65
 2 53 1 50 4 40 0 90 7  7 5 80 3 57 9 60 6 91 8 47
 2 45 0 59 8 81 3 99 6 71 1 19 4 75 7 77 9 94 5 95
```

orb07 (10 x 10)

```
 0 32 1 14 2 15 3 37 4 18 5 43 6 19 7 27 8 28 9 31
```

0	8	3	12	4	49	8	24	9	52	6	19	7	23	5	19	2	17	1	32
0	25	7	19	3	27	2	45	6	21	4	15	1	13	5	16	8	43	9	19
0	24	1	18	3	41	8	29	5	14	2	17	4	23	7	15	6	18	9	23
0	27	6	29	1	39	3	21	5	15	2	15	9	25	7	26	8	44	4	20
4	17	0	15	6	51	8	17	2	46	3	16	9	33	7	25	5	30	1	25
4	15	3	31	0	25	9	12	2	13	1	51	8	19	7	21	6	12	5	26
4	8	6	29	9	25	3	15	8	17	2	22	7	32	5	20	0	11	1	28
2	41	6	10	8	32	7	5	4	21	9	59	3	26	1	10	5	16	0	29
2	20	9	7	5	44	8	22	6	33	3	25	7	29	4	12	1	14	0	0

orb08 (10 x 10)

0	55	1	74	2	45	3	23	4	76	5	19	6	18	7	61	8	44	9	11
0	63	1	43	2	51	3	18	4	42	7	11	6	29	5	52	9	29	8	88
2	88	1	31	0	47	4	10	3	62	5	60	6	58	7	29	8	52	9	92
2	16	1	71	0	55	4	55	3	9	7	49	6	83	5	54	9	7	8	57
1	7	0	41	4	92	3	94	2	46	6	79	5	34	9	38	8	8	7	18
1	25	0	5	4	89	3	94	2	14	6	94	5	20	9	23	8	44	7	39
1	24	2	21	4	47	0	40	3	94	6	71	7	89	9	75	5	97	8	15
1	5	2	7	4	74	0	28	3	72	5	61	7	9	8	53	9	32	6	97
0	34	2	52	3	37	4	6	1	94	6	6	7	56	9	41	5	5	8	16
0	77	2	74	3	82	4	10	1	29	5	15	7	51	8	65	9	37	6	21

orb09 (10 x 10)

0	36	1	96	2	86	3	7	4	20	5	9	6	39	7	79	8	82	9	24
0	16	8	95	3	67	5	63	2	87	1	24	9	62	6	49	7	92	4	16
0	65	6	71	1	9	3	67	8	70	2	48	5	49	4	66	7	5	9	96
0	50	1	31	4	6	8	13	5	98	7	97	6	93	9	30	3	34	2	83
0	99	5	7	9	55	8	78	7	68	4	81	3	90	2	75	1	66	6	40
1	42	5	11	0	5	9	39	2	10	8	30	3	39	6	50	7	20	4	51
0	38	3	68	8	86	7	77	9	32	2	89	4	37	6	53	1	43	5	89
0	19	9	11	4	37	1	41	5	72	7	7	2	52	6	31	3	68	8	10
0	83	1	21	9	23	7	87	8	58	6	89	4	74	5	29	2	74	3	23
0	44	2	57	4	69	9	50	1	65	5	69	8	60	3	58	6	89	7	13

orb10 (10 x 10)

9	66	8	13	0	93	7	91	6	14	5	70	3	99	2	53	4	86	1	16
8	34	9	99	0	62	7	65	5	62	4	64	6	21	2	12	3	9	1	75
9	12	8	26	7	64	6	92	4	67	5	28	3	66	2	83	1	38	0	58
0	77	1	73	3	82	2	75	6	84	4	19	5	18	7	89	8	8	9	73
0	34	1	74	7	48	5	44	4	92	6	40	3	60	2	62	8	22	9	67
9	8	8	85	3	58	7	97	5	92	4	89	6	75	2	77	1	95	0	5
8	52	9	43	6	5	7	78	5	12	3	62	4	21	2	80	1	60	0	31
9	81	8	23	7	23	6	75	4	78	5	56	3	51	2	39	1	53	0	96
9	79	8	55	2	88	4	21	5	83	3	93	6	47	7	10	0	63	1	14
0	43	1	63	2	83	3	29	4	52	5	98	6	54	7	39	8	33	9	23

- 4 problems used by Yamada and Nakano [1992], referred as yn01 – yn04

yn01 (20 x 20)

```
 17 13  2 26 11 35  4 45 12 29 13 21  7 40  0 45  3 16 15 10 18 49 10
43 14 25  8 25  1 40  6 16 19 43  5 48  9 36 16 11
  8 21  6 22 14 15  5 28 10 10  2 46 11 19 19 13 13 18 18 14  3 11  4
21 16 30  1 29  0 16 15 41 17 40 12 38  7 28  9 39
  4 39  3 28  8 32 17 46  0 35 14 14  1 44 10 20 13 12  6 23 18 22  9
15 11 35  7 27 16 26  5 27 15 23  2 27 12 31 19 31
  4 31 10 24  3 34  6 44 18 43 12 32  2 35 15 34 19 21  7 46 13 15  5
10  9 24 14 37 17 38  1 41  8 34  0 32 16 11 11 36
 19 45  1 23  5 34  9 23  7 41 16 10 11 40 12 46 14 27  8 13  4 20  2
40 15 28 13 44 17 34 18 21 10 27  0 12  6 37  3 30
 13 48  2 34  3 22  7 14 12 22 14 10  8 45 19 38  6 32 16 38 11 16  4
20  0 12  5 40  9 33 17 35  1 32 10 15 15 31 18 49
  9 19  5 33 18 32 16 37 12 28  3 16  2 40 10 37  4 10 11 20  1 17 17
48  6 44 13 29 14 44 15 48  8 21  0 31  7 36 19 43
  9 20  6 43  1 13  5 22  2 33  7 28 16 39 12 16 13 34 17 20 10 47 18
43 19 44  8 29 15 22  4 14 11 28 14 44  0 33  3 28
  7 14 12 40  8 19  0 49 13 11 10 13  9 47 18 22  2 27 17 26  3 47  5
37  6 19 15 43 14 41  1 34 11 21  4 30 19 32 16 45
 16 32  7 22 15 30  6 18 18 41 19 34  9 22 11 11 17 29 10 37  4 30  2
25  1 27  0 31 14 16 13 20  3 26 12 14  5 24  8 43
 18 22 17 22 12 30 15 31 13 15  4 13 16 47 19 18  6 33  3 30  7 46  2
48 11 42  0 18  1 16  8 25 10 43  5 21  9 27 14 14
  5 48  1 39  2 21 18 18 13 20  0 28 15 20  8 36  6 24  9 35  7 22 19
36  3 39 14 34  4 49 17 36 11 38 10 46 12 44 16 13
 14 26  1 32  2 11 15 10  9 41 13 10  6 26 19 26 12 13 11 35  5 22  0
11  7 24 17 33  8 11 10 34 16 11  3 22  4 12 18 17
 16 39 10 24 17 43 14 28  3 49 15 34 18 46 13 29  6 31 11 40  7 24  1
47  9 15  2 26  8 40 12 46  5 18 19 16  4 14  0 21
 11 41 19 26 16 14  3 47  0 49  5 16 17 31  9 43 15 20 10 25 14 10 13
49  8 32  6 36  7 19  4 23  2 20 18 15 12 34  1 33
 11 37  5 48 10 31  7 42  2 24  1 13  9 30 15 24  0 19 13 34 19 35  8
42  3 10 14 40  4 39  6 42 12 38 16 12 18 27 17 40
 14 19  1 27  8 39 12 41  5 45 11 40 10 46  6 48  7 37  3 30 17 31  4
16 18 29 15 44  0 41 16 35 13 47  9 21  2 10 19 48
 18 38  0 27 13 32  9 30  7 17 14 21  1 14  4 37 17 15 16 31  5 27 10
25 15 41 11 48  3 48  6 36  2 30 12 45  8 26 19 17
  1 17 10 40  9 16  5 36  4 34 16 47 19 14  0 24 18 10  6 14 13 14  3
30 12 23  2 37 17 11 11 23  8 40 15 15 14 10  7 46
 14 37 10 28 13 13  0 28  2 18  1 43 16 46  8 39  3 30 12 15 11 38 17
38 18 45 19 44  9 16 15 29  5 33  6 20  7 35  4 34
```

yn02 (20 x 20)

```
17 15  2 28 11 10  4 46 12 19 13 13  7 18  0 14  3 11 15 21 18 30 10
29 14 16  8 41  1 40  6 38 19 28  5 39  9 39 16 28
  8 32  6 46 14 35  5 14 10 44  2 20 11 12 19 23 13 22 18 15  3 35  4
27 16 26  1 27  0 23 15 27 17 31 12 31  7 31  9 24
  4 34  3 44  8 43 17 32  0 35 14 34  1 21 10 46 13 15  6 10 18 24  9
37 11 38  7 41 16 34  5 32 15 11  2 36 12 45 19 23
  4 34 10 23  3 41  6 10 18 40 12 46  2 27 15 13 19 20  7 40 13 28  5
44  9 34 14 21 17 27  1 12  8 37  0 30 16 48 11 34
 19 22  1 14  5 22  9 10  7 45 16 38 11 32 12 38 14 16  8 20  4 12  2
40 15 33 13 35 17 32 18 15 10 31  0 49  6 19  3 33
 13 32  2 37  3 28  7 16 12 40 14 37  8 10 19 20  6 17 16 48 11 44  4
29  0 44  5 48  9 21 17 31  1 36 10 43 15 20 18 43
```

```
  9 13  5 22 18 33 16 28 12 39  3 16  2 34 10 20  4 47 11 43  1 44 17
29  6 22 13 14 14 28 15 44  8 33  0 28  7 14 19 40
  9 19  6 49  1 11  5 13  2 47  7 22 16 27 12 26 13 47 17 37 10 19 18
43 19 41  8 34 15 21  4 30 11 32 14 45  0 32  3 22
  7 30 12 18  8 41  0 34 13 22 10 11  9 29 18 37  2 30 17 25  3 27  5
31  6 16 15 20 14 26  1 14 11 24  4 43 19 22 16 22
 16 30  7 31 15 15  6 13 18 47 19 18  9 33 11 30 17 46  4 48 10 42  2
18  1 16  0 25 14 43 13 21  3 27 12 14  5 48  8 39
 18 21 17 18 12 20 15 28 13 20  4 36 16 24 19 35  7 22  3 36  6 39 10
34 11 49  0 36  1 38  8 46  9 44  5 13  2 26 14 32
  9 11  1 10  2 41 11 10 13 26  0 26 12 13 10 35  6 22  5 11  7 24 19
33  3 11 14 34 17 11  4 22 18 12  8 17 15 39 16 24
  1 43 15 28  2 49 14 34  4 46 12 29 18 31 19 40 13 24 11 47  5 15  0
26  7 40 17 46  8 18 10 16 16 14  3 21  9 41  6 26
 16 14  6 47 17 49 10 16  3 31 12 43  4 20  8 25 14 10 18 49  7 32  0
36  9 19  2 23 15 20  5 15 13 34 19 33 11 37  1 48
  4 31 11 42  7 24  6 13  0 30 14 24 17 19 19 34 16 35 10 42 15 10 13
40  2 39  8 42  5 38  9 12  1 27 18 40 12 19  3 27
  6 39  5 41 13 45 15 40  2 46  9 48  7 37  0 30  1 31 12 16 19 29 14
44  3 41  8 35 10 47 11 21  4 10 16 48 18 38 17 27
 16 32  1 30  8 17 18 21  0 14 17 37 10 15 12 31  7 27  3 25  5 41  4
48 13 48  6 36  2 30 15 45 11 26  9 17 14 17 19 40
 18 16 17 36  4 34  2 47 10 14 15 24  1 10  3 14  7 14 12 30  5 23  9
37  8 11 14 23 11 40  6 15 16 10  0 46 13 37 19 28
 17 13 13 28 11 18 16 43  7 46  8 39  3 30  5 15  4 38  2 38 14 45  0
44 10 16  6 29 12 33  1 20 19 35 15 34  9 16 18 40
 17 14  2 30  0 27 15 47 18 43  3 17 14 13  6 43  7 45 12 32 13 13 16
48  1 10  4 14 10 42  9 38  5 43 19 22 11 43  8 23
```

yn03 (20 x 20)

```
13 47 16 21 17 27  8 46  1 27 14 39 19 24  4 34  7 27  3 36  6 11  5
32  0 13  9 40  2 40 15 20 18 45 10 23 12 36 11 31
  1 40 11 20 12 27  6 32 16 26 13 36 10 37  7 26  3 22  4 44 18 18  2
11 17 15  9 27 15 39  5 25  8 16 14 13  0 49 19 25
  9 40  8 11 14 47  2 35 13 41  7 37  1 37 18 28  6 42  3 23 10 41  5
33 17 25  0 19 19 15 16 42 12 37 11 34  4 10 15 41
  2 28  4 18 11 42  5 26 13 27  6 24 12 41  0 25  1 27  7 40 17 40 14
49 10 33  3 30 15 34 16 17  8 49  9 21 18 35 19 42
  7 26  9 27  4 25  3 42 19 28 15 22 17 34  0 15  6 46  1 34 12 47  2
16 16 34 10 31 14 24  5 43 13 45 11 47  8 18 18 15
  4 30  8 48  1 46 15 13  9 20  7 31 14 20  2 20 16 34 19 38 18 12 17
11 11 47  5 19  0 35 13 17 10 23 12 11  3 22  6 11
  3 27  2 11  5 17  0 43  1 25 15 24 18 36  8 12  9 21 13 44 10 17 17
41 16 34 11 14 12 45  7 45 14 27  6 47  4 47 19 11
  5 27  4 41 17 44 16 16 11 42 10 29  3 23  2 15  0 22 13 28  7 16 14
39  9 21 12 15 18 32 15 36  1 29  8 18  6 39 19 33
  4 44 19 38 11 24 17 21 13 34 15 11 10 16  8 43 16 41  7 45  3 37  9
10  6 36 18 31  2 17 14 28 12 43  0 22  1 25  5 15
  7 40 15 23  4 37  2 12  8 28 12 19 10 30 17 40 13 20 18 11  5 23 16
46  3 40  1 37 14 17  0 16 11 31  6 15  9 10 19 22
  5 10  1 37 15 22  2 28  6 10  9 21 19 38 16 35  7 34  0 13 14 33 11
16  4 26  3 20 17 10 18 37 13 21  8 31 10 27 12 23
 16 32  6 32  7 20  1 14  0 11 19 27  3 21 18 32 10 33 13 13 17 36  8
25  4 32  5 41 15 44  2 32 14 12  9 32 12 10 11 28
  7 28  9 33 11 35 17 44  4 43 16 35 12 31  2 14  6 48  8 40 15 28  0
31  3 22  5 30 13 27 10 24 18 47 14 38  1 46 19 22
 12 33  6 33 14 38  9 15 10 16 13 24  1 30  8 18  7 46  2 30 17 37 11
24  5 13  3 14 18 11 16 38  0 31  4 24 19 42 15 30
```

```
 10 15 16 12  6 43 18 27  0 24  9 20  3 41  2 22 12 41 11 30  5 26  4
24  7 45 13 46 14 22 15 11  8 20  1 42 19 11 17 49
  4 14 19 30 17 15  7 17  8 34  2 48  3 45 14 16 12 23 16 29 13 28  6
28 18 24 10 21  5 37  1 38 11 31  0 29  9 42 15 22
 15 41 17 19  5 37  7 36  8 47 12 49 11 29  6 18  9 33 10 30  0 49 16
37  3 11  2 46 14 36 18 35 13 45  1 31  4 33 19 18
  9 42  4 11 15 28 18 48  6 22  8 15  1 37 11 36  3 26 19 21  2 48 16
17 12 30 10 27 13 35 17 20  0 18  7 14 14 20  5 41
 19 35 17 19 16 20 15 36  1 15  3 46  4 13  8 42 18 19  5 37  2 10 13
44 10 30 11 20 14 42  6 35  0 26  9 29  7 21 12 42
 17 33  3 11  7 42 16 45  9 29  0 27  5 15 13 37  2 32 11 25 14 21  8
49 19 34  1 31 15 35  6 32  4 20 18 30 10 24 12 29
```

yn04 (20 x 20)

```
16 34 17 38  0 21  6 15 15 42  8 17  7 41 18 10 10 26 11 24  1 31 19
25 14 31 13 33  4 35  9 30  3 16 12 16  5 30  2 13
  5 41 11 33  6 15 16 38  0 40 14 38  3 37  1 20 13 22  4 34  7 16 17
39  9 15  2 19 10 36 12 39 18 26  8 19 15 39 19 34
 17 34  1 12 16 10  7 47 13 28 15 27  0 19  6 34 19 33 12 40  9 37 14
24  8 15 10 34  2 44  3 37 18 22 11 31  4 39  5 26
  5 48  7 46 16 47 10 45 14 15  8 25  0 34  3 24 12 35 18 15  2 48 13
19 11 10  1 48 17 16 15 28  4 18  6 17  9 44 19 41
 12 47  3 23  9 48 16 45 14 39  6 42  8 32 15 11 13 16  5 14 11 19  1
46 19 10 10 17  7 41  2 47 17 32  4 17  0 21 18 17
 18 14 16 20  1 18 12 14 13 10  6 16  5 24  4 18  0 24 11 18 15 42 19
13  3 23 14 40  9 48  8 12  2 24 10 23  7 45 17 30
  0 27 12 15  4 26 13 19 17 14  5 49  7 16 18 28 16 16  8 20  9 36  2
21 14 30  3 36  1 17 15 22  6 43 11 32 10 23 19 17
  0 32 16 15 17 12  7 46  3 37 18 43 11 40 13 43  9 48  4 36 15 24  8
25  1 33 14 32  5 26  6 37 12 24 10 24  2 15 19 22
 10 34  6 33 15 25  8 46  0 20 18 33  4 19 13 45  2 47  1 32  3 12 11
29 16 29  5 46 12 17  7 48 14 39 17 40 19 41  9 37
 13 26  3 47  5 44  6 49  1 22 17 12 10 28 19 36  9 27  4 25 14 48  7
11 16 49 12 24 11 48  2 19  0 47 18 49  8 46 15 36
 13 23 18 48 14 15  0 42  3 36  8 15  6 32 10 18  1 45 15 23 11 45  2
13 17 21 12 32  7 44  5 25 19 34 16 22  9 11  4 43
 17 37  7 49 15 45  2 28  9 15  8 35 12 29 13 44  1 26  4 25  5 30  3
39  0 15 14 28 18 23  6 42 11 33 16 45 10 10 19 20
  0 10  6 37  3 15 13 13 10 11  2 49  1 28 14 28 15 13  8 29 12 21 16
32 11 21  4 48  5 11 17 26  9 33 18 22  7 21 19 49
 18 38  0 41  4 30 13 43  6 11  2 43 14 27  3 26  9 30 15 19 16 36  1
31 17 47  5 41 10 34  8 40 12 32  7 13 11 18 19 27
  6 24  5 30  7 10 10 35  8 28 16 43 19 12  9 44 15 15  3 15  2 35 18
43  0 38  4 16  1 29 17 40 14 49 13 38 12 16 11 30
  3 48  6 35 13 43  2 37 17 18  5 27  9 27  7 41  1 22 15 28 16 18 10
37 18 48  4 10  8 14 11 18 14 43  0 48 12 12 19 49
  0 13 13 38  7 34  6 42  1 36  5 45 18 24  8 35 14 26 19 30 12 47 16
24 11 47  4 40 10 43  3 16 15 10  2 12  9 39 17 22
 16 30 13 47 19 49  8 20  4 40  3 46 17 21 14 33  6 44  7 23  9 24  0
48 10 43 15 41  2 32  5 29 11 36  1 38 12 47 18 12
 13 10  5 36 12 18 16 48  0 27 14 43 10 46  6 27  7 46 19 35 11 31  2
18  8 24  3 23 17 29 18 14  9 19  1 40 15 38  4 13
  9 45 16 44  0 43 17 31 14 35 13 17 12 42  3 14 18 37 10 39  6 48  7
38 15 26  4 49  2 28 11 35  1 42  5 24  8 44 19 38
```

- 5 problems formulated by Adams et al. [1988], referred as abz05 – abz09

abz05 (10 x 10)

```
 4 88 8 68 6 94 5 99 1 67 2 89 9 77 7 99 0 86 3 92
 5 72 3 50 6 69 4 75 2 94 8 66 0 92 1 82 7 94 9 63
 9 83 8 61 0 83 1 65 6 64 5 85 7 78 4 85 2 55 3 77
 7 94 2 68 1 61 4 99 3 54 6 75 5 66 0 76 9 63 8 67
 3 69 4 88 9 82 8 95 0 99 2 67 6 95 5 68 7 67 1 86
 1 99 4 81 5 64 6 66 8 80 2 80 7 69 9 62 3 79 0 88
 7 50 1 86 4 97 3 96 0 95 8 97 2 66 5 99 6 52 9 71
 4 98 6 73 3 82 2 51 1 71 5 94 7 85 0 62 8 95 9 79
 0 94 6 71 3 81 7 85 1 66 2 90 4 76 5 58 8 93 9 97
 3 50 0 59 1 82 8 67 7 56 9 96 6 58 4 81 5 59 2 96
```

abz06 (10 x 10)

```
 7 62 8 24 5 25 3 84 4 47 6 38 2 82 0 93 9 24 1 66
 5 47 2 97 8 92 9 22 1 93 4 29 7 56 3 80 0 78 6 67
 1 45 7 46 6 22 2 26 9 38 0 69 4 40 3 33 8 75 5 96
 4 85 8 76 5 68 9 88 3 36 6 75 2 56 1 35 0 77 7 85
 8 60 9 20 7 25 3 63 4 81 0 52 1 30 5 98 6 54 2 86
 3 87 9 73 5 51 2 95 4 65 1 86 6 22 8 58 0 80 7 65
 5 81 2 53 7 57 6 71 9 81 0 43 4 26 8 54 3 58 1 69
 4 20 6 86 5 21 8 79 9 62 2 34 0 27 1 81 7 30 3 46
 9 68 6 66 5 98 8 86 7 66 0 56 3 82 1 95 4 47 2 78
 0 30 3 50 7 34 2 58 1 77 5 34 8 84 4 40 9 46 6 44
```

abz07 (20 x 15)

```
  2 24  3 12  9 17  4 27  0 21  6 25  8 27  7 26  1 30  5 31 11 18 14
16 13 39 10 19 12 26
  6 30  3 15 12 20 11 19  1 24 13 15 10 28  2 36  5 26  7 15  0 11  8
23 14 20  9 26  4 28
  6 35  0 22 13 23  7 32  2 20  3 12 12 19 10 23  9 17  1 14  5 16 11
29  8 16  4 22 14 22
  9 20  6 29  1 19  7 14 12 33  4 30  0 32  5 21 11 29 10 24 14 25  2
29  3 13  8 20 13 18
 11 23 13 20  1 28  6 32  7 16  5 18  8 24  9 23  3 24 10 34  2 24  0
24 14 28 12 15  4 18
  8 24 11 19 14 21  1 33  7 34  6 35  5 40 10 36  3 23  2 26  4 15  9
28 13 38 12 13  0 25
 13 27  3 30  6 21  8 19 12 12  4 27  2 39  9 13 14 12  5 36 10 21 11
17  1 29  0 17  7 33
  5 27  4 19  6 29  9 20  3 21 10 40  8 14 14 39 13 39  2 27  1 36 12
12 11 37  7 22  0 13
 13 32 11 29  8 24  3 27  5 40  4 21  9 26  0 27 14 27  6 16  2 21 10
13  7 28 12 28  1 32
 12 35  1 11  5 39 14 18  7 23  0 34  3 24 13 11  8 30 11 31  4 15 10
15  2 28  9 26  6 33
 10 28  5 37 12 29  1 31  7 25  8 13 14 14  4 20  3 27  9 25 13 31 11
14  6 25  2 39  0 36
  0 22 11 25  5 28 13 35  4 31  8 21  9 20 14 19  2 29  7 32 10 18  1
18  3 11 12 17  6 15
 12 39  5 32  2 36  8 14  3 28 13 37  0 38  6 20  7 19 11 12 14 22  1
36  4 15  9 32 10 16
  8 28  1 29 14 40 12 23  4 34  5 33  6 27 10 17  0 20  7 28 11 21  2
```

```
21 13 20  9 33  3 27
  9 21 14 34  3 30 12 38  0 11 11 16  2 14  5 14  1 34  8 33  4 23 13
40 10 12  6 23  7 27
  9 13 14 40  7 36  4 17  0 13  5 33  8 25 13 24 10 23  3 36  2 29  1
18 11 13  6 33 12 13
  3 25  5 15  2 28 12 40  7 39  1 31  8 35  6 31 11 36  4 12 10 33 14
19  9 16 13 27  0 21
 12 22 10 14  0 12  2 20  5 12  1 18 11 17  8 39 14 31  3 31  7 32  9
20 13 29  4 13  6 26
  5 18 10 30  7 38 14 22 13 15 11 20  9 16  3 17  1 12  2 13 12 40  6
17  8 30  4 38  0 13
  9 31  8 39 12 27  1 14  5 33  3 31 11 22 13 36  0 16  7 11 14 14  4
29  6 28  2 22 10 17
```

abz08 (20 x 15)

```
  0 19  9 33  2 32 13 18 10 39  8 34  6 25  4 36 11 40 12 33  1 31 14
30  3 34  5 26  7 13
  9 11 10 22 14 19  5 12  4 25  6 38  0 29  7 39 13 19 11 22  1 23  3
20  2 40 12 19  8 26
  3 25  8 17 11 24 13 40 10 32 14 16  5 39  9 19  0 24  1 39  4 17  2
35  7 38  6 20 12 31
 14 22  3 36  2 34 12 17  4 30 13 12  1 13  6 25  9 12  7 18 10 31  0
39  5 40  8 26 11 37
 12 32 14 15  1 35  7 13  8 32 11 23  6 22  4 21  0 38  2 38  3 40 10
31  5 11 13 37  9 16
 10 23 12 38  8 11 14 27  9 11  6 25  5 14  4 12  2 27 11 26  7 29  3
28 13 21  0 20  1 30
  6 39  8 38  0 15 12 27 10 22  9 27  2 32  4 40  3 12 13 20 14 21 11
22  5 17  7 38  1 27
 11 11 13 24 10 38  8 15  9 19 14 13  5 30  0 26  2 29  6 33 12 21  1
15  3 21  4 28  7 33
  8 20  6 17  5 26  3 34  9 23  0 16  2 18  4 35 12 24 10 16 11 26  7
12 14 13 13 27  1 19
  1 18  7 37 14 27  9 40  5 40  6 17  8 22  3 17 10 30  0 38  4 21 12
32 11 24 13 24  2 30
 11 19  0 22 13 36  6 18  5 22  3 17 14 35 10 34  7 23  8 19  2 29  1
22 12 17  4 33  9 39
  6 32  3 22 12 24  5 13  4 13  1 11  0 11 13 25  8 13  2 15 10 33 11
17 14 16  9 38  7 24
 14 16 13 16  1 37  8 25  2 26  3 11  9 34  4 14  0 20  6 36 12 12  5
29 10 25  7 32 11 12
  8 20 10 24 11 27  9 38  5 34 12 39  7 33  4 37  2 31 13 15 14 34  3
33  6 26  1 36  0 14
  8 31  0 17  9 13  1 21 10 17  7 19 13 14  3 40  5 32 11 25  2 34 14
23  6 13 12 40  4 26
  8 38 12 17  3 14 13 17  4 12  1 35  6 35  0 19 10 36  7 19  9 29  2
31  5 26 11 35 14 37
 14 20  3 16  0 33 10 14  5 27  7 31  8 16  6 31 12 28  9 37  4 37  2
29 11 38  1 30 13 36
 11 18  3 37 14 16  6 15  8 14 12 11 13 32  5 12  1 11 10 29  7 19  4
12  9 18  2 26  0 39
 11 11  2 11 12 22  9 35 14 20  7 31  4 19  3 39  5 28  6 33 10 34  1
38  0 20 13 17  8 28
  2 12 12 25  5 23  8 21  6 27  9 30 14 23 11 39  3 26 13 34  7 17  1
24  4 12  0 19 10 36
```

abz09 (20 x 15)

```
  6 14  5 21  8 13  4 11  1 11 14 35 13 20 11 17 10 18 12 11  2 23  3
13  0 15  7 11  9 35
  1 35  5 31  0 13  3 26  6 14  9 17  7 38 12 20 10 19 13 12  8 16  4
34 11 15 14 12  2 14
  0 30  4 35  2 40 10 35  6 30 14 23  8 29 13 37  7 38  3 40  9 26 12
11  1 40 11 36  5 17
  7 40  5 18  4 12  8 23  0 23  9 14 13 16 12 14 10 23  3 12  6 16 14
32  1 40 11 25  2 29
  2 35  3 15 12 31 11 28  6 32  4 30 10 27  7 29  0 38 13 11  1 23 14
17  5 27  9 37  8 29
  5 33  3 33  6 19 12 40 10 19  0 33 13 26  2 31 11 28  7 36  4 38  1
21 14 25  9 40  8 35
 13 25  0 32 11 33 12 18  4 32  6 28  5 15  3 35  9 14  2 34  7 23 10
32  1 17 14 26  8 19
  2 16 12 33  9 34 11 30 13 40  8 12 14 26  5 26  6 15  3 21  1 40  4
32  0 14  7 30 10 35
  2 17 10 16 14 20  6 24  8 26  3 36 12 22  0 14 13 11  9 20  7 23  1
29 11 23  4 15  5 40
  4 27  9 37  3 40 11 14 13 25  7 30  0 34  2 11  5 15 12 32  1 36 10
12 14 28  8 31  6 23
 13 25  0 22  3 27  8 14  5 25  6 20 14 18  7 14  1 19  2 17  4 27  9
22 12 22 11 27 10 21
 14 34 10 15  0 22  3 29 13 34  6 40  7 17  2 32 12 20  5 39  4 31 11
16  1 37  8 33  9 13
  6 12 12 27  4 17  2 24  8 11  5 19 14 11  3 17  9 25  1 11 11 31 13
33  7 31 10 12  0 22
  5 22 14 15  0 16  8 32  7 20  4 22  9 11 13 19  1 30 12 33  6 29 11
18  3 34 10 32  2 18
  5 27  3 26 10 28  6 37  4 18 12 12 11 11 13 26  7 27  9 40 14 19  1
24  2 18  0 12  8 34
  8 15  5 28  9 25  6 32  1 13  7 38 11 11  2 34  4 25  0 20 10 32  3
23 12 14 14 16 13 20
  1 15  4 13  8 37  3 14 10 22  5 24 12 26  7 22  9 34 14 22 11 19 13
32  0 29  2 13  6 35
  7 36  5 33 13 28  9 20 10 30  4 33 14 29  0 34  3 22 11 12  6 30  8
12  1 35  2 13 12 35
 14 26 11 31  5 35  2 38 13 19 10 35  4 27  8 29  3 39  9 13  6 14  7
26  0 17  1 22 12 15
  1 36  7 34 11 33  8 17 14 38  6 39  5 16  3 27 13 29  2 16  0 16  4
19  9 40 12 35 10 39
```