INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

**SCHEDULING FLEXIBLE FLOW LINES
WITH SEQUENCE DEPENDENT SETUP TIMES**

**By
Mary Elizabeth Kurz**

Copyright © Mary Elizabeth Kurz 2001

**A Dissertation Submitted to the Faculty of the
DEPARTMENT OF SYSTEMS AND INDUSTRIAL ENGINEERING
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
In the Graduate College
THE UNIVERSITY OF ARIZONA**

2001

UMI Number: 3023515

**Copyright 2001 by
Kurz, Mary Elizabeth**

All rights reserved.

UMI[®]

UMI Microform 3023515

Copyright 2001 by Bell & Howell Information and Learning Company.

**All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.**

**Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346**

THE UNIVERSITY OF ARIZONA @
GRADUATE COLLEGE

As members of the Final Examination Committee, we certify that we have
read the dissertation prepared by Mary Elizabeth Kurz
entitled SCHEDULING FLEXIBLE FLOW LINES WITH SEQUENCE DEPENDENT
SETUP TIMES

and recommend that it be accepted as fulfilling the dissertation
requirement for the Degree of DOCTOR OF PHILOSOPHY

<u>7/25/01</u>
Ronald Askin	Date
<u>7/25/01</u>
Jeffrey Goldberg	Date
<u>7/21/01</u>
Pitu Mirchandani	Date
<u>7/25/01</u>
John Leonard	Date
	Date

Final approval and acceptance of this dissertation is contingent upon
the candidate's submission of the final copy of the dissertation to the
Graduate College.

I hereby certify that I have read this dissertation prepared under my
direction and recommend that it be accepted as fulfilling the dissertation
requirement.

<u>7/25/01</u>
Dissertation Director	Date
Ronald Askin	

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: _____

ACKNOWLEDGEMENTS

Throughout the time I have spent on this dissertation, I have been assisted enormously by many people. I'd like to acknowledge their help and support at this time. First of all, my husband, Matthew Kurz, supported me in numerous ways. My advisor, Dr. Ronald Askin, provided invaluable advice and guidance regarding this work, as well in the areas of professional development, employment and life in general. Drs. Jeffrey Goldberg, Pitu Mirchandani, and John Leonard provided very useful comments on earlier drafts of this dissertation and during its defense, greatly improving its readability and sense. Without Dr. Cole Smith's help with CPLEX, the example problem in Chapter 4 would not have been solved to optimality. Bill Ganoe's computer support was essential in the solution of this problem as well. Though not directly involved in supporting my research, the students in my SIE 305 classes and the office staff provided me with encouragement and moral support. All these people deserve my gratitude and thanks.

DEDICATION

TO

CADY SAOIRSE KURZ

Born January 2, 2001

TABLE OF CONTENTS

LIST OF FIGURES	9
LIST OF TABLES	10
ABSTRACT	12
1 INTRODUCTION	14
2 LITERATURE REVIEW	21
2.1 Single Machine Scheduling	23
2.2 Parallel Machines	28
2.3 Flow Lines	34
2.4 Hybrid Flow Lines	43
2.5 Flexible Flow Lines	48
2.6 Some Traveling Salesman Problem Solution Methods	52
2.6.1 Graph Theory Background	53
2.6.1.1 Finding a Minimum Spanning Tree	55
2.6.1.2 Finding an Eulerian Trail	56
2.6.1.3 Finding a Hamiltonian Cycle	56
2.6.2 Minimum Spanning Tree Heuristics	57
2.6.3 Insertion Heuristics	58
2.7 Genetic Algorithms	58
3 PARALLEL MACHINE SCHEDULING	64
3.1 Integer Programming Model	64
3.2 Heuristics with No Ready Times	67
3.2.1 Slicing Heuristic (SL)	67
3.2.2 Multiple MULTI-FIT Heuristic (MMF)	69
3.2.3 Multiple Insertion Heuristic (MI)	70
3.2.4 Genetic Algorithm (GA)	71
3.2.5 Discussion of Heuristics	75
3.3 Heuristics with Ready Times	75
3.4 Experimental Design	75
3.4.1 Ready Times	76
3.4.2 Setup Times	77
3.4.3 Processing Times	77
3.4.4 Place-holder Jobs	78
3.4.5 Problem Data Characterization	78
3.5 Identifying Bounds	79
3.5.1 Lower Bounds	79
3.5.2 Scheduling Computational Complexity	80
3.5.3 Symmetric Matrix Ceiling	82
3.6 Examples	83
3.6.1 SL Example	84
3.6.2 MMF Example	88

TABLE OF CONTENTS - *Continued*

3.6.3	MI Example.....	89
3.6.4	GA Example.....	91
3.7	Comparing Heuristics without Ready Times.....	91
3.7.1	Comparison to Data-dependent Lower Bounds	96
3.7.2	Comparison to Optimal.....	102
3.7.3	Running Time Performance	104
3.8	Comparing Heuristics with Ready Times.....	106
3.8.1	Comparison to Data-dependent Lower Bounds	106
3.8.2	Comparison to Optimal.....	110
3.9	Conclusions.....	112
4	MULTIPLE STAGES.....	114
4.1	Integer Programming Model.....	114
4.2	Permutation Schedules	118
4.2.1	Counterexample 1: No Sequence-dependent Setup Times.....	120
4.2.2	Counterexample 2: Sequence-dependent Setup Times at First Stage	121
4.2.3	Counterexample 3: Sequence-dependent Setup Times at Second Stage....	122
4.2.4	Counterexample 4: Sequence-dependent Setup Times at Each Stage	123
4.3	Lower Bounds.....	124
4.4	Heuristics.....	126
4.4.1	Cyclic Heuristics	127
4.4.1.1	Basic Cyclic Heuristic	127
4.4.1.2	Readymtime Cyclic Heuristic	128
4.4.1.3	SPT Cyclic	128
4.4.2	Multiple Insertion Heuristics	129
4.4.2.1	Flowtime Multiple Insertion Heuristic	129
4.4.2.2	Completion Time Multiple Insertion Heuristic	130
4.4.2.3	Makespan Multiple Insertion Heuristic	131
4.4.3	Johnson's Rule Based Heuristics	131
4.4.3.1	The 1, g Johnson's Rule.....	131
4.4.3.2	The $g/2$, $g/2$ Johnson's Rule.....	132
4.4.4	Genetic Algorithms	133
4.4.4.1	Genetic Algorithm.....	133
4.4.4.2	Random Keys Genetic Algorithm	134
4.5	Examples	135
4.5.1	CH	137
4.5.2	RCH.....	138
4.5.3	SPTCH.....	139
4.5.4	FTMIH.....	141
4.5.5	CTMIH	142
4.5.6	MMIH.....	143
4.5.7	The 1, g Johnson's Rule.....	144
4.5.8	The $g/2$, $g/2$ Johnson's Rule.....	145

TABLE OF CONTENTS - *Continued*

4.5.9	GA	146
4.5.10	RKGA.....	147
4.6	Generation of Test Data	148
4.7	Experimental Results.....	151
4.7.1	Best Heuristic in Each Class	151
4.7.1.1	Cyclic Heuristics	151
4.7.1.2	Multiple Insertion Heuristics.....	153
4.7.1.3	Johnson's Rule Based Heuristics	155
4.7.1.4	Genetic Algorithms.....	157
4.7.2	Comparing Representative Heuristics	162
4.7.3	Quality of the Lower Bounds.....	164
4.7.4	Significance of the Experimental Factors.....	165
4.8	Conclusions.....	168
5	MULTISTAGE EXTENSIONS	169
5.1	Forward and Backward Pass Method.....	169
5.1.1	Condition	170
5.1.2	Theorem.....	171
5.1.3	Backward Pass Heuristic	171
5.2	Bottleneck Methods	173
5.2.1	Bottleneck SPTCH (BNSPTCH)	173
5.2.2	Bottleneck FTMIH (BNFTMIH)	177
5.2.3	Comparing SPTCH and BNSPTCH.....	179
5.2.4	Comparing FTMIH and BNFTMIH.....	180
5.2.5	Value of Bottleneck Knowledge	180
5.3	Conclusions.....	181
6	CONCLUSIONS AND AREAS FOR FUTURE RESEARCH	182
	APPENDIX A: ANOVA - SLICING HEURISTIC.....	187
	APPENDIX B: ANOVA - MULTIPLE MULTI-FIT HEURISTIC.....	188
	APPENDIX C: ANOVA - MULTIPLE INSERTION HEURISTIC.....	190
	APPENDIX D: ANOVA - COMPARING HEURISTICS – NO READY TIMES.....	192
	APPENDIX E: MULTISTAGE HEURISTICS – LOSS STATISTICS	194
	APPENDIX F: MULTISTAGE HEURISTICS – RUNNING TIME STATISTICS.....	224
	REFERENCES	254

LIST OF FIGURES

Figure 1: Example Problem Schematic.....	17
Figure 2: Processing and Setup Times for Example Problem.....	18
Figure 3: Graphical Representation of Feasible Schedule.....	66
Figure 4: Population Initialization.....	72
Figure 5: Chromosome Evaluation.....	73
Figure 6: Incumbent and Stopping Criteria.....	73
Figure 7: Reproduction.....	73
Figure 8: Crossover.....	74
Figure 9: Mutation.....	74
Figure 10: Sample Data Set.....	84
Figure 11: Optimal Schedule.....	84
Figure 12: Minimum Spanning Tree.....	84
Figure 13: Doubled Graph.....	85
Figure 14: Eulerian Trail Edges.....	85
Figure 15: Hamiltonian Graph.....	86
Figure 16: First SL Attempt.....	87
Figure 17: Final Schedule from SL.....	87
Figure 18: Assignment from MMF.....	89
Figure 19: Schedule from MI.....	90
Figure 20: Schedule from GA.....	91
Figure 21: Least Square Means for MST_ALGO Factor.....	93
Figure 22: Least Square Means for SETUP_FUN Factor for MMF.....	95
Figure 23: Least Square Means for ALGO Factor – No Ready Times.....	97
Figure 24: Makespan Divided by Lower Bounds with No Ready Times.....	100
Figure 25: Average Makespan Divided by Optimal – No Ready Times.....	103
Figure 26: Least Square Means for ALGO Factor – Ready Times.....	107
Figure 27: Makespan Divided by Lower Bounds with Ready Times.....	109
Figure 28: Average Makespan Divided by Optimal – Ready Times.....	112
Figure 29: Counterexample 1 Optimal Schedule.....	120
Figure 30: Counterexample 2 Optimal Schedule.....	121
Figure 31: Counterexample 3 Optimal Schedule.....	122
Figure 32: Counterexample 4 Optimal Schedule.....	124
Figure 33: Processing and Setup Times for Example Problem.....	136
Figure 34: Generation Number vs. Loss for the HHLHLHA Case.....	160
Figure 35: Generation Number vs. Loss for the HLHLLA Case.....	161
Figure 36: Processing and Setup Times for Example Problem.....	176

LIST OF TABLES

Table 1: Feasible Schedule.....	19
Table 2: Factor Levels	79
Table 3: New Job Order and Modified Processing Times.....	88
Table 4: MMF Iterations.....	88
Table 5: Partial Machine Schedules	90
Table 6: SL ANOVA – Main Effects	93
Table 7: MMF ANOVA – Main Effects.....	94
Table 8: MI ANOVA – Main Effects.....	96
Table 9: GA Parameter Values.....	96
Table 10: ANOVA – No Ready Time Heuristics.....	97
Table 11: Average MS/LB – No Ready Times	99
Table 12: Number of Times MI and GA were Lowest – No Ready Times.....	101
Table 13: Number of Times Each Heuristic Hit Optimal – No Ready Times	103
Table 14: Ratios of Achieved Makespan to Optimal Makespan – No Ready Times.....	104
Table 15: Average Running Times in Seconds – No Ready Times	105
Table 16: ANOVA - Ready Time Heuristics	106
Table 17: Average MS/LB – Ready Times.....	108
Table 18: Number of Times MI and GA were Lowest – Ready Times.....	110
Table 19: Number of Times Each Heuristic Hit Optimal – Ready Times.....	111
Table 20: Ratios of Achieved Makespan to Optimal Makespan.....	112
Table 21: Machine Numbering Example 1	119
Table 22: Machine Numbering Example 2	119
Table 23: Processing Times	120
Table 24: Permutation Schedule Makespan	120
Table 25: Setup Times at Stage 1	121
Table 26: Permutation Schedule Makespan	122
Table 27: Setup Times at Stage 2.....	122
Table 28: Permutation Schedule Makespan	123
Table 29: Setup Times at Stage 1	123
Table 30: Setup Times at Stage 2.....	123
Table 31: Permutation Schedule Makespan	124
Table 32: Optimal Schedule.....	137
Table 33: CH Schedule	138
Table 34: Order for RCH	138
Table 35: RCH Schedule.....	139
Table 36: Order for SPTCH.....	139
Table 37: Job Completion Time as Criterion for Machine Assignment Example	140
Table 38: SPTCH Schedule	140
Table 39: Order for FTMIH.....	141
Table 40: Sum of Flowtimes as Criterion for Machine Assignment Example	142

LIST OF TABLES - *Continued*

Table 41: FTMIH Schedule.....	142
Table 42: MMIH Schedule.....	143
Table 43: Order for 1, g Johnson's Rule.....	144
Table 44: 1, g Johnson's Rule Schedule	144
Table 45: Aggregated Times for $g/2$, $g/2$ Johnson's Rule	145
Table 46: Order for $g/2$, $g/2$ Johnson's Rule.....	145
Table 47: $g/2$, $g/2$ Johnson's Rule Schedule	146
Table 48: GA Schedule	147
Table 49: RKGA Schedule.....	148
Table 50: Factor Levels.....	150
Table 51: Loss Statistics for Cyclic Heuristics	152
Table 52: Paired T-Tests for Cyclic Heuristics	152
Table 53: Loss Statistics for Multiple Insertion Heuristics.....	154
Table 54: Paired T-Tests for Multiple Insertion Heuristics	155
Table 55: Multiple Insertion Heuristics Running Times	155
Table 56: Loss Statistics for Johnson's Rule Based Heuristics	156
Table 57: Paired T-Test for Johnson's Rule Based Heuristics.....	157
Table 58: GA Generation Statistics	158
Table 59: HLLLLLL Test Problem Incumbent Solutions	159
Table 60: Loss Statistics for Genetic Algorithms.....	162
Table 61: Paired T-Test for Genetic Algorithms.....	162
Table 62: Loss Statistics for Remaining Heuristics.....	163
Table 63: Paired T-Tests for Remaining Heuristics	163
Table 64: Lower Bounds.....	165
Table 65: Factor Indicators for ANOVA.....	166
Table 66: ANOVA – Multistage Heuristics.....	167
Table 67: Paired T-Test for $g/2$, $g/2$ Johnson's Rule Based Heuristic and the Backward Pass Heuristic.....	172
Table 68: Order for BNSPTCH.....	177
Table 69: BNSPTCH Schedule	177
Table 70: Paired T-Test for SPTCH and BNSPTCH	179
Table 71: Paired T-Test for FTMIH and BNFTMIH	180

ABSTRACT

This dissertation examines scheduling in flexible flow lines with sequence-dependent setup times to minimize makespan. This type of manufacturing environment is found in industries such as printed circuit board and automobile manufacture. Lower makespans can be associated with more efficient use of resources. Poor scheduling when sequence-dependent setup times exist can negatively impact productivity. As a building block, minimizing makespan in parallel identical machines with sequence-dependent setup times is examined. Several heuristics are compared empirically using statistical analysis. Experimental results indicate that a heuristic based on the Insertion Heuristic for the Travelling Salesman Problem is effective. Subsequently, minimizing makespan in flexible flow lines with sequence-dependent setup times is considered. An integer program that incorporates all aspects of the problem is formulated. Due to the NP-hard nature of the problem, heuristic methods are considered. The heuristics, based on greedy methods, flow line methods, the Insertion Heuristic for the Travelling Salesman Problem and genetic algorithms are compared empirically using statistical analysis. The heuristics are designed to take advantage of the flow line nature of the problem, the parallel machine nature and the combinatorial features of the problem. Problem data is generated in order to evaluate the heuristics. The characteristics are chosen to reflect those used by previous researchers. An effective lower bound is created in order to evaluate the heuristics. A random keys genetic algorithm is found to be very effective for the problems examined. In addition, several extensions based on backwards pass of the

schedule and focusing on the bottleneck stage are proposed and examined. These proved to be ineffective approaches but yielded insight regarding what features of schedule are important. Most significantly, the first stage is very important in determining the quality of the subsequent schedule. The heuristics considered here focus on setting a schedule for one stage and then considering the next. Areas for future research include developing methods of scheduling job-by-job and examination of branch and bound methods to find optimal solutions, aided by effective lower bounds and theorems regarding schedule domination.

1 Introduction

Traditional manufacturing systems have taken many general forms. Among these are flow lines and parallel machines. In a flow line, jobs consist of several tasks with a linear precedence relation and a single workstation performs each of the required tasks. In a parallel machine environment, jobs consist of a single task. Each machine performs the same task, thereby increasing the capacity available to process jobs. Methods for scheduling in these environments have been thoroughly researched, as shall be seen in the Literature Review in Section 2. However, in the increasingly complex manufacturing environments, more complex manufacturing systems have been created in order to address such factors as limited capacity and complicated process plans. For example, the semi-conductor industry uses re-entrant flow lines, in which multiple machines may exist at each stage and jobs revisit previous stages many times in a cyclic manner. The printed circuit board and automobile industries make use of flow lines with multiple machines at some stages and allow jobs to skip stages. Moreover, these industries encounter sequence-dependent setup times which result in even more difficult scheduling problems. The scheduling objective in such industries may vary. Due date related criteria may be important. The makespan criterion has been used by many researchers and has been selected for this research. Pinedo (1995) indicates that lower makespans usually indicate an efficient utilization of resources. Scheduling to minimize makespan in flow lines with multiple parallel machines, jobs that may skip stages, and sequence-dependent setup times is the focus of this dissertation. This kind of manufacturing environment

introduces new difficulties that scheduling in simple flow lines, for example, did not address.

At this point, we would like to define the scope of the problem considered in this research. First, we shall define the main structure of a flexible flow line. A flexible flow line consists of several stages in series. A stage is a machine or set of machines that perform the same operation. Each job may not revisit a stage that it has already visited. Each stage has at least one machine but may have more. In fact, at least one stage must have more than one machine. At this point, this structure may be considered a hybrid flow line or a flow line with multiple machines. However, the feature that makes our application a *flexible* flow line is that jobs may skip stages. This could occur in an industry in which some jobs do not require an operation. For example, in screenprinting t-shirts, some shirt and ink combinations require an undercoat of white ink so that the intended print color will appear correctly. Only dark shirts require undercoats for dark colored ink. The undercoat printing step is not required for all t-shirts; these t-shirts do not stop at the undercoating printing station.

The potential variants of the basic flexible flow line described above that can be studied are nearly limitless. Now we shall describe the particular features of this research. All data in this problem are assumed to be known deterministically when scheduling is undertaken. Machines are available at all times, with no breakdowns or scheduled or unscheduled maintenance. Jobs are always processed without error. Job processing cannot be interrupted (no preemption is allowed) and jobs have no associated priority values. Infinite buffers exist between stages and before the first stage and after

the last stage; machines cannot be blocked because the current job can always enter the appropriate buffer. There is no travel time between stages; jobs are available for processing at a stage immediately after completing processing at the previous stage. The ready time for each job is the larger of zero and the time the job completes processing on the previous stage. Machines in parallel are identical in capability and processing rate. A key characteristic of this research topic is that sequence-dependent setup times exist between jobs at each stage. After completing processing of one job and beginning processing of the next job, some sort of setup must be performed. Setup can only be performed after the machine is no longer processing any job and the job for which setup is being performed is ready. Moreover, the length of time required to do the setup depends on both the prior job which the machine processed and the current job to be processed. For example, consider a screenprinting operation again. Between printing different colors with the same screen, some cleaning of the screen may be required. But, the degree to which the screen must be cleaned may depend on the color previously on the screen and the color now required on the screen.

An example is now presented to assist in motivating this research. This example has four stages, each of which has two identical machines. Figure 1 shows a symbolic representation of this scenario. The boxes represent the stages and the ovals within each box represent the machines. The top stage is stage 1, followed by stage 2, etc. Each stage is visited by some jobs, which are listed beside each stage. The processing and setup times for this example are shown in Figure 2. Only the information for jobs that visit each stage is shown. We assume that the current setup for each machine is unknown

but that the machine is never setup for the job that we intend to process first. We use a job called 0 to serve as a nominal set point for each machine. The nominal set point could represent the printing press with no screens loaded and no shirt pallets installed, for example. The fictional job 0 requires no processing. The setup from job 0 to each job indicates the amount of time required to setup each job from this nominal set point. The notation p_i^t indicates the processing time for job i at stage t , and s_{ij}^t indicates the sequence-dependent setup time from job i to job j at stage t . The objective of this research is to find a way of ordering the jobs at each stage so that the latest completion time is minimized (makespan is minimized). Table 1 contains a feasible schedule for this example.

Figure 1: Example Problem Schematic

Stage 1					
Job	0	2	3	4	5
p_i^1	20	63	62	54	53

Stage 2					
Job	0	1	2	5	6
p_i^2	16	70	73	52	57

Stage 1				
(s_{ji}^1)	To			
From	2	3	4	5
0	3	2	0	4
2	—	15	15	19
3	23	—	15	15
4	17	22	—	20
5	18	16	20	—

Stage 2				
(s_{ji}^2)	To			
From	1	2	5	6
0	4	4	8	7
1	—	4	8	7
2	0	—	6	0
5	19	13	—	16
6	16	17	20	—

Stage 3				
Job	0	1	2	4
p_i^3	20	73	71	57

Stage 4				
Job	0	1	2	6
p_i^4	18	70	64	63

Stage 3			
(s_{ji}^3)	To		
From	1	2	4
0	3	2	3
1	—	4	5
2	5	—	6
4	20	19	—

Stage 4			
(s_{ji}^4)	To		
From	1	2	6
0	0	3	5
1	—	4	1
2	4	—	2
6	18	21	—

Figure 2: Processing and Setup Times for Example Problem

Table 1: Feasible Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	4	86	155	1	5	152	212
2	0	0	20	2	0	20	36
2	3	20	84	2	2	86	163
2	5	84	152	2	6	163	220

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	4	186	248	1	6	256	320
2	0	36	56	2	0	56	74
2	2	163	236	2	2	236	303

The main contributions of the research described in this dissertation include the following:

1. An investigation of heuristics for minimizing makespan in parallel identical machines has been performed. Makespan is latest completion time of any job on any stage. The heuristics are compared empirically. The results from this investigation inform the development of heuristics for the flexible flow line problem.
2. A lower bound that is effective for makespan in flexible flow lines with sequence-dependent setup times has been developed.
3. An integer program for the makespan in flexible flow lines with sequence-dependent setup times has been formulated. This formulation incorporates the

flexible nature of the flow line; that is, it correctly deals with jobs that need not visit all stages.

4. Several heuristics for minimizing makespan in flexible flow lines with sequence-dependent setup times are developed and compared empirically. These heuristics include approaches based on TSP methods, flow line methods, and genetic algorithms.
5. The random keys genetic algorithm has been applied to the flexible flow line with sequence-dependent setup times and found to be effective.

This dissertation continues with a review of related research and a review of basic graph theory concepts in Chapter 2. Chapter 3 addresses the single stage problem with the intention that our results will be useful in addressing the full multiple stage flexible flow line scheduling problem. In Chapter 4 we investigate the flexible flow line scheduling problem. We devise several heuristics and evaluate them, using several lower bounds to aid in the evaluation. Chapter 5 contains a few extensions to heuristics developed in Chapter 4. Chapter 6 concludes the dissertation and identifies several areas for continued research. Several appendices are included with supporting material. They will be referred to within the text as needed.

2 Literature Review

As in the queuing literature, a notation scheme exists to classify problems in scheduling (Graham *et al.* 1979; Blacewicz *et al.* 1996). The notation has three parts denoted $\alpha|\beta|\gamma$. The following summarizes some common symbols used in this notation. If additional notation is used later in this work, it will be described at that time. The first term α describes the machine environment where $\alpha = \alpha_1\alpha_2$ with

$\alpha_1 \in \{\emptyset, P, F, HF, FFS\}$, standing for machine formation, where

$\emptyset$ indicates no relation between machines (especially used for single machines)

P indicates machines in parallel

F indicates a flow line

HF indicates a hybrid flow line

FFS indicates a flexible flow line

$\alpha_2 \in \{\emptyset, n\}$, standing for number of machines, where

$\emptyset$ indicates an arbitrary, unspecific number

n indicates a specific number, such as 1, 2, etc.

The second term β describes the task and resource properties and has several possible components. Some of the common components include

r_i indicating the presence of ready times (when each job can begin processing)

d_i indicating the presence of due dates

s_{ij} indicating the presence of sequence-dependent setup times

$pmtn$ indicating that jobs may be preempted

tree indicating that a tree precedence structure exists

prec indicating that a general precedence structure exists

pmu indicating that the solution must be a permutation schedule

The third term γ indicates the performance measure of interest. Some of the common objectives include $C_{\max}$, $L_{\max}$, $\sum C_i$, $\sum T_i$, $\sum w_i C_i$, and $\sum w_i T_i$, where

C_i is the completion time of job i

$L_i = C_i - d_i$ and is the lateness of job i

$T_i = \max\{0, L_i\}$ and is the tardiness of job i

w_i is the weight assigned to job i

$C_{\max} = \max_i \{C_i\}$ and is the makespan of the schedule

$L_{\max} = \max_i \{L_i\}$

Note that this list is not intended to completely define this notation. The notation is evolving and can accommodate any elements desired. The problem of interest in this dissertation can be indicated $FFS|s_{ij}|C_{\max}$.

The organization of the rest of literature review is as follows. Single machine scheduling will be first, followed by parallel machine scheduling. Single machine scheduling is of interest because a single machine may exist at a stage. Parallel machine scheduling is of interest for the same reason. In parallel machine scheduling, each job need only be processed by one of the machines in parallel. Then, serial machines or a flow line, will be discussed. Here, all jobs must be processed by each of the stages. The hybrid flow line, where at least one of the stages has more than one machine in parallel,

is next. Similar to the flow line situation, each job must be processed by exactly one machine at each stage. A review of the literature on flexible flow lines follows. In flexible flow lines, jobs may be processed by at most one machine at each stage. Special sections discussing solution methods for Travelling Salesman Problems and genetic algorithms will conclude the literature review.

2.1 Single Machine Scheduling

Scheduling jobs on one machine has been considered for a number of years. The single machine scheduling problem is relevant because a stage in a flexible flow line may have only one machine. Generally, setup times are considered to be negligible and ready times are all zero. In these cases, the makespan is simply the sum of the processing times, regardless of the order of the jobs. The problem of scheduling jobs with setup times on one machine to minimize the makespan is well understood, as long as all jobs are available at time zero (Baker 1974). With the distances between cities representing the sequence-dependent setup times between jobs, the travelling salesman problem (TSP) solves the problem. This results in an sequencing of jobs which yields a schedule when combined with the processing times and sequence-dependent setup times. Several heuristics exist to generate solutions to a Traveling Salesman Problem. These heuristics are discussed in the Travelling Salesman Problem portion of this literature review.

The case of non-equal ready times in the single machine environment has been considered by several researchers. Non-equal ready times are of interest because the arrival time of jobs at stages in a multistage production line depends on the completion

time of those jobs in prior stages. Clearly, a job's ready time at a stage is dependent on its completion time at the previous stage. Holding "early" jobs to be processed at a stage until all have arrived will only increase the makespan, decrease utilization and otherwise diminish the effectiveness of the multistage production line. Oftentimes in the research into single machine scheduling, the inclusion of ready times is accompanied by some other criterion, such as minimization of the mean completion time, the maximum lateness, or the number of tardy jobs. Minimization of maximum lateness has been considered by McMahon and Florian (1975), who developed a branch-and-bound procedure to implicitly search the solution space and results in an optimal schedule. Leon and Wu (1992) additionally consider vacations for the single machine. Asano and Ohta, (1996) in addition to including sequence-dependent setup times, combine the constraint that no job can be completed after its due date with a minimization of earliness objective in light of the just-in-time production environment. Recently, Rosenkrantz *et al.* (2001) developed a linear time algorithm for minimizing makespan in the case where the jobs each have one of a limited number of distinct processing times and ready times, even when jobs must satisfy precedence constraints.

Delivery times are included by a number of researchers (Potts 1980; Hall and Shmoys 1992, Nowicki and Smutnicki 1994) with the accompanying goal of minimizing the time by which all jobs are delivered. A delivery is some additional processing that must occur for a deterministic amount of time immediately following the regular processing on machines with no capacity constraints or setup, or analogously, on an infinite supply of machines. Because multiple deliveries can be handled simultaneously,

we cannot just simply add the delivery time to the processing time. If delivery time for all jobs are set at zero, then this problem is a minimization of makespan in the conventional sense. The delivery time problem with ready times has been addressed by numerous researchers, including Potts(1980), Tinhofer and Farnbacher (1992), Hall and Shmoys (1992), and Nowicki and Smutnicki (1994). The translation of delivery time into due dates is explained by Potts, and due dates can also be translated back to delivery times by a similar process. Thus, the results of these papers can be applied to the problem with ready times and due dates, even if those due dates are identical. However, neither delivery nor due dates are directly relevant to the problem of interest in this work.

Sequence-dependent setup times have been considered under special circumstances. A common restriction is differentiating the setup times occurring between jobs from different part families from those occurring between jobs in the same part families. The part types are somehow, generally through *a priori* knowledge, partitioned into families such that “major” (large) setups occur between processing of parts from different families and “minor” (small) (or sometimes no) setups occur between the processing of parts within the same family. Sometimes, researchers do not explicitly note that this structure has sequence-dependent setup times. For example, Zdrzalka (1996) and Schutten, van de Velde and Zijm (1996) state that the setup times between families depend only on the family for which setup is now required. This implies that sequence is not important; however there does exist a binary dependency based on the possible switch to a new family. Zdrzalka (1996) combines these family-based setup times with ready times and delivery times to minimize the time by which all jobs are delivered, or

the makespan including delivery. Some characteristics in Zdrzalka's work that vary from the general problem addressed in this work include the assumptions that setup *can* begin before the job is ready, as long as the machine is idle, and that permutation schedules are required. Nowicki and Zdrzalka (1996) addressed a slightly different version of this problem, though with permutation schedules still required. The setup times must have an additional special structure beyond the major and minor classification; the minor setup times must satisfy the condition that changing from higher indexed jobs to lower indexed jobs within the same family incurs a positive setup time while any other changes within the same family incur no setup times. All jobs are apparently available at time zero, and a tabu search approach is developed that is appropriate for any regular cost function. Another recent contribution to this general problem by Schutten, van de Velde and Zijm (1996) focuses on minimizing maximum lateness with ready times and family-based sequence-dependent setup times.

The problem with sequence-dependent processing times and no setup times has been addressed by Bianco *et al.* (1988). By creating new processing times which are the sums of the original processing times and the sequence-dependent setup times, under the assumption that setup must occur while the machine is idle and the jobs are available, the $||s_{ij}||C_{max}$ problem can be transformed to the problem considered by Bianco *et al.*. The single machine scheduling problem with both ready and sequence-dependent setup times to minimize the makespan (also denoted $||s_{ij}, r_i||C_{max}$) has only been addressed directly by Chudzik and Janiak (1997), with a genetic algorithm approach.

Note that other criteria exist for the single machine scheduling problem.

Minimization of mean flow times with no due dates has been solved to optimality through use of a branch-and-bound algorithm by Dessouky and Deogun (1981), and then improved upon (in terms of computational efficiency) through development of decomposition methods by Deogun (1983). Further decomposition methods were introduced by Chand, Traub and Uzsoy (1996), based on the SPT (shortest processing time) rule which generally resulted in “more” decomposition than that yielded by the Deogun method. A dominant lower bound for the minimization of the sum of completion times based on the preemptive version of the problem was found by Ahmadi and Bagchi (1990). Minimizing the mean completion time has been considered by Liu and MacCarthy (1991), who used heuristic methods to sequence the jobs. Liu and MacCarthy also provided conditions under which their heuristics yield optimal sequences. Reeves (1995) generalized Lui and MacCarthy’s heuristic, motivated by the observation that under special circumstances the original result would not yield a good solution where a better one could be found through inspection. Jobs may have importance weights. Minimization with respect to the weighted total completion time has been considered by Gelinas and Soumis (1997). An interesting feature of their formulation is that the weights need not be non-negative, so that the objective may be non-regular. Other “advanced” criteria can be seen in heterogeneous criteria scheduling, in which different criterion are applied to different classes of jobs, as Peha (1995) has discussed.

2.2 Parallel Machines

The next level of interest is the problem of scheduling jobs with ready times and sequence-dependent setup times on identical parallel machines to minimize the makespan. This can be summarized as the $Pls_{ij}, r_i | Cmax$ problem. As with the single machine section, there are many versions of this problem which merit attention. Initially, no ready times or sequence-dependent setup times are included. Moreover, idle queuing is forbidden unless otherwise noted.

Although the problem of interest in this work does not allow preemption, some of the ideas developed under the allowance of preemption have been extended to the non-preemption case. McNaughton (1959) provided an algorithm (as part of a constructive proof) to minimize makespan on m parallel identical machines in the case of n jobs with preemption. If each job's processing time is p_i , the makespan value $s \geq \frac{1}{m} \sum_{i=1}^n p_i$ can be achieved as long as $p_i \leq s \forall i$. Hu (1961) developed an algorithm to minimize the makespan for jobs with a tree precedence constraint relationship and equal processing times, though he did not allow preemption. An important result of Hu's work is a labeling algorithm that assists in partitioning the set of jobs in many later algorithms. Muntz and Coffman generalize Hu's labeling algorithm. In Muntz and Coffman (1969), an unequal processing time version of Hu's labeling algorithm is combined with McNaughton's lower bound on the makespan to apply to the case of two machines, arbitrary precedence constraints and preemption. In Muntz and Coffman (1970), a generalization of Hu's labeling algorithm is one step in an optimal method under the

conditions of an assembly tree precedence structure. Later, Monma and Potts (1989) determined that minimizing the makespan on two parallel machines with preemptions and sequence-independent setup times is *NP*-hard. In their problem, jobs are divided into batches, which may be repartitioned. Monma and Potts (1993) also developed heuristics in the case of sequence-dependent setups, one in which the batches may be repartitioned and the other which builds on McNaughton's results. However, the setup times are only somewhat sequence-dependent; setups only occur when switching from jobs in one batch to jobs in another batch, and the setup time is the same regardless of what batch is next. Thus, from a job point of view, setups are sequence-dependent with two values possible while from a batch point of view, setups are sequence independent.

The more relevant case in which preemptions are forbidden will be discussed in much more detail. Graham (1969) developed the well-known Longest Processing Time (LPT) heuristic as well as bounds on its performance for the problem *P||Cmax*. The LPT heuristic orders the jobs from longest processing time to shortest and assigns the first job in the list to the machines as they become available. Graham's bound says that the ratio between the LPT schedule length ω_L and the optimal schedule length ω_o , where n is the number of machines, is $\frac{\omega_L}{\omega_o} \leq \frac{4}{3} - \frac{1}{3n}$. Coffman and Sethi (1976) improved this bound to

$$\frac{\omega_L}{\omega_o} \leq \frac{k+1}{k} - \frac{1}{kn}, \text{ where } k \text{ is the number of jobs on the machine that achieves the schedule}$$

makespan. Sahni (1976) interpreted the *P||Cmax* problem as a n -partition of the jobs such that the largest partition size is minimized. The relationship between partitioning and the *P||Cmax* problem is also recognized by Babel *et al.* (1998), who develop

numerous heuristics for partitioning problems. A widely utilized solution approach to this problem was developed by Coffman, Garey and Johnson (1978) and is called MULTI-FIT. The basic idea of MULTI-FIT is to pack equal-sized bins using the First-Fit-Decreasing heuristic and see if the number of bins required matches the number of machines available. The First Fit Decreasing heuristic can be described as follows, where C is the bin size (capacity), i indexes the jobs, j indexes the machines and $l(j)$ indicates the length of the partial schedule on machine j . A partial schedule is a schedule that does not include all required jobs.

1. Initialize
 - a. Sort the jobs in decreasing order of processing times p_i and re-index them from 1 to n .
 - b. Let $l(j)=0$ for $j=1 \dots m$. Let $i = 1$.
2. Let $j=1$. If $l(j) + p_i > C$, increment j . (Job i will not fit on this machine.)
3. If $l(j) + p_i \leq C$, append job i to machine j . Increment i .
4. If $i \leq n$, go to step 2. Otherwise, return the number of machines used.

The bin size is found through a binary search, which is repeated until the jobs fit into the desired number of bins. Coffman, Garey and Johnson also found bounds on the MULTI-FIT solution which were improved upon by Friesen (1984). Lee and Massey (1988) noted the strengths of both the LPT and MULTI-FIT heuristics and suggested combining them, using LPT to provide an initial solution and then MULTI-FIT as an improvement method. Blocher and Chaud (1991) also combined two approaches to this problem in order to realize a solution within a desired percentage of optimal, and developed improved bounds on the LPT heuristic. Lagrangean methods have been applied recently as well. Guignard (1993) used Lagrangean decomposition to discover plant location and

constrained 0-1 knapsack problems within the *P||Cmax* problem. Punnen and Aneja (1995) developed lower bounds for the general minmax combinatorial problem, of which *P||Cmax* is an application. Recently, genetic algorithms have been applied to this problem (for example Hou *et al.* 1994, Corrêa *et al.* 1999). In these papers, a schedule is represented by a set of strings such that each machine has a string. The string then contains the jobs assigned to that machine, in the order to be processed. In their context, precedence relations exist between jobs and they must include some methods to preserve these relations in the schedules. Min and Cheng (1998) combined genetic algorithms and simulated annealing for the *P||Cmax* problem and found that combining these methods balanced the better solutions of the genetic algorithm with longer running times of the simulated annealing algorithm.

Setup costs and/or times with special characteristics have been considered by many researchers. Sometimes, the setup times are modeled as setup costs, which are then minimized. The idea is that setup costs and setup times are related, so that minimizing setup costs will lead to minimized makespan. Parker, Deane and Holmes (1976) considered the problem with multiple parallel machines, sequence-dependent setup costs and an upper bound on the total workload of each machine, where the total setup costs should be minimized. They modeled this situation as a vehicle routing problem. Withdrawing the workload bound constraint, Bitran and Gilbert (1990) developed a travelling salesman based heuristic to minimize total setup costs on parallel machines, where the setup costs are not only sequence dependent but can be divided into two classes which vary in magnitude by degrees. The classes generally correspond to major

setup times, incurred, for example, when switching from one part family to another, and minor setup times, which accrue when switching between part types within a family. The two class division of setup times has been continued, though with the direct impact of setup times being modeled as part of the makespan. Tang (1990) considers a situation in which the setup times are not entirely sequence-dependent and uses an adaptation of the Multi-Fit method to minimize makespan. Moreover, there are no other restrictions on the setup times. Rajgopal and Bidanda (1991) proceed to include sequence-dependent setup times which can be partitioned into two classes with the aim of minimizing makespan. In this case, the major setup times are identical for all families and the minor setup times are the same for all part types. The restriction on setup times seems to have been made in order to include the sequence-dependent condition. Ovacik and Uzsoy (1993) developed worst-case error bounds in the presence of sequence-dependent setup times that were bounded by the processing times of the job setup, also noting that list schedules need not be optimal for this case. In list scheduling, a list of jobs is created according to some criterion. Then the first job on the list is assigned to the first available machine and processed. The schedule is characterized by the start time of each job. The procedure is repeated until all the jobs have been started.

Non-zero ready times of jobs has been discussed recently by Rosenkrantz *et al.* (2001) who developed a $O(n \log m)$ procedure when the n jobs have ready times and identical processing times to be scheduled on $m \geq 2$ identical parallel machines. The *on-line* version of the problem has been discussed recently (Bartal *et al.* 1995). In the *on-line* problem, only the jobs which have already arrived are known and jobs must be

assigned to machines as they arrive; this is in contrast to the standard problem in which all jobs to be scheduled are known *a priori*. Zhang (1997) studied the on-line problem in the case of two parallel identical machines with one buffer location available to hold an unassigned job in anticipation of other jobs. However, the on-line feature of these works is beyond that required in this work.

Of course, many other variations and criteria have been examined in the area of parallel machines. A variation that may be closely related to this work involves *machines* which are not available at the same time, as opposed to job ready times, under the minimization of makespan objective (Lee 1991). The case of setups that must be performed by a single server has been examined with the makespan criterion (Kravchenko 1997). Uniform (as opposed to identical) parallel machines with sequence dependent setup times have been studied with the goal of minimizing deviations from due dates (Balakrishnan *et al.* 1999). The minimization of weighted tardiness with sequence-dependent setup times (Lee and Pinedo 1997) has been considered as well as the minimization of maximum lateness (Ovacik and Uzsoy 1995; Schutten and Leussink 1996). Reward maximization when machines are capacitated and not all jobs can be scheduled was addressed by So (1990). Other cost-based criteria include minimizing the penalty cost of tardiness (Elmaghraby and Park 1974) and minimizing the sum of setup costs, production costs and deviation from due date costs (Geoffrion and Graves 1976). Drozdowski (1996) considered jobs that could be processed on more than one machine simultaneously.

2.3 Flow Lines

The study of flow lines has differed from that of parallel machines in that the makespan criteria has been very commonly addressed. The problem of minimizing makespan on a two machine flow line (denoted as $F2||C_{max}$) was solved optimally by Johnson in 1954. Johnson's rule states that jobs whose smallest operation time occurs on the first machine should be scheduled first, from smallest time to longest time, followed by those jobs whose smallest operation time occurs on the second machine, from longest time to shortest time. The resulting schedule is often also called a SPT(1)-LPT(2) schedule (Pinedo 1995). The same schedule is used on both machines – thus it is a permutation schedule. For the three machine case, where either the first machine or last machine requires more time than the middle machine, the makespan can be minimized by a permutation schedule (Johnson 1954). However, when there are more than three machines or the processing times do not follow the preceding restriction, permutation schedules do not yield the optimal solution, as can be shown by a counter-example with four machines.

The restriction to permutation schedules in flow line scheduling to minimize makespan is denoted $F||prmulC_{max}$ and has been considered by numerous researchers. Wagner formulated it as a mixed integer program in 1959. In 1976, Garey *et al.* proved that this problem was *NP*-complete for three or more machines. Ignall and Schrage (1965) applied the branch-and-bound technique to the $F3||prmulC_{max}$ problem, as did Lomnicki (1965). Since the optimal solution for a special case of the three machine case

can be found in the set of permutation schedules, this restriction may not actually be a restriction at all, depending on the problem-specific data.

Some heuristics have been developed based on algebraic / arithmetic properties of the problem. The basic idea is that alternatives are compared and one chosen based on the results of these simple comparisons. Dudek and Teuton (1964) developed a multi-step algorithm for M machines that uses comparisons to select jobs and the satisfaction of certain conditions to determine the placement of those jobs. Gupta (1972) described three of these types of algorithms, building on Smith and Dudek's work and Johnson's Rule for two machines. Campbell *et al.* (1970) presented an approach based on Johnson's rule adapted to a general number of machines often called the CDS procedure. The M machines are partitioned iteratively into two psuedo-machines such that in the i th iteration, the first i machines are grouped into the first pseudo-machine and the last i are grouped into the second psuedo-machine. The operation times are then grouped in the same manner and Johnson's Rule applied, resulting in several alternatives with their own makespans. Then the best of the $M-1$ alternatives is selected. Palmer (1965) developed the idea of a slope index so that jobs whose operation times tend to move from short times to long times as the stages are visited are preferred for earlier placement in the schedule. Dannenbring (1977) also forms approximations of a two-machine problem from the M machine data given and provides two methods for improvement based on interchanging jobs in the original solution. Stinson and Smith (1982) used a travelling salesman approach in which the distances between cities represents the amount of blocking/idleness that may occur between successive jobs on machines, in a departure

from the Johnson's Rule-based methods. The approach by Nawaz *et al.* (1983) incrementally determines the relative position of jobs in the final sequence and is often denoted NEH. Partial sequences are created by choosing to place the job under consideration in the position relative to the current sequence (before or after each job) that minimizes the makespan thus far. Ho and Chang (1991) also used an algebraic approach based on the observation that the makespan is increased by gaps between processing of jobs on consecutive machines. They use a measure of this gap to select jobs to be exchanged in the schedule and keep the exchange if the makespan is reduced. A linear programming approach proposed by Sevast'yanov was used by Lourenco (1996) who focused primarily on the computational aspects of this algorithm.

More recently, advanced combinatorial heuristic methods such as tabu search and simulated annealing have been applied. Widmer and Hertz (1989) construct an initial solution through a TSP approach and improving that solution with a tabu search method. The distance between jobs i and j is based on a weighting of the processing times of operations that would occur simultaneously if these jobs were scheduled in this order, penalizing the differences on earlier machines more. The tabu search uses neighborhoods which result from pair-wise interchanges of jobs; the opposite exchange is then tabu. Taillard (1990) utilized tabu search as well, but defined neighborhoods differently. The neighborhoods are generated by insertion, which involves moving a job from one position to another. Moves are tabu if they result in a makespan that has already been achieved. Based on Taillard's empirical demonstration that exchanging is not better than insertion, Nowicki and Smutnicki (1996) used tabu search with insertion. However, for

problems with a large number of jobs, insertion can lead to large neighborhoods whose complete examination could take quite a long time. Thus, the idea of *blocks* is used to reduce the size of the neighborhood examined. Moreover, Back Jump Tracking is used to return to the neighborhood of attractive solutions which were left by following one neighbor when another neighbor could have been chosen. Ben-Daya and Al-Fawzan (1998) commented on the comprehension complexity of the Nowicki and Smutnicki application of tabu search and countered with a implementation of tabu search that advocated simple methods. For example, the neighborhood is generated using either exchange, insertion or block insertion (a set of jobs is moved together as one in a regular insertion), chosen randomly at each iteration. The first neighbor that generates an improvement is accepted. These two decisions are in contrast to the Nowicki and Smutnicki implementation in which much time is devoted to reducing the size of the neighborhood in order to examine more of it. However, advances include the acceptance of tabu moves that improve the makespan is also allowed and the resetting of the seed solution in order to diversify the solution.

Simulated annealing has been applied by numerous researchers as well. Osman and Potts (1989) examined simulated annealing with two types of neighborhoods: the exchange neighborhood as defined for tabu search and the shift neighborhood which has been called the insertion neighborhood in the tabu search discussion. Moreover, the neighborhoods can either be examined in some sort of predetermined order or in a random order. In their investigations, the shift neighborhood and random order performed best of the four combinations possible here; this conclusion held even when

compared to NEH. Zegordi *et al.* (1995) incorporated the concept of desirability of moving jobs in either direction in the sequence into a simulated annealing pair-wise exchange step. In this way, more problem specific information could be utilized.

Ishibuchi *et al.* (1995) varied the basic simulated annealing algorithm by generating multiple neighbors from each current solution for evaluation; parallel implementation was a major motivation for this development.

Werner (1993) developed a new type of approach called a path algorithm, which is an iterative or improvement method such as simulated annealing or tabu search. A graph is developed where each vertex represents the processing of job i on machine j and has weight equal to that processing time. The only arcs included are those that represent the next job being performed on the same machine or the same job being performed on the next machine; that is, the following arcs would exist: (i, j) to $(i, j+1)$ or $(i+1, j)$ only. The longest path represents the makespan. Changing the order of the jobs in this graph changes the length of the longest path. The neighborhood of a schedule is considered to be all the possible results of insertions or shifts, and is called the shift graph. As in Nowicki and Smutnicki, considerable time is spent reducing the size of the neighborhood so that a reasonable search can be performed. The basic reduction idea is that a neighbor must not have the same order of jobs in the longest path after one job has been shifted to a new location. Moreover, simulated annealing ideas are incorporated to include a non-deterministic factor to the algorithm, so that the current solution may escape local optima.

While maintaining the permutation schedule requirement, other conditions have been relaxed. Yoshida and Hitomi (1979) introduced the idea of separable setup times

which means that setup for the next machine can begin before processing has completed on the current machine if the next machine is idle. An adaptation of Johnson's Rule is given for the two machine case. This idea was extended further by Sule (1982), who added removal, which is analogous to setup but occurs after processing. Thus the job is available to the next machine but the machine is not available for the next job's setup until removal is completed. Johnson's Rule was also adapted for this case with two machines. The same type of two machine sub-problem formation seen in the work by Campbell was applied to the three machines with separable setup and removal times case by Sule and Huang (1983). Rajendran and Ziegler (1997) extended the allowance of separable setup and removal times to an arbitrary number of machines (stages). Note that to change from one job to another, a removal must be performed followed by a new setup. The setups are sequence independent, as are the removal times. However, if the two were considered as one, the total time between ending one job's processing time and beginning the next job could be considered sequence-dependent. Proust *et al.* (1991) used this approach to investigate the sequence-dependent, but separable, setup and removal time problem in a flow shop with an arbitrary number of machines.

The sequence dependent setup time problem has been considered directly for quite a while. Corwin and Esogbue (1974) considered the situation of a two stage flow line where one of the machines incurs a sequence-dependent setup time. Dynamic programming formulations were created and discussed with regard to computing time and space requirements in comparison to the TSP. They also show that the set of permutation schedules contains the optimal schedule for this case. Gupta and Darrow

(1986) established that the inclusion of sequence dependent setup times on only one of these machines results in a *NP*-complete problem and that the set of permutation schedules does not necessarily contain the optimal schedule when both stages have sequence-dependent setup times. They attacked the problem by setting up two TSPs, one for each machine, incorporating the sequence dependent setup times. The larger of the shortest path in each of these provide lower bounds on the problem of interest; moreover, if a permutation schedule is found that results in a makespan given by the lower bound, then that permutation schedule is optimal. Gupta (1986) also established the *NP*-completeness of the $F|s_{ij}|C_{max}$ problem for an arbitrary number of stages. In this case, one TSP is developed based on the idea of the amount of time a job must be delayed before beginning processing on the first machine in order to avoid delays on any other machines (continuous processing). . However, in both the Gupta papers mentioned above (Gupta and Darrow 1986; Gupta 1986) the TSP problem still exists, which is *NP*-complete as well. Srikar and Ghosh (1986) developed a mixed integer linear program for the $F|s_{ij}|C_{max}$ problem in which a different set of decision variables was used to represent the ordering of jobs relative to one another in a general rather than a specific manner. That is, the integer variable indicates that one job precedes another, not that one job *immediately* precedes another. Though this approach was useful in this area, several errors in their paper were found and later addressed by Stafford and Tseng (1990). After several corrections, Stafford and Tseng extended the MILP model to handle the cases of independent setup times and no buffers between stages. Recently, Ríos-Mercado and Bard have produced numerous papers in this area. In 1998, they developed valid

inequalities for a mixed integer program that assisted in a branch-and-cut heuristic approach. They also developed a branch-and-bound algorithm for the $F|pmu, s_{ij}|C_{max}$ problem (1999a). Additional work extended research by Simons (1992) that transformed this problem into an asymmetric TSP where the distances between jobs is set as the sum of the processing and setup times over all machines. Ríos-Mercado and Bard (1999b) extended Simons' research by creating different times between the cities that reflect not only the impact of sequence-dependent setup times but also the impact on the entire schedule makespan of scheduling jobs next to each other. Though these research addresses flow lines with sequence-dependent setup times, each stage has only one machine and the schedules are restricted to be a permutation schedule. In a slight variation, Sotskov *et al.* (1996) considered jobs which were partitioned into batches and setup times that occurred whenever the first job of a batch was processed. The jobs in a batch may only be processed on the current machine when all the jobs in the batch have finished processing on the previous machine. Insertion heuristics are used to generate the initial schedule and local search is used for improvement.

Ready times have been considered for the two machine flow line case $F2|pmu, r_i|C_{max}$ by a limited number of researchers. Potts (1985), who also examined the ready time problem in single machine scheduling (Potts 1980), attacked the problem by combining Johnson's Rule with the condition that only jobs that are currently available be considered for placement into the schedule. Hall (1994) also uses adaptations of Johnson's Rule, additionally building on work for the $1|r_i|C_{max}$ problem (Hall and

Shmoys 1992). Tadei *et al.* (1998) addressed this area by focusing on branch-and-bound techniques.

Relaxing the permutation requirement seems to have only been considered by Koulamas (1998). Initially, a priority index is produced for each job by applying Johnson's Rule to the various two-machine sub-problems that can be formed from the original M machine problem (there are M choose 2 of these) and changing the priority of a job based on its location in each of the sub-problems. Then the jobs are sorted in non-decreasing order of priority indices, resulting in a permutation schedule. The second step looks for pairs of jobs which have a certain order in all the two-machine sub-problems defined in machines before machine j and the reversed order in all the two-machine sub-problems defined in machines after machine j ; that is, jobs whose relative order would switch and stay switched after a certain machine is reached in the two-machine sub-problems. The jobs are eligible to switch position in the schedule at this machine, resulting in a non-permutation schedule. If this switch results in a lower makespan, it is retained. This is repeated for every adjacent pair of jobs in the permutation schedule resulting from the first step.

Minimizing a weighted sum of makespan and job flowtime has been considered by multiple authors including Nagar, Heragu and Haddock, beginning with a branch and bound approach (1995) and then combining it with a genetic algorithm approach (1996). Gangadharan and Rajendran (1994) attacked the problem with simulated annealing techniques.

Recently, Schaller (2000) investigated several heuristics for scheduling a flowline manufacturing cell in which jobs belonged to families. The jobs have independent setup times which are incorporated into the processing times while the families have sequence independent setup times which are incurred when switching from one family to another. Variations of the NEH and CDS procedures, tabu search and genetic algorithms are compared. Results indicate that heuristics should form sequences within families and then sequence families, that variations of the NEH and CDS procedures are computationally efficient and that tabu search, while computationally intensive, finds lowest makespans.

Clearly, the *FlpmulCmax* problem has been more closely examined than the *FlCmax* problem. Sequence-dependent setup times have not been extensively studied, except by Ríos-Mercado and Bard, who require permutation schedules and single machines at each stage, in contrast to our interest. Moreover, the problem with non-equal ready times has not been considered for more than a few machines, and has thus far required the permutation schedule restriction. Clearly, the flow line makespan minimization problem has many unexamined areas.

2.4 Hybrid Flow Lines

In the remainder of the literature review, effort has been made to accurately categorize papers by the production system as opposed to the titles of the papers. A hybrid flow shop is one in which every job must be processed at every stage in the same order. A flexible flow line is a hybrid flow shop in which at least one job is processed at

fewer than all the stages. Much of the literature regarding hybrid flow shops has used the phrase “flexible flow line” even though all jobs must be processed at every stage.

The basic assumptions for the hybrid flow shop include the restriction to permutation schedules and unlimited buffers between stages. As in the flow line discussion above, basic approaches can be divided into algebraic / arithmetic approaches and heuristic combinatorial methods.

The algebraic approach began when Salvador (1973) first introduced a combination of a multiple machines at stages in a single line flow shop. However, no buffers exist between stages in Salvador’s version. A graphical representation of a hybrid flow shop is developed that is furthermore dynamic, so it can represent the evolving state of the hybrid flow shop. Branch-and-bound techniques are then applied to determine the optimal permutation schedule in terms of makespan. In 1989, Sriskandarajah and Sethi examined worst case performance for various heuristics applied to two stage hybrid flow shops. The two specific configurations of one machine at the first stage and m machines ($m > 1$) at the second stage and an equal number at each stage were examined, to which various adaptations of Johnson’s Rule are applied. The worst case bounds developed for the solutions derived from the heuristics in this paper were never less than double. Johnson’s Rule was also applied by Gupta (1997) to the case with one machine in the first stage and multiple machines in the second stage

The branch and bound algorithm was once again considered by Brah and Hunsucker (1991) for application to the hybrid flow shop with an arbitrary number of stages and intermediate buffers. Moreover, they provide a means by which non-

permutation schedules or schedules with inserted idle time (non-non-delay schedules or delay schedules) can be created, by basically considering the branch and bound problem for each stage and using the completed schedule on the previous stage as a beginning point. Moreover, non-identical job release times can be considered in the first stage. Rajendran and Chaudhuri (1992) also utilized branch-and-bound but restricted the resultant schedule to the set of permutation schedules. Riane *et al.* (1998) considered a three stage hybrid flow shop with two machines at the second stage and one at each of the other stages. They developed a dynamic programming-based heuristic based on the CDS heuristic and a branch-and-bound heuristic.

Ding and Kittichatphayak (1994) modeled the hybrid flow line as an extension of a single line flow shop. Three constructive heuristics were introduced. Two are adaptations of the CDS heuristic which differ in where a new job is placed in the sequence (either the end or the beginning). The third involves placing jobs at the end of the current sequence in light of the idle time of the machines. Then, these heuristics are combined and the best result among the three is reported. Chen (1995) considered the classification of heuristics for the special case of the hybrid flow shop with two stages where one stage has exactly one machine. Worst case bounds are developed for each of the major classification families and are used for comparison purposes. Brah and Loo (1999) used simulation to evaluate how heuristics developed for the single flow line case would perform in the hybrid environment. They found that problem-specific data such as number of machines at each stage and processing times of operations had much more effect on the solution quality than the actual heuristic applied.

Non-permutation schedules have been considered in at least two papers other than that by Brah and Hunsucker (1991), discussed above. Lee and Vairaktarakis (1994) developed heuristics for multistage hybrid flow shops by extending results for a two stage hybrid flow shop. For a two stage problem, the machines in each stage are aggregated into one and the processing time required at each stage is reduced by dividing by the number of machines at that stage. Then Johnson's Rule is applied. The jobs are assigned to machines in the first stage in the same manner as a list schedule: the next unassigned job in the schedule goes to the machine that becomes idle first. The jobs are assigned to the machines in the second stage in a similar manner; the last job in the schedule is assigned to the machine that has the least amount of work assigned to it so far. Then the jobs at each machine in the second stage are ordered in non-increasing order of their completion times on the first stage machines. For more than two stages, this procedure is applied to two consecutive stages, adding a dummy stage if the number of stages is odd, and concatenating the results. This heuristic need not always result in a permutation schedule. Santos *et al.* (1995) combined permutation schedules with the FIFO queuing discipline at all stages after the first stage. The jobs enter the line according to one of the $n!$ permutations and then begin processing at every stage thereafter in the order they completed processing at the previous stage. The resultant schedule is not a permutation schedule, because the same order is not maintained at every stage. The optimal solution is found by considering every possible permutation of jobs that can enter the line, tempered by the introduction of a lower bound on the optimal makespan. In a later paper, Santos *et al.* (1996) considered the use of heuristics developed for the single flow shop

case as a method to generate an initial permutation schedule that would then be followed by the application of FIFO as above.

Heuristic combinatorial approaches such as tabu search and simulated annealing as applied to the hybrid flow shop seems to have been limited to a recent application of tabu search by Nowicki and Smutnicki in 1998, building on their prior experience with tabu search in single machine scheduling (1994) and flow shops (1996). Neighborhoods of solutions are created by moving a job from one position to another, resulting in permutation schedules only. The interesting development in this paper is in the tabu list. Moves are forbidden based on *attributes* of the solutions, not on the solutions themselves.

Other related variations include parallel flow shops (Sundararaghavan *et al.* 1997), unrelated parallel machines at the stages (Suresh 1997), uniform parallel machines and identical jobs (Dessouky *et al.* 1998) and multiple machines that can process multiple operations simultaneously (Oguz and Ercan 1997). Separable setup and removal times, seen in the single flow line case, have also been considered in the two stage hybrid case where the first stage has only one machine (Gupta and Tunc 1994).

Vairaktarakis and Elhafi (2000) compared the makespan performance of a two stage hybrid flow line where the first stage has fewer machines than the second with a transformed version of the original problem with several two stage parallel hybrid flow lines where the first stage has one machine. The idea is to assign jobs to one of the parallel hybrid flow shops to minimize makespan, and the resulting decrease in scheduling complexity and movement of materials between machines in the two stages will hopefully be offset by little deterioration in makespan.

As can be seen from the above discussion, the hybrid case has been much less well explored than the single flow line case. Oftentimes the environment is limited to two stages, where one of the stages has only one machine. Also, sequence dependent setup times have not been considered for the general case or with non-zero release times.

2.5 Flexible Flow Lines

In contrast to the single flow line and hybrid flow line problems, researchers addressing the flexible flow line problem did not start by adapting Johnson's Rule. Instead, a common approach was to examine the problem in terms of Loading (deciding what enters the line when) and Dispatching (deciding how to empty buffers between stages). Recent approaches have focused on combinatorial heuristics.

The examination of flexible flow lines as defined in this paper seems to have begun with Wittrock in 1985. Wittrock initially considers the scheduling problem to be a two part problem: determine what proportion of each part type is to be produced each day, which is the Mix Allocation problem and results in a part mix, and then determine when parts should begin processing at the first stage, which is the Loading problem. The Loading problem is most relevant to the work here. The salient characteristics of the flexible flow line in this context include no setup time between processing of different part types and infinite buffers between stages which are emptied in a first-come, first-serve manner. In this situation, the objective is to obtain the minimum makespan schedule which also minimizes the longest queue over all the buffers. Wittrock notes that setup times should be included and that the periodic basis for the schedule is flawed. In

1988, Wittrock re-examined the sequencing sub-problem in detail without the periodic basis. As in 1985, he immediately proceeded to a heuristic solution approach, noting that the scheduling problem to minimize makespan was *NP*-complete. Again, machine allocation is performed through application of the LPT heuristic. He focuses on sequencing by developing a heuristic called Workload Approximation (WLA). Wittrock also adapts his basic algorithm to account for some real world conditions. First, it is clear that in real life, buffers are not unlimited. Expediting of parts occur sometimes, often called “hot lots”. And finally, he considers that the line may already be loaded with parts in process when scheduling occurs. This kind of rescheduling may be required if machines fail, production requirements or raw material availability change, or some parts require rework. Computational results of Wittrock’s work indicate that the algorithm can be applied quickly to flexible flow lines of reasonable size (13 part types, 3 stages of machines) with low makespans in both the unlimited buffer and limited buffer cases and low queue lengths in the unlimited buffer constraints.

Kochhar and Morris (1987) also considered the flexible flow line scheduling problem. Their model of flexible flow lines is more complete than Wittrock’s in that it allows for setups between jobs, finite buffers which may cause blocking, and machine down-time. The first part of the problem solution is called Entry Point Sequencing, which is the same as Wittrock’s Loading Problem. The entry point sequence is found by examining permutations of the jobs requiring processing with repeated local search to escape local optima. The second part is called Dispatching and describes how jobs are

selected for processing at the later stages from the buffer. A deterministic simulation is used for evaluation of various operational policies.

Sawik has developed numerous results for the flexible flow line scheduling problem. The basic model in all his papers discussed here include important factors such as transportation time between stages and non-zero release times. However, sequence-dependent setup times are not included. The first heuristic developed is called *Route Idle Time Minimization (RITM)* (1992), which aims to minimize makespan by minimizing the idle time of the machines. In this case, buffers are limited in size meaning blocking can occur. The time that each part begins and ends processing at each stage is determined explicitly in these heuristics. Parts are selected one at a time and scheduled for the entire line at that time. The parts are placed in non-increasing order of the weighted sum of processing times where the processing times for later stages are weighted more heavily. Each job placed in this order also has another associated weight so that the last job has weight 1 and each earlier job has a weight that decreases by a constant multiplier. Each job that has not been scheduled already is considered for selection. The job with minimal weighted idle time (were it to be scheduled now) is selected. Jobs are placed on the machine with the earliest available time at each stage. This is repeated for each job to be scheduled. Later, Sawik extended the *RITM* heuristic to the case of no buffers between stages (1994, 1995). In these papers, the *RITM_NQ* (no queue) and *RITM_NS* (no store) heuristics are presented and appear to be identical. The key difference between these algorithms and the limited buffer version, *RITM*, is that no weights are used in

determining the jobs order for selection consideration or in determining the idle time for selection of the next job.

Improvement techniques differ from those previously discussed. Local search improvement techniques were considered by Leon and Ramamoorthy (1997). Rather than considering neighborhoods of the schedules, they considered neighborhoods of the problem data. The perturbed data is then used by a problem-specific heuristic to generate a solution whose quality is assessed using the original problem data. This approach is interesting because of its built-in consideration of robustness; schedules that are generated by perturbed data but perform well for the original data are favored. Genetic algorithms have been applied to variations of the basic flexible flow line problem. Lee *et al.* (1997) applied genetic algorithms to the joint problem of determining lot sizes and sequence to minimize makespan. This research also included sequence-dependent setup times. Moreover, combining genetic algorithms with simulated annealing was also considered. In our research, however, we do not have lot sizes which need to be determined.

Some other variations that are possible include jobs with operations that can be processed at a given stage each or all at one stage (Kouvelis and Vairaktarakis 1998) and limiting the total number of jobs simultaneously in the system (Hunsucker and Brah 1994).

Oguz *et al.* (1997) examined a problem that could be considered either a three stage flexible flow line with one machine at each stage or a two stage hybrid flow line with unrelated machines in the first stage. The three machines can be called *A*, *B*, and *C*.

Jobs visit either A and C or B and C ; A and B are not identical, so either of the situations defined above applies. Oguz *et al.* modeled this as a three stage flow line where the difference in job routings were handled as different job types. Then Johnson's Rule was used for each of the types of jobs. The resultant schedules were manipulated so that jobs were delayed as little as possible before beginning processing on the common machine, resulting in a permutation schedule of sorts.

Though a significant amount of work has been done in the general areas of scheduling parallel machines with sequence-dependent setup times and flow lines has been done, no work directly addressing our problem of interest has been done. Allahverdi *et al.* (1999) provide a review of scheduling research and note no research in scheduling hybrid flow lines or flexible flow lines with sequence-dependent setup times. This area, though identified by Wittrock in 1985, still has not been directly considered.

2.6 Some Traveling Salesman Problem Solution Methods

In this section we shall discuss two basic methods to find solutions to the Traveling Salesman Problem. Both methods are constructive. The first group heavily depends on graph theoretic concepts and can be called Minimum Spanning Tree Heuristics. The second group is based on the idea of inserting cities in "good" places and are called Insertion Heuristics. These methods are included due to the importance of the TSP in solving sequence-dependent problems.

2.6.1 Graph Theory Background

The TSP can be represented as a graph. In this section we review several concepts from Graph Theory. These concepts can also be found in Wilson (1996).

A **graph** G consists of a non-empty finite set $N(G)$ of elements called **nodes** and a finite family $E(G)$ of pairs of elements of $N(G)$ called **edges**. In general, the number of nodes or cardinality of the set $N(G)$ is denoted by n . Nodes can also be called vertices and edges can also be called arcs. In this work, edges can be either directed (ordered pairs of nodes) or undirected (unordered pairs of nodes). Edges are denoted (a,b) , where a and b are elements of $N(G)$ and a and b may be identical. Edges can have **weights** (or **distances**) associated with them, denoted w_{ab} . Consider a set of three edges denoted by (a,b) , (b,c) , and (c,a) with weights w_{ab} , w_{bc} , and w_{ca} . If $w_{ab} + w_{bc} \geq w_{ca}$, then we say that the edges satisfy the **triangle inequality**. If all sets of three edges of the form (a,b) , (b,c) , and (c,a) satisfy the triangle inequality, then the graph is said to be a **triangle inequality-satisfying graph**. A **connected graph** is one in which at least one sequence of adjacent edges exists between every pair of distinct nodes in the graph. A **complete graph** is a graph with edges between every pair of nodes. A **trail** of a graph is a sequence of adjacent edges in which all edges are distinct. A trail that includes every edge in a connected graph is an **Eulerian trail**. A graph that contains a closed Eulerian trail is called an **Eulerian graph**. The **degree** of a node is the number of edges claiming that node. If every node in a connected graph has even degree, then that graph is an Eulerian graph. A **path** of a graph is a trail in which all nodes (except possibly the first node in the first edge and the last node in the last edge) are distinct. Trails and paths can be

closed, in which case the first node in the first edge and the last node in the last edge are the same. A closed path with at least one edge is a **cycle**. A **Hamiltonian cycle** is a cycle that passes through each node exactly once. A graph that contains an Hamiltonian cycle is called a **Hamiltonian graph**. If a Hamiltonian graph has weighted edges, then the length of the Hamiltonian cycle is the sum of the edges in the cycle. If a Hamiltonian graph does not have weighted edges, we can take the weights to be 1 on each edge and set the length of the Hamiltonian cycle equal to the number of edges in it. Determining that a graph is Hamiltonian is not as easy as determining that a graph is Eulerian. Several theorems regarding sufficient or necessary conditions exist, but there is not a theorem listing necessary and sufficient conditions (Wilson 1996). However, every complete graph is Hamiltonian. A **tree** is a connected graph with no cycles. A **spanning tree**, T , of a graph G , is a tree composed of edges and nodes in G such that every node in G is also in T . A **minimum spanning tree (MST)** is a spanning tree of a graph with weighted edges such that the sum of the weights of the edges in the MST is minimal. A **perfect matching** of a set of nodes V , with $2k$ nodes, is a set of k edges such that every node is contained in exactly one edge. A **minimum weight perfect matching** is a perfect matching where every edge has a weight and the sum of the weights in the perfect matching is minimal.

The TSP can be represented by a graph where each node represents a city and the edges represent the direct travel between cities, with weights equal to the travel times. In our context cities are analogous to jobs and travel times are analogous to setup times. If setup times are symmetric then the edges can be undirected, while if the setup times are

asymmetric, directed edges are required. Generally, edges of the form (a, a) are not explicitly included, but could be included with very high weights to preclude their inclusion in a TSP solution. A Hamiltonian cycle of the TSP graph provides a solution, though not necessarily the optimal solution, to the TSP problem.

2.6.1.1 Finding a Minimum Spanning Tree

A minimum spanning tree can be found using either Prim's or Kruskal's Algorithm. Prim's Algorithm starts with the lowest weighted edge as the tree. Edges are added to the tree by finding the smallest weight edge incident to a node in the tree that would not create a cycle when added. After $n-1$ edges have been added in this manner, a MST has been formed. Prim's algorithm can be implemented in $O(n^2)$ time. Kruskal's Algorithm builds up a tree from the edges, though the graph is not necessarily connected at all times. Kruskal's Algorithm requires a non-increasing sort of all the edges by weight. The smallest weighted edge is added to the graph as long as it does not create a cycle in the graph. After $n-1$ edges have been added in this manner, a MST has been formed. By using *components* to determine if an edge's addition would create a cycle, Kruskal's Algorithm can be implemented in $O(n^2 \log n^2)$ time.

A MST can be used to create an Eulerian graph by simply "doubling" the MST. This means creating a duplicate edge for every edge that exists in the MST. Whichever nodes had odd degree originally will now have even degree, while those with even degree previously will still have even degree. The doubled edges in the Eulerian graph have the same weights as the original edges in the MST.

2.6.1.2 Finding an Eulerian Trail

An Eulerian trail of an Eulerian graph can be found using Fleury's Algorithm.

Fleury's algorithm can be stated as the following theorem, which is proven in Wilson (1996). Note that a **bridge** is defined to be a single edge in a connected graph, which if deleted, would result in the graph becoming disconnected.

Theorem: Let G be an Eulerian graph. The following is always possible and results in a closed Eulerian trail E of G . Start at any node and traverse the edges in an arbitrary manner, subject to the following rules:

- (i) Delete the edges from G and add them to E as they are traversed. Isolated nodes are deleted from G as well.
- (ii) Traverse a bridge only if there is no alternative.

Since every edge in an Eulerian graph formed by doubling a MST is required to produce an Eulerian trail, the Eulerian trail has length equal to twice the minimum spanning tree.

2.6.1.3 Finding a Hamiltonian Cycle

Once a closed Eulerian trail is found, it can be used, with the original graph, to find a Hamiltonian cycle of the nodes in the Eulerian trail. The Eulerian trail E can be represented by a list of nodes, $v_0, v_1, \dots, v_k$ which includes the nodes visited in order.

Note that $v_0 = v_k$ and other nodes may also be repeated. Beginning with an empty graph H , one can build a Hamiltonian cycle in H using the method in Reinelt (1994). This procedure moves through the nodes in E , beginning at an arbitrary node which is added to H and deleted from E . The current node is already in H . We look through E until we

find a node not in H . Then we add this node to H and repeat. The edges in H are then either edges in E or shortcuts, skipping edges in E . If G satisfies the triangle inequality, the shortcut edges are no longer than the length of the skipped edges. Thus, the length of the Hamiltonian cycle is not more than that of the Eulerian trail.

2.6.2 Minimum Spanning Tree Heuristics

If the sequence dependent setup times satisfy the triangle inequality and are symmetric, then the above concepts can be used in the Doubling Method to find a Hamiltonian cycle. The Doubling Method can be summarized as follows:

1. Compute a minimum spanning tree (MST) of the graph G and call it M .
2. Double every edge of M to get an Eulerian Graph called E .
3. Find an Eulerian trail of E .
4. Find a Hamiltonian cycle of E .

The Christofides Method differs from the Doubling Method in step 2. Christofides suggested finding a minimum weight perfect matching, P , of the odd-degree nodes in M . Then $P+M$, formed by adding the edges in P to M , is an Eulerian graph. The Doubling and Christofides Methods are considered the primary heuristic methods for the TSP by Papadimitriou and Steiglitz (1998).

The Doubling Method has a running time of n^2 but a worse case bound of 2 times the optimal cycle length. The Christofides Method has a running time of n^4 or n^3 (depending on the implementation of the minimum weight perfect matching) and a worst case bound of 1.5 times the optimal cycle length. Since the number of jobs could be on

the order of 100 for each stage, a running time of n^4 or n^3 is considered too costly at this time. For this reason, the Doubling Method has been implemented.

2.6.3 Insertion Heuristics

Insertion Heuristics build a TSP cycle directly by inserting nodes into an incrementally built up partial cycle. Insertion Heuristics on a set N of nodes have the following basic steps:

1. Select a starting node for the cycle. Remove this node from N .
2. As long as N is not empty,
 - a. Select the next node i from N according to some criterion.
 - b. Insert i into some position of the cycle.
 - c. Remove i from N .

Insertion Heuristics vary in their criteria for selecting the next node and determining the position for the next node in the tour. Let the nodes in the partial cycle be called **cycle nodes**. A Nearest Insertion Heuristic inserts the node that has the shortest distance to a cycle node. A Farthest Insertion Heuristic inserts the node that has the longest distance to a cycle node. A Cheapest Insertion Heuristic inserts the node that increases the total length of the cycle the least (Reinelt 1994). Insertion Heuristics have a worst case bound of twice the optimal tour length (Johnson and Papadimitriou 1985).

2.7 Genetic Algorithms

Genetic algorithms were introduced in 1975 by Holland. He provided the basic framework for genetic algorithms: chromosomes represent solutions which reproduce

based on how well they solve the problem at hand in a manner analogous to survival of the fittest. A key feature of genetic algorithms is their randomness. Chromosomes are chosen to reproduce randomly and experience changes randomly, as organisms do in the natural environment. Theoretically, the best chromosome will survive to the final generation of chromosomes. A genetic algorithm needs the following five primary components, which will be discussed below (Davis and Steenstrup 1987):

1. a chromosomal representation of solutions to the problem
2. a method to initialize a population
3. an evaluation function that represents how well the individual solutions function in the environment, called their “fitness”
4. genetic operators that change the composition of the chromosomes
5. values for the parameters that are required

Chromosomes represent solutions to problems. The chromosomal representation of a solution is an important design feature of a genetic algorithm. Often, the chromosome is a string of 0s and 1s; however, other possibilities exist, such as strings of non-negative integers. Each piece of information in the chromosome is known as a gene. The population of chromosomes is often initialized randomly, but could be initialized based on the results of a greedy heuristic. The chromosomes are chosen to reproduce based on their fitness, which is a measure of how well a solution solves the problem of interest. For example, if the problem is to find the maximum of some difficult function, such as $f(x) = x \sin(10\pi x) + 1.0$, $x \in [-1, 2]$, the fitness of a chromosome corresponding to a real value x can be simply the value $f(x)$. The evaluation function represents how

well the chromosome performs in the environment. Holland defined three kinds of genetic operators: crossover, mutation and inversion. Only the first two are in common usage now and appear in many different forms (Davis and Steenstrup 1987). Crossover is considered mandatory for an algorithm to be a genetic algorithm. Here we will summarize Holland's definitions, and introduce others as needed. The crossover operator randomly selects two chromosomes, A and B, and a random integer between 1 and one less than the number of genes in the chromosome. This number is the *crossover point* of the chromosomes. The chromosomes are then cut after this point and put back together with the second halves swapped. For example, if A= 010001, B= 111000 and the crossover point is 3, the chromosomes become A'= 010|000 and B'= 111|001, where the 'l' indicates where the cut occurred. The mutation operator selects a gene in a chromosome and randomly changes it. The inversion operation, which as mentioned earlier is rarely used, takes a chromosome and two random integers between 1 and the length of the chromosome, a and b such that $a \leq b$. The genes between and including spots a and b are then reversed. For example, if the chromosome is 001100 and $a=3$ and $b=5$, the new chromosome after inversion is 00|011|0, where the genes between the 'l' symbols have been reversed. The final component of a genetic algorithm is the values of the required parameters. The required parameters include, at a minimum, the number of chromosomes in the population (constant), the number of genes in a chromosome, the probability of selecting any one chromosome for crossover and the number of generations that will be generated. Other parameters could include the probability of selecting any one gene for mutation, the probability of selecting any one chromosome for inversion,

and the difference allowed between successive best solutions before the algorithm stops.

Clearly, additional parameters exist for other variations that could be developed.

Choosing values for parameters is very time consuming and has a great influence on the performance of the genetic algorithm.

The chromosomal representation of the scheduling information can take many forms and influences the types of genetic operators. One option for the chromosomal representation in a scheduling application such as ours is an array indexed by the job number, where each value in the array corresponds to the machine to which the job is assigned. For example, the chromosome [2, 3, 1, 3, 2, 1] means that machine 1 has jobs 3 and 6, machine 2 has jobs 1 and 5, and machine 3 has jobs 2 and 4. This kind of structure is used by Gu and Chung (1999) for assigning gates to flights in airports. In their application, gates can be repeated but the ordering of flights is not part of the problem, since the flight schedule, even if changing due to delay, is an input and not an output. In our application, however, the job order is relevant. The job order could be found by a TSP for each machine. With this representation, mutations consist of changing the assigned machine for one or more jobs. Each job in each chromosome in the current population is eligible for crossover, and any number of jobs can be selected for crossover. Clearly, these operations could result in no change in the schedule, reducing the effective probabilities of crossover and mutation. Sivrikaya-Serifoglu and Ulusoy (1999) use the job and machine, in that order, as a gene. The jobs are processed on the machines in the order in which they appear in the chromosome. Their example chromosome [4-1, 2-1, 5-2, 1-1, 3-2] means that machine 1 processes jobs 4, 2, and 1 in that order and machine 2

processes jobs 5 and 3. With this representation, a swap mutation involves selecting two jobs in a chromosome in the current population and exchanging them. This may result in a different job ordering on up to two machines. A bit mutation involves selecting a job and randomly assigning a machine to that job. This representation and types of mutation could also result in no change to the schedule.

Bean (1994) has introduced an alternative method to encode problem solutions using random numbers called a Random Keys Genetic Algorithm (RKGA), which has been applied to resource allocation problems, quadratic assignment problems, multiple machine tardiness scheduling problems, jobshop makespan minimization problems and the generalized traveling salesman problem (Bean 1994, Norman and Bean 1999, Snyder and Daskin 2001).

RKGA differs from traditional genetic algorithms most notably in the solution representation. Random numbers serve as sort keys in order to decode the solution. The decoded solution is evaluated with a fitness function that is appropriate for the problem at hand. For example, Norman and Bean (1999) suggest using the following solution representation for an identical multiple machine problem. Each job is assigned a real number whose integer part is the machine number to which the job is assigned and whose fractional part is used to sort the jobs assigned to each machine. Once the job assignments and order on each machine is found through the decoding, a schedule can be built incorporating additional factors such as nonzero ready times and sequence-dependent setup times. The desired performance measure can then be found using the schedule. The genetic operators and related parameters used in Bean (1994) are

summarized in the following. Each generation has a population of 100 chromosomes. The initial population is generated randomly. An elitist strategy is used for reproduction. Each chromosome is decoded and the resulting schedule is evaluated for the makespan. Chromosomes with lower makespans are more desirable, so the 20% lowest makespan chromosomes are automatically copied to the next generation. Parametrized uniform crossover is used to select the next 79% chromosomes in the next generation. For each chromosome in the next generation, the following is performed. Two chromosomes in the current generation are selected at random. For each job, a random number is generated. If the value is less than 0.7, the value from the “first” chromosome is copied to the new chromosome, otherwise the value from the “second” chromosome is selected. The remaining 1% of the next generation is filled through “immigration”, in which new chromosomes are randomly generated. The above procedures are repeated until 100 iterations have been performed without finding an improved schedule. The values reported here were selected empirically.

3 Parallel Machine Scheduling

This chapter addresses scheduling in parallel machines with sequence-dependent setup times under a variety of conditions. We begin with a general integer programming formulation that incorporates ready times. We then present heuristics for the cases with and without ready times. One heuristic is a simple adaptation of a TSP solution method. The second heuristic is an adaptation of the MULTI-FIT heuristic. The Insertion Heuristic for the TSP is adapted to a multiple machine environment. A genetic algorithm is presented as well. Several test files are created. The performance of each heuristic is evaluated with respect to lower bounds and the heuristics are compared to discover which one performs best on the problem types considered here.

3.1 Integer Programming Model

An integer programming formulation for the $P|r_i, s_{ij} | C_{max}$ problem is presented in this section. First, we will define the variables and then present the formulation with a discussion of the constraints. We conclude this section with a graphical depiction of a potential solution.

Let n be the number of jobs to be scheduled and m be the number of machines where $n \geq m$. We assume that machines are initially setup for a nominal job 0 and must finish setup for a tear down job $n+1$. We have the following definitions.

p_i = the processing time of job i , $i = 1, \dots, n$

r_i = the ready time of job i , $i = 1, \dots, n$

s_{ij} = the sequence-dependent setup time between job i and job j , $i, j \in \{0, 1, \dots, n\}$

$x_{ij} = 1$ if job i is scheduled immediately before job j , else $x_{ij} = 0$, $i, j \in \{0, 1, \dots, n\}$

c_i = the completion time of job i , $i = 1, \dots, n$

z = the makespan of the schedule

$$\min z \quad (3.1)$$

$$\text{s.t.} \quad \sum_{j=1}^n x_{0j} = m \quad (3.2)$$

$$\sum_{j=1}^{n+1} x_{ij} = 1 \quad i = 1, \dots, n \quad (3.3)$$

$$\sum_{i=0}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (3.4)$$

$$c_0 = 0 \quad (3.5)$$

$$c_j \geq p_j + \sum_{i=0}^n x_{ij} (s_{ij} + c_i) \quad j = 1, \dots, n \quad (3.6)$$

$$c_j \geq r_j + p_j + \sum_{i=0}^n x_{ij} s_{ij} \quad j = 1, \dots, n \quad (3.7)$$

$$z \geq c_j \quad j = 1, \dots, n \quad (3.8)$$

$$x_{ij} \in \{0, 1\} \quad i, j \in \{0, 1, \dots, n, n+1\} \quad (3.9)$$

$$x_{ij} = 0 \quad i = j$$

$$c_j \geq 0 \quad j = 0, 1, \dots, n \quad (3.10)$$

Equation (3.1) defines the objective function, which is to minimize the makespan z . We

assume setup times satisfy some sufficient condition to ensure that an optimal solution

will always exist that utilizes all m machines. For instance, $s_{0j} \leq \min_{i \neq 0} (p_i + s_{ij}) \forall j$ and

$$p_j + s_{0j} \leq \frac{\sum_{i \neq j} (p_i + \min_k s_{ki})}{m-1} \quad \forall j \text{ are both sufficient. Constraint set (3.2) ensures that}$$

exactly m machines can be scheduled. Constraint sets (3.3) and (3.4) ensure that each job

is scheduled on one and only one machine. Constraint (3.5) defines the completion time

of the non-existent job 0 so that the recursive constraint sets (3.6) and (3.7) can be used to find completion times of the jobs. Constraint sets (3.6) and (3.7) also ensure that jobs only begin processing after both the previous job and the required setup have been completed. Note that setup for the current job cannot begin until both the current job is available and the preceding job has completed. Constraint set (3.8) links the decision variables c_i and z . Constraint sets (3.9) and (3.10) provide limits on the decision variables. This formulation incorporates the $P||C_{max}$ problem, which is a case of the $Plr_i, s_{ij}|C_{max}$ problem where $r_i=s_{ij}=0$ for all i and j . Since the single machine makespan problem is NP -complete, the $P||C_{max}$ and $Plr_i, s_{ij}|C_{max}$ problems are NP -complete as well.

A feasible solution could be represented graphically as shown in Figure 3. Each node is a job and edges between nodes represent relative ordering on machines. Node 0 must have m edges exiting and node $n+1$ must have m edges entering. In this example, m must be three.

Figure 3: Graphical Representation of Feasible Schedule

3.2 Heuristics with No Ready Times

In this section we describe several heuristics prepared for the problem $P|s_{ij}|C_{max}$.

3.2.1 Slicing Heuristic (SL)

The first idea can be succinctly described as “Cutting up a single machine solution.” The goal is to use a quick method to find a sequence for a single machine problem and quickly slice it up into m pieces. The motivation is that once jobs have been assigned to machines, the jobs should be sequenced optimally. However, finding the optimal combination of job assignments and ordering is very difficult for m machines. Instead, we attempt to find a good sequence for all the jobs and then break the sequence into m pieces. The general algorithm can be described as follows:

1. Find a solution to the single machine problem with makespan M .
2. Break the single machine sequence into m groups, one for each machine.

A target makespan for each machine is calculated as M/m . Each machine starts with job 0. Then, jobs are taken from the single machine solution and added to the current machine until the schedule length of that machine exceeds M/m . At that time, the current machine is “closed” and the next machine is “opened.” This continues until all jobs have been assigned to a machine. The final job on a machine is the last real job, not the tear down job. This job number can be saved and considered the current setup for the machine.

Important considerations include ensuring that all m machines are used, and what to do if this does not happen (or if one machine is used much less than another) and

ensuring that all jobs are assigned to a machine. At this time, if $m' < m$ machines are used, the target is reset to m'/m times the original target. If not all jobs are assigned to machines, then the target is reset to ($\#$ of jobs to be scheduled / $\#$ of placed jobs) times the original target. Let j be the index of the current job being examined and mc be the index of the current machine.

The implemented heuristic is:

1. Use the Doubling Method to solve a TSP and find a solution to the single machine problem. Call the resultant makespan M . The solution gives the job order to be used in Step 5, starting at the first "real" job. That is, skip job 0 which indicates the start of the sequence.
2. Set an approximate target $t = M/m$.
3. Let $mc=1$. Schedule job 0 on this machine. Let $j=1$.
4. If all the jobs have been scheduled, go to Step 7.
5. If machine number mc has a schedule length $< t$, place job j on machine number mc and let $j=j+1$. Go to Step 4.
6. If machine number mc has a schedule length $\geq t$, close machine number mc . Let $mc=mc+1$. Schedule job 0 on this machine. Go to Step 5.
7. If the number of machines used is less than that available ($mc < m$), let

$$t_{new} = t_{old} \frac{mc}{m}.$$
 Unschedule all the jobs and go to Step 3.
8. If more than m machines were used ($mc > m$), let k be the number of jobs placed on the first m machines. Let

$$t_{new} = t_{old} \frac{n}{k}.$$
 Unschedule all the jobs and go to Step 3.
9. If m machines were used, DONE.

3.2.2 Multiple MULTI-FIT Heuristic (MMF)

The second idea to be explored is an adaptation of the MULTI-FIT Algorithm by Coffman, Garey and Johnson (1978). MULTI-FIT does not consider sequence-dependent setup times. However, the setup times can be accounted for by integrating their values into the processing times. Since the actual setup times are unknown until the sequence is determined, we must estimate them. The obvious choices are the minimum, average and maximum setup time. The modified processing times are found using $p_i' = p_i + \text{fun}_j(S_{ij})$, where $\text{fun} \in \{\min, \text{avg}, \max\}$ and fun is referred to as the setup function. MULTI-FIT can then be performed using these modified processing times. Once jobs have been assigned to machines, the true processing and setup times can be used. The Doubling Method can then be used on each machine's sequence to see if a better ordering of those jobs can be found. Otherwise, the modified MULTI-FIT sequence is used. Let k be the maximum number of iterations. The following describes the heuristic.

1. Create the modified processing times $p_i' \forall i$.
2. Order the jobs in non-increasing order of $p_i' \forall i$.
3. Calculate $CL = \max\left(\frac{1}{m} \sum p_i', \max_i p_i'\right)$.
4. Calculate $CU = \max\left(\frac{2}{m} \sum p_i', \max_i p_i'\right)$.
5. $I=1$
6. If $I > k$, STOP. Go to Step 8.
Else, $C=(CL+CU)/2$.
7. Perform the First Fit Decreasing heuristic with the modified processing times.
Let c be the number of machines required.

If $c \leq m$, let $CU=C$ and $CL=CL$.

Else, let $CU=CU$ and $CL=C$.

Let $I = I + 1$, go to Step 6.

8. For each machine, perform the Doubling Heuristic with the true processing and setup times. If the makespan on that machine is reduced, use the Doubling Heuristic ordering.
9. Compute the actual makespan.

3.2.3 Multiple Insertion Heuristic (MI)

The third idea is a multiple machine adaptation of the Insertion Heuristic for TSP.

This approach can also be considered an adaptation of the NEH method (Nawaz *et al.* 1983). Setup times are accounted for by integrating their values into the processing times. As with MMF, the obvious choices are the minimum, average and maximum setup time. The modified processing times are found using the definition

$$p_i' = p_i + \text{fun}(S_{ij}), \text{ where } \text{fun} \in \{\min, \text{avg}, \max\}. \text{ Again, } \text{fun} \text{ is called the setup function.}$$

The Insertion Heuristic can then be performed using these modified processing times.

Once jobs have been assigned to machines, the true processing and setup times can be used. MI has the following steps:

1. Create the modified processing times $p_i' \forall i$.
2. Order the jobs in non-increasing order of $p_i' \forall i$.
3. Examine every job in the order found in step 2. For each job i ,
 - a. insert job i into every position on each machine
 - b. calculate the true partial makespan for each position of job i using the actual setup times

- c. place job i in the position on the machine with the lowest resultant partial makespan

3.2.4 Genetic Algorithm (GA)

Genetic algorithms have been discussed in an earlier section. At this time, we will focus on the particular application of the genetic algorithm in this work.

Based on our earlier discussion, we have a choice between including the job order in the genetic algorithm or not. At this time, we will use just the machine information in the gene (following Gu and Chung 1999) and then solve the TSP using the Doubling Method. In this representation, each gene represents the machine to which the job corresponding to that location has been assigned. For example, a chromosome may be [2, 3, 1, 3, 2, 1], meaning that machine 1 has jobs 3 and 6, machine 2 has jobs 1 and 5, and machine 3 has jobs 2 and 4. This may prevent the genetic algorithm from finding the optimal schedule, because the optimal sequence on a machine may not be the one found by the Doubling Method, even though the optimal assignment of jobs to machines has been made by the genetic algorithm. The tradeoff is between quality of solution and time to find the solution. If too much time spent solving TSPs each iteration, then, and only then, would we include the job order in the gene information. Based on initial experimentation, the Doubling Method has been found to be sufficient.

The population is initialized randomly, so that every job has an equally likely chance of starting out on any one machine. The fitness function is $(C_{\max})^{-1}$, found using the Doubling Method as described previously.

The important parameters for the genetic algorithm are as follows:

POPSIZE	the constant number of chromosomes in the population
PROBMUTATE	the probability that a particular gene undergoes mutation
PROBCROSS	the probability that a particular chromosome is chosen for crossover
NUMITERBASE	the value, which when multiplied by n , yields the number of iterations that will define a stopping criterion
EPSILON	the difference between two successive incumbent solutions that will define a second stopping criterion

The general structure for the genetic algorithm implemented here follows. . Each step is described in more detail in the corresponding figure.

1. Initialize a population of chromosomes. See Figure 4.
2. Evaluate the chromosomes in the population to get each chromosome's fitness. See Figure 5.
3. Find the new incumbent chromosome and consider stopping. If do not stop, go to step 4. See Figure 6.
4. Reproduce to create the next generation, allowing chromosomes with higher fitness values to have higher chances of reproduction. See Figure 7.
5. Apply crossover to generation g . See Figure 8.
6. Apply mutation to generation g . See Figure 9.
7. Go to step 2.

```

Set the generation (iteration) number  $g=0$ .
For  $i=1$  to POPSIZE,
    For  $j=1$  to  $n$ ,
        Let  $\text{Generation}[g].\text{PopMember}[i].\text{gene}[j]$  be a randomly generated integer
         $\in [1, m]$ .
Best.fitness=0, OldBest.fitness=0

```

Figure 4: Population Initialization

```

For  $i=1$  to POPSIZE,
  For  $k=1$  to  $m$ ,
    Select the set of jobs  $J_k$  such that  $\text{Generation}[g].\text{PopMember}[i].\text{gene}[j]=k$ 
      for  $j \in J_k$ 
    Use the Doubling Method for the jobs  $J_k$  to find an order for machine  $k$ 
  Find the makespan  $C_{\max}$  for population member  $i$ 
  Let the fitness be  $\text{Generation}[g].\text{PopMember}[i].\text{fitness}=(C_{\max})^{-1}$ 

```

Figure 5: Chromosome Evaluation

```

NewBest = population member in generation  $g$  with the maximum fitness.
If ( $\text{NewBest.fitness} > \text{Best.fitness}$ ),
  Copy Best to OldBest, Copy NewBest to Best
  If ( $\text{Best.fitness} - \text{OldBest.fitness} < \text{EPSILON}$ ),
    The Best schedule is represented by Best: STOP
If ( $g = \text{NUMITERBASE} * n$ ),
  The Best schedule is represented by Best: STOP
Else  $g=g+1$  and go to step 4.

```

Figure 6: Incumbent and Stopping Criteria

```

TotalFit=0
For  $i=1$  to POPSIZE,
  TotalFit=TotalFit +  $\text{Generation}[g-1].\text{PopMember}[i].\text{fitness}$ 
Sort the population members in generation  $g-1$  in decreasing order of fitness. Let ( $i$ )
  indicate the  $i$ th element.
 $\text{Generation}[g-1].\text{PopMember}[(1)].ub = \text{Generation}[g-1].\text{PopMember}[(1)].\text{fitness} / \text{TotalFit}$ 
For  $i=2$  to POPSIZE,
   $\text{Generation}[g-1].\text{PopMember}[(i)].ub = \text{Generation}[g-1].\text{PopMember}[(i-1)].ub +$ 
     $\text{Generation}[g-1].\text{PopMember}[(i)].\text{fitness} / \text{TotalFit}$ 
For  $i=1$  to POPSIZE,
  Generate a random real number  $s$  between 0 and 1.
  Find the population member ( $l$ ) such that  $\text{Generation}[g-1].\text{PopMember}[(l-1)].ub <$ 
     $s \leq \text{Generation}[g-1].\text{PopMember}[(l)].ub$ 
  Copy  $\text{Generation}[g-1].\text{PopMember}[(l)]$  to  $\text{Generation}[g].\text{PopMember}[i]$ 

```

Figure 7: Reproduction

```

For  $i=1$  to POPSIZE,
  NumToCross=0
  For  $j=1$  to  $n$ ,
    Generate a random real number  $s$  between 0 and 1.
    If ( $s < \text{PROBCROSS}$ )
      Add  $j$  to the set of jobs to be crossed in population member  $i$ ,  $C_i$ 
      NumToCross=NumToCross+1
  If NumToCross is odd,
    Generate a random real number  $s$  between 0 and 1.
    If  $s < 0.5$ ,
      Generate a random integer  $s \in [1, \text{NumToCross}]$ 
      Remove job ( $s$ ) from  $C_i$ 
      NumToCross=NumToCross-1
    Else, while NumToCross is odd
      Generate a random integer  $s \in [1, n]$  until  $s \notin C_i$ 
      Add job  $s$  to  $C_i$ 
      NumToCross=NumToCross+1
  CrossPairs=NumToCross/2
  For  $j=1$  to CrossPairs,
    Generate two random integers  $s$  and  $t$  such that  $s, t \in [1, \text{NumToCross}]$  and
       $s \neq t$ 
    Swap Generation[ $g$ ].PopMember[ $i$ ].gene[ $(s)$ ] and
      Generation[ $g$ ].PopMember[ $i$ ].gene[ $(t)$ ]
    Remove jobs ( $s$ ) and ( $t$ ) from  $C_i$ 
    NumToCross=NumToCross-2

```

Figure 8: Crossover

```

For  $i=1$  to POPSIZE,
  For  $j=1$  to  $n$ ,
    Generate a random real number  $s$  between 0 and 1.
    If ( $s < \text{PROBMUTATE}$ )
      Let Generation[ $g$ ].PopMember[ $i$ ].gene[ $j$ ] be a randomly generated
        integer  $\in [1, m]$ 

```

Figure 9: Mutation

3.2.5 Discussion of Heuristics

This problem has characteristics of both parallel machine scheduling and the traveling salesman problem. The heuristics developed in this section draw their motivation from methods that have proven effective for these two separate problems. The SL and MI procedures derive from the TSP. Both are then modified for parallel machines. MMF is adapted from parallel processor solution procedures but with an added step to address sequence-dependent setups. GA attempts to equally consider both aspects of the problem.

3.3 Heuristics with Ready Times

MI can be applied to the problem $P|r_i, s_{ij}|Cmax$ with no modifications. In addition, GA developed previously will be used with no modifications

3.4 Experimental Design

An experiment was performed to evaluate the heuristics for problem $P|r_i, s_{ij}|Cmax$. The experimental runs contained on the order of 100 jobs to be scheduled on up to 10 machines. Processing times and sequence-dependent setup times were chosen randomly from a uniform distribution as rational values with two decimal places. We first describe the experimental setup.

3.4.1 Ready Times

Each value r_i is either the zero vector (all jobs available at time 0) or each element is drawn from a uniform distribution with bounds a and b . Our intent is to control the standard deviation σ_r relative to the mean. In setting ready times we want to emulate the arrival of jobs. The lower bound is set to zero. The upper bound should be the expected makespan of the (for now nonexistent) preceding stage – the expected completion time for the last job through that stage. This would of course depend on the structure of the preceding stage. Assume that the preceding stage has essentially the same structure as the current stage. The preceding stage's makespan can be roughly estimated by examining the average number of jobs per machine (n_m), the mean of the processing time (μ_p) and the mean of the setup time (μ_s). Every job must be processed and setup. The makespan can be estimated by $n_m(\mu_p + \mu_s)$. Jobs arrive to the current stage as soon as the first job is done, which happens approximately at time $\mu_p + \mu_s$. Thus, jobs arrive at the current stage over a time period with length $n_m(\mu_p + \mu_s) - (\mu_p + \mu_s) = (n_m - 1)(\mu_p + \mu_s)$. Note that one job must arrive at time 0. Since the length is an estimate only, it is not necessary to force a job to arrive at time $(n_m - 1)(\mu_p + \mu_s)$. The ready times are thus generated from a uniform distribution with lower bound zero and upper bound $(n_m - 1)(\mu_p + \mu_s)$, transformed so that one job arrives at time 0.

3.4.2 Setup Times

Each value $s_{ij}, i \neq j$ is drawn from a uniform distribution with standard deviation σ_s and the values s_{ii} are set to a large value. While setup times always satisfy the triangle inequality, they may either be symmetric or asymmetric. Setup times that satisfy the triangle inequality are desired so that the direct path between two jobs is always no longer than any non-direct path between two jobs. Moreover, the SL Heuristic requires the triangle inequality. If a matrix is generated with two values lower than half the largest value in the matrix, then the triangle inequality may not hold for that matrix. To prevent this, the restriction that the lower bound must be at least half the upper bound has been introduced. Let a be the lower bound and b be the upper bound. Then, for uniformly distributed setups, $\mu_s = \frac{a+b}{2}$ and $\sigma_s^2 = \frac{(b-a)^2}{12}$. Solving these two equations yields $a = \mu_s - \sqrt{3}\sigma_s$ and $b = \mu_s + \sqrt{3}\sigma_s$. If the triangle inequality is to hold, then $2a \geq b$ and so $\sigma_s \leq \frac{1}{3\sqrt{3}}\mu_s = \frac{\sqrt{3}\mu_s}{9}$. Thus only the mean must be specified. We select $\mu_s = 450$ so that σ_s is rational.

3.4.3 Processing Times

Each value p_i is drawn from a uniform distribution with standard deviation σ_p . However, the range of the processing times must correspond to the setup times in some way. Following Morris and Tersine (1990), the mean of the processing times can take on

one of two values: $\mu_p = \mu_s$ or $\mu_p = 10\mu_s$. The range of processing times is set at either $[0.94\mu_p, 1.06\mu_p]$ or $[0.4\mu_p, 1.6\mu_p]$.

3.4.4 Place-holder Jobs

Two jobs must be added for implementation reasons. Jobs 0 and $n+1$ are the place-holder jobs. These jobs are basically markers for the beginning and ending of the sequence of jobs. However, they also indicate the initial machine setup and required ending state. If the start and end states of the machines are known, then a job sequence can be found including these details. If the required start and end states are not known, then the place holder jobs can provide substitute information. With the addition of these two jobs, there are $n+2$ jobs, numbered from 0 to $n+1$. Their ready times are zero and their processing times are zero. Their setup times are found using the equations

$$s_{0i} = \max_{\forall j \neq i} \{s_{ij}\} \text{ and } s_{i, n+1} = s_{0i}.$$

The maximum function was chosen so that the makespan would not be underestimated due to the lack of knowledge regarding the required initial and ending machine states.

3.4.5 Problem Data Characterization

Problem data can now be characterized by 7 factors: ready times, range of processing times, mean of processing times, variability of setup times, setup times structure, the number of machines and average number of jobs per machine. Each of these factors is tested at two levels, low and high. The meaning of these levels are shown

in Table 1. Since each factor has 2 levels, there are $2^7 = 128$ possible experimental designs. 10 data sets for each of these cases have been generated.

Table 2: Factor Levels

Factor	Low	High
Ready Times	$r = 0$	$r \sim \text{Unif}(0, (n_m - 1)(\mu_p + \mu_s))$
Range of Processing Times	$p \sim \text{Unif}(0.94\mu_p, 1.06\mu_p)$	$p \sim \text{Unif}(0.4\mu_p, 1.6\mu_p)$
Mean of Processing Times	$\mu_p = \mu_s$	$\mu_p = 10\mu_s$
Setup Times Structure	Symmetric	Asymmetric
Std Dev of Setup Times $S \sim \text{Unif}(\mu_s - \sqrt{3}\sigma_s, \mu_s + \sqrt{3}\sigma_s)$ $\mu_s = 450$	$\sigma_s = \frac{1}{2} \frac{1.5\mu_s}{9}$	$\sigma_s = \frac{1.5\mu_s}{9}$
Number of Machines	2	10
Ave. Number of Jobs per Machine = n_m	3	10

3.5 Identifying Bounds

In this section, we shall characterize problem solutions. Lower bounds are developed as well as a technique for finding optimal solutions for certain problems. We conclude this section with a discussion of the error introduced by using asymmetric setup time matrices.

3.5.1 Lower Bounds

A lower bound on the makespan can be found by looking at the minimum preemptive schedule makespan (McNaughton 1959). We have n jobs that need to be scheduled, each of which must be processed and setup. The minimum setup time for

each job is the lower bound of the setup time distribution, which is $LB_S = \mu_s - \sqrt{3}\sigma_s$.

The minimum processing time is the lower bound of the processing time distribution LB_P .

Thus, a lower bound on the makespan is $LB_{C_{\max}}^{(1)} = n(LB_S + LB_P)/m$.

Especially in cases with the high range of processing times, the general preemptive lower bound can be very poor. We use a preemptive type lower bound for each of the individual data sets using the actual data. Clearly, a job cannot begin until it is ready. Another lower bound on the makespan is the largest ready time plus the processing time of the job with this ready time. We shall use the larger of preemptive type data-dependent lower bound and the largest ready time plus processing time. A data-dependent lower bound for each data set is

$$LB_{C_{\max}}^{(2)} = \max \left(\frac{1}{m} \left\{ \sum_{i=1}^n \left[p_i + \min_{j \in \{1, \dots, n\}} (s_{ji}) \right] \right\}, \max_i \left\{ r_i + \min_{j \in \{1, \dots, n\}} (s_{ji}) + p_i \right\} \right).$$

3.5.2 Scheduling Computational Complexity

The optimal solution can be found for the case of $m=2$. Consider the case with 6 jobs. The first machine could conceivably have anywhere from 1 to 5 jobs assigned to it. Since machines are identical and indistinguishable, assigning 1 job to machine 1 and 5 jobs to machine 2 is equivalent to assigning 5 jobs to machine 1 and 1 job to machine 2. For this reason, machine 1 can have 1, 2, or 3 jobs and machine 2 can have 5, 4, or 3 jobs, respectively. Examine these configurations of assignments:

- 1: machine 1 has 1 job, machine 2 has 5 in the TSP order
- 2: machine 1 has 2 jobs in the TSP order, machine 2 has 4 in the TSP order

3: machine 1 has 3 jobs in the TSP order, machine 2 has 3 in the TSP order

Finding the best makespan over all the configuration types will yield the best schedule for 2 machines and 6 jobs to minimize makespan. The same argument follows for the cases with 2 machines and 20 jobs. However, the number of different configurations is much larger.

The number of configurations in the 6 jobs case is small enough to enable enumeration to find the optimal solution, while the problem with 20 jobs is much too large to permit finding the optimal solution. Consider the cases where only the 10,10 configurations need to be examined. First, there are $\binom{20}{10}$ ways to choose the 10 jobs to go on the first machine. Since there is no difference between the machines, this can be reduced to $\binom{20}{10} \frac{1}{2}$. Then, the TSP solution for each of the machine assignments must be found. Since there are $10!$ ways to order each machine, there are

$\binom{20}{10} \frac{1}{2} 10! 10! = \frac{20!}{2} \approx 1.216 * 10^{18}$ possibilities. Since the jobs could also be distributed in the 9,11 configuration, this configuration would need to be examined as well. This continues for all the required configurations. Thus, even if we could use data based bounds to reduce the number of configurations to be tested, enumeration would still not be feasible.

If there are more than two machines, it is much more difficult to find the optimal makespan. Consider 30 jobs and 10 machines, under the assumption that 3 jobs will be

on each machine in the optimal solution. The first machine could have $\binom{30}{3}$ different sets of 3 jobs assigned to it. The second machine could have $\binom{27}{3}$ different sets of 3 jobs assigned to it and so on. The TSP solution for each of these machines must then be found. This results in $\binom{30}{3} \binom{27}{3} \binom{24}{3} \dots \binom{6}{3} \binom{3}{3} (3!)^{10} = 30!$ possibilities. Moreover, the assumption that the jobs would be evenly distributed among machines is flawed. The jobs could be distributed among the machines in many possible configurations, each of which would have to be considered. Due to this explosion of scenarios that would need to be examined, the optimal solution has only been found for the 2 machine, average 3 jobs per machine cases.

3.5.3 Symmetric Matrix Ceiling

In our intended application, setup time matrices could be asymmetric. Portions of the heuristics assume symmetric matrices. However, if the setup time matrix is asymmetric, it can be replaced by a symmetric matrix with a known error. Consider the following definitions. Let M^* be the length of the TSP tour of the asymmetric setup time matrix. Let Ms^* be the length of the TSP tour of the asymmetric matrix's symmetric replacement M , called the *Symmetric Matrix Ceiling*. We define $M = (M_{ij})$ where

$$M_{ij} = \max(s_{ij}, s_{ji}) \text{ and } E_{ij} = M_{ij} - s_{ij}.$$

Theorem: $Ms^* \leq M^* + \min \left(\sum_i \max_j \{E_{ij}\}, \sum_j \max_i \{E_{ij}\} \right)$

Proof: Replace s_{ij} and s_{ji} by M_{ij} . Each value E_{ij} is either 0 or positive. Consider a feasible TSP solution to M . This solution contains exactly one element from each row and column. Either element (i, j) has $M_{ij} = s_{ij}$ or $M_{ij} = s_{ij} + E_{ij}$. Let $X_{ij} = 1$ if element (i, j) is in the TSP solution and $X_{ij} = 0$ otherwise. A feasible TSP solution on M then has value $Ms^* = \sum_i \sum_j X_{ij} M_{ij} = \sum_i \sum_j X_{ij} \max(s_{ij}, s_{ji})$. Note that $\sum_i \sum_j X_{ij} \max(s_{ij}, s_{ji}) \leq M^* + \sum_i \max_j \{E_{ij}\}$ and

$$\sum_i \sum_j X_{ij} \max(s_{ij}, s_{ji}) \leq M^* + \sum_j \max_i \{E_{ij}\}. \blacksquare$$

3.6 Examples

Consider an example where the processing time range, processing time mean and standard deviation of setup times take on the high levels and the structure of the setup times, number of machines and average number of jobs per machine take on the low levels. The ready times are at the low level, so the ready times are all zero. The actual data are shown in Figure 10. The lower bound for this case is 6555.14. The optimal solution has a makespan of 17865.51. The notation (4/17148.46) shown in Figure 11 and many additional figures indicates the latest completion time of the jobs on that machine and the number of jobs scheduled on that machine. We shall compare the example makespans to the optimal schedule shown in Figure 11.

NumJobs 6					
NumMachs 2					
Process					
4697.51	7019.87	4115.43	6417.20	4114.94	6007.01
Setup					
90000.00	394.26	455.17	447.97	378.30	380.27
394.26	90000.00	412.94	521.70	482.70	390.24
455.17	412.94	90000.00	463.84	425.67	453.40
447.97	521.70	463.84	90000.00	522.35	408.66
378.30	482.70	425.67	522.35	90000.00	396.63
380.27	390.24	453.40	408.66	396.63	90000.00

Figure 10: Sample Data Set

MC 1:	0	2	1	5	(4/17148.46)
MC 2:	0	6	4	3	(4/17865.51)
Assignment's Makespan is 17865.51					

Figure 11: Optimal Schedule

3.6.1 SL Example

The first step in SL involves finding a minimum spanning tree, shown in Figure

12. In this example, Prim’s algorithm was used to find the minimum spanning tree.

Figure 12: Minimum Spanning Tree

The minimum spanning tree is then doubled and is shown in Figure 13.

Figure 13: Doubled Graph

The closed Eulerian trail visits the nodes in the following order: 0, 7, 0, 6, 1, 5, 1, 6, 2, 3, 2, 6, 4, 6, 0. The closed Eulerian trail was found using Fleury's Algorithm and is shown in Figure 14.

Figure 14: Eulerian Trail Edges

The Hamiltonian cycle is constructed by traversing the edges of the closed Eulerian trail in order as long as no ending node of the edge is encountered a second time. If one is, then an edge from the original graph must be added. The edge will start with the last node encountered. The ending node is found by examining the edges in the closed Eulerian graph until a node already encountered is found. This is the edge to be

added to cycle being constructed. Examination of edges in the closed Eulerian trail, with possible edges being added from the original graph, continues until all the nodes in the closed Eulerian trail are in the cycle being constructed, so this cycle is Hamiltonian. The Hamiltonian graph is shown in Figure 15.

Figure 15: Hamiltonian Graph

Since there is no difference between jobs 0 and 7, other than semantic, the Hamiltonian cycle can be thought of as jobs 0, 6, 1, 5, 2, 3, 4, 7 with makespan 34943.41.

The first attempt to place jobs on the two machines has an approximate target of 17471.67. Jobs are added to machine 1 in the order 0, 6, 1, 5, 2, 3, 4. Adding job 2 to machine 1 would have resulted in a schedule length greater than 17471.67. Thus machine 2 begins with job 2. Placing job 4 on machine 2 would have resulted in a schedule length greater than 17471.67 as well. All the jobs could not be placed on the two machine with a target of 17471.67, so the target must be increased as shown in Figure 16.

```

Tried this assignment: (Approx_target is 17471.67)
MC 1:  0   6   1   5 (4/16031.37)
MC 2:  0   2   3   3 (3/12069.94)
Assignment's Makespan is 16031.37
but didn't place all the jobs - Have to adjust target up

```

Figure 16: First SL Attempt

The second attempt to place the jobs on two machines with a target of 20383.61 concluded successfully with an actual makespan of 18950.97. This is approximately 6% larger than the optimal solution. The schedule and a Gantt Chart are shown in Figure 17.

Figure 17: Final Schedule from SL

3.6.2 MMF Example

The first step in MMF involves finding the modified processing times and sorting the real jobs (1-6) in non-increasing order. In this example, the minimum setup time will be used to modify the processing times. The new order and the modified processing times are shown in Table 3. Now, traditional MULTI-FIT is performed but the makespans are computed using the actual processing and setup times. Five iterations were performed resulting in the machine assignments in Table 4. If the machine assignment results in too many machines being used, the makespan value is shown as 'X'.

Table 3: New Job Order and Modified Processing Times

Job	Modified Processing Time
2	7410.10
4	6825.86
6	6387.26
1	5075.79
3	4528.37
5	4493.22

Table 4: MMF Iterations

Iteration	(Lower Bound, Upper Bound)	Machine Assignments	Makespan
1	(17360.30, 34720.60)	MC 1: 0, 2, 4, 6, 1 MC 2: 0, 3, 5	25973.89
2	(17360.30, 26040.45)	MC 1: 0, 2, 4, 6 MC 2: 0, 1, 3, 5	20896.13
3	(17360.30, 21700.38)	MC 1: 0, 2, 4, 3 MC 2: 0, 6, 1, 5	19059.73
4	(17360.30, 19530.34)	MC 1: 0, 2, 4 MC 2: 0, 6, 1, 3 MC 3: 0, 5	X
5	(18445.32, 19530.34)	MC 1: 0, 2, 4 MC 2: 0, 6, 1, 3 MC 3: 0, 5	X

Clearly, the machine assignments in iterations 4 and 5 are invalid since 3 machines are required. The assignment with the lowest makespan was found in Iteration 3. Using this assignment, the Doubling Heuristic is applied to see if improvement is possible. It was, as shown in Figure 18. Note that the final schedule is the same as for SL, at approximately 6% larger than the optimal. Figure 17 shows the Gantt Chart of this schedule.

Doubling gave better				
After trying to improve with Doubling Heuristic:				
MC 1:	0	3	2	4 (4/18950.97)
MC 2:	0	6	1	5 (4/16031.37)
Assignment's Makespan is 18950.97				

Figure 18: Assignment from MMF

3.6.3 MI Example

The first step in MI involves finding the modified processing times and sorting the real jobs (1-6) in non-increasing order, just as in MMF. Again, the minimum setup time will be used to modify the processing times. The new order and the modified processing times are shown in Table 3. Now each job is considered in the order 2, 4, 6, 1, 3, 5. Table 5 shows the machine assignments as each job is added. Recall that both machines must start with job 0. The final schedule is found and is shown in Figure 19. This solution is preferable to those found by SL and MMF and is approximately 0.007 % larger than the optimal value of 17865.51.

Table 5: Partial Machine Schedules

Current Job	Machine Assignments
	MC 1: 0 MC 2: 0
2	MC 1: 0, 2 MC 2: 0
4	MC 1: 0, 2 MC 2: 0, 4
6	MC 1: 0, 2 MC 2: 0, 6, 4
1	MC 1: 0, 1, 2 MC 2: 0, 6, 4
3	MC 1: 0, 1, 2, 3 MC 2: 0, 6, 4
5	MC 1: 0, 1, 2, 3 MC 2: 0, 5, 6, 4

MC 1: 0 1 2 3 (4/17095.17)
MC 2: 0 5 6 4 (4/17866.76)
Assignment's Makespan is 17866.76

Figure 19: Schedule from MI

3.6.4 GA Example

GA was applied to the same data set as well. The final schedule is shown in Figure 20. Note that this makespan is the optimal value, even though the jobs on machine 1 are not in the exact same order. That is because this is a symmetric case.

Figure 20: Schedule from GA

3.7 Comparing Heuristics without Ready Times

In this section, we shall refine the heuristics introduced previously as well as compare the refined heuristics. We conduct an empirical evaluation of the heuristics, using statistical methods to evaluate the results.

Recall that minimizing the makespan is the nominal objective. ANOVAs and boxplots use the base 10 log of the makespans divided by the data-dependent lower bounds. Other tables use just the makespan divided by the data-dependent lower bound. Additional comparisons for cases where the optimal solution was attainable have been included. In ANOVAs, the main effects are shown, as well as the interactions with P values less than or equal to 0.05.

SL must find a minimum spanning tree (MST), which can be done with either Prim's or Kruskal's algorithm. We can also examine the levels for the factors chosen at this time. Appendix A contains an ANOVA table for SL with the MST algorithm and data generation factors. Table 6 contains the values for the main effects and the error. The MST algorithm (MST_ALGO) is significant as seen in Table 6, and the Prim algorithm is better, as can be seen in Figure 21. The time complexity of Prim's algorithm is lower as well, so it will be used. The Setup Matrix Structure (SETUP_STRUCT) and Number of Machines (NUM_MC) factors were insignificant in this analysis. Since the Setup Matrix Structure tells us whether or not the Doubling Method could be applied without manipulating the data first, it has been continued as a factor. The Number of Machines is insignificant in this heuristic, which makes sense. The difficult part of this heuristic is in finding a TSP cycle, not in slicing the cycle into parts. Since the Number of Machines is significant in the ANOVAs of the other heuristics, it has been continued. Prim's algorithm will be used for SL and the factor levels will not be adjusted.

Table 6: SL ANOVA – Main Effects

Source	Sum of Squares	df	Mean Square	F-ratio	P
MST_ALGO	0.180	1	0.180	12.210	0.000
PROC_RANGE	0.227	1	0.227	15.406	0.000
PROC_MEAN	0.130	1	0.130	8.816	0.003
SETUP_STRUCT	0.003	1	0.003	0.200	0.655
SETUP_STDDEV	0.057	1	0.057	3.896	0.049
NUM_MC	0.025	1	0.025	1.676	0.196
AVG_NUM_JOBS	0.758	1	0.758	51.501	0.000
Error	16.950	1152	0.015		

Least Squares Means **Figure 21: Least Square Means for MST_ALGO Factor**

MMF adds some function of the setup times to the processing time before performing regular MULTI-FIT. In this section, we will decide whether to use the

minimum, average or maximum setup time. Appendix B contains an ANOVA table for MMF with the setup function and data generation factors. Table 7 contains the values for the main effects and the error. The setup function (SETUP_FUN) is significant as seen in Table 7. The minimum function is best, as can be seen in Figure 22, even though on a case by case basis it is not the best function. The minimum function is best 405 times out of 640 cases. The minimum setup time will be used.

Table 7: MMF ANOVA – Main Effects

Source	Sum of Squares	df	Mean Square	F-ratio	P
SETUP_FUN	0.017	2	0.008	167.686	0.000
PROC_RANGE	0.035	1	0.035	709.262	0.000
PROC_MEAN	0.260	1	0.260	5220.344	0.000
SETUP_STRUCT	0.001	1	0.001	27.974	0.000
SETUP_STDDEV	0.079	1	0.079	1587.582	0.000
NUM_MC	0.040	1	0.040	797.688	0.000
AVG_NUM_JOBS	0.009	1	0.009	189.891	0.000
Error	0.086	1728	0.000

Figure 22: Least Square Means for SETUP_FUN Factor for MMF

MI also adds some function of the setup times to the processing time before performing the multiple insertion. Appendix C contains an ANOVA table for MI with the setup function and data generation factors. Table 8 contains the values for the main effects and the error. The setup function (SETUP_FUN) is insignificant as seen in Table 8. In order to compare MMF and MI, the minimum setup time will be used for MI as well.

Table 8: MI ANOVA – Main Effects

Source	Sum of Squares	df	Mean Square	F-ratio	P
SETUP_FUN	0.000	2	0.000	0.743	0.476
PROC_RANGE	0.008	1	0.008	410.566	0.000
PROC_MEAN	0.073	1	0.073	3701.615	0.000
SETUP_STRUCT	0.001	1	0.001	62.814	0.000
SETUP_STDDEV	0.023	1	0.023	1158.658	0.000
NUM_MC	0.004	1	0.004	221.971	0.000
AVG_NUM_JOBS	0.060	1	0.060	3055.707	0.000
Error	0.034	1728	0.000		

GA has several parameters that need to be tuned. After initial experimentation, the following parameters were found:

Table 9: GA Parameter Values

Parameter	Value
POPSIZE	20
PROBMUTATE	0.5
PROBCROSS	0.25
NUMITERBASE	20
EPSILON	0.000001

Recall that the NUMITERBASE is multiplied by the number of jobs to be scheduled to yield the true number of iterations. These parameters were chosen so that GA found many optimal solutions in the six job case.

3.7.1 Comparison to Data-dependent Lower Bounds

Now that the heuristics have been refined, they can be compared. Appendix D contains an ANOVA table for the non-ready time heuristics with the heuristic used and data generation factors. Table 10 contains the values for the main effects and the error. The heuristic used (ALGO) is significant as seen in Table 10. The heuristics are denoted

as follows in the Least Squares Means graph in Figure 23: SL=1, MMF=2, MI=3 and GA=4. Level 3, corresponding to MI, is best, as can be seen in Figure 23.

Table 10: ANOVA – No Ready Time Heuristics

Source	Sum of Squares	df	Mean Square	F-ratio	P
ALGO	0.317	3	0.106	836.382	0.000
PROC_RANGE	0.044	1	0.044	345.532	0.000
PROC_MEAN	0.159	1	0.159	1258.078	0.000
SETUP_STRUCT	0.000	1	0.000	0.359	0.549
SETUP_STDDEV	0.042	1	0.042	336.165	0.000
NUM_MC	0.341	1	0.341	2658.875	0.000
AVG_NUM_JOBS	0.224	1	0.224	1770.202	0.000
Error	0.291	2304	0.000

Figure 23: Least Square Means for ALGO Factor – No Ready Times

Several tables with additional data have been compiled to support this conclusion. The average value for the makespan divided by the data-dependent lower bounds for each heuristic has been summarized in Table 11. Note that for many cases, GA outperforms MI even though the ANOVA indicated that MI was better. In general, GA performs well in cases with 6 or 20 jobs and poorly in cases with 30 or 100 jobs. This may be due to the fact that the number of generations allowed depends on the number of jobs in a way that is too small; meaning that while 20x6 generations may be enough for the 6 job case, 20x100 is not enough generations for the 100 job case. The average value for the makespan divided by the data-dependent lower bounds for each heuristic are also summarized in Figure 24. Each data point is the average over all cases with the shown levels for Range of Processing Times, Standard Deviation of Setup Times, Number of Machines and Average Number of Jobs.

Table 11: Average MS/LB – No Ready Times

Format of each cell: SL, MMF, MI, GA

			Range of Processing Times							
			Low				High			
			Mean of Processing Times				Mean of Processing Times			
			Low		High		Low		High	
			Setup Times Structure		Setup Times Structure		Setup Times Structure		Setup Times Structure	
			Low	High	Low	High	Low	High	Low	High
Setup Times Std Dev Low	# of MC Low	Avg # of Jobs Low	1.078	1.042	1.017	1.013	1.175	1.117	1.127	1.135
			1.043	1.041	1.025	1.025	1.088	1.114	1.059	1.061
			1.031	1.030	1.008	1.007	1.039	1.047	1.051	1.025
			1.028	1.024	1.007	1.006	1.037	1.044	1.031	1.016
		Avg # of Jobs High	1.036	1.032	1.010	1.013	1.052	1.054	1.045	1.034
			1.038	1.055	1.031	1.032	1.054	1.069	1.021	1.031
			1.018	1.017	1.004	1.003	1.022	1.018	1.006	1.006
			1.015	1.020	1.003	1.004	1.016	1.024	1.003	1.005
	# of MC High	Avg # of Jobs Low	1.063	1.064	1.036	1.055	1.178	1.232	1.203	1.208
			1.066	1.070	1.031	1.031	1.129	1.131	1.066	1.070
			1.047	1.043	1.010	1.010	1.056	1.062	1.040	1.048
			1.362	1.338	1.286	1.312	1.211	1.246	1.179	1.177
		Avg # of Jobs High	1.055	1.052	1.049	1.021	1.083	1.080	1.057	1.066
			1.059	1.065	1.057	1.058	1.080	1.086	1.029	1.030
			1.023	1.020	1.003	1.003	1.019	1.017	1.006	1.006
			1.127	1.136	1.100	1.114	1.107	1.122	1.095	1.092
Setup Times Std Dev High	# of MC Low	Avg # of Jobs Low	1.090	1.076	1.022	1.025	1.141	1.112	1.205	1.141
			1.087	1.086	1.027	1.026	1.119	1.120	1.064	1.066
			1.067	1.064	1.013	1.013	1.088	1.062	1.045	1.028
			1.056	1.054	1.011	1.012	1.071	1.073	1.033	1.020
		Avg # of Jobs High	1.060	1.092	1.012	1.015	1.072	1.082	1.035	1.050
			1.105	1.108	1.040	1.042	1.104	1.100	1.032	1.035
			1.041	1.038	1.007	1.006	1.042	1.033	1.010	1.008
			1.030	1.042	1.007	1.008	1.035	1.042	1.007	1.009
	# of MC High	Avg # of Jobs Low	1.125	1.131	1.039	1.041	1.265	1.275	1.228	1.200
			1.123	1.126	1.045	1.052	1.188	1.191	1.079	1.079
			1.105	1.094	1.017	1.017	1.102	1.095	1.060	1.041
			1.399	1.372	1.297	1.298	1.287	1.293	1.191	1.197
		Avg # of Jobs High	1.095	1.094	1.028	1.026	1.103	1.115	1.071	1.060
			1.093	1.115	1.056	1.057	1.118	1.123	1.032	1.034
			1.045	1.044	1.007	1.006	1.039	1.036	1.010	1.010
			1.155	1.174	1.106	1.110	1.146	1.167	1.087	1.099

Figure 24: Makespan Divided by Lower Bounds with No Ready Times

In order to more fully examine the different performances of these two heuristics, Table 12 was formed. Table 12 shows the number of times each heuristic had the lowest makespan. Note that the values in each cell may be more than 10 because the heuristics could have found the same solution. Again in Table 12, we see that, while GA could be considered competitive in the 6 and 20 job cases, MI dominates in the 30 and 100 job cases. In general, GA performs well in cases with the low level for Number of Machines and poorly in cases with the high level for Number of Machines. Out of 640 trials, MI has the lowest makespan of the four heuristics 446 times while GA has the lowest makespan only 207 times.

Table 12: Number of Times MI and GA were Lowest – No Ready Times

Format of each cell: MI, GA

			Range of Processing Times							
			Low				High			
			Mean of Processing Times				Mean of Processing Times			
			Low		High		Low		High	
			Setup Times Structure		Setup Times Structure		Setup Times Structure		Setup Times Structure	
			Low	High	Low	High	Low	High	Low	High
Setup Times Std Dev Low	# of MC Low	Avg # of Jobs Low	5 5	3 7	5 9	5 6	4 8	5 7	2 9	3 7
		Avg # of Jobs High	0 10	9 1	1 8	8 2	0 10	9 1	1 9	5 5
	# of MC High	Avg # of Jobs Low	8 0	10 0	10 0	10 0	10 0	10 0	10 0	9 0
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
Setup Times Std Dev High	# of MC Low	Avg # of Jobs Low	3 10	2 8	2 9	4 8	4 8	7 4	4 8	6 4
		Avg # of Jobs High	0 10	7 3	2 8	9 1	1 9	10 0	0 10	7 3
	# of MC High	Avg # of Jobs Low	9 0	10 0	10 0	10 0	10 0	10 0	8 0	9 0
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0

In comparison to the data-dependent lower bounds, based on the above statistics,

MI is preferable.

3.7.2 Comparison to Optimal

The number of times each heuristic achieved the optimal makespan, for cases with 2 machines and 6 jobs, has been summarized in Table 13. The order of each cell's data is SL, MMF, MI, and GA. MI achieves the optimal makespan 37 times in the 160 data sets examined. GA achieves the optimal makespan 78 times. Moreover, SL only achieves the optimal makespan 6 times and MMF only achieves the optimal makespan twice. The ratio between the makespan and the optimal solution for these cases has been summarized in Figure 25. Each data point is the average over all cases with the shown levels for Range of Processing Times, Standard Deviation of Setup Times, Number of Machines and Average Number of Jobs. GA has the lowest ratio in all but one case. The average and maximum ratio of achieved makespan to optimal makespan for cases with 2 machines and 3 jobs, has been summarized in Table 14. GA has the lowest ratio at 1.047, followed by MI at 1.079. In comparison to the optimal solutions, GA appears preferable. However, the added element of running time must be considered also.

Table 14: Ratios of Achieved Makespan to Optimal Makespan – No Ready Times

Heuristic	Average Makespan divided by Optimal	Maximum Makespan divided by Optimal
SL	1.063	1.621
MMF	1.036	1.152
MI	1.009	1.079
GA	1.003	1.047

3.7.3 Running Time Performance

Each heuristic was written in C, compiled with Microsoft Visual C++ and run on a PC with Pentium II 200 MHz processor and 64 MB RAM. The running times were found using the clock() function and the averages over the ten data files for each case have been tabulated in Appendix H. If the average time was less than 0.01 seconds, a dash has been used. Again, MI performed at least as well as the other heuristics in all cases on average. GA performed poorly on this measure, taking 100 to 1000 times longer than the next slowest alternative. In the 30 and 100 job cases, these long running times cannot be considered in light of the quality of the solution, because the solutions were not very good.

Using many different measures, MI has been found to be best for the types of problems discussed here. While GA is attractive in some instances, its overall performance, especially in light of its running time, is inferior to MI.

3.8 Comparing Heuristics with Ready Times

In this section, we shall compare MI and GA, the two heuristics that apply to the case of ready times. We conduct an empirical evaluation of the heuristics, using statistical methods to evaluate the results.

3.8.1 Comparison to Data-dependent Lower Bounds

The heuristic used (ALGO) is seen to be a significant factor at the 0.001 level in Table 17. The designation 5 is used to MI and the designation 6 is used for GA in Figure 26, which shows MI (5) to be better.

Table 16: ANOVA - Ready Time Heuristics

Source	Sum of Squares	df	Mean Square	F-ratio	P
ALGO	2.674	1	2.674	1139.089	0.000
Error	3.000	1278	0.002

Figure 26: Least Square Means for ALGO Factor – Ready Times

Several tables with additional data have been compiled to support this conclusion. The average value for the makespan divided by the data-dependent lower bounds for each heuristic has been summarized in Table 17. Note that for all cases, MI outperforms GA. Figure 27 graphically depicts GA's performance versus MI's. Each data point is the average over all cases with the shown levels for Range of Processing Times, Standard Deviation of Setup Times, Number of Machines and Average Number of Jobs.

Table 17: Average MS/LB – Ready Times

Format of each cell: MI, GA

			Range of Processing Times							
			Low				High			
			Mean of Processing Times				Mean of Processing Times			
			Low		High		Low		High	
			Setup Times Structure		Setup Times Structure		Setup Times Structure		Setup Times Structure	
			Low	High	Low	High	Low	High	Low	High
Setup Times Std Dev Low	# of MC Low	Avg # of Jobs Low	1.057	1.057	1.116	1.111	1.061	1.056	1.138	1.091
			1.257	1.179	1.254	1.202	1.120	1.165	1.242	1.204
		Avg # of Jobs High	1.147	1.120	1.154	1.156	1.100	1.108	1.105	1.135
			1.396	1.356	1.419	1.386	1.300	1.332	1.315	1.427
	# of MC High	Avg # of Jobs Low	1.150	1.154	1.234	1.241	1.119	1.133	1.152	1.176
			1.405	1.416	1.450	1.489	1.397	1.412	1.412	1.451
Setup Times Std Dev High	# of MC Low	Avg # of Jobs Low	1.150	1.144	1.133	1.140	1.123	1.128	1.114	1.117
			1.648	1.647	1.635	1.639	1.638	1.654	1.663	1.636
		Avg # of Jobs High	1.124	1.090	1.131	1.120	1.120	1.095	1.070	1.102
			1.145	1.186	1.264	1.221	1.127	1.194	1.165	1.205
	# of MC High	Avg # of Jobs Low	1.156	1.158	1.134	1.138	1.150	1.126	1.111	1.118
			1.383	1.448	1.401	1.409	1.369	1.310	1.385	1.373
	# of MC Low	Avg # of Jobs Low	1.180	1.178	1.240	1.233	1.173	1.164	1.153	1.167
			1.463	1.487	1.482	1.495	1.441	1.457	1.438	1.481
	# of MC High	Avg # of Jobs Low	1.155	1.161	1.151	1.158	1.150	1.156	1.125	1.088
			1.687	1.708	1.672	1.675	1.678	1.685	1.648	1.630

Figure 27: Makespan Divided by Lower Bounds with Ready Times

In order to more fully examine the different performances of these two heuristics, Table 18 was formed. Table 18 shows the number of times each heuristic had the lowest makespan. Note that the values in each cell may be more than 10 because the heuristics could have found the same solution. Again in Table 18, we see that MI dominates. Out of 640 trials, MI has the lowest makespan 610 times while GA has the lowest makespan only 30 times.

Table 18: Number of Times MI and GA were Lowest – Ready Times

Format of each cell: MI, GA

			Range of Processing Times							
			Low				High			
			Mean of Processing Times				Mean of Processing Times			
			Low		High		Low		High	
			Setup Times Structure		Setup Times Structure		Setup Times Structure		Setup Times Structure	
			Low	High	Low	High	Low	High	Low	High
Setup Times Std Dev Low	# of MC Low	Avg # of Jobs Low	9 1	7 3	7 3	9 1	8 2	9 1	10 0	9 1
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
	# of MC High	Avg # of Jobs Low	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
Setup Times Std Dev High	# of MC Low	Avg # of Jobs Low	8 2	8 2	6 4	9 1	4 6	9 1	9 1	9 1
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
	# of MC High	Avg # of Jobs Low	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0
		Avg # of Jobs High	10 0	10 0	10 0	10 0	10 0	10 0	10 0	10 0

In comparison to the data-dependent lower bounds, MI is preferable.

3.8.2 Comparison to Optimal

The number of times each heuristic achieved the optimal makespan, for cases with 2 machines and 6 jobs, has been summarized in Table 19. MI achieves the optimal makespan 45 times in the 160 data sets examined. GA achieves the optimal makespan 17

times. The ratio between the makespan and the optimal solution for these cases has been summarized in Figure 28. Each data point is the average over all cases with the shown levels for Range of Processing Times, Standard Deviation of Setup Times, Number of Machines and Average Number of Jobs. MI has the lowest ratio in all cases. The average and maximum ratios of achieved makespan to optimal makespan for cases with 2 machines and 3 jobs has been summarized in Table 20. In comparison to the optimal solutions, MI is preferable.

Table 19: Number of Times Each Heuristic Hit Optimal – Ready Times

(Format of each cell: MI, GA)

			Range of Processing Times							
			Low				High			
			Mean of Processing Times				Mean of Processing Times			
			Low		High		Low		High	
			Setup Times Structure		Setup Times Structure		Setup Times Structure		Setup Times Structure	
			Low	High	Low	High	Low	High	Low	High
Setup Times Std Dev Low	# of MC Low	Avg # of Jobs Low	1	3	2	0	6	5	4	5
			0	2	3	0	2	1	0	1
Setup Times Std Dev High	# of MC Low	Avg # of Jobs Low	1	1	0	4	2	5	3	3
			2	0	1	0	3	0	1	1

Figure 28: Average Makespan Divided by Optimal – Ready Times

Table 20: Ratios of Achieved Makespan to Optimal Makespan

Heuristic	Average Makespan Divided by Optimal	Maximum Makespan Divided by Optimal
MI	1.023	1.232
GA	1.116	1.408

Using many different measures, MI has been found to be best for the types of problems discussed here.

3.9 Conclusions

This work has discussed scheduling up to 100 jobs on up to 10 parallel machines to minimize makespan. Sequence-dependent setup times were included, leading to an NP-hard problem. Two varieties of problems were considered: no ready times and ready times.

In the no ready time case, four heuristics were empirically compared with each other and in some cases with the optimal schedule. Problem data was characterized by range and mean of processing times, structure of setup time matrices, variance of setup times, number of machines and the average number of jobs per machine. Using many different measures, MI has been found to be best for the types of problems discussed here. While GA is attractive in some instances, its overall performance, especially in light of its running time, is inferior to MI. This implies that GA's design is flawed, as the increased running time and the motivation for genetic algorithms in general indicate that the methodology should be superior to, or at least on a par with, a simpler heuristic.

In the ready time case, two heuristics were empirically compared. Using many different measures, MI has been found to be best for the types of problems discussed here as well.

4 Multiple Stages

This chapter considers scheduling in flexible flow lines to minimize makespan. First an integer programming model is presented and described. A discussion of permutation schedules for flexible flow lines follows. Lower bounds are developed for use in evaluating schedules produced with the ten heuristics described. The ten heuristics comprise four classes of heuristics. One class is a naïve group that simply assigns jobs to machines in a greedy fashion. The second class expands on the Multiple Machine Insertion Heuristic used in the previous chapter, in order to take advantage of the sequence-dependent nature of the setup times. The third class is based on Johnson's Rule. The fourth class contains two versions of genetic algorithms. Examples of the ten heuristics are provided. Using randomly generated data and lower bounds, the heuristics are compared with each class to find a representative. The four representatives of each class are then compared. The quality of the lower bounds is also discussed.

4.1 Integer Programming Model

Let g be the number of stages. Let n be the number of jobs to be scheduled and m^t be the number of machines at stage t . We assume that machines are initially setup for a nominal job 0 and must finish setup for a tear down job $n+1$ at every stage. We have the following definitions.

n = number of true jobs to be scheduled
 g = number of serial stages
 g_j = last stage visited by job j

m^t = number of machines at stage t

p_i^t = processing time for job i at stage t

c_i^t = completion time for job i at stage t

s_{ij}^t = setup time from job i to job j stage t

x_{ij}^t = 1 if job i is scheduled immediately before job j at stage t and 0 otherwise

S_i = set of stages visited by job i

S^t = set of jobs that visit stage $t = \{i : p_i^t > 0\}$

The processing times of jobs 0 and $n+1$ are set at 0 and the setup times are times to move from and to the nominal set point state. We assume that all jobs currently in the system must be completed at each stage before the jobs under consideration may begin setup.

The completion times of job 0 at each stage are set to the earliest time setup may begin at that stage. We include the restriction that every stage must be visited by at least as many jobs as there are machines in that stage. This is expressed by the inequality

$|S^t| \geq m^t, t = 1, 2, \dots, g$, so $n \geq \max_{t=1,2,\dots,g} \{m^t\}$. The formulation also assumes that a job that

does not visit a stage has a processing time of 0. That is, $p_i^t = 0$ if $i \notin S^t$. The

formulation becomes

$$P: \min z \quad (4.1)$$

$$\text{s.t.} \quad \sum_{j=1}^n x'_{oj} = m^t \quad t=1, \dots, g \quad (4.2)$$

$$\sum_{j \in \{S^t, n+1\}} x'_{ij} = 1 \quad i = 1, \dots, n, t \in S_i \quad (4.3)$$

$$\sum_{i \in \{0, S^t\}} x'_{ij} = 1 \quad j = 1, \dots, n, t \in S_i \quad (4.4)$$

$$c'_j - c'_i + M^t(1 - x'_{ij}) \geq s'_{ij} + p'_j \quad i = 0, \dots, n, j = 1, \dots, n, t \in S_i \quad (4.5)$$

$$c'_j - c'_{j-1} + M^t(1 - x'_{ij}) \geq s'_{ij} + p'_j \quad i = 0, \dots, n, j = 1, \dots, n, t \in S_i - \{1\} \quad (4.6)$$

$$x'_{ij} \leq p'_j \quad i, j \in \{0, 1, \dots, n, n+1\}, t=1, \dots, g \quad (4.7)$$

$$x'_{ji} \leq p'_j \quad i, j \in \{0, 1, \dots, n, n+1\}, t=1, \dots, g$$

$$c'_j - c'_0 \geq M^t p'_j \quad j = 1, \dots, n \quad (4.8)$$

$$c'_j - c'_{j-1} \geq M^t p'_j \quad j = 1, \dots, n, t=2, \dots, g$$

$$c'_j \geq c'_0 \quad j = 1, \dots, n, t=1, \dots, g \quad (4.9)$$

$$z \geq c'^g_j \quad j = 1, \dots, n \quad (4.10)$$

$$x'_{ij} \in \{0, 1\} \quad i, j \in \{0, 1, \dots, n, n+1\}, t=1, \dots, g \quad (4.11)$$

$$x'_{ij} = 0 \quad i = j, t=1, \dots, g$$

$$c'_j \geq 0 \quad j = 0, \dots, n, t=1, \dots, g \quad (4.12)$$

This formulation is based on the TSP. Each stage exists independently except that stage t 's completion times are stage $t+1$'s ready times. Equation (4.1) defines the objective function, which is to minimize the makespan z . We assume that setup times satisfy some condition sufficient to ensure that an optimal solution will always exist that utilizes all m^t machines at each stage, as in the single stage case. For instance,

$$s'_{0j} \leq \min_{i \neq 0} (p'_i + s'_{ij}) \quad \forall j \text{ and } p'_j + s'_{0j} \leq \frac{\sum_{i \neq j} (p'_i + \min_k s'_{ki})}{m^t - 1} \quad \forall j \text{ are both sufficient.}$$

Constraint set (4.2) ensures that m^t machines are scheduled in each stage. Constraint sets (4.3) and (4.4) ensure that each job is scheduled on one and only one machine in each

stage. Constraint set (4.5) forces job j to follow job i by at least i 's processing time plus the setup time from i to j if i is immediately before j . The value M^t is an upper bound on the time stage t completes processing, similar to the upper bound A in Rios-Mercado and Bard (1998). $M^1 = \sum_{i=1}^n \left(p_i^1 + \max_{j \in \{0, \dots, n\}} s_{ji}^1 \right)$ and $M^t = M^{t-1} + \sum_{i=1}^n \left(p_i^t + \max_{j \in \{0, \dots, n\}} s_{ji}^t \right)$. Constraint set (4.6) forces job j at stage t to complete after the following three actions are completed: job j completes at stage $t-1$, job j completes processing at stage t , setup from job j 's predecessor to j . The value M_j^t is set to $M_j^t = \max_i (s_{ij}^t) + p_j^t$. Constraint sets (4.5) and (4.6) together ensure that a job cannot begin setup until it is available (done at the previous stage) and the previous job at the current stage is complete. Constraint sets (4.5) and (4.6) also serve as sub-tour elimination constraints. Constraint set (4.7) ensures that jobs that do not visit a stage are not assigned to that stage. Constraint sets (4.8) and (4.9) ensure that the completion time of a job at stage t , that does not visit stage t , is set to the job's completion time at stage $t-1$. The value M^t is an upper bound on the time stage t completes processing and is the same as that used in constraint set (4.5). Constraint set (4.10) links the decision variables c_j^s and z . Constraint sets (4.11) and (4.12) provide limits on the decision variables.

Due to the fact that scheduling the jobs assigned to each machine at each stage is a TSP, this problem is NP-hard. In this research, we need not only sequence jobs on machines, we must consider which jobs are to be assigned to the machines. Heuristic approaches will therefore be used in this research.

4.2 Permutation Schedules

In flowshop scheduling to minimize makespan, permutation schedules are often required. These are schedules in which the jobs visit every stage in the same order so that a permutation of the job numbers completely specifies a semi-active schedule.

Permutation schedules have been shown to be optimal in several specific cases. First, they are optimal in the case of two or three machines. Corwin and Esogbue (1974) discuss two stage flow shops where either the first or second stage machine has sequence-dependent setup times and report that there exists a permutation schedule with the optimal makespan in this situation. The essence of the proof is that the schedule of the non-sequence-dependent stage can be changed to be the same as that in the sequence-dependent stage without increasing the makespan. In the case of sequence-dependent setup times at both stages of a two stage flow shop, Gupta and Darrow (1986) show by counterexample that permutation schedules are not necessarily optimal. In all these cases, only one machine exists at each stage. However, this research addresses the more general situation with multiple machines allowed at stages.

Consider the situation with two machines at one stage. The first question that arises is how an assignment of jobs to machines will be written in this context. We will number machines in the order of the smallest job number on that machine. This is best explained by way of multiple examples. For example, consider the job assignment and resultant machine numbering in Table 21. In this case, the lowest scheduled job is job 2, so its machine is machine 1. The lowest scheduled job not on machine 1 is job 4, so its machine is machine 2. Since we have only two machines in this example, we could have

known that machine 2 was so named before examining its jobs, but the method is shown for completeness. The job assignment is written $(2,3,7 / 6,5,4)$. Consider a second example (Table 22) in which job 1 is scheduled on the machine with jobs 6, 5, and 4. Now, the machine numbers change since job 1 is scheduled on what was machine 2, but our convention dictates that this machine is now machine 1. This job assignment is written $(6,5,4,1 / 2,3,7)$. This can be done at every stage. For example, $\{(6,5,4,1 / 2,3,7) / (6,5,4,1 / 2,3,7)\}$ indicates that two stages exist, each with two machines and the jobs are assigned in the same order to the machines with the same name in each stage.

Table 21: Machine Numbering Example 1

Job List	2,3,7	6,5,4
Machine Number	1	2

Table 22: Machine Numbering Example 2

Job List	2,3,7	6,5,4,1
Machine Number	2	1

When the number of machines and the set of jobs to be scheduled are the same at every stage, the concept of a permutation schedule can be extended to the multiple machine case. A permutation schedule would be one in which jobs are assigned to the “same” machines in each stage in the same order. The question arises as to whether or not the set of permutation schedules contains the optimal schedule in this case. The first case has no sequence-dependent setup times, the second has sequence-dependent setup times at only one stage, and the third has sequence-dependent setup times at both stages.

4.2.1 Counterexample 1: No Sequence-dependent Setup Times

With the processing times as shown in Table 23, the following schedule is optimal: $\{(1/3, 2)/(2, 1/3)\}$ with a makespan of 12. This makespan matches the lower bound formed from the largest sum of processing times across all stages, which is 12 and achieved by both jobs 1 and 3. The schedule is shown in Figure 29. Each box is labeled by the job number and the time axis is the bottom line. The six permutation schedules and their makespans are shown in Table 24.

Table 23: Processing Times

Job	Stage	
	1	2
1	11	1
2	1	1
3	1	11

Figure 29: Counterexample 1 Optimal Schedule

Table 24: Permutation Schedule Makespans

Schedule	Makespan
$\{(1/2, 3)/(1/2, 3)\}$	13
$\{(1/3, 2)/(1/3, 2)\}$	13
$\{(1, 2/3)/(1, 2/3)\}$	13
$\{(2, 1/3)/(2, 1/3)\}$	13
$\{(1, 3/2)/(1, 3/2)\}$	23
$\{(3, 1/2)/(3, 1/2)\}$	13

Following Corwin and Esogbue, we consider the case of two stages where either the first or second stage has sequence-dependent setup times. However, we allow two machines at each stage. By counter-example, we show that permutation schedule are sub-optimal when the first or second stage has sequence-dependent setup times.

4.2.2 Counterexample 2: Sequence-dependent Setup Times at First Stage

With the processing times as shown in Table 23 and the setup times in Table 25 , the following schedule is optimal: $\{(1 / 3, 2) / (2, 1 / 3)\}$ with a makespan of 12. The schedule is shown in Figure 30. The schedule shows that the sequence-dependent setup times in the first stage only affected one job, namely job 2, which is not the job that sets the makespan for the schedule. The six permutation schedules and their makespans are shown in Table 26.

Table 25: Setup Times at Stage 1

Job	1	2	3
1	—	1	2
2	1	—	3
3	2	3	—

Figure 30: Counterexample 2 Optimal Schedule

Table 26: Permutation Schedule Makespans

Schedule	Makespan
$\{(1/2, 3)/(1/2, 3)\}$	16
$\{(1/3, 2)/(1/3, 2)\}$	13
$\{(1, 2/3)/(1, 2/3)\}$	14
$\{(2, 1/3)/(2, 1/3)\}$	14
$\{(1, 3/2)/(1, 3/2)\}$	25
$\{(3, 1/2)/(3, 1/2)\}$	15

4.2.3 Counterexample 3: Sequence-dependent Setup Times at Second Stage

With the processing times as shown in Table 23 and the setup times in Table 27, the following schedule is optimal: $\{(1/3, 2)/(2, 1/3)\}$ with a makespan of 13. The schedule is shown in Figure 31. The schedule shows that the sequence-dependent setup times in the second stage only affected one job, namely job 1, which is the job that sets the makespan for the schedule.

Table 27: Setup Times at Stage 2

Job	1	2	3
1	—	1	2
2	1	—	3
3	2	3	—

Figure 31: Counterexample 3 Optimal Schedule

Table 28: Permutation Schedule Makespans

Schedule	Makespan
$\{(1/2, 3)/(1/2, 3)\}$	16
$\{(1/3, 2)/(1/3, 2)\}$	16
$\{(1, 2/3)/(1, 2/3)\}$	14
$\{(2, 1/3)/(2, 1/3)\}$	14
$\{(1, 3/2)/(1, 3/2)\}$	25
$\{(3, 1/2)/(3, 1/2)\}$	15

4.2.4 Counterexample 4: Sequence-dependent Setup Times at Each Stage

With the processing times as shown in Table 23 and the setup times in Table 29 and Table 30 the following schedule is optimal: $\{(1/3, 2)/(2, 1/3)\}$ with a makespan of 13. The schedule is shown in Figure 32. The schedule shows that the sequence-dependent setup times in the first stage only affected one job, namely job 2, which is not the job that sets the makespan for the schedule. However, the sequence-dependent setup times in the second stage affected one job, namely job 1, which is the job that sets the makespan for the schedule.

Table 29: Setup Times at Stage 1

Job	1	2	3
1	—	1	2
2	1	—	3
3	2	3	—

Table 30: Setup Times at Stage 2

Job	1	2	3
1	—	1	2
2	1	—	3
3	2	3	—

Figure 32: Counterexample 4 Optimal Schedule

Table 31: Permutation Schedule Makespans

Schedule	Makespan
$\{(1/2, 3)/(1/2, 3)\}$	19
$\{(1/3, 2)/(1/3, 2)\}$	16
$\{(1, 2/3)/(1, 2/3)\}$	15
$\{(2, 1/3)/(2, 1/3)\}$	15
$\{(1, 3/2)/(1, 3/2)\}$	27
$\{(3, 1/2)/(3, 1/2)\}$	17

In addition to allowing multiple machines per stage, this research allows a non-constant number of machines per stage. For example, the first stage may have two machines and the second stage may have only one machine. A schedule can be represented as discussed above in this case as well, but the definition of a permutation schedule is unclear and is an area for future research.

4.3 Lower Bounds

This section contains a theorem with three lower bounds for the problem P . The notation " $\min_{[k]}$ " will be used to indicate the $(k+1)$ st from the lowest value and

$\min_{[0]} \equiv \min$. For example, given a list of values $\{2, 5, 7, 8, 9\}$, $\min_{[1]} = 5$.

Theorem: The following are lower bounds on any feasible solution to P :

$$LB^{(1)} = \max_{i=1,\dots,n} \left\{ \sum_{i \in S_t} \left(p_i^t + \min_{j=0,\dots,n} s_{ji}^t \right) \right\} \quad (4.13)$$

$$LB^{(2)} = \max_{t=1,\dots,g} \left\{ \frac{\sum_{i \in S^t} \left(p_i^t + \min_{j=0,\dots,n} s_{ji}^t \right)}{m^t} \right\} \quad (4.14)$$

$$LB^{(3)} = \max_{t=1,\dots,g} \left\{ \min_{i \in S^t} \sum_{r=1}^{t-1} \left(p_i^r + \min_{j=0,\dots,n} s_{ji}^r \right) + \frac{\sum_{i \in S^t} \left(p_i^t + \min_{j=0,\dots,n} s_{ji}^t \right)}{m^t} + \min_{i \in S^t} \sum_{r=t+1}^g \left(p_i^r + \min_{j=0,\dots,n} s_{ji}^r \right) + \frac{1}{m^t} \sum_{k=1}^{m^t-1} \left[\min_{i \in S^t} \sum_{j=1}^{t-1} \left(p_i^j + \min_{j=0,\dots,n} s_{ji}^j \right) - \min_{i \in S^t} \sum_{j=1}^{t-1} \left(p_i^j + \min_{j=0,\dots,n} s_{ji}^j \right) \right] \right\} \quad (4.15)$$

Proof:

$LB^{(1)}$: Every job i must be processed at each stage and must also be setup, which requires at least the minimal amount of time required to setup job i . Solutions which are feasible to constraint sets (4.5) and (4.6) satisfy this condition.

$LB^{(2)}$: Every stage t needs time to process job 0 and then the preemptive processing and minimal setup time for the rest of the jobs. Solutions which are feasible to constraint sets (4.5) and (4.6) satisfy this condition. Alternatively, we could solve an assignment problem at each stage to possibly strengthen the second term in parenthesis. In this case, job 0 would need to be replicated m^t times.

$LB^{(3)}$: In addition to minimum setup and processing at each stage, we can add in the minimum time to get to the stage plus the minimum time to finish after the stage. Furthermore, we may be able to bound idle time for parallel machines at each stage waiting for the first available job. This yields the bound $LB^{(3)}$ originally

proposed in Kurz and Askin (2001) as an extension to Leon and Ramamoorthy (1997). The first term represents the minimum time required for a job to reach stage t . The second term is the same as $LB^{(2)}$, which assumes that jobs are processed preemptively at stage t . The third term represented the minimum time for a job to finish processing and setup on the stages after t . The final term requires the observation that the second machine at stage t does not begin processing until at least the second job arrives, and so on for all the machines at stage t . We find the minimum times for the second, third, etc. jobs to reach stage t and allocate this time to all the machines at this stage. Solutions which are feasible to constraint sets (4.5) and (4.6) satisfy this condition.

4.4 Heuristics

Ten heuristics are considered here. The first class of heuristics is based on simplistic assignment of jobs to machines with little or no regard for the setup times. The second class of heuristics is based on the Insertion heuristic for the TSP. For the single stage problem, this approach has been found to be effective. Another class of heuristics is based on Johnson's Rule. The last class of heuristics is based on genetic algorithms. Note that the second class caters to setup aspects of the problem while the third derives from standard flow shops. In light of the discussion in Section 4.2, no restrictions on the form of the resultant schedules is made.

Let $[i]$ indicate the i th job in an ordered sequence in the following. In many of the

following heuristics, a modified processing time is used. It is denoted by $\tilde{p}_i^t$ for job i in stage t and is defined as $\tilde{p}_i^t = p_i^t + \min_j s_{ji}^t$. This time represents the minimum time at a stage t that must elapse before job i could be completed.

4.4.1 Cyclic Heuristics

This section will discuss three variants of a cyclic heuristic. The Cyclic Heuristics are naïve greedy heuristics that assign jobs to machines with little or no regard for setup times or the interactions between stages. Because of their simplistic nature, they provide a basis of comparison for the other heuristic classes.

4.4.1.1 Basic Cyclic Heuristic

The Cyclic Heuristic (CH) arranges jobs in numerical order and then assigns them to machines in a cyclic order. The procedure follows the steps:

1. Arrange jobs in numerical order from 1 to n .
2. At each stage $t=1, \dots, g$, assign job 0 to each machine in that stage.
3. For each stage $t=1, \dots, g$:
 - a. Let $mc=1$
 - b. For $i=1$ to n , $i \in S^t$:
 - If $mc > m'$, let $mc=1$.
 - Place job i last on machine mc .
 - Let $mc=mc+1$.

Though simplistic, CH has a time complexity of O/ng .

4.4.1.2 Readytime Cyclic Heuristic

The Readytime Cyclic Heuristic (RCH) is nearly identical to CH. However, in the RCH the jobs are sequenced in increasing order of ready times at stages 2 to g before being placed on machines. Ready times at stage 0 are 0 for all jobs.

1. Arrange jobs in numerical order from 1 to n .
2. At each stage $t=1, \dots, g$, assign job 0 to each machine in that stage.
3. For each stage $t=1, \dots, g$:
 - a. Update the ready times in stage t to be the completion times in stage $t-1$
 - b. Arrange jobs in increasing order of ready times.
 - c. Let $mc=1$
 - d. For $[i]=1$ to n :

If $mc > m'$, let $mc=1$.

Place job $[i]$ last on machine mc .

Let $mc=mc+1$.

RCH's time complexity depends on the complexity of the sorting routine in Step 3b. If a selection sort ($O(n^2)$) is used, the overall complexity is $O(n^2g + ng) = O(n^2g)$.

4.4.1.3 SPT Cyclic

In the SPT Cyclic Heuristic (SPTCH), the jobs are ordered at stage 1 in increasing order of the modified processing times $\tilde{p}_i^1$. At subsequent stages, jobs are assigned in earliest ready time order. Jobs are assigned to the machine in every stage that allows it to complete at the earliest time as measured in a greedy fashion. SPTCH has a time complexity of $O(n^2g)$.

4.4.2 Multiple Insertion Heuristics

This section will discuss three variants of a multiple insertion heuristic. The Insertion Heuristic for the TSP is an effective method. Because of the TSP nature of ordering jobs at each machine, this method has been adapted and applied to the single stage problem in the previous chapter. This multiple insertion heuristic with completion time criteria was found to be effective for the single stage case. This approach can also be considered an adaptation of the NEH method (Nawaz *et al.* 1983). However, we will consider three different criteria as the method for choosing on which machine and in which position on that machine each job will be placed.

4.4.2.1 Flowtime Multiple Insertion Heuristic

The Flowtime Multiple Insertion Heuristic (FTMIH) is a multiple insertion heuristic to minimize the sum of flowtimes (completion-ready times) at each stage. It is a multiple-machine, multiple-stage adaptation of the Insertion Heuristic for TSP. Setup times are accounted for by integrating their values into the processing times using the $\tilde{p}_i^t$. The Insertion Heuristic can then be performed using these modified processing times at each stage. Once jobs have been assigned to machines, the true processing and setup times can be used. The FTMIH has the following steps for each stage t :

1. Create the modified processing times $\tilde{p}_i^t$.
2. Order the jobs in non-increasing order (LPT) of $\tilde{p}_i^t$.
3. For $[i]=1$ to n , $i \in S^t$:
 - a. insert job $[i]$ into every position on each machine

- b. calculate the true sum of flowtimes using the actual setup times
- c. place job i in the position on the machine with the lowest resultant sum of flowtimes

4. Update the ready times in stage $t+1$ to be the completion times in stage t .

Step 3a requires that each job $[i]$ be placed in $[i]+1$ positions for evaluation in step 3b.

Each evaluation requires $[i]$ calculations. Step 3 requires

$$\sum_{[i]=1}^n ([i]+1)([i]) = \frac{1}{6}(n)(n+1)(2n+1) + \frac{n(n+1)}{2} \text{ steps. Since this must be done for each}$$

stage, the overall complexity of FTMIH is $O(n^3g)$.

4.4.2.2 Completion Time Multiple Insertion Heuristic

The Completion Time Multiple Insertion Heuristic (CTMIH) is a multiple insertion heuristic to minimize the sum of completion times at each stage. It differs from the FTMIH only in that the sum of the completion times in the current stage is considered, regardless of the values of the ready times in that stage. The changed step is step 3.c., shown here:

- 3.c. place job i in the position on the machine with the lowest resultant sum of completion times

The overall complexity of CTMIH is $O(n^3g)$, following the same reasoning as for FTMIH.

4.4.2.3 Makespan Multiple Insertion Heuristic

The Makespan Multiple Insertion Heuristic (MMIH) is a multiple insertion heuristic to minimize the maximum completion time at each stage. It differs from the CTMIH only in that only the maximum completion time (makespan) at each stage is considered. The changed step is step 3.c., shown here:

- 3.c. place job i in the position on the machine with the lowest resultant maximum completion time

The overall complexity of MMIH is $O(n^3g)$, following the same reasoning as for FTMIH.

4.4.3 Johnson's Rule Based Heuristics

This section will discuss two variants of Johnson's Rule adapted for the multiple stage case. Johnson's Rule finds the optimal makespan solution for $F2||C_{max}$ (1954). Variants have been created, for example by Campbell *et al.* (1970), for the flow shop with more than two stages. This section presents two simple adaptations of Johnson's Rule for the sequence-dependent flexible flow line problem under consideration.

4.4.3.1 The 1, g Johnson's Rule

This heuristic is an extension of Johnson's Rule to take into account the setup times. Only the first and last stages are considered to create the order for assignment in stage 1.

1. Create the modified processing times $\bar{p}_i^1$ and $\bar{p}_i^g$.
 2. Let $U = \{j \mid \bar{p}_j^1 < \bar{p}_j^g\}$ and $V = \{j \mid \bar{p}_j^1 \geq \bar{p}_j^g\}$.
 3. Arrange jobs in U in non-decreasing order of $\bar{p}_i^1$ and arrange jobs in V in non-increasing order of $\bar{p}_i^g$. Append the ordered list V to the end of U .
 4. At each stage $t=1, \dots, g$, assign job 0 to each machine in that stage.
 5. For $[i]=1$ to n , $i \in S^1$:
 - a. For $mc=1$ to m^1 :

Place job $[i]$ last on machine mc .

If this placement results in the lowest completion time for job $[i]$, let $m=mc$.
 - b. Place job $[i]$ last on machine m .
 6. For each stage $t=2, \dots, g$:
 - a. Update the ready times in stage t to be the completion times in stage $t-1$
 - b. Arrange jobs in increasing order of ready times.
 - c. For $[i]=1$ to n , $i \in S^t$:
 - (1) For $mc=1$ to m^t :

Place job $[i]$ last on machine mc .

If this placement results in the lowest completion time for job $[i]$, let $m=mc$.
 - (2) Place job $[i]$ last on machine m .
- 1, g Johnson's Rule has time complexity of $O(n^2g)$, following the same reasoning as for SPTCH.

4.4.3.2 The $g/2, g/2$ Johnson's Rule

This rule differs from the 1, g Johnson's Rule in that $\bar{p}_i^1$ is the sum of modified processing times for stages 1 to $\left\lfloor \frac{g}{2} \right\rfloor$ and $\bar{p}_i^g$ uses the sum over stages $\left\lfloor \frac{g}{2} \right\rfloor + 1$ to g .

The changed step is step 1., shown here:

1. Create the modified processing times $\tilde{p}_i^1 = \sum_{t=1}^{\lfloor g/2 \rfloor} p_i^t$ and $\tilde{p}_i^g = \sum_{t=\lfloor g/2 \rfloor + 1}^g p_i^t$.

The $g/2, g/2$ Johnson's Rule has time complexity of $O(n^2g)$, following the same reasoning as for the $1, g$ Johnson's Rule.

4.4.4 Genetic Algorithms

This section will discuss two variants of genetic algorithms applied to this problem. Genetic algorithms have been applied to numerous difficult combinatorial optimization problems with positive results. This section presents two genetic algorithms: the first is an extension of the one developed for the single stage case in the previous chapter and the second is an application of the Random Keys Genetic Algorithm (Bean 1994).

4.4.4.1 Genetic Algorithm

Genetic algorithms have been discussed in two previous chapters. In this chapter, we continue with the same representation and parameter values as those used in the previous chapter. However, we only consider the jobs and machines for stage 1. The jobs are processed on the machines in stage 1 in the order in which they appear in the chromosome. In all other stages, jobs are assigned to machines in ready time order such that they end at the earliest time.

4.4.4.2 *Random Keys Genetic Algorithm*

Bean (1994) introduced an alternative method to encode problem solutions using random numbers called a Random Keys Genetic Algorithm (RKGA), which has been applied to resource allocation problems, quadratic assignment problems, multiple machine tardiness scheduling problems, jobshop makespan minimization problems and the generalized traveling salesman problem (Bean 1994, Norman and Bean 1999, Snyder and Daskin 2001).

RKGA differs from traditional genetic algorithms most notably in the solution representation. Random numbers serve as sort keys in order to decode the solution. The decoded solution is evaluated with a fitness function that is appropriate for the problem at hand. For example, Norman and Bean (1999) suggest using the following solution representation for an identical multiple machine problem. Each job is assigned a real number whose integer part is the machine number to which the job is assigned and whose fractional part is used to sort the jobs assigned to each machine. Once the job assignments and order on each machine is found through the decoding, a schedule can be built incorporating additional factors such as nonzero ready times and sequence-dependent setup times. The desired performance measure can then be found using the schedule. In this research, this representation is used for the jobs in the first stage. The assignment of jobs to machines in subsequent stages follows the method used in SPTCH and the Johnson's Rule Based Heuristics, where jobs are assigned to the machine that allows it to complete at the earliest time as measured in a greedy fashion.

The genetic operators and related parameters used in this research are based on

those in Bean (1994). Each generation has a population of 100 chromosomes. The initial population is generated randomly. An elitist strategy is used for reproduction. Each chromosome is decoded and the resulting schedule is evaluated for the makespan. Chromosomes with lower makespans are more desirable, so the 20% lowest makespan chromosomes are automatically copied to the next generation. Parametrized uniform crossover is used to select the next 79% chromosomes in the next generation. For each chromosome in the next generation, the following is performed. Two chromosomes in the current generation are selected at random. For each job, a random number is generated. If the value is less than 0.7 (following Bean 1994), the value from the “first” chromosome is copied to the new chromosome, otherwise the value from the “second” chromosome is selected. The remaining 1% of the next generation is filled through “immigration”, in which new chromosomes are randomly generated. The above procedures are repeated until we are fairly sure that the population has settled into a good location in the search space. In this research, we continue until 100 generations have been examined without finding an improved schedule. This value was selected empirically.

4.5 Examples

The following data set will be used to illustrate the various heuristics. It has up to 6 jobs in each of four stages. Each stage has two machines. The processing and setup times are shown in Figure 33.

Stage 1					
Job	0	2	3	4	5
p_i^1	20	63	62	54	53
$\tilde{p}_i^1$	20	66	64	54	57

Stage 2					
Job	0	1	2	5	6
p_i^2	16	70	73	52	57
$\tilde{p}_i^2$	16	70	77	58	57

Stage 1				
(s_{ji}^1)	To			
From	2	3	4	5
0	3	2	0	4
2	—	15	15	19
3	23	—	15	15
4	17	22	—	20
5	18	16	20	—
$\min_j s_{ji}^1$	3	2	0	4

Stage 2				
(s_{ji}^2)	To			
From	1	2	5	6
0	4	4	8	7
1	—	4	8	7
2	0	—	6	0
5	19	13	—	16
6	16	17	20	—
$\min_j s_{ji}^2$	0	4	6	0

Stage 3				
Job	0	1	2	4
p_i^3	20	73	71	57
$\tilde{p}_i^3$	20	76	73	60

Stage 4				
Job	0	1	2	6
p_i^4	18	70	64	63
$\tilde{p}_i^4$	18	70	67	64

Stage 3			
(s_{ji}^3)	To		
From	1	2	4
0	3	2	3
1	—	4	5
2	5	—	6
4	20	19	—
$\min_j s_{ji}^3$	3	2	3

Stage 4			
(s_{ji}^4)	To		
From	1	2	6
0	0	3	5
1	—	4	1
2	4	—	2
6	18	21	—
$\min_j s_{ji}^4$	0	3	1

Figure 33: Processing and Setup Times for Example Problem

The lower bound on the makespan for this schedule was found to be 291. The integer programming formulation was used to create a CPLEX input file for this example. CPLEX was used to solve this problem to optimality. The optimal makespan solution is 307. Table 32 shows the optimal schedule.

Table 32: Optimal Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	4	86	155	1	5	110	170
2	0	0	20	1	6	170	243
2	5	20	77	2	0	20	36
2	3	77	155	2	2	86	163

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	110	180
2	4	186	248	1	6	243	307
2	0	36	56	2	0	56	74
2	2	163	236	2	2	236	303

When each schedule is written, it is to be understood that job 0 appears first on each machine in each stage.

4.5.1 CH

CH simply orders the jobs and assigns them to machines in each stage in this order. The schedule is $\{(2, 4 / 3, 5) / (1, 5 / 2, 6) / (1, 4 / 2) / (1, 6 / 2)\}$ with a makespan of 320 and is shown in Table 33.

Table 33: CH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	4	86	155	1	5	152	212
2	0	0	20	2	0	20	36
2	3	20	84	2	2	86	163
2	5	84	152	2	6	163	220

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	4	186	248	1	6	256	320
2	0	36	56	2	0	56	74
2	2	163	236	2	2	236	303

4.5.2 RCH

In RCH, the stage 1 assignments are the same as in CH. However, the subsequent stages first order the jobs in increasing ready time order. Table 34 shows the order in which jobs are placed in the subsequent stages. The schedule is $\{(2, 4 / 3, 5) / (1, 2 / 5, 6) / (1, 2 / 4) / (1 / 6, 2)\}$ with a makespan of 347 and is shown in Table 35.

Table 34: Order for RCH

Stage	2	3	4
Job Order	1, 6, 2, 5	1, 4, 2	6, 1, 2

Table 35: RCH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	4	86	155	1	2	110	187
2	0	0	20	2	0	20	36
2	3	20	84	2	6	36	100
2	5	84	152	2	5	152	224

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	2	187	262	2	0	56	74
2	0	36	56	2	6	100	168
2	4	155	215	2	2	262	347

4.5.3 SPTCH

In SPTCH, stage 1 assignments are made in SPT order. The subsequent stages first order the jobs in increasing ready time order and assign the job to the machine on which it ends earliest. The order in which jobs are assigned in stage 1 and considered in stages 2, 3 and 4 are shown in Table 36.

Table 36: Order for SPTCH

Stage	1	2	3	4
Job Order	4, 5, 3, 2	1, 6, 5, 2	4, 1, 2	6, 1, 2

In stage 2, the jobs are considered for placement on each machine. Recall that job 0 is first assigned to each machine. Job 1 is assigned to machine 1 and job 6 is assigned to machine 2. Next, job 5 is considered for placement on both machine 1 and machine 2.

Table 37 shows the potential completion times of job 5 if placed on either machine.

Recall that the completion time of a job is the larger of the job's ready time and the previous job's completion time, plus the required setup time, plus the job's processing time. As shown, job 5 will end at time 170 if placed on machine 1 or at time 172 if placed on machine 2. Since 170 is less than 172, job 5 is placed on machine 1. These steps are repeated for the remaining jobs on the remaining stages.

Table 37: Job Completion Time as Criterion for Machine Assignment Example

Potential Machine Assignment	Completion Time of Job 5
MC 1: 0, 1, 5 MC 2: 0, 6	$110+8+52 = 170$
MC 1: 0, 1, MC 2: 0, 6, 5	$100+20+52 = 172$

Table 38: SPTCH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	5	20	77	1	1	36	110
1	2	77	158	1	5	110	170
2	0	0	20	1	2	170	256
2	4	20	74	2	0	20	36
2	3	74	158	2	6	36	100

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	2	256	331	1	2	331	399
2	0	36	56	2	0	56	74
2	4	74	134	2	6	100	168

The schedule is $\{(5, 2 / 4, 3) / (1, 5, 2 / 6) / (1, 2 / 4) / (1, 2 / 6)\}$ with a makespan of 399 and is shown in Table 38.

4.5.4 FTMIH

In FTMIH, jobs are considered in each stage in LPT order. The LPT order of each stage is given in Table 39.

Table 39: Order for FTMIH

Stage	1	2	3	4
Job Order	2, 3, 5, 4	2, 1, 5, 6	1, 2, 4	1, 2, 6

In each stage, the jobs are considered for placement on each machine in each potential position. This will be explained in detail by way of an example. Recall that job 0 is first assigned to each machine. Job 2 is assigned to machine 1 and job 3 is assigned to machine 2. Next, job 5 is considered for placement on both machine 1 and machine 2 in every potential position. That is, job 5 is placed before and after job 2 and before and after job 3. Whichever position results in the smallest sum of flowtimes is the assigned position and machine for job 5. As can be seen in Table 40, job 5 is assigned to machine 2 in the first position. The schedule is $\{(4, 2 / 5, 3) / (1, 2 / 6, 5) / (1 / 4, 2) / (1, 2 / 6)\}$ with a makespan of 389 as shown in Table 41.

Table 40: Sum of Flowtimes as Criterion for Machine Assignment Example

Potential Machine Assignment	Sum of Flow Times
MC 1: 0, 5, 2 MC 2: 0, 3	359
MC 1: 0, 2, 5 MC 2: 0, 3	368
MC 1: 0, 2 MC 2: 0, 5, 3	358
MC 1: 0, 2 MC 2: 0, 3, 5	362

Table 41: FTMIH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	4	20	74	1	1	36	110
1	2	74	154	1	2	154	231
2	0	0	20	2	0	20	36
2	5	20	77	2	6	36	100
2	3	77	155	2	5	100	172

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
2	0	36	56	1	2	321	389
2	4	74	134	2	0	56	74
2	2	231	321	2	6	100	168

4.5.5 CTMIH

In CTMIH, jobs are considered in each stage in LPT order. The LPT order of each stage is given in Table 39. Instead of considering the sum of flowtimes as the criterion for placing each job, the sum of completion times is the criterion. The schedule

in this case is the same as that found by FTMIH: $\{(4, 2 / 5, 3) / (1, 2 / 6, 5) / (1 / 4, 2) / (1, 2 / 6)\}$ with a makespan of 389 as shown in Table 41.

4.5.6 MMIH

In MMIH, jobs are considered in each stage in LPT order. The LPT order of each stage is given in Table 39. Instead of considering the sum of flowtimes as the criterion for placing each job, the maximum completion time of all jobs scheduled so far is the criterion. The schedule is $\{(4, 2 / 3, 5) / (1, 2 / 6, 5) / (4, 1 / 2) / (1, 2 / 6)\}$ with a makespan of 372 as shown in Table 42.

Table 42: MMIH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	4	20	74	1	1	36	110
1	2	74	154	1	2	154	231
2	0	0	20	2	0	20	36
2	3	20	84	2	6	36	100
2	5	84	152	2	5	152	224

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	4	74	134	1	1	227	297
1	1	134	227	1	2	304	372
2	0	36	56	2	0	56	74
2	2	231	304	2	6	100	168

4.5.7 The 1, g Johnson's Rule

In the 1, g Johnson's Rule Heuristic, the stage 1 and 4 modified processing times are used to apply Johnson's Rule. The set $U = \{j | \tilde{p}_j^1 < \tilde{p}_j^4\} = \{1, 2, 6\}$ and $V = \{j | \tilde{p}_j^1 \geq \tilde{p}_j^4\} = \{3, 4, 5\}$. The sorted sets are $U = \{1, 6, 2\}$ and $V = \{3, 4, 5\}$. The order in which jobs are considered in each stage is shown in Table 43.

Table 43: Order for 1, g Johnson's Rule

Stage	1	2	3	4
Job Order	2, 3, 4, 5	1, 6, 2, 5	1, 4, 2	6, 1, 2

The jobs are placed on the machine in the last position such that it completes earliest, as demonstrated in Section 4.5.3. The schedule is $\{(2, 4 / 3, 5) / (1, 2, 5 / 6) / (1, 2 / 4) / (1, 2 / 6)\}$ with a makespan of 330 as shown in Table 44.

Table 44: 1, g Johnson's Rule Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	4	86	155	1	2	110	187
2	0	0	20	1	5	187	245
2	3	20	84	2	0	20	36
2	5	84	152	2	6	36	100

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	2	187	262	1	2	262	330
2	0	36	56	2	0	56	74
2	4	155	215	2	6	100	168

4.5.8 The $g/2, g/2$ Johnson's Rule

In the $g/2, g/2$ Johnson's Rule Heuristic, the stage 1 and 2 modified processing times added together and the stage 3 and 4 modified processing times are added together. These times are shown in Table 45. The resulting times are used to apply Johnson's Rule.

Table 45: Aggregated Times for $g/2, g/2$ Johnson's Rule

Job	$\bar{p}_i^l$	$\bar{p}_i^s$
1	70	146
2	143	140
3	64	0
4	54	60
5	115	0
6	57	64

The set $U = \{j | \bar{p}_j^l < \bar{p}_j^s\} = \{1, 4, 6\}$ and $V = \{j | \bar{p}_j^l \geq \bar{p}_j^s\} = \{2, 3, 5\}$. The sorted sets are $U = \{4, 1, 6\}$ and $V = \{2, 3, 5\}$. The order in which jobs are considered in each stage is shown in Table 46.

Table 46: Order for $g/2, g/2$ Johnson's Rule

Stage	1	2	3	4
Job Order	4, 2, 3, 5	1, 6, 2, 5	4, 1, 2	6, 1, 2

The jobs are placed on the machine in the last position such that it completes earliest, as demonstrated in Section 4.5.3. The schedule is $\{(2, 5 / 4, 3) / (1, 2, 5 / 6) / (1, 2 / 4) / (1, 2 / 6)\}$ with a makespan of 330 as shown in Table 47.

Table 47: $g/2, g/2$ Johnson's Rule Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	5	86	158	1	2	110	187
2	0	0	20	1	5	187	245
2	4	20	74	2	0	20	36
2	3	74	158	2	6	36	100

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	2	187	262	1	2	262	330
2	0	36	56	2	0	56	74
2	4	74	134	2	6	100	168

4.5.9 GA

GA was run on the data set using the parameters discussed earlier. The final chromosome returned was [2-1, 1-0, 5-1, 6-0, 3-2, 4-1]. This chromosome contains the following information: jobs 1 and 6 do not exist at stage 1 (are assigned to machine 0); jobs 2, 4, and 5 are assigned to machine 2 in the order 2, 5, 4; job 3 is the only additional job assigned to machine 2. This solution was settled on in 2.25 seconds but this chromosome was found in generation 24 of 120. The jobs in subsequent stages are placed on the machine in the last position such that it completes earliest, as demonstrated in Section 4.5.3. The schedule is $\{(2, 5, 4 / 3) / (1, 6 / 2, 5) / (1, 4 / 2) / (1, 2 / 6)\}$ with a makespan of 324 as shown in Table 48.

Table 48: GA Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
1	5	86	158	1	6	110	174
1	4	158	232	2	0	20	36
2	0	0	20	2	2	86	163
2	3	20	84	2	5	163	221

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	4	232	294	1	2	256	324
2	0	36	56	2	0	56	74
2	2	163	236	2	6	100	168

4.5.10 RKGA

RKGA was run on the data set using the parameters discussed earlier. The final chromosome returned was [0.00 1.33 2.47 2.69 2.53 0.00]. This chromosome contains the following information: jobs 1 and 6 do not exist at stage 1; job 2 is the only additional job assigned to machine 1; jobs 3, 4, and 5 are assigned to machine 2 in the order 3, 5, 4. This solution was settled on in 12.94 seconds but this chromosome was found in generation 0, or in the initialization of the population. The jobs in subsequent stages are placed on the machine in the last position such that it completes earliest, as demonstrated in Section 4.5.3. The schedule is $\{(2 / 3, 5, 4) / (1, 2, 5 / 6) / (1, 4 / 2) / (1, 2 / 6)\}$ with a makespan of 328 as shown in Table 49.

Table 49: RKGA Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	2	20	86	1	1	36	110
2	0	0	20	1	2	110	187
2	3	20	84	1	5	187	245
2	5	84	152	2	0	20	36
2	4	152	226	2	6	36	100

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	1	110	186	1	1	186	256
1	4	226	288	1	2	260	328
2	0	36	56	2	0	56	74
2	2	187	260	2	6	100	168

4.6 Generation of Test Data

An experiment was conducted to test the performance of the heuristics. Data generated was integer. Data required for a problem consist of the number of jobs, range of processing times, number of stages and whether all stages have the same number of machines or not. Each stage requires data telling how many machines exist at that stage, the sequence-dependent setup times, the processing times and the ready times. The ready times for stage 1 are set to 0 for all jobs. The ready times at stage $t+1$ are the completion times at stage t , for $t > 1$, so this data need not be generated. Processing times are distributed uniformly with a mean of 60. Flexible flow lines are considered by allowing some jobs to skip some stages. Following Leon and Ramamoorthy (1997), the

probability of skipping a stage is set at 0, 0.05, or 0.40. The low value of setup times are uniformly distributed from 12 to 24, which is 20% to 40% of the mean of the processing time. A high value is also considered with setup times having the same mean as processing time. The setup time matrices are asymmetric and satisfy the triangle inequality. The triangle inequality assumption is reasonable, because we do not want to allow shortcuts in setting up a particular job by setting up for an intermediate job that is not required. The setup time characteristics follow Rios-Mercado and Bard (1998).

Problem data can be characterized by seven factors: the range of setup times, the probability that a job skips a stage, range of processing times, number of stages, whether the number of machines per stage is constant or variable, range in number of machines per stage and number of jobs. Each of these factors can have at least two levels. When coding the test cases, the letters "L", "M", "H" and "A" will be used. The meaning of these levels are shown in Table 50.

Table 50: Factor Levels

Factor	Levels			
Setup Times	Unif(12-24)		L	
	Unif(54-66)		H	
Skipping Probability	0		L	
	0.05		M	
	0.4		H	
Processing Times	Unif(50-70)		L	
	Unif(20-100)		H	
Number of Stages	2		L	
	4		M	
	8		H	
Machine Distribution	Constant		L	
	Variable		H	
Number of Machines	(Depends on Machine Distribution)			
	<u>Constant</u>		<u>Variable</u>	
	1	L	Unif(1,4)	L
	2	M		
Number of Jobs	10	H	Unif(1,10)	H
	6		L	
	20		M	
	30		H	
	100		A	

In general, all combinations of these levels will be tested. However, some further restrictions are introduced. The variable machine distribution factor requires that at least one stage have a different number of machines than the others. Also, the largest number of machines in a stage must be less than the number of jobs. Thus, the combination with 10 machines at each stage and 6 jobs will be skipped and the combination of 1-10 machines per stage with 6 jobs will be changed to 1-6 machines per stage with 6 jobs. There are 684 test scenarios and ten data sets are generated for each one.

4.7 Experimental Results

This section discusses the effectiveness of the proposed lower bounds and the proposed construction heuristics. The heuristics were implemented in C on a PC with a Pentium II 200 MHz processor with 64 MB of RAM and compiled with Microsoft Visual C++. “Loss” is the (makespan – lower bound)/lower bound. Appendix E contains the average and standard deviation of loss for each of the test scenarios for each of the heuristics. The running times were found using the clock() function. Appendix F contains the average and standard deviations of running times for each of the test scenarios for each of the heuristics with run times greater than 0.01 seconds (Multiple Insertion Heuristics and Genetic Algorithms).

4.7.1 Best Heuristic in Each Class

In this section we determine which representative of each class of heuristics performs best so that we are left with 4 rather than 10 heuristics.

4.7.1.1 Cyclic Heuristics

The Cyclic Heuristics are the most basic heuristics considered. RCH is CH with the additional step that jobs are sorted according to ready times instead of the job’s name. CH and RCH both assign jobs to the next machine numerically, without regard for the resultant completion time of the job on that machine. SPTCH is different from the others in two respects. First, the jobs are ordered in stage 1 according the modified processing times, which takes into account some measure of setup times. Secondly, in subsequent

stages, jobs are assigned to the machine on which the job completes earliest. All the Cyclic Heuristics enforce artificial adjacency of jobs in stage 1 based on initial ordering by only allowing new jobs to be placed at the end of a machine's schedule.

Each of the 6840 test problems was subjected to the Cyclic Heuristics. Several statistics have been gathered on the experimental results for this group of heuristics. Table 51 contains the average, standard deviation, and maximum losses attained by each of the Cyclic Heuristics, as well as the number of times each heuristic attained the lowest makespan solution. Note that the sum of the last column can be larger than 6840, as a heuristic is credited with the best solution even if another heuristic attains that value as well. Table 51 shows us that SPTCH has the best statistics for the average, standard deviation and number of time minimum measures.

In addition, paired T-tests were performed on each pair of the Cyclic Heuristics and the results are summarized in Table 52. These values indicate that SPTCH is the preferred Cyclic Heuristic with at least 95% confidence.

Table 51: Loss Statistics for Cyclic Heuristics

Heuristic	Average Loss	Standard Deviation Loss	Maximum Loss	Number of Times Minimum
CH	0.33	0.23	1.54	789
RCH	0.27	0.16	1.11	2133
SPTCH	0.24	0.16	1.23	4540

Table 52: Paired T-Tests for Cyclic Heuristics

Heuristics	Mean Difference	Standard Deviation Difference	98.5% CI	T-statistic (6839 df)
CH - RCH	0.062	0.107	0.059, 0.066	48.437
CH - SPTCH	0.086	0.123	0.082, 0.089	57.835
RCH - SPTCH	0.023	0.076	0.021, 0.026	25.349

The running times are inconsequential, never being more than 0.01 seconds, even though the time complexities are $O(n^2g)$, $O(n^2g)$ and $O(n^2g)$ respectively.

4.7.1.2 Multiple Insertion Heuristics

The insertion heuristics (FTMIH, CTMIH, MMIH) all have the same time complexity of $O(n^3g)$ and differ only in the performance measure that is considered when evaluating the placement of a job into a particular location on a particular machine. The sum of flowtimes measure is intended to reduce the amount of time each job spends in each stage. The sum of completion times measure is intended to push all jobs to complete as early as possible in each stage. The makespan measure focuses on only the last job through each stage. This class of heuristics is the only one of the four considered here that considers stages somewhat independently. The other classes release a sequence of jobs into the first stage and the ready times at each stage basically indicate how they progress through the remaining stages. Moreover, this class of heuristics, in contrast to the Cyclic and Johnson's Rule Based Heuristics, allows jobs to be inserted in between previously scheduled jobs. In this way, a relative ordering of jobs is maintained but the adjacency of jobs is not enforced. Clearly, this difference in focus affects the time complexity of this class of heuristics. The running times for the Multiple Insertion Heuristics did not vary much, as would be expected.

Each of the 6840 test problems was subjected to the Multiple Insertion Heuristics. Several statistics have been gathered on the experimental results for this group of

heuristics. Table 53 contains the average, standard deviation, and maximum losses attained by each of the Multiple Insertion Heuristics, as well as the number of times each heuristic attained the lowest makespan solution. Note that the sum of the last column can be larger than 6840, as a heuristic is credited with the best solution even if another heuristic attains that value as well. Table 53 shows us that FTMIH has the best performance for all of these statistics. CTMIH performs extremely poorly in cases with 100 jobs, 2 or 1-4 machines per stage and 4 or 8 stages. In these cases, the low number of machines at each stage limits the number of options for a poorly placed job in a previous stage to recover. The high number of stages increases the probability that a job's placement in one stage will adversely affect many other stages. Though the Multiple Insertion Heuristics are the only class that consider each stage somewhat independently, they do so without considering the impact on subsequent stages.

Table 53: Loss Statistics for Multiple Insertion Heuristics

Heuristic	Average Loss	Standard Deviation Loss	Maximum Loss	Number of Times Minimum
FTMIH	0.23	0.15	1.04	5287
CTMIH	0.80	1.79	13.77	4334
MMIH	0.44	0.36	2.24	1750

In addition, paired T-tests were performed on each pair of the Multiple Insertion Heuristics and the results are summarized in Table 54. These values indicate that FTMIH is the preferred Multiple Insertion Heuristic with at least 95% confidence.

Table 54: Paired T-Tests for Multiple Insertion Heuristics

Heuristics	Mean Difference	Standard Deviation Difference	98.5% CI	T-statistic (6839 df)
FTMIH - CTMIH	-0.568	1.769	-0.620, -0.516	-26.558
FTMIH - MMIH	-0.207	0.319	-0.216, -0.198	-53.671
CTMIH - MMIH	0.361	1.626	0.313, 0.409	18.355

The running times of the Multiple Insertion Heuristics are all very similar to each other, as evidenced by Table 55.

Table 55: Multiple Insertion Heuristics Running Times

Heuristic	Average Running Time (s)	Minimum Running Time (s)	Maximum Running Time (s)	Std Dev Running Time (s)
FTMIH	20.35	< 0.01	410.68	56.26
CTMIH	19.85	< 0.01	400.85	54.91
MMIH	19.84	< 0.01	401.61	54.89

4.7.1.3 Johnson's Rule Based Heuristics

The Johnson's Rule Based Heuristics attempt to adapt Johnson's Rule simply in light of the flow line characteristics of the flexible flow lines under consideration. The 1,g Johnson's Rule Heuristic uses stage 1 times as the first value because the order the jobs enter stage 1 affects the order the jobs reach the rest of the stages. The last stage's times are used because the order in which jobs leave the last stage will affect the makespan of the whole schedule. The g/2,g/2 Johnson's Rule Based Heuristic uses the same logic but applies this to the first half and second half of the flexible flow line. The Johnson's Rule Based Heuristics enforce adjacency of jobs in stage 1 by only allowing

new jobs to be placed at the end of a machine's schedule. The Johnson's Rule Based Heuristics have a time complexity of $O(n^2g)$.

Each of the 6840 test problems was subjected to the Johnson's Rule Based Heuristics. Several statistics have been gathered on the experimental results for this group of heuristics. Table 56 contains the average, standard deviation, and maximum losses attained by each of the Johnson's Rule Based Heuristics as well as the number of times each heuristic attained the lowest makespan solution. Note that the sum of the last column can be larger than 6840, as a heuristic is credited with the best solution even if another heuristic attains that value as well. Table 56 shows us that the two heuristics are very close on all these measures. Some of this is accounted for by the fact that the two heuristics are identical when there are only two stages under consideration, as is the case in about one third of the experimental data. However, the $g/2, g/2$ Johnson's Rule Based Heuristic has the better statistics in this table.

Table 56: Loss Statistics for Johnson's Rule Based Heuristics

Heuristic	Average Loss	Standard Deviation Loss	Maximum Loss	Number of Times Minimum
1, g	0.21	0.13	0.88	4271
$g/2, g/2$	0.20	0.12	0.83	5219

In addition, a paired T-test was performed on the Johnson's Rule Based Heuristics and the results are summarized in Table 57. These values indicate that the $g/2, g/2$ Johnson's Rule Based Heuristic is preferred with 95% confidence.

Table 57: Paired T-Test for Johnson's Rule Based Heuristics

Heuristics	Mean Difference	Standard Deviation Difference	95% CI	T-statistic (6839 df)
$1,g - g/2, g/2$	0.009	0.045	0.008, 0.010	16.463

The running times are inconsequential, never being more than 0.01 seconds

4.7.1.4 Genetic Algorithms

The running times of GA and RKGA are much larger than those of the other heuristics. Due to this, only the data sets with the low level of the setup time factor were run. Only 3420 data sets were run. GA will be discussed in detail first, followed by RKGA.

An interesting statistic is that the standard deviation of the running time for GA is never more than 4% of the average running time. The running times are very similar across the data sets for a particular test case, as seen in Appendix F by the low standard deviations. This suggests that either GA is finding its best schedule in late generations often and terminating under the closeness criterion at that late time or GA terminates because the maximum number of generations has been reached. The maximum number of generations in GA depends only on the number of jobs. In the experiments, the best schedule found by GA was found in varying generation numbers. Table 58 shows the maximum number of generations allowed as well as the earliest and latest generation number in which the best schedule was found. Clearly, the best (though not necessarily optimal) schedule could be found in the initial population generated (generation 0) but

possibly was not found until the last allowed generation. The consistent running times and the generation statistics shown in Table 58 indicate that GA had not found the optimal schedule, or had not been able to convince itself that it had using the closeness criterion for stopping.

Table 58: GA Generation Statistics

Number of Jobs	Maximum Number of Generations	Earliest Generation Number	Latest Generation Number
6	140	0	139
20	420	1	420
30	620	1	620
100	2020	0	2020

RKGA differs from GA most notably in its chromosomal representation of a schedule and in the genetic operators used. Though the running time of RKGA is much larger than that of the other non-genetic algorithm heuristics, this varies greatly with the number of stages and jobs. GA and RKGA required nearly the same amount of total time for the 3420 files (158.73 and 158.58 hours respectively), but the contrast to GA, the RKGA running times for a particular type of problem can vary greatly, as seen in Appendix F by the larger standard deviations of running times. One explanation for this difference is the different stopping criteria used. The running times of RKGA tend to be larger than GA's when there are six jobs and smaller than GA's when there are more jobs. As problems with very small number of jobs would be easier to solve to optimality in general, this trend indicates that RKGA is better suited than GA to larger problems. Also, the number of generations required to find a good solution varies greatly. For example, in a test problem from the group encoded HLLLLLL (high setups, 6 jobs that

must visit each stage, two stages, each with exactly one machine, low processing times), 10 generations were required to find a solution which was not improved upon in the next 100 generations. Table 59 shows the makespans of incumbent solutions, the generation number in which they were found and the lower bound for this problem. In the initial population, a schedule with makespan 585, or with loss of 0.023, is found. The ending schedule's makespan of 580 corresponds to a loss of 0.014. The improvement from generation 0 to 10 is not overwhelming, but neither is the amount of time required to finish the 110 generations, which was 2.56 seconds.

Table 59: HLLLLLL Test Problem Incumbent Solutions

Generation	Makespan
0	585
2	584
5	583
10	580
<i>LB</i>	<i>572</i>

In a test problem from the group encoded HHLHLHA (high setups, 100 jobs that may skip each stage with probability 0.4, eight stages, each with exactly 10 machines, low processing times), 803 generations were required to find a solution which was not improved upon in the next 100 generations. The 903 generations took a total of 2236.67 seconds. Though this is about 37 minutes, the loss fell from 0.766 to 0.388. Figure 34 shows how the loss fell with the generation number. Sharp improvements are seen by generation 46 (loss is 0.588), but the remaining improvement to 0.388 occurs after that point. In this case, it seems reasonable that the entire running time was valuable in finding a schedule with lower loss.

Figure 34: Generation Number vs. Loss for the HHLHLHA Case

In a test problem from the group encoded HLHLLA (high setups, 100 jobs that must visit each stage, eight stages, each with exactly one machine, high processing times), 757 generations were required to find a solution which was not improved upon in the next 100 generations. The 857 generations took a total of 1995.85 seconds. Though this is about 33 minutes, the loss fell from 0.128 to 0.054. Figure 35 shows how the loss fell with the generation number. Sharp improvements are seen by generation 191 (loss is 0.067) with the remaining improvement to 0.054 occurring slowly after that point. In this case, the improvement gained by running beyond generation 300 is very small. At this time, determining in general at which point the improvements gained will be no longer worth the extended running time is an area for further research.

Figure 35: Generation Number vs. Loss for the HLHLLA Case

Each of the 3420 test problems was subjected to the Genetic Algorithms. Several statistics have been gathered on the experimental results for this group of heuristics. Table 60 contains the average, standard deviation, and maximum losses attained by each of the Genetic Algorithms as well as the number of times each heuristic attained the lowest makespan solution. Note that the sum of the last column can be larger than 3420, as a heuristic is credited with the best solution even if another heuristic attains that value as well. Table 60 shows us that RKGA finds many more minimal solutions and finds solutions with lower average, standard deviation, and maximum loss.

Table 60: Loss Statistics for Genetic Algorithms

Heuristic	Average Loss	Standard Deviation Loss	Maximum Loss	Number of Times Minimum
GA	0.17	0.12	0.79	517
RKGA	0.12	0.09	0.49	3364

In addition, a paired T-test was performed on the Genetic Algorithms and the results are summarized in Table 61. These values indicate that RKGA is preferred with 95% confidence.

Table 61: Paired T-Test for Genetic Algorithms

Heuristics	Mean Difference	Standard Deviation Difference	95% CI	T-statistic (3419 df)
GA - RKGA	0.049	0.053	0.047, 0.050	54.117

4.7.2 Comparing Representative Heuristics

Ten heuristics were presented in a previous section. The number of heuristics under consideration has been reduced to four. RKGA was run on each of the 6840 data sets discussed previously, so every heuristic considered here was run on the same 6840 data sets. Summary statistics over all the data sets are presented in Table 62. RKGA achieves the lowest values for the loss statistics and finds the minimum loss schedules many more times than the other heuristics.

Table 62: Loss Statistics for Remaining Heuristics

Heuristic	Average Loss	Standard Deviation Loss	Maximum Loss	Number of Times Minimum
SPTCH	0.24	0.16	1.23	79
FTMIH	0.23	0.15	1.04	413
<i>g/2, g/2</i> Johnson's	0.20	0.12	0.83	255
RKGA	0.12	0.09	0.49	6343

In addition, a paired T-test was performed on the remaining heuristics and the results are summarized in Table 63. These values indicate that RKGA is preferred with 95% confidence. The *g/2, g/2* Johnson's Based Rule appears to be the next best heuristic.

Table 63: Paired T-Tests for Remaining Heuristics

Heuristics	Mean Difference	Standard Deviation Difference	98.75% CI	T-statistic (6839 df)
SPTCH - FTMIH	0.016	0.080	0.013, 0.018	16.065
SPTCH - <i>g/2, g/2</i>	0.045	0.076	0.043, 0.047	48.992
SPTCH - RKGA	0.121	0.105	0.118, 0.124	95.602
FTMIH - <i>g/2, g/2</i>	0.029	0.087	0.027, 0.032	27.876
FTMIH - RKGA	0.105	0.104	0.102, 0.108	83.645
<i>g/2, g/2</i> - RKGA	0.076	0.068	0.074, 0.078	92.787

As noted earlier, Appendix F contains the non-trivial running time statistics over the test cases. RKGA clearly has a larger running time than the other heuristics. This larger running time suggests the question of when RKGA should not be used, and instead perhaps the *g/2, g/2* Johnson's Based Rule should be used. As can be observed in Appendix E, RKGA generally dominates the *g/2, g/2* Johnson's Based Rule in loss performance. However, the *g/2, g/2* Johnson's Based Rule performs as well or better than RKGA only 303 of 6840 times. The *g/2, g/2* Johnson's Based Rule tends to perform

competitively in problems with two stages, ten machines per stage, jobs that must visit every stage and processing times that have a smaller range. This is reasonable because this kind of problem somewhat matches the scenario for which Johnson's Rule was designed. In this case, we have a two stage flowline and each job must visit each stage. However, instead of only one machine for each stage, we have ten. This means that a tie in Johnson's order does not force a job that should be first on a machine to be second; the job can be first on another machine at that stage. In this way, we allow several jobs that are early in the Johnson's order to move towards the beginning of the schedule on the first stage. In the second stage, by considering jobs in ready time order and allowing jobs to be placed on the machine on which it ends soonest, we continue with the Johnson's paradigm. We allow jobs that arrive to the second stage and will move through it quickly (before later jobs from stage 1) to be scheduled earlier. These jobs should not impact the overall makespan much because the later jobs in stage 1 will not arrive in stage 2 very soon anyway. The low range of processing times and the fact that all jobs visit both stages indicates that the stages are fairly balanced in terms of workload. Overall, this reasoning indicates that the $g/2, g/2$ Johnson's Based Rule may be more suitable for RKGA in cases with fewer stages and more machines in which the stages are balanced.

4.7.3 Quality of the Lower Bounds

Three lower bounds were presented in a previous section. Table 64 shows the number of times each lower bound was the highest. Recall that $LB^{(1)}$ is the maximum time over all jobs to process and minimally setup each job in all stages. $LB^{(2)}$ is the

maximum time over all stages to preemptively process and minimally setup all jobs in that stage. $LB^{(3)}$ is $LB^{(2)}$ with the added time to reach and leave each stage. Since $LB^{(3)}$ will be at least as effective as $LB^{(2)}$, we will compare $LB^{(1)}$ and $LB^{(3)}$. Table 64 shows that, while $LB^{(3)}$ is the largest 6015 out of 6840 times, $LB^{(1)}$ is larger over 800 times. Since $LB^{(1)}$ is easy to compute, it should be included even though it is not as good a lower bound in general as $LB^{(3)}$.

Table 64: Lower Bounds

Lower Bound	Number of Times Largest
$LB^{(1)}$	825
$LB^{(3)}$	6015

It can also be noted that RKGA occasionally found solutions very close to the lower bound $LB^{(3)}$. In fact, of the 6840 test files, RKGA found solutions equivalent to the lower bound 31 times and within 1% of the lower bound 118 additional times. $LB^{(3)}$ is an effective lower bound in relation to the makespan as well.

4.7.4 Significance of the Experimental Factors

In Section 4.6, several experimental factors and their levels were introduced. In this section, we will discuss their significance. The indicators for each factor are shown in Table 65. Table 66 contains an ANOVA performed with effects limited to the main effects and the two way effects. The data used for the ANOVA are the loss values achieved by SPTCH, FTMIH, the $g/2$, $g/2$ Johnson's Rule Based Heuristic and RKGA.

Table 65: Factor Indicators for ANOVA

Factor	Indicator
Algorithm used	ALGO\$
Setup Time Range	SETUP\$
Skipping Probability	SKIPPING
Processing Time Range	PROC_RANGE\$
Number of Stages	NUM_STAGES
Machine Distribution and Number of Machines per Stage	NUM_MC\$
Number of Jobs	NUM_JOBS

The levels for the ALGO\$ factor are SPTCH, FTMIH, $g/2$, $g/2$ and RKGA. The levels for the remaining factors follow those indicated in Table 50 with two exceptions. The Machine Distribution and Number of Machines per Stage were combined into one factor with the following five levels: 1, 2, 10, 1-4 and 1-10. The two way factor NUM_MC\$ and NUM_JOBS was not included because its inclusion caused SYSTAT to return an ANOVA regarding the entire model rather than the detailed ANOVA desired; the experiment was not complete because the NUM_JOBS factor could not take on the low level (6 jobs) when the NUM_MC\$ factor was at the 10 level.

Table 66: ANOVA – Multistage Heuristics

Source	Sum of Squares	df	Mean Square	F-ratio	P
ALGO\$	56.634	3	18.878	2234.443	0.000
SETUP\$	0.021	1	0.021	2.479	0.115
SKIPPING	126.873	2	63.436	7508.455	0.000
PROC_RANGES	1.859	1	1.859	220.020	0.000
NUM_STAGES	7.651	2	3.826	452.807	0.000
NUM_MC\$	12.743	4	3.186	377.066	0.000
NUM_JOBS	24.957	3	8.319	984.647	0.000
ALGO\$*SETUP\$	0.007	3	0.002	0.281	0.839
ALGO\$*SKIPPING	11.079	6	1.846	218.551	0.000
ALGO\$*PROC_RANGES	0.624	3	0.208	24.604	0.000
ALGO\$*NUM_STAGES	0.685	6	0.114	13.506	0.000
ALGO\$*NUM_MC\$	2.469	12	0.206	24.356	0.000
ALGO\$*NUM_JOBS	2.565	9	0.285	33.734	0.000
SETUP\$*SKIPPING	0.013	2	0.006	0.749	0.473
SETUP\$*PROC_RANGES	0.011	1	0.011	1.335	0.248
SETUP\$*NUM_STAGES	0.005	2	0.002	0.278	0.757
SETUP\$*NUM_MC\$	0.098	4	0.025	2.915	0.020
SETUP\$*NUM_JOBS	0.057	3	0.019	2.229	0.083
SKIPPING*PROC_RANGES	3.642	2	1.821	215.559	0.000
SKIPPING*NUM_STAGES	3.469	4	0.867	102.654	0.000
SKIPPING*NUM_MC\$	8.320	8	1.040	123.092	0.000
SKIPPING*NUM_JOBS	12.638	6	2.106	249.316	0.000
PROC_RANGES*NUM_STAGES	0.184	2	0.092	10.885	0.000
PROC_RANGES*NUM_MC\$	0.121	4	0.030	3.587	0.006
PROC_RANGES*NUM_JOBS	0.393	3	0.131	15.500	0.000
NUM_STAGES*NUM_MC\$	9.208	8	1.151	136.229	0.000
NUM_STAGES*NUM_JOBS	0.485	6	0.081	9.567	0.000
Error	230.217	27249	0.008		

Table 66 indicates that all the main effects are significant at the 5% level with exception of the SETUP\$ factor. This is as expected because, though the values of the setup times are different in the two levels, the *range* of the two levels is identical at 12. The two way interactions indicate that two way interactions with SETUP\$ are not significant at the 5% level except SETUP\$ and NUM_MC\$. The other two way interactions are significant at the 5% level. These results indicate that most of the experimental levels were sufficiently different to represent different scenarios.

4.8 Conclusions

This chapter has examined ten heuristics to find schedules minimizing makespan in flexible flow lines. These methods included simplistic greedy methods, approaches based on the TSP nature of the problem and the flow line nature of the problem and genetic algorithm approaches. Original lower bounds were developed for use in the evaluation of these heuristics. The lower bounds were investigated for performance and found to be efficient. The data characteristics investigated were designed to reflect characteristics used by other researchers. Through ANOVA, these data characteristics were found to be significant. Six of the ten original heuristics were discarded through comparisons within each class of heuristics. The remaining heuristics (SPTCH, FTMIH, the $g/2$, $g/2$ Johnson's Based Heuristic and RKGA) were compared on 6840 data files. Through paired T-tests it was determined that RKGA performed best on the problems examined here. However, in the specific situation of two stages, jobs that visit each stage and 10 machines per stage, the $g/2$, $g/2$ Johnson's Based Heuristic was found to be effective as well.

5 Multistage Extensions

This chapter addresses extensions of the basic multistage research presented in the previous chapter. First we will consider an improvement method based on a backward pass over an initial schedule. Then we consider using the bottleneck stage as the key stage for scheduling in the heuristics developed in the previous chapter.

5.1 Forward and Backward Pass Method

The makespan criterion allows one job to define the entire objective function value. In the problems examined so far, we have observed that usually, only one job has a completion time equivalent to the makespan. The other jobs end at some time before the makespan. These jobs could be pushed back to end at the makespan time, with no increase in the makespan value. Consider forcing all the last jobs on the machines in the last stage to end at the makespan time. Even if the makespan is not achieved by a job on the last stage, this will not increase the makespan. Now, the other jobs on this stage have some room to move around, between the time that later jobs must begin setup and the time these jobs must begin setup themselves. If we move these jobs to begin as late as possible, we are increasing their *waiting time*, which we define as the time between the job's ready time and start-of-setup time. At each stage, if we can maximize the minimum waiting time, and have this value be larger than 0, we may be able to rearrange the jobs such that stage 1 has a positive waiting time value. In that case, the earliest job begins at some time greater than 0. We have not increased the time the last job ends processing but

have increased the time the first job begins processing. The overall makespan has thus been reduced.

The first step is to generate an initial schedule. This can be done with any of the heuristics developed in the previous chapter. The second step is to schedule the jobs in the last stage to begin setup as late as possible but finish before the makespan time. The previous stages are then scheduled in descending order with the jobs arranged in due date order and being placed first on the machine so that it starts latest. The succeeding sections include a condition under which no rearrangement of jobs can be made that would allow a reduction in the makespan, a theorem that tells us which jobs are eligible for placement before a particular job, and a heuristic developed to perform the backwards pass. We then evaluate the performance of this backward pass heuristic.

5.1.1 Condition

Let $j > 0$ denote a job that defines the current makespan which is achieved in stage t and d_j^t denote the due date of job j in stage t . If $c_j^t = d_j^t$, $r_j^t = d_j^t - p_j^t - \min_{i \neq j, i \in S^t} s_{ij}^t$ and

$$\sum_{i \neq j, i \in S^t} s_{ij}^t x_{ij}^t = \min_{i \neq j, i \in S^t} s_{ij}^t \text{ (job } j \text{ is preceded by the job that has the smallest setup to job } j),$$

then no improvement in the makespan can be made. The job j has no room to move earlier in the schedule without violating its ready time.

5.1.2 Theorem

Consider a job $i > 0$ at stage t in a feasible schedule. Only jobs k such that $s_{ki}^t \leq a_i^t$, where $a_i^t = d_i^t - r_i^t - p_i^t$ need be considered for placement before i at stage t .

Proof:

Let a_i^t be the available time for setting up a potential job before job i at stage t . The available time is the current due date minus the ready time minus the required processing time, or $a_i^t = d_i^t - r_i^t - p_i^t$. Consider a job k such that $s_{ki}^t > a_i^t$. This job cannot be scheduled before i because it will force i to be completed after the due date, even if setup begins as soon as job i is available. Thus, the only jobs that need be considered for scheduling before i at stage t are those jobs k with $s_{ki}^t \leq a_i^t$.

5.1.3 Backward Pass Heuristic

The Backward Pass Heuristic requires an initial schedule. For this research, the $g/2$, $g/2$ Johnson's Rule Based Heuristic will be used because of its low computational requirements. Its inferior performance when compared to RKGA is an advantage here as far as the Backward Pass Heuristic is concerned; since the $g/2$, $g/2$ Johnson's Rule Based Heuristic in general returns schedules with higher makespans, we know that better schedules exist. If the Backward Pass Heuristic cannot find improved schedules with these initial schedules, it is probably not an effective improvement method.

1. Set the due dates of all jobs in stage g to the makespan of the initial solution MS
2. Arrange the jobs in increasing order of due date minus ready time
3. Place each job $[i]$ first on the machine in stage g on which it begins setup latest,

subject to its due date

4. For each stage $t=g, \dots, 1$:
 - a. Update the due dates in stage t to be the start of setup times in stage $t+1$
 - b. Arrange the jobs in decreasing due date order
 - c. Place each job $[i]$ first on the machine in stage t on which it begins setup latest
5. Find the job that begins setup the earliest over all stages. Call this time e .
6. Return the makespan of the new schedule $MS_{new} = MS - e$

This heuristic has been applied to the 6840 test problems described in the previous chapter. Since the $g/2, g/2$ Johnson's Rule Based Heuristic was used to generate the initial schedule, it shall be used to evaluate the performance of the Backward Pass Heuristic. As in the previous chapter, the loss shall be calculated for each problem. The Backward Pass Heuristic caused a worsening of makespan in 5480 of the 6840 test problems. In 745 cases, the makespan did not change after application of the Backward Pass Heuristic. In only 615 cases did the makespan improve after the Backward Pass heuristic. Table 67 contains the paired T-test results comparing the $g/2, g/2$ Johnson's Rule Based Heuristic without and with the Backward Pass Heuristic performed afterwards.

Table 67: Paired T-Test for $g/2, g/2$ Johnson's Rule Based Heuristic and the Backward Pass Heuristic

Heuristics	Mean Difference	Standard Deviation Difference	95 % CI	T-statistic (6839 df)
$g/2, g/2$ – Backward Pass	-0.327	0.558	-0.340, -0.314	-48.468

The results of the paired T-test clearly indicate that the use of the Backward Pass Heuristic results in worse solutions.

5.2 Bottleneck Methods

In this section we adapt two heuristics from the previous chapter to use the bottleneck stage as the key stage for ordering jobs. The reasoning is that the first stage may not be the key stage for controlling the makespan of the whole schedule. Instead, it seems reasonable that a good schedule in the bottleneck stage, with as little idle time as possible, will control the overall makespan of the schedule. The non-bottleneck stages will then accommodate the schedule in the bottleneck stage, facilitating the preservation of the schedule in the bottleneck stage. This method was used in adapting SPTCH and FTMIH from the previous chapter.

BN will be used to indicate the bottleneck stage. Of the 6840 test files discussed in the previous chapter, 2003 have stage 1 as the bottleneck stage. In these, the Bottleneck methods discussed below will give identical results as the original method.

5.2.1 Bottleneck SPTCH (BNSPTCH)

Recall that in SPTCH, the first stage's modified processing times were used to create a SPT order of jobs. At the first stage, jobs were examined in this order and placed last on the machine on which it completes earliest. In the subsequent stages, jobs are examined in ready time order and placed last on the machine on which it ends earliest. BNSPTCH introduces several changes to this. The first change is that the bottleneck

stage is used for the SPT ordering and the ordering is performed first on this stage. Next, the stages before the bottleneck stage are considered. Due dates are created for each job, which are the times setup begins in the next stage. We order the jobs in decreasing due date order and place the jobs first on the machine on which it begins latest. After each stage is scheduled in this way, we update the starting and completion times of each job in each stage scheduled so far. After all the pre-bottleneck stages have been considered, we have a partial schedule for stages 1 to BN. At this point, we revert to the regular SPTCH method. These steps are described below.

1. Find the bottleneck stage, BN. BN is the stage which has the largest value for

$$\frac{1}{m'} \left\{ (m' - 1) p'_0 + \sum_{i \in S'} \left(p'_i + \min_{j \in S'} s'_{ji} \right) \right\}. \text{ Note that job 0 must be included once for}$$

each machine in each stage.

2. Arrange the jobs in stage BN in the SPT order of $\bar{p}_i^{BN}$.
3. Place each job $[i]$ last on the machine in stage t on which it end earliest.
4. For each stage $t=BN-1, \dots, 1$:
 - a. Update the due dates in stage t to be the start of setup times in stage $t+1$
 - b. Arrange the jobs in decreasing due date order
 - c. Place each job $[i]$ on the first on the machine in stage t on which it begins setup latest
5. For each stage $t=1, \dots, BN$:
 - a. Find the completion times of each job
 - b. Set the ready times of the jobs in the subsequent stages to the completion times at stage t
6. For each stage $t=BN+1, \dots, g$:
 - a. Arrange the jobs in increasing ready time order
 - b. Place each job $[i]$ last on the machine in stage t on which it end earliest

- c. Set the ready times of the jobs in the subsequent stages to the completion times at stage t

This heuristic shall be demonstrated with an example. The data set is the same used in Chapter 4 but is repeated here for convenience in Figure 36. Stage 2 is the bottleneck stage with the most workload per machine. Due to this, stage 2 will be ordered first, followed by 1. Then, the start and completion times for stages 1 and 2 will be recalculated. Then, stages 3 and 4 will be scheduled.

Stage 1					
Job	0	2	3	4	5
p_i^1	20	63	62	54	53
$\bar{p}_i^1$	20	66	64	54	57

Stage 2					
Job	0	1	2	5	6
p_i^2	16	70	73	52	57
$\bar{p}_i^2$	16	70	77	58	57

Stage 1				
(s_{ji}^1)	To			
From	2	3	4	5
0	3	2	0	4
2	—	15	15	19
3	23	—	15	15
4	17	22	—	20
5	18	16	20	—
$\min_j s_{ji}^1$	3	2	0	4

Stage 2				
(s_{ji}^2)	To			
From	1	2	5	6
0	4	4	8	7
1	—	4	8	7
2	0	—	6	0
5	19	13	—	16
6	16	17	20	—
$\min_j s_{ji}^2$	0	4	6	0

Stage 3				
Job	0	1	2	4
p_i^3	20	73	71	57
$\bar{p}_i^3$	20	76	73	60

Stage 4				
Job	0	1	2	6
p_i^4	18	70	64	63
$\bar{p}_i^4$	18	70	67	64

Stage 3			
(s_{ji}^3)	To		
From	1	2	4
0	3	2	3
1	—	4	5
2	5	—	6
4	20	19	—
$\min_j s_{ji}^3$	3	2	3

Stage 4			
(s_{ji}^4)	To		
From	1	2	6
0	0	3	5
1	—	4	1
2	4	—	2
6	18	21	—
$\min_j s_{ji}^4$	0	3	1

Figure 36: Processing and Setup Times for Example Problem

The jobs in each stage are considered in the order shown in Table 68. After stage 2 is scheduled, the jobs are sequenced (5, 1 / 6, 2). Stage 1 then has due dates set for each of its jobs. The stage 1 order is then found and is shown in Table 68. Stage 1 jobs are sequenced (4, 2 / 3, 5).

Table 68: Order for BNSPTCH

Stage	2	1	3	4
Job Order	5, 6, 1, 2	2, 5, 4, 3	4, 2, 1	6, 2, 1

After stage 1 has been scheduled, we recalculate the start and completion times for the first two stages. Now we continue with regular SPTCH. The final schedule returned is $\{(4, 2 / 3, 5) / (5, 1 / 6, 2) / (4, 1 / 2) / (2, 1 / 6)\}$ with a makespan of 468 and is shown in

Table 69: BNSPTCH Schedule

Stage 1				Stage 2			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	0	20	1	0	20	36
1	4	20	74	1	5	152	212
1	2	74	154	1	1	212	301
2	0	0	20	2	0	20	36
2	3	20	84	2	6	36	100
2	5	84	152	2	2	154	244

Stage 3				Stage 4			
MC	Job	Setup Starts	Completion Time	MC	Job	Setup Starts	Completion Time
1	0	36	56	1	0	56	74
1	4	74	134	1	2	317	384
1	1	301	394	1	1	394	468
2	0	36	56	2	0	56	74
2	2	244	317	2	6	100	168

5.2.2 Bottleneck FTMIH (BNFTMIH)

Recall that in FTMIH, each stage is scheduled in order. That is, the first is scheduled, then the second, et cetera. Each stage's modified processing times were used to create a LPT order of jobs. At each stage, jobs were examined in this order and placed

in the position on the machine on which the sum of flowtimes for that stage is smallest. BNFTMIH introduces several changes to this. The first change is that the bottleneck stage is used for the LPT ordering and the basic FTMIH is performed first on this stage. Next, the stages before the bottleneck stage are considered. Due dates are created for each job which are the times setup begins in the next stage. We order the jobs in decreasing due date order and place the jobs first on the machine on which it begins latest. After each stage is scheduled in this way, we update the starting and completion times of each job in each stage scheduled so far. After all the pre-bottleneck stages have been considered, we have a partial schedule for stages 1 to BN. At this point, we revert to the regular SPTCH method. These steps are described below.

1. Find the bottleneck stage, BN. BN is the stage which has the largest value for $\frac{1}{m'} \left\{ (m' - 1) p_0' + \sum_{i \in S^t} \left(p_i' + \min_{j \in S^t} s_{ji}' \right) \right\}$. Note that job 0 must be included once for each machine in each stage.
2. Arrange the jobs in stage BN in the LPT order of $\bar{p}_i^{BN}$.
3. For $[i]=1$ to n , $i \in S^{BN}$:
 - a. insert job $[i]$ into every position on each machine
 - b. calculate the true sum of flowtimes using the actual setup times
 - c. place job i in the position on the machine with the lowest resultant sum of flowtimes
4. For each stage $t=BN-1, \dots, 1$:
 - a. Update the due dates in stage t to be the start of setup times in stage $t+1$
 - b. Arrange the jobs in decreasing due date order
 - c. Place each job $[i]$ on the first on the machine in stage t on which it begins setup latest.
5. For each stage $t=1, \dots, BN$:

- a. Find the completion times of each job
- b. Set the ready times of the jobs in the subsequent stages to the completion times at stage t
6. For each stage $t=BN+1, \dots, g$:
 - a. Arrange the jobs in increasing ready time order
 - b. Place each job $[i]$ last on the machine in stage t on which it end earliest.
 - c. Set the ready times of the jobs in the subsequent stages to the completion times at stage t

5.2.3 Comparing SPTCH and BNSPTCH

The 6840 test problems discussed in the previous chapter were subjected to BNSPTCH. Of the 4837 problems in which the bottleneck stage was not stage 1, only 329 saw improvement with BNSPTCH, 43 problems had the same makespan and 4465 problems had worse makespans. The losses of SPTCH and BNSPTCH were compared in a paired T-test which is shown in Table 57. With 95% confidence, SPTCH outperforms BNSPTCH.

Table 70: Paired T-Test for SPTCH and BNSPTCH

Heuristics	Mean Difference	Standard Deviation Difference	95 % CI	T-statistic (6839 df)
SPTCH - BNSPTCH	-0.209	0.384	-0.219, -0.200	-45.062

Given the results of the paired T-test and the low number of improved makespans with BNSPTCH, SPTCH is a better method.

5.2.4 Comparing FTMIH and BNFTMIH

The 6840 test problems discussed in the previous chapter were subjected to BNFTMIH. Of the 4837 problems in which the bottleneck stage was not stage 1, only 994 saw improvement with BNFTMIH. The losses of FTMIH and BNFTMIH were compared in a paired T-test which is shown in Table 71. With 95% confidence, FTMIH outperforms BNFTMIH.

Table 71: Paired T-Test for FTMIH and BNFTMIH

Heuristics	Mean Difference	Standard Deviation Difference	95% CI	T-statistic (6839 df)
FTMIH-BNFTMIH	-0.200	0.379	-0.209, -0.191	-43.642

Given the results of the paired T-test and the low number of improved makespans with BNFTMIH, FTMIH is a better method.

5.2.5 Value of Bottleneck Knowledge

Using the bottleneck stage as the key stage for scheduling a flexible flow line has been attempted in two methods. SPTCH and FTMIH were adapted to use the bottleneck stage scheduling method. In both cases, the bottleneck stage scheduling method generally resulted in worse schedules. The order in which jobs enter the system, whether all the jobs enter in stage 1 or not, seems to have a much larger effect on the quality of the solution than the amount of idle time at the bottleneck stage. In conclusion, flexible flow lines should be scheduled in numerical stage order rather than trying to accommodate workload at different stages.

5.3 Conclusions

This chapter has contained two methodologies for the flexible flow line with sequence-dependent setup times makespan minimization problem. The first was an improvement technique which intended to rearrange jobs to end as late as possible but not violate the original makespan as the due dates for all jobs. This technique proved to be worse than ineffectual; it worsens the majority of schedules in terms of makespans. The second methodology was based on the observation that the bottleneck stage may be fundamentally important in determining the makespan of the entire schedule. This observation motivated the adaptation of two of the previously developed heuristics, SPTCH and FTMIH, to set the schedule at the bottleneck stage first, followed by the stages before the bottleneck and concluding with the stages after the bottleneck. This approach also proved to be ineffective.

6 Conclusions and Areas for Future Research

There are potentially unlimited opportunities for research in scheduling to minimize makespan in flexible flow lines with sequence-dependent setup times. As noted in the Literature Review in Section 2, no research has been published in this area. In this dissertation, we have addressed only a few areas. We have examined the use of single stage heuristics in the multiple stage environment. Both the NEH method for flow lines (developed here as an application of the Insertion heuristic) and the MULTI-FIT method for parallel machines were investigated for application to the single stage problem. An integer programming model for the flexible flow line with sequence-dependent setup times was developed, but not thoroughly explored. This model incorporates the ability of jobs to skip stages. Discussion of permutation schedules in flexible flow lines indicated that simple extensions of the definition of permutation schedule would not suffice. This indicates that not restricting our search for good solutions to the class of permutation schedules may result in improved schedules. We adapted TSP methods to the flexible flow line in the Multiple Insertion Heuristics. We also investigated the adaptation of flow line solution methods, such as Johnson's Rule, to the flexible flow line. Two types of genetic algorithms were considered which differed most notable in their problem representation, genetic operators and stopping criteria. In all, ten heuristics were developed. These heuristics were compared through an empirical evaluation, using data generated with characteristics proposed by previous researchers. The heuristics were evaluated in comparison to lower bounds developed for the flexible

flow line with sequence-dependent setup times makespan minimization problem. Some extensions to the basic methodologies for the multiple stage scheduling problem were offered, which, while unsuccessful, yielded insight that helps guide our future research through the conclusions drawn. Through this research, we can make the following conclusions:

1. The lower bounds introduced in Chapter 4 are very good. Since RKGa found some schedules with makespans equivalent to the lower bound or within 1% of the lower bound, it is unlikely the lower bound can be significantly strengthened without solving the problem.
2. The order in which jobs enter the flexible flow line strongly affects the resulting makespan. This is supported by the results in Chapters 4 and 5. Good starting sequences in the heuristics in Chapter 4 resulted in better makespans. Using the bottleneck stage as the focal stage did not improve, but often worsened, the makespan of the resultant schedule.
3. Improvement methods based on backward passes over an initial schedule were not successful. This is interpreted to mean that the relative positioning of jobs in an initial schedule is “strong”, meaning that the jobs have relatively little freedom in the schedule. Again, the entering order of the jobs strongly affects the makespan, and using a backward pass to change this damages the order in the initial schedule.

The research in this dissertation has lead to many more questions regarding flexible flow lines with sequence-dependent setup times. Some areas of future research are discussed below.

1. Is there a definition of a permutation schedule that is general enough to handle multiple machines in serial stages where not all jobs visit each stage and sequence-dependent setup times exist? Potential definitions may include a condition that cannot be violated, but does not tell explicitly what all the permutations may look like. For example, a permutation schedule may be one in which the relative ordering of start of setup times at each stage does not change from stage to stage. This definition in no way tells us what jobs should be assigned to what machine in stages with multiple machines and in fact allows for several alternative assignments at each stage. However, this definition also includes the traditional definition of a permutation schedule in a flow line. By creating a general permutation schedule definition, we may be able to find a class of schedules that contains the optimal makespan schedule for some special case, such as two stages with one machine at the first stage and two machines at the second, which sequence-dependent setup times at both.
2. The integer programming formulation in Chapter 4 for the flexible flow line with sequence-dependent setup times was not thoroughly investigated. It has been solved for some specific cases, such as the example problem in Chapter 4. But its performance on much larger problems is unknown.

3. The integer programming formulation developed for the flexible flow line with sequence-dependent setup times can be used to provide optimal solutions to many problem instances. These optimal makespans can be used to further evaluate the effectiveness of the lower bounds developed in Chapter 4.
4. A theorem regarding schedule domination can also be used to help in a branch-and-bound solution methodology in order to find optimal solutions. This branch-and-bound methodology may be able to find optimal solutions that are not computationally feasible with the integer programming formulation.
5. An improvement method may be possible using the insight that a single job may be “binding” the schedule. This is the same insight that motivated the Backward Pass Heuristic, though that heuristic only examined each job in a single stage. By looking at which job defines the makespan, we may be able to rearrange it in all stages to reduce the makespan. An application of the motivation for the Shifting Bottleneck Heuristic may be applicable to this concept.
6. Two genetic algorithms were developed for the multiple stage problem. The random keys genetic algorithm performed much better despite nearly identical or lower running times for most problems. RKGGA used only stage 1 as the chromosomal information; the schedules in the subsequent stages were determined by a ready time greedy method. Would using the entire schedule at each stage result in better performance? Would the larger chromosome,

with each job being explicitly considered in each stage, result in better makespans at less time cost? This computational experiment will be performed in the future.

7. This research can also lead to other research in genetic algorithms. Is there any way to know what chromosomal representation is appropriate for a particular type of problem? Do the various genetic operators matter in the time required to find a good solution, or in a genetic algorithm's ability to even find good solutions? These are large questions. By thoroughly experimenting with the flexible flow line problem, some insights can be found in these areas.

Appendix A: ANOVA - Slicing Heuristic

Source	Sum-of-Squares	df	Mean-Square	F-ratio	P
<u>(ALL MAIN EFFECTS)</u>					
MST_ALGO	0.180	1	0.180	12.210	0.000
PROC_RANGE	0.227	1	0.227	15.406	0.000
PROC_MEAN	0.130	1	0.130	8.816	0.003
SETUP_STRUCT	0.003	1	0.003	0.200	0.655
SETUP_STDDEV	0.057	1	0.057	3.896	0.049
NUM_MC	0.025	1	0.025	1.676	0.196
AVG_NUM_JOBS	0.758	1	0.758	51.501	0.000
<u>(TWO WAY INTERACTION EFFECTS WITH $P \leq 0.05$)</u>					
MST_ALGO*NUM_MC	0.173	1	0.173	11.740	0.001
MST_ALGO*AVG_NUM_JOBS	0.175	1	0.175	11.887	0.001
PROC_RANGE					
*AVG_NUM_JOBS	0.080	1	0.080	5.439	0.020
NUM_MC					
*AVG_NUM_JOBS	0.086	1	0.086	5.838	0.016
MST_ALGO*NUM_MC					
*AVG_NUM_JOBS	0.161	1	0.161	10.921	0.001
<u>(NO LARGER WAY INTERACTION EFFECTS HAD $P \leq 0.05$)</u>					
Error	16.950	1152	0.015		

Appendix B: ANOVA - Multiple MULTI-FIT Heuristic

Source	Sum-of-Squares	df	Mean-Square	F-ratio	P
<u>(ALL MAIN EFFECTS)</u>					
SETUP_FUN	0.017	2	0.008	167.686	0.000
PROC_RANGE	0.035	1	0.035	709.262	0.000
PROC_MEAN	0.260	1	0.260	5220.344	0.000
SETUP_STRUCT	0.001	1	0.001	27.974	0.000
SETUP_STDDEV	0.079	1	0.079	1587.582	0.000
NUM_MC	0.040	1	0.040	797.688	0.000
AVG_NUM_JOBS	0.009	1	0.009	189.891	0.000
<u>(TWO WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
SETUP_FUN*PROC_MEAN	0.007	2	0.004	73.486	0.000
SETUP_FUN*SETUP_STDDEV	0.004	2	0.002	44.211	0.000
SETUP_FUN*NUM_MC	0.001	2	0.001	14.258	0.000
SETUP_FUN*AVG_NUM_JOBS	0.004	2	0.002	40.035	0.000
PROC_RANGE					
*PROC_MEAN	0.009	1	0.009	189.317	0.000
PROC_RANGE					
*SETUP_STDDEV	0.000	1	0.000	7.937	0.005
PROC_RANGE					
*AVG_NUM_JOBS	0.029	1	0.029	577.814	0.000
PROC_MEAN					
*SETUP_STRUCT	0.000	1	0.000	9.111	0.003
PROC_MEAN					
*SETUP_STDDEV	0.044	1	0.044	882.033	0.000
PROC_MEAN*NUM_MC	0.004	1	0.004	86.241	0.000
SETUP_STRUCT					
*SETUP_STDDEV	0.000	1	0.000	8.544	0.004
SETUP_STDDEV					
*NUM_MC	0.001	1	0.001	14.161	0.000
SETUP_STDDEV					
*AVG_NUM_JOBS	0.001	1	0.001	12.536	0.000
NUM_MC					
*AVG_NUM_JOBS	0.002	1	0.002	33.570	0.000
<u>(THREE WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
SETUP_FUN*PROC_RANGE					
*AVG_NUM_JOBS	0.002	2	0.001	18.928	0.000
SETUP_FUN*PROC_MEAN					
*SETUP_STDDEV	0.003	2	0.001	28.496	0.000
SETUP_FUN*PROC_MEAN					
*NUM_MC	0.001	2	0.000	5.982	0.003
SETUP_FUN*PROC_MEAN					
*AVG_NUM_JOBS	0.003	2	0.002	32.014	0.000
SETUP_FUN*SETUP_STDDEV					
*AVG_NUM_JOBS	0.001	2	0.001	12.860	0.000
SETUP_FUN*NUM_MC					
*AVG_NUM_JOBS	0.000	2	0.000	3.313	0.037
PROC_RANGE					
*PROC_MEAN					
*SETUP_STDDEV	0.001	1	0.001	15.191	0.000
PROC_RANGE					
*PROC_MEAN*NUM_MC	0.001	1	0.001	23.475	0.000
PROC_RANGE					
*PROC_MEAN					
*AVG_NUM_JOBS	0.001	1	0.001	10.784	0.001
PROC_RANGE					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	4.156	0.042
PROC_RANGE					
*SETUP_STDDEV					
*NUM_MC	0.000	1	0.000	4.278	0.039
PROC_RANGE*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	11.674	0.001

PROC_MEAN					
*SETUP STDDEV					
*NUM_MC	0.001	1	0.001	17.857	0.000
PROC_MEAN					
*SETUP STDDEV					
*AVG_NUM_JOBS	0.001	1	0.001	23.782	0.000
PROC_MEAN*NUM_MC					
*AVG_NUM_JOBS	0.003	1	0.003	62.525	0.000
SETUP_STRUCT					
*SETUP STDDEV					
*NUM_MC	0.000	1	0.000	4.578	0.033
SETUP STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.002	1	0.002	35.365	0.000
(FOUR WAY INTERACTION EFFECTS WITH P≤0.05)					
SETUP_FUN*PROC_RANGE					
*PROC_MEAN*NUM_MC	0.000	2	0.000	4.265	0.014
SETUP_FUN*PROC_MEAN					
*SETUP STDDEV					
*NUM_MC	0.000	2	0.000	3.203	0.041
SETUP_FUN*PROC_MEAN					
*SETUP STDDEV					
*AVG_NUM_JOBS	0.001	2	0.000	9.420	0.000
SETUP_FUN*PROC_MEAN					
*NUM_MC					
*AVG_NUM_JOBS	0.000	2	0.000	4.737	0.009
PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	4.511	0.034
PROC_MEAN					
*SETUP_STRUCT					
*SETUP STDDEV					
*NUM_MC	0.000	1	0.000	5.339	0.021
PROC_MEAN					
*SETUP STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	7.490	0.006
(FIVE WAY INTERACTION EFFECTS WITH P≤0.05)					
SETUP_FUN*PROC_RANGE					
*PROC_MEAN					
*SETUP STDDEV					
*AVG_NUM_JOBS	0.000	2	0.000	3.698	0.025
(SIX WAY INTERACTION EFFECTS WITH P≤0.05)					
PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*SETUP STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	6.275	0.012
Error	0.086	1728	0.000		

Appendix C: ANOVA - Multiple Insertion Heuristic

Source	Sum-of-Squares	df	Mean-Square	F-ratio	P
<u>(ALL MAIN EFFECTS)</u>					
SETUP_FUN	0.000	2	0.000	0.743	0.476
PROC_RANGE	0.008	1	0.008	410.566	0.000
PROC_MEAN	0.073	1	0.073	3701.615	0.000
SETUP_STRUCT	0.001	1	0.001	62.814	0.000
SETUP_STDDEV	0.023	1	0.023	1158.658	0.000
NUM_MC	0.004	1	0.004	221.971	0.000
AVG_NUM_JOBS	0.060	1	0.060	3055.707	0.000
<u>(TWO WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
PROC_RANGE					
*PROC_MEAN	0.004	1	0.004	196.721	0.000
PROC_RANGE					
*SETUP_STRUCT	0.000	1	0.000	6.071	0.014
PROC_RANGE					
*SETUP_STDDEV	0.000	1	0.000	18.859	0.000
PROC_RANGE					
*AVG_NUM_JOBS	0.008	1	0.008	411.781	0.000
PROC_MEAN					
*SETUP_STDDEV	0.013	1	0.013	673.365	0.000
PROC_MEAN*NUM_MC	0.001	1	0.001	59.221	0.000
PROC_MEAN					
*AVG_NUM_JOBS	0.003	1	0.003	146.918	0.000
SETUP_STRUCT					
*SETUP_STDDEV	0.000	1	0.000	21.627	0.000
SETUP_STRUCT					
*NUM_MC	0.000	1	0.000	6.170	0.013
SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	20.437	0.000
SETUP_STDDEV					
*NUM_MC	0.001	1	0.001	28.439	0.000
SETUP_STDDEV					
*AVG_NUM_JOBS	0.002	1	0.002	88.713	0.000
NUM_MC					
*AVG_NUM_JOBS	0.003	1	0.003	164.802	0.000
<u>(THREE WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT	0.000	1	0.000	19.991	0.000
PROC_RANGE					
*PROC_MEAN					
*SETUP_STDDEV	0.000	1	0.000	8.267	0.004
PROC_RANGE					
*PROC_MEAN*NUM_MC	0.000	1	0.000	18.166	0.000
PROC_RANGE					
*PROC_MEAN					
*AVG_NUM_JOBS	0.001	1	0.001	61.820	0.000
PROC_RANGE					
*SETUP_STRUCT					
*SETUP_STDDEV	0.000	1	0.000	8.642	0.003
PROC_RANGE					
*SETUP_STRUCT					
*NUM_MC	0.000	1	0.000	15.614	0.000
PROC_RANGE					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	9.452	0.002
PROC_RANGE					
*SETUP_STDDEV					
*AVG_NUM_JOBS	0.000	1	0.000	4.384	0.036
PROC_RANGE*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	4.637	0.031
PROC_MEAN					

*SETUP_STRUCT					
*SETUP_STDDEV	0.000	1	0.000	5.266	0.022
PROC_MEAN					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	4.930	0.027
PROC_MEAN					
*SETUP_STDDEV					
*NUM_MC	0.000	1	0.000	8.635	0.003
PROC_MEAN					
*SETUP_STDDEV					
*AVG_NUM_JOBS	0.001	1	0.001	73.350	0.000
PROC_MEAN*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	34.333	0.000
SETUP_STRUCT					
*SETUP_STDDEV					
*AVG_NUM_JOBS	0.000	1	0.000	11.836	0.001
SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	13.294	0.000

(FOUR WAY INTERACTION EFFECTS WITH P≤0.05)

PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.000	1	0.000	12.484	0.000
PROC_RANGE					
*PROC_MEAN					
*SETUP_STDDEV					
*NUM_MC	0.000	1	0.000	5.645	0.018
PROC_RANGE					
*SETUP_STRUCT					
*SETUP_STDDEV					
*AVG_NUM_JOBS	0.000	1	0.000	10.995	0.001
PROC_RANGE					
*SETUP_STRUCT					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	11.073	0.001
PROC_MEAN					
*SETUP_STRUCT					
*SETUP_STDDEV					
*NUM_MC	0.000	1	0.000	9.320	0.002
PROC_MEAN					
*SETUP_STRUCT					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	4.489	0.034

(FIVE WAY INTERACTION EFFECTS WITH P≤0.05)

PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*SETUP_STDDEV					
*NUM_MC	0.000	1	0.000	17.575	0.000
PROC_MEAN					
*SETUP_STRUCT					
*SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	9.792	0.002

(SIX WAY INTERACTION EFFECTS WITH P≤0.05)

PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.000	1	0.000	17.740	0.000
Error	0.034	1728	0.000		

Appendix D: ANOVA - Comparing Heuristics – No Ready Times

Source	Sum-of-Squares	df	Mean-Square	F-ratio	P
<u>(ALL MAIN EFFECTS)</u>					
ALGO	0.317	3	0.106	836.382	0.000
PROC_RANGE	0.044	1	0.044	345.532	0.000
PROC_MEAN	0.159	1	0.159	1258.078	0.000
SETUP_STRUCT	0.000	1	0.000	0.359	0.549
SETUP_STDDEV	0.042	1	0.042	336.165	0.000
NUM_MC	0.341	1	0.341	2698.875	0.000
AVG_NUM_JOBS	0.224	1	0.224	1770.202	0.000
<u>(TWO WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
ALGO*PROC_RANGE	0.120	3	0.040	315.267	0.000
ALGO*PROC_MEAN	0.007	3	0.002	17.514	0.000
ALGO*SETUP_STRUCT	0.001	3	0.000	3.403	0.017
ALGO*SETUP_STDDEV	0.001	3	0.000	2.818	0.038
ALGO*NUM_MC	0.420	3	0.140	1108.906	0.000
ALGO*AVG_NUM_JOBS	0.063	3	0.021	167.313	0.000
PROC_RANGE					
*PROC_MEAN	0.001	1	0.001	6.132	0.013
PROC_RANGE*NUM_MC	0.001	1	0.001	10.318	0.001
PROC_RANGE					
*AVG_NUM_JOBS	0.027	1	0.027	210.898	0.000
PROC_MEAN					
*SETUP_STDDEV	0.023	1	0.023	178.516	0.000
PROC_MEAN*NUM_MC	0.005	1	0.005	41.674	0.000
PROC_MEAN					
*AVG_NUM_JOBS	0.001	1	0.001	11.253	0.001
SETUP_STRUCT					
*NUM_MC	0.001	1	0.001	4.738	0.030
SETUP_STRUCT					
*AVG_NUM_JOBS	0.001	1	0.001	7.906	0.005
SETUP_STDDEV					
*NUM_MC	0.001	1	0.001	8.557	0.003
SETUP_STDDEV					
*AVG_NUM_JOBS	0.001	1	0.001	4.708	0.030
NUM_MC					
*AVG_NUM_JOBS	0.053	1	0.053	416.539	0.000
<u>(THREE WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
ALGO*PROC_RANGE					
*PROC_MEAN	0.011	3	0.004	29.387	0.000
ALGO*PROC_RANGE					
*NUM_MC	0.030	3	0.010	79.030	0.000
ALGO*PROC_RANGE					
*AVG_NUM_JOBS	0.053	3	0.018	140.901	0.000
ALGO*PROC_MEAN					
*AVG_NUM_JOBS	0.002	3	0.001	5.832	0.001
ALGO*NUM_MC					
*AVG_NUM_JOBS	0.053	3	0.018	140.421	0.000
PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT	0.001	1	0.001	4.421	0.036
PROC_RANGE					
*PROC_MEAN					
*SETUP_STDDEV	0.001	1	0.001	5.065	0.025
PROC_RANGE					
*PROC_MEAN*NUM_MC	0.001	1	0.001	4.256	0.039
PROC_RANGE					
*PROC_MEAN					
*AVG_NUM_JOBS	0.001	1	0.001	9.823	0.002
PROC_RANGE*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	5.219	0.022
PROC_MEAN					
*SETUP_STDDEV					

*NUM_MC	0.001	1	0.001	11.017	0.001
PROC_MEAN*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	11.837	0.001
SETUP_STRUCT					
*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	4.667	0.031
SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	8.702	0.003
<u>(FOUR WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
ALGO*PROC_RANGE					
*PROC_MEAN*NUM_MC	0.001	3	0.000	2.972	0.031
ALGO*PROC_RANGE					
*NUM_MC					
*AVG_NUM_JOBS	0.016	3	0.005	43.521	0.000
ALGO*PROC_MEAN					
*SETUP_STDDEV					
*NUM_MC	0.001	3	0.000	2.713	0.043
ALGO*SETUP_STRUCT					
*NUM_MC					
*AVG_NUM_JOBS	0.001	3	0.000	3.492	0.015
PROC_RANGE					
*PROC_MEAN					
*SETUP_STRUCT					
*AVG_NUM_JOBS	0.001	1	0.001	6.260	0.012
PROC_MEAN					
*SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.001	1	0.001	5.771	0.016
<u>(FIVE WAY INTERACTION EFFECTS WITH P≤0.05)</u>					
ALGO*PROC_MEAN					
*SETUP_STDDEV					
*NUM_MC					
*AVG_NUM_JOBS	0.002	3	0.001	4.572	0.003
<u>(NO LARGER WAY INTERACTION EFFECTS HAD P≤0.05)</u>					
Error	0.291	2304	0.000		

Appendix E: Multistage Heuristics – Loss Statistics

Format of each cell: average (standard deviation) in loss = (makespan - lb)/lb
CH, RCH, SPTCH, FTMIH, CTMIH, MMIH, 1, g Johnson's, g/2, g/2 Johnson's, GA, RKGA

Note: GA was only run for low setup time files, shown as X in high setup time files

Low Setup Times, 1 Machine per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.07 (0.02)	0.17 (0.06)	0.10 (0.07)	0.21 (0.15)	0.32 (0.17)	0.40 (0.20)
		0.07 (0.02)	0.17 (0.06)	0.10 (0.07)	0.15 (0.08)	0.21 (0.11)	0.25 (0.20)
		0.06 (0.02)	0.06 (0.03)	0.09 (0.06)	0.12 (0.09)	0.26 (0.21)	0.22 (0.09)
		0.04 (0.01)	0.07 (0.05)	0.08 (0.06)	0.11 (0.09)	0.20 (0.14)	0.18 (0.09)
		0.04 (0.01)	0.07 (0.05)	0.08 (0.06)	0.11 (0.09)	0.20 (0.14)	0.18 (0.09)
		0.10 (0.06)	0.14 (0.09)	0.10 (0.05)	0.15 (0.09)	0.18 (0.11)	0.21 (0.17)
		0.05 (0.01)	0.05 (0.03)	0.05 (0.02)	0.07 (0.03)	0.15 (0.08)	0.16 (0.09)
		0.05 (0.01)	0.05 (0.03)	0.05 (0.02)	0.07 (0.03)	0.15 (0.08)	0.16 (0.09)
		0.02 (0.01)	0.02 (0.02)	0.02 (0.01)	0.04 (0.03)	0.10 (0.08)	0.10 (0.08)
		0.02 (0.01)	0.02 (0.01)	0.06 (0.01)	0.04 (0.02)	0.14 (0.01)	0.04 (0.01)
		0.09 (0.02)	0.11 (0.03)	0.17 (0.08)	0.24 (0.10)	0.36 (0.09)	0.35 (0.11)
		0.09 (0.02)	0.11 (0.03)	0.15 (0.07)	0.21 (0.09)	0.29 (0.08)	0.26 (0.05)
	20	0.07 (0.01)	0.10 (0.02)	0.13 (0.06)	0.17 (0.09)	0.29 (0.09)	0.26 (0.06)
		0.05 (0.01)	0.06 (0.01)	0.11 (0.06)	0.14 (0.08)	0.22 (0.06)	0.23 (0.05)
		0.05 (0.01)	0.06 (0.01)	0.11 (0.06)	0.14 (0.08)	0.22 (0.06)	0.23 (0.05)
		0.20 (0.05)	0.16 (0.04)	0.24 (0.11)	0.27 (0.10)	0.22 (0.09)	0.22 (0.08)
		0.07 (0.01)	0.07 (0.01)	0.11 (0.06)	0.14 (0.08)	0.23 (0.05)	0.23 (0.03)
		0.07 (0.01)	0.07 (0.01)	0.11 (0.06)	0.14 (0.08)	0.23 (0.05)	0.23 (0.03)
		0.04 (0.00)	0.05 (0.01)	0.08 (0.06)	0.12 (0.07)	0.13 (0.05)	0.15 (0.06)
		0.02 (0.01)	0.03 (0.01)	0.06 (0.01)	0.07 (0.01)	0.03 (0.01)	0.08 (0.02)
		0.09 (0.01)	0.12 (0.02)	0.18 (0.07)	0.24 (0.06)	0.40 (0.15)	0.33 (0.11)
		0.09 (0.01)	0.12 (0.02)	0.16 (0.07)	0.22 (0.07)	0.28 (0.05)	0.27 (0.04)
		0.08 (0.01)	0.08 (0.01)	0.16 (0.06)	0.19 (0.06)	0.28 (0.04)	0.27 (0.05)
		0.06 (0.02)	0.05 (0.01)	0.11 (0.06)	0.15 (0.07)	0.23 (0.03)	0.25 (0.05)
	30	0.06 (0.02)	0.05 (0.01)	0.11 (0.06)	0.15 (0.07)	0.23 (0.03)	0.25 (0.05)
		0.23 (0.04)	0.20 (0.04)	0.27 (0.10)	0.33 (0.09)	0.27 (0.09)	0.24 (0.05)
		0.08 (0.01)	0.08 (0.01)	0.12 (0.05)	0.16 (0.06)	0.26 (0.03)	0.26 (0.04)
		0.08 (0.01)	0.08 (0.01)	0.12 (0.05)	0.16 (0.06)	0.26 (0.03)	0.26 (0.04)
		0.06 (0.01)	0.06 (0.01)	0.11 (0.06)	0.14 (0.06)	0.20 (0.03)	0.20 (0.04)
		0.03 (0.01)	0.05 (0.02)	0.07 (0.01)	0.06 (0.01)	0.03 (0.01)	0.04 (0.01)
		0.09 (0.00)	0.09 (0.01)	0.24 (0.06)	0.27 (0.05)	0.34 (0.05)	0.35 (0.03)
		0.09 (0.00)	0.09 (0.01)	0.23 (0.06)	0.25 (0.05)	0.29 (0.01)	0.29 (0.02)
		0.08 (0.00)	0.08 (0.00)	0.22 (0.06)	0.24 (0.05)	0.29 (0.01)	0.29 (0.02)
		0.07 (0.01)	0.06 (0.00)	0.19 (0.06)	0.21 (0.05)	0.23 (0.01)	0.24 (0.02)
		0.84 (0.12)	0.82 (0.14)	1.08 (0.20)	0.98 (0.18)	0.79 (0.35)	0.68 (0.32)
		0.29 (0.02)	0.26 (0.02)	0.43 (0.09)	0.40 (0.05)	0.32 (0.06)	0.32 (0.04)
	100	0.08 (0.00)	0.08 (0.01)	0.21 (0.06)	0.23 (0.05)	0.29 (0.01)	0.28 (0.01)
		0.08 (0.00)	0.08 (0.01)	0.21 (0.06)	0.23 (0.05)	0.29 (0.01)	0.28 (0.01)
		0.07 (0.00)	0.07 (0.00)	0.20 (0.06)	0.21 (0.05)	0.22 (0.05)	0.21 (0.05)
		0.04 (0.00)	0.07 (0.09)	0.10 (0.01)	0.07 (0.00)	0.04 (0.01)	0.04 (0.01)

Loss Values - Low Setup Times, 1 Machine per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.02)	0.19 (0.03)	0.17 (0.06)	0.29 (0.11)	0.39 (0.14)	0.57 (0.28)
		0.09 (0.02)	0.19 (0.03)	0.12 (0.05)	0.22 (0.11)	0.31 (0.16)	0.34 (0.13)
		0.07 (0.02)	0.14 (0.09)	0.10 (0.04)	0.19 (0.10)	0.34 (0.19)	0.29 (0.16)
		0.07 (0.02)	0.13 (0.07)	0.12 (0.04)	0.21 (0.11)	0.28 (0.17)	0.28 (0.15)
		0.07 (0.02)	0.13 (0.07)	0.12 (0.04)	0.21 (0.11)	0.28 (0.17)	0.28 (0.15)
		0.16 (0.07)	0.23 (0.09)	0.19 (0.05)	0.30 (0.13)	0.27 (0.12)	0.24 (0.15)
		0.07 (0.02)	0.13 (0.07)	0.09 (0.04)	0.16 (0.06)	0.26 (0.13)	0.29 (0.14)
		0.07 (0.01)	0.12 (0.05)	0.08 (0.06)	0.16 (0.07)	0.23 (0.13)	0.22 (0.12)
		0.04 (0.01)	0.08 (0.04)	0.06 (0.04)	0.10 (0.06)	0.11 (0.07)	0.12 (0.07)
		0.05 (0.02)	0.05 (0.05)	0.10 (0.02)	0.07 (0.00)	0.06 (0.01)	0.05 (0.01)
	20	0.11 (0.01)	0.20 (0.05)	0.20 (0.06)	0.28 (0.10)	0.58 (0.13)	0.53 (0.15)
		0.11 (0.01)	0.20 (0.05)	0.17 (0.06)	0.26 (0.09)	0.35 (0.10)	0.33 (0.10)
		0.09 (0.01)	0.17 (0.04)	0.17 (0.07)	0.22 (0.05)	0.38 (0.12)	0.33 (0.10)
		0.09 (0.01)	0.14 (0.04)	0.16 (0.07)	0.19 (0.06)	0.33 (0.10)	0.28 (0.08)
		0.09 (0.01)	0.14 (0.04)	0.16 (0.07)	0.19 (0.06)	0.33 (0.10)	0.28 (0.08)
		0.54 (0.08)	0.49 (0.09)	0.58 (0.14)	0.55 (0.09)	0.45 (0.08)	0.39 (0.13)
		0.09 (0.01)	0.13 (0.02)	0.14 (0.06)	0.16 (0.05)	0.28 (0.09)	0.30 (0.08)
		0.08 (0.01)	0.13 (0.04)	0.14 (0.05)	0.18 (0.06)	0.26 (0.10)	0.25 (0.04)
		0.06 (0.01)	0.09 (0.02)	0.12 (0.05)	0.13 (0.04)	0.20 (0.06)	0.19 (0.03)
		0.04 (0.01)	0.03 (0.02)	0.14 (0.02)	0.07 (0.01)	0.03 (0.02)	0.18 (0.04)
	30	0.11 (0.01)	0.18 (0.05)	0.21 (0.05)	0.25 (0.06)	0.51 (0.10)	0.60 (0.12)
		0.11 (0.01)	0.18 (0.05)	0.17 (0.05)	0.22 (0.07)	0.32 (0.05)	0.39 (0.09)
		0.09 (0.01)	0.13 (0.03)	0.18 (0.05)	0.19 (0.07)	0.36 (0.07)	0.33 (0.06)
		0.09 (0.01)	0.11 (0.03)	0.16 (0.04)	0.16 (0.06)	0.30 (0.06)	0.30 (0.06)
		0.09 (0.01)	0.11 (0.03)	0.16 (0.04)	0.16 (0.06)	0.30 (0.06)	0.30 (0.06)
		0.64 (0.04)	0.58 (0.08)	0.66 (0.10)	0.62 (0.10)	0.54 (0.10)	0.58 (0.13)
		0.09 (0.01)	0.11 (0.02)	0.15 (0.03)	0.19 (0.08)	0.30 (0.05)	0.31 (0.06)
		0.09 (0.01)	0.11 (0.02)	0.14 (0.04)	0.17 (0.07)	0.30 (0.06)	0.29 (0.04)
		0.07 (0.01)	0.09 (0.02)	0.13 (0.04)	0.14 (0.06)	0.23 (0.04)	0.25 (0.06)
		0.04 (0.01)	0.07 (0.03)	0.03 (0.02)	0.07 (0.01)	0.03 (0.01)	0.19 (0.06)
	100	0.10 (0.01)	0.14 (0.02)	0.27 (0.03)	0.29 (0.06)	0.42 (0.04)	0.47 (0.09)
		0.10 (0.01)	0.14 (0.02)	0.25 (0.04)	0.25 (0.05)	0.33 (0.03)	0.32 (0.03)
		0.09 (0.01)	0.12 (0.01)	0.24 (0.04)	0.25 (0.07)	0.31 (0.03)	0.31 (0.03)
		0.09 (0.01)	0.10 (0.02)	0.22 (0.04)	0.23 (0.12)	0.26 (0.02)	0.31 (0.16)
		2.58 (0.13)	2.57 (0.13)	3.02 (0.32)	2.89 (0.30)	2.25 (0.32)	1.95 (0.55)
		0.83 (0.04)	0.74 (0.05)	0.98 (0.11)	0.89 (0.08)	0.74 (0.05)	0.63 (0.07)
		0.09 (0.01)	0.10 (0.01)	0.23 (0.04)	0.24 (0.06)	0.30 (0.03)	0.29 (0.03)
		0.09 (0.00)	0.10 (0.01)	0.22 (0.03)	0.23 (0.05)	0.30 (0.02)	0.29 (0.02)
		0.08 (0.00)	0.09 (0.01)	0.22 (0.04)	0.21 (0.05)	0.25 (0.04)	0.26 (0.02)
		0.06 (0.01)	0.04 (0.01)	0.03 (0.02)	0.08 (0.01)	0.04 (0.01)	0.13 (0.03)

Loss Values - Low Setup Times, 1 Machine per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.10 (0.02)	0.24 (0.05)	0.21 (0.06)	0.31 (0.09)	0.62 (0.23)	0.66 (0.22)
		0.10 (0.02)	0.24 (0.05)	0.19 (0.05)	0.29 (0.09)	0.38 (0.07)	0.39 (0.15)
		0.10 (0.02)	0.19 (0.07)	0.18 (0.07)	0.26 (0.07)	0.46 (0.11)	0.42 (0.18)
		0.10 (0.01)	0.18 (0.05)	0.18 (0.07)	0.27 (0.08)	0.39 (0.08)	0.38 (0.20)
		0.10 (0.01)	0.18 (0.05)	0.18 (0.07)	0.27 (0.08)	0.39 (0.08)	0.38 (0.20)
		0.24 (0.08)	0.42 (0.10)	0.27 (0.08)	0.38 (0.11)	0.45 (0.10)	0.46 (0.20)
		0.09 (0.02)	0.19 (0.06)	0.17 (0.06)	0.23 (0.07)	0.40 (0.07)	0.37 (0.14)
		0.09 (0.02)	0.17 (0.06)	0.16 (0.06)	0.22 (0.06)	0.37 (0.07)	0.36 (0.15)
		0.06 (0.01)	0.13 (0.05)	0.12 (0.05)	0.17 (0.07)	0.27 (0.07)	0.25 (0.10)
		0.10 (0.02)	0.04 (0.01)	0.03 (0.01)	0.08 (0.01)	0.06 (0.01)	0.07 (0.04)
	20	0.13 (0.01)	0.25 (0.04)	0.21 (0.04)	0.36 (0.06)	0.81 (0.08)	0.81 (0.13)
		0.13 (0.01)	0.25 (0.04)	0.18 (0.04)	0.33 (0.08)	0.56 (0.11)	0.55 (0.12)
		0.12 (0.01)	0.21 (0.03)	0.18 (0.05)	0.30 (0.07)	0.58 (0.12)	0.61 (0.14)
		0.11 (0.01)	0.21 (0.03)	0.19 (0.04)	0.26 (0.07)	0.53 (0.10)	0.57 (0.12)
		0.11 (0.01)	0.21 (0.03)	0.19 (0.04)	0.26 (0.07)	0.53 (0.10)	0.57 (0.12)
		1.05 (0.08)	1.01 (0.10)	1.06 (0.10)	1.02 (0.15)	0.93 (0.21)	0.91 (0.16)
		0.11 (0.01)	0.21 (0.04)	0.16 (0.04)	0.27 (0.08)	0.52 (0.08)	0.54 (0.12)
		0.11 (0.01)	0.20 (0.05)	0.17 (0.05)	0.24 (0.06)	0.44 (0.08)	0.51 (0.11)
		0.09 (0.01)	0.15 (0.02)	0.14 (0.04)	0.21 (0.05)	0.38 (0.06)	0.38 (0.08)
		0.11 (0.01)	0.05 (0.01)	0.06 (0.02)	0.09 (0.01)	0.02 (0.02)	0.03 (0.01)
	30	0.13 (0.01)	0.24 (0.03)	0.23 (0.07)	0.36 (0.07)	0.73 (0.13)	0.72 (0.11)
		0.13 (0.01)	0.24 (0.03)	0.20 (0.06)	0.32 (0.05)	0.46 (0.08)	0.53 (0.12)
		0.12 (0.01)	0.22 (0.03)	0.20 (0.06)	0.28 (0.07)	0.46 (0.08)	0.50 (0.08)
		0.11 (0.01)	0.19 (0.03)	0.20 (0.06)	0.26 (0.06)	0.41 (0.08)	0.45 (0.09)
		0.11 (0.01)	0.19 (0.03)	0.20 (0.06)	0.26 (0.06)	0.41 (0.08)	0.45 (0.09)
		1.27 (0.06)	1.21 (0.11)	1.33 (0.12)	1.21 (0.14)	1.11 (0.17)	1.00 (0.13)
		0.11 (0.01)	0.19 (0.03)	0.18 (0.06)	0.25 (0.06)	0.40 (0.05)	0.49 (0.13)
		0.11 (0.01)	0.16 (0.02)	0.18 (0.07)	0.24 (0.07)	0.41 (0.07)	0.44 (0.11)
		0.10 (0.01)	0.14 (0.02)	0.16 (0.06)	0.21 (0.05)	0.33 (0.07)	0.34 (0.06)
		0.12 (0.03)	0.05 (0.01)	0.02 (0.01)	0.08 (0.01)	0.02 (0.00)	0.03 (0.01)
	100	0.11 (0.01)	0.18 (0.03)	0.28 (0.06)	0.35 (0.07)	0.58 (0.09)	0.56 (0.07)
		0.11 (0.01)	0.18 (0.03)	0.26 (0.05)	0.32 (0.06)	0.38 (0.04)	0.36 (0.03)
		0.11 (0.00)	0.16 (0.03)	0.27 (0.05)	0.31 (0.04)	0.38 (0.05)	0.35 (0.03)
		0.11 (0.01)	0.15 (0.03)	0.26 (0.06)	0.29 (0.05)	0.33 (0.04)	0.31 (0.03)
		6.06 (0.15)	5.85 (0.24)	6.66 (0.28)	6.47 (0.43)	5.23 (0.66)	5.44 (0.62)
		1.86 (0.04)	1.65 (0.07)	2.01 (0.15)	1.89 (0.16)	1.53 (0.17)	1.36 (0.11)
		0.10 (0.01)	0.15 (0.02)	0.24 (0.05)	0.30 (0.05)	0.35 (0.03)	0.35 (0.04)
		0.10 (0.00)	0.14 (0.02)	0.24 (0.05)	0.27 (0.05)	0.33 (0.03)	0.33 (0.03)
		0.09 (0.00)	0.12 (0.02)	0.23 (0.05)	0.27 (0.05)	0.32 (0.03)	0.30 (0.03)
		0.04 (0.02)	0.05 (0.02)	0.04 (0.02)	0.12 (0.01)	0.02 (0.01)	0.04 (0.01)

Loss Values - Low Setup Times, 2 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.04)	0.26 (0.07)	0.27 (0.19)	0.27 (0.15)	0.48 (0.17)	0.52 (0.22)
		0.12 (0.04)	0.22 (0.07)	0.25 (0.18)	0.25 (0.14)	0.27 (0.19)	0.40 (0.25)
		0.09 (0.02)	0.18 (0.07)	0.22 (0.12)	0.35 (0.14)	0.38 (0.21)	0.38 (0.16)
		0.07 (0.02)	0.17 (0.07)	0.23 (0.16)	0.28 (0.11)	0.33 (0.19)	0.30 (0.14)
		0.07 (0.02)	0.17 (0.07)	0.23 (0.16)	0.28 (0.11)	0.33 (0.19)	0.30 (0.14)
		0.15 (0.09)	0.22 (0.07)	0.24 (0.15)	0.25 (0.14)	0.29 (0.18)	0.29 (0.15)
		0.06 (0.01)	0.16 (0.05)	0.21 (0.11)	0.19 (0.09)	0.17 (0.08)	0.32 (0.12)
		0.06 (0.01)	0.16 (0.05)	0.21 (0.11)	0.19 (0.09)	0.17 (0.08)	0.32 (0.12)
		0.05 (0.02)	0.10 (0.02)	0.14 (0.08)	0.12 (0.06)	0.11 (0.07)	0.17 (0.10)
	20	0.05 (0.05)	0.05 (0.04)	0.10 (0.02)	0.03 (0.01)	0.09 (0.04)	0.19 (0.04)
		0.11 (0.02)	0.22 (0.07)	0.23 (0.10)	0.30 (0.07)	0.42 (0.14)	0.49 (0.18)
		0.11 (0.02)	0.22 (0.06)	0.21 (0.09)	0.27 (0.10)	0.33 (0.10)	0.37 (0.15)
		0.09 (0.02)	0.14 (0.04)	0.20 (0.10)	0.22 (0.07)	0.35 (0.10)	0.35 (0.10)
		0.07 (0.01)	0.10 (0.04)	0.17 (0.09)	0.18 (0.08)	0.31 (0.11)	0.31 (0.11)
		0.07 (0.01)	0.10 (0.04)	0.17 (0.09)	0.18 (0.08)	0.31 (0.11)	0.31 (0.11)
		0.20 (0.03)	0.19 (0.07)	0.28 (0.09)	0.26 (0.06)	0.36 (0.13)	0.29 (0.12)
		0.08 (0.02)	0.10 (0.03)	0.16 (0.07)	0.19 (0.07)	0.28 (0.04)	0.28 (0.08)
		0.08 (0.02)	0.10 (0.03)	0.16 (0.07)	0.19 (0.07)	0.28 (0.04)	0.28 (0.08)
	30	0.06 (0.01)	0.08 (0.02)	0.13 (0.06)	0.16 (0.06)	0.18 (0.05)	0.20 (0.04)
		0.05 (0.03)	0.17 (0.02)	0.07 (0.02)	0.05 (0.01)	0.20 (0.04)	0.06 (0.04)
		0.10 (0.02)	0.18 (0.05)	0.23 (0.08)	0.34 (0.09)	0.43 (0.07)	0.49 (0.21)
		0.11 (0.02)	0.17 (0.05)	0.18 (0.08)	0.29 (0.09)	0.35 (0.07)	0.43 (0.07)
		0.10 (0.00)	0.12 (0.02)	0.19 (0.08)	0.25 (0.07)	0.36 (0.08)	0.36 (0.07)
		0.07 (0.02)	0.08 (0.01)	0.14 (0.06)	0.18 (0.08)	0.29 (0.06)	0.30 (0.06)
		0.07 (0.02)	0.08 (0.01)	0.14 (0.06)	0.18 (0.08)	0.29 (0.06)	0.30 (0.06)
		0.20 (0.03)	0.20 (0.03)	0.28 (0.11)	0.32 (0.10)	0.30 (0.06)	0.30 (0.08)
		0.09 (0.01)	0.10 (0.04)	0.16 (0.04)	0.19 (0.08)	0.29 (0.02)	0.30 (0.04)
	100	0.09 (0.01)	0.10 (0.04)	0.16 (0.04)	0.19 (0.08)	0.29 (0.02)	0.30 (0.04)
		0.07 (0.00)	0.08 (0.01)	0.13 (0.06)	0.18 (0.08)	0.20 (0.03)	0.19 (0.05)
		0.08 (0.05)	0.18 (0.02)	0.06 (0.01)	0.07 (0.04)	0.20 (0.06)	0.08 (0.05)
		0.10 (0.01)	0.14 (0.03)	0.29 (0.04)	0.32 (0.08)	0.39 (0.04)	0.42 (0.07)
		0.10 (0.01)	0.13 (0.02)	0.28 (0.04)	0.29 (0.08)	0.31 (0.02)	0.36 (0.04)
		0.09 (0.01)	0.11 (0.03)	0.27 (0.04)	0.26 (0.06)	0.30 (0.02)	0.32 (0.02)
		0.35 (0.01)	0.28 (0.03)	0.53 (0.06)	0.43 (0.06)	0.24 (0.02)	0.25 (0.02)
		1.51 (0.10)	1.15 (0.16)	1.75 (0.33)	1.25 (0.27)	0.25 (0.03)	0.25 (0.02)
		0.24 (0.02)	0.22 (0.02)	0.42 (0.06)	0.39 (0.07)	0.30 (0.03)	0.33 (0.07)
		0.09 (0.00)	0.10 (0.01)	0.25 (0.04)	0.24 (0.06)	0.29 (0.01)	0.31 (0.06)
		0.09 (0.00)	0.10 (0.01)	0.25 (0.04)	0.24 (0.06)	0.29 (0.01)	0.31 (0.06)
		0.08 (0.00)	0.08 (0.01)	0.24 (0.04)	0.22 (0.06)	0.24 (0.03)	0.23 (0.05)
		0.07 (0.02)	0.05 (0.05)	0.15 (0.04)	0.06 (0.04)	0.22 (0.08)	0.08 (0.03)

Loss Values - Low Setup Times, 2 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.03)	0.33 (0.10)	0.27 (0.14)	0.31 (0.11)	0.36 (0.21)	0.49 (0.25)
		0.10 (0.04)	0.30 (0.07)	0.25 (0.13)	0.28 (0.08)	0.32 (0.17)	0.41 (0.20)
		0.09 (0.03)	0.27 (0.10)	0.25 (0.13)	0.31 (0.11)	0.37 (0.15)	0.38 (0.12)
		0.09 (0.02)	0.26 (0.08)	0.23 (0.13)	0.29 (0.11)	0.43 (0.15)	0.30 (0.15)
		0.09 (0.02)	0.26 (0.08)	0.23 (0.13)	0.29 (0.11)	0.43 (0.15)	0.30 (0.15)
		0.22 (0.07)	0.32 (0.09)	0.32 (0.10)	0.34 (0.11)	0.35 (0.13)	0.30 (0.16)
		0.09 (0.03)	0.23 (0.08)	0.22 (0.14)	0.27 (0.07)	0.33 (0.17)	0.30 (0.11)
		0.09 (0.03)	0.23 (0.10)	0.21 (0.13)	0.24 (0.07)	0.32 (0.15)	0.25 (0.09)
		0.06 (0.02)	0.17 (0.07)	0.17 (0.10)	0.18 (0.07)	0.14 (0.08)	0.14 (0.09)
		0.08 (0.04)	0.03 (0.01)	0.14 (0.04)	0.03 (0.01)	0.29 (0.05)	0.17 (0.04)
	20	0.13 (0.02)	0.30 (0.06)	0.27 (0.06)	0.40 (0.13)	0.85 (0.16)	0.86 (0.26)
		0.14 (0.02)	0.29 (0.06)	0.24 (0.07)	0.34 (0.07)	0.50 (0.12)	0.59 (0.19)
		0.11 (0.01)	0.20 (0.05)	0.23 (0.10)	0.29 (0.10)	0.54 (0.15)	0.49 (0.16)
		0.11 (0.03)	0.17 (0.03)	0.22 (0.08)	0.26 (0.08)	0.51 (0.10)	0.43 (0.14)
		0.11 (0.03)	0.17 (0.03)	0.22 (0.08)	0.26 (0.08)	0.51 (0.10)	0.43 (0.14)
		0.42 (0.06)	0.45 (0.08)	0.53 (0.08)	0.50 (0.10)	0.58 (0.13)	0.57 (0.14)
		0.11 (0.02)	0.19 (0.05)	0.18 (0.04)	0.24 (0.06)	0.40 (0.12)	0.42 (0.14)
		0.11 (0.01)	0.17 (0.04)	0.18 (0.05)	0.23 (0.07)	0.39 (0.09)	0.40 (0.15)
		0.09 (0.01)	0.14 (0.02)	0.16 (0.05)	0.20 (0.08)	0.32 (0.08)	0.30 (0.08)
		0.04 (0.02)	0.03 (0.02)	0.12 (0.02)	0.05 (0.02)	0.10 (0.11)	0.15 (0.10)
	30	0.13 (0.01)	0.26 (0.05)	0.21 (0.05)	0.40 (0.11)	0.68 (0.14)	0.73 (0.21)
		0.13 (0.01)	0.25 (0.03)	0.20 (0.03)	0.32 (0.07)	0.49 (0.08)	0.46 (0.11)
		0.11 (0.01)	0.19 (0.05)	0.18 (0.04)	0.27 (0.08)	0.50 (0.08)	0.48 (0.09)
		0.10 (0.02)	0.15 (0.03)	0.19 (0.03)	0.22 (0.07)	0.47 (0.07)	0.42 (0.10)
		0.10 (0.02)	0.15 (0.03)	0.19 (0.03)	0.22 (0.07)	0.47 (0.07)	0.42 (0.10)
		0.55 (0.06)	0.49 (0.06)	0.57 (0.08)	0.57 (0.10)	0.68 (0.14)	0.54 (0.14)
		0.10 (0.01)	0.19 (0.03)	0.16 (0.03)	0.22 (0.05)	0.40 (0.07)	0.44 (0.07)
		0.10 (0.01)	0.17 (0.03)	0.16 (0.03)	0.23 (0.05)	0.39 (0.05)	0.37 (0.08)
		0.09 (0.01)	0.15 (0.02)	0.15 (0.03)	0.20 (0.06)	0.32 (0.05)	0.29 (0.04)
		0.04 (0.01)	0.06 (0.00)	0.09 (0.01)	0.02 (0.01)	0.11 (0.07)	0.11 (0.04)
	100	0.12 (0.01)	0.19 (0.02)	0.31 (0.05)	0.38 (0.05)	0.48 (0.06)	0.52 (0.06)
		0.11 (0.01)	0.17 (0.02)	0.28 (0.06)	0.32 (0.05)	0.35 (0.04)	0.39 (0.04)
		0.11 (0.01)	0.15 (0.02)	0.28 (0.05)	0.31 (0.05)	0.36 (0.05)	0.35 (0.04)
		0.34 (0.02)	0.27 (0.04)	0.51 (0.06)	0.36 (0.09)	0.30 (0.03)	0.30 (0.05)
		4.66 (0.23)	4.16 (0.39)	5.00 (0.29)	4.50 (0.49)	0.37 (0.22)	0.30 (0.05)
		0.66 (0.06)	0.61 (0.05)	0.85 (0.07)	0.79 (0.08)	0.68 (0.09)	0.60 (0.08)
		0.10 (0.00)	0.13 (0.02)	0.25 (0.05)	0.28 (0.04)	0.33 (0.03)	0.32 (0.01)
		0.09 (0.00)	0.11 (0.01)	0.25 (0.05)	0.29 (0.05)	0.32 (0.03)	0.34 (0.02)
		0.09 (0.00)	0.11 (0.01)	0.25 (0.04)	0.27 (0.05)	0.28 (0.02)	0.26 (0.04)
		0.05 (0.01)	0.06 (0.04)	0.02 (0.02)	0.05 (0.06)	0.13 (0.06)	0.10 (0.02)

Loss Values - Low Setup Times, 2 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.01)	0.29 (0.06)	0.22 (0.05)	0.32 (0.05)	0.37 (0.13)	0.37 (0.16)
		0.11 (0.02)	0.26 (0.04)	0.20 (0.04)	0.29 (0.07)	0.19 (0.07)	0.25 (0.09)
		0.11 (0.01)	0.24 (0.06)	0.23 (0.07)	0.29 (0.05)	0.30 (0.07)	0.23 (0.08)
		0.09 (0.01)	0.24 (0.05)	0.21 (0.03)	0.26 (0.05)	0.21 (0.07)	0.21 (0.08)
		0.09 (0.01)	0.24 (0.05)	0.21 (0.03)	0.26 (0.05)	0.21 (0.07)	0.21 (0.08)
		0.24 (0.03)	0.29 (0.06)	0.29 (0.05)	0.36 (0.06)	0.17 (0.08)	0.24 (0.11)
		0.10 (0.03)	0.24 (0.06)	0.21 (0.06)	0.28 (0.06)	0.24 (0.11)	0.25 (0.09)
		0.10 (0.03)	0.23 (0.04)	0.22 (0.06)	0.28 (0.05)	0.21 (0.08)	0.24 (0.11)
		0.08 (0.02)	0.17 (0.04)	0.16 (0.03)	0.19 (0.03)	0.11 (0.04)	0.11 (0.05)
		0.15 (0.07)	0.06 (0.04)	0.10 (0.03)	0.07 (0.05)	0.20 (0.02)	0.20 (0.04)
	20	0.16 (0.01)	0.37 (0.06)	0.30 (0.05)	0.55 (0.12)	0.82 (0.11)	1.00 (0.12)
		0.16 (0.01)	0.33 (0.06)	0.29 (0.04)	0.47 (0.09)	0.60 (0.09)	0.74 (0.15)
		0.14 (0.01)	0.27 (0.04)	0.30 (0.06)	0.43 (0.12)	0.63 (0.11)	0.70 (0.16)
		0.14 (0.02)	0.26 (0.04)	0.30 (0.05)	0.40 (0.11)	0.62 (0.10)	0.64 (0.18)
		0.14 (0.02)	0.26 (0.04)	0.30 (0.05)	0.40 (0.11)	0.62 (0.10)	0.64 (0.18)
		0.75 (0.10)	0.81 (0.08)	0.85 (0.09)	0.96 (0.14)	0.95 (0.12)	0.98 (0.19)
		0.12 (0.01)	0.25 (0.05)	0.26 (0.05)	0.38 (0.09)	0.58 (0.11)	0.67 (0.15)
		0.12 (0.01)	0.26 (0.05)	0.25 (0.05)	0.38 (0.07)	0.52 (0.07)	0.65 (0.19)
		0.11 (0.01)	0.22 (0.04)	0.22 (0.04)	0.33 (0.08)	0.42 (0.05)	0.48 (0.08)
		0.08 (0.01)	0.04 (0.01)	0.22 (0.02)	0.16 (0.06)	0.05 (0.06)	0.16 (0.03)
	30	0.14 (0.01)	0.33 (0.03)	0.29 (0.06)	0.50 (0.11)	0.89 (0.13)	1.09 (0.09)
		0.14 (0.01)	0.31 (0.03)	0.25 (0.05)	0.42 (0.07)	0.61 (0.13)	0.72 (0.11)
		0.14 (0.01)	0.26 (0.04)	0.26 (0.05)	0.36 (0.07)	0.57 (0.13)	0.71 (0.12)
		0.15 (0.02)	0.24 (0.03)	0.27 (0.06)	0.34 (0.06)	0.55 (0.09)	0.63 (0.12)
		0.15 (0.02)	0.24 (0.03)	0.27 (0.06)	0.34 (0.06)	0.55 (0.09)	0.63 (0.12)
		0.91 (0.08)	0.92 (0.09)	1.04 (0.06)	1.08 (0.12)	0.95 (0.19)	1.05 (0.13)
		0.13 (0.01)	0.25 (0.04)	0.23 (0.05)	0.32 (0.05)	0.55 (0.15)	0.60 (0.09)
		0.13 (0.01)	0.24 (0.03)	0.22 (0.05)	0.33 (0.05)	0.47 (0.13)	0.57 (0.08)
		0.11 (0.01)	0.21 (0.02)	0.21 (0.04)	0.29 (0.05)	0.42 (0.11)	0.50 (0.08)
		0.07 (0.02)	0.06 (0.02)	0.08 (0.02)	0.11 (0.08)	0.12 (0.08)	0.21 (0.05)
	100	0.13 (0.01)	0.25 (0.03)	0.30 (0.05)	0.45 (0.07)	0.64 (0.06)	0.74 (0.05)
		0.13 (0.01)	0.23 (0.02)	0.27 (0.05)	0.39 (0.05)	0.42 (0.03)	0.47 (0.04)
		0.12 (0.01)	0.21 (0.03)	0.27 (0.04)	0.38 (0.05)	0.43 (0.04)	0.47 (0.05)
		0.33 (0.02)	0.27 (0.06)	0.45 (0.05)	0.42 (0.09)	0.37 (0.03)	0.39 (0.05)
		10.68 (0.32)	10.48 (0.41)	11.62 (0.62)	11.49 (0.68)	1.71 (1.43)	1.37 (0.82)
		1.38 (0.04)	1.31 (0.06)	1.54 (0.12)	1.59 (0.10)	1.29 (0.12)	1.16 (0.12)
		0.12 (0.01)	0.19 (0.03)	0.25 (0.05)	0.34 (0.03)	0.40 (0.04)	0.46 (0.05)
		0.11 (0.01)	0.17 (0.03)	0.23 (0.05)	0.32 (0.04)	0.35 (0.03)	0.37 (0.05)
		0.11 (0.00)	0.17 (0.02)	0.24 (0.04)	0.32 (0.04)	0.36 (0.02)	0.36 (0.03)
		0.06 (0.01)	0.13 (0.05)	0.03 (0.02)	0.08 (0.05)	0.14 (0.08)	0.33 (0.10)

Loss Values - Low Setup Times, 10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.16 (0.03)	0.47 (0.06)	0.38 (0.12)	0.46 (0.05)	0.49 (0.09)	0.33 (0.24)
		0.17 (0.03)	0.45 (0.07)	0.36 (0.13)	0.45 (0.04)	0.34 (0.22)	0.21 (0.14)
		0.14 (0.02)	0.36 (0.05)	0.39 (0.16)	0.38 (0.03)	0.27 (0.20)	0.23 (0.14)
		0.10 (0.02)	0.30 (0.05)	0.35 (0.14)	0.30 (0.05)	0.26 (0.19)	0.14 (0.14)
		0.10 (0.02)	0.30 (0.05)	0.35 (0.14)	0.30 (0.05)	0.26 (0.19)	0.14 (0.14)
		0.27 (0.09)	0.28 (0.05)	0.46 (0.17)	0.26 (0.05)	0.22 (0.18)	0.04 (0.05)
		0.12 (0.02)	0.23 (0.05)	0.37 (0.13)	0.24 (0.07)	0.13 (0.09)	0.07 (0.06)
		0.12 (0.02)	0.23 (0.05)	0.37 (0.13)	0.24 (0.07)	0.13 (0.09)	0.07 (0.06)
		0.24 (0.08)	0.30 (0.06)	0.39 (0.10)	0.29 (0.05)	0.11 (0.10)	0.02 (0.03)
		0.07 (0.06)	0.10 (0.03)	0.36 (0.05)	0.04 (0.03)	0.34 (0.10)	0.23 (0.04)
	30	0.17 (0.02)	0.45 (0.08)	0.45 (0.20)	0.80 (0.17)	0.72 (0.25)	0.58 (0.20)
		0.17 (0.02)	0.42 (0.09)	0.44 (0.20)	0.79 (0.18)	0.61 (0.20)	0.54 (0.18)
		0.15 (0.02)	0.35 (0.05)	0.45 (0.23)	0.64 (0.11)	0.62 (0.16)	0.46 (0.16)
		0.09 (0.01)	0.29 (0.05)	0.38 (0.21)	0.51 (0.08)	0.55 (0.15)	0.31 (0.13)
		0.09 (0.01)	0.29 (0.05)	0.38 (0.21)	0.51 (0.08)	0.55 (0.15)	0.31 (0.13)
		0.28 (0.01)	0.31 (0.05)	0.49 (0.17)	0.51 (0.08)	0.46 (0.14)	0.26 (0.14)
		0.12 (0.01)	0.24 (0.04)	0.40 (0.20)	0.46 (0.10)	0.37 (0.20)	0.21 (0.15)
		0.12 (0.01)	0.24 (0.04)	0.40 (0.20)	0.46 (0.10)	0.37 (0.20)	0.21 (0.15)
		0.25 (0.04)	0.34 (0.04)	0.43 (0.18)	0.57 (0.08)	0.43 (0.11)	0.21 (0.12)
		0.12 (0.07)	0.11 (0.06)	0.07 (0.04)	0.08 (0.07)	0.26 (0.09)	0.10 (0.06)
	100	0.15 (0.01)	0.32 (0.04)	0.45 (0.05)	0.58 (0.11)	0.54 (0.05)	0.69 (0.11)
		0.15 (0.01)	0.28 (0.02)	0.41 (0.06)	0.49 (0.12)	0.47 (0.07)	0.59 (0.09)
		0.12 (0.01)	0.21 (0.03)	0.41 (0.07)	0.42 (0.09)	0.51 (0.07)	0.57 (0.09)
		0.11 (0.02)	0.14 (0.03)	0.33 (0.05)	0.30 (0.09)	0.36 (0.07)	0.39 (0.06)
		0.11 (0.02)	0.14 (0.03)	0.33 (0.05)	0.30 (0.09)	0.36 (0.07)	0.39 (0.06)
		0.21 (0.02)	0.22 (0.02)	0.43 (0.06)	0.40 (0.09)	0.43 (0.06)	0.38 (0.08)
		0.10 (0.02)	0.17 (0.03)	0.31 (0.06)	0.32 (0.06)	0.36 (0.03)	0.48 (0.05)
		0.10 (0.02)	0.17 (0.03)	0.31 (0.06)	0.32 (0.06)	0.36 (0.03)	0.48 (0.05)
		0.16 (0.01)	0.22 (0.02)	0.39 (0.04)	0.41 (0.09)	0.37 (0.04)	0.41 (0.04)
		0.09 (0.06)	0.17 (0.04)	0.09 (0.07)	0.10 (0.06)	0.07 (0.05)	0.07 (0.03)

Loss Values - Low Setup Times, 10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.18 (0.01)	0.28 (0.05)	0.29 (0.05)	0.25 (0.09)	0.28 (0.12)	0.27 (0.12)
		0.17 (0.02)	0.26 (0.05)	0.27 (0.05)	0.22 (0.08)	0.26 (0.14)	0.20 (0.13)
		0.17 (0.01)	0.21 (0.03)	0.32 (0.04)	0.23 (0.06)	0.29 (0.13)	0.17 (0.05)
		0.12 (0.02)	0.18 (0.04)	0.28 (0.04)	0.18 (0.05)	0.19 (0.12)	0.12 (0.07)
		0.12 (0.02)	0.18 (0.04)	0.28 (0.04)	0.18 (0.05)	0.19 (0.12)	0.12 (0.07)
		0.27 (0.02)	0.21 (0.06)	0.35 (0.06)	0.21 (0.07)	0.14 (0.10)	0.14 (0.10)
		0.16 (0.02)	0.15 (0.05)	0.32 (0.07)	0.16 (0.06)	0.16 (0.09)	0.16 (0.09)
		0.15 (0.02)	0.16 (0.05)	0.32 (0.04)	0.18 (0.11)	0.14 (0.06)	0.14 (0.08)
		0.23 (0.04)	0.14 (0.06)	0.28 (0.03)	0.11 (0.06)	0.07 (0.03)	0.03 (0.02)
		0.19 (0.04)	0.14 (0.03)	0.06 (0.04)	0.17 (0.05)	0.09 (0.07)	0.07 (0.05)
	30	0.19 (0.02)	0.49 (0.07)	0.45 (0.10)	0.53 (0.08)	0.51 (0.12)	0.47 (0.16)
		0.18 (0.02)	0.48 (0.05)	0.43 (0.09)	0.45 (0.05)	0.43 (0.11)	0.34 (0.13)
		0.17 (0.02)	0.43 (0.03)	0.46 (0.10)	0.41 (0.07)	0.39 (0.09)	0.36 (0.08)
		0.12 (0.01)	0.34 (0.03)	0.40 (0.07)	0.34 (0.04)	0.34 (0.09)	0.25 (0.09)
		0.12 (0.01)	0.34 (0.03)	0.40 (0.07)	0.34 (0.04)	0.34 (0.09)	0.25 (0.09)
		0.35 (0.04)	0.45 (0.06)	0.58 (0.10)	0.40 (0.08)	0.31 (0.08)	0.25 (0.17)
		0.15 (0.02)	0.35 (0.05)	0.44 (0.10)	0.31 (0.08)	0.30 (0.11)	0.34 (0.15)
		0.17 (0.02)	0.33 (0.05)	0.45 (0.08)	0.32 (0.07)	0.25 (0.10)	0.24 (0.10)
		0.22 (0.03)	0.35 (0.04)	0.42 (0.09)	0.32 (0.05)	0.20 (0.06)	0.15 (0.07)
		0.07 (0.05)	0.17 (0.04)	0.21 (0.04)	0.10 (0.06)	0.15 (0.07)	0.16 (0.05)
	100	0.17 (0.01)	0.44 (0.04)	0.47 (0.08)	0.74 (0.05)	0.90 (0.07)	1.12 (0.14)
		0.17 (0.01)	0.39 (0.03)	0.42 (0.06)	0.65 (0.05)	0.73 (0.13)	0.80 (0.10)
		0.15 (0.01)	0.30 (0.03)	0.42 (0.07)	0.58 (0.07)	0.70 (0.12)	0.81 (0.13)
		0.17 (0.01)	0.23 (0.03)	0.37 (0.05)	0.44 (0.06)	0.63 (0.11)	0.65 (0.13)
		0.17 (0.01)	0.23 (0.03)	0.37 (0.05)	0.44 (0.06)	0.63 (0.11)	0.65 (0.13)
		0.45 (0.02)	0.51 (0.04)	0.73 (0.07)	0.79 (0.08)	0.85 (0.12)	0.81 (0.07)
		0.13 (0.01)	0.26 (0.03)	0.36 (0.06)	0.51 (0.07)	0.60 (0.13)	0.65 (0.14)
		0.13 (0.01)	0.26 (0.04)	0.35 (0.07)	0.51 (0.05)	0.50 (0.07)	0.56 (0.07)
		0.17 (0.01)	0.30 (0.02)	0.42 (0.07)	0.56 (0.05)	0.59 (0.06)	0.61 (0.08)
		0.12 (0.10)	0.14 (0.03)	0.05 (0.04)	0.10 (0.05)	0.19 (0.05)	0.16 (0.06)

Loss Values - Low Setup Times, 10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.16 (0.02)	0.19 (0.04)	0.19 (0.04)	0.21 (0.06)	0.23 (0.14)	0.14 (0.13)
		0.16 (0.02)	0.16 (0.03)	0.19 (0.02)	0.19 (0.07)	0.21 (0.10)	0.12 (0.05)
		0.16 (0.02)	0.15 (0.03)	0.27 (0.03)	0.23 (0.03)	0.21 (0.08)	0.12 (0.06)
		0.12 (0.02)	0.11 (0.03)	0.19 (0.03)	0.14 (0.03)	0.11 (0.05)	0.07 (0.05)
		0.12 (0.02)	0.11 (0.03)	0.19 (0.03)	0.14 (0.03)	0.11 (0.05)	0.07 (0.05)
		0.29 (0.02)	0.22 (0.06)	0.26 (0.07)	0.25 (0.06)	0.23 (0.10)	0.21 (0.07)
		0.15 (0.01)	0.14 (0.03)	0.24 (0.04)	0.20 (0.07)	0.19 (0.07)	0.11 (0.05)
		0.17 (0.02)	0.14 (0.04)	0.22 (0.05)	0.18 (0.07)	0.16 (0.07)	0.09 (0.04)
		0.17 (0.02)	0.08 (0.04)	0.18 (0.03)	0.11 (0.05)	0.06 (0.03)	0.03 (0.01)
		0.13 (0.06)	0.19 (0.05)	0.08 (0.06)	0.13 (0.07)	0.05 (0.05)	0.12 (0.05)
	30	0.19 (0.03)	0.32 (0.05)	0.34 (0.04)	0.36 (0.06)	0.36 (0.14)	0.40 (0.17)
		0.18 (0.03)	0.30 (0.03)	0.33 (0.06)	0.31 (0.11)	0.32 (0.11)	0.28 (0.10)
		0.17 (0.02)	0.26 (0.02)	0.38 (0.04)	0.31 (0.04)	0.33 (0.06)	0.31 (0.05)
		0.12 (0.02)	0.21 (0.04)	0.32 (0.03)	0.25 (0.05)	0.24 (0.08)	0.22 (0.07)
		0.12 (0.02)	0.21 (0.04)	0.32 (0.03)	0.25 (0.05)	0.24 (0.08)	0.22 (0.07)
		0.38 (0.02)	0.36 (0.05)	0.48 (0.06)	0.37 (0.05)	0.28 (0.08)	0.32 (0.14)
		0.16 (0.02)	0.23 (0.05)	0.33 (0.05)	0.29 (0.06)	0.27 (0.12)	0.23 (0.06)
		0.17 (0.02)	0.24 (0.04)	0.31 (0.04)	0.26 (0.05)	0.24 (0.06)	0.23 (0.05)
		0.18 (0.02)	0.20 (0.04)	0.31 (0.04)	0.22 (0.05)	0.16 (0.05)	0.12 (0.05)
		0.22 (0.09)	0.10 (0.03)	0.11 (0.07)	0.19 (0.06)	0.14 (0.06)	0.11 (0.04)
	100	0.19 (0.01)	0.50 (0.05)	0.54 (0.07)	0.85 (0.10)	1.22 (0.12)	1.12 (0.21)
		0.18 (0.01)	0.43 (0.06)	0.48 (0.07)	0.67 (0.07)	0.87 (0.07)	0.77 (0.16)
		0.17 (0.01)	0.38 (0.04)	0.50 (0.08)	0.64 (0.08)	0.92 (0.07)	0.75 (0.14)
		0.18 (0.01)	0.30 (0.05)	0.47 (0.07)	0.53 (0.07)	0.81 (0.09)	0.68 (0.11)
		1.14 (0.21)	0.78 (0.24)	0.55 (0.14)	0.59 (0.14)	0.81 (0.09)	0.68 (0.11)
		0.69 (0.04)	0.82 (0.06)	1.07 (0.10)	1.16 (0.10)	1.25 (0.13)	0.97 (0.20)
		0.16 (0.01)	0.35 (0.03)	0.45 (0.06)	0.59 (0.07)	0.80 (0.08)	0.64 (0.12)
		0.16 (0.01)	0.35 (0.04)	0.46 (0.07)	0.56 (0.05)	0.65 (0.10)	0.59 (0.11)
		0.17 (0.00)	0.36 (0.03)	0.48 (0.06)	0.60 (0.07)	0.71 (0.06)	0.56 (0.15)
		0.12 (0.05)	0.20 (0.02)	0.19 (0.04)	0.18 (0.03)	0.16 (0.07)	0.21 (0.04)

Loss Values - Low Setup Times, 1-4 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.10 (0.04)	0.24 (0.12)	0.23 (0.15)	0.23 (0.14)	0.33 (0.19)	0.28 (0.21)
		0.10 (0.03)	0.20 (0.11)	0.19 (0.14)	0.16 (0.10)	0.30 (0.19)	0.25 (0.20)
		0.09 (0.03)	0.22 (0.13)	0.18 (0.14)	0.18 (0.11)	0.23 (0.13)	0.24 (0.22)
		0.07 (0.03)	0.20 (0.10)	0.17 (0.15)	0.18 (0.13)	0.25 (0.18)	0.21 (0.18)
		0.07 (0.03)	0.20 (0.10)	0.17 (0.15)	0.18 (0.13)	0.25 (0.18)	0.21 (0.18)
		0.11 (0.08)	0.14 (0.06)	0.13 (0.11)	0.15 (0.12)	0.21 (0.17)	0.19 (0.20)
		0.09 (0.02)	0.16 (0.09)	0.15 (0.13)	0.17 (0.12)	0.16 (0.09)	0.15 (0.11)
		0.09 (0.02)	0.16 (0.09)	0.15 (0.13)	0.17 (0.12)	0.16 (0.09)	0.15 (0.11)
		0.05 (0.02)	0.08 (0.04)	0.10 (0.11)	0.07 (0.06)	0.09 (0.09)	0.07 (0.07)
		0.04 (0.05)	0.14 (0.04)	0.11 (0.07)	0.16 (0.05)	0.16 (0.08)	0.06 (0.04)
	20	0.09 (0.02)	0.15 (0.07)	0.26 (0.14)	0.28 (0.18)	0.34 (0.08)	0.46 (0.20)
		0.09 (0.02)	0.15 (0.06)	0.23 (0.13)	0.25 (0.11)	0.27 (0.05)	0.36 (0.16)
		0.08 (0.02)	0.12 (0.05)	0.25 (0.12)	0.23 (0.08)	0.28 (0.03)	0.35 (0.14)
		0.04 (0.02)	0.07 (0.05)	0.19 (0.10)	0.17 (0.08)	0.23 (0.05)	0.30 (0.13)
		0.04 (0.02)	0.07 (0.05)	0.19 (0.10)	0.17 (0.08)	0.23 (0.05)	0.30 (0.13)
		0.04 (0.03)	0.09 (0.06)	0.20 (0.12)	0.17 (0.13)	0.22 (0.04)	0.25 (0.09)
		0.08 (0.02)	0.10 (0.03)	0.21 (0.10)	0.16 (0.08)	0.27 (0.04)	0.27 (0.06)
		0.08 (0.02)	0.10 (0.03)	0.21 (0.10)	0.16 (0.08)	0.27 (0.04)	0.27 (0.06)
		0.04 (0.01)	0.06 (0.02)	0.15 (0.10)	0.13 (0.08)	0.16 (0.06)	0.20 (0.07)
		0.15 (0.04)	0.12 (0.04)	0.20 (0.06)	0.22 (0.05)	0.11 (0.07)	0.18 (0.06)
	30	0.09 (0.01)	0.11 (0.04)	0.25 (0.07)	0.37 (0.14)	0.38 (0.13)	0.44 (0.17)
		0.09 (0.01)	0.10 (0.05)	0.21 (0.06)	0.31 (0.12)	0.32 (0.11)	0.37 (0.13)
		0.09 (0.01)	0.09 (0.03)	0.21 (0.06)	0.29 (0.10)	0.31 (0.09)	0.37 (0.10)
		0.03 (0.01)	0.06 (0.03)	0.16 (0.06)	0.22 (0.10)	0.25 (0.08)	0.31 (0.11)
		0.03 (0.01)	0.06 (0.03)	0.16 (0.06)	0.22 (0.10)	0.25 (0.08)	0.31 (0.11)
		0.04 (0.03)	0.04 (0.02)	0.18 (0.09)	0.22 (0.10)	0.26 (0.09)	0.28 (0.10)
		0.09 (0.01)	0.08 (0.01)	0.20 (0.06)	0.24 (0.10)	0.27 (0.02)	0.27 (0.05)
		0.09 (0.01)	0.08 (0.01)	0.20 (0.06)	0.24 (0.10)	0.27 (0.02)	0.27 (0.05)
		0.05 (0.01)	0.05 (0.01)	0.16 (0.05)	0.19 (0.08)	0.17 (0.08)	0.16 (0.07)
		0.36 (0.08)	0.18 (0.05)	0.06 (0.03)	0.07 (0.07)	0.15 (0.06)	0.15 (0.07)
	100	0.09 (0.01)	0.12 (0.05)	0.28 (0.03)	0.29 (0.07)	0.32 (0.03)	0.35 (0.05)
		0.09 (0.01)	0.11 (0.04)	0.28 (0.03)	0.29 (0.08)	0.30 (0.03)	0.32 (0.04)
		0.09 (0.01)	0.10 (0.02)	0.27 (0.04)	0.27 (0.06)	0.30 (0.02)	0.31 (0.03)
		0.12 (0.12)	0.10 (0.08)	0.29 (0.13)	0.36 (0.16)	0.28 (0.13)	0.24 (0.02)
		0.57 (0.45)	0.73 (0.10)	0.95 (0.37)	0.78 (0.33)	0.34 (0.15)	0.37 (0.22)
		0.03 (0.03)	0.05 (0.03)	0.23 (0.04)	0.22 (0.08)	0.22 (0.02)	0.22 (0.02)
		0.08 (0.01)	0.09 (0.01)	0.26 (0.03)	0.25 (0.05)	0.29 (0.01)	0.31 (0.03)
		0.08 (0.01)	0.09 (0.01)	0.26 (0.03)	0.25 (0.05)	0.29 (0.01)	0.31 (0.03)
		0.06 (0.00)	0.07 (0.01)	0.23 (0.03)	0.23 (0.05)	0.22 (0.06)	0.20 (0.07)
		0.08 (0.06)	0.17 (0.02)	0.12 (0.08)	0.28 (0.16)	0.18 (0.04)	0.17 (0.09)

Loss Values - Low Setup Times, 1-4 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.08 (0.02)	0.19 (0.08)	0.17 (0.10)	0.25 (0.10)	0.28 (0.15)	0.23 (0.17)
		0.07 (0.02)	0.16 (0.08)	0.14 (0.10)	0.16 (0.08)	0.14 (0.09)	0.14 (0.10)
		0.07 (0.02)	0.15 (0.06)	0.11 (0.08)	0.14 (0.08)	0.18 (0.11)	0.18 (0.14)
		0.06 (0.02)	0.14 (0.06)	0.11 (0.08)	0.14 (0.09)	0.12 (0.12)	0.17 (0.13)
		0.06 (0.02)	0.14 (0.06)	0.11 (0.08)	0.14 (0.09)	0.12 (0.12)	0.17 (0.13)
		0.11 (0.08)	0.14 (0.07)	0.14 (0.10)	0.18 (0.08)	0.13 (0.11)	0.20 (0.13)
		0.07 (0.03)	0.14 (0.06)	0.12 (0.07)	0.16 (0.07)	0.15 (0.11)	0.12 (0.10)
		0.06 (0.02)	0.11 (0.05)	0.11 (0.06)	0.16 (0.09)	0.12 (0.08)	0.13 (0.09)
		0.03 (0.02)	0.06 (0.05)	0.08 (0.07)	0.09 (0.06)	0.06 (0.04)	0.10 (0.09)
		0.12 (0.06)	0.20 (0.06)	0.12 (0.06)	0.21 (0.04)	0.20 (0.05)	0.15 (0.11)
	20	0.11 (0.04)	0.20 (0.06)	0.29 (0.11)	0.37 (0.18)	0.59 (0.26)	0.68 (0.27)
		0.10 (0.02)	0.16 (0.05)	0.26 (0.11)	0.29 (0.12)	0.49 (0.24)	0.49 (0.25)
		0.09 (0.02)	0.14 (0.03)	0.26 (0.11)	0.30 (0.13)	0.52 (0.22)	0.42 (0.10)
		0.07 (0.02)	0.10 (0.03)	0.23 (0.11)	0.22 (0.08)	0.47 (0.23)	0.39 (0.16)
		0.07 (0.02)	0.10 (0.03)	0.23 (0.11)	0.22 (0.08)	0.47 (0.23)	0.39 (0.16)
		0.21 (0.11)	0.22 (0.10)	0.39 (0.19)	0.30 (0.13)	0.48 (0.21)	0.41 (0.19)
		0.08 (0.02)	0.12 (0.03)	0.22 (0.10)	0.23 (0.08)	0.39 (0.12)	0.39 (0.10)
		0.09 (0.02)	0.10 (0.03)	0.21 (0.09)	0.23 (0.08)	0.38 (0.10)	0.35 (0.12)
		0.06 (0.02)	0.06 (0.02)	0.17 (0.09)	0.17 (0.08)	0.26 (0.12)	0.21 (0.11)
		0.14 (0.07)	0.17 (0.04)	0.21 (0.04)	0.08 (0.09)	0.13 (0.07)	0.18 (0.09)
	30	0.09 (0.01)	0.15 (0.04)	0.25 (0.14)	0.30 (0.11)	0.49 (0.14)	0.43 (0.08)
		0.09 (0.02)	0.15 (0.04)	0.20 (0.11)	0.27 (0.09)	0.37 (0.10)	0.34 (0.05)
		0.09 (0.01)	0.16 (0.07)	0.22 (0.14)	0.27 (0.07)	0.39 (0.11)	0.33 (0.05)
		0.06 (0.03)	0.09 (0.04)	0.17 (0.11)	0.21 (0.08)	0.33 (0.12)	0.26 (0.04)
		0.06 (0.03)	0.09 (0.04)	0.17 (0.11)	0.21 (0.08)	0.33 (0.12)	0.26 (0.04)
		0.23 (0.16)	0.21 (0.13)	0.37 (0.17)	0.37 (0.15)	0.38 (0.14)	0.36 (0.11)
		0.09 (0.01)	0.11 (0.04)	0.19 (0.12)	0.23 (0.10)	0.34 (0.09)	0.30 (0.05)
		0.08 (0.01)	0.10 (0.03)	0.19 (0.10)	0.22 (0.09)	0.31 (0.07)	0.29 (0.05)
		0.06 (0.01)	0.07 (0.03)	0.14 (0.10)	0.19 (0.07)	0.21 (0.08)	0.21 (0.05)
		0.20 (0.05)	0.22 (0.04)	0.10 (0.08)	0.15 (0.07)	0.22 (0.07)	0.19 (0.07)
	100	0.10 (0.02)	0.12 (0.04)	0.26 (0.05)	0.31 (0.09)	0.37 (0.05)	0.43 (0.13)
		0.09 (0.01)	0.12 (0.05)	0.26 (0.05)	0.28 (0.07)	0.33 (0.03)	0.35 (0.04)
		0.09 (0.01)	0.11 (0.03)	0.26 (0.05)	0.28 (0.05)	0.33 (0.03)	0.33 (0.03)
		0.10 (0.08)	0.15 (0.10)	0.28 (0.09)	0.37 (0.19)	0.29 (0.08)	0.26 (0.02)
		2.92 (1.31)	2.18 (1.31)	2.63 (1.06)	2.93 (0.81)	0.77 (0.45)	0.36 (0.22)
		0.26 (0.16)	0.24 (0.14)	0.32 (0.17)	0.54 (0.23)	0.37 (0.14)	0.37 (0.12)
		0.10 (0.01)	0.11 (0.03)	0.24 (0.04)	0.27 (0.05)	0.33 (0.02)	0.33 (0.03)
		0.09 (0.01)	0.10 (0.02)	0.24 (0.05)	0.26 (0.05)	0.31 (0.02)	0.32 (0.03)
		0.07 (0.01)	0.08 (0.02)	0.22 (0.04)	0.24 (0.05)	0.22 (0.07)	0.25 (0.07)
		0.07 (0.05)	0.04 (0.02)	0.10 (0.05)	0.15 (0.08)	0.20 (0.03)	0.25 (0.02)

Loss Values - Low Setup Times, 1-4 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.03)	0.23 (0.08)	0.18 (0.09)	0.25 (0.10)	0.34 (0.18)	0.44 (0.23)
		0.09 (0.02)	0.18 (0.07)	0.14 (0.05)	0.17 (0.09)	0.19 (0.14)	0.27 (0.13)
		0.07 (0.02)	0.19 (0.05)	0.15 (0.05)	0.15 (0.11)	0.21 (0.20)	0.25 (0.08)
		0.06 (0.02)	0.15 (0.05)	0.13 (0.06)	0.16 (0.12)	0.18 (0.09)	0.23 (0.12)
		0.06 (0.02)	0.15 (0.05)	0.13 (0.06)	0.16 (0.12)	0.18 (0.09)	0.23 (0.12)
		0.18 (0.06)	0.26 (0.09)	0.21 (0.08)	0.19 (0.09)	0.25 (0.12)	0.28 (0.11)
		0.07 (0.02)	0.14 (0.05)	0.15 (0.06)	0.19 (0.10)	0.22 (0.20)	0.25 (0.10)
		0.07 (0.02)	0.18 (0.09)	0.12 (0.06)	0.14 (0.10)	0.20 (0.13)	0.26 (0.14)
		0.04 (0.01)	0.08 (0.02)	0.08 (0.05)	0.08 (0.07)	0.10 (0.07)	0.14 (0.07)
		0.09 (0.06)	0.09 (0.04)	0.17 (0.03)	0.19 (0.09)	0.23 (0.07)	0.27 (0.07)
	20	0.10 (0.01)	0.21 (0.09)	0.24 (0.08)	0.30 (0.11)	0.76 (0.17)	0.65 (0.22)
		0.10 (0.02)	0.19 (0.07)	0.21 (0.11)	0.26 (0.13)	0.50 (0.13)	0.45 (0.14)
		0.10 (0.01)	0.14 (0.04)	0.20 (0.08)	0.28 (0.13)	0.47 (0.12)	0.44 (0.14)
		0.08 (0.02)	0.12 (0.07)	0.17 (0.09)	0.22 (0.11)	0.42 (0.10)	0.44 (0.17)
		0.08 (0.02)	0.12 (0.07)	0.17 (0.09)	0.22 (0.11)	0.42 (0.10)	0.44 (0.17)
		0.45 (0.23)	0.44 (0.16)	0.42 (0.19)	0.64 (0.27)	0.68 (0.17)	0.54 (0.13)
		0.09 (0.01)	0.14 (0.06)	0.18 (0.11)	0.23 (0.11)	0.42 (0.11)	0.39 (0.12)
		0.08 (0.01)	0.13 (0.05)	0.16 (0.08)	0.23 (0.11)	0.39 (0.12)	0.38 (0.11)
		0.06 (0.01)	0.07 (0.04)	0.12 (0.08)	0.16 (0.09)	0.29 (0.08)	0.24 (0.10)
		0.13 (0.11)	0.38 (0.07)	0.18 (0.06)	0.06 (0.06)	0.04 (0.02)	0.23 (0.03)
	30	0.10 (0.02)	0.19 (0.05)	0.21 (0.06)	0.30 (0.16)	0.71 (0.23)	0.64 (0.10)
		0.10 (0.02)	0.18 (0.06)	0.18 (0.04)	0.25 (0.13)	0.49 (0.18)	0.44 (0.09)
		0.09 (0.02)	0.15 (0.05)	0.17 (0.05)	0.24 (0.10)	0.50 (0.16)	0.43 (0.08)
		0.06 (0.03)	0.12 (0.04)	0.13 (0.05)	0.19 (0.09)	0.45 (0.15)	0.38 (0.06)
		0.06 (0.03)	0.12 (0.04)	0.13 (0.05)	0.19 (0.09)	0.45 (0.15)	0.38 (0.06)
		0.56 (0.26)	0.53 (0.20)	0.53 (0.31)	0.62 (0.29)	0.84 (0.29)	0.75 (0.17)
		0.09 (0.02)	0.15 (0.05)	0.14 (0.04)	0.22 (0.10)	0.46 (0.14)	0.35 (0.04)
		0.08 (0.01)	0.12 (0.03)	0.13 (0.04)	0.22 (0.10)	0.38 (0.12)	0.32 (0.04)
		0.06 (0.01)	0.09 (0.03)	0.10 (0.04)	0.17 (0.10)	0.32 (0.11)	0.25 (0.04)
		0.20 (0.06)	0.08 (0.07)	0.11 (0.07)	0.15 (0.12)	0.09 (0.04)	0.21 (0.05)
	100	0.09 (0.01)	0.13 (0.03)	0.30 (0.04)	0.34 (0.07)	0.44 (0.11)	0.47 (0.13)
		0.09 (0.01)	0.12 (0.03)	0.28 (0.03)	0.30 (0.07)	0.36 (0.06)	0.34 (0.05)
		0.09 (0.01)	0.12 (0.02)	0.28 (0.03)	0.28 (0.07)	0.36 (0.05)	0.33 (0.05)
		0.18 (0.15)	0.29 (0.18)	0.28 (0.06)	0.34 (0.14)	0.31 (0.08)	0.28 (0.04)
		5.72 (1.67)	5.15 (0.74)	5.95 (0.76)	7.18 (2.20)	3.08 (1.76)	2.01 (2.07)
		0.68 (0.27)	0.74 (0.31)	0.91 (0.33)	0.98 (0.22)	0.80 (0.32)	0.69 (0.18)
		0.09 (0.01)	0.11 (0.02)	0.27 (0.03)	0.27 (0.07)	0.36 (0.06)	0.32 (0.04)
		0.08 (0.01)	0.10 (0.02)	0.27 (0.03)	0.26 (0.06)	0.32 (0.04)	0.31 (0.04)
		0.07 (0.01)	0.08 (0.01)	0.25 (0.03)	0.24 (0.06)	0.30 (0.05)	0.27 (0.05)
		0.17 (0.07)	0.10 (0.07)	0.15 (0.06)	0.15 (0.08)	0.33 (0.05)	0.24 (0.02)

Loss Values - Low Setup Times, 1-10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.13 (0.09)	0.18 (0.12)	0.10 (0.08)	0.25 (0.19)	0.26 (0.24)	0.09 (0.15)
		0.13 (0.13)	0.16 (0.10)	0.10 (0.08)	0.17 (0.17)	0.20 (0.22)	0.08 (0.13)
		0.13 (0.15)	0.18 (0.12)	0.10 (0.08)	0.19 (0.18)	0.15 (0.17)	0.09 (0.11)
		0.09 (0.11)	0.12 (0.07)	0.08 (0.07)	0.15 (0.14)	0.18 (0.18)	0.08 (0.10)
		0.09 (0.11)	0.12 (0.07)	0.08 (0.07)	0.15 (0.14)	0.18 (0.18)	0.08 (0.10)
		0.08 (0.09)	0.12 (0.08)	0.08 (0.07)	0.13 (0.12)	0.10 (0.13)	0.10 (0.15)
		0.13 (0.15)	0.11 (0.08)	0.10 (0.07)	0.14 (0.15)	0.11 (0.13)	0.04 (0.06)
		0.13 (0.15)	0.11 (0.08)	0.10 (0.07)	0.14 (0.15)	0.11 (0.13)	0.04 (0.06)
		0.08 (0.12)	0.05 (0.05)	0.06 (0.07)	0.07 (0.07)	0.06 (0.06)	0.03 (0.06)
		0.11 (0.05)	0.16 (0.07)	0.10 (0.10)	0.19 (0.05)	0.20 (0.10)	0.24 (0.04)
	20	0.13 (0.08)	0.23 (0.12)	0.27 (0.13)	0.47 (0.20)	0.50 (0.14)	0.52 (0.18)
		0.12 (0.07)	0.21 (0.12)	0.25 (0.14)	0.41 (0.21)	0.45 (0.18)	0.49 (0.16)
		0.14 (0.07)	0.20 (0.13)	0.26 (0.13)	0.43 (0.15)	0.43 (0.19)	0.45 (0.22)
		0.09 (0.07)	0.11 (0.07)	0.21 (0.13)	0.31 (0.13)	0.35 (0.17)	0.37 (0.13)
		0.09 (0.07)	0.11 (0.07)	0.21 (0.13)	0.31 (0.13)	0.35 (0.17)	0.37 (0.13)
		0.08 (0.07)	0.10 (0.06)	0.18 (0.11)	0.27 (0.16)	0.34 (0.14)	0.27 (0.11)
		0.12 (0.07)	0.14 (0.09)	0.20 (0.08)	0.27 (0.10)	0.28 (0.16)	0.20 (0.08)
		0.12 (0.07)	0.14 (0.09)	0.20 (0.08)	0.27 (0.10)	0.28 (0.16)	0.20 (0.08)
		0.08 (0.06)	0.09 (0.06)	0.15 (0.09)	0.24 (0.14)	0.20 (0.10)	0.11 (0.06)
		0.28 (0.04)	0.22 (0.08)	0.23 (0.03)	0.22 (0.02)	0.08 (0.07)	0.01 (0.00)
	30	0.10 (0.02)	0.36 (0.18)	0.34 (0.20)	0.46 (0.19)	0.49 (0.27)	0.47 (0.20)
		0.09 (0.02)	0.28 (0.13)	0.30 (0.20)	0.41 (0.18)	0.38 (0.16)	0.42 (0.15)
		0.09 (0.02)	0.26 (0.12)	0.31 (0.21)	0.37 (0.14)	0.39 (0.16)	0.44 (0.18)
		0.03 (0.02)	0.18 (0.09)	0.25 (0.19)	0.29 (0.14)	0.30 (0.13)	0.32 (0.11)
		0.03 (0.02)	0.18 (0.09)	0.25 (0.19)	0.29 (0.14)	0.30 (0.13)	0.32 (0.11)
		0.03 (0.01)	0.17 (0.10)	0.26 (0.18)	0.28 (0.12)	0.33 (0.16)	0.32 (0.15)
		0.08 (0.01)	0.20 (0.10)	0.25 (0.15)	0.29 (0.11)	0.28 (0.07)	0.37 (0.10)
		0.08 (0.01)	0.20 (0.10)	0.25 (0.15)	0.29 (0.11)	0.28 (0.07)	0.37 (0.10)
		0.05 (0.02)	0.16 (0.09)	0.24 (0.18)	0.26 (0.11)	0.23 (0.14)	0.22 (0.10)
		0.26 (0.07)	0.23 (0.07)	0.07 (0.08)	0.14 (0.10)	0.12 (0.09)	0.06 (0.03)
	100	0.11 (0.03)	0.17 (0.05)	0.30 (0.05)	0.39 (0.14)	0.36 (0.05)	0.42 (0.10)
		0.11 (0.03)	0.16 (0.05)	0.27 (0.06)	0.38 (0.14)	0.35 (0.05)	0.41 (0.09)
		0.11 (0.02)	0.14 (0.06)	0.27 (0.05)	0.31 (0.08)	0.32 (0.04)	0.37 (0.08)
		0.12 (0.11)	0.07 (0.03)	0.27 (0.11)	0.28 (0.06)	0.27 (0.04)	0.29 (0.06)
		0.42 (0.44)	0.30 (0.28)	0.54 (0.41)	0.36 (0.20)	0.28 (0.04)	0.28 (0.05)
		0.07 (0.07)	0.07 (0.05)	0.21 (0.05)	0.25 (0.10)	0.25 (0.04)	0.26 (0.07)
		0.10 (0.02)	0.11 (0.02)	0.25 (0.03)	0.29 (0.05)	0.30 (0.02)	0.34 (0.06)
		0.10 (0.02)	0.11 (0.02)	0.25 (0.03)	0.29 (0.05)	0.30 (0.02)	0.34 (0.06)
		0.09 (0.02)	0.09 (0.02)	0.24 (0.04)	0.27 (0.07)	0.21 (0.05)	0.23 (0.09)
		0.26 (0.02)	0.10 (0.08)	0.16 (0.06)	0.18 (0.05)	0.18 (0.11)	0.25 (0.10)

Loss Values - Low Setup Times, 1-10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.08 (0.02)	0.17 (0.10)	0.21 (0.10)	0.27 (0.17)	0.38 (0.26)	0.30 (0.23)
		0.08 (0.03)	0.17 (0.11)	0.17 (0.08)	0.21 (0.12)	0.24 (0.15)	0.13 (0.10)
		0.06 (0.02)	0.18 (0.08)	0.14 (0.07)	0.21 (0.15)	0.27 (0.13)	0.16 (0.16)
		0.04 (0.02)	0.14 (0.07)	0.14 (0.08)	0.18 (0.14)	0.29 (0.16)	0.14 (0.11)
		0.04 (0.02)	0.14 (0.07)	0.14 (0.08)	0.18 (0.14)	0.29 (0.16)	0.14 (0.11)
		0.04 (0.02)	0.17 (0.06)	0.13 (0.08)	0.18 (0.14)	0.32 (0.22)	0.16 (0.12)
		0.05 (0.02)	0.16 (0.09)	0.11 (0.06)	0.16 (0.11)	0.24 (0.14)	0.16 (0.16)
		0.04 (0.01)	0.18 (0.09)	0.13 (0.08)	0.20 (0.15)	0.23 (0.14)	0.14 (0.13)
		0.02 (0.01)	0.10 (0.06)	0.09 (0.05)	0.09 (0.05)	0.15 (0.11)	0.09 (0.08)
		0.03 (0.02)	0.13 (0.06)	0.11 (0.08)	0.18 (0.03)	0.21 (0.08)	0.13 (0.08)
	20	0.13 (0.04)	0.30 (0.12)	0.32 (0.15)	0.33 (0.13)	0.68 (0.26)	0.47 (0.20)
		0.12 (0.04)	0.23 (0.09)	0.28 (0.14)	0.33 (0.14)	0.51 (0.30)	0.43 (0.19)
		0.12 (0.03)	0.23 (0.12)	0.29 (0.15)	0.27 (0.13)	0.60 (0.28)	0.38 (0.22)
		0.07 (0.03)	0.17 (0.09)	0.24 (0.14)	0.24 (0.11)	0.43 (0.21)	0.31 (0.20)
		0.07 (0.03)	0.17 (0.09)	0.24 (0.14)	0.24 (0.11)	0.43 (0.21)	0.31 (0.20)
		0.16 (0.11)	0.26 (0.12)	0.26 (0.13)	0.26 (0.09)	0.53 (0.30)	0.33 (0.14)
		0.10 (0.03)	0.19 (0.11)	0.25 (0.14)	0.25 (0.13)	0.43 (0.17)	0.28 (0.13)
		0.11 (0.02)	0.17 (0.08)	0.25 (0.11)	0.25 (0.17)	0.36 (0.18)	0.29 (0.14)
		0.07 (0.03)	0.13 (0.07)	0.18 (0.12)	0.16 (0.09)	0.27 (0.16)	0.17 (0.12)
		0.07 (0.02)	0.17 (0.05)	0.18 (0.10)	0.20 (0.07)	0.25 (0.10)	0.17 (0.07)
	30	0.14 (0.04)	0.18 (0.09)	0.31 (0.11)	0.34 (0.20)	0.55 (0.25)	0.78 (0.38)
		0.11 (0.03)	0.17 (0.07)	0.27 (0.10)	0.30 (0.19)	0.49 (0.24)	0.51 (0.28)
		0.11 (0.03)	0.15 (0.06)	0.27 (0.09)	0.29 (0.18)	0.47 (0.20)	0.53 (0.18)
		0.08 (0.05)	0.11 (0.06)	0.22 (0.10)	0.20 (0.15)	0.41 (0.20)	0.43 (0.16)
		0.08 (0.05)	0.11 (0.06)	0.22 (0.10)	0.20 (0.15)	0.41 (0.20)	0.43 (0.16)
		0.18 (0.12)	0.15 (0.10)	0.37 (0.09)	0.27 (0.18)	0.49 (0.21)	0.45 (0.23)
		0.11 (0.03)	0.13 (0.06)	0.23 (0.09)	0.27 (0.18)	0.41 (0.18)	0.44 (0.20)
		0.11 (0.04)	0.13 (0.06)	0.21 (0.08)	0.27 (0.20)	0.37 (0.10)	0.40 (0.18)
		0.07 (0.02)	0.08 (0.04)	0.19 (0.08)	0.20 (0.16)	0.29 (0.13)	0.28 (0.17)
		0.39 (0.05)	0.17 (0.06)	0.03 (0.06)	0.03 (0.02)	0.26 (0.07)	0.19 (0.04)
	100	0.11 (0.02)	0.16 (0.06)	0.32 (0.09)	0.33 (0.12)	0.47 (0.14)	0.46 (0.10)
		0.10 (0.02)	0.14 (0.04)	0.29 (0.08)	0.31 (0.09)	0.36 (0.07)	0.40 (0.10)
		0.10 (0.02)	0.13 (0.03)	0.30 (0.10)	0.30 (0.06)	0.35 (0.08)	0.37 (0.08)
		0.15 (0.14)	0.16 (0.11)	0.29 (0.06)	0.33 (0.16)	0.29 (0.08)	0.29 (0.05)
		2.17 (1.77)	1.64 (1.19)	1.78 (1.41)	2.05 (1.03)	0.43 (0.23)	0.29 (0.04)
		0.19 (0.16)	0.16 (0.09)	0.44 (0.20)	0.28 (0.15)	0.43 (0.15)	0.30 (0.08)
		0.09 (0.01)	0.12 (0.03)	0.27 (0.07)	0.27 (0.06)	0.33 (0.05)	0.35 (0.06)
		0.09 (0.01)	0.12 (0.06)	0.27 (0.08)	0.28 (0.07)	0.33 (0.05)	0.35 (0.05)
		0.08 (0.02)	0.09 (0.03)	0.26 (0.09)	0.24 (0.06)	0.30 (0.06)	0.26 (0.04)
		0.10 (0.07)	0.15 (0.09)	0.05 (0.04)	0.05 (0.03)	0.22 (0.04)	0.22 (0.06)

Loss Values - Low Setup Times, 1-10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.08 (0.02)	0.23 (0.07)	0.14 (0.06)	0.21 (0.10)	0.39 (0.15)	0.28 (0.14)
		0.07 (0.02)	0.19 (0.08)	0.13 (0.05)	0.17 (0.09)	0.25 (0.11)	0.22 (0.12)
		0.06 (0.02)	0.16 (0.07)	0.12 (0.06)	0.16 (0.06)	0.27 (0.13)	0.22 (0.11)
		0.05 (0.02)	0.14 (0.07)	0.12 (0.06)	0.15 (0.06)	0.22 (0.10)	0.22 (0.13)
		0.05 (0.02)	0.14 (0.07)	0.12 (0.06)	0.15 (0.06)	0.22 (0.10)	0.22 (0.13)
		0.11 (0.08)	0.23 (0.08)	0.16 (0.09)	0.23 (0.06)	0.25 (0.12)	0.24 (0.16)
		0.06 (0.02)	0.16 (0.07)	0.11 (0.06)	0.13 (0.03)	0.25 (0.11)	0.22 (0.14)
		0.06 (0.02)	0.15 (0.08)	0.10 (0.06)	0.10 (0.03)	0.26 (0.13)	0.21 (0.13)
		0.04 (0.01)	0.08 (0.04)	0.06 (0.04)	0.07 (0.03)	0.13 (0.07)	0.12 (0.05)
		0.21 (0.06)	0.16 (0.04)	0.14 (0.07)	0.30 (0.05)	0.24 (0.03)	0.12 (0.05)
	20	0.10 (0.03)	0.27 (0.12)	0.25 (0.09)	0.40 (0.15)	0.64 (0.20)	0.65 (0.28)
		0.10 (0.02)	0.22 (0.10)	0.22 (0.10)	0.30 (0.17)	0.44 (0.18)	0.38 (0.18)
		0.10 (0.03)	0.19 (0.10)	0.23 (0.10)	0.31 (0.16)	0.40 (0.15)	0.43 (0.21)
		0.06 (0.02)	0.16 (0.10)	0.19 (0.09)	0.27 (0.15)	0.39 (0.14)	0.34 (0.16)
		0.06 (0.02)	0.16 (0.10)	0.19 (0.09)	0.27 (0.15)	0.39 (0.14)	0.34 (0.16)
		0.22 (0.12)	0.34 (0.18)	0.41 (0.21)	0.44 (0.19)	0.46 (0.21)	0.51 (0.19)
		0.09 (0.02)	0.17 (0.07)	0.20 (0.10)	0.27 (0.16)	0.40 (0.18)	0.38 (0.21)
		0.10 (0.02)	0.15 (0.08)	0.17 (0.10)	0.25 (0.12)	0.39 (0.16)	0.35 (0.16)
		0.06 (0.02)	0.12 (0.06)	0.13 (0.08)	0.19 (0.12)	0.22 (0.09)	0.19 (0.09)
		0.22 (0.08)	0.15 (0.06)	0.17 (0.10)	0.09 (0.10)	0.11 (0.05)	0.14 (0.09)
	30	0.11 (0.03)	0.16 (0.08)	0.28 (0.10)	0.36 (0.13)	0.69 (0.24)	0.65 (0.27)
		0.10 (0.04)	0.15 (0.07)	0.21 (0.08)	0.29 (0.11)	0.49 (0.17)	0.46 (0.24)
		0.10 (0.02)	0.13 (0.05)	0.23 (0.11)	0.28 (0.10)	0.49 (0.19)	0.48 (0.22)
		0.07 (0.04)	0.08 (0.04)	0.19 (0.11)	0.25 (0.10)	0.44 (0.17)	0.39 (0.19)
		0.07 (0.04)	0.08 (0.04)	0.19 (0.11)	0.25 (0.10)	0.44 (0.17)	0.39 (0.19)
		0.26 (0.21)	0.24 (0.15)	0.39 (0.17)	0.49 (0.21)	0.61 (0.17)	0.61 (0.20)
		0.09 (0.03)	0.13 (0.05)	0.21 (0.10)	0.24 (0.08)	0.45 (0.19)	0.45 (0.20)
		0.09 (0.03)	0.11 (0.05)	0.19 (0.08)	0.26 (0.11)	0.45 (0.15)	0.42 (0.18)
		0.06 (0.03)	0.07 (0.03)	0.16 (0.08)	0.19 (0.08)	0.29 (0.11)	0.29 (0.11)
		0.25 (0.03)	0.20 (0.07)	0.12 (0.07)	0.15 (0.09)	0.32 (0.08)	0.21 (0.05)
	100	0.10 (0.02)	0.12 (0.03)	0.30 (0.07)	0.36 (0.10)	0.56 (0.25)	0.55 (0.23)
		0.10 (0.02)	0.12 (0.03)	0.26 (0.04)	0.32 (0.08)	0.41 (0.13)	0.41 (0.09)
		0.09 (0.01)	0.11 (0.02)	0.26 (0.05)	0.31 (0.07)	0.43 (0.15)	0.40 (0.09)
		0.20 (0.16)	0.14 (0.13)	0.32 (0.15)	0.31 (0.09)	0.36 (0.14)	0.34 (0.08)
		6.99 (2.68)	4.73 (1.63)	6.02 (4.01)	5.59 (2.89)	0.83 (0.63)	1.03 (1.63)
		0.39 (0.22)	0.29 (0.29)	0.51 (0.27)	0.65 (0.33)	0.62 (0.32)	0.59 (0.29)
		0.09 (0.01)	0.10 (0.01)	0.25 (0.04)	0.29 (0.06)	0.40 (0.14)	0.39 (0.07)
		0.09 (0.01)	0.09 (0.01)	0.24 (0.04)	0.27 (0.06)	0.36 (0.11)	0.35 (0.06)
		0.07 (0.01)	0.07 (0.01)	0.23 (0.04)	0.25 (0.06)	0.32 (0.13)	0.30 (0.08)
		0.13 (0.07)	0.01 (0.00)	0.14 (0.04)	0.16 (0.06)	0.28 (0.06)	0.22 (0.06)

Loss Values - High Setup Times, 1 Machine per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.07 (0.01)	0.10 (0.06)	0.13 (0.09)	0.14 (0.07)	0.27 (0.20)	0.21 (0.15)
		0.07 (0.01)	0.10 (0.06)	0.10 (0.09)	0.09 (0.05)	0.17 (0.10)	0.14 (0.10)
		0.05 (0.02)	0.07 (0.05)	0.12 (0.08)	0.08 (0.05)	0.14 (0.07)	0.16 (0.13)
		0.04 (0.01)	0.07 (0.04)	0.11 (0.08)	0.08 (0.05)	0.10 (0.07)	0.16 (0.14)
		0.04 (0.01)	0.07 (0.04)	0.11 (0.08)	0.08 (0.05)	0.10 (0.07)	0.16 (0.14)
		0.09 (0.06)	0.12 (0.06)	0.13 (0.08)	0.10 (0.04)	0.09 (0.07)	0.18 (0.22)
		0.05 (0.01)	0.05 (0.02)	0.04 (0.02)	0.06 (0.04)	0.12 (0.06)	0.13 (0.08)
		0.05 (0.01)	0.05 (0.02)	0.04 (0.02)	0.06 (0.04)	0.12 (0.06)	0.13 (0.08)
		X	X	X	X	X	X
		0.02 (0.01)	0.02 (0.02)	0.02 (0.01)	0.02 (0.02)	0.06 (0.06)	0.07 (0.05)
	20	0.09 (0.01)	0.11 (0.03)	0.20 (0.06)	0.21 (0.10)	0.33 (0.11)	0.44 (0.16)
		0.09 (0.01)	0.11 (0.03)	0.19 (0.07)	0.19 (0.10)	0.27 (0.06)	0.27 (0.08)
		0.07 (0.01)	0.09 (0.03)	0.19 (0.05)	0.14 (0.08)	0.28 (0.06)	0.26 (0.07)
		0.06 (0.02)	0.05 (0.01)	0.13 (0.05)	0.12 (0.06)	0.24 (0.04)	0.22 (0.07)
		0.06 (0.02)	0.05 (0.01)	0.13 (0.05)	0.12 (0.06)	0.24 (0.04)	0.22 (0.07)
		0.20 (0.04)	0.20 (0.08)	0.30 (0.05)	0.24 (0.11)	0.25 (0.05)	0.23 (0.07)
		0.08 (0.01)	0.07 (0.01)	0.14 (0.06)	0.12 (0.07)	0.23 (0.04)	0.22 (0.05)
		0.08 (0.01)	0.07 (0.01)	0.14 (0.06)	0.12 (0.07)	0.23 (0.04)	0.22 (0.05)
		X	X	X	X	X	X
		0.02 (0.01)	0.02 (0.01)	0.08 (0.05)	0.06 (0.06)	0.11 (0.06)	0.14 (0.07)
	30	0.09 (0.01)	0.12 (0.02)	0.17 (0.07)	0.23 (0.07)	0.42 (0.11)	0.42 (0.12)
		0.09 (0.01)	0.12 (0.02)	0.15 (0.07)	0.22 (0.06)	0.29 (0.05)	0.30 (0.05)
		0.08 (0.01)	0.09 (0.02)	0.14 (0.06)	0.18 (0.08)	0.30 (0.06)	0.27 (0.05)
		0.07 (0.01)	0.05 (0.01)	0.11 (0.06)	0.14 (0.07)	0.25 (0.04)	0.23 (0.04)
		0.07 (0.01)	0.05 (0.01)	0.11 (0.06)	0.14 (0.07)	0.25 (0.04)	0.23 (0.04)
		0.24 (0.03)	0.22 (0.02)	0.26 (0.07)	0.32 (0.05)	0.22 (0.05)	0.27 (0.08)
		0.08 (0.01)	0.08 (0.01)	0.12 (0.06)	0.16 (0.07)	0.26 (0.04)	0.25 (0.03)
		0.08 (0.01)	0.08 (0.01)	0.12 (0.06)	0.16 (0.07)	0.26 (0.04)	0.25 (0.03)
		X	X	X	X	X	X
		0.03 (0.01)	0.02 (0.01)	0.07 (0.05)	0.10 (0.06)	0.12 (0.06)	0.15 (0.04)
	100	0.09 (0.00)	0.10 (0.01)	0.28 (0.05)	0.25 (0.06)	0.34 (0.05)	0.32 (0.04)
		0.09 (0.00)	0.10 (0.01)	0.26 (0.05)	0.23 (0.05)	0.29 (0.02)	0.30 (0.02)
		0.08 (0.00)	0.09 (0.01)	0.26 (0.05)	0.22 (0.05)	0.30 (0.02)	0.30 (0.02)
		0.07 (0.01)	0.06 (0.00)	0.23 (0.05)	0.19 (0.05)	0.24 (0.02)	0.31 (0.26)
		0.84 (0.07)	0.79 (0.14)	1.17 (0.14)	1.03 (0.14)	0.74 (0.24)	0.82 (0.35)
		0.29 (0.01)	0.27 (0.04)	0.47 (0.07)	0.40 (0.07)	0.29 (0.05)	0.30 (0.06)
		0.08 (0.00)	0.08 (0.01)	0.25 (0.05)	0.22 (0.05)	0.29 (0.01)	0.28 (0.02)
		0.08 (0.00)	0.08 (0.01)	0.25 (0.05)	0.22 (0.05)	0.29 (0.01)	0.28 (0.02)
		X	X	X	X	X	X
		0.04 (0.00)	0.04 (0.01)	0.20 (0.05)	0.16 (0.04)	0.17 (0.06)	0.17 (0.06)

Loss Values - High Setup Times, 1 Machine per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.02)	0.21 (0.08)	0.17 (0.10)	0.28 (0.12)	0.46 (0.18)	0.47 (0.13)
		0.09 (0.02)	0.21 (0.08)	0.15 (0.08)	0.25 (0.09)	0.21 (0.12)	0.30 (0.13)
		0.08 (0.02)	0.15 (0.07)	0.17 (0.09)	0.16 (0.09)	0.31 (0.15)	0.33 (0.11)
		0.08 (0.02)	0.15 (0.08)	0.16 (0.10)	0.16 (0.09)	0.24 (0.16)	0.32 (0.16)
		0.08 (0.02)	0.15 (0.08)	0.16 (0.10)	0.16 (0.09)	0.24 (0.16)	0.32 (0.16)
		0.19 (0.08)	0.23 (0.08)	0.21 (0.07)	0.30 (0.08)	0.23 (0.16)	0.30 (0.17)
		0.08 (0.02)	0.14 (0.07)	0.14 (0.09)	0.15 (0.05)	0.28 (0.12)	0.30 (0.14)
		0.07 (0.01)	0.14 (0.06)	0.14 (0.09)	0.16 (0.07)	0.21 (0.08)	0.29 (0.10)
		X	X	X	X	X	X
		0.05 (0.01)	0.08 (0.04)	0.11 (0.07)	0.10 (0.06)	0.10 (0.06)	0.16 (0.08)
	20	0.11 (0.01)	0.18 (0.03)	0.19 (0.04)	0.29 (0.09)	0.52 (0.16)	0.48 (0.12)
		0.11 (0.01)	0.18 (0.03)	0.17 (0.02)	0.27 (0.08)	0.34 (0.12)	0.33 (0.11)
		0.09 (0.01)	0.14 (0.04)	0.16 (0.03)	0.23 (0.06)	0.36 (0.07)	0.36 (0.12)
		0.08 (0.02)	0.12 (0.04)	0.15 (0.03)	0.19 (0.06)	0.33 (0.10)	0.29 (0.08)
		0.08 (0.02)	0.12 (0.04)	0.15 (0.03)	0.19 (0.06)	0.33 (0.10)	0.29 (0.08)
		0.52 (0.03)	0.48 (0.09)	0.58 (0.07)	0.57 (0.09)	0.47 (0.17)	0.41 (0.08)
		0.09 (0.01)	0.13 (0.04)	0.12 (0.03)	0.19 (0.05)	0.30 (0.09)	0.30 (0.08)
		0.09 (0.01)	0.11 (0.04)	0.11 (0.03)	0.19 (0.06)	0.30 (0.08)	0.25 (0.06)
		X	X	X	X	X	X
		0.04 (0.01)	0.04 (0.02)	0.06 (0.02)	0.09 (0.04)	0.16 (0.05)	0.15 (0.04)
	30	0.11 (0.01)	0.17 (0.05)	0.23 (0.08)	0.26 (0.06)	0.55 (0.10)	0.57 (0.11)
		0.11 (0.01)	0.17 (0.05)	0.19 (0.07)	0.22 (0.06)	0.33 (0.05)	0.40 (0.08)
		0.10 (0.01)	0.13 (0.03)	0.19 (0.06)	0.18 (0.05)	0.35 (0.06)	0.38 (0.08)
		0.10 (0.01)	0.10 (0.02)	0.17 (0.07)	0.16 (0.06)	0.30 (0.05)	0.33 (0.07)
		0.10 (0.01)	0.10 (0.02)	0.17 (0.07)	0.16 (0.06)	0.30 (0.05)	0.33 (0.07)
		0.62 (0.07)	0.56 (0.08)	0.68 (0.10)	0.61 (0.09)	0.55 (0.16)	0.53 (0.13)
		0.09 (0.01)	0.12 (0.02)	0.16 (0.05)	0.17 (0.05)	0.29 (0.04)	0.34 (0.06)
		0.09 (0.01)	0.10 (0.02)	0.16 (0.06)	0.17 (0.05)	0.28 (0.02)	0.32 (0.06)
		X	X	X	X	X	X
		0.04 (0.01)	0.04 (0.01)	0.10 (0.05)	0.08 (0.04)	0.19 (0.03)	0.17 (0.05)
	100	0.10 (0.01)	0.13 (0.02)	0.24 (0.05)	0.29 (0.04)	0.47 (0.08)	0.43 (0.07)
		0.10 (0.01)	0.13 (0.02)	0.23 (0.05)	0.27 (0.05)	0.31 (0.02)	0.32 (0.04)
		0.09 (0.01)	0.11 (0.01)	0.23 (0.05)	0.26 (0.05)	0.30 (0.02)	0.31 (0.03)
		0.09 (0.01)	0.10 (0.01)	0.20 (0.05)	0.22 (0.05)	0.26 (0.02)	0.31 (0.13)
		2.50 (0.15)	2.50 (0.11)	2.50 (0.21)	2.50 (0.27)	2.21 (0.69)	2.25 (0.47)
		0.81 (0.03)	0.72 (0.04)	0.99 (0.13)	0.91 (0.08)	0.70 (0.10)	0.63 (0.10)
		0.09 (0.01)	0.10 (0.01)	0.21 (0.05)	0.23 (0.05)	0.30 (0.02)	0.29 (0.02)
		0.09 (0.00)	0.09 (0.01)	0.21 (0.05)	0.24 (0.06)	0.30 (0.02)	0.28 (0.03)
		X	X	X	X	X	X
		0.05 (0.00)	0.04 (0.01)	0.16 (0.05)	0.16 (0.05)	0.23 (0.02)	0.21 (0.04)

Loss Values - High Setup Times, 1 Machine per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.02)	0.27 (0.09)	0.20 (0.05)	0.33 (0.09)	0.51 (0.16)	0.52 (0.17)
		0.11 (0.02)	0.27 (0.09)	0.18 (0.04)	0.31 (0.09)	0.34 (0.11)	0.35 (0.16)
		0.09 (0.02)	0.27 (0.08)	0.18 (0.05)	0.26 (0.06)	0.41 (0.16)	0.37 (0.13)
		0.10 (0.02)	0.25 (0.08)	0.16 (0.04)	0.25 (0.06)	0.44 (0.18)	0.35 (0.14)
		0.10 (0.02)	0.25 (0.08)	0.16 (0.04)	0.25 (0.06)	0.44 (0.18)	0.35 (0.14)
		0.29 (0.09)	0.41 (0.13)	0.33 (0.09)	0.42 (0.05)	0.42 (0.12)	0.44 (0.17)
		0.09 (0.02)	0.21 (0.07)	0.16 (0.04)	0.24 (0.05)	0.32 (0.08)	0.39 (0.17)
		0.09 (0.02)	0.19 (0.06)	0.17 (0.04)	0.25 (0.06)	0.33 (0.06)	0.35 (0.10)
		X	X	X	X	X	X
		0.07 (0.01)	0.16 (0.05)	0.13 (0.03)	0.19 (0.05)	0.19 (0.04)	0.23 (0.09)
	20	0.12 (0.01)	0.27 (0.04)	0.21 (0.05)	0.33 (0.05)	0.72 (0.18)	0.69 (0.17)
		0.12 (0.01)	0.27 (0.04)	0.20 (0.04)	0.26 (0.03)	0.52 (0.15)	0.48 (0.12)
		0.12 (0.01)	0.21 (0.04)	0.19 (0.05)	0.26 (0.04)	0.53 (0.16)	0.42 (0.11)
		0.11 (0.02)	0.21 (0.04)	0.19 (0.04)	0.25 (0.03)	0.50 (0.15)	0.39 (0.09)
		0.11 (0.02)	0.21 (0.04)	0.19 (0.04)	0.25 (0.03)	0.50 (0.15)	0.39 (0.09)
		1.03 (0.11)	0.93 (0.08)	1.05 (0.11)	1.00 (0.07)	0.87 (0.19)	0.75 (0.12)
		0.11 (0.02)	0.19 (0.03)	0.16 (0.04)	0.23 (0.05)	0.48 (0.14)	0.41 (0.10)
		0.11 (0.01)	0.19 (0.04)	0.16 (0.05)	0.23 (0.05)	0.42 (0.14)	0.34 (0.08)
		X	X	X	X	X	X
		0.07 (0.01)	0.09 (0.02)	0.10 (0.03)	0.10 (0.04)	0.21 (0.08)	0.17 (0.04)
	30	0.13 (0.01)	0.23 (0.03)	0.20 (0.02)	0.32 (0.08)	0.71 (0.11)	0.73 (0.15)
		0.13 (0.01)	0.23 (0.03)	0.18 (0.02)	0.31 (0.08)	0.52 (0.04)	0.47 (0.09)
		0.12 (0.01)	0.19 (0.03)	0.17 (0.02)	0.29 (0.10)	0.50 (0.05)	0.48 (0.06)
		0.12 (0.01)	0.17 (0.03)	0.17 (0.02)	0.27 (0.08)	0.48 (0.06)	0.44 (0.10)
		0.12 (0.01)	0.17 (0.03)	0.17 (0.02)	0.27 (0.08)	0.48 (0.06)	0.44 (0.10)
		1.32 (0.09)	1.20 (0.10)	1.31 (0.06)	1.27 (0.12)	1.12 (0.14)	1.02 (0.11)
		0.11 (0.01)	0.18 (0.04)	0.15 (0.02)	0.26 (0.07)	0.47 (0.04)	0.45 (0.08)
		0.10 (0.01)	0.17 (0.04)	0.14 (0.03)	0.26 (0.09)	0.43 (0.07)	0.40 (0.09)
		X	X	X	X	X	X
		0.06 (0.01)	0.07 (0.02)	0.08 (0.02)	0.13 (0.06)	0.23 (0.02)	0.22 (0.03)
	100	0.11 (0.01)	0.19 (0.02)	0.28 (0.06)	0.33 (0.05)	0.50 (0.06)	0.60 (0.06)
		0.11 (0.01)	0.19 (0.02)	0.26 (0.06)	0.31 (0.05)	0.38 (0.04)	0.38 (0.04)
		0.11 (0.01)	0.16 (0.02)	0.25 (0.07)	0.29 (0.06)	0.37 (0.04)	0.38 (0.04)
		0.11 (0.01)	0.15 (0.02)	0.24 (0.07)	0.27 (0.06)	0.33 (0.03)	0.33 (0.04)
		2.50 (0.18)	2.50 (0.25)	2.50 (0.56)	2.50 (0.35)	2.50 (0.57)	2.50 (0.72)
		1.86 (0.04)	1.66 (0.03)	1.98 (0.14)	1.85 (0.13)	1.52 (0.10)	1.39 (0.10)
		0.10 (0.01)	0.15 (0.02)	0.24 (0.06)	0.28 (0.06)	0.35 (0.04)	0.37 (0.04)
		0.10 (0.00)	0.13 (0.01)	0.23 (0.05)	0.26 (0.06)	0.32 (0.04)	0.33 (0.03)
		X	X	X	X	X	X
		0.06 (0.00)	0.06 (0.01)	0.16 (0.06)	0.17 (0.05)	0.25 (0.02)	0.22 (0.03)

Loss Values - High Setup Times, 2 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.02)	0.18 (0.07)	0.16 (0.11)	0.28 (0.18)	0.49 (0.18)	0.48 (0.17)
		0.09 (0.03)	0.17 (0.07)	0.15 (0.12)	0.25 (0.16)	0.41 (0.22)	0.34 (0.23)
		0.08 (0.03)	0.13 (0.03)	0.14 (0.10)	0.20 (0.17)	0.34 (0.21)	0.39 (0.22)
		0.06 (0.02)	0.12 (0.05)	0.14 (0.13)	0.18 (0.17)	0.33 (0.21)	0.34 (0.17)
		0.06 (0.02)	0.12 (0.05)	0.14 (0.13)	0.18 (0.17)	0.33 (0.21)	0.34 (0.17)
		0.13 (0.08)	0.15 (0.09)	0.22 (0.10)	0.20 (0.12)	0.41 (0.15)	0.29 (0.16)
		0.07 (0.01)	0.10 (0.04)	0.10 (0.10)	0.14 (0.09)	0.20 (0.14)	0.21 (0.17)
		0.07 (0.01)	0.10 (0.04)	0.10 (0.10)	0.14 (0.09)	0.20 (0.14)	0.21 (0.17)
		X	X	X	X	X	X
		0.04 (0.01)	0.06 (0.03)	0.06 (0.07)	0.09 (0.07)	0.15 (0.10)	0.14 (0.09)
	20	0.11 (0.01)	0.21 (0.04)	0.22 (0.12)	0.27 (0.09)	0.48 (0.13)	0.48 (0.19)
		0.11 (0.01)	0.20 (0.05)	0.19 (0.09)	0.22 (0.08)	0.34 (0.11)	0.32 (0.10)
		0.08 (0.01)	0.12 (0.04)	0.20 (0.10)	0.17 (0.08)	0.34 (0.10)	0.31 (0.14)
		0.07 (0.02)	0.10 (0.03)	0.17 (0.11)	0.15 (0.07)	0.30 (0.09)	0.29 (0.09)
		0.07 (0.02)	0.10 (0.03)	0.17 (0.11)	0.15 (0.07)	0.30 (0.09)	0.29 (0.09)
		0.20 (0.04)	0.20 (0.05)	0.26 (0.08)	0.23 (0.05)	0.29 (0.07)	0.25 (0.11)
		0.09 (0.02)	0.09 (0.02)	0.13 (0.07)	0.11 (0.05)	0.26 (0.05)	0.22 (0.07)
		0.09 (0.02)	0.09 (0.02)	0.13 (0.07)	0.11 (0.05)	0.26 (0.05)	0.22 (0.07)
		X	X	X	X	X	X
		0.04 (0.01)	0.04 (0.01)	0.08 (0.07)	0.06 (0.03)	0.15 (0.05)	0.12 (0.05)
	30	0.11 (0.01)	0.19 (0.05)	0.24 (0.08)	0.30 (0.11)	0.43 (0.09)	0.47 (0.19)
		0.11 (0.01)	0.18 (0.05)	0.22 (0.07)	0.26 (0.08)	0.34 (0.08)	0.42 (0.13)
		0.09 (0.02)	0.11 (0.03)	0.22 (0.08)	0.19 (0.07)	0.34 (0.06)	0.40 (0.09)
		0.09 (0.02)	0.08 (0.02)	0.16 (0.07)	0.16 (0.05)	0.28 (0.06)	0.32 (0.06)
		0.09 (0.02)	0.08 (0.02)	0.16 (0.07)	0.16 (0.05)	0.28 (0.06)	0.32 (0.06)
		0.21 (0.04)	0.18 (0.04)	0.32 (0.08)	0.28 (0.07)	0.28 (0.08)	0.32 (0.12)
		0.08 (0.01)	0.09 (0.01)	0.15 (0.06)	0.15 (0.05)	0.28 (0.04)	0.31 (0.08)
		0.08 (0.01)	0.09 (0.01)	0.15 (0.06)	0.15 (0.05)	0.28 (0.04)	0.31 (0.08)
		X	X	X	X	X	X
		0.04 (0.01)	0.04 (0.01)	0.10 (0.05)	0.10 (0.05)	0.15 (0.07)	0.14 (0.06)
	100	0.10 (0.01)	0.14 (0.02)	0.29 (0.05)	0.32 (0.05)	0.40 (0.06)	0.42 (0.07)
		0.10 (0.01)	0.13 (0.01)	0.26 (0.05)	0.30 (0.05)	0.32 (0.02)	0.36 (0.05)
		0.09 (0.00)	0.09 (0.01)	0.25 (0.05)	0.26 (0.05)	0.30 (0.01)	0.31 (0.03)
		0.35 (0.01)	0.29 (0.05)	0.52 (0.06)	0.43 (0.05)	0.25 (0.02)	0.25 (0.02)
		1.52 (0.17)	1.24 (0.17)	1.66 (0.22)	1.30 (0.22)	0.25 (0.02)	0.25 (0.02)
		0.25 (0.02)	0.22 (0.03)	0.40 (0.08)	0.38 (0.08)	0.33 (0.04)	0.31 (0.06)
		0.09 (0.01)	0.09 (0.01)	0.23 (0.05)	0.25 (0.06)	0.30 (0.02)	0.30 (0.04)
		0.09 (0.01)	0.09 (0.01)	0.23 (0.05)	0.25 (0.06)	0.30 (0.02)	0.30 (0.04)
		X	X	X	X	X	X
		0.05 (0.01)	0.05 (0.01)	0.18 (0.05)	0.20 (0.06)	0.23 (0.05)	0.20 (0.05)

Loss Values - High Setup Times, 2 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.12 (0.03)	0.26 (0.04)	0.15 (0.10)	0.38 (0.13)	0.45 (0.23)	0.54 (0.22)
		0.12 (0.03)	0.26 (0.05)	0.15 (0.08)	0.28 (0.06)	0.32 (0.25)	0.32 (0.21)
		0.10 (0.03)	0.21 (0.04)	0.17 (0.11)	0.24 (0.09)	0.33 (0.25)	0.30 (0.12)
		0.09 (0.03)	0.21 (0.03)	0.16 (0.12)	0.24 (0.09)	0.31 (0.24)	0.32 (0.15)
		0.09 (0.03)	0.21 (0.03)	0.16 (0.12)	0.24 (0.09)	0.31 (0.24)	0.32 (0.15)
		0.23 (0.05)	0.26 (0.07)	0.25 (0.11)	0.35 (0.13)	0.26 (0.18)	0.36 (0.10)
		0.09 (0.02)	0.22 (0.04)	0.15 (0.09)	0.22 (0.07)	0.27 (0.24)	0.24 (0.16)
		0.08 (0.02)	0.20 (0.06)	0.14 (0.07)	0.22 (0.05)	0.26 (0.26)	0.23 (0.15)
		X	X	X	X	X	X
		0.06 (0.01)	0.13 (0.03)	0.10 (0.07)	0.14 (0.05)	0.15 (0.12)	0.13 (0.07)
	20	0.13 (0.01)	0.30 (0.05)	0.24 (0.06)	0.42 (0.16)	0.74 (0.16)	0.72 (0.21)
		0.12 (0.01)	0.27 (0.03)	0.20 (0.05)	0.32 (0.06)	0.50 (0.15)	0.51 (0.12)
		0.11 (0.02)	0.19 (0.03)	0.20 (0.05)	0.26 (0.07)	0.47 (0.15)	0.48 (0.12)
		0.13 (0.04)	0.19 (0.03)	0.19 (0.04)	0.25 (0.07)	0.46 (0.12)	0.43 (0.11)
		0.13 (0.04)	0.19 (0.03)	0.19 (0.04)	0.25 (0.07)	0.46 (0.12)	0.43 (0.11)
		0.41 (0.06)	0.40 (0.07)	0.51 (0.08)	0.52 (0.07)	0.62 (0.19)	0.61 (0.14)
		0.11 (0.01)	0.15 (0.02)	0.17 (0.03)	0.21 (0.07)	0.41 (0.13)	0.43 (0.14)
		0.10 (0.01)	0.15 (0.02)	0.16 (0.03)	0.21 (0.07)	0.35 (0.10)	0.38 (0.09)
		X	X	X	X	X	X
		0.06 (0.01)	0.08 (0.02)	0.10 (0.03)	0.11 (0.05)	0.19 (0.06)	0.19 (0.05)
	30	0.13 (0.02)	0.26 (0.05)	0.31 (0.10)	0.40 (0.12)	0.67 (0.09)	0.72 (0.14)
		0.12 (0.01)	0.24 (0.05)	0.28 (0.10)	0.30 (0.07)	0.47 (0.06)	0.48 (0.13)
		0.10 (0.01)	0.17 (0.05)	0.27 (0.09)	0.22 (0.06)	0.46 (0.07)	0.43 (0.09)
		0.11 (0.02)	0.17 (0.04)	0.25 (0.08)	0.21 (0.06)	0.44 (0.07)	0.37 (0.07)
		0.11 (0.02)	0.17 (0.04)	0.25 (0.08)	0.21 (0.06)	0.44 (0.07)	0.37 (0.07)
		0.50 (0.05)	0.54 (0.05)	0.63 (0.09)	0.55 (0.09)	0.56 (0.11)	0.55 (0.13)
		0.10 (0.02)	0.15 (0.03)	0.23 (0.09)	0.19 (0.05)	0.38 (0.09)	0.34 (0.10)
		0.09 (0.02)	0.14 (0.04)	0.22 (0.10)	0.18 (0.04)	0.35 (0.06)	0.35 (0.08)
		X	X	X	X	X	X
		0.06 (0.01)	0.06 (0.02)	0.16 (0.09)	0.10 (0.04)	0.20 (0.02)	0.19 (0.02)
	100	0.12 (0.01)	0.19 (0.02)	0.29 (0.06)	0.40 (0.07)	0.48 (0.04)	0.54 (0.08)
		0.12 (0.01)	0.17 (0.02)	0.26 (0.05)	0.33 (0.06)	0.34 (0.03)	0.37 (0.04)
		0.10 (0.01)	0.14 (0.02)	0.26 (0.05)	0.30 (0.06)	0.35 (0.04)	0.34 (0.03)
		0.35 (0.02)	0.27 (0.05)	0.49 (0.06)	0.40 (0.09)	0.29 (0.01)	0.30 (0.03)
		2.50 (0.20)	2.50 (0.35)	2.50 (0.40)	2.50 (0.52)	0.39 (0.19)	0.42 (0.38)
		0.65 (0.04)	0.60 (0.04)	0.83 (0.10)	0.79 (0.09)	0.65 (0.06)	0.61 (0.09)
		0.10 (0.00)	0.11 (0.02)	0.24 (0.05)	0.27 (0.06)	0.32 (0.02)	0.33 (0.04)
		0.10 (0.00)	0.11 (0.02)	0.23 (0.04)	0.26 (0.05)	0.30 (0.03)	0.30 (0.03)
		X	X	X	X	X	X
		0.07 (0.01)	0.07 (0.01)	0.19 (0.05)	0.21 (0.05)	0.25 (0.02)	0.25 (0.01)

Loss Values - High Setup Times, 2 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.02)	0.24 (0.07)	0.24 (0.06)	0.31 (0.09)	0.43 (0.14)	0.44 (0.21)
		0.11 (0.02)	0.22 (0.05)	0.21 (0.06)	0.26 (0.06)	0.27 (0.08)	0.24 (0.09)
		0.10 (0.02)	0.20 (0.07)	0.21 (0.06)	0.24 (0.06)	0.26 (0.08)	0.22 (0.10)
		0.10 (0.02)	0.21 (0.06)	0.21 (0.05)	0.25 (0.06)	0.25 (0.11)	0.23 (0.08)
		0.10 (0.02)	0.21 (0.06)	0.21 (0.05)	0.25 (0.06)	0.25 (0.11)	0.23 (0.08)
		0.25 (0.05)	0.31 (0.09)	0.30 (0.07)	0.32 (0.10)	0.25 (0.10)	0.22 (0.09)
		0.10 (0.02)	0.18 (0.07)	0.20 (0.05)	0.24 (0.08)	0.26 (0.08)	0.21 (0.08)
		0.10 (0.02)	0.18 (0.05)	0.20 (0.05)	0.22 (0.06)	0.24 (0.09)	0.25 (0.12)
		X	X	X	X	X	X
		0.07 (0.02)	0.11 (0.05)	0.16 (0.04)	0.15 (0.05)	0.13 (0.06)	0.12 (0.06)
	20	0.15 (0.01)	0.37 (0.08)	0.32 (0.09)	0.51 (0.08)	0.91 (0.22)	1.00 (0.11)
		0.14 (0.01)	0.33 (0.05)	0.27 (0.06)	0.44 (0.08)	0.60 (0.18)	0.66 (0.14)
		0.13 (0.02)	0.26 (0.04)	0.26 (0.06)	0.37 (0.06)	0.61 (0.12)	0.66 (0.19)
		0.14 (0.03)	0.26 (0.05)	0.27 (0.06)	0.37 (0.07)	0.54 (0.10)	0.61 (0.12)
		0.14 (0.03)	0.26 (0.05)	0.27 (0.06)	0.37 (0.07)	0.54 (0.10)	0.61 (0.12)
		0.73 (0.09)	0.79 (0.07)	0.87 (0.15)	0.92 (0.13)	0.92 (0.11)	0.88 (0.14)
		0.13 (0.01)	0.24 (0.04)	0.22 (0.05)	0.32 (0.05)	0.54 (0.10)	0.59 (0.17)
		0.13 (0.01)	0.23 (0.06)	0.23 (0.05)	0.31 (0.06)	0.49 (0.11)	0.53 (0.12)
		X	X	X	X	X	X
		0.08 (0.01)	0.15 (0.03)	0.16 (0.05)	0.20 (0.03)	0.27 (0.06)	0.30 (0.06)
	30	0.15 (0.02)	0.34 (0.04)	0.33 (0.09)	0.48 (0.08)	0.87 (0.14)	0.99 (0.09)
		0.15 (0.02)	0.33 (0.04)	0.28 (0.09)	0.37 (0.07)	0.58 (0.05)	0.66 (0.08)
		0.14 (0.01)	0.25 (0.03)	0.28 (0.09)	0.33 (0.04)	0.58 (0.10)	0.60 (0.08)
		0.15 (0.01)	0.25 (0.03)	0.29 (0.10)	0.31 (0.05)	0.56 (0.09)	0.60 (0.07)
		0.15 (0.01)	0.25 (0.03)	0.29 (0.10)	0.31 (0.05)	0.56 (0.09)	0.60 (0.07)
		0.96 (0.05)	0.90 (0.07)	0.99 (0.14)	1.03 (0.10)	1.10 (0.13)	1.06 (0.21)
		0.13 (0.01)	0.22 (0.02)	0.25 (0.08)	0.29 (0.05)	0.52 (0.06)	0.52 (0.08)
		0.12 (0.01)	0.22 (0.02)	0.25 (0.09)	0.27 (0.04)	0.48 (0.09)	0.49 (0.05)
		X	X	X	X	X	X
		0.08 (0.01)	0.13 (0.01)	0.18 (0.07)	0.17 (0.04)	0.27 (0.05)	0.27 (0.03)
	100	0.13 (0.01)	0.25 (0.02)	0.36 (0.04)	0.45 (0.05)	0.68 (0.10)	0.76 (0.08)
		0.13 (0.00)	0.23 (0.02)	0.32 (0.04)	0.38 (0.05)	0.45 (0.07)	0.43 (0.03)
		0.12 (0.01)	0.18 (0.02)	0.32 (0.04)	0.34 (0.03)	0.42 (0.05)	0.42 (0.05)
		0.33 (0.03)	0.28 (0.05)	0.50 (0.06)	0.38 (0.06)	0.39 (0.05)	0.37 (0.05)
		2.50 (0.22)	2.50 (0.42)	2.50 (0.75)	2.50 (0.59)	1.22 (0.86)	1.18 (0.44)
		1.40 (0.04)	1.26 (0.05)	1.64 (0.12)	1.54 (0.10)	1.29 (0.14)	1.16 (0.07)
		0.11 (0.01)	0.16 (0.02)	0.29 (0.04)	0.31 (0.03)	0.41 (0.06)	0.38 (0.04)
		0.11 (0.01)	0.14 (0.01)	0.28 (0.04)	0.29 (0.05)	0.36 (0.04)	0.37 (0.05)
		X	X	X	X	X	X
		0.09 (0.00)	0.09 (0.01)	0.24 (0.04)	0.22 (0.03)	0.26 (0.02)	0.26 (0.02)

Loss Values - High Setup Times, 10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.16 (0.02)	0.43 (0.09)	0.27 (0.07)	0.42 (0.06)	0.31 (0.24)	0.33 (0.17)
		0.15 (0.03)	0.43 (0.09)	0.27 (0.05)	0.39 (0.10)	0.30 (0.24)	0.37 (0.16)
		0.13 (0.03)	0.31 (0.07)	0.29 (0.08)	0.34 (0.06)	0.25 (0.22)	0.27 (0.09)
		0.08 (0.01)	0.28 (0.05)	0.24 (0.06)	0.30 (0.06)	0.19 (0.19)	0.20 (0.11)
		0.08 (0.01)	0.28 (0.05)	0.24 (0.06)	0.30 (0.06)	0.19 (0.19)	0.20 (0.11)
		0.28 (0.08)	0.28 (0.04)	0.32 (0.14)	0.27 (0.09)	0.12 (0.16)	0.16 (0.11)
		0.09 (0.02)	0.21 (0.06)	0.22 (0.07)	0.21 (0.08)	0.11 (0.10)	0.10 (0.08)
		0.09 (0.02)	0.21 (0.06)	0.22 (0.07)	0.21 (0.08)	0.11 (0.10)	0.10 (0.08)
		X	X	X	X	X	X
		0.10 (0.01)	0.18 (0.05)	0.21 (0.06)	0.14 (0.08)	0.06 (0.09)	0.01 (0.00)
	30	0.17 (0.02)	0.49 (0.12)	0.35 (0.10)	0.62 (0.11)	0.63 (0.14)	0.59 (0.15)
		0.16 (0.03)	0.45 (0.09)	0.32 (0.09)	0.57 (0.15)	0.55 (0.10)	0.49 (0.14)
		0.12 (0.02)	0.33 (0.08)	0.35 (0.10)	0.45 (0.11)	0.64 (0.16)	0.36 (0.10)
		0.09 (0.02)	0.29 (0.06)	0.30 (0.08)	0.39 (0.11)	0.52 (0.10)	0.34 (0.12)
		0.09 (0.02)	0.29 (0.06)	0.30 (0.08)	0.39 (0.11)	0.52 (0.10)	0.34 (0.12)
		0.26 (0.01)	0.28 (0.05)	0.39 (0.10)	0.35 (0.11)	0.45 (0.10)	0.22 (0.11)
		0.11 (0.01)	0.20 (0.04)	0.28 (0.06)	0.29 (0.08)	0.36 (0.14)	0.17 (0.06)
		0.11 (0.01)	0.20 (0.04)	0.28 (0.06)	0.29 (0.08)	0.36 (0.14)	0.17 (0.06)
		X	X	X	X	X	X
		0.11 (0.01)	0.20 (0.04)	0.25 (0.07)	0.26 (0.09)	0.31 (0.11)	0.07 (0.03)
	100	0.14 (0.02)	0.33 (0.03)	0.40 (0.07)	0.57 (0.10)	0.64 (0.06)	0.69 (0.09)
		0.13 (0.01)	0.29 (0.02)	0.37 (0.06)	0.51 (0.10)	0.52 (0.11)	0.58 (0.11)
		0.11 (0.01)	0.19 (0.02)	0.36 (0.08)	0.40 (0.06)	0.46 (0.08)	0.51 (0.12)
		0.11 (0.02)	0.13 (0.02)	0.28 (0.07)	0.31 (0.05)	0.41 (0.06)	0.39 (0.09)
		0.11 (0.02)	0.13 (0.02)	0.28 (0.07)	0.31 (0.05)	0.41 (0.06)	0.39 (0.09)
		0.21 (0.02)	0.24 (0.03)	0.40 (0.07)	0.40 (0.09)	0.42 (0.09)	0.40 (0.07)
		0.09 (0.02)	0.15 (0.03)	0.27 (0.05)	0.30 (0.08)	0.35 (0.05)	0.38 (0.09)
		0.09 (0.02)	0.15 (0.03)	0.27 (0.05)	0.30 (0.08)	0.35 (0.05)	0.38 (0.09)
		X	X	X	X	X	X
		0.11 (0.02)	0.14 (0.03)	0.27 (0.05)	0.29 (0.06)	0.24 (0.04)	0.21 (0.03)

Loss Values - High Setup Times, 10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.14 (0.02)	0.27 (0.07)	0.25 (0.05)	0.36 (0.11)	0.33 (0.15)	0.31 (0.10)
		0.13 (0.02)	0.25 (0.06)	0.24 (0.06)	0.29 (0.13)	0.26 (0.11)	0.16 (0.11)
		0.13 (0.03)	0.18 (0.05)	0.28 (0.07)	0.22 (0.05)	0.22 (0.13)	0.11 (0.08)
		0.09 (0.01)	0.17 (0.03)	0.24 (0.08)	0.19 (0.05)	0.20 (0.12)	0.08 (0.08)
		0.09 (0.01)	0.17 (0.03)	0.24 (0.08)	0.19 (0.05)	0.20 (0.12)	0.08 (0.08)
		0.24 (0.01)	0.24 (0.05)	0.32 (0.05)	0.24 (0.08)	0.18 (0.11)	0.15 (0.05)
		0.12 (0.02)	0.14 (0.04)	0.24 (0.06)	0.19 (0.03)	0.17 (0.08)	0.06 (0.05)
		0.12 (0.03)	0.12 (0.06)	0.25 (0.05)	0.19 (0.07)	0.13 (0.07)	0.06 (0.04)
		X	X	X	X	X	X
		0.10 (0.01)	0.06 (0.05)	0.16 (0.04)	0.08 (0.06)	0.06 (0.04)	0.02 (0.01)
	30	0.18 (0.02)	0.49 (0.06)	0.34 (0.13)	0.60 (0.07)	0.48 (0.13)	0.50 (0.08)
		0.17 (0.02)	0.41 (0.04)	0.33 (0.12)	0.52 (0.09)	0.37 (0.12)	0.36 (0.14)
		0.13 (0.01)	0.31 (0.05)	0.33 (0.13)	0.40 (0.09)	0.37 (0.07)	0.24 (0.07)
		0.10 (0.02)	0.29 (0.06)	0.29 (0.13)	0.38 (0.08)	0.33 (0.07)	0.22 (0.07)
		0.10 (0.02)	0.29 (0.06)	0.29 (0.13)	0.38 (0.08)	0.33 (0.07)	0.22 (0.07)
		0.32 (0.03)	0.38 (0.03)	0.46 (0.11)	0.44 (0.07)	0.28 (0.07)	0.27 (0.06)
		0.11 (0.01)	0.25 (0.04)	0.27 (0.09)	0.33 (0.10)	0.23 (0.10)	0.16 (0.06)
		0.13 (0.02)	0.24 (0.03)	0.28 (0.10)	0.32 (0.07)	0.19 (0.06)	0.13 (0.05)
		X	X	X	X	X	X
		0.11 (0.02)	0.17 (0.03)	0.22 (0.08)	0.20 (0.05)	0.09 (0.04)	0.05 (0.03)
	100	0.17 (0.01)	0.41 (0.04)	0.47 (0.08)	0.67 (0.11)	0.93 (0.09)	1.05 (0.14)
		0.16 (0.01)	0.36 (0.06)	0.42 (0.07)	0.55 (0.11)	0.72 (0.13)	0.75 (0.13)
		0.13 (0.01)	0.21 (0.04)	0.41 (0.08)	0.45 (0.09)	0.64 (0.10)	0.64 (0.14)
		0.16 (0.02)	0.20 (0.02)	0.38 (0.06)	0.37 (0.10)	0.60 (0.08)	0.60 (0.13)
		0.16 (0.02)	0.20 (0.02)	0.38 (0.06)	0.37 (0.10)	0.60 (0.08)	0.60 (0.13)
		0.43 (0.03)	0.47 (0.03)	0.71 (0.09)	0.68 (0.10)	0.80 (0.05)	0.79 (0.12)
		0.12 (0.01)	0.16 (0.04)	0.32 (0.05)	0.37 (0.12)	0.51 (0.10)	0.56 (0.16)
		0.12 (0.01)	0.16 (0.03)	0.32 (0.05)	0.36 (0.11)	0.43 (0.10)	0.43 (0.08)
		X	X	X	X	X	X
		0.14 (0.02)	0.17 (0.04)	0.33 (0.06)	0.29 (0.10)	0.30 (0.03)	0.31 (0.08)

Loss Values - High Setup Times, 10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.14 (0.02)	0.20 (0.06)	0.20 (0.04)	0.25 (0.11)	0.17 (0.08)	0.17 (0.11)
		0.14 (0.01)	0.16 (0.06)	0.21 (0.05)	0.18 (0.06)	0.17 (0.07)	0.16 (0.10)
		0.11 (0.02)	0.11 (0.02)	0.23 (0.03)	0.16 (0.03)	0.19 (0.06)	0.07 (0.05)
		0.09 (0.01)	0.10 (0.03)	0.20 (0.02)	0.15 (0.04)	0.11 (0.08)	0.06 (0.03)
		0.09 (0.01)	0.10 (0.03)	0.20 (0.02)	0.15 (0.04)	0.11 (0.08)	0.06 (0.03)
		0.25 (0.02)	0.22 (0.03)	0.26 (0.05)	0.26 (0.08)	0.16 (0.09)	0.26 (0.14)
		0.11 (0.02)	0.08 (0.03)	0.20 (0.03)	0.13 (0.05)	0.11 (0.05)	0.06 (0.02)
		0.12 (0.01)	0.09 (0.04)	0.17 (0.05)	0.14 (0.06)	0.08 (0.03)	0.05 (0.02)
		X	X	X	X	X	X
		0.08 (0.01)	0.02 (0.02)	0.11 (0.04)	0.04 (0.02)	0.03 (0.01)	0.02 (0.01)
	30	0.17 (0.01)	0.31 (0.08)	0.32 (0.05)	0.35 (0.03)	0.38 (0.12)	0.38 (0.14)
		0.16 (0.01)	0.26 (0.08)	0.30 (0.05)	0.27 (0.06)	0.31 (0.09)	0.27 (0.06)
		0.13 (0.01)	0.18 (0.04)	0.32 (0.04)	0.24 (0.05)	0.27 (0.06)	0.23 (0.05)
		0.12 (0.01)	0.19 (0.04)	0.30 (0.05)	0.21 (0.04)	0.24 (0.07)	0.21 (0.03)
		0.12 (0.01)	0.19 (0.04)	0.30 (0.05)	0.21 (0.04)	0.24 (0.07)	0.21 (0.03)
		0.36 (0.02)	0.35 (0.06)	0.45 (0.04)	0.33 (0.03)	0.22 (0.11)	0.34 (0.11)
		0.13 (0.01)	0.15 (0.05)	0.31 (0.04)	0.17 (0.04)	0.19 (0.06)	0.21 (0.09)
		0.14 (0.02)	0.16 (0.04)	0.28 (0.05)	0.23 (0.04)	0.19 (0.07)	0.17 (0.04)
		X	X	X	X	X	X
		0.12 (0.01)	0.08 (0.04)	0.20 (0.03)	0.07 (0.02)	0.06 (0.02)	0.06 (0.03)
	100	0.18 (0.01)	0.45 (0.03)	0.48 (0.06)	0.80 (0.07)	1.13 (0.10)	1.26 (0.17)
		0.18 (0.01)	0.39 (0.03)	0.42 (0.06)	0.64 (0.07)	0.81 (0.14)	0.82 (0.13)
		0.13 (0.01)	0.26 (0.05)	0.42 (0.07)	0.52 (0.07)	0.78 (0.15)	0.70 (0.14)
		0.17 (0.01)	0.28 (0.03)	0.39 (0.06)	0.48 (0.07)	0.72 (0.08)	0.69 (0.11)
		0.99 (0.15)	0.94 (0.18)	0.50 (0.18)	0.48 (0.07)	0.72 (0.08)	0.69 (0.11)
		0.69 (0.04)	0.76 (0.03)	0.98 (0.07)	1.09 (0.08)	1.13 (0.12)	1.05 (0.09)
		0.13 (0.01)	0.24 (0.04)	0.36 (0.06)	0.45 (0.05)	0.69 (0.09)	0.68 (0.12)
		0.13 (0.00)	0.23 (0.03)	0.37 (0.05)	0.46 (0.08)	0.54 (0.06)	0.57 (0.09)
		X	X	X	X	X	X
		0.14 (0.01)	0.20 (0.02)	0.34 (0.05)	0.39 (0.06)	0.41 (0.07)	0.31 (0.08)

Loss Values - High Setup Times, 1-4 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.04)	0.19 (0.07)	0.13 (0.10)	0.22 (0.15)	0.35 (0.22)	0.19 (0.13)
		0.10 (0.04)	0.18 (0.10)	0.11 (0.10)	0.22 (0.15)	0.23 (0.18)	0.17 (0.14)
		0.08 (0.03)	0.14 (0.07)	0.11 (0.09)	0.22 (0.16)	0.20 (0.17)	0.13 (0.08)
		0.07 (0.03)	0.10 (0.07)	0.09 (0.08)	0.19 (0.16)	0.18 (0.17)	0.14 (0.07)
		0.07 (0.03)	0.10 (0.07)	0.09 (0.08)	0.19 (0.16)	0.18 (0.17)	0.14 (0.07)
		0.06 (0.03)	0.12 (0.09)	0.09 (0.12)	0.17 (0.14)	0.18 (0.16)	0.14 (0.11)
		0.06 (0.03)	0.10 (0.07)	0.10 (0.09)	0.12 (0.10)	0.16 (0.12)	0.11 (0.07)
		0.06 (0.03)	0.10 (0.07)	0.10 (0.09)	0.12 (0.10)	0.16 (0.12)	0.11 (0.07)
		X	X	X	X	X	X
		0.03 (0.02)	0.03 (0.03)	0.05 (0.07)	0.06 (0.05)	0.11 (0.10)	0.06 (0.06)
	20	0.08 (0.02)	0.17 (0.06)	0.21 (0.06)	0.37 (0.15)	0.46 (0.13)	0.40 (0.23)
		0.08 (0.02)	0.16 (0.07)	0.20 (0.07)	0.27 (0.09)	0.36 (0.13)	0.30 (0.11)
		0.08 (0.01)	0.12 (0.06)	0.20 (0.07)	0.24 (0.11)	0.35 (0.17)	0.30 (0.10)
		0.03 (0.01)	0.08 (0.05)	0.16 (0.07)	0.21 (0.09)	0.26 (0.11)	0.28 (0.08)
		0.03 (0.01)	0.08 (0.05)	0.16 (0.07)	0.21 (0.09)	0.26 (0.11)	0.28 (0.08)
		0.03 (0.03)	0.10 (0.06)	0.14 (0.07)	0.20 (0.09)	0.27 (0.12)	0.26 (0.08)
		0.07 (0.01)	0.10 (0.03)	0.16 (0.05)	0.19 (0.08)	0.26 (0.05)	0.26 (0.07)
		0.07 (0.01)	0.10 (0.03)	0.16 (0.05)	0.19 (0.08)	0.26 (0.05)	0.26 (0.07)
		X	X	X	X	X	X
		0.02 (0.01)	0.03 (0.02)	0.10 (0.05)	0.12 (0.06)	0.11 (0.06)	0.11 (0.07)
	30	0.09 (0.01)	0.14 (0.05)	0.24 (0.11)	0.28 (0.07)	0.50 (0.14)	0.42 (0.10)
		0.08 (0.01)	0.12 (0.05)	0.24 (0.11)	0.26 (0.07)	0.40 (0.11)	0.33 (0.08)
		0.08 (0.01)	0.11 (0.05)	0.24 (0.11)	0.23 (0.09)	0.38 (0.11)	0.30 (0.08)
		0.03 (0.01)	0.08 (0.04)	0.16 (0.10)	0.19 (0.08)	0.34 (0.10)	0.26 (0.05)
		0.03 (0.01)	0.08 (0.04)	0.16 (0.10)	0.19 (0.08)	0.34 (0.10)	0.26 (0.05)
		0.03 (0.01)	0.07 (0.04)	0.16 (0.11)	0.17 (0.07)	0.31 (0.09)	0.24 (0.05)
		0.07 (0.01)	0.09 (0.03)	0.20 (0.10)	0.17 (0.06)	0.29 (0.04)	0.27 (0.04)
		0.07 (0.01)	0.09 (0.03)	0.20 (0.10)	0.17 (0.06)	0.29 (0.04)	0.27 (0.04)
		X	X	X	X	X	X
		0.02 (0.01)	0.04 (0.01)	0.12 (0.08)	0.10 (0.05)	0.13 (0.06)	0.12 (0.08)
	100	0.09 (0.01)	0.10 (0.02)	0.29 (0.04)	0.28 (0.06)	0.31 (0.04)	0.39 (0.07)
		0.09 (0.01)	0.09 (0.01)	0.28 (0.04)	0.27 (0.07)	0.31 (0.04)	0.34 (0.05)
		0.09 (0.01)	0.09 (0.01)	0.27 (0.04)	0.25 (0.07)	0.32 (0.05)	0.31 (0.04)
		0.16 (0.14)	0.19 (0.14)	0.28 (0.12)	0.26 (0.12)	0.25 (0.05)	0.26 (0.04)
		0.77 (0.57)	0.71 (0.34)	0.86 (0.34)	0.97 (0.25)	0.27 (0.09)	0.26 (0.04)
		0.06 (0.04)	0.04 (0.03)	0.24 (0.08)	0.20 (0.06)	0.24 (0.04)	0.26 (0.05)
		0.08 (0.01)	0.08 (0.01)	0.26 (0.03)	0.24 (0.06)	0.29 (0.01)	0.31 (0.04)
		0.08 (0.01)	0.08 (0.01)	0.26 (0.03)	0.24 (0.06)	0.29 (0.01)	0.31 (0.04)
		X	X	X	X	X	X
		0.06 (0.01)	0.05 (0.01)	0.21 (0.03)	0.19 (0.05)	0.15 (0.08)	0.11 (0.09)

Loss Values - High Setup Times, 1-4 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.03)	0.16 (0.10)	0.18 (0.10)	0.21 (0.10)	0.36 (0.30)	0.35 (0.13)
		0.08 (0.02)	0.14 (0.07)	0.13 (0.09)	0.15 (0.08)	0.20 (0.13)	0.25 (0.11)
		0.06 (0.03)	0.12 (0.07)	0.09 (0.06)	0.09 (0.06)	0.21 (0.16)	0.19 (0.07)
		0.06 (0.03)	0.12 (0.07)	0.10 (0.07)	0.09 (0.05)	0.20 (0.20)	0.20 (0.10)
		0.06 (0.03)	0.12 (0.07)	0.10 (0.07)	0.09 (0.05)	0.20 (0.20)	0.20 (0.10)
		0.09 (0.08)	0.10 (0.09)	0.13 (0.08)	0.14 (0.11)	0.15 (0.14)	0.23 (0.12)
		0.06 (0.03)	0.07 (0.04)	0.10 (0.08)	0.11 (0.07)	0.16 (0.13)	0.16 (0.07)
		0.05 (0.02)	0.10 (0.06)	0.08 (0.08)	0.11 (0.09)	0.20 (0.14)	0.18 (0.08)
		X	X	X	X	X	X
		0.02 (0.02)	0.03 (0.03)	0.05 (0.06)	0.04 (0.04)	0.09 (0.10)	0.12 (0.05)
	20	0.10 (0.02)	0.12 (0.03)	0.26 (0.06)	0.28 (0.14)	0.53 (0.08)	0.60 (0.25)
		0.10 (0.02)	0.12 (0.02)	0.22 (0.07)	0.24 (0.14)	0.34 (0.11)	0.43 (0.21)
		0.08 (0.02)	0.10 (0.03)	0.21 (0.09)	0.23 (0.15)	0.39 (0.10)	0.39 (0.14)
		0.06 (0.04)	0.07 (0.03)	0.16 (0.07)	0.18 (0.13)	0.34 (0.12)	0.34 (0.15)
		0.06 (0.04)	0.07 (0.03)	0.16 (0.07)	0.18 (0.13)	0.34 (0.12)	0.34 (0.15)
		0.15 (0.11)	0.17 (0.17)	0.28 (0.15)	0.23 (0.16)	0.34 (0.09)	0.50 (0.21)
		0.08 (0.02)	0.08 (0.02)	0.18 (0.08)	0.18 (0.11)	0.33 (0.10)	0.36 (0.13)
		0.07 (0.02)	0.08 (0.02)	0.18 (0.08)	0.17 (0.09)	0.31 (0.05)	0.31 (0.11)
		X	X	X	X	X	X
		0.03 (0.02)	0.02 (0.01)	0.10 (0.05)	0.10 (0.08)	0.17 (0.04)	0.17 (0.07)
	30	0.10 (0.02)	0.19 (0.07)	0.22 (0.09)	0.24 (0.16)	0.52 (0.16)	0.47 (0.20)
		0.09 (0.02)	0.16 (0.06)	0.19 (0.07)	0.21 (0.10)	0.38 (0.11)	0.41 (0.21)
		0.08 (0.01)	0.13 (0.03)	0.18 (0.08)	0.20 (0.08)	0.39 (0.14)	0.38 (0.13)
		0.06 (0.02)	0.10 (0.04)	0.14 (0.07)	0.15 (0.09)	0.34 (0.13)	0.33 (0.14)
		0.06 (0.02)	0.10 (0.04)	0.14 (0.07)	0.15 (0.09)	0.34 (0.13)	0.33 (0.14)
		0.21 (0.14)	0.19 (0.10)	0.26 (0.12)	0.25 (0.13)	0.40 (0.10)	0.42 (0.25)
		0.07 (0.01)	0.12 (0.05)	0.17 (0.06)	0.14 (0.06)	0.31 (0.08)	0.35 (0.14)
		0.08 (0.01)	0.10 (0.04)	0.16 (0.06)	0.14 (0.07)	0.28 (0.03)	0.31 (0.06)
		X	X	X	X	X	X
		0.03 (0.01)	0.03 (0.01)	0.09 (0.06)	0.07 (0.06)	0.17 (0.07)	0.15 (0.09)
	100	0.09 (0.01)	0.13 (0.05)	0.25 (0.06)	0.31 (0.04)	0.35 (0.04)	0.35 (0.04)
		0.09 (0.01)	0.12 (0.03)	0.23 (0.06)	0.28 (0.04)	0.31 (0.03)	0.30 (0.03)
		0.08 (0.01)	0.10 (0.01)	0.24 (0.07)	0.27 (0.03)	0.30 (0.02)	0.30 (0.02)
		0.13 (0.13)	0.26 (0.14)	0.21 (0.07)	0.35 (0.20)	0.25 (0.03)	0.26 (0.05)
		2.66 (0.68)	2.54 (1.32)	2.64 (0.74)	2.41 (1.16)	0.58 (0.36)	0.88 (0.52)
		0.25 (0.13)	0.30 (0.12)	0.47 (0.23)	0.33 (0.15)	0.38 (0.13)	0.34 (0.12)
		0.08 (0.01)	0.10 (0.01)	0.22 (0.07)	0.26 (0.04)	0.29 (0.02)	0.29 (0.02)
		0.08 (0.01)	0.09 (0.02)	0.22 (0.07)	0.26 (0.03)	0.29 (0.01)	0.28 (0.02)
		X	X	X	X	X	X
		0.05 (0.02)	0.05 (0.01)	0.17 (0.06)	0.22 (0.03)	0.20 (0.08)	0.18 (0.07)

Loss Values - High Setup Times, 1-4 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.09 (0.01)	0.18 (0.06)	0.20 (0.12)	0.25 (0.12)	0.39 (0.20)	0.38 (0.15)
		0.08 (0.01)	0.18 (0.05)	0.14 (0.07)	0.22 (0.12)	0.20 (0.09)	0.35 (0.13)
		0.06 (0.01)	0.13 (0.06)	0.12 (0.09)	0.20 (0.12)	0.25 (0.13)	0.31 (0.10)
		0.06 (0.01)	0.11 (0.06)	0.11 (0.06)	0.19 (0.09)	0.23 (0.12)	0.26 (0.10)
		0.06 (0.01)	0.11 (0.06)	0.11 (0.06)	0.19 (0.09)	0.23 (0.12)	0.26 (0.10)
		0.16 (0.07)	0.23 (0.11)	0.16 (0.14)	0.22 (0.09)	0.27 (0.10)	0.32 (0.11)
		0.05 (0.01)	0.11 (0.04)	0.11 (0.07)	0.19 (0.09)	0.24 (0.13)	0.31 (0.10)
		0.06 (0.02)	0.10 (0.06)	0.08 (0.06)	0.18 (0.09)	0.23 (0.11)	0.28 (0.09)
		X	X	X	X	X	X
		0.03 (0.01)	0.06 (0.04)	0.06 (0.06)	0.09 (0.06)	0.11 (0.09)	0.15 (0.07)
	20	0.11 (0.02)	0.24 (0.07)	0.22 (0.06)	0.33 (0.12)	0.62 (0.19)	0.65 (0.19)
		0.11 (0.02)	0.21 (0.07)	0.19 (0.07)	0.27 (0.12)	0.44 (0.12)	0.42 (0.17)
		0.09 (0.01)	0.17 (0.07)	0.18 (0.06)	0.26 (0.07)	0.40 (0.07)	0.40 (0.17)
		0.09 (0.01)	0.15 (0.06)	0.17 (0.07)	0.22 (0.06)	0.38 (0.10)	0.36 (0.17)
		0.09 (0.01)	0.15 (0.06)	0.17 (0.07)	0.22 (0.06)	0.38 (0.10)	0.36 (0.17)
		0.51 (0.13)	0.46 (0.17)	0.47 (0.22)	0.63 (0.14)	0.57 (0.19)	0.64 (0.22)
		0.09 (0.01)	0.15 (0.06)	0.17 (0.05)	0.22 (0.09)	0.36 (0.07)	0.37 (0.14)
		0.09 (0.02)	0.13 (0.05)	0.16 (0.05)	0.19 (0.08)	0.34 (0.06)	0.34 (0.13)
		X	X	X	X	X	X
		0.04 (0.01)	0.05 (0.04)	0.08 (0.05)	0.10 (0.06)	0.13 (0.05)	0.18 (0.06)
	30	0.09 (0.01)	0.17 (0.02)	0.24 (0.09)	0.31 (0.13)	0.56 (0.18)	0.57 (0.14)
		0.09 (0.01)	0.16 (0.03)	0.22 (0.09)	0.28 (0.11)	0.44 (0.12)	0.40 (0.09)
		0.09 (0.01)	0.13 (0.02)	0.21 (0.08)	0.26 (0.08)	0.43 (0.09)	0.35 (0.08)
		0.06 (0.03)	0.11 (0.03)	0.17 (0.08)	0.22 (0.08)	0.37 (0.12)	0.32 (0.07)
		0.06 (0.03)	0.11 (0.03)	0.17 (0.08)	0.22 (0.08)	0.37 (0.12)	0.32 (0.07)
		0.48 (0.26)	0.59 (0.13)	0.58 (0.21)	0.67 (0.17)	0.64 (0.19)	0.66 (0.11)
		0.09 (0.02)	0.12 (0.04)	0.19 (0.08)	0.21 (0.07)	0.38 (0.11)	0.36 (0.08)
		0.08 (0.01)	0.10 (0.02)	0.16 (0.07)	0.20 (0.09)	0.34 (0.09)	0.31 (0.09)
		X	X	X	X	X	X
		0.03 (0.01)	0.03 (0.02)	0.09 (0.06)	0.10 (0.06)	0.19 (0.05)	0.17 (0.05)
	100	0.09 (0.01)	0.13 (0.03)	0.28 (0.05)	0.32 (0.09)	0.43 (0.05)	0.42 (0.08)
		0.09 (0.01)	0.13 (0.04)	0.26 (0.04)	0.29 (0.07)	0.34 (0.03)	0.35 (0.05)
		0.09 (0.01)	0.12 (0.02)	0.27 (0.05)	0.28 (0.08)	0.33 (0.04)	0.33 (0.04)
		0.17 (0.12)	0.18 (0.13)	0.37 (0.16)	0.34 (0.21)	0.32 (0.09)	0.31 (0.06)
		5.28 (0.58)	5.85 (1.42)	6.15 (0.83)	6.48 (1.74)	3.22 (2.07)	2.80 (1.23)
		0.74 (0.38)	0.72 (0.34)	0.87 (0.23)	1.09 (0.39)	0.77 (0.21)	0.64 (0.22)
		0.09 (0.01)	0.11 (0.02)	0.26 (0.04)	0.26 (0.07)	0.32 (0.04)	0.33 (0.03)
		0.09 (0.01)	0.10 (0.02)	0.25 (0.04)	0.25 (0.06)	0.30 (0.02)	0.31 (0.02)
		X	X	X	X	X	X
		0.06 (0.02)	0.06 (0.01)	0.20 (0.05)	0.18 (0.06)	0.22 (0.05)	0.20 (0.08)

Loss Values - High Setup Times, 1-10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.15 (0.11)	0.27 (0.14)	0.17 (0.13)	0.30 (0.18)	0.27 (0.17)	0.32 (0.24)
		0.13 (0.10)	0.20 (0.09)	0.16 (0.13)	0.26 (0.20)	0.27 (0.17)	0.30 (0.24)
		0.12 (0.08)	0.16 (0.09)	0.16 (0.13)	0.17 (0.11)	0.24 (0.17)	0.32 (0.21)
		0.10 (0.09)	0.11 (0.07)	0.15 (0.13)	0.15 (0.11)	0.22 (0.15)	0.29 (0.21)
		0.10 (0.09)	0.11 (0.07)	0.15 (0.13)	0.15 (0.11)	0.22 (0.15)	0.29 (0.21)
		0.09 (0.08)	0.11 (0.08)	0.14 (0.11)	0.17 (0.14)	0.26 (0.15)	0.22 (0.20)
		0.11 (0.08)	0.11 (0.08)	0.13 (0.13)	0.11 (0.10)	0.13 (0.05)	0.20 (0.15)
		0.11 (0.08)	0.11 (0.08)	0.13 (0.13)	0.11 (0.10)	0.13 (0.05)	0.20 (0.15)
		X	X	X	X	X	X
		0.08 (0.08)	0.05 (0.05)	0.10 (0.12)	0.05 (0.06)	0.05 (0.04)	0.11 (0.12)
	20	0.14 (0.08)	0.30 (0.16)	0.33 (0.16)	0.28 (0.13)	0.53 (0.19)	0.43 (0.25)
		0.14 (0.06)	0.28 (0.16)	0.27 (0.13)	0.26 (0.13)	0.43 (0.09)	0.40 (0.26)
		0.12 (0.08)	0.20 (0.14)	0.28 (0.15)	0.25 (0.11)	0.41 (0.13)	0.40 (0.12)
		0.09 (0.07)	0.17 (0.14)	0.22 (0.13)	0.19 (0.11)	0.33 (0.11)	0.31 (0.13)
		0.09 (0.07)	0.17 (0.14)	0.22 (0.13)	0.19 (0.11)	0.33 (0.11)	0.31 (0.13)
		0.07 (0.05)	0.17 (0.12)	0.23 (0.13)	0.17 (0.08)	0.32 (0.11)	0.30 (0.14)
		0.11 (0.04)	0.17 (0.09)	0.21 (0.10)	0.18 (0.07)	0.31 (0.11)	0.21 (0.11)
		0.11 (0.04)	0.17 (0.09)	0.21 (0.10)	0.18 (0.07)	0.31 (0.11)	0.21 (0.11)
		X	X	X	X	X	X
		0.06 (0.06)	0.08 (0.07)	0.16 (0.09)	0.09 (0.06)	0.19 (0.11)	0.08 (0.07)
	30	0.13 (0.06)	0.26 (0.12)	0.32 (0.14)	0.40 (0.13)	0.45 (0.13)	0.60 (0.15)
		0.13 (0.07)	0.21 (0.08)	0.28 (0.14)	0.34 (0.13)	0.36 (0.12)	0.53 (0.20)
		0.12 (0.07)	0.14 (0.07)	0.30 (0.15)	0.28 (0.11)	0.35 (0.10)	0.42 (0.13)
		0.07 (0.06)	0.11 (0.05)	0.23 (0.11)	0.21 (0.09)	0.31 (0.09)	0.40 (0.15)
		0.07 (0.06)	0.11 (0.05)	0.23 (0.11)	0.21 (0.09)	0.31 (0.09)	0.40 (0.15)
		0.08 (0.07)	0.12 (0.08)	0.23 (0.14)	0.22 (0.11)	0.31 (0.12)	0.31 (0.14)
		0.09 (0.05)	0.09 (0.02)	0.24 (0.10)	0.23 (0.10)	0.27 (0.04)	0.27 (0.11)
		0.09 (0.05)	0.09 (0.02)	0.24 (0.10)	0.23 (0.10)	0.27 (0.04)	0.27 (0.11)
		X	X	X	X	X	X
		0.05 (0.04)	0.05 (0.02)	0.18 (0.10)	0.13 (0.08)	0.16 (0.08)	0.12 (0.06)
	100	0.11 (0.02)	0.17 (0.06)	0.30 (0.08)	0.34 (0.07)	0.35 (0.07)	0.56 (0.18)
		0.11 (0.03)	0.17 (0.06)	0.28 (0.07)	0.33 (0.06)	0.31 (0.04)	0.41 (0.10)
		0.09 (0.02)	0.09 (0.01)	0.26 (0.07)	0.29 (0.05)	0.31 (0.04)	0.38 (0.09)
		0.06 (0.03)	0.09 (0.08)	0.25 (0.11)	0.28 (0.10)	0.25 (0.03)	0.31 (0.07)
		0.33 (0.37)	0.29 (0.20)	0.63 (0.46)	0.54 (0.36)	0.26 (0.06)	0.32 (0.07)
		0.08 (0.06)	0.06 (0.05)	0.22 (0.09)	0.21 (0.04)	0.23 (0.05)	0.30 (0.09)
		0.09 (0.02)	0.08 (0.01)	0.25 (0.06)	0.27 (0.05)	0.28 (0.02)	0.31 (0.05)
		0.09 (0.02)	0.08 (0.01)	0.25 (0.06)	0.27 (0.05)	0.28 (0.02)	0.31 (0.05)
		X	X	X	X	X	X
		0.07 (0.03)	0.06 (0.01)	0.20 (0.05)	0.21 (0.04)	0.16 (0.08)	0.22 (0.04)

Loss Values - High Setup Times, 1-10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.11 (0.06)	0.21 (0.10)	0.17 (0.10)	0.28 (0.13)	0.20 (0.19)	0.26 (0.17)
		0.09 (0.06)	0.14 (0.07)	0.13 (0.07)	0.21 (0.16)	0.14 (0.13)	0.20 (0.14)
		0.08 (0.04)	0.10 (0.06)	0.13 (0.07)	0.17 (0.08)	0.16 (0.13)	0.19 (0.12)
		0.07 (0.05)	0.12 (0.09)	0.11 (0.07)	0.15 (0.12)	0.13 (0.13)	0.20 (0.12)
		0.07 (0.05)	0.12 (0.09)	0.11 (0.07)	0.15 (0.12)	0.13 (0.13)	0.20 (0.12)
		0.10 (0.07)	0.12 (0.07)	0.15 (0.06)	0.19 (0.09)	0.17 (0.17)	0.17 (0.14)
		0.08 (0.05)	0.10 (0.06)	0.11 (0.06)	0.15 (0.09)	0.14 (0.13)	0.18 (0.13)
		0.08 (0.05)	0.11 (0.08)	0.11 (0.06)	0.18 (0.13)	0.17 (0.16)	0.19 (0.12)
		X	X	X	X	X	X
		0.05 (0.05)	0.03 (0.03)	0.09 (0.06)	0.08 (0.05)	0.09 (0.10)	0.10 (0.12)
	20	0.10 (0.03)	0.24 (0.12)	0.29 (0.15)	0.37 (0.16)	0.52 (0.10)	0.53 (0.24)
		0.09 (0.03)	0.18 (0.09)	0.24 (0.13)	0.31 (0.13)	0.34 (0.12)	0.39 (0.20)
		0.08 (0.02)	0.16 (0.08)	0.26 (0.12)	0.29 (0.15)	0.43 (0.21)	0.37 (0.16)
		0.05 (0.03)	0.12 (0.07)	0.21 (0.13)	0.24 (0.14)	0.33 (0.15)	0.35 (0.15)
		0.05 (0.03)	0.12 (0.07)	0.21 (0.13)	0.24 (0.14)	0.33 (0.15)	0.35 (0.15)
		0.10 (0.09)	0.18 (0.10)	0.27 (0.16)	0.30 (0.14)	0.40 (0.15)	0.36 (0.18)
		0.08 (0.02)	0.11 (0.04)	0.24 (0.11)	0.24 (0.11)	0.33 (0.18)	0.32 (0.13)
		0.08 (0.02)	0.14 (0.08)	0.21 (0.12)	0.22 (0.10)	0.29 (0.08)	0.28 (0.14)
		X	X	X	X	X	X
		0.03 (0.02)	0.04 (0.02)	0.13 (0.10)	0.12 (0.08)	0.16 (0.05)	0.16 (0.10)
	30	0.11 (0.03)	0.28 (0.12)	0.33 (0.14)	0.41 (0.18)	0.63 (0.17)	0.65 (0.24)
		0.10 (0.03)	0.21 (0.13)	0.28 (0.14)	0.34 (0.16)	0.49 (0.17)	0.49 (0.17)
		0.09 (0.02)	0.15 (0.07)	0.27 (0.12)	0.29 (0.15)	0.45 (0.14)	0.46 (0.17)
		0.06 (0.04)	0.14 (0.07)	0.22 (0.12)	0.24 (0.14)	0.46 (0.20)	0.40 (0.14)
		0.06 (0.04)	0.14 (0.07)	0.22 (0.12)	0.24 (0.14)	0.46 (0.20)	0.40 (0.14)
		0.08 (0.06)	0.19 (0.10)	0.31 (0.19)	0.33 (0.21)	0.46 (0.15)	0.46 (0.19)
		0.08 (0.02)	0.12 (0.04)	0.25 (0.10)	0.25 (0.12)	0.37 (0.14)	0.39 (0.14)
		0.08 (0.02)	0.11 (0.04)	0.22 (0.10)	0.24 (0.12)	0.35 (0.13)	0.36 (0.09)
		X	X	X	X	X	X
		0.03 (0.01)	0.04 (0.02)	0.15 (0.09)	0.14 (0.07)	0.19 (0.10)	0.20 (0.05)
	100	0.09 (0.02)	0.18 (0.08)	0.33 (0.06)	0.34 (0.10)	0.41 (0.10)	0.49 (0.11)
		0.09 (0.02)	0.14 (0.05)	0.29 (0.04)	0.32 (0.10)	0.34 (0.08)	0.38 (0.06)
		0.09 (0.01)	0.11 (0.04)	0.28 (0.04)	0.29 (0.08)	0.32 (0.06)	0.37 (0.06)
		0.13 (0.13)	0.13 (0.07)	0.26 (0.04)	0.33 (0.12)	0.27 (0.07)	0.34 (0.12)
		1.30 (1.50)	1.47 (1.18)	1.32 (1.04)	1.78 (1.10)	0.43 (0.43)	0.33 (0.08)
		0.09 (0.09)	0.19 (0.17)	0.40 (0.13)	0.34 (0.15)	0.35 (0.12)	0.34 (0.09)
		0.08 (0.01)	0.10 (0.03)	0.26 (0.04)	0.28 (0.08)	0.31 (0.05)	0.32 (0.05)
		0.08 (0.01)	0.10 (0.04)	0.25 (0.03)	0.27 (0.06)	0.29 (0.02)	0.31 (0.04)
		X	X	X	X	X	X
		0.06 (0.01)	0.07 (0.03)	0.22 (0.03)	0.21 (0.05)	0.23 (0.05)	0.23 (0.06)

Loss Values - High Setup Times, 1-10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.07 (0.02)	0.24 (0.06)	0.14 (0.05)	0.25 (0.10)	0.37 (0.16)	0.26 (0.18)
		0.07 (0.02)	0.19 (0.08)	0.11 (0.05)	0.18 (0.06)	0.27 (0.12)	0.15 (0.08)
		0.06 (0.02)	0.16 (0.07)	0.12 (0.06)	0.17 (0.06)	0.28 (0.14)	0.19 (0.10)
		0.05 (0.02)	0.14 (0.09)	0.11 (0.07)	0.17 (0.08)	0.23 (0.14)	0.18 (0.11)
		0.05 (0.02)	0.14 (0.09)	0.11 (0.07)	0.17 (0.08)	0.23 (0.14)	0.18 (0.11)
		0.12 (0.07)	0.22 (0.12)	0.13 (0.05)	0.16 (0.07)	0.29 (0.10)	0.24 (0.11)
		0.06 (0.02)	0.16 (0.07)	0.10 (0.05)	0.14 (0.05)	0.27 (0.12)	0.18 (0.11)
		0.05 (0.02)	0.15 (0.07)	0.11 (0.07)	0.15 (0.05)	0.26 (0.12)	0.17 (0.09)
		X	X	X	X	X	X
		0.03 (0.02)	0.08 (0.06)	0.05 (0.04)	0.08 (0.04)	0.12 (0.07)	0.08 (0.04)
	20	0.11 (0.03)	0.22 (0.09)	0.28 (0.12)	0.39 (0.13)	0.66 (0.19)	0.60 (0.22)
		0.11 (0.02)	0.18 (0.11)	0.25 (0.13)	0.28 (0.09)	0.40 (0.10)	0.45 (0.16)
		0.08 (0.01)	0.16 (0.06)	0.23 (0.12)	0.23 (0.07)	0.49 (0.13)	0.37 (0.12)
		0.07 (0.02)	0.13 (0.08)	0.20 (0.13)	0.21 (0.09)	0.41 (0.15)	0.33 (0.13)
		0.07 (0.02)	0.13 (0.08)	0.20 (0.13)	0.21 (0.09)	0.41 (0.15)	0.33 (0.13)
		0.30 (0.19)	0.36 (0.12)	0.46 (0.20)	0.39 (0.14)	0.54 (0.19)	0.53 (0.19)
		0.07 (0.01)	0.14 (0.06)	0.21 (0.10)	0.22 (0.07)	0.36 (0.11)	0.39 (0.14)
		0.08 (0.02)	0.12 (0.05)	0.21 (0.11)	0.20 (0.07)	0.35 (0.11)	0.35 (0.12)
		X	X	X	X	X	X
		0.03 (0.01)	0.04 (0.03)	0.13 (0.08)	0.11 (0.06)	0.17 (0.06)	0.18 (0.06)
	30	0.09 (0.02)	0.19 (0.07)	0.32 (0.13)	0.33 (0.10)	0.53 (0.21)	0.70 (0.18)
		0.09 (0.01)	0.17 (0.06)	0.26 (0.10)	0.29 (0.08)	0.40 (0.14)	0.51 (0.16)
		0.08 (0.01)	0.14 (0.04)	0.25 (0.09)	0.26 (0.10)	0.42 (0.13)	0.44 (0.14)
		0.05 (0.01)	0.12 (0.06)	0.22 (0.11)	0.21 (0.10)	0.34 (0.11)	0.40 (0.12)
		0.05 (0.01)	0.12 (0.06)	0.22 (0.11)	0.21 (0.10)	0.34 (0.11)	0.40 (0.12)
		0.24 (0.17)	0.35 (0.22)	0.47 (0.28)	0.33 (0.10)	0.48 (0.22)	0.65 (0.23)
		0.07 (0.01)	0.10 (0.03)	0.24 (0.09)	0.24 (0.08)	0.38 (0.14)	0.41 (0.12)
		0.07 (0.01)	0.11 (0.03)	0.22 (0.11)	0.21 (0.09)	0.34 (0.11)	0.43 (0.17)
		X	X	X	X	X	X
		0.03 (0.01)	0.04 (0.02)	0.14 (0.07)	0.11 (0.05)	0.20 (0.05)	0.20 (0.05)
	100	0.09 (0.01)	0.14 (0.05)	0.29 (0.03)	0.37 (0.14)	0.50 (0.19)	0.43 (0.07)
		0.09 (0.01)	0.14 (0.04)	0.28 (0.03)	0.33 (0.11)	0.40 (0.14)	0.36 (0.06)
		0.09 (0.01)	0.11 (0.02)	0.28 (0.03)	0.30 (0.09)	0.39 (0.11)	0.33 (0.04)
		0.12 (0.12)	0.21 (0.10)	0.33 (0.16)	0.34 (0.10)	0.36 (0.13)	0.33 (0.09)
		5.56 (3.04)	7.48 (3.71)	7.02 (3.19)	6.90 (2.99)	1.93 (1.88)	1.72 (1.70)
		0.33 (0.16)	0.38 (0.20)	0.57 (0.24)	0.70 (0.22)	0.65 (0.23)	0.53 (0.30)
		0.09 (0.01)	0.11 (0.02)	0.25 (0.02)	0.28 (0.08)	0.36 (0.09)	0.31 (0.03)
		0.08 (0.01)	0.10 (0.02)	0.24 (0.03)	0.26 (0.07)	0.33 (0.06)	0.30 (0.03)
		X	X	X	X	X	X
		0.06 (0.00)	0.06 (0.02)	0.20 (0.03)	0.20 (0.06)	0.25 (0.08)	0.24 (0.04)

Appendix F: Multistage Heuristics – Running Time Statistics

Format of each cell: average (standard deviation) in seconds

FTMIH, CTMIH, MMIH, GA, RKGA

Note: CH, RCH, SPTCH and the Johnson's Rule Based Heuristics had no running time greater than 0.01 seconds.

GA was only run for low setup time files, shown as X in high setup time files.

Low Setup Times, 1 Machine per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.00 (0.00)	0.01 (0.01)	0.01 (0.00)	0.01 (0.00)	0.00 (0.01)	0.00 (0.00)
		0.01 (0.01)	0.01 (0.01)	0.00 (0.00)	0.00 (0.01)	0.00 (0.00)	0.00 (0.01)
		0.00 (0.01)	0.00 (0.01)	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
		1.22 (0.00)	1.23 (0.01)	1.24 (0.01)	1.23 (0.02)	1.22 (0.01)	1.22 (0.01)
		2.44 (0.32)	2.34 (0.07)	2.34 (0.08)	2.36 (0.10)	2.30 (0.07)	2.40 (0.10)
	20	0.06 (0.01)	0.06 (0.00)	0.06 (0.01)	0.05 (0.01)	0.04 (0.00)	0.04 (0.01)
		0.06 (0.00)	0.06 (0.00)	0.05 (0.01)	0.06 (0.00)	0.03 (0.01)	0.04 (0.01)
		0.06 (0.00)	0.06 (0.00)	0.05 (0.00)	0.06 (0.01)	0.03 (0.00)	0.03 (0.00)
		7.47 (0.01)	7.49 (0.01)	7.49 (0.03)	7.47 (0.04)	7.23 (0.06)	7.24 (0.08)
		6.98 (1.36)	6.40 (2.24)	6.74 (1.98)	6.48 (1.70)	5.65 (1.56)	5.85 (1.23)
	30	0.17 (0.00)	0.18 (0.01)	0.16 (0.01)	0.16 (0.01)	0.10 (0.02)	0.10 (0.02)
		0.17 (0.00)	0.17 (0.00)	0.16 (0.01)	0.16 (0.00)	0.10 (0.01)	0.10 (0.02)
		0.17 (0.00)	0.17 (0.00)	0.16 (0.01)	0.16 (0.01)	0.10 (0.01)	0.09 (0.02)
		17.10	17.24	17.22	17.14	16.51	16.50
		(0.04)	(0.05)	(0.08)	(0.06)	(0.16)	(0.21)
	100	10.94	12.00	10.14	12.26	9.14 (2.72)	9.08 (2.81)
		(4.96)	(2.88)	(1.49)	(3.38)		
		8.86 (0.10)	8.84 (0.05)	8.20 (0.21)	8.19 (0.24)	4.13 (0.44)	4.23 (0.39)
		8.49 (0.07)	8.50 (0.06)	8.05 (0.22)	7.98 (0.24)	3.97 (0.43)	4.16 (0.39)
		8.50 (0.06)	8.54 (0.05)	7.97 (0.22)	7.94 (0.24)	3.95 (0.44)	4.09 (0.38)
		323.48	334.18	332.09	331.94	308.84	309.79
		(0.80)	(0.20)	(1.61)	(1.59)	(3.19)	(2.20)
		46.84	43.54	43.55	54.68	39.20	39.54
		(9.87)	(9.37)	(8.79)	(21.60)	(7.05)	(11.28)

Running Times - Low Setup Times, 1 Machine per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.01 (0.01)	0.01 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)
		2.25 (0.01)	2.24 (0.01)	2.23 (0.01)	2.22 (0.01)	2.20 (0.01)	2.20 (0.01)
		11.36 (0.28)	11.55 (0.31)	11.41 (0.27)	11.52 (0.22)	11.25 (0.27)	11.52 (0.21)
	20	0.12 (0.00)	0.12 (0.01)	0.11 (0.01)	0.11 (0.01)	0.07 (0.01)	0.07 (0.01)
		0.12 (0.00)	0.12 (0.00)	0.11 (0.00)	0.11 (0.00)	0.07 (0.01)	0.07 (0.01)
		0.12 (0.01)	0.12 (0.00)	0.11 (0.00)	0.11 (0.01)	0.06 (0.01)	0.06 (0.01)
		13.35 (0.11)	13.23 (0.14)	13.15 (0.15)	13.04 (0.13)	12.46 (0.16)	12.36 (0.16)
		28.06 (9.46)	29.27 (6.85)	28.02 (7.53)	26.85 (3.93)	24.36 (3.99)	26.84 (4.86)
	30	0.35 (0.01)	0.35 (0.01)	0.33 (0.01)	0.33 (0.01)	0.19 (0.02)	0.19 (0.02)
		0.34 (0.00)	0.34 (0.00)	0.32 (0.01)	0.32 (0.01)	0.18 (0.03)	0.19 (0.02)
		0.34 (0.00)	0.33 (0.01)	0.31 (0.01)	0.32 (0.01)	0.18 (0.02)	0.18 (0.02)
		28.33 (0.04)	28.08 (0.04)	27.99 (0.15)	27.85 (0.16)	25.87 (0.45)	25.92 (0.29)
		42.38 (11.13)	47.60 (9.03)	52.98 (14.33)	43.54 (8.68)	46.59 (11.42)	41.33 (7.58)
	100	17.03 (0.08)	16.95 (0.05)	15.41 (0.20)	15.64 (0.19)	7.47 (0.67)	7.84 (0.29)
		16.45 (0.06)	16.49 (0.05)	14.89 (0.19)	15.17 (0.19)	7.25 (0.65)	7.57 (0.29)
		16.40 (0.07)	16.49 (0.07)	14.82 (0.19)	15.11 (0.19)	7.24 (0.64)	7.54 (0.29)
		508.17 (1.46)	515.35 (0.33)	508.05 (1.94)	508.83 (2.62)	445.44	449.98
		258.95 (45.23)	281.69 (95.58)	278.83 (30.99)	273.27 (30.56)	6.60	3.18
						201.75	176.36
						(97.19)	(55.92)

Running Times - Low Setup Times, 1 Machine per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.00)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.01 (0.00)	0.02 (0.01)	0.01 (0.01)	0.02 (0.00)	0.01 (0.00)	0.01 (0.00)
		4.45 (0.02)	4.43 (0.01)	4.40 (0.01)	4.40 (0.01)	4.34 (0.01)	4.35 (0.03)
		48.40	48.80	48.14	48.56	48.12	48.88
	20	(0.62)	(0.86)	(0.50)	(0.47)	(0.88)	(0.25)
		0.25 (0.01)	0.25 (0.01)	0.23 (0.01)	0.23 (0.01)	0.14 (0.01)	0.14 (0.01)
		0.24 (0.00)	0.24 (0.01)	0.23 (0.01)	0.23 (0.01)	0.13 (0.01)	0.13 (0.01)
		0.23 (0.00)	0.23 (0.01)	0.22 (0.00)	0.23 (0.01)	0.13 (0.01)	0.13 (0.01)
		25.41	25.32	25.00	25.00	23.39	23.55
	30	(0.09)	(0.03)	(0.10)	(0.08)	(0.16)	(0.28)
		134.78	124.84	130.52	141.52	130.88	127.51
		(27.54)	(16.87)	(26.07)	(44.05)	(24.55)	(25.92)
		0.71 (0.01)	0.71 (0.01)	0.66 (0.01)	0.66 (0.01)	0.40 (0.03)	0.38 (0.02)
		0.69 (0.00)	0.69 (0.00)	0.64 (0.01)	0.65 (0.01)	0.39 (0.03)	0.37 (0.02)
	100	0.68 (0.00)	0.68 (0.00)	0.63 (0.01)	0.64 (0.01)	0.38 (0.03)	0.37 (0.01)
		54.91	54.86	54.03	54.06	49.84	49.87
		(0.03)	(0.08)	(0.16)	(0.10)	(0.51)	(0.23)
		209.14	212.37	191.39	233.96	206.42	200.74
		(32.75)	(35.40)	(32.25)	(56.45)	(58.14)	(43.80)
		34.04	34.10	30.87	31.16	15.22	15.31
		(0.12)	(0.12)	(0.44)	(0.26)	(0.70)	(0.79)
		33.17	32.98	29.97	30.36	14.90	14.88
		(0.18)	(0.09)	(0.43)	(0.26)	(0.70)	(0.77)
		33.06	32.86	29.81	30.27	14.79	14.79
		(0.12)	(0.10)	(0.41)	(0.22)	(0.70)	(0.77)
		934.39	939.24	920.34	920.21	781.29	780.67
		(2.60)	(2.45)	(2.11)	(3.08)	(7.87)	(8.17)
		1339.44	1464.66	1612.06	1679.29	960.34	646.66
		(463.20)	(232.74)	(531.32)	(307.72)	(390.82)	(282.68)

Running Times - Low Setup Times, 2 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.00)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.00)
		1.26 (0.01)	1.25 (0.01)	1.25 (0.01)	1.25 (0.01)	1.24 (0.01)	1.24 (0.01)
		2.37 (0.10)	2.53 (0.42)	2.44 (0.14)	2.69 (0.67)	2.29 (0.02)	2.44 (0.17)
	20	0.12 (0.01)	0.12 (0.00)	0.12 (0.01)	0.12 (0.01)	0.06 (0.01)	0.08 (0.02)
		0.12 (0.00)	0.12 (0.00)	0.12 (0.01)	0.11 (0.01)	0.06 (0.01)	0.07 (0.01)
		0.12 (0.00)	0.12 (0.00)	0.11 (0.01)	0.11 (0.01)	0.06 (0.01)	0.07 (0.01)
		7.61 (0.02)	7.60 (0.02)	7.57 (0.04)	7.62 (0.03)	7.33 (0.08)	7.38 (0.07)
		6.87 (1.97)	7.87 (2.19)	7.47 (2.59)	6.57 (0.96)	6.62 (3.21)	7.49 (2.97)
	30	0.36 (0.01)	0.36 (0.01)	0.34 (0.02)	0.34 (0.01)	0.20 (0.02)	0.19 (0.03)
		0.35 (0.00)	0.35 (0.00)	0.32 (0.02)	0.33 (0.01)	0.19 (0.02)	0.18 (0.03)
		0.35 (0.01)	0.34 (0.00)	0.32 (0.02)	0.32 (0.01)	0.18 (0.02)	0.18 (0.03)
		17.37 (0.02)	17.46 (0.04)	17.40 (0.06)	17.46 (0.06)	16.77 (0.16)	16.65 (0.20)
		12.72 (4.21)	10.08 (2.66)	10.83 (1.75)	10.64 (4.11)	10.36 (5.10)	9.08 (2.66)
	100	18.06 (0.09)	18.03 (0.10)	16.22 (0.39)	16.46 (0.36)	8.17 (0.74)	8.10 (0.89)
		17.30 (0.07)	17.27 (0.10)	15.99 (0.35)	15.87 (0.35)	7.83 (0.72)	7.91 (0.88)
		17.37 (0.08)	17.38 (0.10)	15.83 (0.36)	15.81 (0.35)	7.80 (0.72)	7.78 (0.87)
		324.96 (0.22)	336.51 (0.69)	333.48 (1.31)	333.96 (1.64)	311.84 (3.15)	311.28 (2.91)
		62.98 (30.42)	60.58 (16.38)	61.48 (26.28)	55.01 (16.47)	23.49 (10.35)	33.59 (9.09)

Running Times - Low Setup Times, 2 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.00)	0.01 (0.01)
		0.02 (0.00)	0.02 (0.00)	0.01 (0.00)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.00)	0.02 (0.01)	0.02 (0.00)	0.01 (0.00)	0.01 (0.00)
		2.27 (0.01)	2.26 (0.01)	2.26 (0.02)	2.26 (0.01)	2.22 (0.01)	2.22 (0.01)
		11.79 (0.90)	11.55 (0.38)	11.52 (0.33)	11.55 (0.33)	11.50 (0.32)	11.58 (0.25)
	20	0.25 (0.01)	0.25 (0.01)	0.24 (0.01)	0.24 (0.01)	0.14 (0.01)	0.15 (0.02)
		0.24 (0.00)	0.25 (0.01)	0.23 (0.01)	0.23 (0.01)	0.14 (0.01)	0.14 (0.01)
		0.25 (0.03)	0.24 (0.00)	0.23 (0.01)	0.23 (0.01)	0.13 (0.01)	0.14 (0.01)
		13.30 (0.12)	13.38 (0.15)	13.34 (0.12)	13.32 (0.14)	12.74 (0.19)	12.73 (0.21)
		38.91 (6.30)	34.34 (8.55)	37.65 (11.15)	31.29 (5.74)	29.63 (4.52)	28.32 (5.64)
	30	0.72 (0.01)	0.72 (0.01)	0.70 (0.01)	0.69 (0.02)	0.40 (0.02)	0.38 (0.04)
		0.70 (0.00)	0.70 (0.00)	0.68 (0.01)	0.67 (0.01)	0.39 (0.02)	0.37 (0.04)
		0.70 (0.01)	0.70 (0.01)	0.67 (0.01)	0.65 (0.02)	0.38 (0.02)	0.36 (0.04)
		28.59 (0.03)	28.57 (0.05)	28.40 (0.08)	28.33 (0.15)	26.64 (0.11)	26.31 (0.28)
		54.80 (10.31)	54.68 (16.06)	52.39 (12.41)	47.26 (5.81)	45.47 (11.13)	39.96 (9.81)
	100	34.79 (0.16)	34.61 (0.16)	31.53 (0.44)	31.70 (0.34)	15.76 (1.29)	15.83 (1.42)
		33.54 (0.13)	33.77 (0.14)	30.54 (0.42)	30.73 (0.34)	15.30 (1.25)	15.30 (1.34)
		33.45 (0.14)	33.60 (0.17)	30.39 (0.42)	30.62 (0.33)	15.30 (1.26)	15.22 (1.34)
		513.33 (1.74)	521.72 (1.59)	513.65 (1.91)	512.03 (1.28)	452.93 (5.08)	453.47 (6.40)
		242.60 (111.75)	229.32 (78.43)	279.12 (106.29)	181.97 (81.30)	132.01 (53.63)	180.62 (110.18)

Running Times - Low Setup Times, 2 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.04 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.01)
		0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.01)
		0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.01)
		4.49 (0.01)	4.47 (0.01)	4.51 (0.04)	4.46 (0.01)	4.38 (0.03)	4.42 (0.04)
		50.08 (1.43)	50.08 (0.85)	49.25 (0.41)	49.17 (0.82)	50.13 (1.72)	49.34 (1.67)
	20	0.50 (0.00)	0.50 (0.00)	0.47 (0.01)	0.47 (0.01)	0.29 (0.02)	0.29 (0.03)
		0.49 (0.01)	0.49 (0.01)	0.46 (0.01)	0.46 (0.01)	0.28 (0.02)	0.28 (0.03)
		0.48 (0.01)	0.48 (0.00)	0.45 (0.01)	0.45 (0.01)	0.28 (0.02)	0.27 (0.03)
		25.81 (0.03)	25.69 (0.04)	25.59 (0.08)	25.48 (0.13)	24.00 (0.15)	24.14 (0.31)
		175.43 (45.46)	144.71 (27.45)	168.90 (24.85)	145.49 (26.99)	145.62 (34.44)	133.10 (16.91)
	30	1.45 (0.00)	1.46 (0.01)	1.36 (0.04)	1.36 (0.03)	0.78 (0.03)	0.77 (0.03)
		1.41 (0.01)	1.41 (0.00)	1.32 (0.03)	1.33 (0.04)	0.76 (0.03)	0.75 (0.03)
		1.40 (0.01)	1.40 (0.01)	1.31 (0.03)	1.31 (0.04)	0.75 (0.03)	0.74 (0.03)
		55.91 (0.04)	55.63 (0.04)	55.25 (0.34)	55.43 (0.27)	50.36 (0.27)	50.58 (0.37)
		271.42 (81.47)	267.81 (57.66)	245.00 (73.02)	240.44 (56.58)	246.12 (89.57)	226.73 (40.80)
	100	69.45 (0.23)	69.82 (0.34)	63.12 (1.01)	63.96 (0.72)	31.88 (2.06)	31.43 (1.23)
		67.73 (0.34)	67.42 (0.21)	61.30 (0.98)	62.40 (0.72)	31.15 (2.03)	30.54 (1.20)
		67.48 (0.27)	67.20 (0.19)	61.12 (1.04)	62.18 (0.70)	30.96 (2.01)	30.40 (1.19)
		945.62 (2.52)	949.03 (2.09)	931.28 (3.50)	931.15 (4.37)	794.89 (10.37)	790.68 (6.87)
		789.90 (211.92)	1342.80 (451.69)	987.62 (398.24)	1099.19 (339.46)	732.87 (269.06)	820.78 (300.06)

Running Times - Low Setup Times, 10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.71 (0.01)	0.72 (0.01)	0.67 (0.02)	0.68 (0.04)	0.40 (0.05)	0.39 (0.05)
		0.69 (0.00)	0.70 (0.01)	0.65 (0.02)	0.66 (0.04)	0.39 (0.05)	0.38 (0.05)
		0.69 (0.00)	0.69 (0.01)	0.65 (0.02)	0.65 (0.04)	0.39 (0.05)	0.37 (0.05)
		8.25 (0.02)	8.34 (0.04)	8.25 (0.02)	8.26 (0.05)	7.99 (0.07)	8.01 (0.07)
		7.76 (2.71)	10.32 (6.68)	7.94 (1.82)	9.61 (2.43)	5.59 (1.53)	4.94 (0.96)
	30	2.11 (0.01)	2.11 (0.01)	1.99 (0.08)	1.98 (0.05)	1.19 (0.13)	1.14 (0.19)
		2.05 (0.01)	2.05 (0.00)	1.93 (0.07)	1.93 (0.05)	1.16 (0.13)	1.11 (0.18)
		2.04 (0.01)	2.04 (0.01)	1.92 (0.08)	1.91 (0.05)	1.15 (0.13)	1.11 (0.18)
		18.64 (0.04)	18.92 (0.04)	18.71 (0.06)	18.68 (0.05)	18.02 (0.18)	18.05 (0.21)
		13.71 (3.80)	15.19 (5.53)	11.72 (3.57)	13.08 (3.48)	8.12 (0.73)	8.72 (3.10)
	100	100.55 (0.49)	101.48 (0.60)	90.60 (2.33)	94.17 (2.06)	47.78 (4.06)	49.42 (4.96)
		97.16 (0.38)	97.39 (0.59)	89.02 (2.23)	90.50 (2.04)	45.88 (3.93)	47.82 (4.81)
		98.29 (0.37)	97.52 (0.61)	88.87 (2.28)	90.60 (2.11)	45.88 (3.92)	47.79 (4.79)
		341.97 (1.09)	352.89 (1.15)	350.53 (1.86)	349.89 (1.35)	326.01 (2.61)	327.66 (3.13)
		64.32 (30.56)	63.69 (27.83)	60.07 (23.04)	59.25 (15.86)	52.44 (10.02)	47.74 (18.76)

Running Times - Low Setup Times, 10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	1.48 (0.01)	1.48 (0.00)	1.40 (0.04)	1.40 (0.03)	0.85 (0.10)	0.84 (0.07)
		1.44 (0.00)	1.44 (0.01)	1.37 (0.04)	1.36 (0.03)	0.83 (0.09)	0.82 (0.07)
		1.42 (0.00)	1.43 (0.01)	1.35 (0.04)	1.35 (0.03)	0.82 (0.09)	0.81 (0.06)
		15.11	15.14	15.05	15.04	14.39	14.39
		(0.02)	(0.02)	(0.05)	(0.05)	(0.14)	(0.10)
		40.65	43.40	34.97	33.72	22.80	18.18
		(14.36)	(14.95)	(10.09)	(7.84)	(9.45)	(2.45)
	30	4.40 (0.01)	4.39 (0.01)	4.16 (0.08)	4.15 (0.09)	2.40 (0.21)	2.37 (0.24)
		4.27 (0.01)	4.28 (0.01)	4.04 (0.08)	4.04 (0.08)	2.34 (0.20)	2.31 (0.24)
		4.24 (0.01)	4.26 (0.02)	4.01 (0.08)	4.01 (0.09)	2.33 (0.18)	2.29 (0.23)
		31.94	32.00	31.76	31.80	29.79	29.73
		(0.03)	(0.05)	(0.06)	(0.10)	(0.27)	(0.34)
		67.66	74.19	67.18	66.93	49.91	43.88
		(18.35)	(12.98)	(26.71)	(8.09)	(12.57)	(13.04)
	100	201.69	200.71	184.94	186.91	101.13	94.23
		(0.99)	(0.72)	(2.35)	(2.41)	(7.29)	(7.97)
		194.99	195.57	178.54	181.40	98.43	91.50
		(0.56)	(0.86)	(2.30)	(2.31)	(7.05)	(7.71)
		195.06	195.34	178.30	181.58	98.55	91.41
		(0.57)	(1.08)	(2.36)	(2.44)	(7.00)	(7.69)
		549.79	559.15	550.68	551.58	496.11	505.58
		(0.24)	(1.70)	(1.30)	(1.51)	(6.08)	(19.20)
		200.59	302.82	253.56	285.15	321.34	298.49
		(146.90)	(130.33)	(136.35)	(139.04)	(79.51)	(55.01)

Running Times - Low Setup Times, 10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	3.00 (0.01)	3.00 (0.01)	2.81 (0.06)	2.82 (0.07)	1.73 (0.16)	1.68 (0.08)
		2.92 (0.01)	2.92 (0.01)	2.74 (0.05)	2.75 (0.07)	1.69 (0.16)	1.63 (0.08)
		2.89 (0.01)	2.89 (0.01)	2.71 (0.05)	2.72 (0.07)	1.67 (0.16)	1.61 (0.08)
		29.20	29.19	29.16	29.16	27.40	27.53
		(0.02)	(0.02)	(0.05)	(0.11)	(0.25)	(0.13)
		230.13	183.01	207.64	171.29	104.73	78.73
		(78.35)	(52.46)	(53.35)	(39.22)	(32.95)	(27.52)
	30	8.91 (0.02)	8.96 (0.02)	8.31 (0.08)	8.38 (0.14)	5.01 (0.24)	4.92 (0.28)
		8.71 (0.01)	8.67 (0.02)	8.09 (0.08)	8.20 (0.15)	4.90 (0.24)	4.79 (0.27)
		8.66 (0.02)	8.61 (0.01)	8.04 (0.08)	8.14 (0.13)	4.85 (0.24)	4.75 (0.27)
		63.22	63.27	62.94	63.00	58.12	58.47
		(0.05)	(0.05)	(0.25)	(0.26)	(0.31)	(0.42)
		200.15	373.19	360.02	313.79	244.65	201.31
		(91.05)	(90.06)	(54.37)	(39.49)	(62.65)	(84.84)
	100	407.34	408.16	373.04	379.00	196.32	191.29
		(0.79)	(1.52)	(2.93)	(4.73)	(3.93)	(11.16)
		395.94	395.38	363.55	367.77	191.52	188.29
		(1.03)	(1.00)	(3.18)	(5.89)	(3.66)	(12.23)
		395.70	395.44	363.10	366.84	190.91	186.16
		(0.92)	(0.92)	(2.79)	(5.20)	(3.89)	(10.94)
		1030.92	1030.68	1012.26	1012.49	876.58	871.21
		(2.56)	(2.48)	(3.08)	(3.07)	(4.27)	(10.25)
		752.29	1049.33	951.75	1708.42	1507.28	1648.69
		(349.10)	(496.23)	(504.23)	(869.85)	(414.81)	(314.35)

Running Times - Low Setup Times, 1-4 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.01)	0.01 (0.00)	0.01 (0.01)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)	0.01 (0.01)
		1.27 (0.01)	1.26 (0.01)	1.25 (0.01)	1.26 (0.01)	1.23 (0.01)	1.25 (0.04)
		2.58 (0.41)	2.43 (0.14)	2.36 (0.09)	2.46 (0.14)	2.31 (0.04)	2.43 (0.14)
	20	0.17 (0.03)	0.16 (0.03)	0.16 (0.05)	0.16 (0.04)	0.08 (0.03)	0.10 (0.04)
		0.16 (0.03)	0.15 (0.04)	0.15 (0.04)	0.16 (0.04)	0.08 (0.03)	0.10 (0.04)
		0.16 (0.03)	0.15 (0.04)	0.15 (0.04)	0.15 (0.04)	0.08 (0.03)	0.10 (0.03)
		7.66 (0.04)	7.64 (0.06)	7.63 (0.07)	7.63 (0.07)	7.37 (0.10)	7.40 (0.12)
		7.65 (2.19)	7.01 (1.48)	7.15 (1.79)	7.01 (1.92)	5.57 (0.86)	5.37 (1.04)
	30	0.40 (0.14)	0.41 (0.08)	0.39 (0.12)	0.48 (0.11)	0.26 (0.08)	0.30 (0.05)
		0.38 (0.13)	0.40 (0.08)	0.38 (0.12)	0.46 (0.11)	0.25 (0.08)	0.28 (0.04)
		0.38 (0.13)	0.40 (0.08)	0.37 (0.12)	0.46 (0.10)	0.25 (0.09)	0.28 (0.04)
		17.36 (0.16)	17.48 (0.11)	17.37 (0.12)	17.46 (0.13)	16.65 (0.17)	16.78 (0.15)
		13.15 (5.89)	12.83 (4.63)	9.71 (3.74)	9.09 (2.25)	9.06 (1.71)	7.78 (1.77)
	100	19.42 (4.33)	21.85 (8.59)	19.51 (6.22)	19.44 (7.05)	8.86 (3.14)	10.27 (2.70)
		18.68 (4.16)	21.00 (8.21)	19.12 (6.12)	18.52 (6.67)	8.47 (3.00)	9.90 (2.60)
		18.85 (4.21)	21.03 (8.26)	19.03 (6.11)	18.46 (6.66)	8.46 (3.01)	9.88 (2.60)
		325.20 (1.20)	337.42 (1.81)	335.29 (1.68)	335.47 (3.00)	311.35	312.45
		41.06 (21.02)	59.56 (17.26)	51.09 (28.40)	81.89 (33.72)	38.74 (15.21)	312.45 (3.38)
							33.53 (18.30)

Running Times - Low Setup Times, 1-4 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.01)
		0.02 (0.01)	0.03 (0.01)	0.02 (0.00)	0.02 (0.01)	0.01 (0.00)	0.01 (0.01)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.00)	0.01 (0.00)	0.01 (0.01)
		2.27 (0.02)	2.28 (0.01)	2.28 (0.02)	2.27 (0.02)	2.23 (0.02)	2.23 (0.02)
		12.51 (1.66)	11.96 (0.62)	12.17 (0.42)	12.38 (1.01)	11.90 (0.58)	12.23 (0.63)
	20	0.34 (0.08)	0.35 (0.06)	0.33 (0.07)	0.31 (0.07)	0.18 (0.04)	0.20 (0.04)
		0.33 (0.08)	0.34 (0.06)	0.32 (0.06)	0.31 (0.06)	0.18 (0.03)	0.20 (0.04)
		0.33 (0.08)	0.33 (0.06)	0.32 (0.06)	0.30 (0.06)	0.17 (0.03)	0.20 (0.04)
		13.63 (0.17)	13.56 (0.16)	13.53 (0.18)	13.39 (0.18)	12.70 (0.24)	12.77 (0.20)
		35.14 (9.47)	31.31 (7.38)	36.03 (12.65)	28.45 (6.17)	27.70 (7.61)	25.55 (7.45)
	30	0.80 (0.23)	0.89 (0.14)	0.87 (0.21)	0.78 (0.14)	0.58 (0.13)	0.53 (0.11)
		0.78 (0.23)	0.87 (0.13)	0.85 (0.22)	0.76 (0.14)	0.56 (0.12)	0.52 (0.10)
		0.77 (0.23)	0.86 (0.13)	0.84 (0.21)	0.75 (0.14)	0.55 (0.12)	0.51 (0.10)
		28.61 (0.25)	28.74 (0.17)	28.73 (0.33)	28.41 (0.21)	26.72 (0.39)	26.82 (0.37)
		48.63 (14.91)	58.87 (13.54)	52.98 (23.67)	60.50 (20.00)	38.63 (12.75)	46.69 (8.93)
	100	47.76 (9.38)	43.81 (12.74)	43.76 (3.55)	38.25 (9.69)	19.46 (2.87)	21.76 (6.07)
		46.35 (9.10)	42.48 (12.34)	42.37 (3.53)	37.13 (9.38)	18.96 (2.79)	21.12 (5.90)
		46.25 (9.10)	42.39 (12.32)	42.22 (3.50)	37.07 (9.38)	18.91 (2.80)	21.03 (5.87)
		516.65 (3.04)	522.25 (3.35)	517.28 (2.30)	515.51 (3.50)	455.50 (5.27)	455.23 (5.57)
		188.81 (88.26)	124.31 (46.33)	210.36 (120.63)	213.12 (110.72)	214.50 (120.53)	159.49 (100.14)

Running Times - Low Setup Times, 1-4 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.04 (0.01)	0.04 (0.01)	0.05 (0.01)	0.04 (0.01)	0.03 (0.01)	0.03 (0.01)
		0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.02 (0.01)	0.03 (0.01)
		0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.03 (0.01)	0.02 (0.00)
		4.50 (0.02)	4.50 (0.02)	4.50 (0.02)	4.49 (0.02)	4.41 (0.04)	4.43 (0.03)
		50.34 (2.98)	50.12 (1.78)	51.19 (2.41)	49.93 (1.70)	50.17 (0.68)	48.94 (0.94)
	20	0.67 (0.10)	0.63 (0.12)	0.64 (0.08)	0.61 (0.12)	0.34 (0.04)	0.39 (0.07)
		0.66 (0.10)	0.61 (0.11)	0.62 (0.07)	0.59 (0.12)	0.33 (0.04)	0.39 (0.07)
		0.65 (0.10)	0.60 (0.11)	0.62 (0.07)	0.58 (0.12)	0.33 (0.04)	0.37 (0.07)
		25.93 (0.18)	25.86 (0.20)	25.73 (0.15)	25.70 (0.24)	24.16 (0.16)	24.38 (0.24)
		159.69 (27.76)	173.08 (53.20)	163.41 (56.63)	149.22 (31.63)	122.99 (22.35)	126.74 (18.11)
	30	1.76 (0.28)	1.94 (0.36)	1.81 (0.33)	1.85 (0.12)	1.01 (0.23)	0.99 (0.17)
		1.71 (0.27)	1.89 (0.35)	1.76 (0.32)	1.80 (0.12)	0.99 (0.24)	0.99 (0.17)
		1.70 (0.26)	1.87 (0.35)	1.74 (0.32)	1.78 (0.11)	0.97 (0.23)	0.95 (0.16)
		55.92 (0.39)	56.14 (0.45)	55.48 (0.51)	55.51 (0.27)	50.91 (0.57)	51.26 (0.49)
		221.47 (99.45)	276.91 (84.60)	311.54 (61.31)	223.31 (35.81)	226.00 (50.89)	198.53 (34.54)
	100	91.27 (14.50)	83.59 (16.53)	77.74 (11.94)	88.71 (8.99)	41.08 (8.15)	43.34 (7.58)
		88.21 (13.86)	81.37 (16.14)	75.71 (11.59)	85.70 (8.71)	39.78 (7.88)	43.45 (7.62)
		87.97 (13.86)	81.43 (16.17)	75.24 (11.55)	85.41 (8.69)	39.59 (7.81)	42.21 (7.39)
		955.41 (3.66)	953.62 (5.11)	935.15 (4.31)	937.99 (5.03)	800.50 (8.63)	803.29 (8.36)
		784.13 (511.26)	1368.85 (827.49)	1099.56 (520.55)	1161.36 (761.21)	877.32 (706.41)	604.55 (407.35)

Running Times - Low Setup Times, 1-10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.00)	0.01 (0.01)	0.02 (0.01)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.00)	0.01 (0.00)	0.01 (0.01)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.00)
		1.28 (0.01)	1.27 (0.02)	1.26 (0.01)	1.26 (0.01)	1.26 (0.02)	1.26 (0.01)
		2.46 (0.14)	2.69 (0.64)	2.45 (0.14)	2.55 (0.38)	2.38 (0.13)	2.41 (0.03)
	20	0.37 (0.14)	0.31 (0.12)	0.25 (0.12)	0.37 (0.15)	0.20 (0.11)	0.23 (0.09)
		0.36 (0.13)	0.30 (0.12)	0.25 (0.12)	0.36 (0.15)	0.20 (0.11)	0.22 (0.09)
		0.36 (0.13)	0.30 (0.12)	0.24 (0.12)	0.36 (0.14)	0.19 (0.10)	0.21 (0.08)
		7.88 (0.18)	7.82 (0.14)	7.75 (0.17)	7.88 (0.19)	7.63 (0.22)	7.67 (0.18)
		7.07 (2.17)	8.15 (1.62)	7.28 (1.60)	7.99 (2.13)	6.46 (2.69)	6.77 (4.03)
	30	0.99 (0.37)	1.25 (0.49)	1.21 (0.42)	0.99 (0.34)	0.57 (0.30)	0.65 (0.20)
		0.96 (0.36)	1.21 (0.48)	1.17 (0.41)	0.96 (0.33)	0.56 (0.29)	0.63 (0.20)
		0.96 (0.36)	1.20 (0.48)	1.16 (0.41)	0.96 (0.33)	0.55 (0.29)	0.62 (0.20)
		17.86	18.18	18.13	17.92	17.31	17.27
		(0.32)	(0.35)	(0.40)	(0.34)	(0.42)	(0.40)
	100	14.79	12.29	10.06	13.09	7.48 (0.86)	10.95
		(5.06)	(2.11)	(3.91)	(3.59)		(5.29)
		57.27	55.13	50.40	50.73	24.34	21.86
		(24.55)	(18.73)	(19.05)	(17.83)	(7.85)	(10.56)
		55.14	52.96	49.42	48.47	23.59	20.96
		(23.61)	(18.00)	(18.71)	(17.08)	(7.60)	(10.19)
		55.24	53.29	49.32	48.42	23.56	20.93
		(23.69)	(18.14)	(18.70)	(17.04)	(7.58)	(10.17)
		332.99	344.45	341.31	340.15	318.20	316.22
		(4.69)	(3.44)	(4.42)	(4.56)	(3.84)	(5.01)
		48.53	31.34	61.70	51.47	40.17	38.38
		(26.93)	(14.44)	(33.61)	(29.20)	(16.01)	(15.04)

Running Times - Low Setup Times, 1-10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.03 (0.01)	0.04 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.02 (0.01)
		0.02 (0.01)	0.04 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.02 (0.01)
		0.02 (0.01)	0.03 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.02 (0.01)
		2.29 (0.03)	2.30 (0.03)	2.30 (0.01)	2.30 (0.03)	2.25 (0.03)	2.27 (0.04)
		13.42	12.50	12.11	12.30	11.89	12.36
	20	(1.92)	(0.34)	(0.21)	(0.54)	(0.27)	(0.51)
		0.64 (0.18)	0.83 (0.19)	0.77 (0.22)	0.69 (0.16)	0.41 (0.16)	0.47 (0.19)
		0.63 (0.18)	0.80 (0.19)	0.75 (0.21)	0.67 (0.16)	0.40 (0.15)	0.46 (0.19)
		0.62 (0.18)	0.79 (0.19)	0.73 (0.20)	0.66 (0.16)	0.39 (0.15)	0.45 (0.18)
		14.05	14.24	14.22	14.02	13.30	13.54
	30	(0.34)	(0.35)	(0.31)	(0.33)	(0.46)	(0.46)
		35.92	33.82	28.81	33.37	27.41	26.89
		(10.39)	(9.16)	(6.98)	(7.80)	(6.26)	(6.25)
		2.32 (0.82)	2.18 (0.40)	1.97 (0.81)	2.15 (0.77)	1.22 (0.32)	1.17 (0.28)
		2.25 (0.80)	2.11 (0.39)	1.91 (0.79)	2.09 (0.75)	1.19 (0.32)	1.14 (0.27)
	100	2.23 (0.79)	2.10 (0.39)	1.90 (0.78)	2.07 (0.75)	1.18 (0.33)	1.13 (0.27)
		30.14	29.95	29.67	29.86	27.87	27.87
		(0.79)	(0.47)	(0.85)	(0.85)	(0.59)	(0.41)
		42.87	48.42	52.11	49.29	41.75	40.45
		(17.61)	(8.52)	(6.77)	(16.07)	(9.82)	(6.35)
		102.72	91.31	100.63	91.51	51.18	52.15
		(21.52)	(33.78)	(37.17)	(30.57)	(20.39)	(11.89)
		99.69	88.51	97.70	89.10	49.82	50.65
		(20.91)	(32.83)	(36.09)	(29.74)	(19.84)	(11.57)
		99.86	88.45	97.53	89.21	49.81	50.55
		(20.96)	(32.88)	(36.11)	(29.80)	(19.86)	(11.57)
		529.35	535.07	531.93	526.72	471.18	470.13
		(5.78)	(8.09)	(7.70)	(7.03)	(11.68)	(6.42)
		149.98	239.79	223.60	253.33	131.74	165.89
		(57.07)	(105.18)	(154.42)	(85.60)	(60.43)	(67.41)

Running Times - Low Setup Times, 1-10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.06 (0.01)	0.07 (0.01)	0.06 (0.01)	0.05 (0.01)	0.04 (0.01)	0.03 (0.01)
		0.06 (0.01)	0.06 (0.01)	0.05 (0.01)	0.05 (0.01)	0.04 (0.01)	0.03 (0.01)
		0.06 (0.01)	0.06 (0.01)	0.05 (0.01)	0.05 (0.01)	0.03 (0.01)	0.03 (0.01)
		4.57 (0.03)	4.58 (0.04)	4.54 (0.04)	4.52 (0.03)	4.46 (0.04)	4.48 (0.04)
		51.28 (2.36)	50.05 (1.38)	50.04 (0.69)	50.09 (1.30)	49.90 (0.35)	48.86 (0.44)
	20	1.62 (0.31)	1.47 (0.25)	1.37 (0.18)	1.56 (0.15)	0.89 (0.17)	0.90 (0.18)
		1.57 (0.30)	1.43 (0.24)	1.33 (0.17)	1.52 (0.15)	0.87 (0.17)	0.90 (0.18)
		1.55 (0.30)	1.41 (0.24)	1.32 (0.17)	1.50 (0.15)	0.85 (0.17)	0.87 (0.17)
		27.52 (0.45)	27.33 (0.36)	26.92 (0.27)	27.22 (0.28)	25.50 (0.39)	25.73 (0.47)
		188.51 (27.05)	165.27 (39.68)	171.37 (36.06)	178.39 (42.96)	115.85 (24.28)	122.94 (33.82)
	30	4.91 (1.11)	4.80 (0.62)	4.28 (0.70)	4.61 (1.08)	2.30 (0.76)	2.42 (0.52)
		4.76 (1.08)	4.67 (0.61)	4.17 (0.68)	4.46 (1.04)	2.24 (0.74)	2.42 (0.53)
		4.71 (1.06)	4.65 (0.61)	4.13 (0.68)	4.42 (1.03)	2.21 (0.73)	2.33 (0.50)
		59.63 (1.23)	59.43 (0.67)	58.37 (0.82)	58.65 (1.22)	53.36 (1.41)	54.00 (0.90)
		207.99 (70.58)	276.49 (105.26)	253.65 (89.78)	301.87 (88.35)	244.33 (49.59)	253.43 (59.77)
	100	237.56 (38.98)	207.11 (29.34)	191.05 (48.99)	177.97 (29.32)	107.14 (19.93)	109.12 (23.90)
		229.92 (37.79)	201.97 (28.53)	184.85 (47.28)	172.79 (28.44)	103.70 (19.39)	109.44 (24.03)
		229.78 (37.77)	201.50 (28.33)	184.39 (47.20)	172.44 (28.42)	103.43 (19.41)	106.27 (23.32)
		991.37 (9.30)	984.87 (7.45)	965.71 (13.22)	962.65 (11.65)	835.05 (13.07)	832.21 (10.92)
		915.05 (560.67)	1041.12 (615.49)	904.98 (585.50)	892.53 (354.43)	807.02 (507.58)	918.01 (575.16)

Running Times - High Setup Times, 1 Machine per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.00 (0.00)	0.01 (0.01)	0.00 (0.00)	0.01 (0.01)	0.00 (0.00)	0.00 (0.00)
		0.01 (0.01)	0.01 (0.01)	0.00 (0.00)	0.01 (0.01)	0.00 (0.00)	0.01 (0.01)
		0.00 (0.00)	0.00 (0.01)	0.00 (0.00)	0.00 (0.01)	0.00 (0.01)	0.00 (0.00)
		X	X	X	X	X	X
		2.41 (0.09)	2.42 (0.06)	2.40 (0.08)	2.42 (0.13)	2.39 (0.14)	2.34 (0.07)
	20	0.06 (0.00)	0.06 (0.00)	0.06 (0.01)	0.06 (0.00)	0.04 (0.01)	0.03 (0.01)
		0.06 (0.01)	0.06 (0.00)	0.05 (0.00)	0.06 (0.01)	0.03 (0.00)	0.03 (0.01)
		0.06 (0.00)	0.06 (0.00)	0.06 (0.00)	0.06 (0.00)	0.03 (0.01)	0.03 (0.01)
		X	X	X	X	X	X
		7.49 (2.32)	6.97 (1.90)	7.33 (1.70)	6.31 (2.03)	6.01 (1.42)	5.91 (1.34)
	30	0.18 (0.00)	0.17 (0.01)	0.16 (0.01)	0.16 (0.01)	0.09 (0.01)	0.09 (0.01)
		0.17 (0.00)	0.17 (0.00)	0.16 (0.01)	0.16 (0.01)	0.09 (0.01)	0.09 (0.01)
		0.17 (0.01)	0.17 (0.00)	0.16 (0.01)	0.16 (0.01)	0.09 (0.01)	0.09 (0.01)
		X	X	X	X	X	X
		10.28 (2.34)	11.48 (2.25)	9.56 (2.60)	12.30 (3.96)	8.09 (2.60)	9.45 (2.85)
	100	8.80 (0.05)	8.70 (0.06)	7.97 (0.27)	8.18 (0.23)	4.08 (0.34)	4.30 (0.62)
		8.63 (0.05)	8.55 (0.05)	7.74 (0.24)	7.91 (0.20)	3.99 (0.35)	4.14 (0.59)
		8.60 (0.05)	8.53 (0.06)	7.74 (0.25)	7.88 (0.18)	3.98 (0.36)	4.14 (0.60)
		X	X	X	X	X	X
		44.24 (6.24)	49.02 (15.80)	49.74 (20.87)	60.34 (29.15)	45.43 (12.51)	42.68 (13.22)

Running Times - High Setup Times, 1 Machine per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.00 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.00 (0.00)
		X	X	X	X	X	X
	20	11.71	11.65	11.64	11.64	11.49	11.59
		(0.40)	(0.30)	(0.22)	(0.22)	(0.13)	(0.51)
		0.12 (0.00)	0.13 (0.01)	0.12 (0.01)	0.11 (0.01)	0.07 (0.01)	0.07 (0.01)
		0.12 (0.00)	0.12 (0.00)	0.11 (0.01)	0.11 (0.00)	0.07 (0.01)	0.07 (0.01)
	30	0.12 (0.00)	0.12 (0.00)	0.11 (0.01)	0.11 (0.00)	0.07 (0.00)	0.07 (0.00)
		X	X	X	X	X	X
		33.77	28.72	31.36	28.38	29.27	26.28
		(8.75)	(7.84)	(9.94)	(3.39)	(6.80)	(5.08)
	100	0.35 (0.01)	0.35 (0.01)	0.33 (0.01)	0.33 (0.01)	0.19 (0.01)	0.19 (0.02)
		0.34 (0.00)	0.34 (0.00)	0.32 (0.01)	0.32 (0.01)	0.18 (0.02)	0.19 (0.02)
		0.34 (0.00)	0.34 (0.00)	0.32 (0.01)	0.32 (0.01)	0.18 (0.02)	0.19 (0.01)
		X	X	X	X	X	X
		44.90	51.47	46.27	48.96	41.76	47.89
		(9.88)	(11.26)	(20.36)	(12.29)	(6.47)	(11.89)
		16.82	16.86	15.32	15.51	7.54 (0.46)	7.78 (0.54)
		(0.06)	(0.05)	(0.25)	(0.27)	7.40 (0.47)	7.60 (0.53)
		16.54	16.54	15.02	15.26	7.41 (0.47)	7.62 (0.52)
		(0.05)	(0.08)	(0.25)	(0.23)	X	X
		16.51	16.50	14.98	15.19	240.86	259.17
		(0.07)	(0.06)	(0.25)	(0.25)	(76.00)	(95.28)
		X	X	X	X		
		293.49	238.19	277.49	284.87		
		(51.68)	(65.93)	(67.82)	(58.88)		

Running Times - High Setup Times, 1 Machine per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		X	X	X	X	X	X
	20	49.59	50.27	49.52	49.52	49.49	49.03
		(0.58)	(0.91)	(0.72)	(0.59)	(0.79)	(0.64)
		0.24 (0.01)	0.24 (0.01)	0.23 (0.01)	0.23 (0.01)	0.14 (0.01)	0.13 (0.01)
		0.24 (0.01)	0.24 (0.01)	0.23 (0.01)	0.22 (0.01)	0.13 (0.01)	0.13 (0.01)
	30	0.24 (0.01)	0.24 (0.01)	0.22 (0.01)	0.22 (0.00)	0.13 (0.01)	0.13 (0.01)
		X	X	X	X	X	X
		144.85	132.05	139.79	130.37	131.19	117.21
		(25.06)	(16.89)	(19.58)	(21.90)	(22.02)	(23.04)
	100	0.70 (0.01)	0.70 (0.01)	0.66 (0.01)	0.66 (0.02)	0.38 (0.02)	0.38 (0.02)
		0.68 (0.00)	0.68 (0.00)	0.64 (0.01)	0.64 (0.02)	0.37 (0.02)	0.37 (0.02)
		0.68 (0.00)	0.68 (0.00)	0.64 (0.01)	0.64 (0.02)	0.37 (0.02)	0.37 (0.02)
		X	X	X	X	X	X
		247.31	244.55	224.04	221.76	200.89	194.76
		(59.04)	(51.08)	(39.66)	(55.77)	(34.28)	(35.37)
		33.69	33.71	30.46	31.00	15.39	15.44
		(0.09)	(0.08)	(0.34)	(0.43)	(0.61)	(0.53)
		33.08	33.14	30.09	30.38	15.04	15.12
		(0.11)	(0.08)	(0.32)	(0.42)	(0.58)	(0.51)
		32.97	32.98	29.95	30.34	15.06	15.21
		(0.13)	(0.08)	(0.31)	(0.41)	(0.61)	(0.50)
		X	X	X	X	X	X
		1657.53	1581.55	1485.70	1560.72	849.33	1053.72
		(297.85)	(237.92)	(454.80)	(182.92)	(447.55)	(488.55)

Running Times - High Setup Times, 2 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)	0.01 (0.01)
		0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)
		X	X	X	X	X	X
		2.40 (0.05)	2.42 (0.06)	2.46 (0.10)	2.45 (0.12)	2.43 (0.14)	2.38 (0.07)
	20	0.12 (0.00)	0.12 (0.01)	0.12 (0.01)	0.12 (0.00)	0.06 (0.01)	0.07 (0.02)
		0.12 (0.00)	0.13 (0.01)	0.11 (0.01)	0.11 (0.00)	0.06 (0.01)	0.07 (0.02)
		0.12 (0.00)	0.12 (0.00)	0.11 (0.01)	0.11 (0.01)	0.06 (0.01)	0.07 (0.01)
		X	X	X	X	X	X
		8.25 (3.76)	6.89 (2.17)	7.70 (1.50)	6.96 (1.78)	6.12 (1.38)	6.11 (1.64)
	30	0.36 (0.01)	0.36 (0.01)	0.33 (0.02)	0.34 (0.02)	0.19 (0.02)	0.18 (0.01)
		0.35 (0.00)	0.35 (0.00)	0.32 (0.01)	0.33 (0.02)	0.19 (0.01)	0.18 (0.02)
		0.35 (0.01)	0.35 (0.01)	0.32 (0.02)	0.33 (0.02)	0.19 (0.02)	0.18 (0.01)
		X	X	X	X	X	X
		11.28 (3.87)	9.58 (1.85)	10.37 (2.76)	10.25 (3.09)	7.97 (2.68)	8.61 (1.97)
	100	17.86 (0.07)	17.74 (0.10)	15.95 (0.27)	16.56 (0.34)	8.19 (0.87)	8.17 (0.69)
		17.47 (0.09)	17.46 (0.12)	15.54 (0.27)	16.03 (0.32)	8.02 (0.86)	7.87 (0.67)
		17.44 (0.08)	17.42 (0.09)	15.51 (0.25)	16.00 (0.31)	8.05 (0.86)	7.89 (0.68)
		X	X	X	X	X	X
		69.85 (21.72)	71.61 (27.73)	60.48 (14.97)	71.78 (20.69)	25.99 (10.66)	30.29 (18.12)

Running Times - High Setup Times, 2 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.00)	0.01 (0.00)	0.01 (0.00)
		0.02 (0.00)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.00)
		X	X	X	X	X	X
	20	12.08	11.99	12.15	11.66	11.60	11.73
		(0.50)	(0.60)	(0.65)	(0.18)	(0.39)	(0.46)
		0.25 (0.01)	0.25 (0.01)	0.24 (0.01)	0.23 (0.01)	0.14 (0.01)	0.14 (0.02)
		0.24 (0.00)	0.24 (0.00)	0.23 (0.01)	0.23 (0.01)	0.14 (0.01)	0.14 (0.02)
	30	0.24 (0.00)	0.24 (0.00)	0.23 (0.01)	0.22 (0.01)	0.14 (0.01)	0.14 (0.01)
		X	X	X	X	X	X
		36.05	34.78	34.44	27.92	25.79	29.60
		(9.90)	(7.82)	(8.24)	(5.10)	(4.06)	(8.31)
	100	0.72 (0.01)	0.72 (0.01)	0.66 (0.02)	0.68 (0.03)	0.41 (0.03)	0.40 (0.03)
		0.70 (0.00)	0.70 (0.00)	0.65 (0.02)	0.65 (0.02)	0.40 (0.03)	0.39 (0.03)
		0.70 (0.00)	0.70 (0.01)	0.65 (0.02)	0.65 (0.02)	0.39 (0.03)	0.39 (0.03)
		X	X	X	X	X	X
		51.59	48.19	50.27	55.79	47.83	35.57
		(6.46)	(11.34)	(13.48)	(11.22)	(9.69)	(8.22)
		34.42	34.37	30.94	31.75	15.81	15.83
		(0.10)	(0.14)	(0.38)	(0.47)	(0.96)	(0.71)
		33.98	33.73	30.39	31.14	15.51	15.46
		(0.17)	(0.14)	(0.38)	(0.47)	(0.94)	(0.70)
		33.92	33.69	30.30	31.25	15.57	15.47
		(0.19)	(0.12)	(0.36)	(0.53)	(0.95)	(0.69)
		X	X	X	X	X	X
		208.86	255.44	272.92	219.88	116.34	133.11
		(54.05)	(117.87)	(122.36)	(97.51)	(46.63)	(52.29)

Running Times - High Setup Times, 2 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.03 (0.01)	0.03 (0.01)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.00)
		0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.01)
		0.03 (0.01)	0.03 (0.00)	0.03 (0.00)	0.03 (0.00)	0.02 (0.00)	0.02 (0.00)
		X	X	X	X	X	X
	20	49.92	51.74	50.65	50.62	50.28	50.89
		(0.47)	(1.51)	(0.85)	(1.37)	(2.87)	(3.42)
		0.50 (0.01)	0.50 (0.00)	0.47 (0.01)	0.47 (0.00)	0.28 (0.02)	0.27 (0.02)
		0.48 (0.00)	0.48 (0.00)	0.45 (0.01)	0.45 (0.01)	0.27 (0.02)	0.26 (0.02)
	30	0.48 (0.00)	0.49 (0.00)	0.46 (0.01)	0.45 (0.01)	0.27 (0.02)	0.26 (0.02)
		X	X	X	X	X	X
		200.28	144.75	166.58	156.39	141.65	136.07
		(42.55)	(32.69)	(40.34)	(26.91)	(36.21)	(35.38)
	100	1.43 (0.01)	1.44 (0.01)	1.33 (0.04)	1.35 (0.03)	0.80 (0.04)	0.78 (0.05)
		1.41 (0.01)	1.40 (0.01)	1.30 (0.03)	1.32 (0.03)	0.78 (0.04)	0.76 (0.05)
		1.40 (0.01)	1.41 (0.01)	1.29 (0.04)	1.32 (0.03)	0.78 (0.05)	0.76 (0.05)
		X	X	X	X	X	X
		255.43	277.59	277.66	255.13	209.56	250.10
		(85.73)	(49.49)	(92.73)	(55.80)	(39.81)	(32.66)
		68.96	69.02	62.34	63.14	31.15	31.70
		(0.27)	(0.15)	(0.64)	(0.43)	(1.24)	(1.37)
		67.59	67.79	61.52	61.70	30.28	31.18
		(0.21)	(0.22)	(0.63)	(0.48)	(1.25)	(1.38)
		67.51	67.66	61.42	61.62	30.37	31.23
		(0.21)	(0.23)	(0.65)	(0.44)	(1.28)	(1.37)
		X	X	X	X	X	X
		802.80	1653.27	1293.09	1400.97	857.23	755.18
		(217.90)	(563.63)	(712.46)	(613.51)	(237.74)	(183.01)

Running Times - High Setup Times, 10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	0.71 (0.01)	0.71 (0.01)	0.68 (0.02)	0.67 (0.02)	0.37 (0.09)	0.42 (0.07)
		0.69 (0.00)	0.69 (0.00)	0.66 (0.01)	0.65 (0.02)	0.36 (0.08)	0.41 (0.06)
		0.69 (0.00)	0.70 (0.01)	0.67 (0.03)	0.65 (0.02)	0.36 (0.09)	0.41 (0.06)
		X	X	X	X	X	X
	30	7.57 (1.81)	10.43 (2.97)	9.41 (3.29)	10.80 (3.34)	5.64 (2.27)	5.57 (2.30)
		2.11 (0.01)	2.10 (0.01)	1.95 (0.05)	1.98 (0.07)	1.19 (0.09)	1.10 (0.13)
		2.06 (0.01)	2.05 (0.01)	1.90 (0.05)	1.93 (0.07)	1.16 (0.08)	1.07 (0.12)
		2.05 (0.01)	2.05 (0.01)	1.90 (0.06)	1.93 (0.07)	1.16 (0.09)	1.07 (0.13)
	100	X	X	X	X	X	X
		15.19 (3.64)	14.10 (3.82)	12.28 (2.26)	13.11 (4.33)	9.51 (1.77)	10.03 (3.33)
		101.11 (0.92)	100.07 (0.42)	90.88 (2.53)	93.78 (1.51)	46.41 (3.38)	49.75 (3.27)
		98.86 (0.73)	98.91 (0.78)	88.72 (2.39)	90.88 (1.45)	45.44 (3.31)	48.00 (3.16)
		99.13 (0.97)	99.37 (0.57)	88.57 (2.38)	90.85 (1.40)	45.66 (3.37)	48.14 (3.15)
		X	X	X	X	X	X
		60.99 (16.55)	63.42 (25.14)	66.84 (19.49)	63.38 (24.31)	44.87 (12.35)	52.53 (12.71)

Running Times - High Setup Times, 10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	1.47 (0.00)	1.47 (0.01)	1.39 (0.04)	1.41 (0.04)	0.85 (0.06)	0.83 (0.07)
		1.43 (0.00)	1.44 (0.03)	1.36 (0.05)	1.38 (0.04)	0.83 (0.06)	0.81 (0.07)
		1.43 (0.01)	1.43 (0.01)	1.36 (0.05)	1.38 (0.04)	0.83 (0.07)	0.81 (0.07)
		X	X	X	X	X	X
		43.96	36.94	39.28	36.19	23.78	18.88
		(14.15)	(9.08)	(6.51)	(11.43)	(5.90)	(3.58)
	30	4.36 (0.02)	4.35 (0.02)	4.12 (0.11)	4.01 (0.13)	2.43 (0.22)	2.47 (0.19)
		4.27 (0.02)	4.25 (0.02)	4.02 (0.12)	3.91 (0.12)	2.37 (0.22)	2.41 (0.19)
		4.28 (0.02)	4.25 (0.02)	4.03 (0.11)	3.92 (0.13)	2.37 (0.22)	2.41 (0.19)
		X	X	X	X	X	X
		68.46	67.41	70.83	79.37	38.42	35.60
		(20.68)	(10.02)	(18.06)	(19.56)	(5.27)	(9.71)
	100	199.68	200.52	182.41	185.64	92.04	95.77
		(0.72)	(0.45)	(3.33)	(2.38)	(4.55)	(5.97)
		196.65	196.01	178.84	182.80	90.28	93.55
		(0.68)	(0.69)	(3.28)	(2.42)	(4.50)	(5.81)
		196.87	195.94	178.89	182.88	90.26	94.16
		(0.75)	(0.75)	(3.37)	(2.39)	(4.44)	(5.95)
		X	X	X	X	X	X
		230.56	222.70	261.57	388.45	303.78	281.01
		(136.60)	(136.93)	(140.02)	(141.48)	(49.91)	(39.53)

Running Times - High Setup Times, 10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	20	2.98 (0.02)	2.97 (0.01)	2.78 (0.08)	2.83 (0.06)	1.72 (0.13)	1.60 (0.13)
		2.90 (0.01)	2.90 (0.01)	2.71 (0.07)	2.74 (0.04)	1.68 (0.13)	1.56 (0.13)
		2.91 (0.01)	2.91 (0.01)	2.71 (0.07)	2.75 (0.04)	1.69 (0.13)	1.56 (0.13)
		X	X	X	X	X	X
		214.18	195.20	199.11	192.54	86.40	84.88
		(82.84)	(66.72)	(22.14)	(42.25)	(17.74)	(20.70)
	30	8.90 (0.04)	8.86 (0.02)	8.32 (0.14)	8.31 (0.16)	5.08 (0.26)	4.96 (0.29)
		8.66 (0.04)	8.64 (0.03)	8.15 (0.14)	8.09 (0.16)	4.94 (0.25)	4.86 (0.31)
		8.66 (0.03)	8.65 (0.03)	8.17 (0.15)	8.11 (0.15)	4.94 (0.25)	4.86 (0.30)
		X	X	X	X	X	X
		233.92	352.25	337.94	363.30	204.02	194.59
		(88.26)	(83.67)	(82.52)	(61.53)	(64.66)	(43.08)
	100	406.50	404.21	370.41	373.05	196.54	187.76
		(1.23)	(1.17)	(3.99)	(4.05)	(10.69)	(8.90)
		397.65	397.78	363.21	364.70	191.16	184.47
		(1.02)	(1.52)	(3.49)	(4.00)	(10.28)	(8.89)
		398.08	398.32	363.33	365.04	192.18	185.02
		(1.15)	(1.35)	(3.78)	(4.05)	(10.20)	(8.77)
		X	X	X	X	X	X
		815.83	1288.25	1378.56	1914.53	1553.61	1591.37
		(248.74)	(889.99)	(686.84)	(1231.65)	(491.88)	(346.88)

Running Times - High Setup Times, 1-4 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)
		0.02 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.01)
		0.01 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)
		X	X	X	X	X	X
		2.63 (0.47)	2.53 (0.32)	2.57 (0.44)	2.47 (0.23)	2.36 (0.03)	2.33 (0.02)
	20	0.13 (0.03)	0.17 (0.04)	0.13 (0.03)	0.17 (0.03)	0.09 (0.03)	0.08 (0.02)
		0.13 (0.03)	0.16 (0.04)	0.13 (0.03)	0.16 (0.03)	0.09 (0.03)	0.08 (0.02)
		0.13 (0.03)	0.16 (0.04)	0.13 (0.03)	0.16 (0.03)	0.09 (0.03)	0.08 (0.02)
		X	X	X	X	X	X
		8.43 (2.40)	8.24 (1.98)	7.07 (1.55)	7.51 (1.88)	5.32 (1.04)	5.37 (2.21)
	30	0.36 (0.09)	0.50 (0.15)	0.41 (0.06)	0.48 (0.14)	0.27 (0.11)	0.27 (0.10)
		0.35 (0.09)	0.49 (0.15)	0.40 (0.06)	0.47 (0.13)	0.26 (0.10)	0.26 (0.10)
		0.35 (0.09)	0.49 (0.15)	0.40 (0.06)	0.47 (0.13)	0.26 (0.11)	0.26 (0.10)
		X	X	X	X	X	X
		11.62 (3.88)	9.49 (2.20)	12.00 (4.96)	10.65 (2.73)	11.51 (3.57)	7.62 (2.21)
	100	25.92 (4.38)	21.51 (4.28)	19.78 (7.14)	21.43 (6.33)	10.57 (2.19)	11.88 (4.52)
		25.15 (4.28)	21.37 (4.25)	19.28 (6.96)	20.84 (6.14)	10.37 (2.15)	11.47 (4.37)
		25.36 (4.33)	21.29 (4.26)	19.24 (6.91)	20.86 (6.15)	10.38 (2.17)	11.48 (4.37)
		X	X	X	X	X	X
		37.64 (33.98)	47.68 (22.43)	40.01 (21.10)	45.67 (19.80)	37.29 (25.15)	47.36 (21.40)

Running Times - High Setup Times, 1-4 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.03 (0.01)	0.02 (0.00)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)
		0.02 (0.00)	0.02 (0.00)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.00)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)
		X	X	X	X	X	X
	20	12.89	12.73	12.46	12.24	12.81	12.04
		(0.77)	(0.64)	(0.58)	(0.41)	(2.09)	(0.29)
		0.33 (0.08)	0.30 (0.05)	0.32 (0.07)	0.32 (0.07)	0.18 (0.04)	0.18 (0.05)
		0.33 (0.08)	0.30 (0.05)	0.31 (0.06)	0.31 (0.07)	0.18 (0.04)	0.18 (0.05)
	30	0.32 (0.07)	0.30 (0.05)	0.31 (0.06)	0.31 (0.07)	0.18 (0.04)	0.17 (0.05)
		X	X	X	X	X	X
		34.35	39.37	37.84	38.19	27.36	28.07
		(6.86)	(11.44)	(10.21)	(5.35)	(7.90)	(7.30)
	100	0.88 (0.12)	1.09 (0.19)	0.84 (0.19)	0.83 (0.24)	0.52 (0.15)	0.51 (0.11)
		0.85 (0.12)	1.07 (0.19)	0.82 (0.18)	0.81 (0.23)	0.50 (0.14)	0.50 (0.11)
		0.86 (0.11)	1.06 (0.18)	0.82 (0.18)	0.81 (0.24)	0.51 (0.14)	0.50 (0.11)
		X	X	X	X	X	X
		56.16	50.82	53.97	51.94	38.02	44.36
		(16.37)	(17.12)	(16.83)	(14.65)	(11.54)	(17.86)
		45.07	43.59	38.63	42.15	21.23	19.66
		(8.65)	(11.70)	(9.49)	(6.61)	(3.96)	(3.74)
		44.51	42.52	37.94	41.61	20.72	19.24
		(8.52)	(11.40)	(9.36)	(6.53)	(3.88)	(3.68)
		44.23	42.43	37.97	41.26	20.78	19.25
		(8.46)	(11.36)	(9.38)	(6.48)	(3.85)	(3.68)
		X	X	X	X	X	X
		172.74	205.35	232.98	214.74	142.46	185.14
		(115.64)	(100.99)	(88.41)	(131.00)	(86.83)	(105.12)

Running Times - High Setup Times, 1-4 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.03 (0.01)	0.02 (0.01)
		0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.03 (0.01)	0.02 (0.00)
		0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.04 (0.01)	0.03 (0.01)	0.02 (0.00)
		X	X	X	X	X	X
	20	50.31	51.11	51.55	50.86	51.04	49.95
		(0.75)	(2.08)	(2.19)	(1.15)	(2.10)	(0.77)
		0.62 (0.08)	0.67 (0.12)	0.65 (0.11)	0.59 (0.10)	0.35 (0.06)	0.36 (0.07)
		0.61 (0.08)	0.66 (0.12)	0.63 (0.11)	0.57 (0.09)	0.34 (0.06)	0.36 (0.07)
	30	0.61 (0.08)	0.66 (0.12)	0.63 (0.11)	0.57 (0.10)	0.34 (0.06)	0.36 (0.07)
		X	X	X	X	X	X
		173.04	151.75	173.59	159.21	120.33	113.09
		(41.76)	(24.67)	(38.49)	(21.87)	(30.78)	(29.25)
	100	1.74 (0.30)	1.82 (0.24)	1.77 (0.32)	1.69 (0.20)	1.04 (0.12)	0.93 (0.12)
		1.70 (0.29)	1.78 (0.23)	1.72 (0.31)	1.65 (0.20)	1.02 (0.11)	0.91 (0.12)
		1.71 (0.30)	1.78 (0.23)	1.73 (0.31)	1.65 (0.20)	1.01 (0.11)	0.91 (0.12)
		X	X	X	X	X	X
		274.97	245.01	233.63	296.78	223.59	204.42
		(73.67)	(53.38)	(48.44)	(57.42)	(60.24)	(57.22)
		81.89	91.94	79.56	78.47	43.47	42.32
		(14.48)	(17.99)	(11.86)	(21.19)	(3.48)	(6.32)
		80.56	90.57	77.70	77.18	42.28	41.40
		(14.31)	(17.76)	(11.55)	(20.87)	(3.44)	(6.20)
		80.59	90.23	77.74	77.14	42.34	41.50
		(14.22)	(17.55)	(11.53)	(20.86)	(3.41)	(6.26)
		X	X	X	X	X	X
		813.38	803.98	1013.75	1093.57	864.40	950.72
		(574.26)	(362.50)	(648.72)	(475.54)	(609.19)	(401.01)

Running Times - High Setup Times, 1-10 Machines per Stage, 2 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.01)	0.01 (0.00)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.00)	0.01 (0.01)
		0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.02 (0.01)	0.01 (0.01)	0.01 (0.01)
		X	X	X	X	X	X
		2.68 (0.44)	2.47 (0.09)	2.58 (0.32)	2.52 (0.12)	2.41 (0.09)	2.45 (0.14)
	20	0.39 (0.14)	0.38 (0.19)	0.39 (0.18)	0.33 (0.12)	0.21 (0.08)	0.23 (0.07)
		0.38 (0.14)	0.37 (0.19)	0.38 (0.17)	0.32 (0.12)	0.20 (0.08)	0.22 (0.07)
		0.38 (0.13)	0.37 (0.18)	0.38 (0.17)	0.32 (0.12)	0.20 (0.08)	0.22 (0.06)
		X	X	X	X	X	X
		7.72 (2.15)	9.67 (2.82)	8.88 (2.78)	8.42 (2.32)	6.82 (2.06)	6.73 (2.07)
	30	0.99 (0.41)	0.89 (0.18)	1.07 (0.42)	1.01 (0.34)	0.50 (0.27)	0.75 (0.18)
		0.97 (0.40)	0.86 (0.17)	1.04 (0.41)	0.98 (0.34)	0.49 (0.27)	0.73 (0.18)
		0.97 (0.40)	0.86 (0.17)	1.04 (0.42)	0.99 (0.34)	0.49 (0.27)	0.73 (0.18)
		X	X	X	X	X	X
		13.17 (3.24)	12.22 (3.98)	10.86 (4.01)	12.14 (2.58)	8.97 (3.12)	9.55 (3.30)
	100	53.36 (27.39)	56.29 (9.26)	47.26 (16.56)	41.42 (11.32)	24.15 (13.28)	28.23 (11.16)
		51.85 (26.64)	55.97 (9.30)	45.55 (16.00)	40.62 (11.13)	23.70 (13.06)	27.46 (10.89)
		51.79 (26.61)	55.80 (9.14)	45.40 (15.97)	40.63 (11.13)	23.94 (13.19)	27.52 (10.95)
		X	X	X	X	X	X
		36.25 (15.76)	41.18 (17.17)	48.41 (21.05)	50.59 (27.26)	45.85 (17.16)	37.54 (16.63)

Running Times - High Setup Times, 1-10 Machines per Stage, 4 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.03 (0.01)	0.04 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.02 (0.01)
		0.03 (0.01)	0.03 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.01 (0.01)
		0.03 (0.01)	0.04 (0.01)	0.03 (0.01)	0.03 (0.01)	0.02 (0.01)	0.02 (0.00)
		X	X	X	X	X	X
	20	12.82	12.68	12.58	12.61	12.23	12.29
		(0.60)	(0.31)	(0.72)	(0.31)	(0.24)	(0.46)
		0.73 (0.19)	0.78 (0.22)	0.76 (0.26)	0.74 (0.17)	0.42 (0.07)	0.46 (0.11)
		0.70 (0.19)	0.76 (0.22)	0.73 (0.25)	0.72 (0.16)	0.41 (0.07)	0.44 (0.11)
	30	0.71 (0.19)	0.76 (0.22)	0.73 (0.25)	0.72 (0.16)	0.41 (0.07)	0.45 (0.11)
		X	X	X	X	X	X
		41.52	36.10	34.29	37.28	24.60	26.02
		(12.00)	(7.81)	(9.00)	(12.16)	(6.23)	(7.03)
	100	2.32 (0.60)	2.29 (0.45)	2.22 (0.48)	2.41 (0.55)	1.53 (0.63)	1.12 (0.35)
		2.27 (0.59)	2.23 (0.44)	2.17 (0.47)	2.36 (0.53)	1.49 (0.61)	1.09 (0.35)
		2.27 (0.59)	2.24 (0.44)	2.16 (0.46)	2.37 (0.54)	1.50 (0.61)	1.10 (0.35)
		X	X	X	X	X	X
		60.85	55.05	52.58	53.68	40.88	48.62
		(24.18)	(19.11)	(16.48)	(9.70)	(8.14)	(7.79)
		101.17	101.67	94.98	89.60	46.39	60.61
		(38.18)	(28.96)	(22.63)	(24.02)	(15.54)	(11.69)
		99.42	99.21	93.51	87.90	45.31	59.42
		(37.34)	(28.26)	(22.25)	(23.54)	(15.22)	(11.46)
		99.13	99.19	93.58	87.86	45.46	59.70
		(37.35)	(28.29)	(22.27)	(23.57)	(15.25)	(11.44)
		X	X	X	X	X	X
		150.21	192.88	201.32	256.12	189.81	165.82
		(65.67)	(109.25)	(68.91)	(138.23)	(112.71)	(58.71)

Running Times - High Setup Times, 1-10 Machines per Stage, 8 Stages

		Skipping Prob.					
		0.00		0.05		0.40	
		Processing Range		Processing Range		Processing Range	
		L	H	L	H	L	H
# of Jobs	6	0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.04 (0.01)	0.04 (0.01)
		0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.03 (0.01)	0.04 (0.01)
		0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.06 (0.01)	0.04 (0.01)	0.04 (0.01)
		X	X	X	X	X	X
		51.44 (1.73)	52.48 (2.36)	52.17 (2.44)	51.75 (2.13)	50.83 (3.00)	50.39 (0.64)
	20	1.60 (0.44)	1.38 (0.31)	1.32 (0.36)	1.36 (0.22)	0.85 (0.13)	0.85 (0.18)
		1.56 (0.42)	1.34 (0.30)	1.29 (0.35)	1.32 (0.22)	0.83 (0.13)	0.83 (0.17)
		1.57 (0.42)	1.34 (0.30)	1.29 (0.35)	1.32 (0.22)	0.83 (0.13)	0.83 (0.17)
		X	X	X	X	X	X
		170.21 (42.39)	176.13 (34.49)	168.23 (41.26)	167.71 (37.75)	123.68 (34.77)	143.64 (55.01)
	30	4.52 (0.61)	4.56 (0.82)	4.38 (0.95)	4.23 (1.08)	2.50 (0.50)	2.69 (0.53)
		4.40 (0.60)	4.46 (0.81)	4.26 (0.92)	4.13 (1.05)	2.43 (0.48)	2.63 (0.52)
		4.42 (0.60)	4.46 (0.80)	4.27 (0.93)	4.14 (1.06)	2.44 (0.49)	2.63 (0.53)
		X	X	X	X	X	X
		222.61 (99.44)	297.46 (102.17)	296.26 (70.48)	272.86 (85.18)	217.92 (47.94)	219.86 (47.68)
	100	205.95 (31.48)	198.99 (38.37)	197.22 (40.29)	205.75 (46.85)	97.65 (27.74)	99.87 (21.53)
		203.20 (31.02)	194.86 (37.57)	193.13 (39.53)	202.53 (45.89)	95.29 (27.02)	97.29 (20.91)
		203.19 (31.04)	194.79 (37.66)	192.91 (39.49)	202.82 (46.43)	95.54 (27.01)	97.72 (21.00)
		X	X	X	X	X	X
		568.14 (262.80)	933.34 (374.74)	825.33 (365.61)	1426.63 (1023.01)	862.18 (629.05)	775.86 (384.59)

References

- Agnetis, A., A. Pacifici, F. Rossi, M. Lucertini, S. Nicoletti, F. Nicolo, G. Oriolo, D. Pacciarelli, and E. Pesaro. "Scheduling of flexible flow lines in an automobile assembly plant." *European Journal of Operational Research*. Vol. 97, 1997, pp. 348-362.
- Ahmadi, R.H. and U. Bagchi. "Lower bounds for single-machine scheduling problems." *Naval Research Logistics*. Vol. 37, 1990, pp. 967-979.
- Allahverdi, A., J.N.D. Gupta, and T. Aldowaisan. "A review of scheduling research involving setup considerations." *OMEGA*. Vol. 27, 1999, pp. 219-239.
- Asano, M. and H. Ohta. "Single machine scheduling using dominance relation to minimize earliness subject to ready and due times." *International Journal of Production Economics*. Vol. 44, 1996, pp. 35-43.
- Babel, L., H. Kellerer, and V. Kotov. "The k -partitioning problem." *Mathematical Methods of Operations Research*. Vol. 47, 1998, pp. 59-82.
- Baker, K. *Introduction to Sequencing and Scheduling*. New York: John Wiley, 1974.
- Balakrishnan, N., J.J. Kanet, and S. V. Sridharan. "Early/tardy scheduling with sequence dependent setups on uniform parallel machines." *Computers and Operations Research*. Vol. 26, 1999, pp. 127-141.
- Bartal, Y., A. Fiat, H. Karloff, and R. Vohra. "New algorithms for an ancient scheduling problem." *Journal of Computer and System Sciences*. Vol. 51, 1995, pp. 359-366.
- Bean, J.C. "Genetic algorithms and random keys for sequencing and optimization." *ORSA Journal on Computing*. Vol. 6, 1994, pp. 154-160.
- Ben-Daya, M. and M. Al-Fawzan. "A tabu search approach for the flow shop scheduling problem." *European Journal of Operational Research*. Vol. 109, 1998, pp. 88-95.
- Bianco, L., S. Ricciardelli, G. Rinaldi, and A. Sassano. "Scheduling tasks with sequence-dependent processing times." *Naval Research Logistics*. Vol. 35, 1988, pp. 177-184.
- Bitran, G. R. and S. M. Gilbert. "Sequencing production on parallel machines with two magnitudes of sequence dependent set-up cost." *Journal of Manufacturing and Operations Management*. Vol. 3, 1990, pp. 24-52.

- Blazewicz, J., K.H. Ecker, E. Pesch, G. Schmidt, and J. Weglaz. *Scheduling Computer and Manufacturing Processes*. Berlin: Springer-Verlag, 1996.
- Blocher, J.D. and S. Chand. "Scheduling of parallel processors: a posterior bound on LPT sequencing and a two-step algorithm." *Naval Research Logistics*. Vol. 38, 1991, pp. 273-287.
- Brah, S.A. and J.L. Hunsucker. "Branch and bound algorithm for the flow shop with multiple processors." *European Journal of Operational Research*. Vol. 51, 1991, pp. 88-99.
- Brah, S.A. and L.L. Loo. "Heuristics for scheduling in a flow shop with multiple processors." *European Journal of Operational Research*. Vol. 113, 1999, pp. 113-122.
- Campbell, H.G., R.A. Dudek, and M.L. Smith. "A heuristic algorithm for the n job, m machine sequencing problem." *Management Science*. Vol. 16, No. 10, 1970, pp. B630-B637.
- Carrier, J. "Scheduling jobs with release dates and tails on identical machines to minimize the makespan." *European Journal of Operational Research*. Vol. 29, 1987, pp. 298-306.
- Chand, S., R. Traub, and R. Uzsoy. "Single-machine scheduling with dynamic arrivals: decomposition results and an improved algorithm." *Naval Research Logistics*. Vol. 43, 1996, pp. 709-719.
- Chen, B. "Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage." *Journal of the Operational Research Society*. Vol. 46, 1995, pp. 234-244.
- Chudzik, K. and A. Janiak. "Single machine scheduling with job ready and setup times – genetic approach." *Fourth International Symposium on Methods and Models in Automation and Robotics*. Międzyzdroje, Poland. August 1997, pp. 1071-1075.
- Coffman Jr., E.G. and R. Sethi. "A generalized bound on LPT sequencing." *RAIRO-Informatique*. Vol. 10, 1976, pp. 17-25.
- Coffman Jr., E.G., M.R. Garey, and D.S. Johnson. "An application of bin-packing to multiprocessor scheduling." *SIAM Journal on Computing*. Vol. 7, No. 1, 1978, pp. 1-17.
- Corrêa, R. C, A. Ferreira, and P. Rebreyend. "Scheduling mulitprocessor tasks with genetic algorithms." *IEEE Transactions on Parallel and Distributed Systems* Vol. 10, No. 8, 1999, pp. 825-837.

- Corwin, B.D. and A.O. Esogbue. "Two machine flow shop scheduling problems with sequence dependent setup times: a dynamic programming approach." *Naval Research Logistics Quarterly*. Vol. 21, 1974, pp. 515-524.
- Dannenbring, D.G. "An evaluation of flow shop sequencing heuristics." *Management Science*. Vol. 23, No. 11, 1977, pp. 1174-1182.
- Davis, L. and M. Streenstrup. "Genetic algorithms and simulated annealing: an overview." *Genetic Algorithms and Simulated Annealing*. Ed. L. Davis. London: Pitman, 1987, pp. 1-11.
- Deogun, J. S. "On scheduling with ready times to minimize mean flow time." *The Computer Journal*. Vol. 26, No. 4, 1983, pp. 320-328.
- Dessouky, M. I. And J. S. Deogun. "Sequencing jobs with unequal ready times to minimize mean flow time." *SIAM Journal on Computing*. Vol. 10, No. 1, 1981, pp. 192-202.
- Dessouky, M.M., M.I. Dessouky, and S.K. Verma. "Flowshop scheduling with identical jobs and uniform parallel machines." *European Journal of Operational Research*. Vol. 109, 1998, pp. 620-631.
- Ding, F.-Y. and D. Kittichatphayak. "Heuristics for scheduling flexible flow lines". *Computers and Industrial Engineering*. Vol. 26, No. 1, 1994, pp. 27-34.
- Drozdzowski, M. "Scheduling multiprocessor tasks – an overview." *European Journal of Operational Research*. Vol. 94, 1996, pp. 215-230.
- Dudek, R.A. and O.F. Teuton. "Development of M -stage decision rule for scheduling n jobs through M machines." *Operations Research*, Vol. 12, No. 3, 1964, pp. 471-497.
- Dumas, Y., J. Desrosiers, E. Gelinas, and M.M. Solomon. "An optimal algorithm for the traveling salesman problem with time windows." *Operations Research*, Vol. 43, No. 2, 1995, pp. 367-371.
- Elmaghraby, S.E. and S.H. Park. "Scheduling jobs on a number of identical machines." *AIIE Transactions*. Vol. 6, No. 1, 1974, pp. 1-13.
- Freisen, D.K. "Tighter bounds for the MULTIFIT processor scheduling algorithm." *SIAM Journal on Computing*. Vol. 13, No. 1, 1984, pp. 170-181.
- Gangadharan, R. and C. Rajendran. "A simulated annealing heuristic for scheduling in a flowshop with bicriteria." *Computers and Industrial Engineering*. Vol. 27, 1994, pp. 473-476.

- Garey, M.R. and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979.
- Garey, M.R., D.S. Johnson, and R. Sethi. "The complexity of flowshop and jobshop scheduling". *Mathematics of Operations Research*. Vol. 1, No. 2, 1976, pp. 117-129.
- Gelinas, S. and F. Soumis. "A dynamic programming algorithm for single machine scheduling with ready times." *Annals of Operations Research*. Vol. 69, 1997, pp. 135-156.
- Geoffrion, A. M. and G.W. Graves. "Scheduling parallel production lines with changeover costs: practical application of a quadratic assignment / LP approach." *Operations Research*. Vol. 24, No. 4, 1976, pp. 595-610.
- Graham, R. L., E. L. Lawler, J. K. Lenstra and A. H. G.. Rinnooy Kan. "Optimization and approximation in deterministic sequencing and scheduling theory: a survey." *Ann. Discrete Math*. Vol. 5, 1979, pp. 287-326.
- Graham, R.L. "Bounds on multiprocessing timing anomalies." *SIAM Journal on Applied Mathematics*. Vol. 17, No. 2, 1969, pp. 416-429.
- Gu, Y. and C.A. Chung. "Genetic algorithm approach to aircraft gate reassignment problem." *Journal of Transportation Engineering*. Vol. 125, 1999, 384-389.
- Guignard, M. "Solving makespan minimization problems with Lagrangean decomposition." *Discrete Applied Mathematics*. Vol. 42, 1993, pp. 17-29.
- Gupta, J.N.D. "A flowshop scheduling problem with two operations per job." *International Journal of Production Research*. Vol. 35, No. 8, 1997, pp. 2309-2325.
- Gupta, J.N.D. "Flowshop schedules with sequence dependent setup times." *Journal of the Operational Research Society of Japan*. Vol. 29, No. 3, 1986, pp. 206-219.
- Gupta, J.N.D. "Heuristic algorithms for multistage flowshop scheduling problem." *AIIE Transactions*. Vol. 4, No. 1, 1972, pp. 11-18.
- Gupta, J.N.D. and E.A. Tunc. "Scheduling a two-stage hybrid flowshop with separable setup and removal times." *European Journal of Operational Research*. Vol. 77, 1994, pp. 415-428.
- Gupta, J.N.D. and W.P. Darrow. "The two-machine sequence dependent flowshop scheduling problem." *European Journal of Operational Research*. Vol. 24, 1986, pp. 439-446.

- Hall, L.A. "A polynomial approximation scheme for a constrained flow-shop scheduling problem." *Mathematics of Operations Research*. Vol. 19, No. 1, 1994, pp. 68-85.
- Hall, L.A. and D.B. Shmoys. "Jackson's rule for single-machine scheduling: making a good heuristic better." *Mathematics of Operations Research*. Vol. 17, No. 1, 1992, pp. 22-35.
- Ho, J.C. and Y.L. Chang. "A new heuristic for the n -job, M -machine flow-shop problem." *European Journal of Operational Research*. Vol. 52, 1991, pp. 194-202.
- Holland, J.H. *Adaptation in Natural and Artificial Systems*. Ann Arbor: The University of Michigan Press, 1975.
- Hou, E., N. Ansari, and H. Ren. "A genetic algorithm for multiprocessor scheduling." *IEEE Transactions on Parallel and Distributed Systems*. Vol. 5, No. 2, 1994, pp. 113-120.
- Hu, T.C. "Parallel sequencing and assembly line problems." *Operations Research*. Vol. 9, No. 6, 1961, pp. 841-848.
- Huber, Alfred and Detlev Buenz. "Using GRAI to specify expert systems for the control and the supervision of flexible flow lines". *Knowledge Based Production Management Systems. Proceedings of the IFIP WG 5.7 Working Conference*. 1989, pp. 295-308.
- Hunsucker, J.L. and S.A. Brah. "Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment." *European Journal of Operational Research*. Vol. 72, 1994, pp. 102-114.
- Ignall, E. and L. Schrage. "Application of the branch and bound technique to some flow-shop scheduling problems." *Operations Research*. Vol. 13, No. 3, 1965, pp. 400-412.
- Ishibuchi, H, S. Misaki and H. Tanaka. "Modified simulated annealing algorithms for the flow shop sequencing problem." *European Journal of Operational Research*. Vol. 81, 1995, pp. 388-398.
- Johnson, D.S. and C.H. Papadimitriou. "Computational complexity." *The Traveling Salesman Problem*. Ed. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy, and D.B. Shmoys. Chichester: John Wiley and Sons, 1985, pp. 37-85.
- Johnson, S.M. "Optimal two- and three-stage production schedules with setup times included." *Naval Research Logistics Quarterly*. Vol. 1, 1954, pp. 61-67.

- Kochhar, S. and R.J.T. Morris. "Heuristic methods for flexible flow line scheduling." *Journal of Manufacturing Systems*. Vol. 6, No. 4, 1987, pp. 299-314.
- Koulamas, C. "A new constructive heuristic for the flowshop scheduling problem." *European Journal of Operational Research*. Vol. 105, 1998, pp. 66-71.
- Kouvelis, P. and G. Vairaktarakis. "Flowshops with processing flexibility across production stages." *IIE Transactions*. Vol. 30, 1998, pp. 735-746.
- Kravchenko, S. A. and F. Werner. "Parallel machine scheduling problems with a single server." *Mathematical Computer Modelling*. Vol. 26, No. 12, 1997, pp. 1-11.
- Kurz, M. E. and R. G. Askin.. "Note on An adaptable problem-space-based search method for flexible flow line scheduling." *IIE Transactions*. Vol. 33, 2001, pp. 691-693.
- Lee, C.-Y. "Parallel machines scheduling with non-simultaneous machine available time." *Discrete Applied Mathematics*. Vol. 30, 1991, pp. 53-61.
- Lee, C.-Y. and G.L. Vairaktarakis. "Minimizing makespan in hybrid flowshops." *Operations Research Letters*. Vol. 16, 1994, pp. 149-158.
- Lee, C.-Y. and J.D. Massey. "Multiprocessor scheduling: combining LPT and MULTIFIT." *Discrete Applied Mathematics*. Vol. 20, 1988, pp. 233-242.
- Lee, C.Y., T.C.E. Cheng and B.M.T. Lin. "Minimizing the makespan in the 3-machine assembly-type flowshop scheduling problem." *Management Science*. Vol. 39, No. 5, 1993, pp. 617-625.
- Lee, I., R. Sikora, and M.J. Shaw. "A genetic algorithm-based approach to flexible flow-line scheduling with variable lot sizes." *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*. Vol. 27, No. 1, 1997, pp. 36-54.
- Lee, Y.H., and M. Pinedo. "Scheduling jobs on parallel machines with sequence-dependent setup times." *European Journal of Operational Research*. Vol. 100, 1997, pp. 464-474.
- Leon, V. J., and S. D. Wu. "On scheduling with ready-times, due-dates and vacations." *Naval Research Logistics*. Vol. 39, 1992, pp. 53-65.
- Leon, V.J. and B. Ramamoorthy. "An adaptable problem-space-based search method for flexible flow line scheduling." *IIE Transactions*. Vol. 29, 1997, pp. 115-125.
- Liu, J. and B.L. MacCarthy. "Effective heuristics for the single machine sequencing problem with ready times." *International Journal of Production Research*. Vol. 29, No. 8, 1991, pp. 1521-1533.

- Lomnicki, Z.A. "A 'branch-and-bound' algorithm for the exact solution of the three-machine scheduling problem". *Operational Research Quarterly*. Vol. 16, No. 1, 1965, pp. 89-100.
- Lourenco, H.R. "Sevast'yanov's algorithm for the flow-shop scheduling problem." *European Journal of Operational Research*. Vol. 91, 1996, pp. 176-189.
- McMahon, G. and M. Florian. "On scheduling with ready times and due dates to minimize maximum lateness." *Operations Research*. Vol. 23, No. 3, 1975, pp. 475-482.
- McNaughton, R. "Scheduling with deadlines and loss functions." *Management Science*. Vol. 6, No. 1, 1959, pp. 1-12.
- Min, L. and W. Cheng. "Identical parallel machine scheduling problem for minimizing the makespan using genetic algorithm combined by simulated annealing." *Chinese Journal of Electronics*. Vol. 7, No. 4, 1998, pp. 317-321.
- Monma, C.L. and C.N. Potts. "Analysis of heuristics for preemptive parallel machine scheduling with batch setup times." *Operations Research*. Vol. 41, No. 5, 1993, pp. 981-993.
- Monma, C.L. and C.N. Potts. "On the complexity of scheduling with batch setup times." *Operations Research*. Vol. 37, No. 5, 1989, pp. 798-814.
- Morris, J.S. and R.J. Tersine. "A simulation analysis of factors influencing the attractiveness of group technology cellular layouts." *Management Science*. Vol. 36, No. 12, 1990, pp. 1567-1578.
- Muntz, R.R. and E.G. Coffman. "Optimal preemptive scheduling on two-processor systems." *IEEE Transactions on Computers*. Vol. 18, No. 11, 1969, pp. 1014 - 1020.
- Muntz, R.R. and E.G. Coffman. "Preemptive scheduling of real-time tasks on multiprocessor systems." *Journal of the ACM*. Vol. 17, No. 2, 1970, pp. 324-338.
- Nagar, A., A.A. Heragu and J. Haddock. "A branch-and-bound approach for a two machine flowshop scheduling problem." *Journal of the Operational Research Society*. Vol. 46, 1995, pp. 721-734.
- Nagar, A., A.A. Heragu and J. Haddock. "A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem." *Annals of Operations Research*. Vol. 63, 1996, pp. 397-414.

- Nawaz, M., E.E. Enscore, and I. Ham. "A heuristic algorithm for the m -machine, n -job flow-shop sequencing problem." *OMEGA*. Vol. 11, No. 1, 1983, pp. 91-95.
- Norman, B.A. and J.C. Bean. "A genetic algorithm methodology for complex scheduling problems." *Naval Research Logistics*. Vol. 46, 1999, pp. 199-211.
- Nowicki, E. and C. Smutnicki. "A fast tabu search algorithm for the permutation flow-shop problem." *European Journal of Operational Research*. Vol. 91, 1996, pp. 160-175.
- Nowicki, E. and C. Smutnicki. "An approximation algorithm for a single-machine scheduling problem with release times and delivery times." *Discrete Applied Mathematics*. Vol. 48, 1994, pp. 69-79.
- Nowicki, E. and C. Smutnicki. "The flow shop with parallel machines: a tabu search approach." *European Journal of Operational Research*. Vol. 106, 1998, pp. 226-253.
- Nowicki, E. and S. Zdrzalka. "Single machine scheduling with major and minor setup times: a tabu search approach." *Journal of the Operational Research Society*. Vol. 47, 1996, pp. 1054-1064.
- Oguz, C. and M.F. Ercan. "Scheduling multiprocessor tasks in a two-stage flow-shop environment." *Computers and Industrial Engineering*. Vol. 33, 1997, pp. 269-272.
- Oguz, C., B.M.T. Lin and T.C.E. Cheng. "Two-stage flowshop scheduling with a common second-stage machine." *Computers and Operations Research*. Vol. 24, No. 12, 1997, pp. 1169-1174.
- Osman, I.H and C.N. Potts. "Simulated annealing for permutation flow-shop scheduling." *OMEGA*. Vol. 17, No. 6, 1989, pp. 551-557.
- Ovacik, I.M. and R. Uzsoy. "Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times." *International Journal of Production Research*. Vol. 33, No. 11, 1995, pp. 3173-3192.
- Ovacik, I.M. and R. Uzsoy. "Worst-case error bounds for parallel machine scheduling problems with bounded sequence-dependent setup times." *Operations Research Letters*. Vol. 14, 1993, pp. 251-256.
- Palmer, D.S. "Sequencing jobs through a multi-stage process in the minimum total time – a quick method of obtaining a near optimum." *Operational Research Quarterly*. Vol. 16, No. 1, 1965, pp. 101-107.

- Papadimitriou, C.H. and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Mineola, New York: Dover Publications, 1998.
- Parker, R.G., R.H. Deane and R.A. Holmes. "On the use of a vehicle routing algorithm for the parallel processor problem with sequence dependent changeover costs." *AIIE Transactions*. Vol. 9, No. 2, 1976, pp. 155-160.
- Peha, J. M. "Heterogeneous-criteria scheduling: minimizing weighted number of tardy jobs and weighted completion time." *Computers and Operations Research*. Vol. 22, No. 10, 1995, pp. 1089-1100.
- Pinedo, Michael. *Scheduling: theory, algorithms and systems*. Prentice-Hall: New Jersey, 1995.
- Piramuthu, S., N. Raman, and M.J. Shaw. "Learning-based scheduling in a flexible manufacturing flow line." *IEEE Transactions on Engineering Management*. Vol. 41, No. 2, 1994, pp. 172-182.
- Potts, C.N. "Analysis of a heuristic for one machine sequencing with release dates and delivery times." *Operations Research*. Vol. 28, No. 6, 1980, pp. 1436-1441.
- Potts, C.N. "Analysis of heuristics for two-machine flow-shop sequencing subject to release dates." *Mathematics of Operations Research*. Vol. 10, No. 4, 1985, pp. 576-584.
- Proust, C, J.N.D. Gupta and V. Deschamps. "Flowshop scheduling with set-up, processing and removal times separated." *International Journal of Production Research*. Vol. 29, No. 3, 1991, pp. 479-493.
- Punnen, A.P., and Y.P. Aneja. "Minmax combinatorial optimization." *European Journal of Operational Research*. Vol. 81, 1995, pp. 634-643.
- Raban, Simon and Roger N. Nagel. "Constraint-based control of flexible flow lines". *International Journal of Production Research*. Vol. 29, No. 10, 1991, pp. 1941-1951.
- Rajendran, C. and D. Chaudhuri. "Scheduling in n -job, m -stage flowshop with parallel processors to minimize makespan." *International Journal of Production Economics*. Vol. 27, 1992, pp. 137-143.
- Rajendran, C. and H. Ziegler. "Heuristics for scheduling in a flowshop with setup, processing nad removal times separated." *Production Planning and Control*. Vol. 8, No. 6, 1997, pp. 568-576.

- Rajgopal, J. and B. Bidanda. "On scheduling parallel machines with two setup classes." *International Journal of Production Research*. Vol. 29, No. 12, 1991, pp. 2443-2458.
- Reeves, C. "Heuristics for scheduling a single machine subject to unequal job release times." *European Journal of Operational Research*. Vol. 80, 1995, pp. 397-403.
- Reinelt, G. *The Traveling Salesman: Combinatorial Solutions for TSP Applications*. Berlin: Springer-Verlag, 1994.
- Riane, F., A. Artiba, and S.E. Elmaghraby. "A hybrid three-stage flowshop problem: efficient heuristics to minimize makespan." *European Journal of Operational Research*. Vol. 109, 1998, pp. 321-329.
- Ríos-Mercado, R.Z. and J.F. Bard. "A branch-and-bound algorithm for permutation flow shops with sequence-dependent setup times." *IIE Transactions*. Vol. 31, 1999, pp. 721-731.
- Ríos-Mercado, R.Z. and J.F. Bard. "An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times." *Journal of Heuristics*. Vol. 5, 1999, pp. 53-70.
- Rios-Mercado, R.Z. and J.F. Bard. "Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups." *Computers and Operations Research*. Vol. 25, No. 5, 1998, pp. 351-366.
- Rosenkrantz, D.J., L. Yu and S.S. Ravi. "Efficient construction of minimum makespan schedules for tasks with a fixed number of distinct execution times." *Algorithmica*. Vol. 30, 2001, pp. 83-100.
- Sahni S., and T. Gonzalez. "P-complete approximation problems." *Journal of the ACM*. Vol. 23, 1976, pp. 555-565.
- Sahni, S.K. "Algorithms for scheduling independent tasks." *Journal of the ACM*. Vol. 23, No. 1, 1976, pp. 116-127.
- Salvador, M.S. "A solution to a special case of flow shop scheduling problems." In *Symposium on the Theory of Scheduling and Its Applications*. Ed. S.E. Elmaghraby. Springer-Verlag: Berlin, 1973, pp. 83-91.
- Santos, D.L., J.L. Hunsucker, and D.E. Deal. "An evaluation of sequencing heuristics in flow shops with multiple processors." *Computers and Industrial Engineering*. Vol. 30, No. 4, 1996, pp. 681-692.

- Santos, D.L., J.L. Hunsucker, and D.E. Deal. "FLOWMULT: permutation sequences for flow shops with multiple processors." *Journal of Information and Optimization Sciences*. Vol. 16, No. 2, 1995, pp. 351-366.
- Sawik, T.J. "A scheduling algorithm for flexible flow lines with limited intermediate buffers." *Proceedings of the 8th International Conference on CAD/CAM, Robotics and Factories of the Future*. Vol. 2, 1992, pp. 1711-1722.
- Sawik, T.J. "New algorithms for scheduling flexible flow lines." *Proceedings of the 1994 Japan-USA Symposium on Flexible Automation*. Vol. 3, 1994, pp. 1091-1094.
- Sawik, T.J. "Scheduling flexible flow lines with no in-process buffers." *International Journal of Production Research*. Vol. 33, No. 5, 1995, pp. 1357-1367.
- Schaller, J. "A comparison of heuristics for family and job scheduling in a flow-line manufacturing cell." *International Journal of Production Research*. Vol. 38, 2000, pp. 287-308.
- Schutten, J.M.J. and R.A.M. Leussink. "Parallel machine scheduling with release dates, due dates and family setup times." *International Journal of Production Economics*. Vol. 46-47, 1996, pp. 119-125.
- Schutten, J.M.J. "List scheduling revisited." *Operations Research Letters*. Vol. 18, 1996, pp. 167-170.
- Schutten, J.M.J., S.L. van de Velde, and W.H.M. Zijm. "Single-machine scheduling with release dates, due dates and family setup times." *Management Science*. Vol. 42, No. 8, 1996, pp. 1165-1174.
- Sivrikaya-Serifoglu, F. and G. Ulusoy. "Parallel machine scheduling with earliness and tardiness penalties." *Computers and Operations Research*. Vol. 26, 1999, 773-787.
- Snyder, L.V. and M.S. Daskin.. "A random-key genetic algorithm for the generalized traveling salesman problem." Working Paper, Department of Industrial Engineering and Management Sciences, Northwestern University, 2001.
- So, K.C. "Some heuristics for scheduling jobs on parallel machines with setups." *Management Science*. Vol. 36, No. 4, 1990, pp. 467-475.
- Sotskov, Y.N., T. Tautenhahn, and F. Werner. "Heuristics for permutation flow shop scheduling with batch setup times." *OR Spektrum*. Vol. 18, 1996, pp. 67-80.

- Srikar, B.N. and S. Ghosh. "A MILP model for the n -job, M -stage flowshop with sequence dependent set-up times." *International Journal of Production Research*. Vol. 24, No. 6, 1986, pp. 1459-1474.
- Sriskandarajah, C. and S.P. Sethi. "Scheduling algorithms for flexible flowshops: worst and average case performance." *European Journal of Operational Research*. Vol. 43, 1989, pp. 143-160.
- Stafford, E.F. and F.T. Tseng. "On the Srikar-Ghosh model for the $N \times M$ SDST flowshop problem." *International Journal of Production Research*. Vol. 28, No. 10, 1990, pp. 1817-1830.
- Stinson, J.P. and A.W. Smith. "A heuristic programming procedure for sequencing the static flowshop." *International Journal of Production Research*. Vol. 20, No. 6, 1982, pp. 753-764.
- Sule, D.R. "Sequencing n jobs on two machines with setup, processing and removal times separated." *Naval Research Logistics Quarterly*. Vol. 29, No.3, 1982, pp. 517-519.
- Sule, D.R. and K.Y.Huang. "Sequency on two and three machines with setup, processing and removal times separated." *International Journal of Production Research*. Vol. 21, No. 5, 1983, pp. 723-732.
- Sundararaghavan, P.S., A.S. Kunnathur and I. Viswanathan. "Minimizing makespan in parallel flowshops." *Journal of the Operational Research Society*. Vol. 48, 1997, pp. 834-842.
- Suresh, V. "A note on scheduling of two-stage flow shop with multiple processors." *International Journal of Production Economics*. Vol. 49, 1997, pp. 77-82.
- Tadei, R., J.N.D. Gupta, F.D. Croce, and M. Cortesi. "Minimising makespan in the two-machine flow-shop with release times". *Journal of the Operational Research Society*. Vol. 49, 1998, pp. 77-85.
- Taillard, E. "Some efficient heuristic methods for the flow shop sequencing problem." *European Journal of Operational Research*. Vol. 47, 1990, pp. 65-74.
- Tang, C.S. "Scheduling batches on parallel machines with major and minor set-ups." *European Journal of Operational Research*. Vol. 46, 1990, pp. 28-37.
- Tinhofer, G. and E. Farnbacher. "A new lower bound for the makespan of a single machine scheduling problem." *System Modelling and Optimization: Proceedings of the 15th IFIP Conference*. Springer-Verlag: Berlin, 1992, pp. 209-218. [verify this citation for style]

- Tzafestas, S. and A. Triantafyllakis. "Deterministic scheduling in computing and manufacturing systems: a survey of models and algorithms." *Mathematics and Computers in Simulation*. Vol. 35, 1993, pp. 397-434.
- Vairaktarakis, G. and M. Elhafi. "The use of flowlines to simplify routing complexity in two-stage flowshops." *IIE Transactions*. Vol. 32, 2000, pp. 687-699.
- Wagner, H.M. "An integer programming model for machine scheduling." *Naval Research Logistics Quarterly*. Vol. 6, 1959, pp. 131-140.
- Werner, F. "On the heuristic solution of the permutation flow shop problem by path algorithms." *Computers and Operations Research*. Vol. 20, No. 7, 1993, pp. 707-722.
- Widmer, M. and A. Hertz. "A new heuristic method for the flow shop sequencing problem." *European Journal of Operational Research*. Vol. 41, 1989, pp. 186-193.
- Wilson, Robin J. *Introduction to Graph Theory*. Essex, England: Longman Group Ltd., 1996.
- Wittrock, Robert. "An adaptable scheduling algorithm for flexible flow lines." *Operations Research*. Vol. 36, No. 3, 1988, pp. 445-453.
- Wittrock, Robert. "Scheduling algorithms for flexible flow lines." *IBM Journal of Research and Development*. Vol. 29, No. 24, 1985, pp. 401-412.
- Yoshida, T and K. Hitomi. "Optimal two-stage production scheduling with setup times separated." *AIEE Transactions*. Vol. 11, No. 3, 1979, pp. 261-263.
- Zdrzalka, S. "A sequencing problem with family setup times." *Discrete Applied Mathematics*. Vol. 66, 1996, pp. 161-183.
- Zegordi, S.H., K. Itoh and T. Enkawa. "Minimizing makespan for flow shop scheduling by combining simulated annealing with sequencing knowledge." *European Journal of Operational Research*. Vol. 85, 1995, pp. 515-531.
- Zhang, G. "A simple semi on-line algorithm for $P2||C_{max}$ with a buffer." *Information Processing Letters*. Vol. 61, 1997, pp. 145-148