

To the Graduate Council:

I am submitting herewith a dissertation written by Robin Ann Hutcheson Lovgren entitled "Mixed-Model Assembly Line Sequencing Problem". I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.



Kenneth Gilbert, Major Professor

We have read this dissertation
and recommend its acceptance:







Accepted for the Council:



Associate Vice Chancellor and
Dean of The Graduate School

**MIXED-MODEL ASSEMBLY LINE
SEQUENCING PROBLEM**

A Dissertation
Presented for the
Doctor of Philosophy Degree
The University of Tennessee, Knoxville

Robin Ann Hutcheson Lovgren
August, 1996

Copyright © Robin H. Lovgren, 1996

All rights reserved

DEDICATION

This dissertation is dedicated to my husband

Martin C. Lovgren

for his loving respect, encouragement, and support

and to my parents

Dr. Aaron A. Hutcheson

and

Lorene H. Hutcheson

who gave me the will to excel and the encouragement to succeed.

ACKNOWLEDGMENTS

I would like to thank my major professor, Dr. Kenneth Gilbert, for his wisdom and guidance, and for his valuable comments and suggestions during the research and writing of this dissertation. Also for always asking the right questions at the right times and for communicating so well over the miles. I would also like to thank the other committee members, Dr. Melissa Bowers, Dr. Mary Leitnaker, and Dr. Charles Noon for their valuable comments, insight, and assistance. I would like to thank Dr. Michael Racer for his enthusiastic interest in exploring alternative theories and ideas.

I would like to express my deep appreciation to my parents and family who always believe that I can do anything I set my mind on, and who give me encouragement and support at each stage of my endeavors. Finally, I would like to thank my husband Marty for all of his love, faith, patience, and for showing that 'two are better than one'.

ABSTRACT

The focus of this research is the development of a solution approach that will aid decision makers in understanding the tradeoffs associated with sequencing to meet due dates versus sequencing to level component usage for the mixed-model sequencing problem with specific customer due dates. In this problem, the product due dates are determined by the customer and each product on the assembly line is associated with a specific customer due date. This method of manufacturing leads to 'build-to-order' production rather than 'build-to-stock' production.

The problem of sequencing in the 'build-to-order' environment with specific customer due dates is a new research area that is the main focus of this work. Mixed-model sequencing techniques have been developed for use in the Just-In-Time, repetitive manufacturing, build-to-stock environment. A majority of these scheduling methods are designed to optimize component usage by keeping a constant rate of usage of parts in the system. Other techniques incorporate the goal of leveling the work load at each station on the assembly line. Along with studying the due date, part usage tradeoff, this research also introduces a new sequencing goal: reducing the time spread of producing a customer's order. This new goal is currently an important issue in industry, but is not studied in the mixed-model sequencing literature.

This research adds insight to the problem of determining which items and how many items should be sequenced late to better meet the goal of leveling the component usage for a specified time period. A set of new sequencing methods are designed and tested on two populations with varying results. Also, a new sequencing goal for leveling component usage over the short term as well as for the long run, is discussed and incorporated into the algorithms.

TABLE OF CONTENTS

1. INTRODUCTION AND PRELIMINARIES	1
1. PROBLEM DEFINITION AND RESEARCH CONTRIBUTIONS	1
2. TERMINOLOGY	4
3. PRODUCTION SCHEDULING GOALS	6
1. <i>Meet The Customer Due Date</i>	6
2. <i>Level The Pull Of Component Parts Through The System</i>	7
3. <i>Balance The Work Load On The Assembly Line</i>	7
4. <i>Minimize The Production Time Spread Of Customer Orders</i>	8
4. RESEARCH OBJECTIVES.....	9
5. ORGANIZATION OF DISSERTATION.....	9
2. LITERATURE SURVEY.....	11
1. INTRODUCTION.....	11
2. LITERATURE SURVEY.....	13
3. MIXED-MODEL SEQUENCING PROBLEM	19
1. THE SEQUENCING MODEL.....	19
2. OBJECTIVE FUNCTION.....	20
1. <i>1st Objective</i>	20
2. <i>2nd Objective</i>	21
4. A SOLUTION APPROACH FOR THE MIXED-MODEL SEQUENCING PROBLEM.....	26
1. GENERAL APPROACH.....	27
2. THE BRANCH-AND-BOUND APPROACH FOR OPTIMAL SEQUENCING.....	28
3. THREE BASIC TYPES OF ALGORITHMS FOR SEQUENCING.....	31
1. <i>Monden's Goal Chasing</i>	32
2. <i>The 2-Optimal Algorithm</i>	35
3. <i>The Border Swap Algorithm</i>	36
4. THE DECOMPOSITION OF THE PART LEVELING GOAL INTO TWO SEPARATE GOALS	38
1. <i>The Multi-period Goal Versus the Single Period Goal</i>	39
2. <i>Application of These Results to the Problem</i>	42
5. A SUMMARY OF THE THREE OBJECTIVES AND THE HEURISTICS APPLICABLE TO EACH.....	44
6. ALGORITHMS TO MINIMIZE MSD FROM LINEAR PART USAGE WITH ZERO LATE ITEMS.....	46
1. <i>GC1-DD Heuristic Algorithm For Sequencing</i>	46
2. <i>GC1-Look-Ahead-DD Heuristic Algorithm For Sequencing</i>	47
3. <i>GC1-Look-Ahead-DD / 2-Optimal-DD Heuristic Algorithm For Sequencing</i>	47
4. <i>Sort-DD / 2-Optimal-DD Heuristic Algorighm For Sequencing</i>	48
7. ALGORITHMS TO MINIMIZE MSD FROM LINEAR PART USAGE WITH NO RESTRICTION ON LATE ITEMS	48
1. <i>GC1 Heuristic Algorithm</i>	48
2. <i>GC1-Look-Ahead Heuristic Algorithm</i>	49
3. <i>Border Swap Heuristic Algorithm</i>	49
8. ALGORITHMS TO PRODUCE SEQUENCES MINIMIZING MSD FROM LINEAR PART USAGE WITH SPECIFIC NUMBERS OF LATE ITEMS	49
1. <i>GC1-Look-Ahead-DD / 2-Opt Heuristic Algorithm For Sequencing</i>	49
2. <i>Sort-DD / 2-Opt Heuristic Algorithm For Sequencing</i>	50
3. <i>Border Swap Heuristic Algorithm For Sequencing</i>	50
5. IMPLEMENTATION AND RESULTS.....	51
1. EVALUATING THE SEQUENCING METHODS.....	51

2. THE SMALL TEST PROBLEM.....	53
3. RESULTS OF THE SMALL TEST PROBLEM	57
1. <i>Goal 1: Level Part Usage with Zero Items Late and Problem Size = 20 Items</i>	58
2. <i>Goal 2: Level Part Usage with No Due Date Restrictions and Problem Size = 20 Items</i>	61
3. <i>Goal 3: Level Part Usage with X Items Late and Problem Size = 20 Items</i>	63
6. ANALYSIS, CONCLUSIONS, AND FUTURE RESEARCH	120
1. ANALYSIS OF MONDEN'S GOAL CHASING 1 METHOD WITH ZERO LATE ITEMS.....	120
2. ANALYSIS OF MONDEN'S GOAL CHASING 1 METHOD WITH ZERO LATE ITEMS ACCORDING TO THE TOTAL PERIOD GOAL	125
3. EVALUATION OF BORDER SWAP ALGORITHM	130
4. EVALUATION OF THE TWO OPTIMAL HEURISTIC	133
5. CONCLUSION.....	137
6. FUTURE RESEARCH.....	138
1. <i>Direct Extensions Of This Dissertation</i>	138
2. <i>Broad Extensions And General Questions</i>	139
BIBLIOGRAPHY.....	142
B. EXAMPLE OF THE BORDER SWAP HEURISTIC.....	147
APPENDIX	151
A. INPUT FILE FORMAT AND A SAMPLE INPUT FILE.....	151
VITA.....	153

LIST OF TABLES

Table 1: Sample Problem.....	22
Table 2: Total Parts For Sample Problem.....	22
Table 3: Goal Production Per Part For Sample Problem.....	23
Table 4: Part Usage Rate And Deviation For Sample Problem.....	23
Table 5: Example Of Inherent Rounding Error For Sample Problem	30
Table 6: Example of a Problem Comparing the Two Conflicting Goals.....	42
Table 7: Summary Of Objectives And Heuristics	45
Table 8: Model Composition By Component	54
Table 9: Initial Demand Patterns A - J.....	55
Table 10: Final Demand Patterns A - J.....	56
Table 11: Due Dates in Order by Model for Each Demand Pattern A - J.	56
Table 12: Algorithms to be Tested and Analyzed for Each Objective	57
Table 13: Total Mean Square Deviation from Overall Uniform Component Usage with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	59
Table 14: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Optimal Component Usage for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	59
Table 15: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	60
Table 16: Total Mean Square Deviation from Uniform Component Usage Over All Periods for Sequences Adjusted and Leveled Within each Period with Zero Late Items.	60
Table 17: Mean Square Deviation from Uniform Component Usage, and Number of Items Late Respectively with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions	62
Table 18: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Optimal Component Usage Deviation for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions	62

Table 19: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage with No Due Date Restrictions	63
Table 20: Total Mean Square Deviation from Uniform Component Usage Over All Periods and Number of Items Late Respectively for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions	20
Table 21: Average Adjusted Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions	21
Table 22: The Average Percent from Optimal for each Sequencing Method with the Sequence Based on Over All Period Goals for Demand Patterns A-J with 20 Items	66
Table 23: The Average Percent from Best for each Sequencing Method with the Sequence Adjusted Based on Single Period Goals for Demand Patterns A-J with 20 Items	67
Table 24: MSD Demand Pattern A	68
Table 25: Adjusted MSD Demand Pattern A	69
Table 26: MSD Demand Pattern B	70
Table 27: Adjusted MSD Demand Pattern B	71
Table 28: MSD Demand Pattern C	72
Table 29: Adjusted MSD Demand Pattern C	73
Table 30: MSD Demand Pattern D	74
Table 31: Adjusted MSD Demand Pattern D	75
Table 32: MSD Demand Pattern E	76
Table 33: Adjusted MSD Demand Pattern E	77
Table 34: MSD Demand Pattern F	78
Table 35: Adjusted MSD Demand Pattern F	79
Table 36: MSD Demand Pattern G	80
Table 37: Adjusted MSD Demand Pattern G	81
Table 38: MSD Demand Pattern H	82
Table 39: Adjusted MSD Demand Pattern H	83

Table 40: MSD Demand Pattern I.....	84
Table 41: Adjusted MSD Demand Pattern I	85
Table 42: MSD Demand Pattern J	86
Table 43: Adjusted MSD Demand Pattern J	87
Table 44: Initial Demand Patterns AA-JJ	88
Table 45: Final Demand Patterns AA-JJ.....	89
Table 46: Due Dates in Order by Model for Each Demand Pattern AA - JJ.	89
Table 47: Total Mean Square Deviation from Overall Uniform Component Usage with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	92
Table 48: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	92
Table 49: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.....	93
Table 50: Total Mean Square Deviation from Uniform Component Usage Over All Periods for Sequences Adjusted and Leveled Within each Period with Zero Late Items	93
Table 51: Mean Square Deviation from Uniform Component Usage, and Number of Items Late Respectively with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions	94
Table 52: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions	95
Table 53: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage with No Due Date Restrictions	95
Table 54: Total Mean Square Deviation from Uniform Component Usage Over All Periods and Number of Items Late Respectively for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions	95
Table 55: Average Adjusted Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions	96
Table 56: The Average Percent from Best for each Sequencing Method with the Sequence Based on Over All Period Goals for Demand Patterns AA-JJ with 60 Items.....	97

Table 57: The Average Percent from Best for each Sequencing Method with the Sequence Adjusted Based on Single Period Goals for Demand Patterns AA-JJ with 60 Items	99
Table 58: MSD Demand Pattern AA	100
Table 59: Adjusted MSD Demand Pattern AA.....	101
Table 60: MSD Demand Pattern BB.....	102
Table 61: Adjusted MSD Demand Pattern BB	103
Table 62: MSD Demand Pattern CC.....	104
Table 63: Adjusted MSD Demand Pattern CC	105
Table 64: MSD Demand Pattern DD	106
Table 65: Adjusted MSD Demand Pattern DD.....	107
Table 66: MSD Demand Pattern EE	108
Table 67: Adjusted MSD Demand Pattern EE.....	109
Table 68: MSD Demand Pattern FF.....	110
Table 69: Adjusted MSD Demand Pattern FF	111
Table 70: MSD Demand Pattern GG	112
Table 71: Adjusted MSD Demand Pattern GG.....	113
Table 72: MSD Demand Pattern HH	114
Table 73: Adjusted MSD Demand Pattern HH.....	115
Table 74: MSD Demand Pattern II	116
Table 75: Adjusted MSD Demand Pattern II.....	117
Table 76: MSD Demand Pattern JJ.....	118
Table 77: Adjusted MSD Demand Pattern JJ	119
Table 78: Cumulative Part Totals for Pattern AA, Time Period 1.....	122
Table 79: Partial Sequences with Associated Square Errors	129
Table 80: Series of Sequences for Demand Pattern D Using the GC1 / 2opt Method with One Item Allowed Late	134

Table 81: Series of Sequences for Demand Pattern D Using the GC1 / 2opt Method with Two
Items Allowed Late 134

Table 82: Series of Sequences for Demand Pattern D Using the Sort / 2opt Method with One
Item Allowed Late 136

Table 83: Series of Sequences for Demand Pattern D Using the Sort / 2opt Method with Two
Items Allowed Late 136

LIST OF FIGURES

Figure 1: A Branch and Bound Solution Tree with Three Items Sorted in Order A, B, C that Includes All Possible Solutions, Feasible and In-feasible.....	29
Figure 2: Sequence Based on Weekly Goal.....	41
Figure 3: Deviation from Weekly Goal	41
Figure 4: Sequence Based on Daily Goal.	41
Figure 5: New Deviation from Weekly Goal.....	41
Figure 6: Graph of the Mean Square Deviation for the Example Demand Pattern	43
Figure 7: Graph of the Mean Square Deviation for Demand Pattern C.....	44
Figure 8: Graph of the Average Percent from the Optimal Mean Square Deviation for Demand Patterns A through J.	66
Figure 9: Graph of the Adjusted Average Percent from the Best Mean Square Deviation for Demand Patterns A through J.....	67
Figure 10: Graph of the Overall Mean Square Deviation for Demand Pattern A.	68
Figure 11: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern A.....	69
Figure 12: Graph of the Overall Mean Square Deviation for Demand Pattern B.....	70
Figure 13: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern B.	71
Figure 14: Graph of the Overall Mean Square Deviation for Demand Pattern C.....	72
Figure 15: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern C	73
Figure 16: Graph of the Overall Mean Square Deviation for Demand Pattern D.	74
Figure 17: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern D.....	75
Figure 18: Graph of the Overall Mean Square Deviation for Demand Pattern E.....	76
Figure 19: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern E.	77
Figure 20: Graph of the Overall Mean Square Deviation for Demand Pattern F	78
Figure 21: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern F.....	79

Figure 22: Graph of the Overall Mean Square Deviation for Demand Pattern G	80
Figure 23: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern G.....	81
Figure 24: Graph of the Overall Mean Square Deviation for Demand Pattern H	82
Figure 25: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern H.....	83
Figure 26: Graph of the Overall Mean Square Deviation for Demand Pattern I.....	84
Figure 27: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern I	85
Figure 28: Graph of the Overall Mean Square Deviation for Demand Pattern J.....	86
Figure 29: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern J.	87
Figure 30: Graph of the Average Percent from the Best Mean Square Deviation for Demand Patterns AA through JJ.....	97
Figure 31: Graph of the Adjusted Average Percent from the Best Mean Square Deviation for Demand Patterns AA through JJ	99
Figure 32: Graph of the Overall Mean Square Deviation for Demand Pattern AA.....	100
Figure 33: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern AA.	101
Figure 34: Graph of the Overall Mean Square Deviation for Demand Pattern BB	102
Figure 35: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern BB	103
Figure 36: Graph of the Overall Mean Square Deviation for Demand Pattern CC.	104
Figure 37: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern CC	105
Figure 38: Graph of the Overall Mean Square Deviation for Demand Pattern DD.....	106
Figure 39: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern DD.	107
Figure 40: Graph of the Overall Mean Square Deviation for Demand Pattern EE	108
Figure 41: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern EE.....	109
Figure 42: Graph of the Overall Mean Square Deviation for Demand Pattern FF	110
Figure 43: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern FF.	111
Figure 44: Graph of the Overall Mean Square Deviation for Demand Pattern GG.....	112
Figure 45: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern GG	113

Figure 46: Graph of the Overall Mean Square Deviation for Demand Pattern HH.....	114
Figure 47: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern HH.....	115
Figure 48: Graph of the Overall Mean Square Deviation for Demand Pattern II.....	116
Figure 49: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern II.....	117
Figure 50: Graph of the Overall Mean Square Deviation for Demand Pattern JJ.....	118
Figure 51: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern JJ.....	119
Figure 52: Square Deviation Pattern AA.....	121
Figure 53: Square Deviation Pattern BB.....	121
Figure 54: Actual Part 1 Usage.....	123
Figure 55: Cumulative Part 1 Usage.....	123
Figure 56: Actual Part 2 Usage.....	123
Figure 57: Cumulative Part 2 Usage.....	123
Figure 58: Deviation Part 1 Adjusted.....	124
Figure 59: Deviation Part 2 Adjusted.....	124
Figure 60: Deviation Part 3 Adjusted.....	124
Figure 61: Deviation Part 4 Adjusted.....	124
Figure 62: Square Deviation Pattern AA.....	126
Figure 63: Square Deviation Pattern BB.....	126
Figure 64: Square Deviation Pattern CC.....	126
Figure 65: Deviation Part 1.....	127
Figure 66: Deviation Part 2.....	127
Figure 67: Deviation Part 3.....	127
Figure 68: Deviation Part 4.....	127
Figure 69: Deviation Part 5.....	128
Figure 70: Deviation Part 6.....	128

Figure 71: Deviation Part 7	128
Figure 72: Deviation Part 8	128
Figure 73: Deviation Part 9	128
Figure 74: Deviation Part 10	128
Figure 75: Square Deviation GC1	130
Figure 76: Sq. Dev., 1 Per. Goal, Evaluated on Total Goal.	130
Figure 77: GC1, Pattern AA.....	131
Figure 78: Border Swap, 1 Late	131
Figure 79: Border Swap, 2 Late.	131
Figure 80: Border Swap, 3 Late	131
Figure 81: Border Swap, 4 Late	132
Figure 82: Border Swap, 3 Late Repeat	132
Figure 83: Border Swap, 5 Late.	132
Figure 84: Border Swap, 5 Late Repeat	132
Figure 85: Border Swap, 6 Late	132
Figure 86: GC1-No DD Restrictions.....	132

CHAPTER 1

Introduction and Preliminaries

We begin this chapter with an introduction to the Mixed-Model Sequencing Problem (MMSP) and the contributions of this research to the body of knowledge as it pertains to the MMSP. In section 2 we provide terminology to clarify the problem definition. Section 3 describes the main goals of sequencing in the build-to-order environment. In section 4 we discuss the objectives of our research and section 5 concludes the chapter with the outline of this dissertation.

1. Problem Definition and Research Contributions

A key aspect of lean production is the ability of a company to produce an ever-growing variety of product. This concept is to combine the economies of the old mass production techniques and the customization of craft production ideas. With this new type of production comes the responsibility of companies to determine an efficient method for scheduling a variety, or mix, of products. *The problem is to develop efficient and appropriate techniques for dynamic, mixed-model sequencing for a discrete lot, build-to-order, high volume, Just-In-Time, final assembly line.* This research focuses on providing information about sequencing algorithms that can be used to evaluate the due date - level part usage tradeoff. These are two conflicting goals which will be explained in section 1.3. The research also redefines the part usage goal to better meet the objectives of sequencing in the build-to-order environment.

The following is an outline of the contributions of this dissertation to the body of knowledge pertaining to the mixed-model sequencing problem. The contributions are presented

with respect to two areas: first from a theoretical standpoint, and second from a practical perspective.

1. From a math programming standpoint, this dissertation provides a new definition of the goal used in a simple greedy algorithm. In this research the goal of interest was to level components over a specified time period. The first attempt at leveling components was based on an overall goal minimizing deviation of actual part usage from level part usage across the entire production sequence. Observations which will be explained later, indicate that optimizing this goal by minimizing the associated objective does not necessarily provide the optimal level part usage. A new goal is defined which can be used to provide more desirable results.
2. This research also explores new low-order polynomial heuristics for solving the Mixed-Model sequencing problem, one of which extends an existing intelligent greedy method and incorporates a local search heuristic. Some of these heuristics are found to perform within approximately 10% of the optimal solution for the problems tested.
3. An evaluative methodology for the multi-objective environment is established.
4. A branch-and-bound procedure is developed that exploits embedded assignment problem constraints. The optimization program is used to compare heuristic solutions to optimal solutions for small sequences.
5. From a practical standpoint, this dissertation provides an improvement to the well known and widely used Goal-Chasing 1 algorithm by Monden.
6. This research also provides information concerning tradeoffs between two objectives which

have not previously been studied in the multi-objective sequencing environment.

The following is a list of tasks used to accomplish the theoretical and practical contributions explained above.

1. Introduced a set of new methods for a two-objective mixed-model sequencing problem and tested several sequencing heuristics on two populations with respect to two of the sequencing goals: meet the customer due date, and level the pull of component parts through the system. The following is a list of the methods tested:
 - Monden's Goal Chasing 1 (GC1) - Tested the existing heuristic, for comparison with new algorithms.
 - An improvement to Monden's Goal Chasing 1 - Adapted and tested the heuristic which is designed to improve the quality of the solution with a longer running time.
 - A 2-optimal heuristic - Introduced and tested the heuristic to try another approach to improving the quality of the solution with a relatively longer running time than Monden's GC1.
 - Border Swap heuristic - Designed and tested the heuristic as a relatively fast method for determining a sequence.
2. Redefined the part usage goal as two individual goals: level component usage over all periods; and level component usage over each period.
 - Recognized the problem with current procedures, and designed a heuristic two-phased method which conditionally improves daily sequencing to better meet the sequencing goals.
3. Investigated the implications of algorithm selection on two populations, and evaluated the results of tradeoffs between two of the sequencing goals: meet the customer due date, and

level the pull of component parts through the system.

4. Defined an additional sequencing goal: minimize the production time spread of customer orders.

2. Terminology

The key terms that define the elements of the problem are: dynamic, mixed-model, discrete lot, build-to-order, Just-In-Time, and high volume assembly line. Explanations of the various aspects that define the problem statement follow.

Mixed-model sequencing is used when a variety of products, similar in nature, are produced alternately on the same production line. Differences in the products may vary in small ways such as the color of paint on a component, or they may vary greatly, such as a base model versus a luxury package, or even front wheel drive and rear wheel drive automobiles. Intentionally mixing the products facilitates line balancing and smoothed demand on internal and supplier resources. This method of manufacturing replaces the well-known large batch production method.

Products are ordered by the customers in *discrete lot sizes* at varying times throughout the year. This characteristic eliminates products of a continual nature such as paper or batches of chemicals. The products, along with their associated lead times, are added to a master calendar to facilitate materials acquisition of long lead time components. A preliminary schedule is created at predetermined time intervals from the master calendar. Then, revised schedules with updated information are used to guide the actual product flow on the assembly line.

In order to meet the customer requests there are two basic methodologies the

manufacturer can use. Some producers use the basic build-to-stock approach. This means that the products will be produced in any desired order (not based on individual customer orders) and stored in finished goods inventory. Then, as the customers' due dates near, the orders will be separated and delivered to the appropriate customers. The second approach, which defines the problem environment to be studied in this research, is referred to as build-to-order and is more complex, in terms of applying mixed-model sequencing. The *build-to-order* approach is used when the manufacturer does not schedule the product until the customer has placed the order. The manufacturer may produce all of a single customer's order, then build the next customer's order. A customer order may consist of a single type of product or a variety of several products. The advantage of this procedure is that it ensures fulfillment of customer demands from current production, thus maintaining minimum finished goods inventories. Producers of refrigeration units, furniture, tractors, heavy industrial and machine equipment, and defense hardware provide current-day examples of companies that use the build-to-order approach.

Customer requests for products can either be constant and non-changing, which is called static demand, or they can be continually changing, having *dynamic demand*. The environment that will be studied is considered to have a dynamic demand pattern with continual changes in the amount ordered of each type of product, as well as changes in the individual customer order after it has been placed.

Just-In-Time (JIT) systems are very important in the world of manufacturing today. The JIT pull system, with the special use of kanbans to control inventory flow for certain batch items, aides the producer in striving to supply the customer with the desired type of product, in the exact amount needed, at the requested time. Kanbans are used for certain items which require batch processing. These items are produced in a feeder line which is separate from the

main assembly line. The results include reduced inventory, shortened lead times, greater flexibility, and improved quality.

High volume final assembly lines are used to produce standard products which may each have a selection of various options. The line itself can either be machine paced or worker paced. All production that cannot be performed on the final assembly line according to a standard tact time is done separately on a feeder line so as not to slow down the final assembly process. Parts in feeder lines are not considered in this research due to the additional complexity that would be added to the problem. In this research, only whole unit finished components or parts that feed the final assembly line are considered.

3. Production Scheduling Goals

When a company is determining a production schedule in the build-to-order environment, there are several criteria that must be met and goals that are considered in order to efficiently and effectively meet the customers' demands. Four of the goals companies consider when scheduling in this environment are described in this section.

1. Meet The Customer Due Date

A basic goal is to *meet the customer due date*. The due date can affect the production schedule in varying ways. For instance, if the production sequence is scheduled in 'time buckets' of one week, or two weeks, etc., as is the case in many build-to-stock environments, then 'meeting the due date' means that the demand mix for each time period is met. If due dates are given as specific days, then meeting the due dates leaves little flexibility to meet other objectives.

2. Level The Pull Of Component Parts Through The System

A basic goal is to *level the pull of component parts through the system*. Smoothing the component pull consists of keeping a constant rate of usage of work-in-process, subassemblies, and raw materials in the system. This consistency helps to minimize inventory storage space and minimize required machine capacity while at the same time encouraging consistent and predictable deliveries from internal and external suppliers.

For example, assume the internal demand for a subassembly is allowed to fluctuate freely between high demand and low demand. During the peak demand periods, the capacity of the machines which process the subassembly must be able to meet the maximum demand, and the inventory of raw materials that supply the subassembly must be large. On the other hand, assume the internal demand for a subassembly is kept constant. There will be no peak demand periods, and the capacity of the processing machines and the inventory of raw materials will be minimized, while the system produces the desired amount of output.

Reducing machine capacity leads to a reduction in capital expenditures and possibly in direct labor costs. Reducing inventory leads to savings of investments, improved quality, and a more efficient production operation, with a reduction in maintenance problems, work methodology problems, and work center bottlenecks.

3. Balance The Work Load On The Assembly Line

Another important goal is to *balance the work load on the assembly line*. If a producer schedules several high work content items in series, then either of two events can occur. If the number of workers remains constant, then the elapsed time from completion of one product to

completion of the next (time between exits) will increase and the production efficiency decreases. An alternative to this scenario is to increase the personnel and maintain the desirable time between product exits.

Both alternatives may be considered disadvantageous for the producer. If the producer alternately schedules high work content items with low work content items, then the number of workers can be kept constant and the system can remain properly utilized. This form of production smoothing leads to a level work force that can flex to produce highly labor intensive products as well as products that require less labor. It also aids in minimizing the flow time of a product (the time it enters the production line, until the time it exits the system). This will result in a shorter lead time and therefore more flexibility in production and faster delivery of the product to the customer.

4. Minimize The Production Time Spread Of Customer Orders

Usually finished goods are delivered to customers in lot sizes based on packaging and shipping constraints. For this reason it is desirable to produce the portion of a single customer's order that is to be transported at one time. The production time spread within an order is the length of time for an order to be produced, that is, the finishing time of production of the last item in an order minus the starting time of production of the first item in the order. It is desirable to produce the necessary products in the amounts required to fill a load that will be shipped together. This method of sequencing jobs according to customer outbound loads avoids the situation of holding finished goods inventory longer than necessary and it can also minimize product damage due to excessive handling.

4. Research Objectives

The main objective of this research is to provide a set of new methods that can be used for adding insight concerning the tradeoffs between two of the four main sequencing goals: meeting customer due dates and leveling the pull of component parts on the final assembly line. These goals are two of the four associated with sequencing mixed-models in a build-to-order environment. This research will provide information that will increase the body of knowledge pertaining to mixed-model scheduling, while at the same time providing the working schedulers, in the build-to-order environment, pertinent information which will aid them in making knowledgeable scheduling decisions.

To achieve this objective, the research entailed the following:

- analytical description of existing objective functions to be used in the research,
- development of a new objective function,
- development of heuristic algorithms to generate high quality feasible solutions to the sequencing problem,
- analysis of sequencing procedures with regard to the two objectives on two populations,
- computational testing of the methods on two populations of test problems.

5. Organization Of Dissertation

In Chapter 2, we review the relevant literature. Chapter 3 presents the Mixed-Model Sequencing Problem. In Chapter 4 we provide details of solution approaches for the MMSP including the Border Swap heuristic, the 2-Optimal heuristic, and a Branch-and-Bound

procedure. Chapter 5 provides the results of computational testing of the various approaches. In Chapter 6, we present our conclusions and our directions for future research.

CHAPTER 2

Literature Survey

In this chapter we present the relevant literature on the MMSP. Section 1 describes the two general focuses of the literature to this point in time. Section 2 historically lists the specific references and provides a brief description of each. This is a broad survey which covers several aspects of mixed-model sequencing found in reference journals. The two research efforts most important to this research are Monden (1983) and Miltenburg (1989).

1. Introduction

Mixed-model scheduling was originally studied in the 1960's. The initial focus was based on the idea of balancing the assembly line so that the work was distributed evenly between work stations. More precisely, it was considered desirable to level the load on each work station so that the total assembly time for each model was equally balanced over all stations on the assembly line (Dar-El and Cucuy 1977). This logic promoted the design of a continual, feasible work schedule that reduced the number of times the workers had to stop the conveyor belt to complete unfinished work. All products do not have the same operation time at each station on the assembly line. If products with relatively longer operation times are successively scheduled, delays and line stoppages will result. Products should be sequenced so as to level the work load at each station. One recent similar goal is to sequence the jobs to minimize the total work overload, or equivalently to maximize the total work completed (Yano and Rachamadugu 1991).

With the evolution of the auto industry came the development of new questions concerning mixed-model scheduling. By the 1980's researchers began to address the issue of

leveling or balancing the schedule to keep a constant rate of usage of all parts used by the assembly line. That is, the quantity of each part used by the mixed-model assembly line per unit time should be kept as constant as possible. There should be very little variability in the usage of each part from one time period to the next (Hall 1983, p. 123, Japanese Management Association 1985, Miltenburg 1989). This goal was considered to be more important than the first goal with respect to JIT production systems (Monden 1983, p.182).

These papers do not specifically discuss two of the ideas presented in this dissertation. The first idea is that in the build-to-order environment, meeting the customer due date can be treated as a goal of production scheduling. This goal does not necessarily have to be met 100% of the time, but rather can be optimized in conjunction with the other sequencing goals. The second idea is that the customer shipment loads should be considered in the sequencing process, so that the customer order is produced within as small of a time frame as is feasible with consideration for the other goals.

This research concerning Mixed-Model sequencing is designed for use mainly in the simple final assembly line production environment. In the case of a flexible manufacturing system, there is a hierarchical framework with three levels as described in Raju and Chetty (1993): the strategic level that deals with long-term decision making; the tactical level that considers the master production plan; and the operational level which is associated with real-time operation. At the operational level, the schedule performance is very sensitive to system disturbances as is described in Kim and Kim (1994). The system requires quick responses to change based on the states of the elements of the system at the time. The sequencing in this research only deals with the products and components being produced and not with the choices

of production work flow through alternative paths.

What follows is a review of some of the published papers, with brief descriptions of each, pertaining to the two main goals of scheduling described in the first part of the introduction.

2. Literature Survey

The earliest work related to the mixed-model sequencing problem is by Kilbridge and Wester (1963), who defined the mixed-model assembly line sequencing problem and presented the variable-rate and the fixed-rate launching systems. Since then, several sequencing objectives have been studied.

Thomopoulos (1970) showed how to modify mixed-model line balancing algorithms to lead to smoother, more consistent, station assignments on a model by model basis. The objective was to distribute evenly the total daily workload to stations so there is an even flow of work along the line for any given model.

Early work such as Macaskill (1972) focused on the mixed-model balance problem, which is concerned with allocation of work to operators. The objective of this problem is to efficiently assign tasks to workstations when a variety of products are assembled on the same production line. The author introduces a formulation that facilitates the achievement of large-scale mixed-model balances by computer.

Macaskill (1973) used computer simulation to study the problem of sequencing a mixed-model assembly line under the assumption that a suitable mixed model balance has already been achieved. Operator idle time, congestion, work deficiency, and utility work are used to measure

assembly performance.

Dar-El and Cother (1975) developed a heuristic that can be used in the design of a mixed-model assembly line. The heuristic sequences models to minimize the overall length of a mixed-model assembly line for no operator interference. A complete factorial experiment was made on five factors to determine their influence on the overall assembly line length and recommendations regarding selection of these parameters are discussed relating to the design of mixed-model assembly lines.

Dar-El and Cucuy (1977) then extend the work of Dar-El and Cother (1975). A search procedure and integer programming techniques were used to solve the mixed-model sequencing problem for the case where the total assembly time for each model is equally balanced over all stations in the assembly line and all stations are closed. This condition allows for the study of only the model sequences in the first station since these are repeated identically in each of the other stations in the assembly line. An optimal sequence was obtained with the minimization of the overall assembly line length for zero station idle time. The algorithm is useful in the design and planning of new mixed-model assembly lines.

Okamura and Yamashina(1979) presented a heuristic procedure for scheduling various products (with different operation times) with the objective of minimizing the risk of stopping the conveyor when variability exists. They proposed a heuristic procedure for a single station for the objective of minimizing the maximum distance an assembly worker would need to proceed downstream from his workstation (if he were permitted to do so) to ensure completion of all work.

In the 1980's, researchers began to address the issue of leveling or balancing the

schedule so as to keep a constant rate of usage of all parts used by the assembly line, the second goal for sequencing that was described previously. They worked on scheduling techniques which result in very little variability in the usage of each part from one time period to the next, so that the quantity of each part used by the mixed-model assembly line per unit time is kept as constant as possible. This goal of emphasizing efficiency of component production can in general be achieved by minimizing the deviations between the linear usage of each part over time and the actual usage required by the assembly sequence. Variations on this "Goal Chasing" objective, which will be defined later, can be found in several recent publications.

Monden (1983) included information on mixed-model scheduling in his book which describes the Toyota production system. He indicated that Toyota considered the goal of leveling the load on each process within the line in Toyota's sequencing program and incorporated it in the solution algorithm which mainly considered a different goal of keeping a constant speed in consuming each part on the line. A Goal-Chasing Method (GC1) was summarized and evaluated. Then, a simplified algorithm Goal-Chasing Method II (GC2) was summarized. Both methods consider two levels of assembly and strive to linearize the usage of fabricated components which feed the assembly line.

Because Monden's GC1 method is one of the few heuristics that can be used when there is production of different models requiring different components, it lends itself well to the research of this dissertation. A description of the GC1 method can be found in Chapter 4.

Miltenburg (1989) developed sequencing procedures for the case where products require approximately the same number and mix of parts. This assumption was used so that a constant rate of part usage could be achieved by only considering the demand rates for the products. The

study is then limited to a single production level which when smoothed, automatically smoothes the part rate usage in the other levels of production. A nonlinear integer programming formulation was introduced which was designed to minimize the variation of the actual production from the desired production. He developed a procedure to obtain an optimal infeasible schedule, then used heuristics, based on variation from uniform part usage, to make the schedule feasible. He then discussed the property that the schedule is usually made up of a short sequence which is repeated.

Miltenburg and Sinnamon (1989,1992) extended the work done by Miltenburg (1989). This new work included multi-level production in the case where parts vary from product to product.

Inman and Bulfin (1991) use the Earliest Due Date rule to very rapidly and easily solve the optimization problem of minimizing the difference between completion times and due dates. These due dates correspond to calculated completion times of the individual products and are considered "ideal", but not necessarily integer feasible. This objective function is intuitively similar to other related objective functions.

Kubiak and Sethi (1991) formulate an assignment problem with the goal of leveling the schedule of a mixed-model assembly line in a Just-In-Time production system. This formulation is designed to accomplish the same objective as that of Miltenburg (1989).

Ding and Cheng (1993) present a much simpler and faster heuristic procedure compared to Miltenburg's best algorithm, with a solution of equal quality. The heuristic, like Miltenburg's, is based on minimizing the variation from a linear schedule, over two stages. This problem only considers products requiring approximately the same number and mix of parts.

Steiner and Yeomans (1993) use a graph-theoretic approach to determine an optimal schedule which maintains a constant rate of part usage. They show that a solution may be reduced to solving a series of decision problems, where it is possible to answer each decision problem in linear time, and that a schedule always exists where the difference between the actual number and the ideal number of units produced at any point in the schedule is at most one.

Other recent research has been done in an area which emphasizes the efficiency of assembly performance. Bolat (1988) provides solutions to the mixed-model assembly line sequencing problem which minimizes total set-up cost and utility work. Parrello (1988) describes the difficulties associated with his attempt to use an expert system to schedule the cars on the assembly line of an automobile manufacturer that averages a production level of 5000 cars per day. McCormick, Pinedo, Shenker, and Wolf (1989) sequence to operate the line so as to maximize throughput. They propose heuristic approaches based on an equivalent maximum flow problem and on critical path techniques.

Yano and Rachamadugu (1991) sequence to minimize the total amount of incomplete work on a paced assembly line. Their objective is concerned with achieving good average performance. For the case where there is one worker per station, this is similar to the objective of Okamura and Yamashina (1979), who sequence to minimize line stoppages (without guaranteeing good average performance).

Sumichrast, Russell, and Taylor (1992) use simulation to compare five sequencing methods according to work not completed (in time units), worker idleness, worker station time (working and idle), and mean squared deviation from uniform component usage. The methods compared include Monden's Goal Chasing 1 (GC1) and Goal Chasing 2 (GC2), Miltenburg's

Algorithm 3 using Heuristic 2 (M-A3H2), batch sequencing, and a procedure developed by the authors called Time Spread (TS) which is similar to GC1, but with the objective of smoothing the work at each station. For the component usage measure, M-A3H2 performed the best. For the three measures of assembly line efficiency, TS performed best in general.

Most of the research papers focus on only one goal at a time. In a new edition of Monden's book describing the Toyota production system, there is reference to heuristic procedures which attempt to sequence products based on the component usage goal, then update the sequence based on the workload balancing goal. There does not seem to be much published research that attempts to combine goals. In the build-to-order environment there is emphasis on not only the two goals mentioned, but also on sequencing orders according to due dates and final customer shipment loads.

CHAPTER 3

Mixed-Model Sequencing Problem

Chapter 3 presents a physical example of the Mixed-Model Sequencing Problem and defines the two main objectives involving due date restrictions and minimizing the Mean Square Deviation (MSD) associated with the deviation of actual component usage from a linear goal.

1. The Sequencing Model

Let D = the number of items to be sequenced. A feasible solution I to the sequencing problem is a set of D items comprising a feasible schedule. The constraints of the problem are written as

$$I \subseteq F, \quad (3.1)$$

$$\|I\| = D, \quad (3.2)$$

where F is the set of items from which we can select. The set of allocations satisfying (1) and (2) will be denoted X . There are $O(D!)$ possible solutions. The problem description might be clarified with a physical example.

Suppose there are three different models of a product to be produced. If we are sequencing cars, we might consider one to be a red car with a sunroof and manual transmission, the second a blue car with a plain roof and automatic transmission, and the third a white car with a plain roof and automatic transmission. The models can be represented as model 1, model 2, and model 3 or simply as 1, 2, and 3. Each model requires different components or parts. The sequencing goals are created concerning those different parts, the amount of time it takes to put

in a part, the inventory ordering policies for ordering the part, etc.

Suppose we are to produce two model 1's, three model 2's and four model 3's. A sequence based on batch techniques might be as follows:

Sequence 1: 1, 1, 2, 2, 2, 3, 3, 3, 3.

If we decide to 'mix' the models to take advantage of some of the results associated with mixed-model sequencing, we might determine two possible sequences as follows:

Sequence 2: 3, 2, 1, 3, 2, 3, 1, 2, 3

Sequence 3: 1, 2, 3, 3, 2, 1, 2, 3, 3

There are several ways a sequence can be derived. The number of possible sequences associated with this example are $\frac{9!}{2!3!4!} = \frac{362,880}{288} = 1260$.

2. Objective Function

This research deals with two objective functions:

$$\text{minimize } L(I), \quad (3.3)$$

$$\text{minimize } P(I), \quad (3.4)$$

where $L(I)$ is the due date (number late) objective, and $P(I)$ is the part objective.

1. First Objective

The number late objective function is

$$L(I) = \sum_{i \in I} \sigma(l_i) \quad (3.5)$$

where

$$l_i = C_i - r_i, \quad (3.6)$$

$$\begin{aligned} \sigma(l_i) &= 1 \text{ if } l_i \geq 1 \\ &= 0 \text{ otherwise} \end{aligned} \quad (3.7)$$

L = the total number of late items,

l_i = the lateness of the i^{th} item,

C_i = the completion date of the i^{th} item, and

r_i = the request date or due date of the i^{th} item.

Because the focus of this research is on the relationship between the two objectives, meeting the due dates and leveling the part usage, it is not necessary to specifically attempt to minimize the number of late items. Instead, the number of allowable late items will be specified for each problem and the square deviation of actual part usage from uniform part usage will be minimized. This methodology will produce the required results.

2. Second Objective

An important objective in mixed-model scheduling is to level or balance the schedule so that the quantity of each part used by the assembly line per unit time is kept as constant as possible. That is, there should be very little variability in the usage of each part from one time period to the next.

We will more specifically define this goal. When products are systematically placed in a sequence one at a time, they are placed in 'stages'. Determining which product to sequence first, is the determination of the stage 1 product. At stage 2 there are two products in the sequence, etc. At each stage of the sequence, we can calculate the mean square deviation of actual part usage from the linear part usage goal. For example, suppose we wish to create a

sequence with 5 items, 2 items of model type 1 and 3 items of model type 2 as shown in Table 1. They each have 2 parts. Model 1 has parts 1 and 2, while model 2 contains parts 2 and 3.

From this information we know we will have 2 part 1's, 5 part 2's, and 3 part 3's as shown in Table 2. To maintain linear part usage it is desirable at the first stage of production to sequence an item that contains $2/5$ of a part 1, $1 (5/5)$ part 2, and $3/5$ of a part 3. This is the goal at stage 1 as can be seen in Table 3.

If we choose to sequence model 2 at stage 1 the part usage rate at stage 1 will be 0, 1, 1, representing the usage of parts 1, 2, and 3 respectively. The deviation is then $-2/5, 0, +2/5$ in stage 1. If we choose to sequence model 1 in stage 2, the usage rate at stage 2 will be 1, 2, 1 with deviations of $+1/5, 0, -1/5$. This process is continued in the same way for the remainder of the sequence as illustrated in Table 4.

Table 1: Sample Problem.

<i>Model</i>	<i>Model 1</i>	<i>Model 2</i>
# Produced	(2)	(3)
Parts:	1,2	2,3

Table 2: Total Parts for Sample Problem.

	<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>
Total Parts	2	5	3

Table 3: Goal Production per Part for Sample Problem.

	<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>
Stage 1	2/5	1	3/5
Stage 2	4/5	2	6/5
Stage 3	6/5	3	9/5
Stage 4	8/5	4	12/5
Stage 5	2	5	3

Table 4: Part Usage Rate and Deviation for Sample Problem.

<i>Stage</i>	<i>Model</i>	<u><i>Cum.</i></u>	<u><i>Part</i></u>	<u><i>Usage</i></u>	<u><i>Deviation</i></u>		
		<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>	<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>
1	2	0	1	1	-2/5	0	2/5
2	1	1	2	1	1/5	0	-1/5
3	2	1	3	2	1/5	0	-1/5
4	1	2	4	2	-2/5	0	2/5
5	2	2	5	3	0	0	0

The mean square deviation or error is derived as follows:

$$\text{MSD} = \frac{\sum_{\text{Stages}} \sum_{\text{Parts}} \left[\left(\frac{\text{Cummulative Production}}{\text{Goal At Each Stage}} - \frac{\text{Actual Cummulative Part}}{\text{Usage At Each Stage}} \right)^2 \right]}{(\# \text{ Stages} - 1) \bullet (\# \text{ Parts})} \quad (3.8)$$

The “Production Goal At Each Stage minus Actual Part Usage At Each Stage” represents the deviation. For this example the total square deviation = $(-2/5)^2 + (0)^2 + (2/5)^2 + (1/5)^2 + (0)^2 + (-1/5)^2 + (1/5)^2 + (0)^2 + (-1/5)^2 + (-2/5)^2 + (0)^2 + (2/5)^2 + (0)^2 + (0)^2 + (0)^2 = 0.8$. The mean square deviation is then $\frac{0.8}{(4)(3)} = 0.0667$.

This second objective can be expressed as follows:

$$\text{minimize } \frac{\sum_{k=1}^Q \sum_{j=1}^c \left(\frac{kN_j}{Q} - A_{jk} \right)^2}{(Q-1) \cdot c} \quad (3.9)$$

Where,

A_{jk} = the quantity of the fabricated component part j required to assemble the first k units of the actual sequence of end items

c = the number of different components

k = the current position in the sequence

N_j = the total quantity of the fabricated component part j required to produce all Q items in the sequence

Q = the total production of all models.

The first term in the equation, $\frac{kN_j}{Q}$, represents the quantity of component j required to assemble the first k items in the sequence given a constant usage rate. The other term in the equation represents the actual cumulative usage for a specific partial sequence.

CHAPTER 4

A Solution Approach For The Mixed-Model Sequencing Problem

In Chapter 4 we provide details of solution approaches for the MMSP. We begin with a general approach for evaluating multi-objective problems, then we describe an exact Branch-and-Bound procedure for optimally solving the sequencing problem. Next, three general types of heuristic algorithms for sequencing are described; Monden's Goal Chasing 1 (GC1), along with a new version of the Goal Chasing method called GC1-Look-Ahead, a 2-Optimal heuristic, and the new Border Swap heuristic.

Following the three general heuristic summarizations is a discussion of the evaluation tool that is used to produce and compare the various sequencing solutions. The effects of the evaluation tool on the selection of a sequence are studied. A heuristic algorithm is given for the general case of developing and evaluating a sequence using any combination of the three basic heuristic algorithms.

Next, we present various combinations of the three basic heuristic algorithms all of which are designed to achieve particular objectives. The first objective to be studied involves minimizing the mean square deviation of actual part usage from the linear part usage goal, while having zero late items. Then we present algorithms for sequencing with the second objective of minimizing the mean square deviation from linear part usage while allowing any number of items to be sequenced late. Last, we present algorithms for sequencing with the third objective of producing sequences that attempt to minimize part deviation with 1 item late, 2 items late, etc. so that a Mean Square Deviation (MSD) - Lateness tradeoff curve is developed. This curve is designed to show the effect of an additional late item on the MSD of component usage.

1. General Approach

The approach will involve the development of a sequencing method which incorporates the two objectives listed below and a study of the relationship between them.

1. Meet the customer due date.
2. Keep a constant rate of usage of all parts on the final assembly line.

In multiobjective optimization, when it is not possible to optimize a schedule with regard to all of the goals, there is sometimes a pattern of tradeoffs which leads to a better solution. For instance, if the optimization of the first goal is relaxed by a small percentage, then it might be possible to bring the second goal closer to optimization. Stated more formally, given a set of feasible schedules X and a set f_i , in this case $i = 1, 2$ of objective functions on X , the problem is to find the most 'preferred' vector of objective function values $F(x) = (f_1(x), f_2(x))$ (Gilbert et. al. 1985). A survey by Rosenthal (1985) identified three general classes of approaches for multiobjective problems; explicit utility maximization, implicit utility maximization, and partial generation of the efficient set. We have chosen a method from the third category for dealing with the scheduling problem.

Partial generation of the efficient set involves identifying a subset of efficient solutions. Efficiency, (also known as noninferiority, nondominance, Pareto-admissability, and Pareto-optimality) is defined in terms of dominance. "Given feasible points $y, z \in X$, z is dominated by y if and only if

$$f_i(y) \geq f_i(z), i = 1, \# \text{ goals}$$

with strict inequality for at least one objective. Then, a feasible solution $x \in X$ is efficient if and

only if there does not exist a solution $y \in X$ which dominates x " (Rosenthal 1985).

When an efficient solution is generated, alternate efficient solutions can be formulated with the following characteristics: one of the objective values is improved, one of the objective values is degraded, and the remaining objective values, if any, remain the same. In other words, with multiobjective problems, there is a series of tradeoffs associated with various efficient solutions. Our work should give insight into the tradeoffs which will aid the decision maker. This process is explained in Gilbert et. al. (1985) as a tradeoff ratio = (improvement in f_j) / (decrement in f_m) for each new efficient point generated where $j \neq m$ and $j, m = 1, 2$. For example, a schedule that optimizes the first objective of sequencing orders according to customer due dates, can be changed so that this objective is slightly degraded in order to accomplish a large improvement in the second objective of keeping a constant rate of usage of all parts on the final assembly line.

As part of solving the scheduling problem, heuristic procedures will be used to find solutions that are close to nondominated. From these schedules the process described above involving tradeoffs in objective values can be tested, then the results will be analyzed regarding these tradeoffs that provide information to the scheduling decision makers.

2. The Branch-And-Bound Approach For Optimal Sequencing

This Branch-and-Bound procedure is designed to produce an optimal solution to the sequencing problem in terms of minimizing the mean square deviation from linear component usage, while meeting the specifications concerning due date restrictions. It is based on a depth first tree search with procedures for fathoming branches of solutions and backtracking up the tree until every possible solution has been explored. The 'best solution' is updated and compared

with each feasible solution and the resulting best solution after all branches have been evaluated is considered to be a feasible optimal solution for that problem.

A solution tree is created from a data list that is sorted by due date first, then model within due date. This sorted list forms the left most branch of the tree. The branches to the right are then formed by systematically creating all possible solutions beginning with the exchange of the last two items sequenced as shown in Figure 1. No exchanges are made that break due date restrictions.

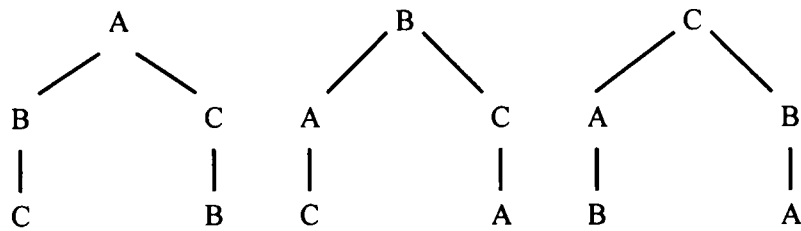


Figure 1: A Branch and Bound Solution Tree with Three Items Sorted in Order A, B, C that Includes All Possible Solutions, Feasible and In-feasible.

There are two main procedures for fathoming solution branches, the upper bound procedure and the total rounding error procedure. The upper bound fathoming procedure follows. An upper bound for the mean square deviation from linear component usage is found from a heuristic procedure designed to minimize the mean square deviation of component usage. Examples of such heuristics are described later in the chapter. This upper bound is then used as input to the branch-and-bound program. As the program creates a partial solution by adding one item at a time in sequence, a running square deviation is tracked. The deviation measure is a result of the square deviation of all parts over each stage of the solution. Note that the individual deviations of each part are not tracked separately, so all parts are treated equally. If this

deviation sum exceeds the upper bound at any point in time, the branch is cut and fathomed, then a backtracking procedure begins until a new branch is explored.

The second main procedure for fathoming branches is based on rounding error. There is an inherent error associated with integer sequencing when the goal is linear as will be shown. The following example uses the data of the problem from Tables 1, 2, and 3 in Chapter 3. The goal for part 1 in stage 1 is $2/5$ but only integer sequencing is permitted. The closest integer to $2/5$ is 0. The absolute value of the difference between the rational goal and the integer goal is $2/5$. This is the inherent rounding error associated with part 1 in stage 1. The total square rounding error associated with stage 1 is the sum of the squared errors for the three parts; $(2/5)^2 + 0^2 + (2/5)^2 = 0.32$ as shown in Table 5.

When the branch-and-bound program is calculating which item to sequence in the stage 1 position, it first checks the amount of inherent remaining rounding error that will be unavoidable in the remaining stages of sequencing. This number is the sum of the remaining total square rounding errors including the current stage. In this example the inherent remaining

Table 5: Example of Inherent Rounding Error for Sample Problem.

<i>Stage</i>	<u><i>Part Usage Goal</i></u>			<u><i>Inherent Rounding Error</i></u>			<i>Total Square Rounding Error</i>	<i>Remaining Rounding Error</i>
	<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>	<i>Part 1</i>	<i>Part 2</i>	<i>Part 3</i>		
1	$2/5$	1	$3/5$	$2/5$	0	$2/5$	0.32	0.8
2	$4/5$	2	$6/5$	$1/5$	0	$1/5$	0.08	0.48
3	$6/5$	3	$9/5$	$1/5$	0	$1/5$	0.08	0.4
4	$8/5$	4	$12/5$	$2/5$	0	$2/5$	0.32	0.32
5	2	5	3	0	0	0	0	0

rounding error is 0.8.

This same method of evaluation is used at each stage of sequencing. For each component part, the best integer solution (not necessarily feasible) is compared with the optimal rational solution. The resulting square deviations are summed over all parts at each stage giving a total minimum square deviation resulting from rounding for each stage of sequencing. Each time the branch-and-bound program generates a partial solution of any length, there is a check for fathoming based on the rounding error. If the square deviation of the partial sequence added to the remaining inherent rounding deviation is greater than the upper bound, then the branch is fathomed.

In the case where there are no due date restrictions for sequencing, there is a much larger set of feasible solutions or branches. The upper bound determined by a heuristic procedure is generally smaller and therefore allows for a relatively faster fathoming of each branch. The result is a solution that may have some late items but with the smallest feasible mean square deviation from linear part usage.

As mentioned earlier, the branch-and-bound program is designed to find a solution with zero items late, or the number of late items that will allow for the smallest possible mean square deviation from linear part usage. In addition, there is the option to specify the exact number of late items to be sequenced. Steps in the algorithm can be found in Appendix A.

3. Three Basic Types Of Algorithms For Sequencing

In this section we discuss the three basic types of heuristic algorithms for sequencing; Monden's Goal Chasing 1 method along with a new GC1-Look-Ahead method, the 2-Optimal Algorithm, and the Border Swap Algorithm. All of the other algorithms that appear in this

research are derived from one or a combination of these three.

1. Monden's Goal Chasing 1

GC1 is the method for mixed-model sequencing that Monden originated for use by Toyota. The following is a description of GC1 as presented in R. T. Sumichrast et al. 1992:

"GC1 constructs a model sequence one position at a time. After $k - 1$ positions in the sequence have been assigned, GC1 selects a model for position k if (1) more units of this model are required and (2) if no other model for which production is required has a smaller 'penalty' for non-uniform component usage. The formulation of this procedure including the penalty for non-uniform parts consumption is as follows.

Starting at the first position in the sequence, a model is chosen that will

$$\underset{i \in M'}{\text{minimize}} \sum_{j=1}^c \left(\frac{kN_j}{Q} - A_{j(k-1)} - b_{ij} \right)^2 \quad (2.1)$$

where,

$A_{j(k-1)}$ = the quantity of the fabricated component part j required to assemble the first $(k - 1)$ units of the actual sequence of end items

b_{ij} = the quantity of component j required to assemble one end item i

c = the number of different components

k = the current position in the sequence

M' = the set of models which requires more production than is scheduled in the sequence up to position $k - 1$. M' is updated each time an item is added to the sequence

N_j = the total quantity of the fabricated component part j required to produce all Q items in the sequence

Q = the total production of all models.

The first term in the equation, $\frac{kN_j}{Q}$, represents the quantity of component j required to assemble the first k items in the sequence given a constant usage rate. The other two terms in the equation represent the actual usage for a specific partial sequence with the last term having value of either 0 or 1. The squared differences between these terms are summed over all component parts to provide a measure of non-uniform usage. Position k in the sequence is filled by the model which minimizes this measure. As the sequence is constructed, at each step (i.e. each time a model is assigned to a position in the sequence) the 'distance' between the expected usage of components and actual usage is minimized, however overall minimization is not assured.

This heuristic does not require excessive computational effort if programmed efficiently. One model, from M models, is selected for each of Q positions. Thus the number of comparisons of models is bounded by QM . For each comparison c additions are required, thus the number of addition operations is approximately cQM ."

"Goal Chasing 1 Look Ahead" is our effort to update GC1 to obtain an improved sequencing solution. GC1 is a myopic heuristic because it greedily selects a model to sequence in the current position without regard to the variation in the future stages. GC1-Look-Ahead has a built in 'look-ahead' capability as compared with the original GC1. This idea was incorporated into another variation of the sequencing problem by Miltenburg (1989) for the case when all products use similar parts.

For each stage, all eligible products are noted and an eligible product is tentatively scheduled. Next, a product is sequenced in the following position by the GC1 method. Now with two tentative products in the current and following position, the total variation is calculated. The process is repeated with each eligible product available for sequencing in the current position.

The product that will be permanently sequenced in the current position is selected based on the pair which minimizes the total variation. The current position is sequenced, and the process begins again with the new current position which was previously the 'following' position.

This new method requires the extra step of tentatively trying every eligible product in each position of the sequence. This extra step is designed to reduce the effects of the greed designed into the GC1 heuristic.

GC1-Look-Ahead Algorithm

Step 1: Temporarily schedule an eligible product in the current position and set the variation equal to some large number. This item is the incumbent and the variation is the minimum variation.

Step 2: Sequence a product in the following position using the GC1 method.

Step 3: Calculate the total variation. If the calculated variation is less than the minimum variation, then the new variation becomes the minimum variation and the item sequenced in the current position is the new incumbent.

Step 4: Select the next eligible product and go to step 1. If all products have been tested, then stop. The product sequenced permanently in the current position is the incumbent associated with the minimum variation.

Step 5: Let the new current position equal the current position plus one. Repeat the process beginning with step 1 until all positions have been permanently filled.

2. The 2-Optimal Algorithm

This algorithm begins with an existing sequence and swaps each feasible pair of items in the sequence, within due date restrictions, to try and reduce the total deviation of part usage from linearity. Each time a swap is made, the procedure begins again with this 'new' sequence until there are no more combinations of 2 items that, if exchanged, will reduce the deviation. The 2-optimal algorithm can be designed in three different ways. The first method does not allow any late items. All of the items are not swapped, only those with common due dates are exchanged. The second method has no regard for due dates in terms of swapping items in the sequence; therefore, the resulting sequence may have any number of late items. The third method allows a specified number of late items (X) and only allows swaps with less than or equal to X items late. This procedure will find the two optimal solution for $\leq X$ items late.

A General Form of the 2-Optimal Algorithm

Step 1: Obtain an initial sequence by the specified method.

Step 2: Temporarily swap every combination of two items that does not break any due date restrictions. Determine the combination that produces the largest decrease in the mean square deviation of actual part usage from linear part usage.

Step 3: If there is a decrease in the MSD, swap the two items determined in the last step to produce a new initial sequence and go back to step 2. If there is no decrease in the MSD then stop, the most recent sequence is the 2-optimal sequence

An example of this method is given later in Chapter 6, section 4.

3. The Border Swap Algorithm

The following algorithm will sequence items based on the goal of minimizing the mean square deviation associated with leveling the usage of component parts over time while allowing any number of late items. The algorithm is designed to start with a sequence in which no items are produced late, and which has a relatively large deviation. The next sequence generated allows for one item to be produced late while decreasing the overall deviation of component usage as much as possible. A series of sequences is produced until there is no improvement in the deviation. This final sequence is a heuristic solution for minimizing the deviation in component usage over time. In essence, this algorithm will produce a set of data points which can be graphed to show the relationship and tradeoffs between the number of items produced late, and the deviation of component part usage. An example of the Border Swap algorithm can be found in Appendix B.

Border Swap Heuristic

Step 1: Sort items in order by due date.

Step 2: Produce a sequence based on the Goal Chasing 1 Look Ahead (GC1-LA) method such that all of the items are used in the calculations, but only the items which match the desired due date are allowed to be sequenced so that there are no late items.

Step 3: Examine the individual deviations of the items sequenced at the end of a time period and at the beginning of a time period. Add the two deviations of each time period 'border', that is the deviation of the last item sequenced in a period, and the deviation of the first item sequenced in the following period. Determine the highest 'border pair', that is the highest sum of deviations at the time period border.

Step 4: Now consider the high border pair. Swap the production dates of the two items so that the item produced last in its time period will now be produced in the following time period, and the item produced first in the next time period, will be produced in the previous time period.

Step 5: Produce a sequence based on GC1 with this new change thus developing a sequence with one item late and a reduced overall deviation.

Step 6: If the overall absolute deviation is reduced then go back to Step 3. Otherwise return to the sequence which generated the lowest deviation and continue with Step 7.

Step 7: Find the second highest deviation of a border pair and swap the production dates.

Step 8: Produce a sequence based on GC1-LA with this new change. If the overall deviation is lower, then go back to Step 7 (replacing second with third, etc.), else go back to the sequence that generated the lowest deviation and Stop.

The Border Swap heuristic incorporates the GC1-LA method in each iteration. In this heuristic, the terms $\frac{kV_j}{Q}$, and $A_{j(k-1)}$, are calculated based on the total demand of all products over all of the time periods. But only the products which have the same due date as the production period for which we are scheduling, are considered eligible to be sequenced. The exception to this is following the first iteration when specific items are assigned to be produced one period earlier or one period later than the original due date.

During the first iteration, because of the due date restriction in sequencing, products which ideally would have been sequenced in a later time period were forced to be sequenced earlier. The effect of this is a product sequenced at the end of a time period with a relatively

large deviation. This means that one or several of the components used in assembling this product are 'overused' for that time period. If leveling the component usage is the main goal in sequencing, then it might be advantageous to move the product to production on a later date.

The due date restrictions also force products to be sequenced later than the component usage dictates. A product sequenced at the beginning of a time period (other than the first), will have a relatively large deviation if the components used to assemble the product are 'under used' in the previous time periods. If leveling the component usage is the main goal, then it might be advantageous to move the product to production on an earlier date.

In each iteration, the product 'border pair' with the highest combined deviation is reassigned to another time period for production. The product originally at the end of the time period is reassigned to the following period and the product at the beginning of the time period at the border, is reassigned to be produced in the previous time period. Basically the due dates of the pair are 'swapped'. After the products are reassigned, the GC1 Look Ahead method is used to recalculate the sequence. The result is a sequence with the two products systematically sequenced within their new time periods for production in a way that moves toward the goal of leveling the usage of component parts.

4. The Decomposition of the Part Leveling Goal into Two Separate Goals

This section illustrates the current problem with sequencing to level component production when customer due dates are considered, if one of the common heuristics is used (for example, Monden's Goal Chasing 1). The heuristic considers the sequence with respect to the big picture, while ignoring the effects on the smaller picture. This problem will be explained and addressed in this section, along with a suggested solution for overcoming the problem.

This research involves three different circumstances for sequencing: level part usage when there are zero items allowed late, level part usage allowing exactly 'X' items to be sequenced late, and level part usage when there are no due date restrictions and any number of items may be sequenced late without penalty. These three circumstances span all of the possible situations. There are two objectives that we will look at with respect to each of the three circumstances. A given sequence can be derived based on a multi-period, or overall goal for part usage, or it can be derived based on a single period goal thus becoming several independent problems.

After the sequence is derived, it can be evaluated with respect to an overall goal or with respect to single period goals. Logically the sequence derived according to an overall goal will perform better when evaluated by an overall objective. Likewise, the sequence derived according to single period goals will perform better when evaluated by the single period objectives. The following section will explain the relationship between the two objectives and how they affect this sequencing problem.

1. The Multi-period Goal Versus the Single Period Goal

In the problem examples tested in this paper, there are four time periods involved. When the sequence must be constructed allowing no late items, the problem decomposes into several independent smaller problems, in this case four. The appropriate objective is to level the parts based only on each individual time period. The logic behind this reasoning can be explained by the following simple example.

Assume that we are establishing a sequence based only on 'part 1'. The average level usage of part 1 over one week's production is calculated to be six per hour. Suppose that the

products due in day one contain less than half as many part 1's as do the products in the rest of the week. If we estimate the level usage of part 1 in day one it is three per hour. Here we look at two alternative methods of production. In the first case we sequence the part 1's based on the weekly goal of six per hour as shown in Figure 2. Since the goal is to minimize deviations, and the algorithms used to sequence are based on greedy methods, the items containing part 1 will be sequenced first until they run out, then there will be no more part 1's left to sequence. The total deviation from day one calculated from Figure 3 is: $\text{Deviation} = 0^2 + 0^2 + 0^2 + 0^2 + 6^2 + 12^2 + 18^2 + 24^2 = 1080$.

If on the other hand, in day one the part 1's are sequenced based on level usage for day one only, the results are shown in Figure 4. The total day one deviation from the weekly usage goal is calculated from Figure 5: $\text{Deviation} = 3^2 + 6^2 + 9^2 + 12^2 + 15^2 + 18^2 + 21^2 + 24^2 = 1836$. This shows a higher deviation due to the cumulative effect of the deviations from hour 1, 2, etc. Intuitively, the second method for sequencing, based on a daily goal, provides a more desirable sequence even though it shows a higher deviation when evaluated according to the weekly goal. The part usage is smoother and more evenly distributed than in the first method which sequences according to the weekly goal. This discrepancy leads to the need for distinguishing the sequencing situation and establishing the goals accordingly.

Now we will look at an example of a sequence that begins with zero items late, then one item late, then any late, which in this case is two items. Suppose we are sequencing eighteen items over three time periods, or six items per day. There are 3 model 1's to be produced, and 15 model 2's. The first sequence in Table 6 is a result of sequencing based on the overall or three period goal. 1.172 is the Mean Square Deviation from uniform component usage based on an

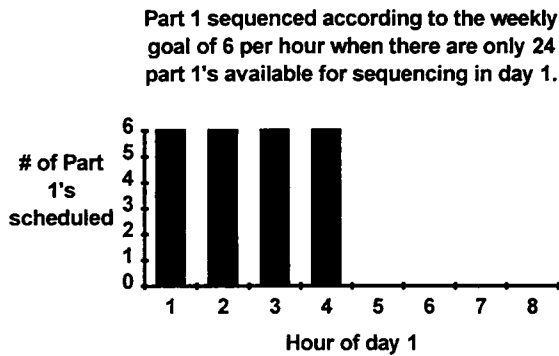


Figure 2: Sequence Based on Weekly Goal.

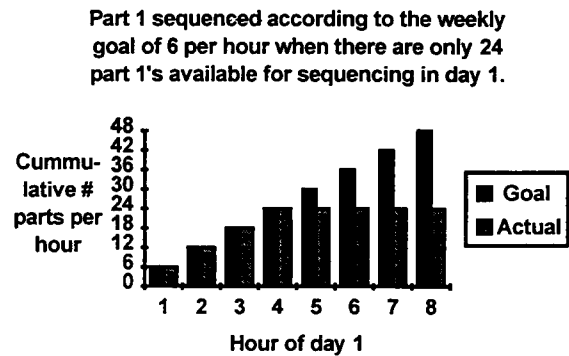


Figure 3: Deviation from Weekly Goal.

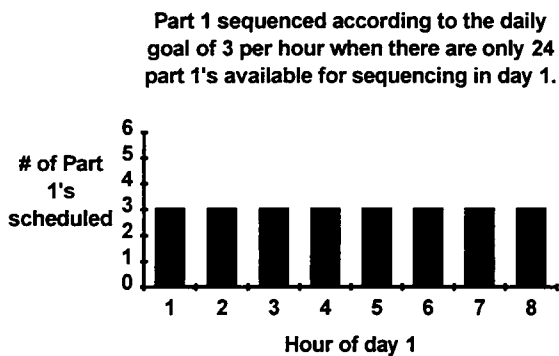


Figure 4: Sequence Based on Daily Goal.

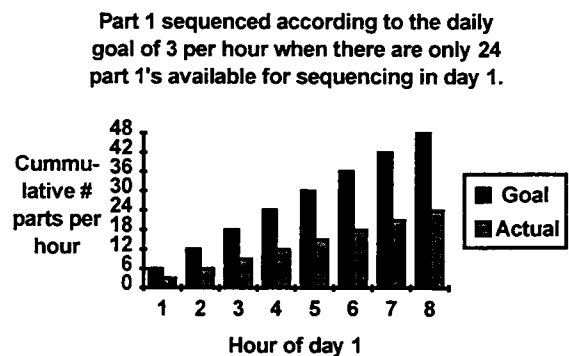


Figure 5: New Deviation from Weekly Goal.

overall goal. The MSD from uniform component usage based on three one period goals is 0.162. This MSD is the sum of the three individual MSD's.

The second sequence in Table 6 is a result of sequencing based on three individual goals. The MSD based on the overall goal is 1.446, while the MSD based on the one period goals is 0.044. We can see that as in the example before, the smallest MSD depends on the criterion used for evaluating the objective. Figure 6 also shows this to be the case for one item late and two items late.

Table 6: Example of a Problem Comparing the Two Conflicting Goals.

	Period 1	Period 2	Period 3	Overall Goal MSD	1 Period Goal MSD
0 Late	2 2 1 2 1 1	2 2 2 2 2 2	2 2 2 2 2 2	1.172	0.162
	1 2 1 2 2 1	2 2 2 2 2 2	2 2 2 2 2 2	1.446	0.044
1 Late	2 2 1 2 2 1	2 2 2 2 2 1	2 2 2 2 2 2	0.328	0.155
	2 1 2 2 1 2	2 2 1 2 2 2	2 2 2 2 2 2	0.721	0.057
2 Late	2 2 1 2 2 2	2 2 1 2 2 2	2 2 1 2 2 2	0.093	0.093

This concept of an overall goal and a single period goal are important in the area of sequencing. Using a greedy heuristic can be misleading. The heuristic will guide the sequence based on a partially unachievable goal. In general, it appears that it is better to design the heuristic to do something achievable based on the parts on hand that can be sequenced during the time period. This concept supports the idea of evaluating the length of the time periods based on the specific sequencing environment; a topic which can be studied in the future.

2. Application of These Results to the Problem

For each number of items late, the first sequence was generated using the overall goal. This is a fast easy method for determining which model will be produced late, and which will be produced early. When these models are chosen, the second sequence is generated based on the new designated production dates of the models. To generalize, the proposed heuristic sequencing method for 'X' late and 'any' late involves first producing a sequence based on the overall goal, then modifying the sequence, given the new production dates, according to the single period goals. The final sequence then is conditioned on the preprocessing of the initial sequence with respect to the overall goal.

Figure 6 shows a graph of the MSD's for the example problem with two models. From the graph it can be seen that the MSD of the sequence produced with a multi-period, overall goal improves with the addition of each late item up to two late items. This is viewed in the 'mp-mp' curve, where the first 'mp' indicates that the sequence was derived based on multi-period or overall goals, and the second 'mp' indicated that the sequence is evaluated according to a multi-period objective. Given each new improved sequence from the overall goal, the single period goals were considered by 'juggling' the order of the models within each period. The result of the new 'juggled' or leveled sequence is seen in the 'sp-sp' line.

Figure 6 also reveals another important relationship between the mp-mp curve and the sp-sp curve. In the example given, the final sequence with two items late is the same when the models are ordered to meet the overall goals as when they are ordered to meet single period goals. There can be no improvements for either goal. They are bounded by each other. The final sequence is optimal for the problem with no due date restrictions.

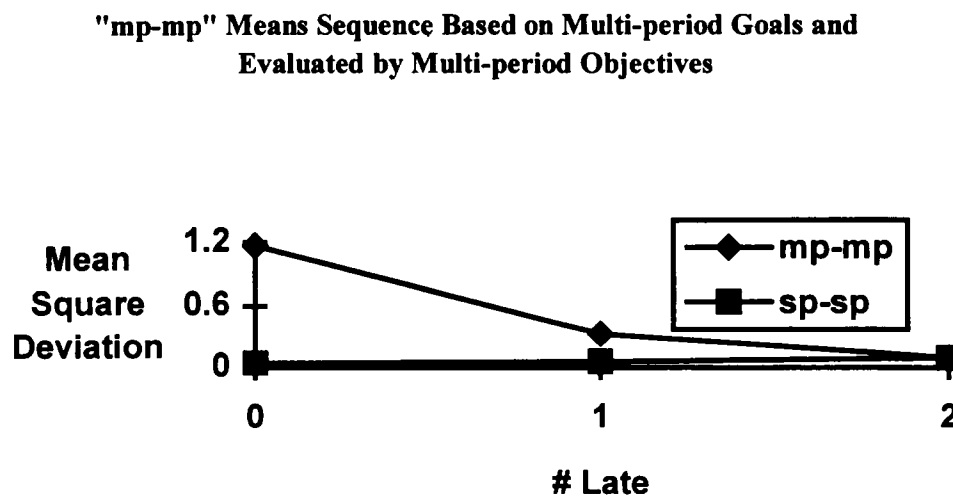


Figure 6: Graph of the Mean Square Deviation for the Example Demand Pattern.

Since heuristic methods are used to obtain the sequences, the optimal solution will not always be produced. Figure 7 shows this to be the case for the problem with twenty items and demand pattern C. When five items are allowed to be late, the MSD based on the multi-period objective is 0.121 while the MSD based on the single period objective is 0.118. There is a gap of 0.003.

5. A Summary Of The Three Objectives And The Heuristics Applicable To Each

A summary of the three objectives, the heuristics that are applicable for each objective, and a brief description of each heuristic can be found in Table 7.

"mp-mp" Means Sequence Based on Multi-period Goals and Evaluated by Multi-period Objectives

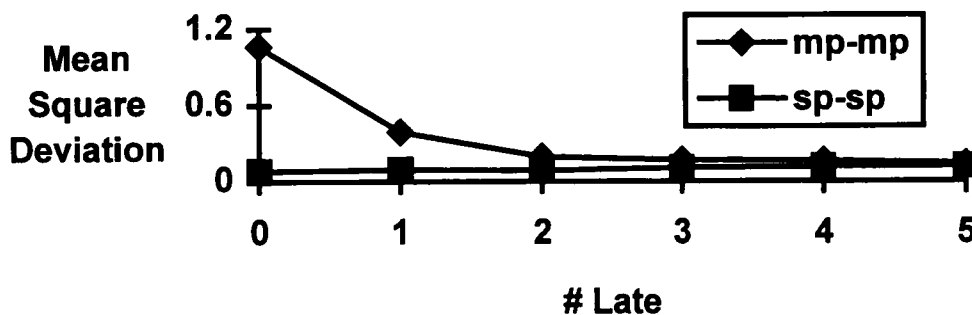


Figure 7: Graph of the Mean Square Deviation for Demand Pattern C.

Table 7: Summary of Objectives and Heuristics.

Objective	Heuristic	Brief Heuristic Description
Level Part Usage (Zero Items Late)	GC1-dd (4.6.1)	Monden's Goal Chasing 1 method is used to produce a sequence with no late items.
	GC1LookAhead-dd (4.6.2)	Monden's Goal Chasing 1 method with an additional feature to take into account two periods of deviations is used to produce a sequence with no late items.
	GC1LookAhead-dd / 2 Opt-dd (4.6.3)	The 2 Optimal procedure with zero late items is performed on a sequence obtained from the GC1LookAhead-dd method.
	Sort-dd / 2 Opt-dd (4.6.4)	The 2 Optimal procedure with zero late items is performed on a sequence obtained from a sorted sequence.
Level Part Usage (No Due Date Restriction)	GC1 (4.7.1)	Monden's Goal Chasing 1 method produces a sequence with no restriction on the number of late items.
	GC1LookAhead (4.7.2)	The GC1LookAhead method produces a sequence with no restrictions on the number of late items.
	Border Swap (4.7.3)	An initial sequence is produced from GC1LA with zero late items then the 'border pair' with the highest combined deviation is swapped so that one item is sequenced late and one item early. The GC1 method is performed again to reshuffle the items within due date periods. This process is repeated until there is no further decrease in part deviation.
Level Part Usage With $\leq$ Specified Number (X) of Items Late	GC1LookAhead-dd / 2 Opt (4.8.1)	The GC1LookAhead-dd method is used to generate a sequence. The 2 optimal procedure without due date restrictions is then used to determine sequences with various numbers of late items.
	Sort-dd / 2-Opt (4.8.2)	The Sort-dd method is used to generate a sequence. The 2 optimal procedure without due date restrictions is then used to determine sequences with various numbers of late items.
	Border Swap (4.8.3)	The Border Swap heuristic as described above is used to generate a series of sequences. The deviations and late items are tracked at each iteration to obtain a sequence with various numbers of late items.

6. Algorithms To Minimize MSD From Linear Part Usage With Zero Late Items

The following algorithms are combinations of Monden's Goal Chasing 1 or GC1-Look-Ahead, and the 2-Optimal heuristic described earlier. The algorithms are designed to allow zero late items in the sequence while at the same time minimizing the mean square deviation of actual part usage from the linear part usage goal.

The existing GC1 algorithm is compared with the newly applied GC1-Look-Ahead (GC1-LA). Since there seems to be evidence that the GC1-LA algorithm gives improved results, it was chosen between the two to be the input to the 2-Optimal (2-Opt) heuristic. This GC1-LA / 2-Opt is then compared with a 2-Optimal heuristic that has a sorted input sequence with respect to component usage.

The algorithms that follow in this section as well as 4.7 and 4.8 include only the steps relative to the specific algorithm. The final sequences attempt to minimize the MSD from linear component usage over all time periods. To obtain the final results that will follow in Chapter 5, the resulting sequence from each algorithm is resequenced within each time period to minimize the MSD from linear component usage for that time period only, given the items determined by the overall goal. This final step is the same for all algorithms and thus will not be stated in the individual explanations.

1. GC1-dd Heuristic Algorithm For Sequencing

Monden's Goal Chasing 1 method as described earlier can be used with due date restrictions allowing zero late items to be sequenced. In this case all of the items are used in the calculations involving summing over all items, but only the items which match the desired due

date are allowed to be sequenced so that there are no late items. This algorithm was described earlier in the chapter. This particular version of GC1 will be referred to as Algorithm (4.6.1)

2. GC1-Look-Ahead-dd Heuristic Algorithm For Sequencing

The Goal Chasing 1 Look Ahead method as described earlier can be used with due date restrictions allowing zero late items to be sequenced. In this case all of the items are used in the calculations involving summing over all items, but only the items which match the desired due date are allowed to be sequenced so that there are no late items. This algorithm was described earlier in the chapter. This particular version of GC1-LA will be referred to as Algorithm (4.6.2).

3. GC1-Look-Ahead-dd / 2-Optimal-dd Heuristic Algorithm For Sequencing

The first part of the heuristic's title states the initial sequence on which the 2-Optimal procedure is performed. For example, in this case, the Goal Chasing 1 Look Ahead heuristic produces the initial sequence. The letters dd at the end of the title indicate that all due dates are observed, and no due date restrictions are broken.

Algorithm (4.6.3) - GC1-Look-Ahead-dd / 2-Optimal-dd Heuristic

Step 1: Sort items in order by due date and model within due date.

Step 2: Produce a sequence based on Goal Chasing 1 Look Ahead Due Date (GC1-LA-dd) method such that all of the items are used in the calculations, but only the items which match the desired due date are allowed to be sequenced so that there are no late items.

Step 3: Starting with this sequence, temporarily swap every combination of 2 items within each

due date period and determine the combination that produces the largest decrease in the mean square deviation of actual part usage from linear part usage.

Step 4: If there is a decrease in the MSD, swap the two items determined in step 3 thus producing a new sequence and go to step 3. If not, stop.

4. Sort-dd / 2-Optimal-dd Heuristic Algorithm For Sequencing

This algorithm is the same as the GC1-LA-dd / 2-Optimal-dd Heuristic, with the exception of the initial sequence. For this algorithm, the initial sequence is produced by simply sorting the items by due date. The steps in the algorithm are identical to the steps above with the elimination of step 2. This algorithm will be referred to as Algorithm (4.6.4).

7. Algorithms To Minimize MSD From Linear Part Usage With No Restriction On The Number Of Late Items

The following algorithms are designed to minimize the mean square deviation of actual part usage from the linear part usage goal. These algorithms allow for the sequencing of late items, but the number of late items are inherently kept to a minimum.

1. GC1 Heuristic Algorithm

Monden's Goal Chasing 1 (GC1) with no due date restrictions produces the first algorithm of this type. The algorithm was described in detail earlier in the chapter. This version with no due date restrictions will be referred to as Algorithm (4.7.1).

2. GC1-Look-Ahead Heuristic Algorithm

The Goal Chasing 1 Look Ahead (GC1-LA) with no due date restrictions was described in detail earlier in the chapter. This version with no due date restrictions will be referred to as Algorithm (4.7.2).

3. Border Swap Heuristic Algorithm

The Border Swap heuristic algorithm is described earlier in the chapter. It will be referred to as Algorithm (4.7.3).

8. Algorithms To Produce Sequences Minimizing MSD From Linear Part Usage With A Specified Number (X) Of Late Items

The following algorithms attempt to minimize the mean square deviation of actual part usage from the linear part usage goal with X late items. These algorithms are all two optimal algorithms. The difference in the various methods is the initial sequence on which the two optimal algorithm is performed.

1. GC1-Look-Ahead-dd / 2-Opt Heuristic Algorithm For Sequencing

In this algorithm, an initial sequence is developed with zero late items by using the GC1-LA method with due dates, then the 2-Optimal method with due dates. The 2-Optimal procedure without due date restrictions is then used to exchange items until there are no more than X late items and there is no improvement in the deviation from the swap of any two items that meet the due date restrictions. This will be referred to as Algorithm (4.8.1).

Algorithm (4.8.1) - GC1-LA-dd / 2-Opt

Step 1: Sort items in order by due date and model within due date.

Step 2: Produce a sequence based on GC1-LA-dd. This is a sequence with no late items.

Step 3: Starting with this sequence, temporarily swap every combination of 2 items that meet the due date restrictions and determine the combination that produces the largest decrease in the mean square deviation of actual part usage from linear part usage.

Step 4: If there is a decrease in the MSD, swap the two items determined in step 3 thus producing a new sequence. This is a sequence with no more than 1 additional late item. Go to step 3. If there is not a decrease in the MSD, stop; the existing deviation is the best for this algorithm with X items late.

2. Sort-dd / 2-Opt Heuristic Algorithm For Sequencing

This algorithm is the same as the GC1-LA-dd / 2-Opt Heuristic, with the exception of the initial sequence. For this algorithm, the initial sequence is produced by simply sorting the items in order by due date first, and model within due date second, then using the 2-Opt procedure to produce the initial sequence. The steps in the algorithm are identical to the steps above with the exception in step 2 as described. This will be referred to as Algorithm (4.8.2).

3. Border Swap Heuristic Algorithm For Sequencing

The Border Swap heuristic as described earlier is used to generate a series of sequences. The deviations and late items are tracked to obtain a tradeoff curve. This is referred to as Algorithm (4.8.3).

CHAPTER 5

Implementation and Results.

In Chapter 5 we describe the method for evaluating the performance of the sequencing algorithms in section 1. First the small test problems are presented in section 2, then their results are shown in section 3. The larger test problems are presented with their respective results in sections 4 and 5. Chapter 6 provides analysis and insights into these results.

1. Evaluating The Sequencing Methods

To evaluate the mean square deviation from uniform component usage we have devised two separate methods. These methods are used due to the complexity of the problem and the large amount of time required to obtain an optimal solution - the mixed-model sequencing problem with multiple objectives is NP-complete. These methods apply when different components are required for the various models, and when all parts are considered of equal importance in mixed-model sequencing. The first method of evaluation is to compare the heuristic MSD solution with an optimal branch-and-bound integer programming solution. This is the best method of comparison for evaluating solutions, but it is not practical for large problems due to the large amount of computer time required to obtain the optimal solution. Therefore, we will evaluate small problems by comparing them to the branch-and-bound optimal solution, but use a different method for evaluating larger problem solutions. The evaluation of the small problems will give us information about the accuracy of each heuristic algorithm for sequencing.

After the performance of the heuristic algorithms on small problems has been evaluated,

we will evaluate the solutions of larger problems. This will be done by comparing the solutions from the various algorithms with the 'best' solution provided by the same algorithms.

For both small and large test problems, the method for evaluation will be the same. The problems will first be evaluated based on the goal of minimizing the Mean Square Deviation from level component usage over all periods of production. This MSD, as defined earlier, is in the form of a cumulative deviation at each stage of production in a sequence. The sequences produced from the various heuristics based on this overall goal are then readjusted within each period to better distribute the components in accordance with the associated single period goal. This method is in essence a 2-phase approach to solving the problem.

In phase one, the initial sequence is used to determine which items if any will be produced in a time period other than the one corresponding to it's original due date. This initial sequence may have some items scheduled late in order to level the overall component usage in accordance with the overall period goal.

The second goal is then considered. An adjusted sequence is created from the original sequence. Each time period is studied independently and the products within the time period are re-sequenced with the goal of leveling the component usage within that time period only. This provides a better sequence on a day-to-day basis for reasons we have seen in previous sections, such as smooth part consumption within a time period.

This new sequence is then evaluated with respect to the goal of leveling the component usage over all periods. This is because of the need for level part usage not only within a sequence period, but also between sequencing periods for reasons discussed earlier such as predictable vendor deliveries for example. The final result is a sequence with a specified number

of items allowed late which strives to meet the objectives of having level component usage within a period, while at the same time minimizing the deviation from level component usage between periods. These two goals could be competing when the demand for a product is not consistent across several time periods of production.

2. The Small Test Problem

To evaluate the algorithms using small test problems, a problem set with 5 models, 10 components, 20 total items to be produced, and 4 production time periods with 5 items produced per time period is used. A random product structure is designed to show the components that are required by each model. For example, Table 8 shows that the assembly of one model 2 requires one component 1, one component 6, one component 8, and one component 10. This structure was designed based on a zero-one option where a model either requires one of a given component or it requires none at all. The distribution of the sample is a uniform $[0,1]$ where there is a 50% chance of the model requiring the particular part. Sample data is shown in Appendix C.

If this structure with only 10 components seems too small to be of any practical use, each component can be thought of as a group of parts or features required in production. This zero-one structure with groups of parts can be thought to represent for example, the situation in the auto industry where a single car has either an automatic transmission or a manual transmission, either a sun roof, or a plain roof, either an air conditioner, or no air conditioner, etc.

This product structure is similar to that defined by Sumichrast, Russell, and Taylor (1992), and by Sumichrast and Russell (1990), with the exception of the zero-one component structure, and the fact that this research incorporates a division of work periods. In these two

Table 8: Model Composition by Component.

Model	COMPONENT										Total Parts
	1	2	3	4	5	6	7	8	9	10	
1	1	0	0	1	0	0	1	0	1	1	5
2	1	0	0	0	0	1	0	1	0	1	4
3	1	1	0	1	0	1	0	1	1	1	7
4	1	0	1	1	1	0	1	0	0	0	5
5	0	1	0	0	1	1	1	0	0	0	4

papers, the researchers set the component structure and varied the demand patterns associated with the component structure.

Ten different demand patterns are used to test each algorithm. The demand patterns are generated randomly from a uniform distribution over the $[0,8]$ range. With five models and a demand mean of 4, the expected total demand for the patterns is 20 items. For our problem it is important to compare each algorithm based on the same total demand. This is due to the importance of being able to directly compare the number of late items in a final sequence between the various solutions.

To adjust the demand patterns so that the total demand for each pattern is exactly 20 items, some additional rules for establishing the demand patterns must be administered. If the demand total for a pattern is less than 20 items then two models are chosen at random to make up the difference. The difference is divided evenly between the two. In the case of an odd difference, the random model chosen first will have added to it the largest portion of the difference. For example, after the initial random demand patterns are established in Table 9, demand pattern C has a total demand of only 17. To bring this total demand up to 20, models 4 and 1 are randomly chosen. Two additional items are added to model 4 changing the demand from 3 to 5, and 1 additional item is added to model 1 changing the demand from 0 to 1. The

results of this update are shown in Table 10.

If the total demand for a demand pattern is greater than 20 items, then a similar procedure is initiated. For example, demand pattern A has a total demand of 22 items. Models 2 and 3 are randomly chosen to reduce the 'extra' demand. The model 2 demand is reduced from 5 to 4, and the model 3 demand is reduced from 6 to 5. In the case where the model demand minus half of the excess is less than zero, a third or additional random model is chosen from which to reduce the demand until the total demand for the pattern is equal to 20.

The due dates for the various demand patterns are randomly chosen from a set of twenty due dates. The set is designed to allow for production of five items within each of four due date periods as follows: {1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4}. Table 11 shows the due dates for each demand pattern.

Table 9: Initial Demand Patterns A - J.
Random Over Uniform [0,8] Distribution

<i>Demand Pattern</i>	<i>Model</i>					<i>Demand Total</i>
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	
A	1	4	5	6	4	20
B	1	0	4	3	2	10
C	0	4	2	3	8	17
D	3	1	1	1	3	9
E	2	2	1	5	5	15
F	1	6	8	1	6	22
G	5	2	2	7	1	17
H	5	5	0	1	2	13
I	6	8	3	7	3	27
J	5	2	6	7	0	20

Table 10: Final Demand Patterns A - J.
Random Over Uniform [0,8] Distribution Plus Alterations To Make Total Demand For Each Demand Pattern Equal 20

Demand Pattern	Model					Demand Total
	1	2	3	4	5	
A	1	4	5	6	4	20
B	1	0	9	3	7	20
C	1	4	2	5	8	20
D	9	1	6	1	3	20
E	4	2	4	5	5	20
F	1	6	7	1	5	20
G	5	2	3	9	1	20
H	8	5	0	1	6	20
I	6	5	0	7	2	20
J	5	2	6	7	0	20

Table 11: Due Dates in Order by Model for Each Demand Pattern A - J.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	2	1	1	2	2	1	2	3	3	4	1	2	3	3	4	4	1	3	4	4
B	3	4	4	2	1	3	4	3	2	1	4	2	1	2	3	1	1	2	3	4
C	2	3	1	3	1	3	4	2	1	2	2	1	4	4	3	4	2	1	3	4
D	2	1	3	4	2	1	2	4	3	4	2	3	1	1	3	1	2	4	4	3
E	3	1	2	4	1	2	2	1	3	3	3	4	4	4	1	2	4	2	3	1
F	1	2	2	1	3	3	4	1	3	3	4	1	4	4	3	2	2	4	1	2
G	2	4	3	1	3	3	3	1	4	1	3	2	2	2	4	4	1	2	4	1
H	1	1	3	4	2	2	3	4	2	1	3	2	1	3	4	4	3	1	2	4
I	4	3	4	2	3	2	1	4	1	1	4	3	2	2	1	1	3	3	2	4
J	4	4	2	4	3	1	1	3	4	4	2	2	3	2	1	3	3	1	2	1

There are three different goals to be analyzed and various heuristic algorithms designed to accomplish each goal. Table 12 is a review of the goals, and the algorithms that will be tested for each goal.

3. Results Of The Small Test Problem

The small problem set was used to test all of the algorithms. The results of these tests are broken down by objective. Shown first are the results of the algorithms designed to level part usage while at the same time allowing no late items in the sequence. Next, are the results of the algorithms designed to level part usage with no restrictions on the number of late items. Lastly, the results are shown for the algorithms designed to level part usage with no more than X late items.

Table 12: Algorithms to be Tested and Analyzed for Each Objective.

Objective	Algorithm	Description
Goal 1: Level Part Usage (Zero Items Late)	(4.6.1) GC1-dd	Mondon's method with 0 late.
	(4.6.2) GC1LookAhead-dd	Improvement to Monden's with 0 late.
	(4.6.3) GC1LookAhead-dd/2opt-dd	2-Opt. starting with GC1LA solution.
	(4.6.4) Sort-dd/2opt-dd	2-Opt. starting with a sorted solution.
	Branch-And-Bound	Optimal solution with 0 late.
Goal 2: Level Part Usage (No Due Date Restrictions)	(4.7.1) GC1	Mondon's method.
	(4.7.2) GC1LookAhead	Improvement to Monden's.
	(4.7.3) Border Swap	Swap products at period borders.
	Branch-And-Bound	Optimal solution with any late.
Goal 3: Level Part Usage With $\leq$ Specified Number (X) of Items Late	(4.8.1) GC1LookAhead-dd/2opt	2-Opt. starting with GC1LA solution.
	(4.8.2) Sort-dd/2opt	2-Opt. starting with a sorted solution.
	(4.8.3) Border Swap	Swap products at period borders.
	Branch-And-Bound	solution based on optimal with $\leq X$ late.

1. Goal 1: Level Part Usage with Zero Items Late and Problem Size = 20 Items

The Branch-and-Bound method provides an optimal solution that is used as a basis for comparing the performance of the other sequencing methods. All of the sequencing methods performed very well as is indicated by the actual Mean Square Deviation data given in Table 13, and the small average percent from optimal for each as shown in Table 14. For the small test problems, the GC1LookAhead-dd/2-Optimal-dd method requiring on average 0.29 seconds as shown in Table 15, along with the Sort-dd/2-Optimal-dd method requiring on average 0.50 seconds, provided the solutions that were closest to optimal in terms of the average deviation from uniform component usage based on an overall goal (not a single period goal). The solutions for both methods were 0.09 percent from the optimal solution. The GC1LookAhead-dd method requiring on average 0.20 seconds provided the next best solutions being on average 0.16 percent from optimal. Monden's original GC1 method requiring on average 0.22 seconds fell in last place among the four being on average 0.36 percent from optimal. All of these results are less than 1 percent from optimal.

After the sequences have been established based on the overall leveling goal, the products designated within each time period are then re-sequenced based on that period's single period goal according to the GC1LookAhead-dd algorithm. The newly adjusted sequences are then evaluated according to the same criteria: the overall deviation from uniform part usage. The results shown in Table 16 indicate no differences between any of the sequencing methods. This indicates that for this set of small test problems, any sequencing method can be used as part of the 2-phase leveling algorithm to obtain the same results after the second phase of adjustment.

Table 13: Total Mean Square Deviation from Overall Uniform Component Usage with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
GC1-dd	0.5	0.138	1.063	0.544	0.472	0.284	0.276	0.369	0.417	0.779
GC1LookAhead-dd	0.497	0.138	1.064	0.544	0.470	0.281	0.276	0.369	0.417	0.779
GC1LA-dd / 2opt-dd	0.497	0.138	1.063	0.544	0.467	0.281	0.276	0.369	0.417	0.779
Sort-dd /2opt-dd	0.497	0.138	1.063	0.544	0.467	0.281	0.276	0.369	0.417	0.779
Branch-and-Bound	0.497	0.138	1.063	0.544	0.463	0.281	0.276	0.369	0.417	0.779

Table 14: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Optimal Component Usage for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Average MSD From Level Part Usage</i>	<i>Average % From Optimal</i>
GC1-dd	0.0015	0.36%
GC1LookAhead-dd	0.0008	0.16%
GC1LA-dd / 2opt-dd	0.0004	0.09%
Sort-dd /2opt-dd	0.0004	0.09%

Table 15: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items (Elapsed processing time based on an IBM-Compatible Pentium Processor, 166 Mhz, 12 Mb RAM)

<i>Sequencing Method</i>	<i>Demand Pattern</i>										<i>Avg. Time</i>
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	
GC1-dd	0.22	0.22	0.22	0.22	0.16	0.22	0.22	0.22	0.22	0.27	0.22
GC1LookAhead-dd	0.11	0.22	0.11	0.22	0.11	0.33	0.11	0.22	0.33	0.27	0.20
GC1LA-dd / 2opt-dd	0.22	0.49	0.22	0.22	0.71	0.22	0.22	0.22	0.33	0.27	0.29
Sort-dd /2opt-dd	1.04	0.49	0.22	0.22	0.71	0.61	0.22	0.77	0.44	0.27	0.50
Branch-and-Bound	3.19	0.16	1.81	0.39	22.85	0.44	0.16	0.22	0.33	0.27	2.98

Table 16: Total Mean Square Deviation from Uniform Component Usage Over All Periods for Sequences Adjusted and Leveled Within each Period with Zero Late Items.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
GC1-dd	.836	.282	1.317	.887	.665	.428	.330	.622	.539	.879
GC1LookAhead-dd	.836	.282	1.317	.887	.665	.428	.330	.622	.539	.879
GC1LA-dd / 2opt-dd	.836	.282	1.317	.887	.665	.428	.330	.622	.539	.879
Sort-dd /2opt-dd	.836	.282	1.317	.887	.665	.428	.330	.622	.539	.879
Branch-and-Bound	.836	.282	1.317	.887	.665	.428	.330	.622	.539	.879

Note that the adjusted Branch-and-Bound sequence does not provide an optimal solution, but rather an adjustment based on an initial solution that was optimal for the overall leveling goal. This is due to the fact that the sequence obtained from the Branch-and-Bound is adjusted with a heuristic method - the GC1LookAhead-dd algorithm.

2. Goal 2: Level Part Usage with No Due Date Restrictions and Problem Size = 20 Items

As in the previous section, the Branch-and-Bound method provides an optimal solution that is used as a basis for comparing the performance of the other sequencing methods. There was a little more distinction between the sequencing methods for the problems tested when there are no due date restrictions, as shown in Tables 17 and 18. For the small test problems, the GC1LookAhead method requiring on average 0.241 seconds as shown in Table 19, provided the solutions that were closest to optimal in terms of the average deviation from uniform component usage based on an overall goal (not a single period goal). The GC1LA method was on average 0.9 percent from the optimal solution for the ten test problems. The new Border Swap method provided solutions that were on average 6.02 percent from optimal in an average time of 1.84 seconds. Monden's original GC1 method requiring on average 0.07 seconds, the smallest time, fell in last place among the three, being on average 9.9 percent from optimal.

After the sequences have been established based on the overall leveling goal, the products designated within each time period are then re-sequenced based on that period's single period goal according to the GC1LookAhead-dd algorithm. The newly adjusted sequences are then evaluated according to the same criteria: the overall deviation from uniform part usage. Note that the adjusted Branch-and-Bound sequence does not provide an optimal solution, but rather an adjustment based on an initial solution that was optimal for the stated goal. The results

Table 17: Mean Square Deviation from Uniform Component Usage, and Number of Items Late Respectively with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
GC1	0.155 2	0.135 2	0.140 5	0.133 3	0.139 2	0.141 4	0.131 3	0.111 2	0.122 2	0.118 3
GC1LA	0.136 2	0.111 1	0.121 5	0.133 3	0.130 3	0.136 4	0.129 3	0.105 2	0.113 4	0.105 3
Border Swap	0.136 2	0.111 1	0.121 5	0.141 3	0.130 2	0.136 3	0.164 2	0.105 1	0.133 3	0.105 3
Branch-Bound	0.136 2	0.111 1	0.121 5	0.129 3	0.126 2	0.136 3	0.128 2	0.105 1	0.113 3	0.103 3

Table 18: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Optimal Component Usage Deviation for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance

<i>Sequencing Method</i>	<i>Average MSD From Level Part Usage</i>	<i>Average % From Optimal</i>
GC1	0.0117	9.90%
GC1LA	0.0011	0.90%
Border Swap	0.0074	6.02%

Table 19: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage with No Due Date Restrictions. (Elapsed processing time based on an IBM-Compatible Pentium Processor, 166 Mhz, 12 Mb RAM)

<i>Sequencing Method</i>	<i>Demand Pattern</i>										<i>Avg. Time</i>
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	
GC1	0.05	0.22	0.06	0.06	0.05	0.05	0.05	0.05	0.06	0.05	0.07
GC1LookAhd	0.16	0.16	0.33	0.17	0.16	0.33	0.11	0.33	0.33	0.33	0.241
Border Swap	0.32	0.20	1.40	0.51	0.32	0.98	0.25	0.38	1.22	0.87	0.645
Branch-Bound	107.1	1.10	0.22	227.9	0.93	2.36	4.95	0.11	0.44	0.82	34.593

shown in Tables 20 and 21 indicate similar results between the GC1LA algorithm, 10.20 percent from optimal, and the Border Swap algorithm, 11.67 percent from optimal. A difference between the two is that the Border Swap algorithm provides more solutions with fewer late items and the same MSD as does the GC1LA algorithm. This important fact indicates that the Border Swap algorithm might be a better algorithm in this case.

3. Goal 3: Level Part Usage With X Items Late And Problem Size = 20 Items

First, the results of the average performance of the three sequencing heuristics are evaluated, then the individual tradeoff curves for each demand pattern are examined. For the small test problems, the deviation from level component usage was measured for one item late, two items late, etc. until there was no longer a decrease in the deviation. Each demand pattern has its own unique tradeoff curve as will be shown later.

The sequencing heuristic can have an effect on the number of items late to minimize the MSD from level component usage. The average percent from optimal for each sequencing

Table 20: Total Mean Square Deviation from Uniform Component Usage Over All Periods and Number of Items Late Respectively for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
GC1	0.176 2	0.161 2	0.191 5	0.179 3	0.178 2	0.198 4	0.218 3	0.166 2	0.184 2	0.153 3
GC1LookAhd	0.139 2	0.148 1	0.199 5	0.179 3	0.178 3	0.185 4	0.227 3	0.161 2	0.175 4	0.153 3
Border Swap	0.139 2	0.135 1	0.167 5	0.213 3	0.178 2	0.185 3	0.248 2	0.161 1	0.191 4	0.153 3
Branch-Bound	0.139 2	0.135 1	0.199 5	0.173 3	0.178 2	0.185 3	0.188 2	0.161 1	0.126 3	0.139 3

Table 21: Average Adjusted Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance

<i>Sequencing Method</i>	<i>Average MSD From Best Part Usage</i>	<i>Average % From Best</i>
GC1	0.0213	14.59%
GC1LookAhd	0.0153	10.20%
Border Swap	0.0179	11.67%
Branch-Bound	0.0032	1.92%

method with the sequence based on the overall period goals is shown in Table 22 and Figure 8. For each number of late items, the total number of observations from ten possible observations are shown.

The concavity of the average percent from optimal for each sequencing method ranging over the number of late items, can best be explained by the decreasing number of observations associated with each number late. In this case, when there are fewer observations, in some cases only one, the observations tended to be closer to the optimal solution thus giving the effect that the average over all observations is closer to optimal.

For the number of late items with the largest numbers of observations, 0 late, 1 late, and 2 late, the 2-Optimal heuristic starting with the Goal Chasing LookAhead - dd solution provided the best solutions on average. Next was the 2-Optimal heuristic starting with a sorted solution which provided similar, but slightly better results on average than the Border Swap heuristic.

Table 23 and Figure 9 show the results after the sequences have been adjusted and balanced within each due date period. The newly adjusted sequences are then compared based on the same overall goal of minimizing the MSD from level component usage over all periods. The performance of the heuristics changes slightly based on the new sequences. Goal Chasing LookAhead-dd/2-Optimal still has the best results, with the Sort-dd/2-Optimal next followed by the Border Swap heuristic. As was expected, all sequences provided a larger MSD when compared to the unadjusted sequence results.

Tables 24 through 43 and the corresponding Figures 10 through 29 respectively, show the results of the individual tradeoff curves for each small test problem with demand patterns A

Table 22: The Average Percent from Optimal for each Sequencing Method with the Sequence Based on Over All Period Goals for Demand Patterns A-J with 20 Items.

	# Observations Per Method Respectively, and # Late					
	10,10,10 0 late	10,10,10 1 late	8,8,8 2 late	5,5,5 3 late	2,2,1 4 late	0,0,1 5 late
S/2	0%	14%	14%	9%	12%	
G/2	0%	5%	5%	4%	0%	
BS	0%	16%	18%	8%	4%	0%

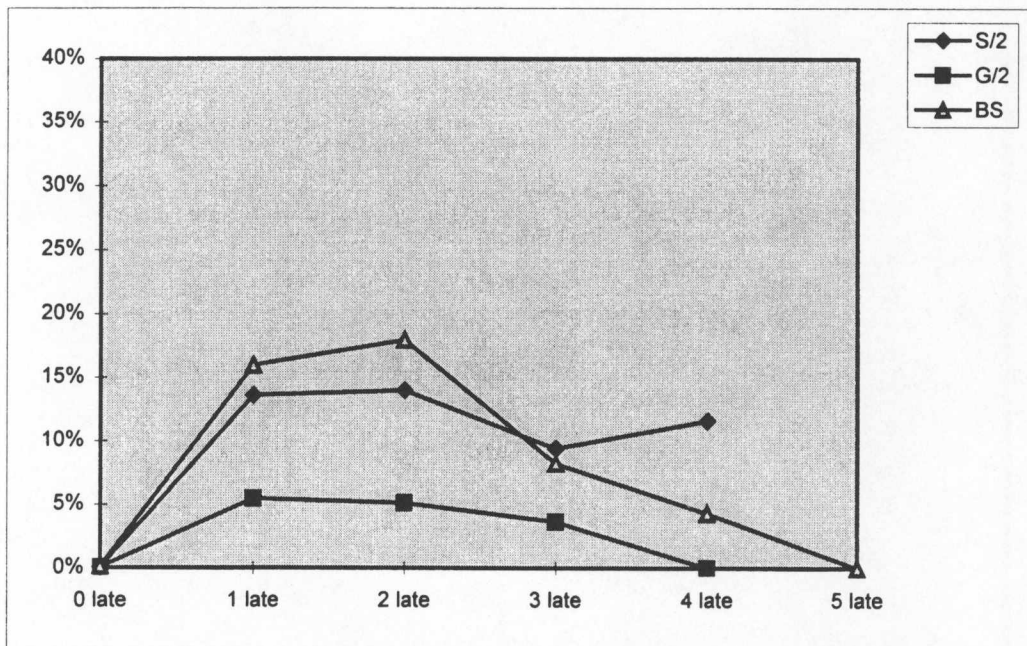


Figure 8: Graph of the Average Percent from the Optimal Mean Square Deviation for Demand Patterns A through J.

through J respectively. The results for all problems and all methods show that allowing one item to be sequenced late will increase performance with respect to reducing the MSD from level component usage. In some cases there is also a large decrease in MSD by allowing two items to be sequenced late. There seems to be less of an advantage in allowing three or more items late.

Table 23: The Average Percent from Best for each Sequencing Method with the Sequence Adjusted Based on Single Period Goals for Demand Patterns A-J with 20 Items.

# Observations Per Method Respectively, and # Late						
	10,10,10	10,10,10	8,8,8	5,5,5	2,2,1	0,0,1
	0 late	1 late	2 late	3 late	4 late	5 late
S/2	16%	22%	24%	11%	0%	
G/2	16%	20%	11%	15%	17%	
BS	16%	34%	36%	31%	20%	0%

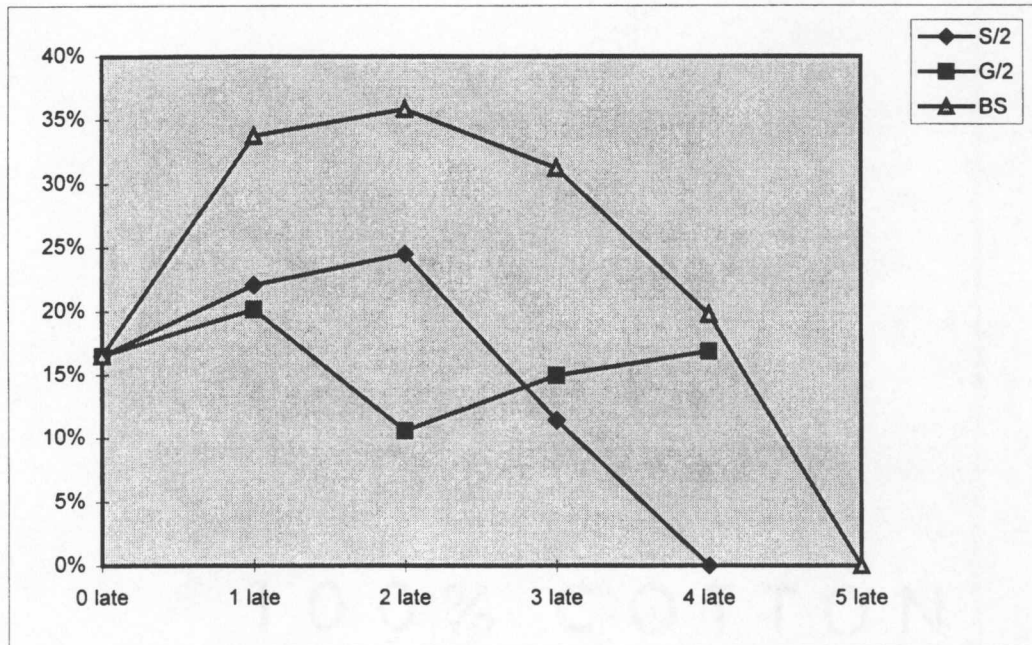


Figure 9: Graph of the Adjusted Average Percent from the Best Mean Square Deviation for Demand Patterns A through J.

Table 24: MSD Demand Pattern A.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.50	0.21	0.17
GC1LookAhead-DD/2-Opt	0.50	0.19	0.16
Border Swap	0.50	0.21	0.14
Branch-and-Bound	0.50	0.19	0.14

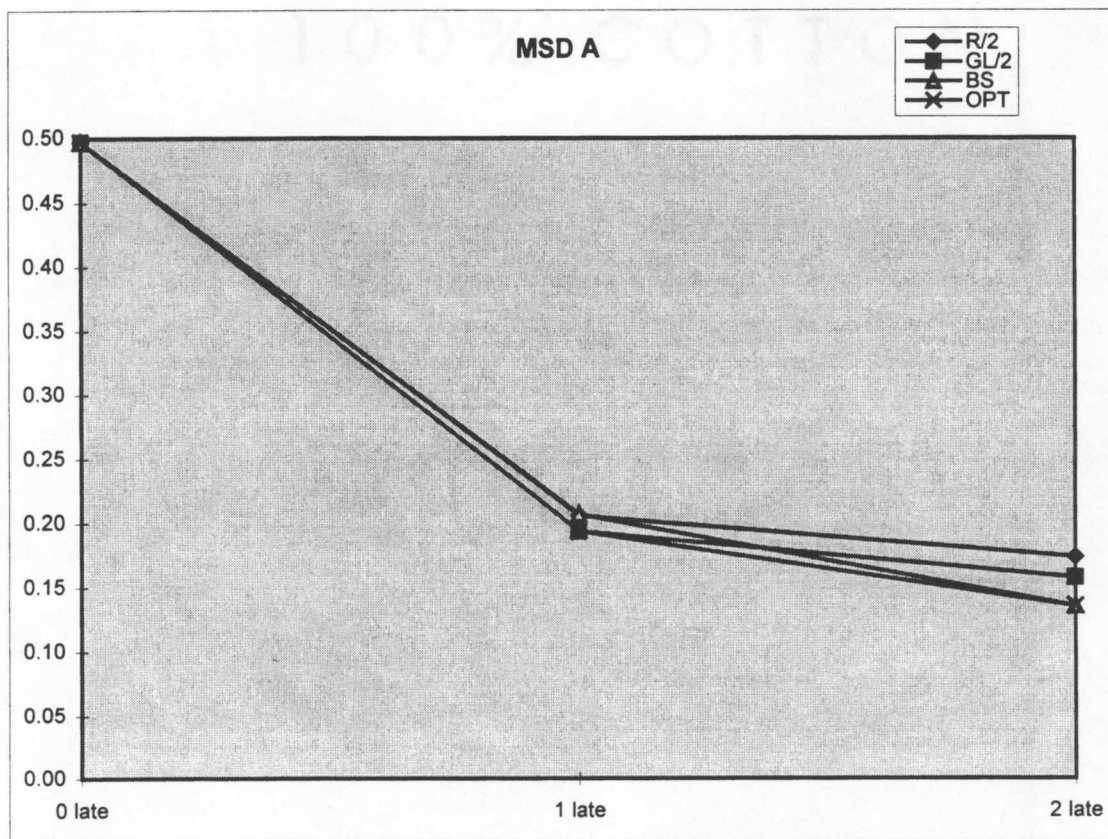


Figure 10: Graph of the Overall Mean Square Deviation for Demand Pattern A.

Table 25: Adjusted MSD Demand Pattern A.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.84	0.30	0.19
GC1LookAhead-DD/2-Opt	0.84	0.32	0.18
Border Swap	0.84	0.32	0.14
Branch-and-Bound	0.84	0.32	0.14

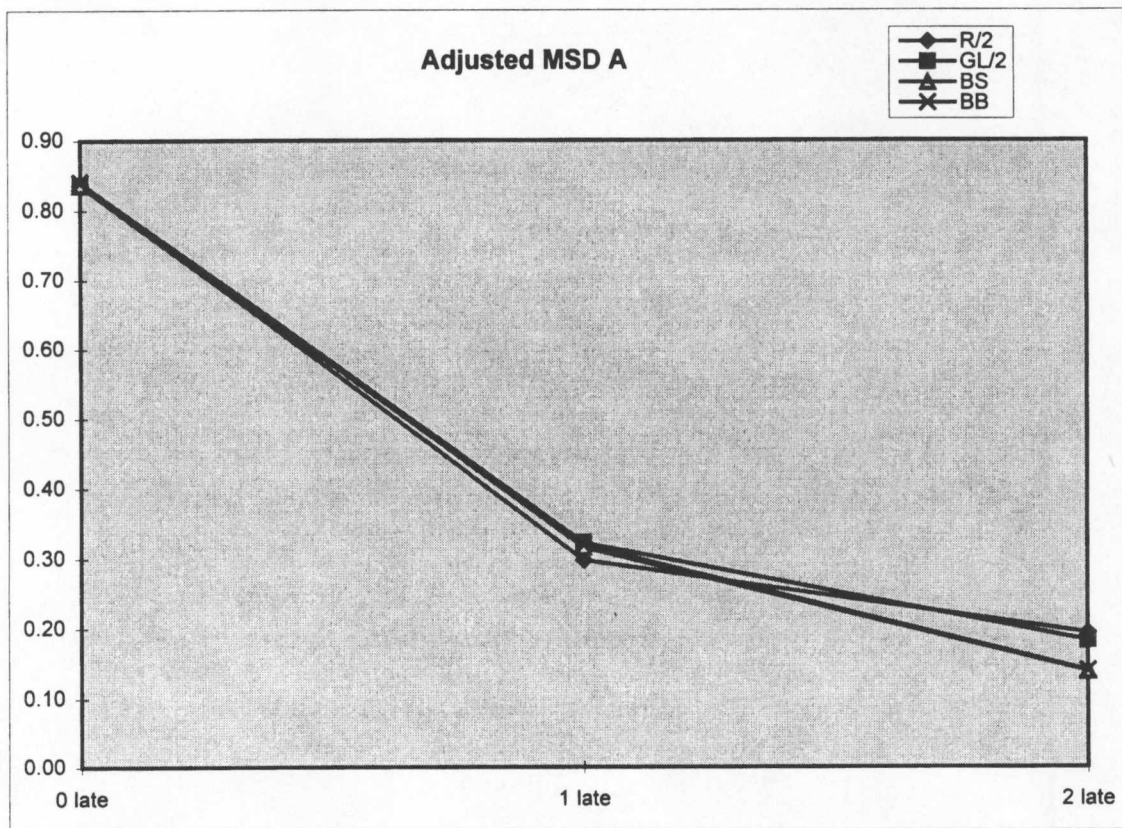


Figure 11: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern A.

Table 26: MSD Demand Pattern B.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late
Random-DD / 2-Opt	0.14	0.12
GC1LookAhead-DD/2-Opt	0.14	0.12
Border Swap	0.14	0.11
Branch-and-Bound	0.14	0.11

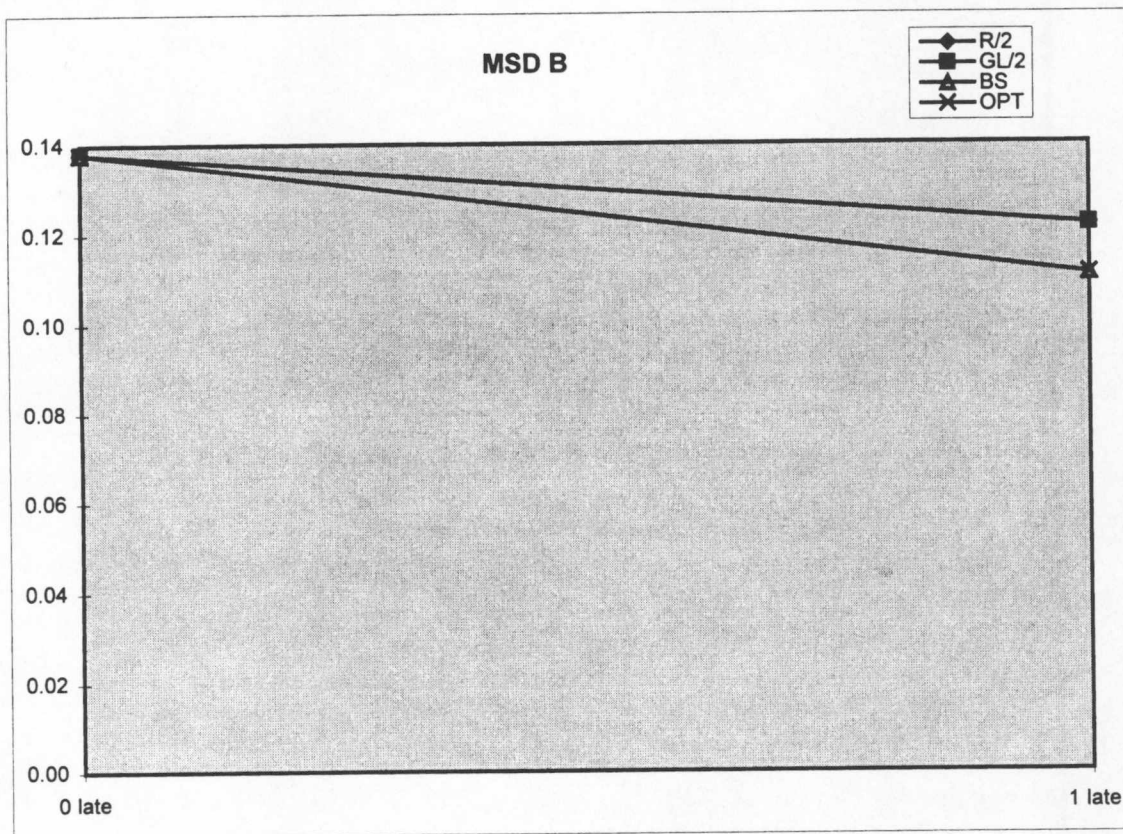


Figure 12: Graph of the Overall Mean Square Deviation for Demand Pattern B.

Table 27: Adjusted MSD Demand Pattern B.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late
Random-DD / 2-Opt	0.28	0.15
GC1LookAhead-DD/2-Opt	0.28	0.15
Border Swap	0.28	0.14
Branch-and-Bound	0.28	0.14

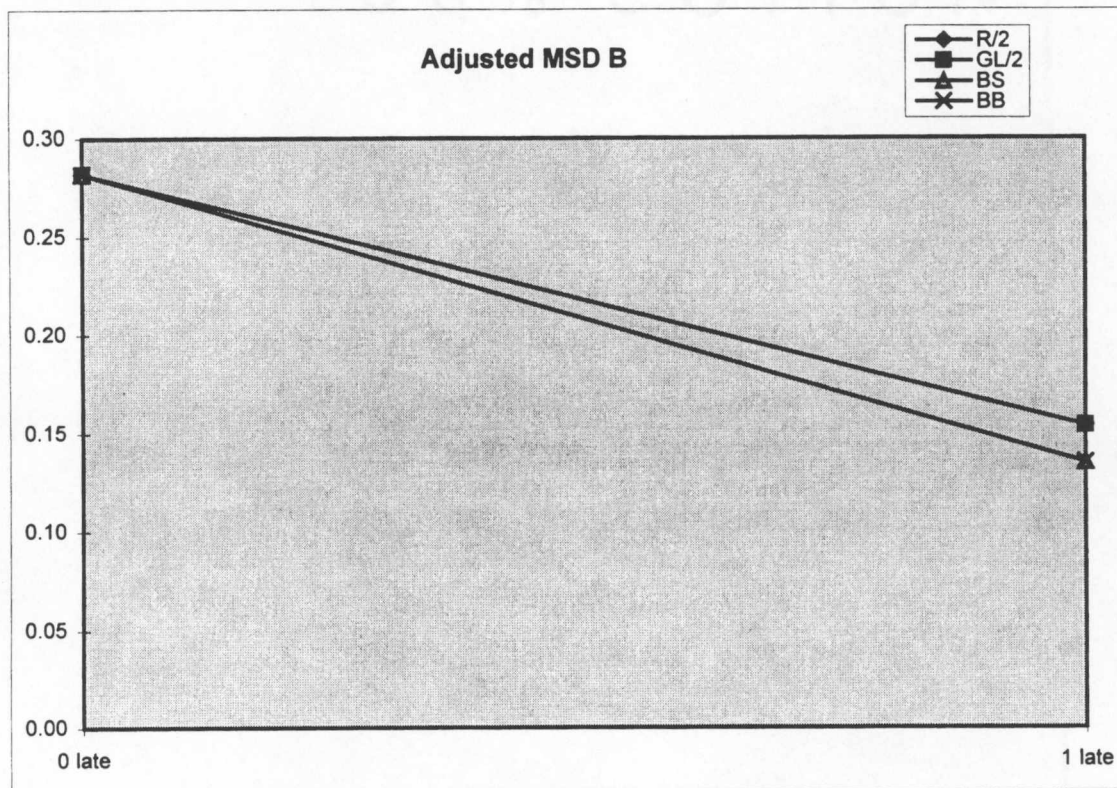


Figure 13: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern B.

Table 28: MSD Demand Pattern C.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	1.06	0.39	0.20	0.18	0.17	
GC1LookAhead-DD/2-Opt	1.06	0.39	0.20	0.17	0.14	
Border Swap	1.06	0.59	0.21	0.17	0.15	0.12
Branch-and-Bound	1.06	0.39	0.20	0.15	0.14	0.12

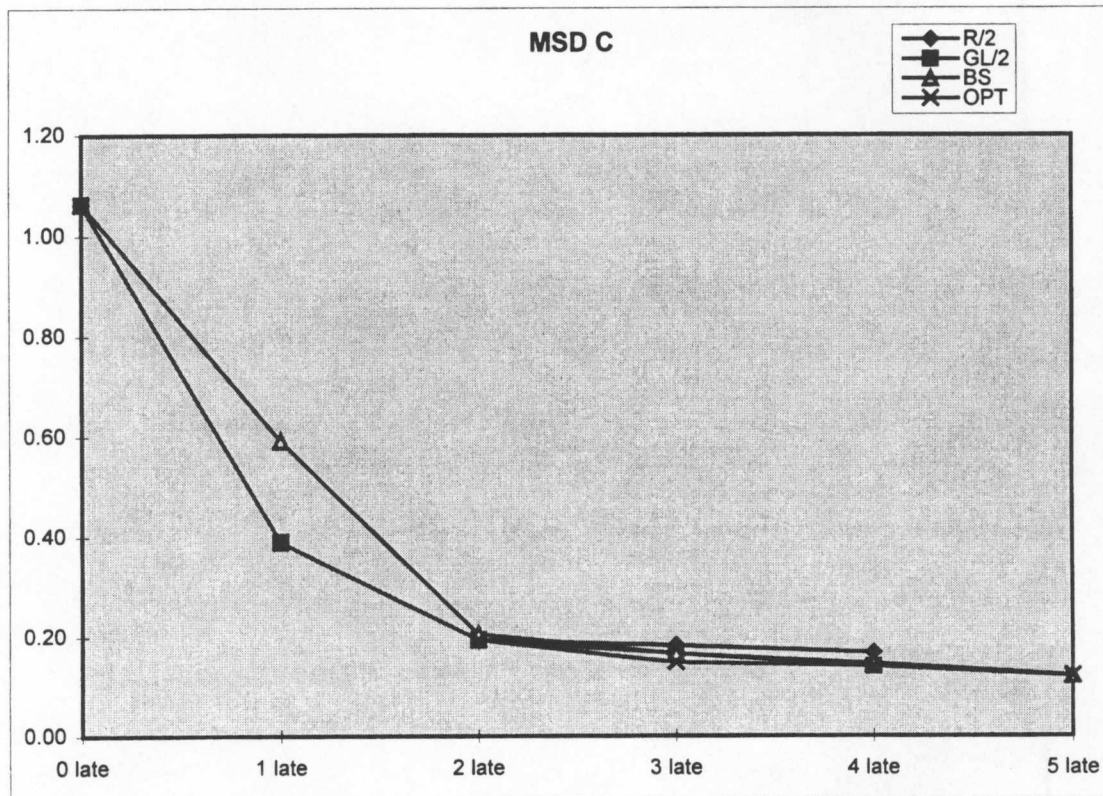


Figure 14: Graph of the Overall Mean Square Deviation for Demand Pattern C.

Table 29: Adjusted MSD Demand Pattern C.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	1.32	0.50	0.25	0.21	0.17	
GC1LookAhead-DD/2-Opt	1.32	0.50	0.25	0.25	0.19	
Border Swap	1.32	0.76	0.36	0.36	0.20	0.17
Branch-and-Bound	0.50	0.25	0.23	0.21	0.20	0.19

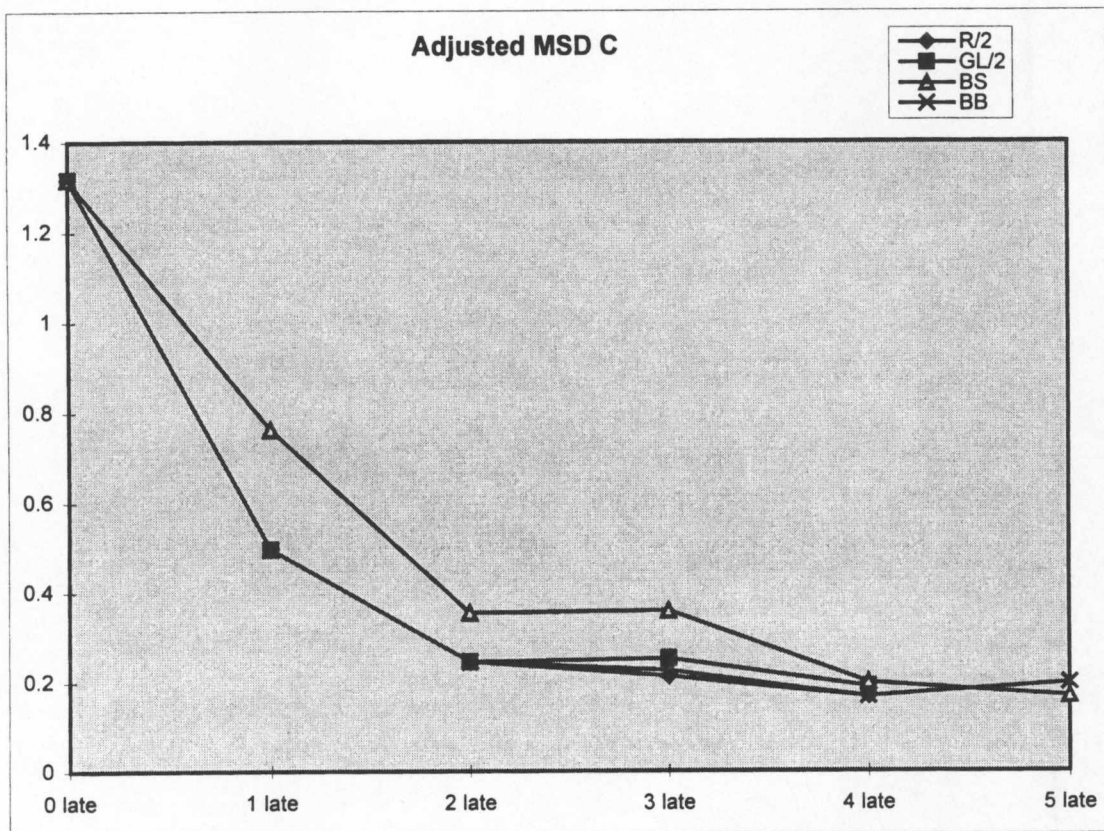


Figure 15: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern C.

Table 30: MSD Demand Pattern D.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.54	0.27	0.20	0.13
GC1LookAhead-DD/2-Opt	0.54	0.15	0.14	0.13
Border Swap	0.54	0.15	0.23	0.14
Branch-and-Bound	0.54	0.15	0.13	0.13

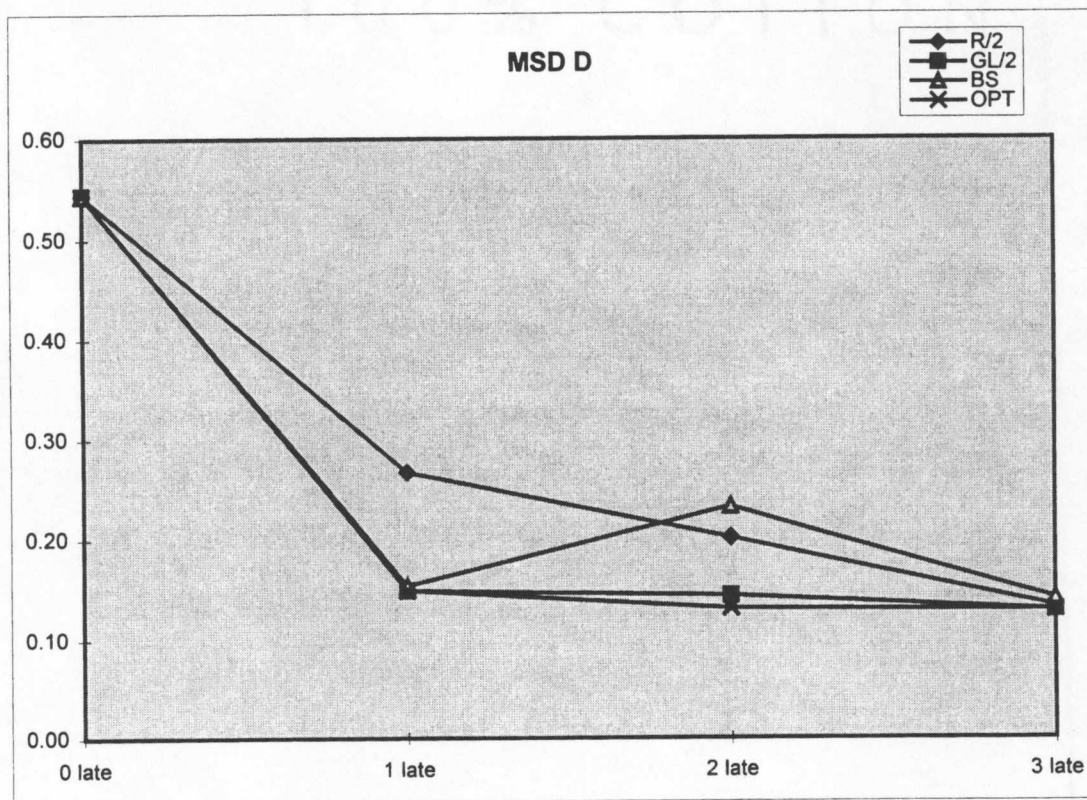


Figure 16: Graph of the Overall Mean Square Deviation for Demand Pattern D.

Table 31: Adjusted MSD Demand Pattern D.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.89	0.40	0.30	0.18
GC1LookAhead-DD/2-Opt	0.89	0.23	0.24	0.18
Border Swap	0.89	0.23	0.37	0.21
Branch-and-Bound	0.89	0.23	0.20	0.17

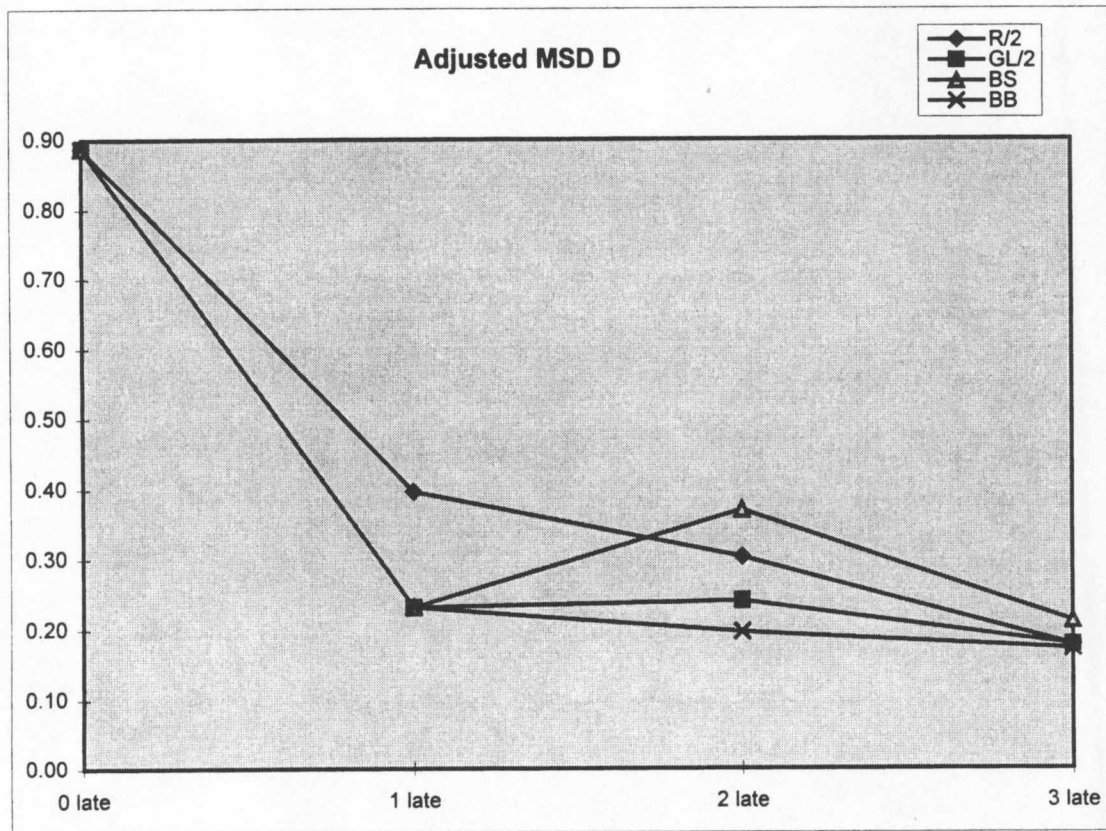


Figure 17: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern D.

Table 32: MSD Demand Pattern E.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.47	0.17	0.13
GC1LookAhead-DD/2-Opt	0.47	0.16	0.14
Border Swap	0.47	0.27	0.13
Branch-and-Bound	0.46	0.15	0.13

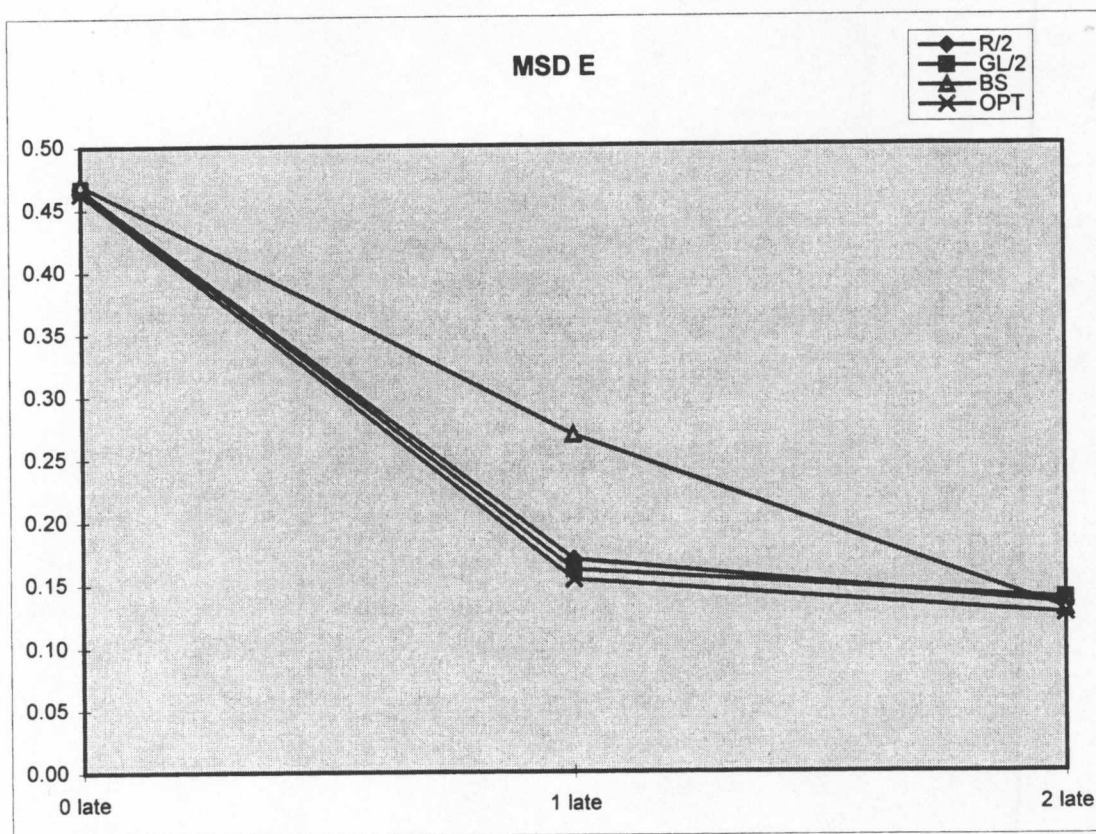


Figure 18: Graph of the Overall Mean Square Deviation for Demand Pattern E.

Table 33: Adjusted MSD Demand Pattern E.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.67	0.21	0.18
GC1LookAhead-DD/2-Opt	0.67	0.21	0.18
Border Swap	0.67	0.37	0.18
Branch-and-Bound	0.67	0.21	0.18

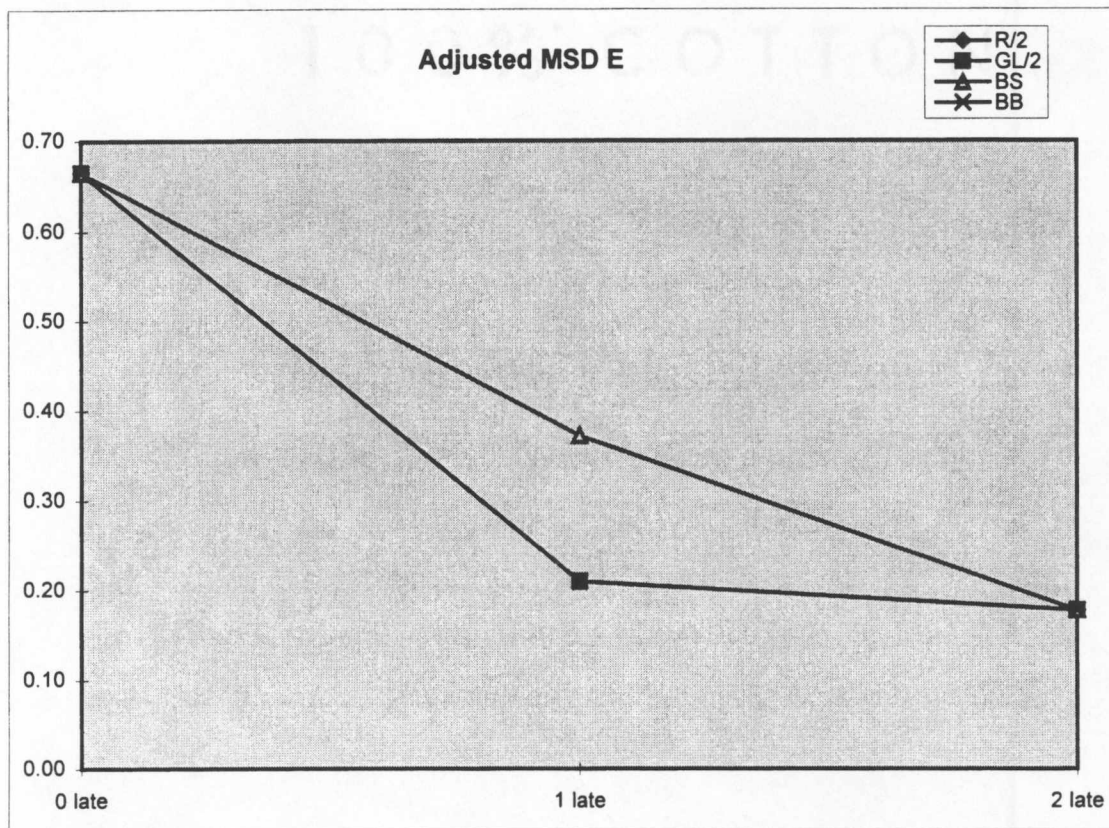


Figure 19: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern E.

Table 34: MSD Demand Pattern F.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.28	0.17	0.14	0.14
GC1LookAhead-DD/2-Opt	0.28	0.17	0.14	0.14
Border Swap	0.28	0.15	0.14	0.14
Branch-and-Bound	0.28	0.15	0.14	0.14

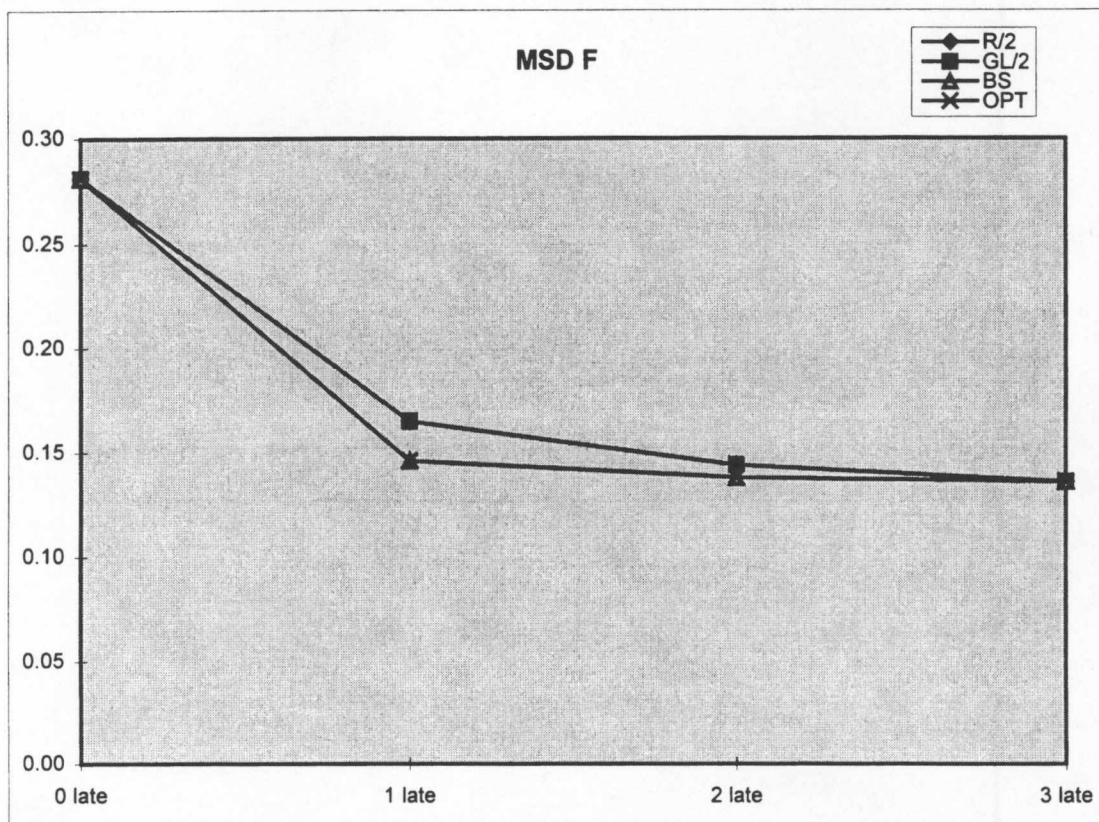


Figure 20: Graph of the Overall Mean Square Deviation for Demand Pattern F.

Table 35: Adjusted MSD Demand Pattern F.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.43	0.25	0.21	0.19
GC1LookAhead-DD/2-Opt	0.43	0.25	0.21	0.19
Border Swap	0.43	0.20	0.18	0.19
Branch-and-Bound	0.43	0.20	0.18	0.19

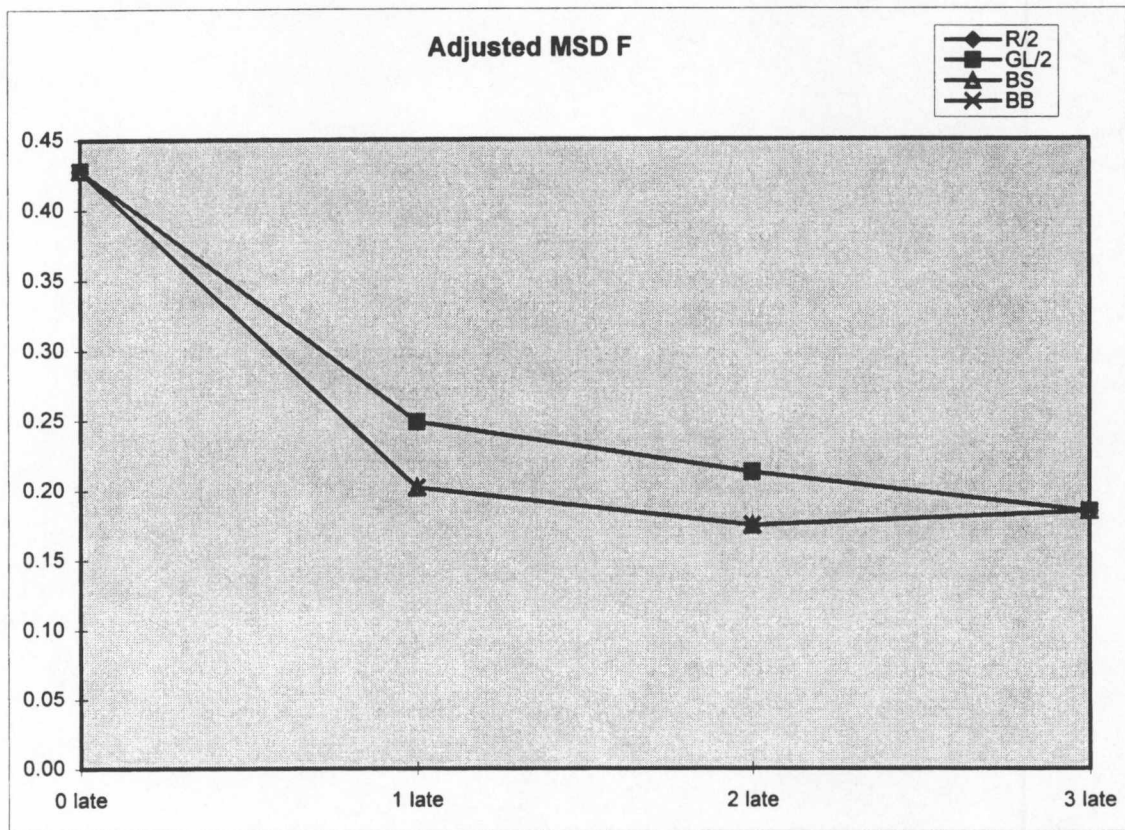


Figure 21: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern F.

Table 36: MSD Demand Pattern G.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.28	0.18	0.13
GC1LookAhead-DD/2-Opt	0.28	0.19	0.13
Border Swap	0.28	0.17	0.16
Branch-and-Bound	0.28	0.17	0.13

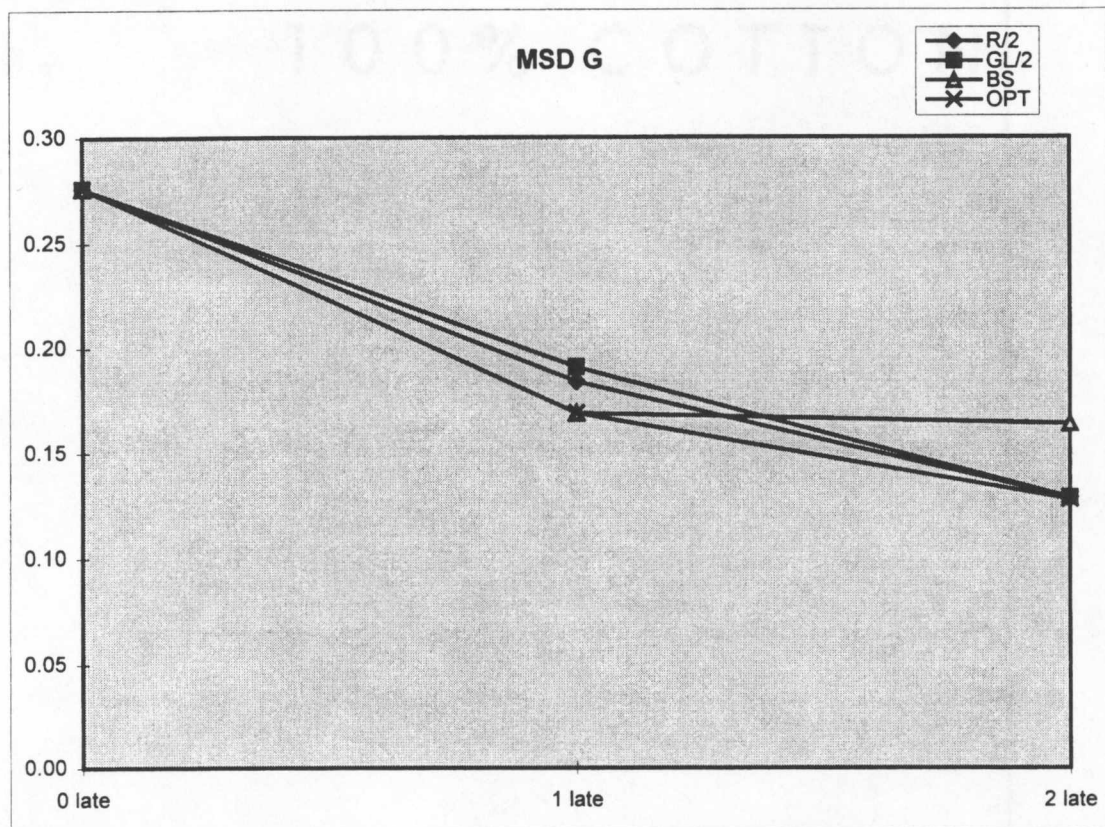


Figure 22: Graph of the Overall Mean Square Deviation for Demand Pattern G.

Table 37: Adjusted MSD Demand Pattern G.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late
Random-DD / 2-Opt	0.33	0.23	0.19
GC1LookAhead-DD/2-Opt	0.33	0.28	0.19
Border Swap	0.33	0.25	0.25
Branch-and-Bound	0.33	0.25	0.19

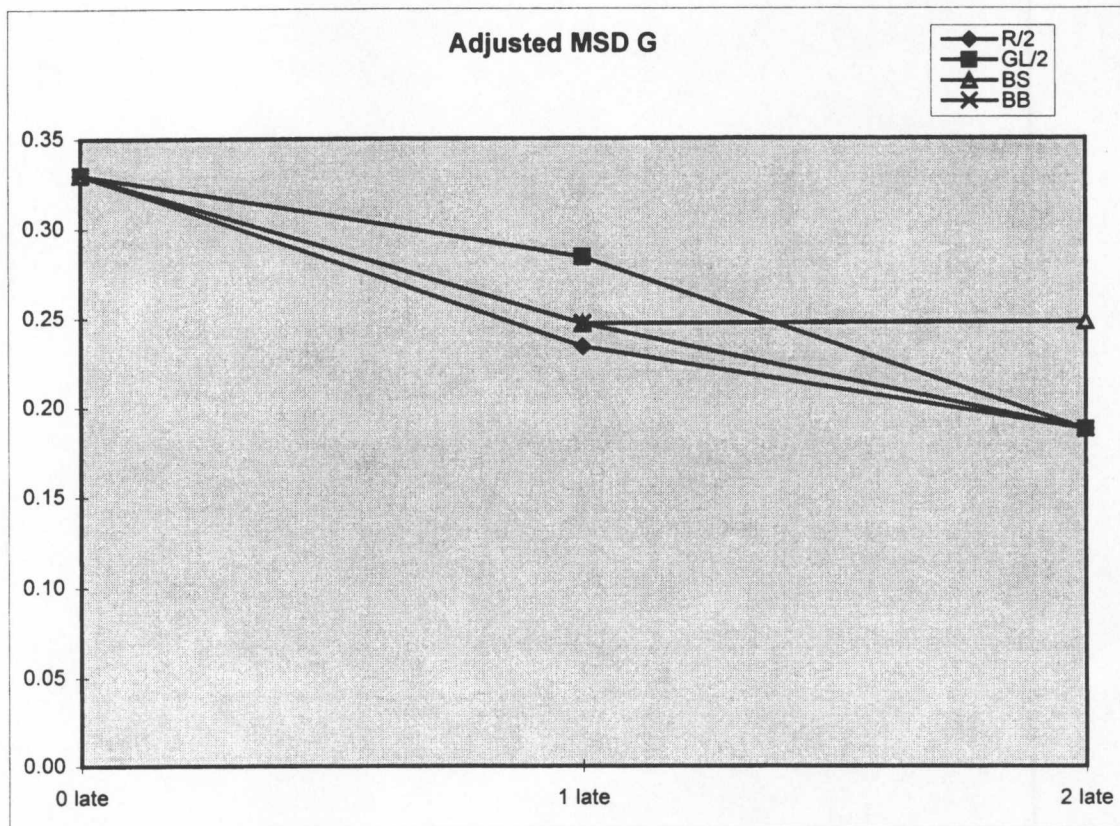


Figure 23: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern G.

Table 38: MSD Demand Pattern H.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late
Random-DD / 2-Opt	0.37	0.11
GC1LookAhead-DD/2-Opt	0.37	0.12
Border Swap	0.37	0.11
Branch-and-Bound	0.37	0.11

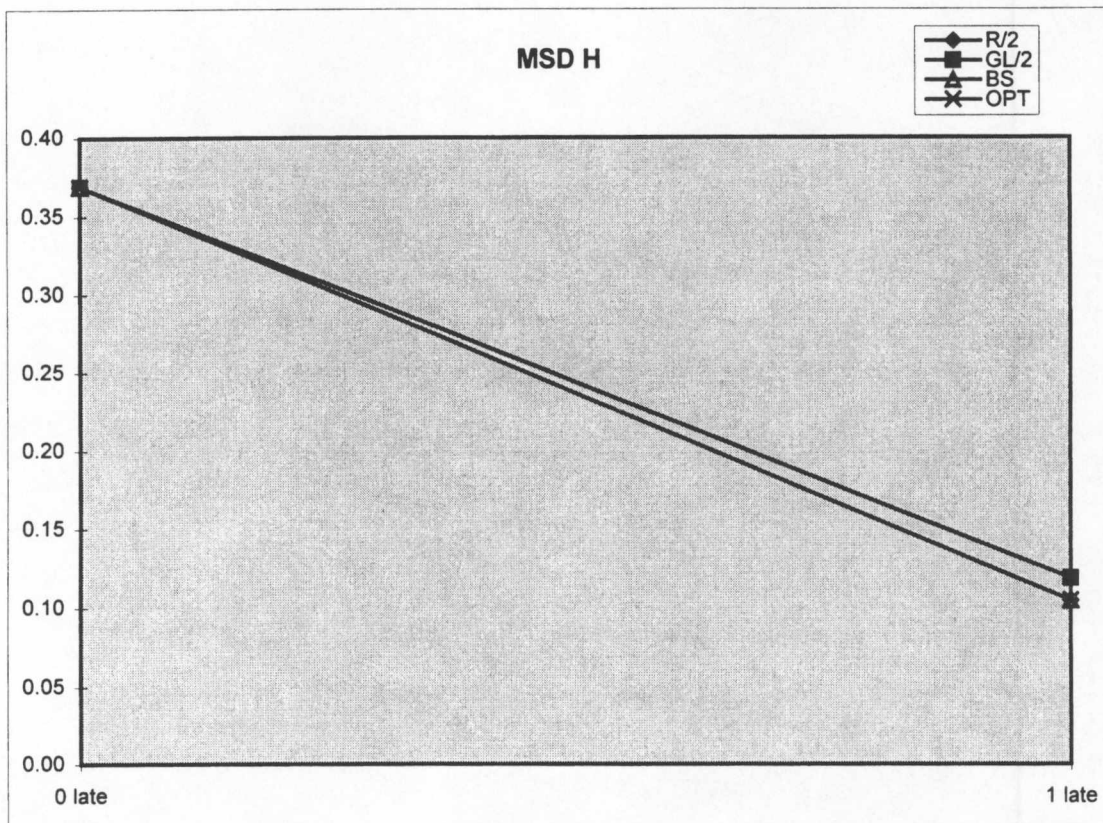


Figure 24: Graph of the Overall Mean Square Deviation for Demand Pattern H.

Table 39: Adjusted MSD Demand Pattern H.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late
Random-DD / 2-Opt	0.62	0.16
GC1LookAhead-DD/2-Opt	0.62	0.21
Border Swap	0.62	0.16
Branch-and-Bound	0.62	0.16

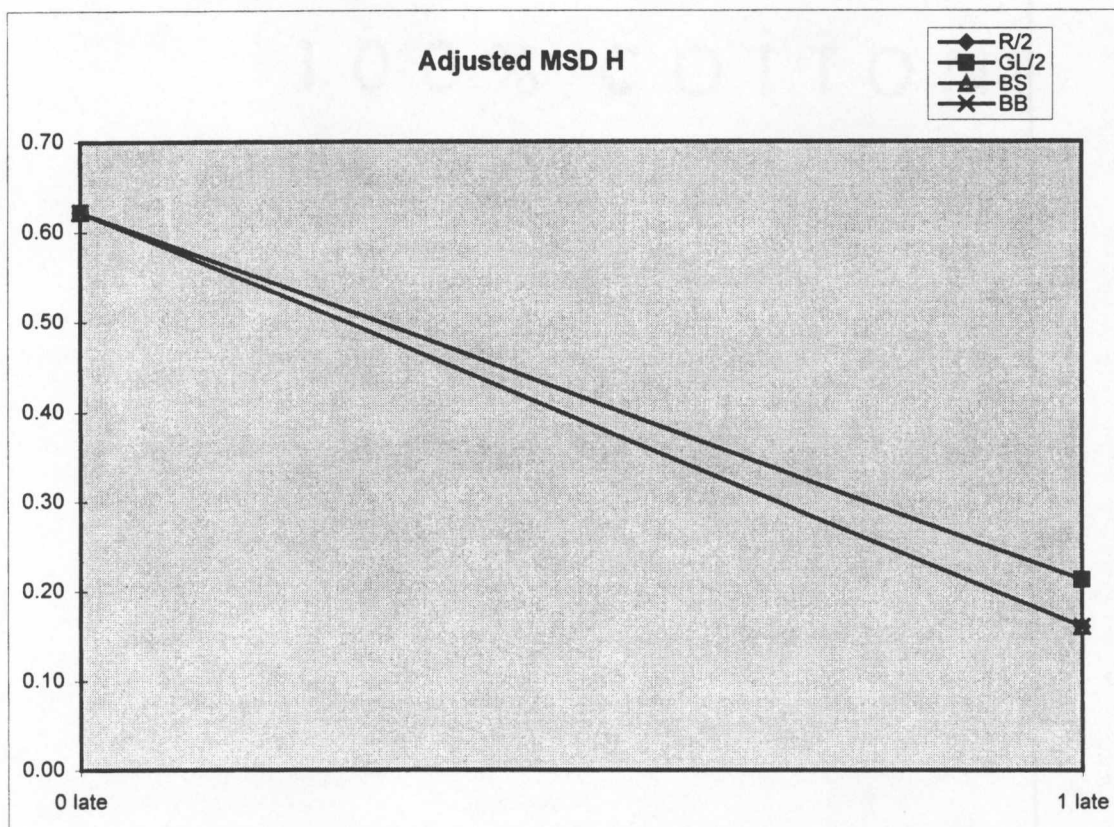


Figure 25: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern H.

Table 40: MSD Demand Pattern I.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late
Random-DD / 2-Opt	0.42	0.25	0.14	0.14	0.12
GC1LookAhead-DD/2-Opt	0.42	0.23	0.12	0.12	0.11
Border Swap	0.42	0.25	0.14	0.13	
Branch-and-Bound	0.42	0.23	0.12	0.11	0.11

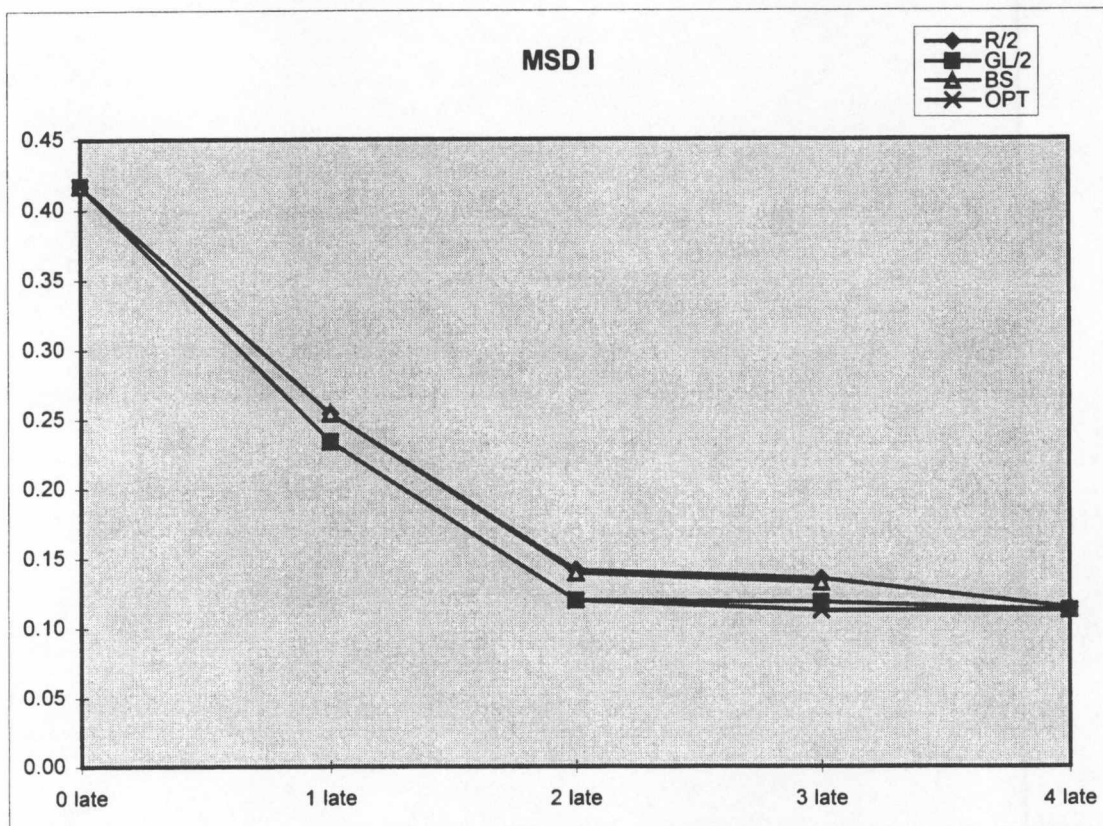


Figure 26: Graph of the Overall Mean Square Deviation for Demand Pattern I.

Table 41: Adjusted MSD Demand Pattern I.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late
Random-DD / 2-Opt	0.54	0.44	0.29	0.19	0.18
GC1LookAhead-DD/2-Opt	0.54	0.40	0.17	0.19	0.22
Border Swap	0.54	0.44	0.29	0.19	
Branch-and-Bound	0.54	0.40	0.18	0.13	0.21

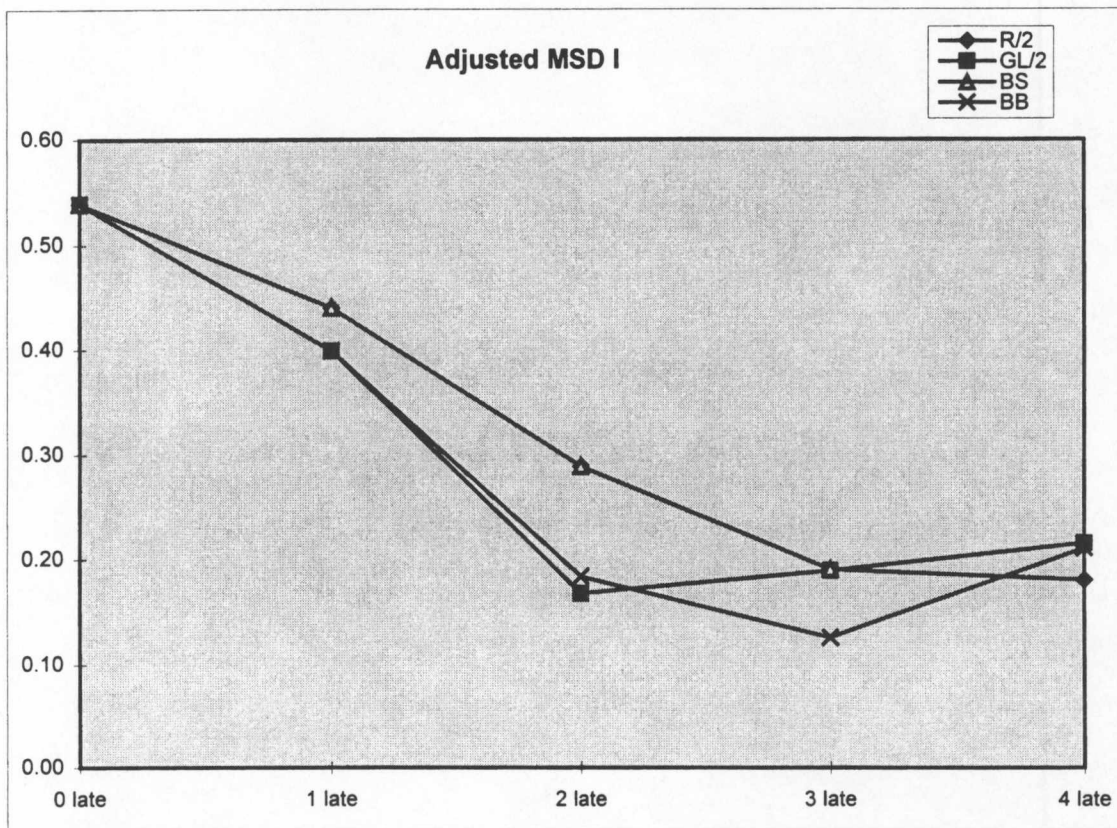


Figure 27: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern I.

Table 42: MSD Demand Pattern J.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.78	0.32	0.11	0.10
GC1LookAhead-DD/2-Opt	0.78	0.32	0.11	0.10
Border Swap	0.78	0.37	0.12	0.11
Branch-and-Bound	0.78	0.32	0.11	0.10

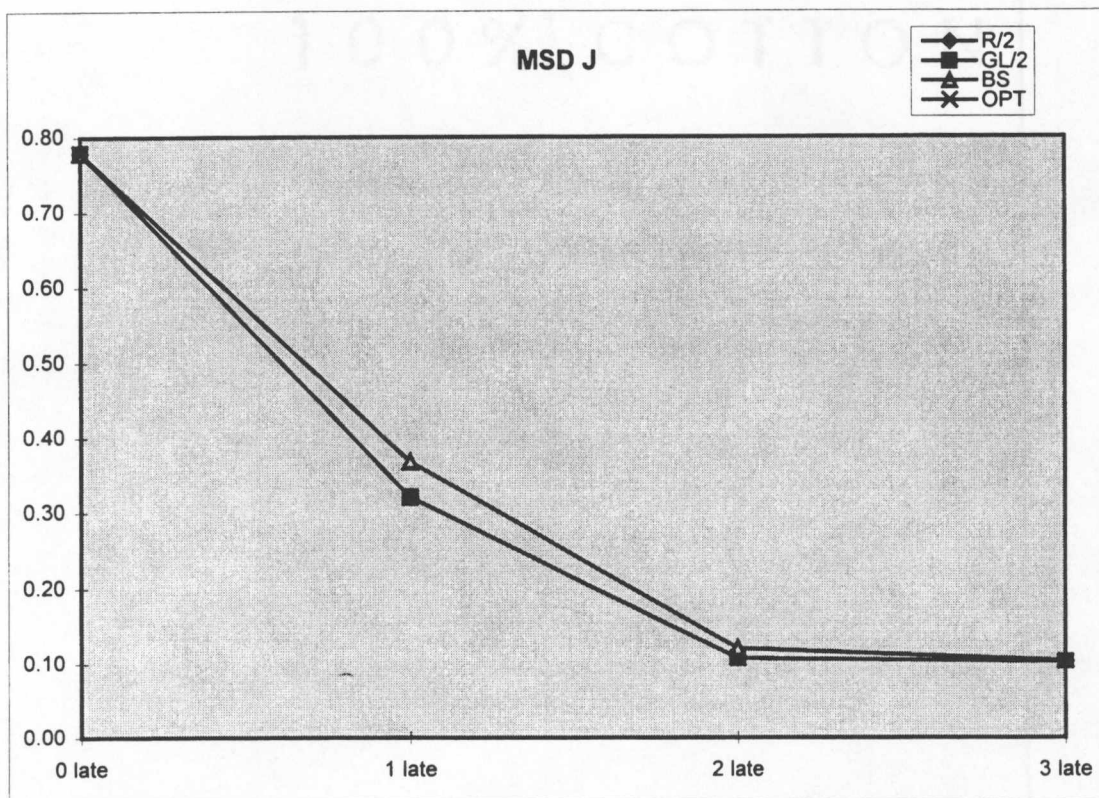


Figure 28: Graph of the Overall Mean Square Deviation for Demand Pattern J.

Table 43: Adjusted MSD Demand Pattern J.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.88	0.35	0.13	0.14
GC1LookAhead-DD/2-Opt	0.88	0.35	0.13	0.14
Border Swap	0.88	0.45	0.17	0.15
Branch-and-Bound	0.88	0.35	0.13	0.14

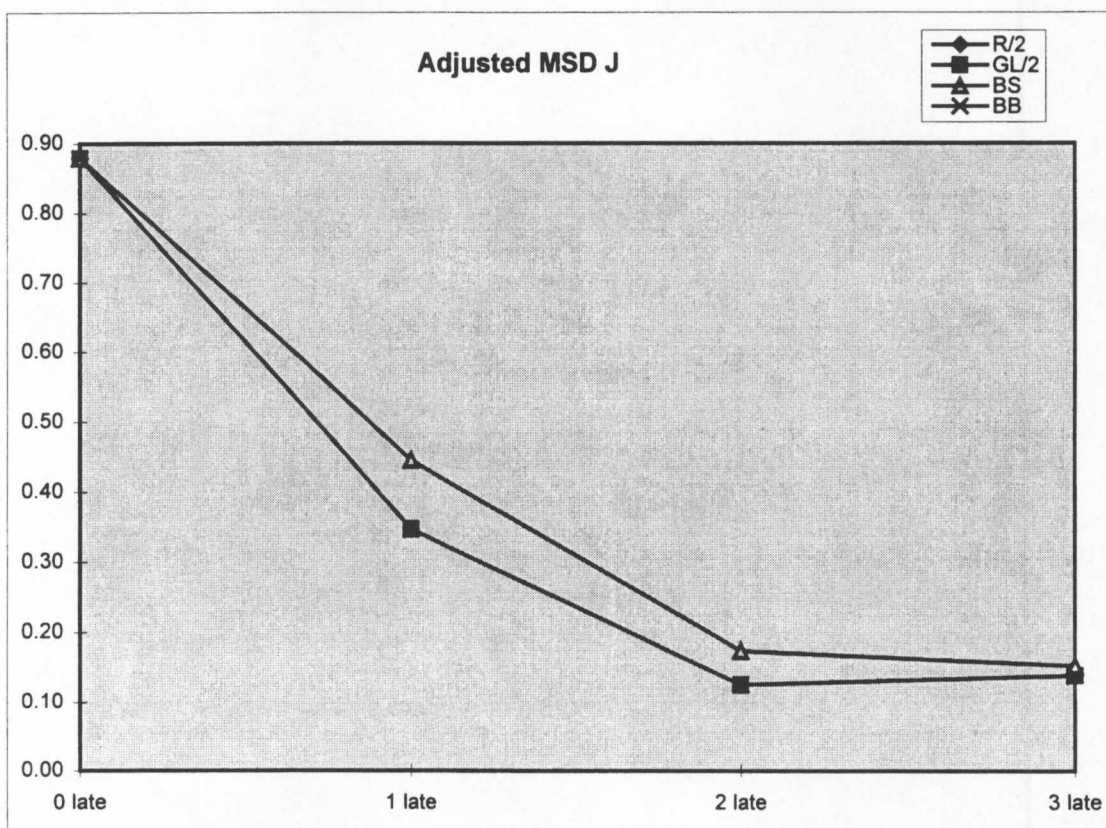


Figure 29: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern J.

4. The Large Test Problems

[illegible]

**Table 44: Initial Demand Patterns AA-JJ.
Random Over Uniform [0,24] Distribution.**

<i>Demand Pattern</i>	<i>Model</i>					<i>Demand Total</i>
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	
AA	16	17	10	9	22	74
BB	13	4	15	24	5	61
CC	14	22	8	20	7	71
DD	16	9	9	12	22	68
EE	8	0	23	16	16	63
FF	5	11	16	17	10	59
GG	9	22	1	2	8	42
HH	0	10	24	19	4	57
II	11	7	2	9	6	35
JJ	13	4	12	13	6	48

Table 45: Final Demand Patterns AA-JJ.
Random Over Uniform [0,24] Distribution Plus Alterations to Make Total Demand for Each Demand Pattern 60.

<i>Demand Pattern</i>	<i>Model</i>					<i>Demand Total</i>
	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	
AA	9	17	10	2	22	60
BB	13	3	15	24	5	60
CC	9	16	8	20	7	60
DD	16	5	5	12	22	60
EE	6	0	23	15	16	60
FF	5	11	16	17	11	60
GG	18	22	10	2	8	60
HH	1	10	24	19	6	60
II	24	7	2	9	18	60
JJ	19	4	12	19	6	60

Table 46: Due Dates in Order by Model for Each Demand Pattern AA - JJ.

	AA	BB	CC	DD	EE	FF	GG	HH	II	JJ
1	4	4	4	2	2	2	1	1	1	1
2	3	4	4	2	3	1	2	1	3	3
3	2	2	3	4	3	3	2	1	1	1
4	4	2	3	4	2	4	3	2	4	1
5	1	3	3	1	1	3	4	2	2	4
6	4	2	4	1	2	3	4	1	1	3
7	4	1	2	3	1	3	4	3	3	3
8	1	1	1	3	3	3	4	2	2	1
9	4	2	3	4	1	1	1	3	3	3
10	1	1	1	1	3	2	3	2	2	2
11	1	3	1	1	1	2	2	1	2	2
12	1	1	4	1	2	4	1	2	4	1
13	4	2	3	3	4	4	4	4	3	4
14	3	3	1	1	3	3	2	2	2	4
15	4	2	4	4	4	1	1	3	2	4
16	3	4	2	3	4	4	3	4	4	1
17	3	3	2	4	3	1	4	4	2	3
18	4	3	2	2	4	4	4	2	4	2
19	2	4	1	2	3	1	3	3	1	3
20	4	3	1	2	4	2	1	4	1	1
21	2	3	1	2	1	3	2	2	3	3

Table 46 (continued)

	AA	BB	CC	DD	EE	FF	GG	HH	II	JJ
22	3	1	3	3	1	1	2	2	2	4
23	2	2	3	4	1	3	4	2	4	4
24	1	3	2	1	4	2	2	3	2	3
25	1	3	4	2	2	2	2	2	2	1
26	3	4	3	1	4	1	2	4	2	1
27	2	4	4	2	3	4	3	3	2	4
28	2	3	3	1	2	4	4	2	1	4
29	2	1	2	1	4	4	4	1	3	4
30	2	2	3	2	1	1	1	4	4	3
31	3	3	2	3	2	2	1	1	1	4
32	4	3	2	1	4	4	3	1	1	3
33	1	4	4	4	2	4	3	3	1	2
34	4	1	4	1	1	4	2	2	3	3
35	2	4	4	3	3	3	3	3	4	4
36	2	4	2	2	1	2	3	4	3	2
37	3	4	4	2	4	2	1	1	1	4
38	1	1	1	1	3	3	3	4	3	3
39	2	1	4	4	2	2	2	1	1	2
40	4	3	4	3	2	1	4	3	2	3
41	1	4	3	3	4	4	1	4	2	2
42	3	1	1	3	3	3	2	4	3	1
43	3	1	4	3	4	1	2	3	3	3
44	3	1	2	4	3	2	4	3	3	1
45	4	3	1	4	1	3	4	3	3	1
46	4	2	3	4	1	1	1	4	1	2
47	1	1	3	3	2	4	4	4	2	4
48	2	2	1	3	3	1	3	4	4	2
49	3	2	2	3	2	4	3	1	4	2
50	3	1	2	1	3	1	1	3	4	2
51	1	4	1	2	4	3	1	2	4	4
52	1	2	1	1	1	1	3	4	4	3
53	2	3	4	4	2	2	4	3	4	2
54	1	1	2	3	4	3	3	1	4	1
55	1	4	2	2	4	2	2	2	1	2
56	2	2	1	2	1	2	1	1	3	1
57	4	4	3	4	3	3	1	4	4	1
58	3	4	2	4	2	2	3	1	1	2
59	3	2	3	2	2	4	1	1	1	4
60	2	2	1	4	1	1	2	3	3	2

5. Results Of The Large Test Problems

1. Goal 1: Level Part Usage with Zero Items Late and Problem Size = 60 Items

For the case of zero items allowed late, in terms of the average deviation from uniform component usage based on an overall goal (not a single period goal), all methods performed within a small percentage from each other as shown in Tables 47 and 48. For the large test problems, the GC1LookAhead-dd/2-Optimal-dd method requiring on average 7.22 seconds as shown in Table 49 provided the best solution for all demand patterns.

All other methods were then compared with this best solution. The Sort-dd/2-Optimal-dd method requiring on average the longest running time, 77.13 seconds, provided the solutions that were closest to the best. The solutions for this method were 0.15 percent from the best solution. The GC1LookAhead-dd method requiring on average 1.23 seconds provided the next best solutions being on average 0.48 percent from the best. Monden's original GC1 method requiring the shortest running time, on average 0.33 seconds fell in last place among the four being on average 0.68 percent from best. All methods provided solutions within 1 percent of each other.

After the sequences have been established based on the overall leveling goal, the products designated within each time period are then re-sequenced based on that period's single period goal according to the GC1LookAhead-dd algorithm. The newly adjusted sequences are then evaluated according to the same criteria: the overall deviation from uniform part usage. The results shown in Table 50 indicate no differences between any of the sequencing methods when no items are allowed to be sequenced late.

Table 47: Total Mean Square Deviation from Overall Uniform Component Usage with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>AA</i>	<i>BB</i>	<i>CC</i>	<i>DD</i>	<i>EE</i>	<i>FF</i>	<i>GG</i>	<i>HH</i>	<i>II</i>	<i>JJ</i>
GC1-dd	.461	1.200	.907	2.078	.384	.277	.360	1.820	1.635	1.488
GC1LookAhead-dd	.466	1.203	.911	2.078	.389	.268	.354	1.818	1.636	1.488
GC1LA-dd / 2opt-dd	.460	1.199	.904	2.078	.381	.268	.354	1.817	1.634	1.486
Sort-dd /2opt-dd	.460	1.202	.905	2.084	.381	.268	.357	1.817	1.634	1.486

Table 48: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Average MSD From Level Part Usage</i>	<i>Average % From Best</i>
GC1-dd	0.0029	0.68%
GC1LookAhead-dd	0.0030	0.48%
GC1LA-dd / 2opt-dd	0.0000	0.00%
Sort-dd /2opt-dd	0.0013	0.15%

Table 49: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with Zero Late Items. (Elapsed processing time based on an IBM-Compatible Pentium Processor, 166 Mhz, 12 Mb RAM)

<i>Sequencing Method</i>	<i>Demand Pattern</i>										<i>Avg. Time</i>
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>	
GC1-dd	.33	.33	.33	.33	.33	.33	.33	.33	.33	.33	.33
GC1LookAhead-dd	1.21	1.15	1.21	1.27	1.16	1.43	1.31	1.15	1.21	1.21	1.23
GC1LA-dd / 2opt-dd	8.89	6.32	9.06	4.34	8.63	4.94	6.48	3.90	4.23	15.43	7.22
Sort-dd /2opt-dd	84.09	70.47	67.44	68.05	87.83	99.69	68.38	80.57	72.56	72.23	77.13

Table 50: Total Mean Square Deviation from Uniform Component Usage Over All Periods for Sequences Adjusted and Leveled Within each Period with Zero Late Items.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>	<i>H</i>	<i>I</i>	<i>J</i>
GC1-dd	.808	2.572	1.942	4.257	1.218	.698	1.002	3.906	3.375	3.073
GC1LookAhead-dd	.808	2.572	1.942	4.257	1.218	.698	1.002	3.906	3.375	3.073
GC1LA-dd / 2opt-dd	.808	2.572	1.942	4.257	1.218	.698	1.002	3.906	3.375	3.073
Sort-dd /2opt-dd	.808	2.572	1.942	4.257	1.218	.698	1.002	3.906	3.375	3.073

2. Goal 2: Level Part Usage with No Due Date Restrictions and Problem Size = 60 Items

There is a little more distinction between the sequencing methods for the problems tested when there are no due date restrictions as shown in Tables 51 and 52. For the large test problems, the GC1LookAhead method requiring on average 1.44 seconds as shown in Table 53 provided the solutions that were best in terms of the average deviation from uniform component usage based on an overall goal (not a single period goal). The new Border Swap method provided solutions that were on average 0.43 percent from best in an average time of 18.39 seconds. Monden's original GC1 method requiring on average 0.30 seconds, the smallest time, fell in last place among the three being on average 8.9 percent from the best.

After the sequences have been established based on the overall leveling goal, the products designated within each time period are then re-sequenced based on that period's single period goal according to the GC1LookAhead-dd algorithm. The newly adjusted sequences are then evaluated according to the same criteria: the overall deviation from uniform part usage. The results shown in Tables 54 and 55 indicate similar results after the sequences have been modified. The GC1LookAhead algorithm performed the best being on average 0.62 percent

Table 51: Mean Square Deviation from Uniform Component Usage, and Number of Items Late Respectively with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>AA</i>	<i>BB</i>	<i>CC</i>	<i>DD</i>	<i>EE</i>	<i>FF</i>	<i>GG</i>	<i>HH</i>	<i>II</i>	<i>JJ</i>
GC1	.119 6	.137 7	.127 5	.131 7	.122 3	.123 3	.126 3	.154 4	.122 5	.137 5
GC1LookAhd	.116 6	.122 7	.118 5	.128 8	.107 3	.119 3	.118 5	.125 6	.120 5	.119 6
Border Swap	.120 6	.123 6	.118 5	.128 8	.107 3	.119 3	.118 3	.125 6	.120 6	.119 6

Table 52: Average Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Each Sequencing Method with the Objective of Leveling Part Usage Over All Periods with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance

<i>Sequencing Method</i>	<i>Average MSD From Level Part Usage</i>	<i>Average % From Best</i>
GC1	0.0106	8.90%
GC1LookAhd	0.0000	0.00%
Border Swap	0.0005	0.43%

Table 53: Computer Processing Time in Seconds for Each Sequencing Method with the Objective of Leveling Part Usage with No Due Date Restrictions. (Elapsed processing time based on an IBM-Compatible Pentium Processor, 166 Mhz, 12 Mb RAM)

Sequencing Method	Demand Pattern										Avg. Time
	AA	BB	CC	DD	EE	FF	GG	HH	II		
GC1	.113	.113	.38	.33	.33	.33	.38	.33	.33	.33	.30
GC1LookAhd	1.65	1.59	1.64	.160	1.37	1.64	1.59	1.49	1.60	1.65	1.44
Border Swap	9.08	8.64	7.45	11.6 0	3.66	4.47	4.32	8.04	8.70	9.00	7.50

Table 54: Total Mean Square Deviation from Uniform Component Usage Over All Periods and Number of Items Late Respectively for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance.

<i>Sequencing Method</i>	<i>Demand Pattern</i>									
	<i>AA</i>	<i>BB</i>	<i>CC</i>	<i>DD</i>	<i>EE</i>	<i>FF</i>	<i>GG</i>	<i>HH</i>	<i>II</i>	<i>JJ</i>
GC1	.202 6	.202 7	.158 5	.183 7	.222 3	.193 3	.162 3	.217 4	.192 5	.221 5
GC1LookAhd	.186 6	.202 7	.158 5	.180 8	.162 3	.193 3	.172 5	.189 6	.192 5	.182 6
Border Swap	.202 6	.211 6	.178 5	.180 8	.162 3	.193 3	.162 3	.189 6	.211 6	.182 6

Table 55: Average Adjusted Mean Square Deviation from Uniform Component Usage, and Average Percent from the Best Component Usage Deviation for Sequences Adjusted and Leveled Within each Period with No Due Date Restrictions.

Note: Numbers in boldface represent the best performance

<i>Sequencing Method</i>	<i>Average MSD From Best Part Usage</i>	<i>Average % From Best</i>
GC1	0.0146	8.35%
GC1LookAhd	0.0010	0.62%
Border Swap	0.0064	3.56%

from best. The Border Swap algorithm was next at 3.56 percent from optimal on average. The original GC1 algorithm fell in last place being on average 8.35 percent from the best solution.

3. Goal 3: Level Part Usage with X Items Late and Problem Size = 60 Items

As with the small test problems, first the results of the performance of the three sequencing heuristics are evaluated, then the individual tradeoff curves for each demand pattern are examined. For the large test problems, the deviation from level component usage was measured for one item late, two items late, etc. until there was no longer a decrease in the deviation. Each demand pattern has its own unique tradeoff curve which will be shown later.

The sequencing heuristic can have an effect on the number of items late to minimize the MSD from level component usage. Table 56 and Figure 30 shows the average percent from optimal for each sequencing method with the sequence based on the overall period goals. For each number of late items, the total number of observations from ten possible observations are shown.

Table 56: The Average Percent from Best for each Sequencing Method
with the Sequence Based on Over All Period Goals
for Demand Patterns AA-JJ with 60 Items.

	# Observations Per Method Respectively, and # Late								
	10,10,10 0 late	10,10,10 1 late	10,10,10 2 late	10,10,10 3 late	7,7,7 4 late	7,7,7 5 late	4,4,6 6 late	1,1,1 7 late	0,1,1 8 late
S/2	0%	53%	45%	32%	82%	48%	56%	8%	8%
G/2	0%	8%	18%	2%	12%	16%	13%	6%	
BS	0%	8%	22%	23%	22%	4%	0%	0%	

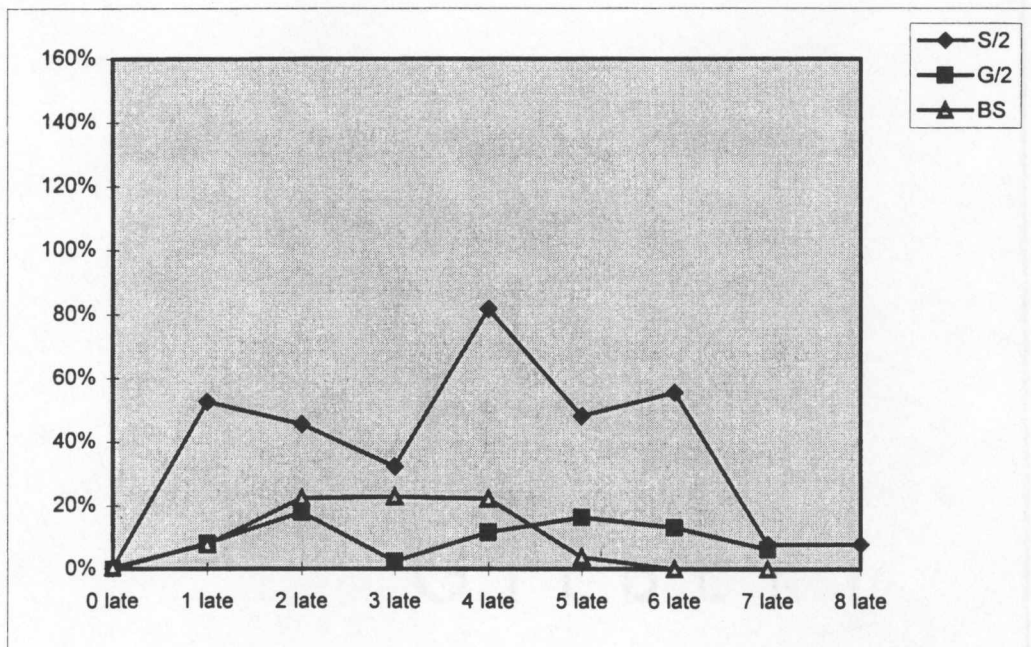


Figure 30: Graph of the Average Percent from the Best Mean Square Deviation for
Demand Patterns AA through JJ.

The sequencing heuristic can also have an effect on the number of items late to minimize the MSD from level component usage. Table 56 shows the number of total observations from ten possible observations for each number of items late.

For the number of late items with the largest numbers of observations, 0 late, 1 late, 2 late, and 3 late, the 2-Optimal heuristic starting with the Goal Chasing LookAhead-dd solution provided the best solutions on average. The Border Swap heuristic provided very similar good results. The 2-Optimal heuristic starting with a sorted solution provided less desirable results as compared with the other two methods.

Table 57 and Figure 31 shows the results after the sequences have been adjusted and balanced within each due date period. The newly adjusted sequences are then compared based on the same overall goal of minimizing the MSD from level component usage over all periods. The performance of the heuristics are even more exaggerated. The Goal Chasing LookAhead-dd/2-Optimal and the 'faster' Border Swap heuristic have similar results for 0 late through 3 late, then for 4 late through 6 late, the Border Swap method performs the best. As before, the Sort-dd/2-Optimal heuristic with the longest running time falls in last place.

Tables 58 through 77 and the corresponding Figures 32 through 51 respectively, show the results of the individual tradeoff curves for each large test problem with demand patterns AA through JJ respectively. The results for all problems show that allowing one item to be sequenced late will increase performance with respect to reducing the MSD from level component usage. This is true for all sequencing methods. In some cases there is also a large decrease in MSD by allowing two or more items to be sequenced late. There seems to be less of an advantage the more items allowed late, and in some cases there is no advantage.

Table 57: The Average Percent from Best for each Sequencing Method
with the Sequence Adjusted Based on Single Period Goals
for Demand Patterns AA-JJ with 60 Items.

# Observations Per Method Respectively, and # Late									
	10,10,10	10,10,10	10,10,10	10,10,10	7,7,7	7,7,7	4,4,6	1,1,1	0,1,1
	0 late	1 late	2 late	3 late	4 late	5 late	6 late	7 late	8 late
S/2	0%	70%	62%	28%	138%	105%	156%	0%	8%
G/2	0%	13%	21%	11%	53%	34%	42%	55%	34%
BS	0%	11%	25%	19%	15%	6%	0%	13%	

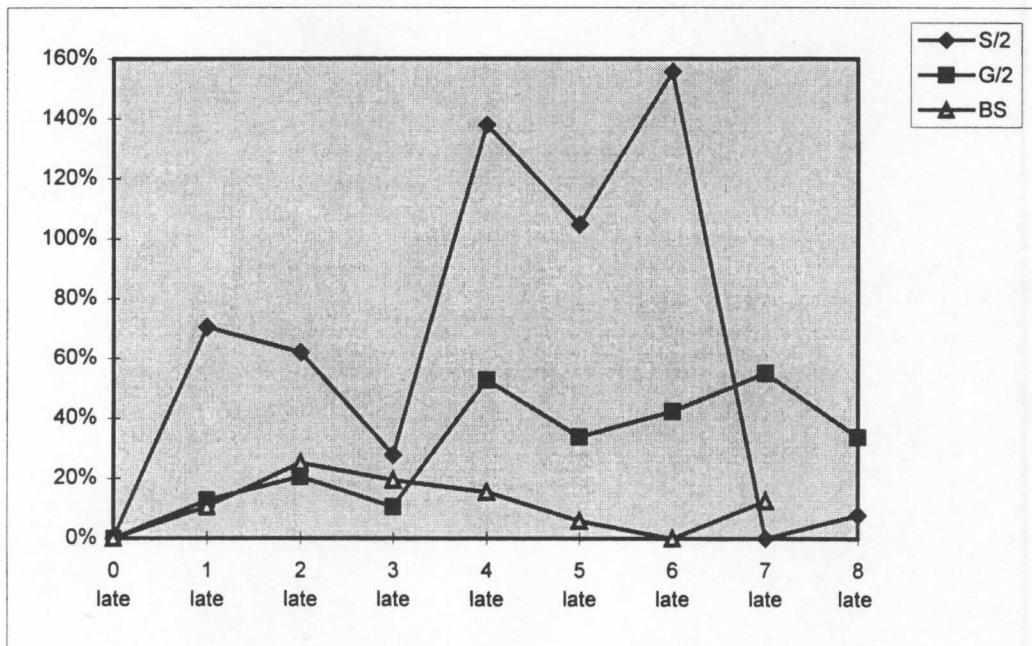


Figure 31: Graph of the Adjusted Average Percent from the Best
Mean Square Deviation for Demand Patterns AA through JJ.

Table 58: MSD Demand Pattern AA.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 lat	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	0.46	0.54	0.35	0.16	0.15	0.14	
GC1LookAhead-DD/2-Opt	0.46	0.27	0.17	0.14	0.13	0.12	
Border Swap	0.47	0.29	0.21	0.17	0.13	0.12	0.12

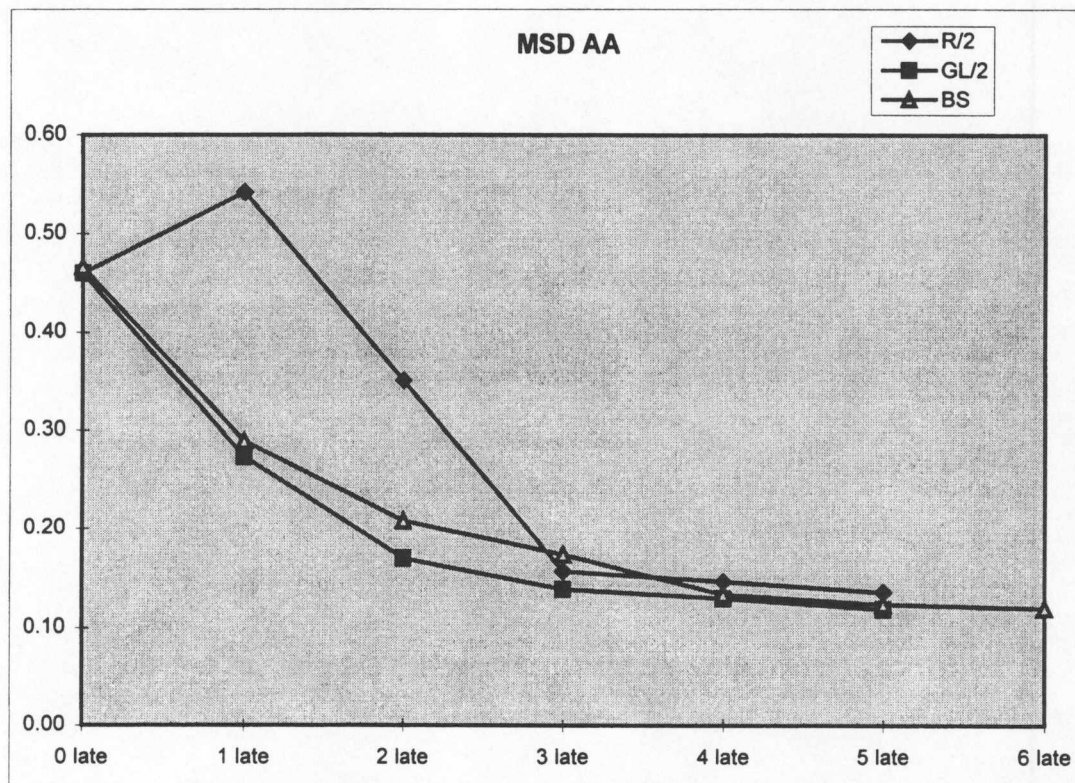


Figure 32: Graph of the Overall Mean Square Deviation for Demand Pattern AA.

Table 59: Adjusted MSD Demand Pattern AA.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 lat	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	0.81	1.26	1.03	0.27	0.28	0.20	
GC1LookAhead-DD/2-Opt	0.81	0.54	0.34	0.27	0.28	0.20	
Border Swap	0.81	0.56	0.39	0.31	0.21	0.22	0.20

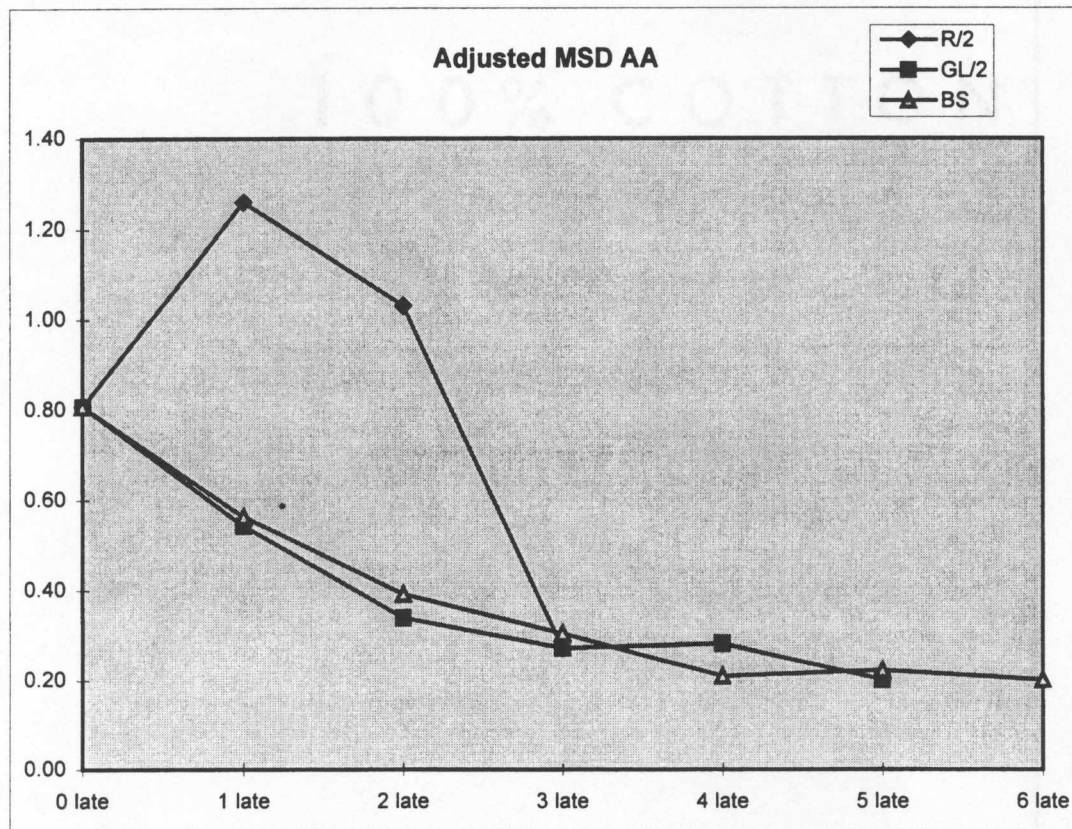


Figure 33: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern AA.

Table 60: MSD Demand Pattern BB.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	1.20	0.68	0.47	0.34	0.33	0.14	0.14
GC1LookAhead-DD/2-Opt	1.20	0.59	0.33	0.24	0.14	0.13	0.13
Border Swap	1.20	0.60	0.41	0.32	0.20	0.13	0.12

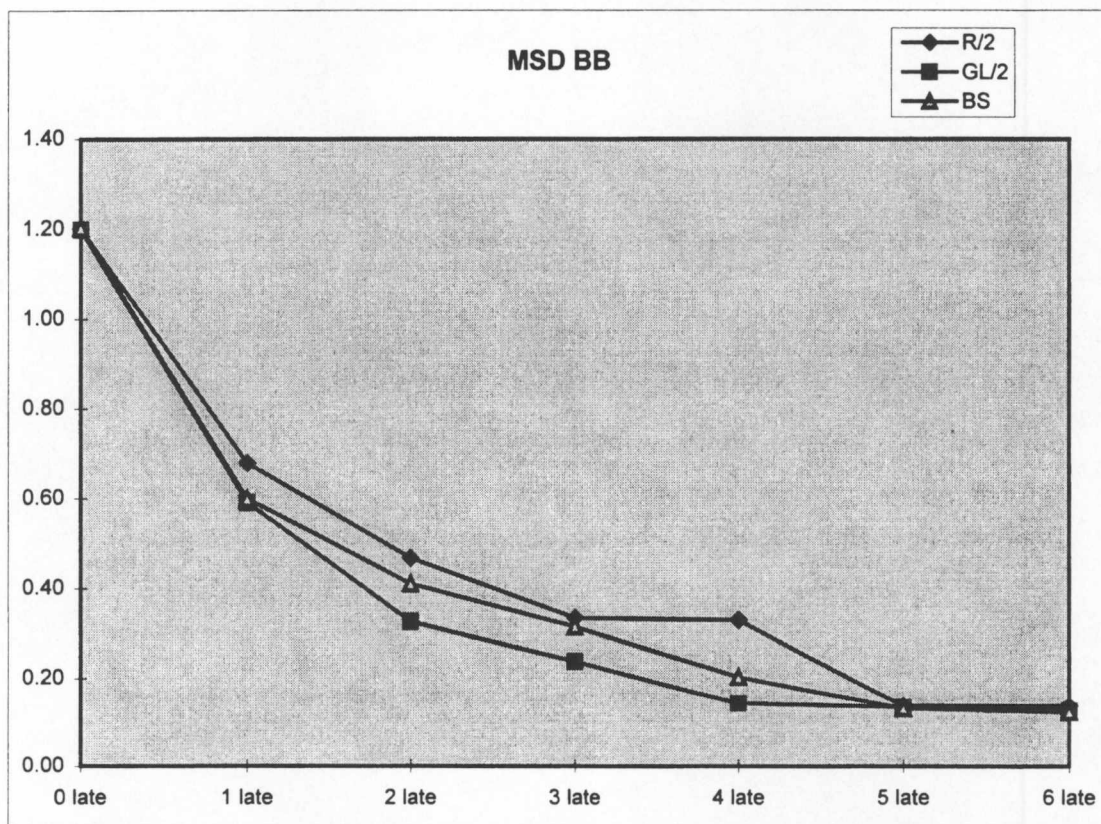


Figure 34: Graph of the Overall Mean Square Deviation for Demand Pattern BB.

Table 61: Adjusted MSD Demand Pattern BB.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	2.57	1.67	1.09	0.90	0.95	0.30	0.24
GC1LookAhead-DD/2-Opt	2.57	1.38	0.78	0.63	0.31	0.24	0.21
Border Swap	2.57	1.38	1.11	0.75	0.54	0.24	0.21

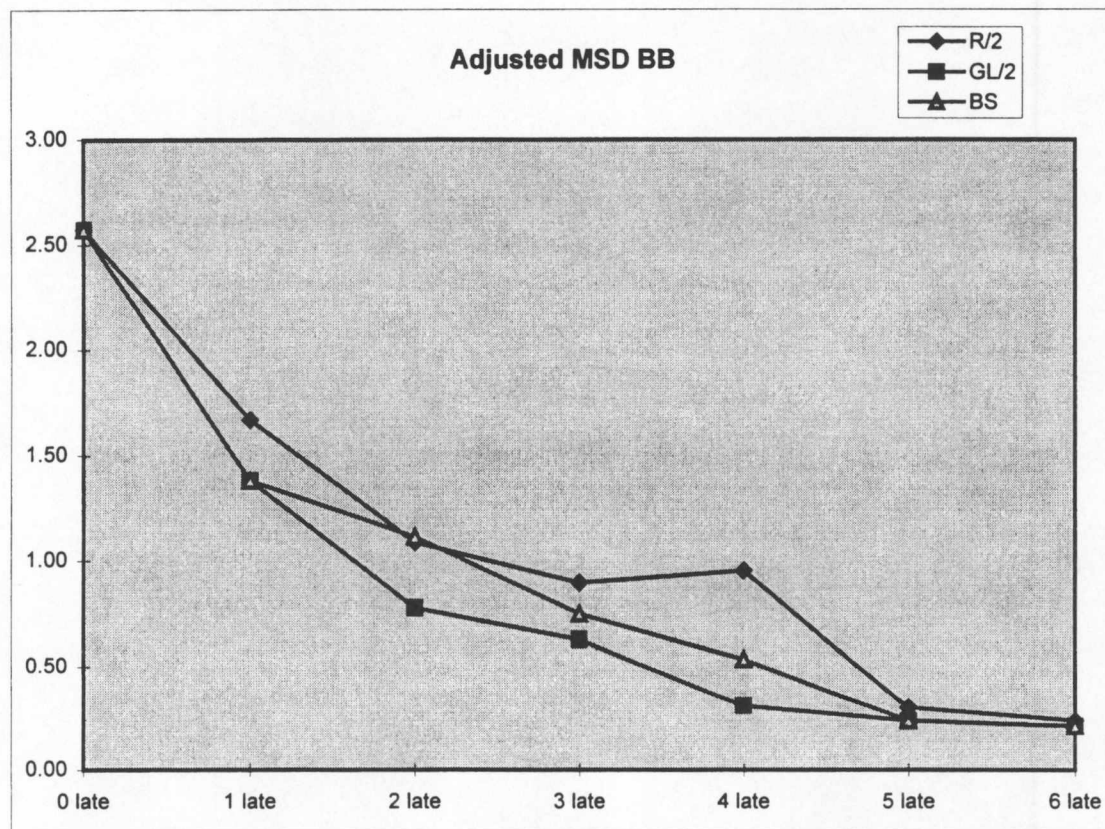


Figure 35: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern BB.

Table 62: MSD Demand Pattern CC.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	0.91	0.50	0.37	0.34	0.17	0.12
GC1LookAhead-DD/2-Opt	0.90	0.50	0.25	0.15	0.13	0.12
Border Swap	0.91	0.49	0.36	0.16	0.14	0.12

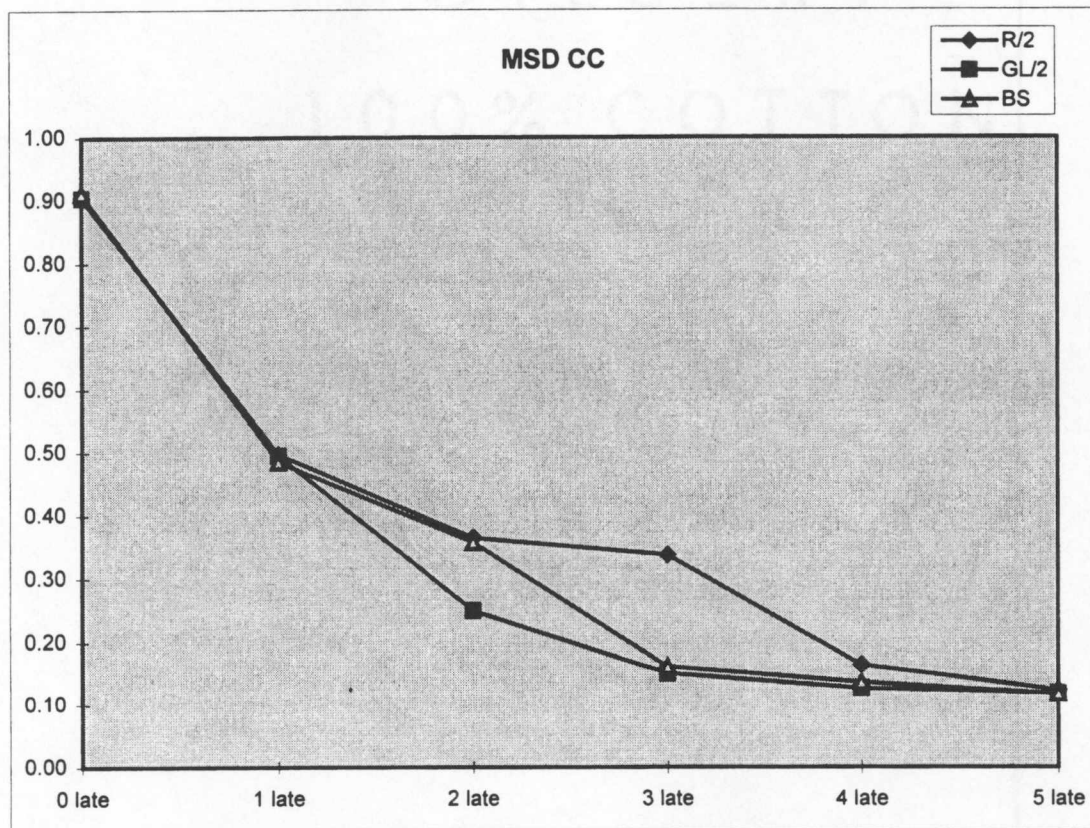


Figure 36: Graph of the Overall Mean Square Deviation for Demand Pattern CC.

Table 63: Adjusted MSD Demand Pattern CC.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	1.94	1.18	0.87	0.85	0.34	0.16
GC1LookAhead-DD/2-Opt	1.94	1.18	0.69	0.35	0.26	0.16
Border Swap	1.94	1.16	0.97	0.41	0.24	0.18

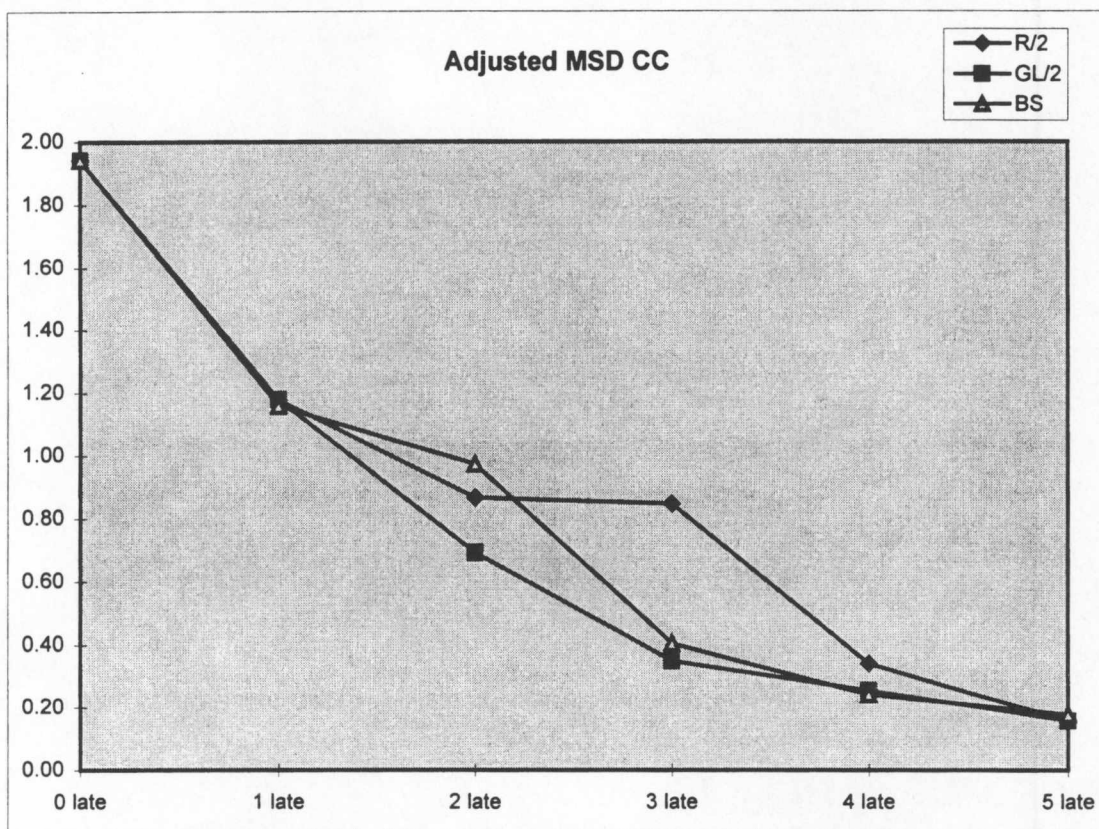


Figure 37: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern CC.

Table 64: MSD Demand Pattern DD.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Seq. Meth	0 late	1 late	2 late	3 late	4 late	5 late	6 late	7 late	8 late
Ran/2Opt	2.08	1.05	0.54	0.41	0.34	0.24	0.26	0.14	0.14
GC1LA/2	2.08	1.04	0.55	0.35	0.31	0.24	0.19	0.14	0.13
Bor Swap	2.08	1.24	0.88	0.71	0.29	0.14	0.14	0.13	0.13

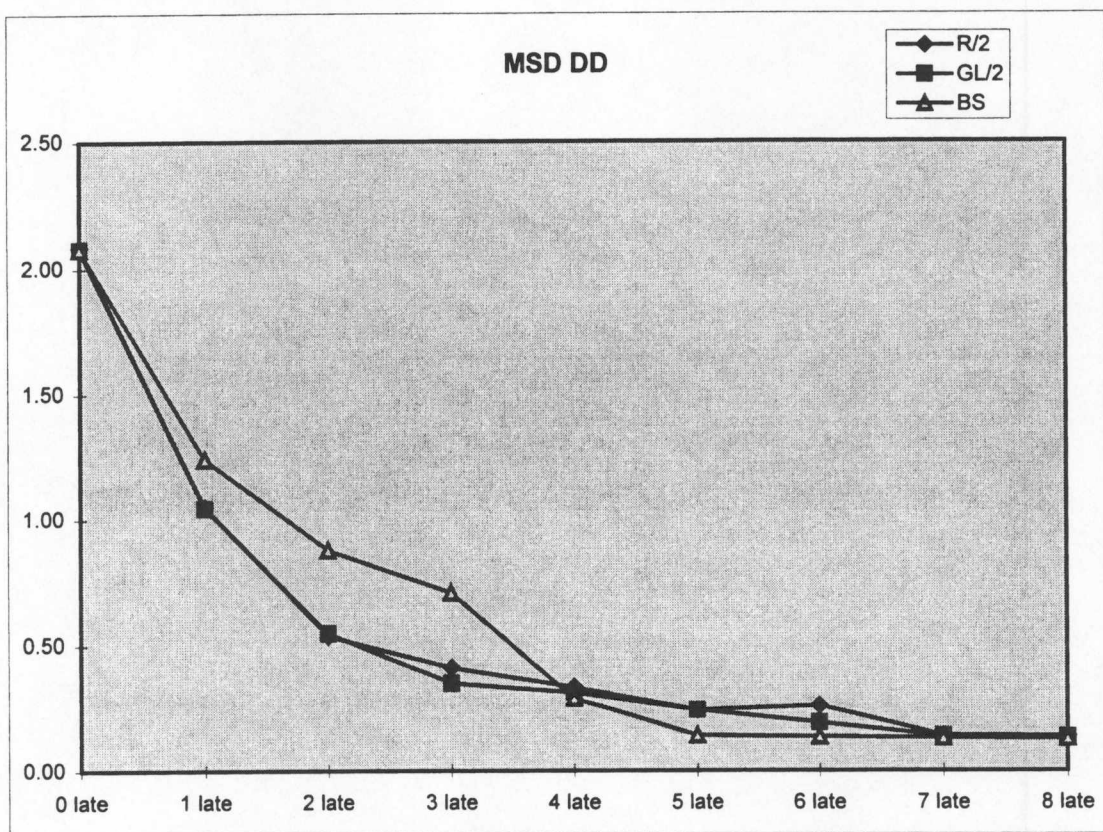


Figure 38: Graph of the Overall Mean Square Deviation for Demand Pattern DD.

Table 65: Adjusted MSD Demand Pattern DD.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Seq. Meth	0 late	1 late	2 late	3 late	4 late	5 late	6 late	7 late	8 late
Ran/2Opt	4.26	2.52	1.20	0.80	0.62	0.55	0.71	0.18	0.00
GC1LA/2	4.26	2.52	1.35	1.01	1.02	0.84	0.64	0.28	0.24
Bor Swap	4.26	2.79	2.20	1.77	0.78	0.34	0.25	0.21	0.18

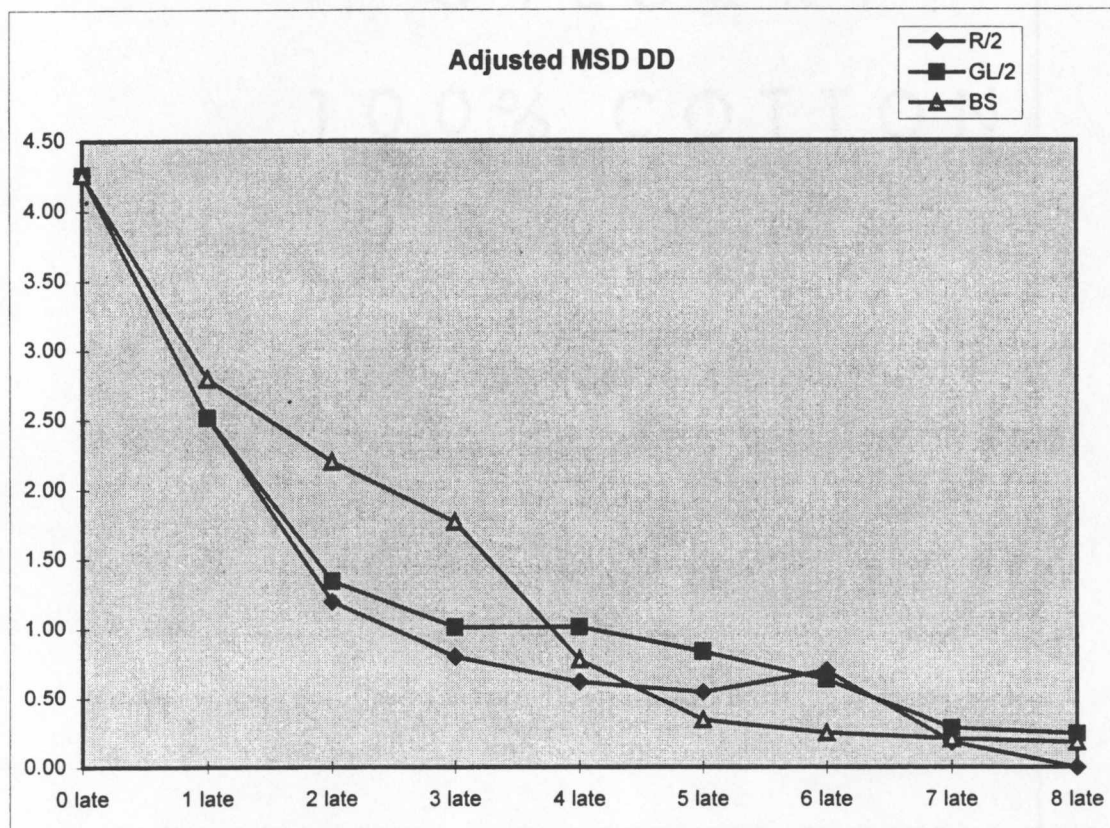


Figure 39: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern DD.

Table 66: MSD Demand Pattern EE.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.38	0.50	0.15	0.11
GC1LookAhead-DD/2-Opt	0.38	0.25	0.17	0.11
Border Swap	0.39	0.19	0.12	0.11

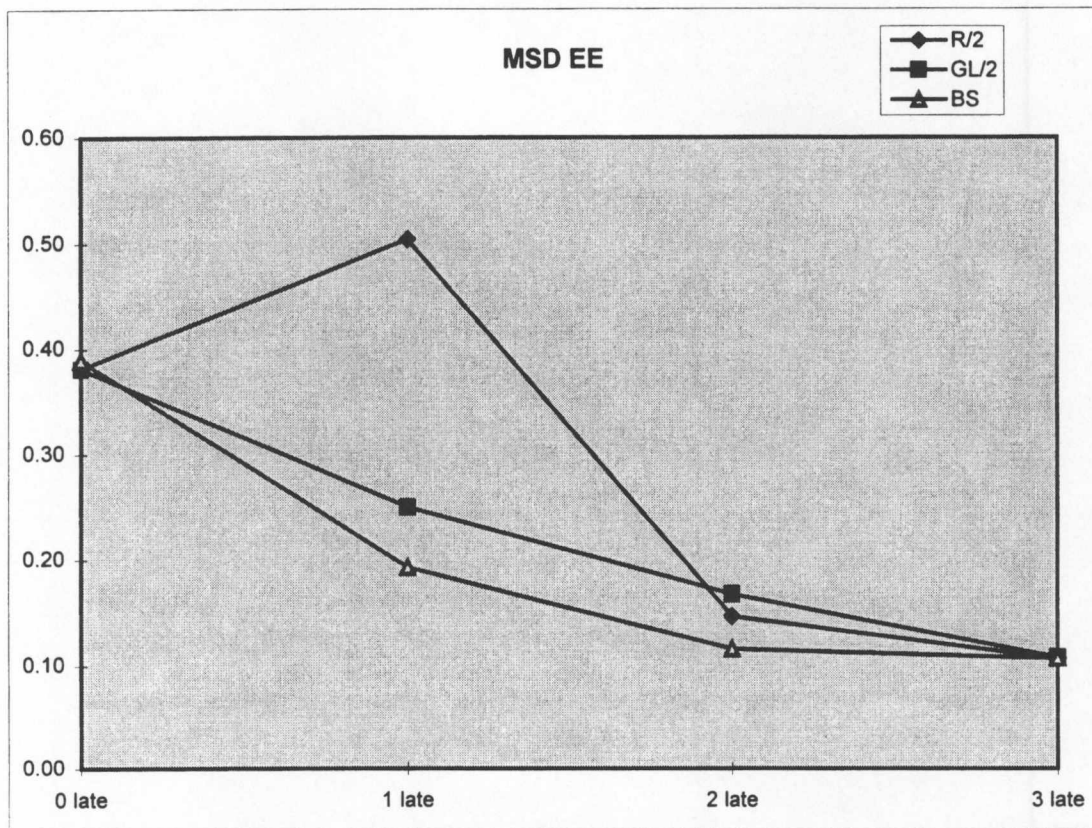


Figure 40: Graph of the Overall Mean Square Deviation for Demand Pattern EE.

Table 67: Adjusted MSD Demand Pattern EE.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	1.22	1.75	0.49	0.16
GC1LookAhead-DD/2-Opt	1.22	0.77	0.56	0.16
Border Swap	1.22	0.58	0.25	0.16

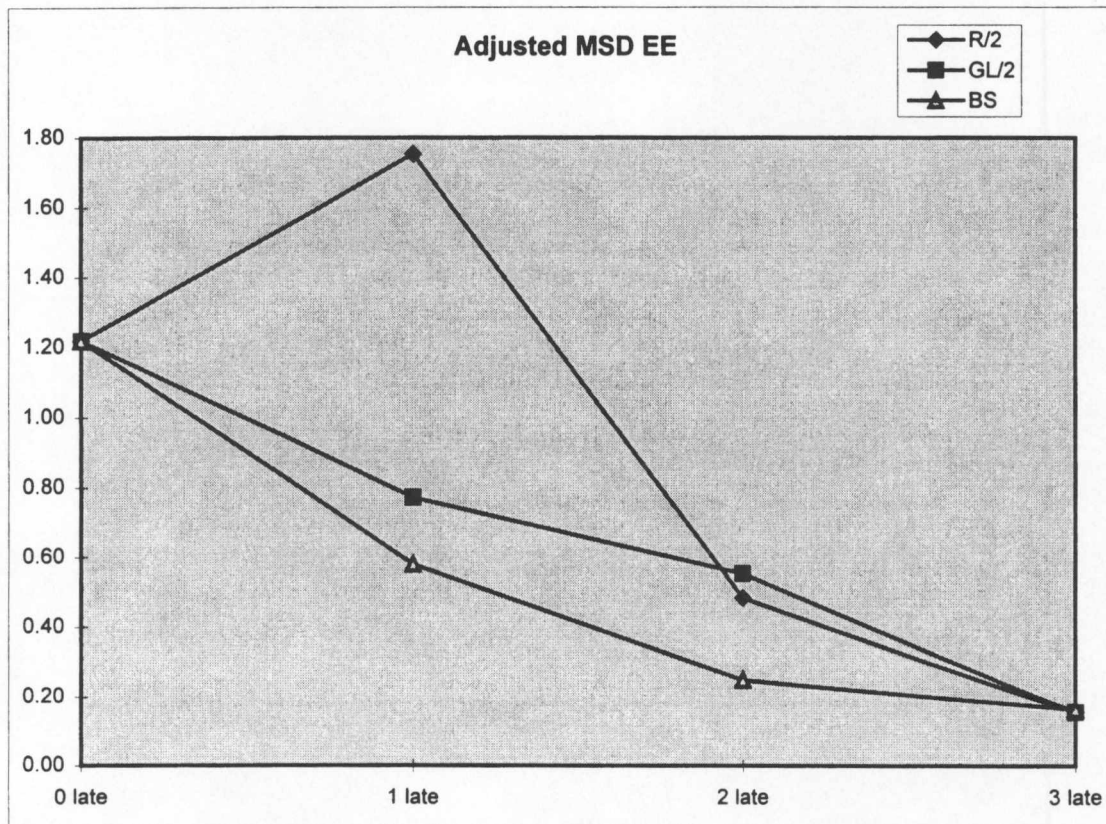


Figure 41: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern EE.

Table 68: MSD Demand Pattern FF.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.27	0.37	0.24	0.13
GC1LookAhead-DD/2-Opt	0.27	0.20	0.14	0.13
Border Swap	0.27	0.14	0.13	0.12

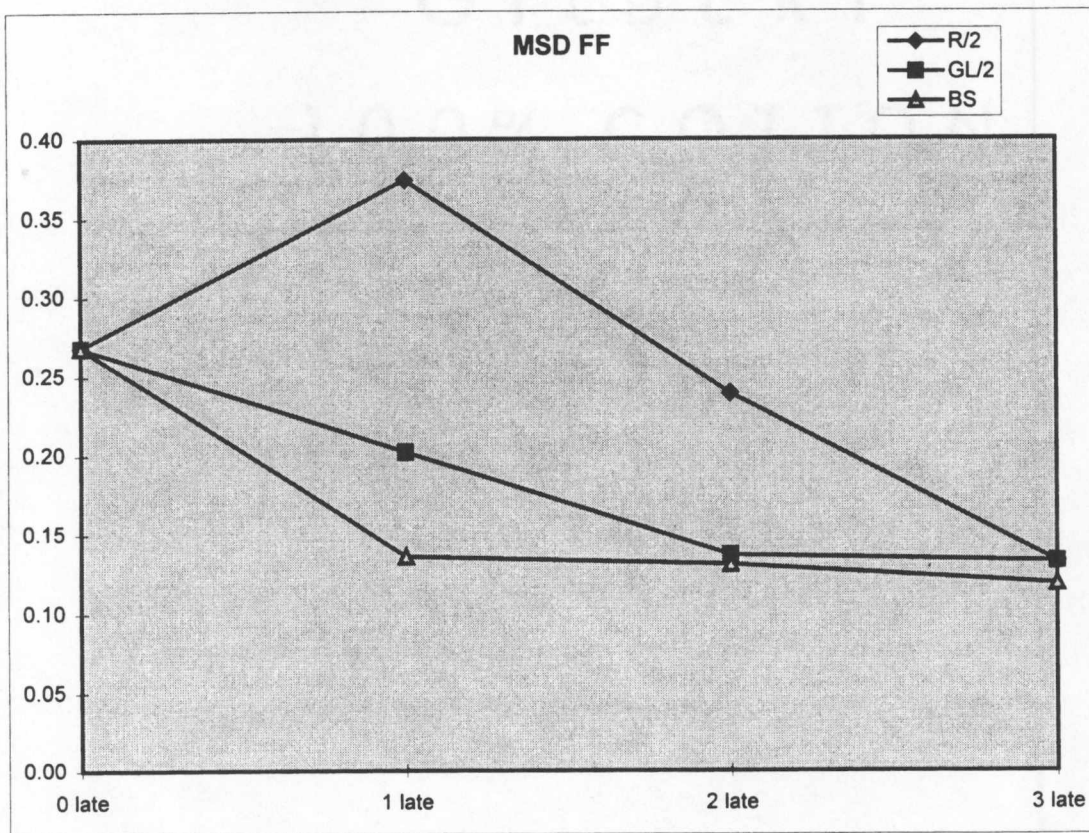


Figure 42: Graph of the Overall Mean Square Deviation for Demand Pattern FF.

Table 69: Adjusted MSD Demand Pattern FF.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.70	1.18	0.77	0.20
GC1LookAhead-DD/2-Opt	0.70	0.48	0.30	0.28
Border Swap	0.70	0.29	0.28	0.19

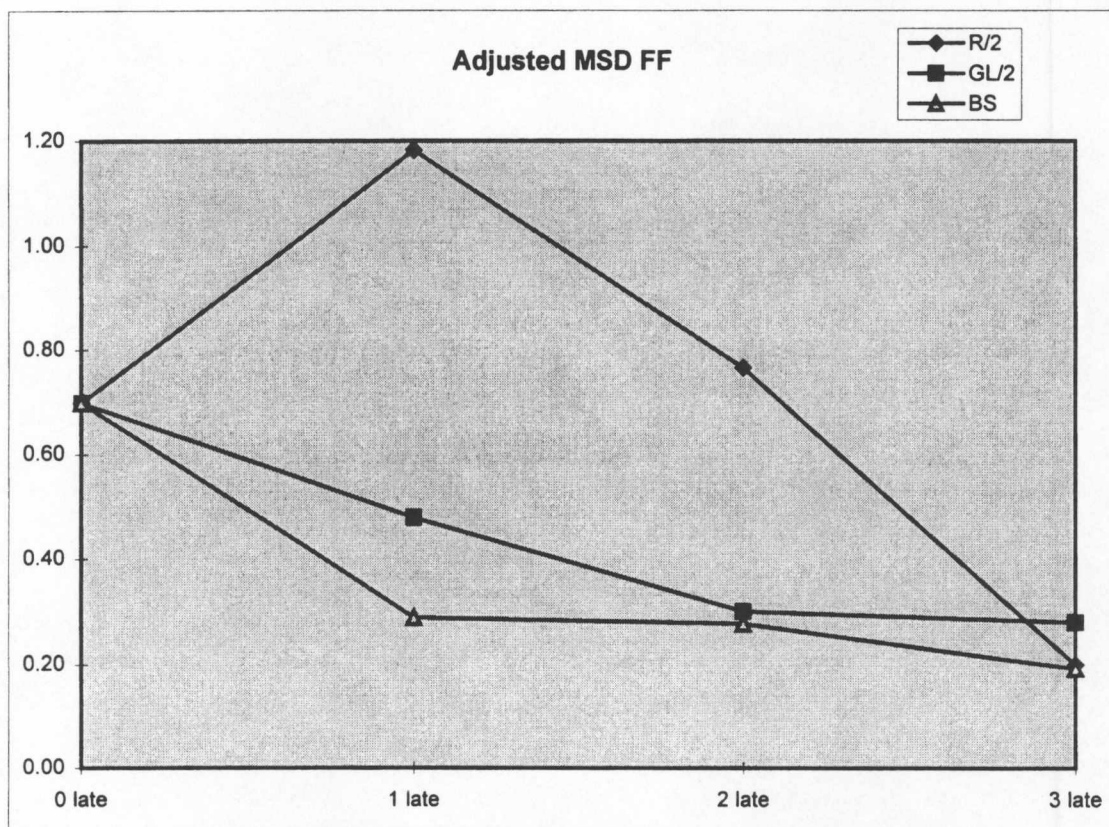


Figure 43: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern FF.

Table 70: MSD Demand Pattern GG.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	0.36	0.31	0.20	0.13
GC1LookAhead-DD/2-Opt	0.35	0.23	0.20	0.13
Border Swap	0.35	0.23	0.12	0.12

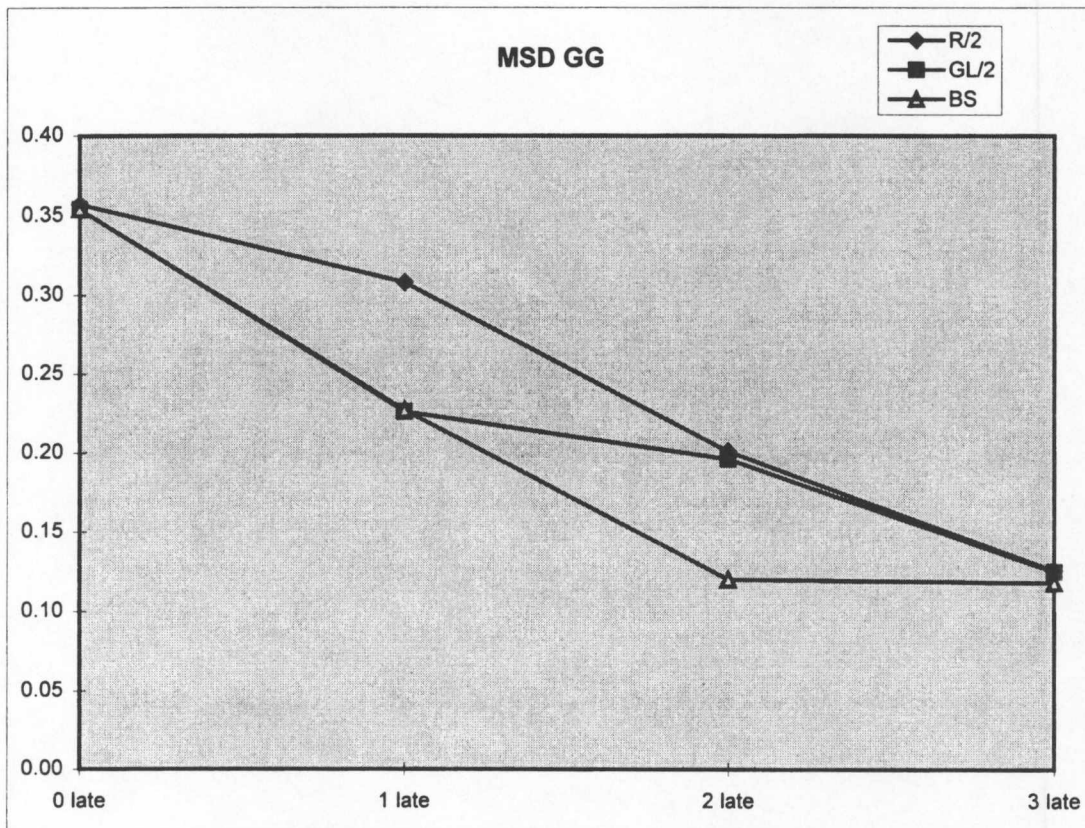


Figure 44: Graph of the Overall Mean Square Deviation for Demand Pattern GG.

Table 71: Adjusted MSD Demand Pattern GG.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late
Random-DD / 2-Opt	1.00	0.81	0.53	0.16
GC1LookAhead-DD/2-Opt	1.00	0.66	0.50	0.16
Border Swap	1.00	0.66	0.17	0.16

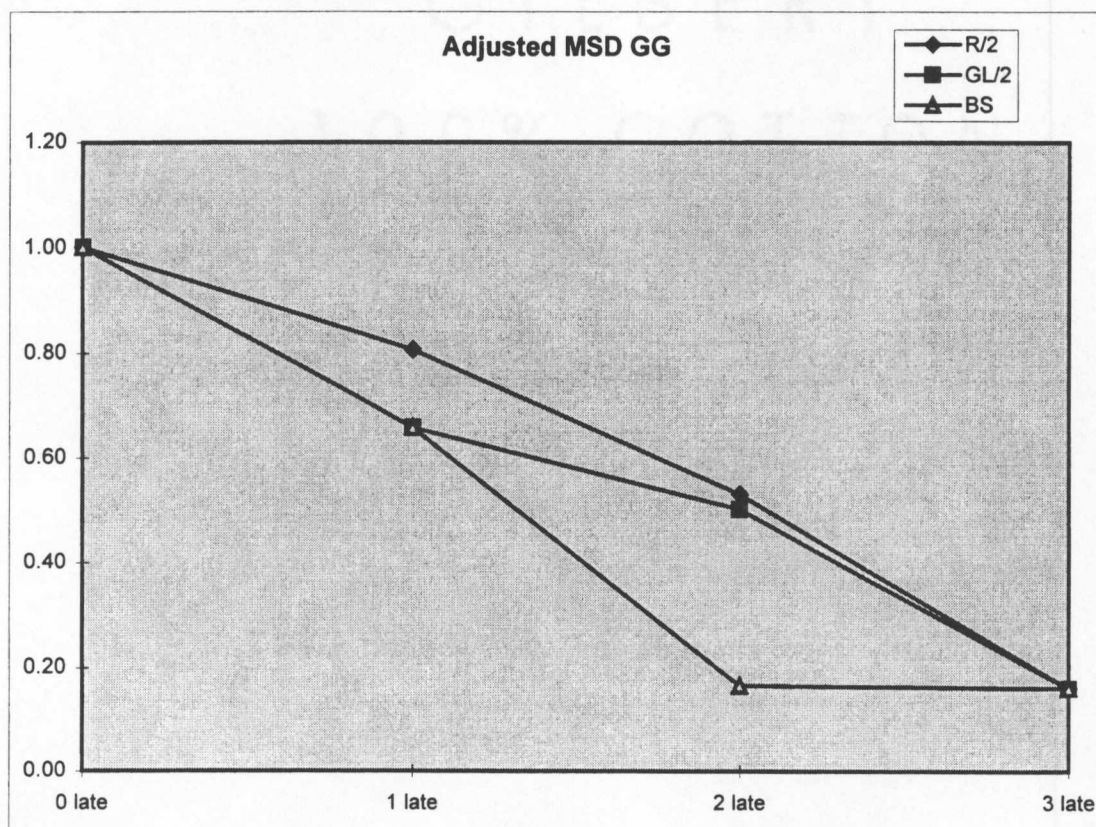


Figure 45: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern GG.

Table 72: MSD Demand Pattern HH.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	1.82	0.89	0.48	0.39	0.39	0.25	0.18
GC1LookAhead-DD/2-Opt	1.82	0.89	0.48	0.26	0.22	0.14	0.13
Border Swap	1.82	0.88	0.59	0.33	0.14	0.13	0.13

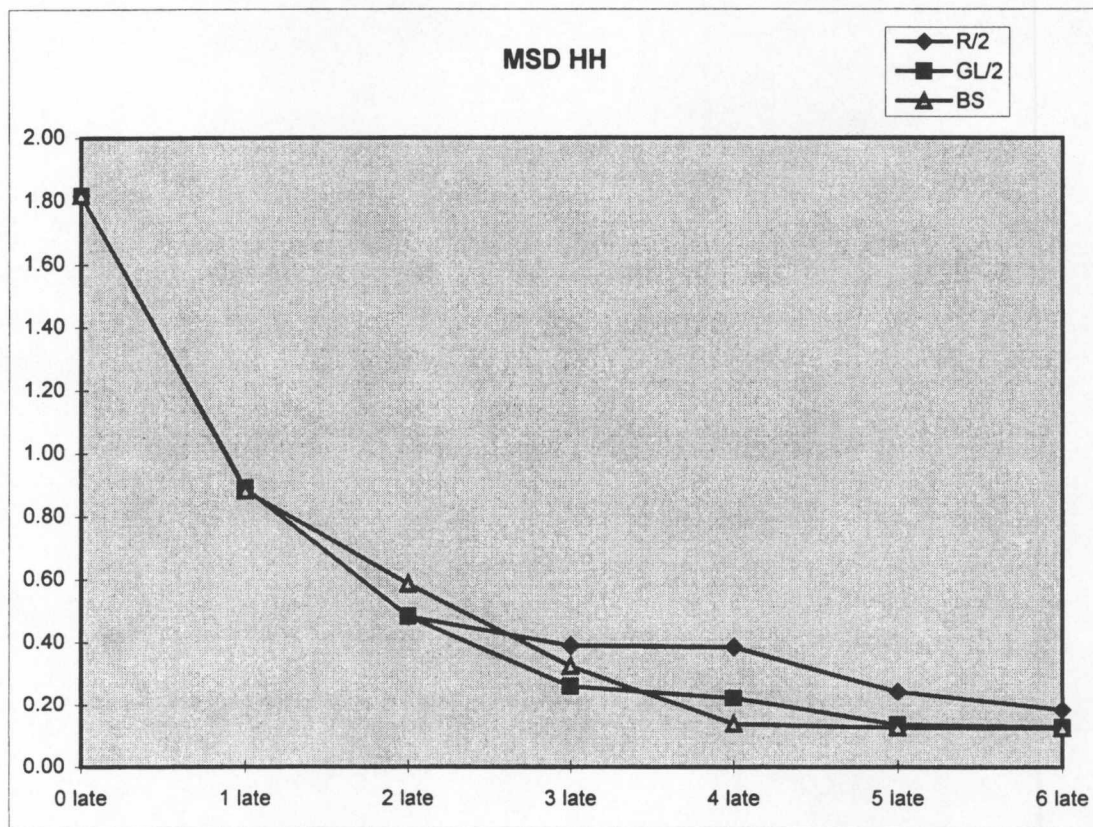


Figure 46: Graph of the Overall Mean Square Deviation for Demand Pattern HH.

Table 73: Adjusted MSD Demand Pattern HH.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	3.91	2.25	1.24	1.05	1.11	0.67	0.51
GC1LookAhead-DD/2-Opt	3.91	2.25	1.24	0.89	0.75	0.28	0.21
Border Swap	3.91	2.19	1.59	1.03	0.22	0.21	0.19

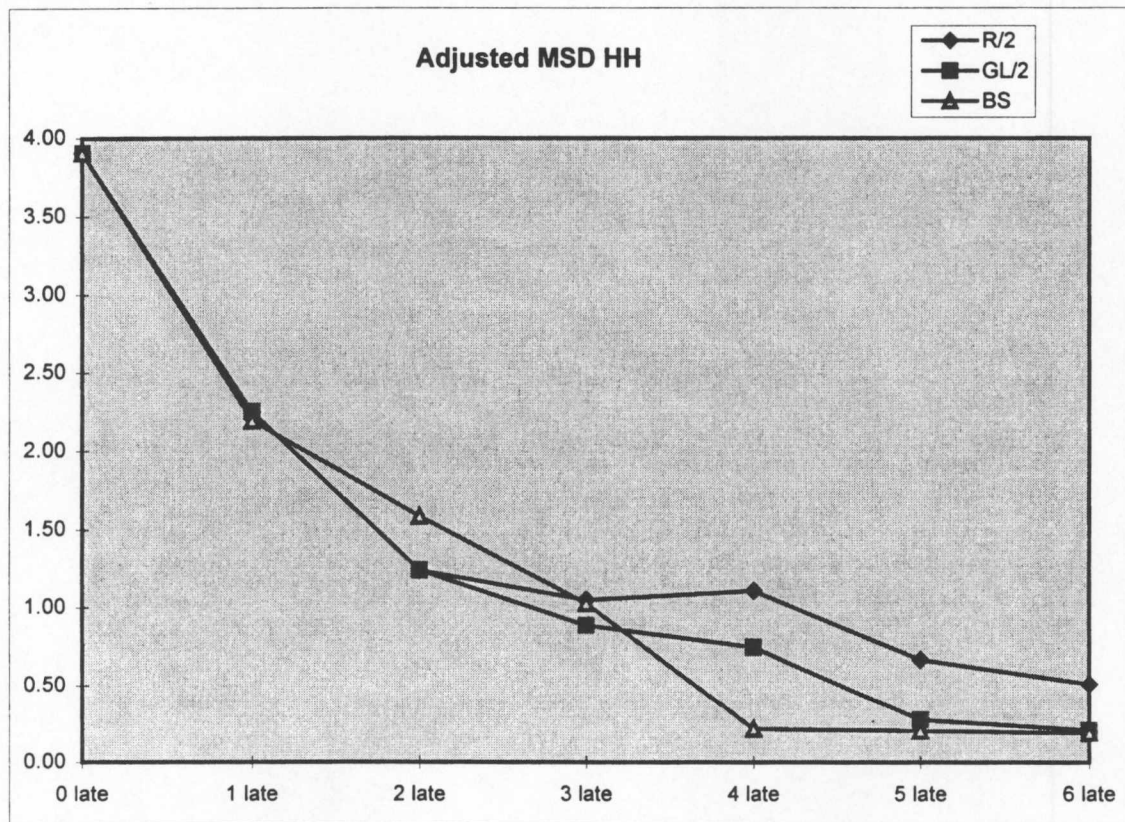


Figure 47: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern HH.

Table 74: MSD Demand Pattern II.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	1.63	0.75	0.42	0.37	0.36	0.22	0.21
GC1LookAhead-DD/2-Opt	1.63	0.75	0.37	0.24	0.16	0.14	0.13
Border Swap	1.64	1.13	0.54	0.33	0.29	0.17	0.12

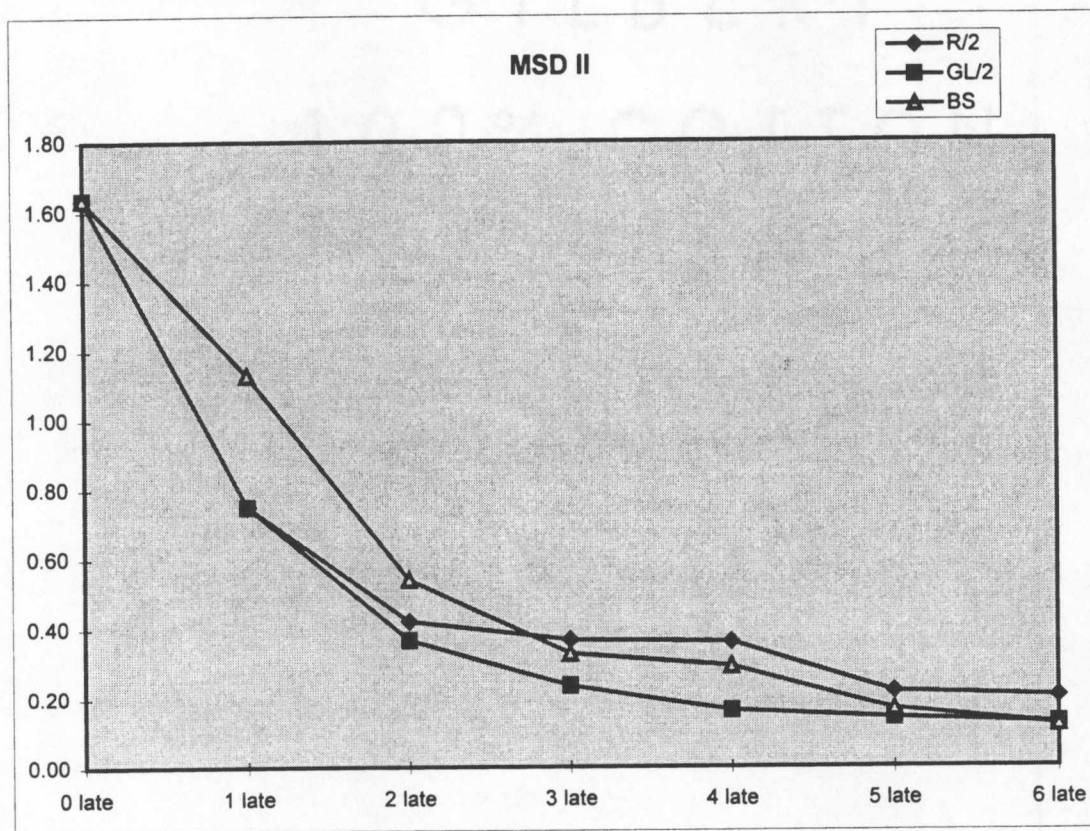


Figure 48: Graph of the Overall Mean Square Deviation for Demand Pattern II.

Table 75: Adjusted MSD Demand Pattern II.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late	6 late
Random-DD / 2-Opt	3.38	1.77	1.04	0.76	0.87	0.65	0.75
GC1LookAhead-DD/2-Opt	3.38	1.77	1.00	0.68	0.47	0.33	0.21
Border Swap	3.38	2.68	1.36	0.73	0.52	0.28	0.21

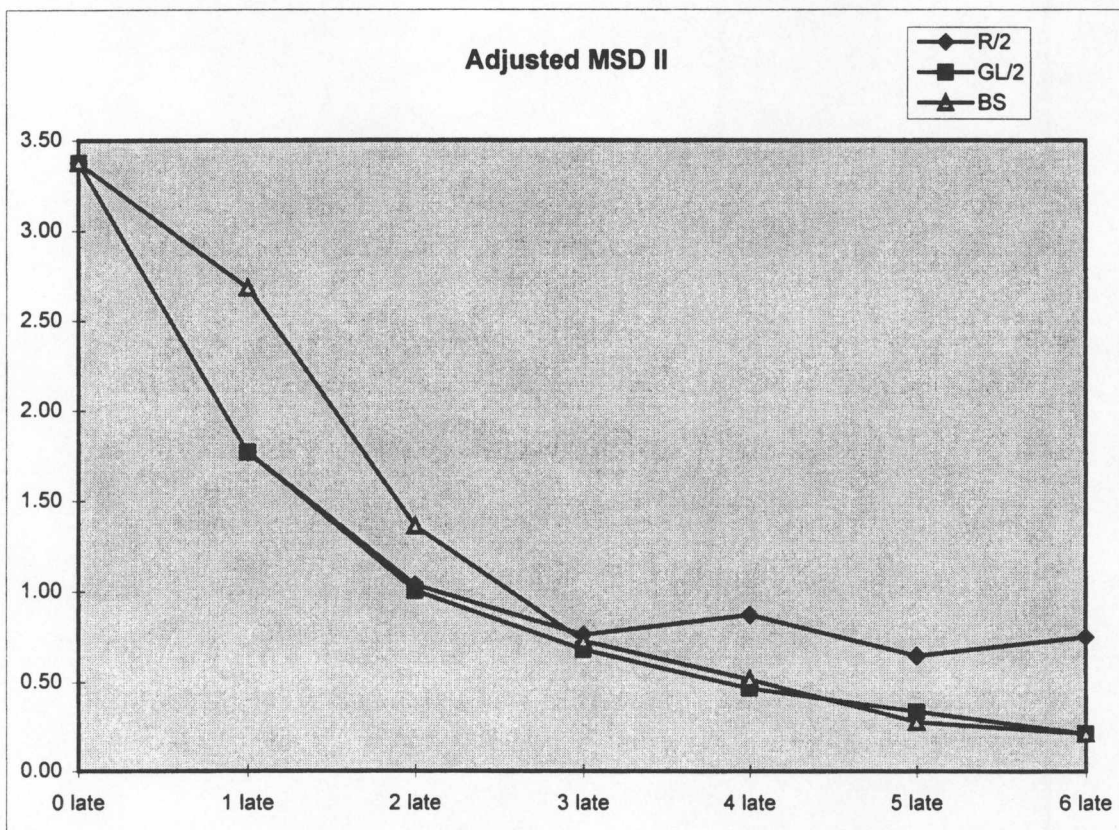


Figure 49: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern II.

Table 76: MSD Demand Pattern JJ.
Total Mean Square Deviation For X Items Late For Each Sequencing Method
Based On Items Balanced Over All Periods

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	1.49	0.72	0.57	0.27	0.27	0.20
GC1LookAhead-DD/2-Opt	1.49	0.74	0.34	0.19	0.15	0.12
Border Swap	1.49	0.92	0.36	0.20	0.15	0.12

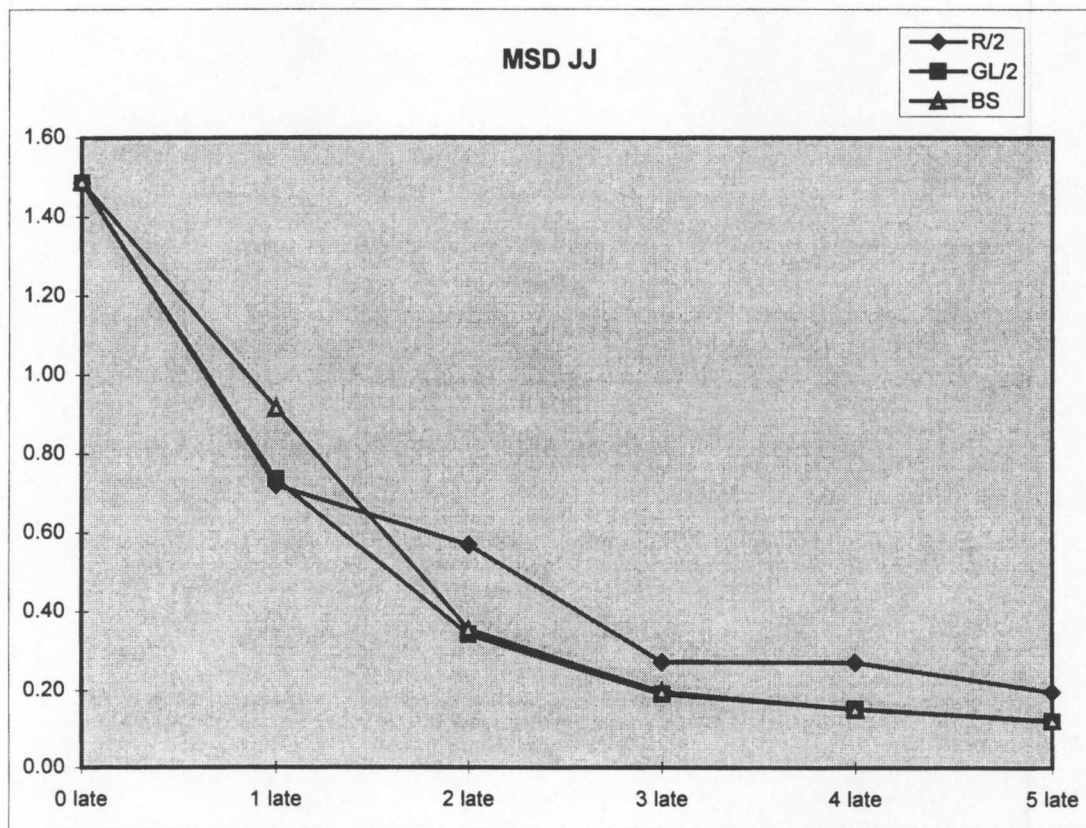


Figure 50: Graph of the Overall Mean Square Deviation for Demand Pattern JJ.

Table 77: Adjusted MSD Demand Pattern JJ.
Adjusted Total Mean Square Deviation for X Items Late for Each Sequencing Method
Based On Items Balanced Within Each Period

Sequencing Method	0 late	1 late	2 late	3 late	4 late	5 late
Random-DD / 2-Opt	3.07	1.60	1.13	0.70	0.76	0.51
GC1LookAhead-DD/2-Opt	3.07	1.98	1.02	0.59	0.33	0.18
Border Swap	3.07	2.22	1.07	0.46	0.27	0.21

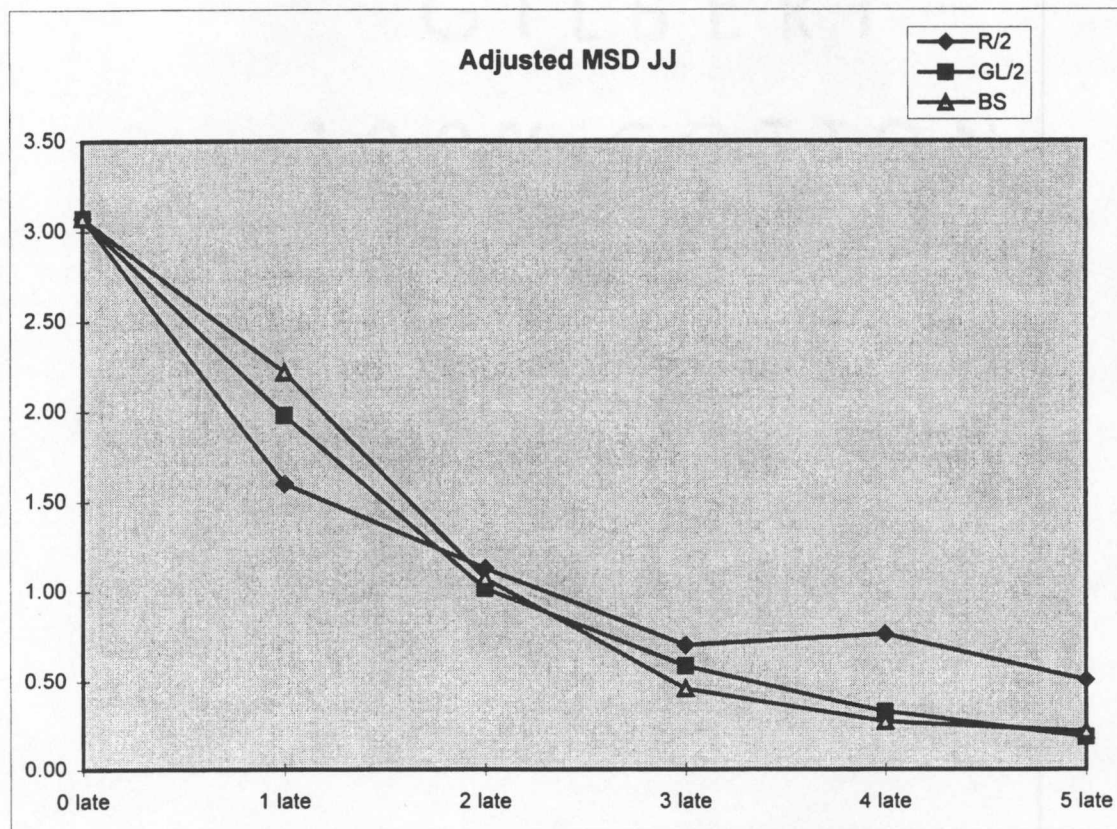


Figure 51: Graph of the Adjusted Overall Mean Square Deviation for Demand Pattern JJ.

CHAPTER 6

Analysis, Conclusions, and Future Research.

To better understand the results of the twenty problems, the individual heuristics were examined. Some of the noteworthy characteristics and patterns of performance are observed and detailed in this chapter. Section 1 describes trends associated with sequences based on Monden's GC1 with zero late items and a one period goal. Section 2 shows how Monden's GC1 with zero late items and a total overall goal were studied and lead to the development of the new Border Swap algorithm, which is examined in section 3. Section 4 gives an evaluation of the newly applied 2-Optimal heuristic. Section 5 gives a conclusion to the research and section 6 describes several future research possibilities.

1. Analysis of Monden's Goal Chasing 1 Method with Zero Late Items and a One Period Goal.

In our problems we sequenced 20 and 60 items respectively in a series of four due dates, or time buckets. In the first case, there were five items due in each time period. In the second problem set, there were fifteen items due in each time period. When no items are allowed to be sequenced late, the problem reduces to four individual problems. Monden's method is a greedy heuristic that was originally described for use mainly in only one time period (Monden, 1983). When used in a series of independent and dependent time buckets, as in this research, certain characteristics can be noted.

Figures 52 and 53 show the sum of the square deviations at each stage in the sequence when no items are allowed late for demand patterns AA and BB and all items are sequenced

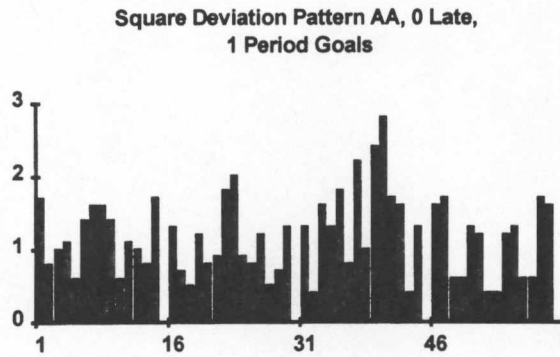


Figure 52: Square Deviation Pattern AA.

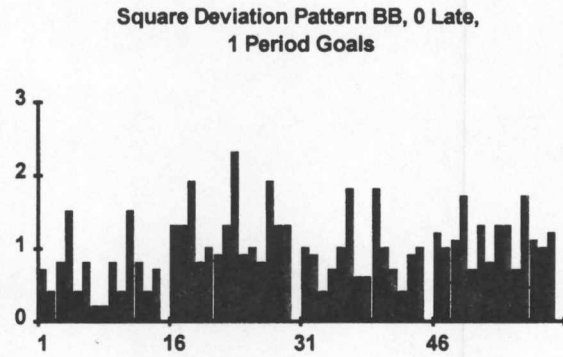


Figure 53: Square Deviation Pattern BB.

according to the associated one period goal. Each problem has 60 items in the sequence. The first item in each time period is marked on the scale; 16, 31, and 46 represent the divisions between time periods.

The sequence produced from the problem with demand pattern AA is as follows:

Period 1 (1 through 15): 2 5 1 5 2 5 3 4 2 5 2 5 1 5 2
 Period 2 (16 through 30): 3 5 2 3 5 3 1 5 2 3 5 3 2 5 3
 Period 3 (31 through 45): 5 2 5 1 2 5 2 5 4 3 5 2 5 2 5
 Period 4 (46 through 60): 2 1 5 3 1 5 2 1 2 5 1 3 5 1 2.

The sequence produced from the problem with demand pattern BB is as follows:

Period 1 (1 through 15): 4 1 4 3 4 4 1 4 1 4 4 3 4 1 4
 Period 2 (16 through 30): 1 5 4 3 1 4 2 1 5 4 1 3 4 5 1
 Period 3 (31 through 45): 3 4 3 1 3 4 3 2 3 4 3 1 3 4 3
 Period 4 (46 through 60): 4 3 4 5 1 4 2 4 3 4 1 5 4 3 4.

Characteristic 1: For each time period, the first half of the sequence is close to and in some cases exactly a mirror image of the second half. When there is a difference, it lies in the 'middle' of the sequence with models that have an odd number of items to be sequenced. Table

78 shows the mechanics of this characteristic. If the sequence is derived from a dual perspective, starting with all of the items in the sequence then eliminating one item at a time using the Goal Chasing method, it can be seen that the same solution would be the result. For instance if we begin with 9 part 1's, 7 part 2's, etc. at the end of period 1, then remove model 2 from the sequence the associated square deviation is 1.7. We are defining the same sequence as we would starting with no items in the sequence and continuing in a forward manner where the deviation is also 1.7.

Figures 54 and 56 show the actual usage of part 1 and part 2 respectively over the 60 item interval. In these graphs, the more evenly distributed the usage, the closer we are to meeting the level usage goal. Figures 55 and 57 show the cumulative usage of part 1 and part 2 respectively over the 60 item interval. In these graphs, the more smooth the ascent from zero to the part total, the closer we are to meeting the level part usage goal.

Table 78: Cumulative Part Totals for Pattern AA, Time Period 1.

Model #	Parts										Sq. Dev.
	1	2	3	4	5	6	7	8	9	10	
2	1	0	0	0	0	1	0	1	0	1	1.7
5	1	1	0	0	1	2	1	1	0	1	0.8
1	2	1	0	1	1	2	2	1	1	2	1.0
5	2	2	0	1	2	3	3	1	1	2	1.1
2	3	2	0	1	2	4	3	2	1	3	0.6
5	3	3	0	1	3	5	4	2	1	3	1.4
3	4	4	0	2	3	6	4	3	2	4	1.6
4	5	4	1	3	4	6	5	3	2	4	1.6
2	6	4	1	3	4	7	5	4	2	5	1.4
5	6	5	1	3	5	8	6	4	2	5	0.6
2	7	5	1	3	5	9	6	5	2	6	1.1
5	7	6	1	3	6	10	7	5	2	6	1.0
1	8	6	1	4	6	10	8	5	3	7	0.8
5	8	7	1	4	7	11	9	5	3	7	1.7
2	9	7	1	4	7	12	9	6	3	8	0

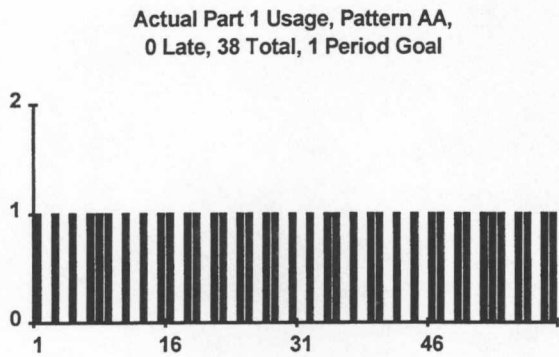


Figure 54: Actual Part 1 Usage.

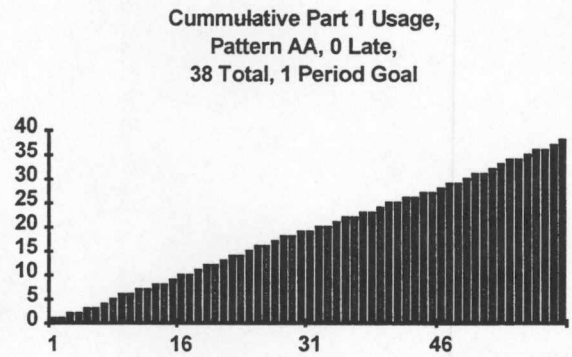


Figure 55: Cumulative Part 1 Usage.

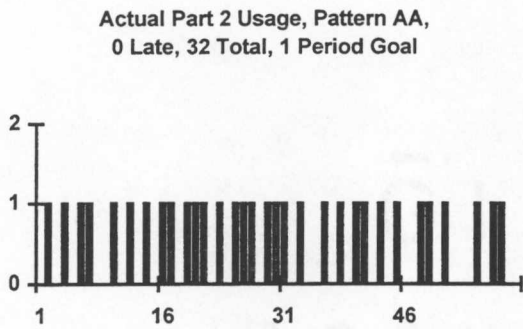


Figure 56: Actual Part 2 Usage.

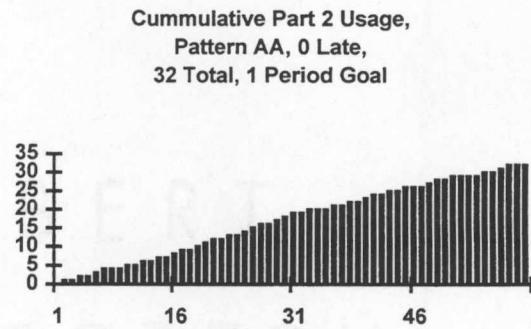


Figure 57: Cumulative Part 2 Usage.

Figures 58 through 61 show the deviation of parts 1 through 4 respectively for demand pattern AA sequenced with the goal of leveling part usage according to the parts in each time period only. On these graphs, deviations less than 0.5 represent the fractional difference between the goal and the integer actual. These deviations are not considered to be necessarily less than optimal.

Another important observation to note in interpreting the graphs is the relationship between the number of a particular part available for sequencing and the changes in the slope of the deviation. This can be seen easily in Figure 60, where there are only 2 part 3's available for sequencing. The first part 3 is sequenced in period 8. Prior to period 8 there is a negative

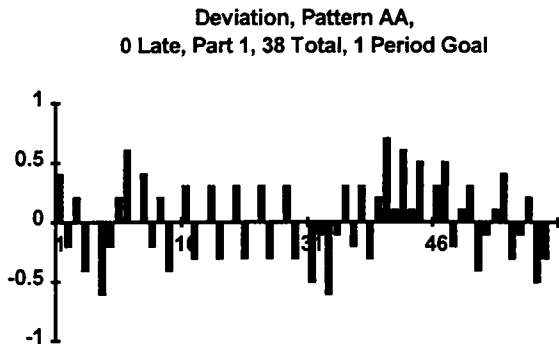


Figure 58: Deviation Part 1 Adjusted.

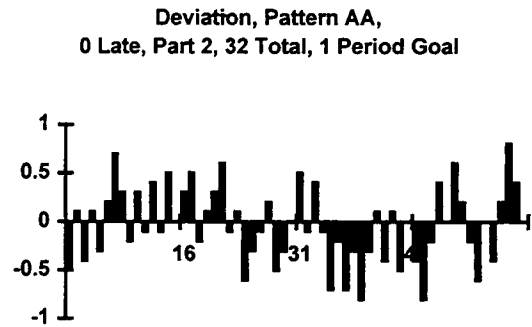


Figure 59: Deviation Part 2 Adjusted.

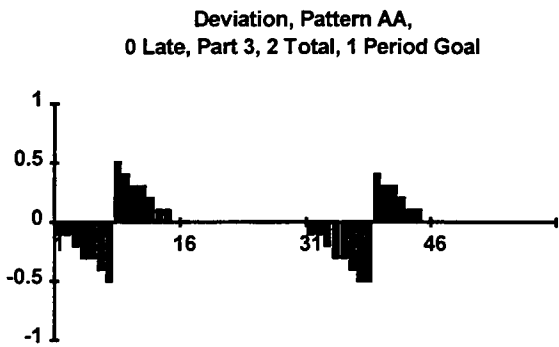


Figure 60: Deviation Part 3 Adjusted.

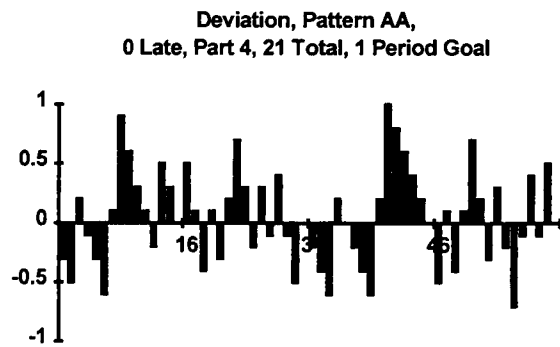


Figure 61: Deviation Part 4 Adjusted.

fractional deviation; when the product containing the part is sequenced, the deviation becomes positive. The same effect can be seen around period 39 when the second product containing part 3 is sequenced. Note that the deviation at the end of each time period is always zero due to the restriction of zero late items allowed in the sequence.

Figures 56 and 59 show that there are 32 part 2's available for sequencing. Thirty-two of the jumps in the figure show the changes in deviations associated with the sequencing of a part 2, and the other jumps show changes in the deviation due to no activity related to part 2. The sequence resulting from the heuristic method is repeated here again for convenience as follows:

Period 1: 2 5 1 5 2 5 3 4 2 5 2 5 1 5 2

Period 2: 3 5 2 3 5 3 1 5 2 3 5 3 2 5 3

Period 3: 5 2 5 1 2 5 2 5 4 3 5 2 5 2 5

Period 4: 2 1 5 3 1 5 2 1 2 5 1 3 5 1 2.

One unit of part 2 is required to produce one unit of models 3 and 5. Each time model 3 or model 5 are produced, the deviation associated with part 2 changes. The sequencing related jumps occur at periods 2, 4, 6, 7, 10, 12, 14, 16, 17, 19, 20, 21, 23, 25, 26, 27, 29, 30, 31, 33, 36, 38, 40, 41, 43, 45, 48, 49, 51, 55, 57, and 58. These are the periods where the products containing a part 1 were scheduled.

2. Analysis of Monden's Goal Chasing 1 Method with Zero Late Items According to the Total Period Goal.

This section analyzes Monden's GC1 as it was originally tested. The purpose of this investigation is to give insight into the development of the Border Swap heuristic.

Figures 62, 63, and 64 show the sum of the square deviations at each stage in the sequence when no items are allowed late for demand patterns AA, BB, and CC respectively and all items are sequenced according to the total overall goal which spans the four time periods. Each problem has 60 items in the sequence.

Characteristic 1: For each demand pattern the square deviation has a 'bridge' effect. The deviation is smaller in the middle of the time period, and larger at the beginning and end of each time period with the exception of stage 1 and stage 60, the beginning and end of the sequence.

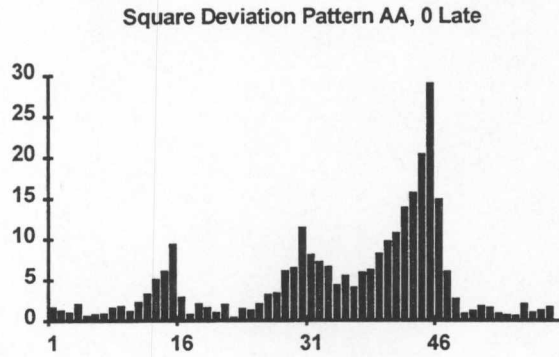


Figure 62: Square Deviation Pattern AA.

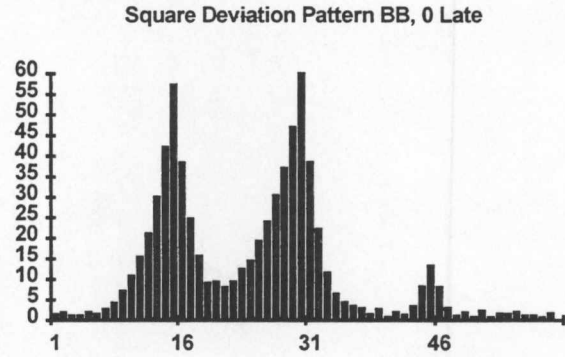


Figure 63: Square Deviation Pattern BB.

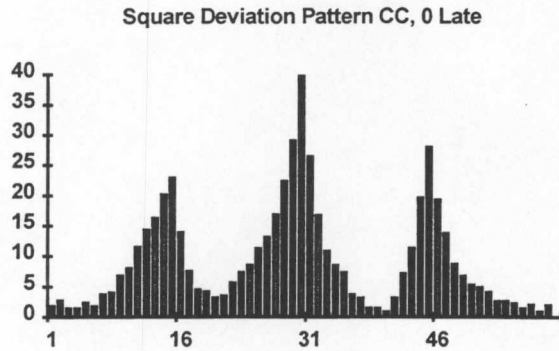


Figure 64: Square Deviation Pattern CC.

This effect can be explained by the nature of the GC1 sequencing method. The deviation at the end of each time period is due to the fact that no items are allowed to be sequenced late. If an item contains parts which have actual usage greater than the goal usage, then the product, if sequenced, will contribute a large amount of variation to the sequence. The product will not be chosen by the GC1 method early in the sequence due to the large deviation that it would contribute. The GC1 method is a greedy method designed to pick products which minimize deviation. The product will be sequenced at the end of the time period, thus the deviation is large at the ends of each time period.

At the beginning of a time period, the products are sequenced which contain parts that are under used to that point in time, that is, parts that need to 'catch up'. This can be seen in graphs of the actual deviations from the goal part usage rate for each of the ten parts in demand pattern AA. Graphs representing the deviations for parts 1 through 10 are shown in Figures 65 through 74 respectively.

Characteristic 2: The square deviation at the last stage of each time period will be the same regardless of the order of the sequence within the time period when there are no late items. The square deviation at a particular point in time is independent of the order of the items sequenced prior to that stage. For example, for part one, the deviation for each stage is

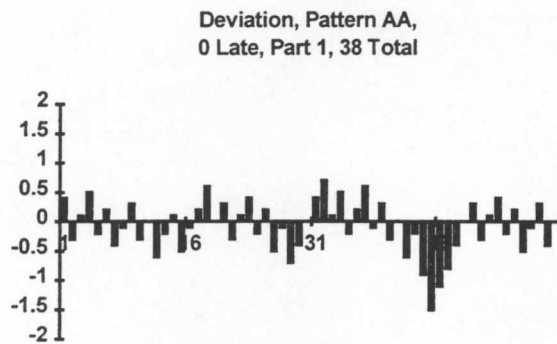


Figure 65: Deviation Part 1.

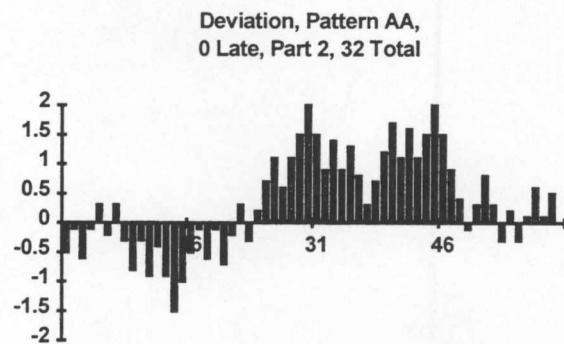


Figure 66: Deviation Part 2.

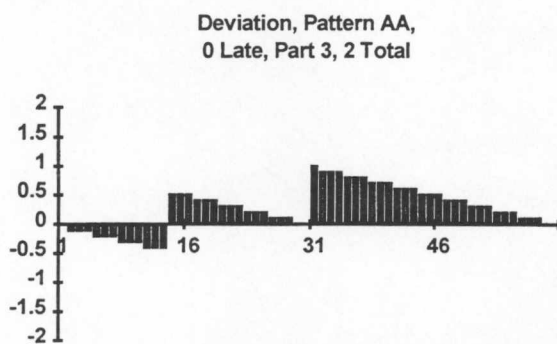


Figure 67: Deviation Part 3.

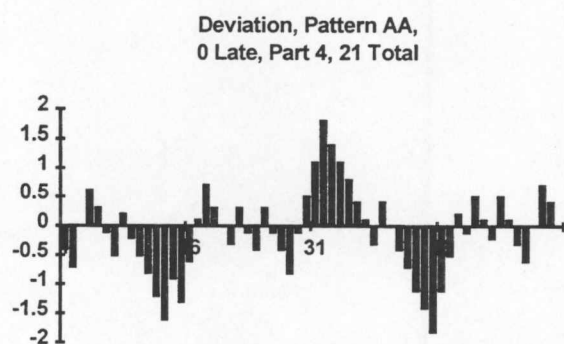


Figure 68: Deviation Part 4.

Deviation, Pattern AA,
0 Late, Part 5, 24 Total

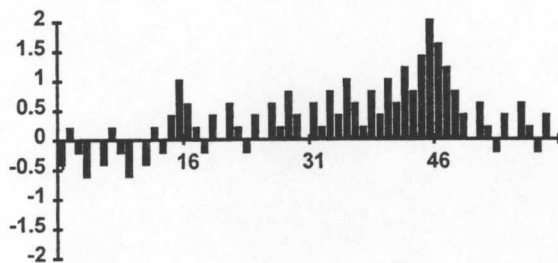


Figure 69: Deviation Part 5.

Deviation, Pattern AA,
0 Late, Part 6, 49 Total

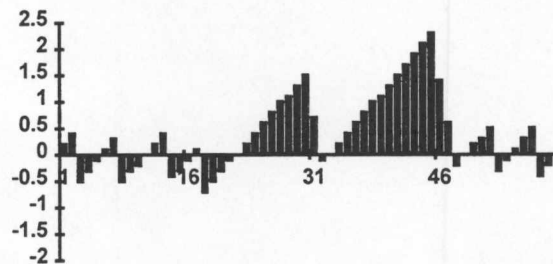


Figure 70: Deviation Part 6.

Deviation, Pattern AA,
0 Late, Part 7, 33 Total

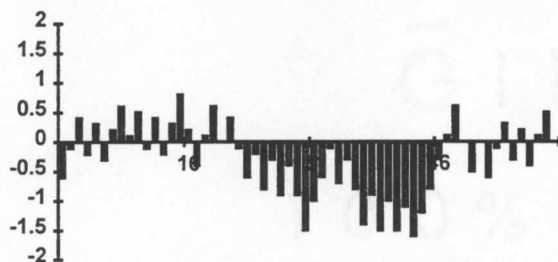


Figure 71: Deviation Part 7.

Deviation, Pattern AA,
0 Late, Part 8, 27 Total

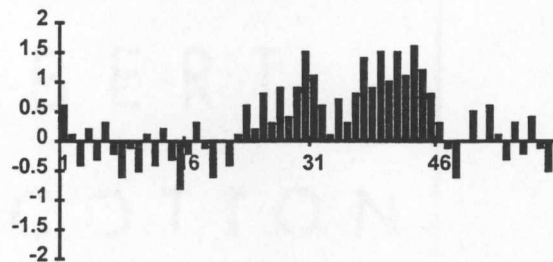


Figure 72: Deviation Part 8.

Deviation, Pattern AA,
0 Late, Part 9, 19 Total

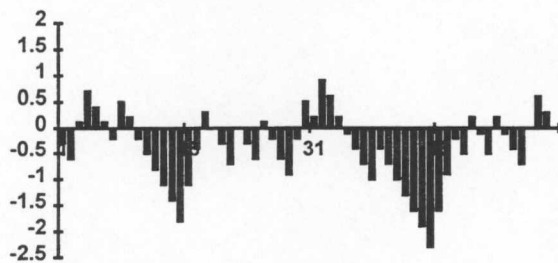


Figure 73: Deviation Part 9.

Deviation, Pattern AA,
0 Late, Part 10, 36 Total



Figure 74: Deviation Part 10.

calculated based on the goal for part 1 at that stage, and the actual total of part one at that stage.

The deviations for each part are summed over all of the parts. Simply stated, for each stage the square deviation is calculated as

$$\sum_{Parts} (Goal - Actual)^2,$$

where the 'Goal' for part j at each stage is calculated as follows:

$$(Stage \#) * \left(\frac{Total \# \text{ of Part } j's \text{ used in Entire Sequence}}{Total \# \text{ of Items in Entire Sequence}} \right).$$

The 'Actual' for part j at a given stage is the actual total number of part j 's used in the products sequenced up to that point in time, regardless of the order of how they were sequenced.

To better illustrate this point, three five item partial sequences follow in Table 79. The square error at stage 5 is 7.5 for all three sequences because these partial sequences contain the same items within the five stages. The first and second sequence contain the same items through stages 3 and 4 thus having identical deviation in those stages as well.

Table 79: Partial Sequences with Associated Square Errors.

Model	2	5	2	1	5
Square Error	1.6	1.2	4.8	2.2	7.5
Model	5	2	2	1	5
Square Error	2.8	3.2	4.8	2.2	7.5
Model	1	5	2	5	2
Square Error	2	6.8	3.4	4.8	7.5

Another example can be seen when comparing the resulting GC1 sequence with an alternate sequence for pattern AA. The sequencing method designed to sequence items based only on the associated individual time periods was tested. This method when evaluated according to the goal of leveling the part usage over all time periods does not show good results. It can be seen in Figure 75 that the last stage in each time period has the same variation as the last stage in each time period in Figure 76.

3. Evaluation of Border Swap Algorithm.

The first characteristic noted above concerning the 'bridge' effect of the square deviations for the GC1 method led to the simple idea that initiated the Border Swap heuristic. The Border Swap heuristic is based on looking at the highest peak in the square deviation sequence and reducing the variation associated with that peak. Figure 77 shows the greatest variation occurring at stages 45 and 46. To reduce the variation, the item sequenced at stage 45 is forced to be sequenced in the fourth time period, and the item sequenced at stage 46 is forced to be sequenced in the third time period. The resulting sequence has one item late and the resulting square deviation from the overall total time period goal is shown in Figure 78.

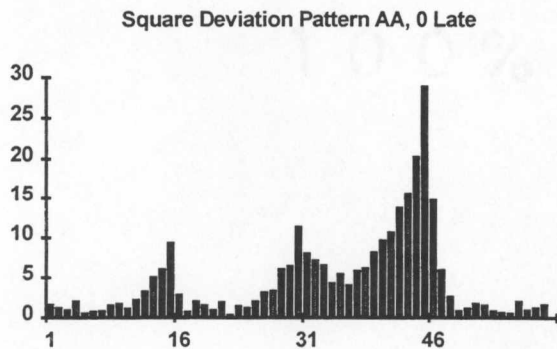


Figure 75: Square Deviation GC1.

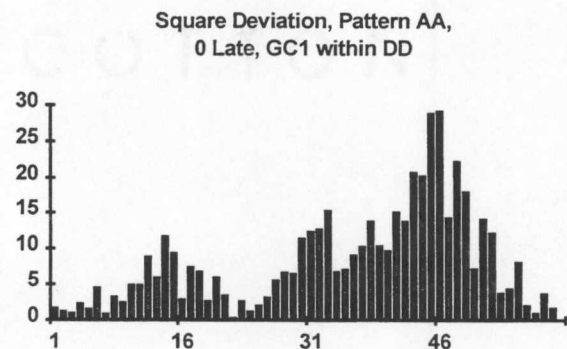


Figure 76: Sq. Dev., 1 Per. Goal,
Evaluated on Total Goal.

The next variation to reduce occurs at stages 30 and 31. The results of the improvement are shown in Figure 79. This process is continued with results shown in Figures 80 through 85. Using this method for sequencing 1 late, 2 late, etc., each time the two items to 'switch' are determined, the new sequence with the specified item late and the specified item early should be recalculated based on the single period goal. In this example, only the total period goal deviations are shown. In this example the Border Swap heuristic provides the same final solution as does the GC1 method with no due date restrictions which is shown in Figure 86.

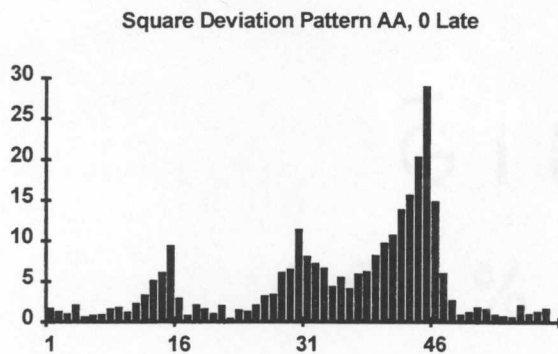


Figure 77: GC1, Pattern AA.

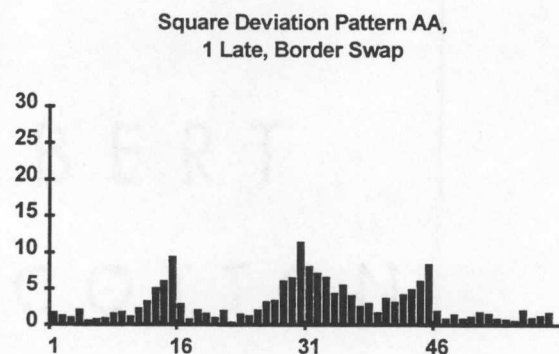


Figure 78: Border Swap, 1 Late.

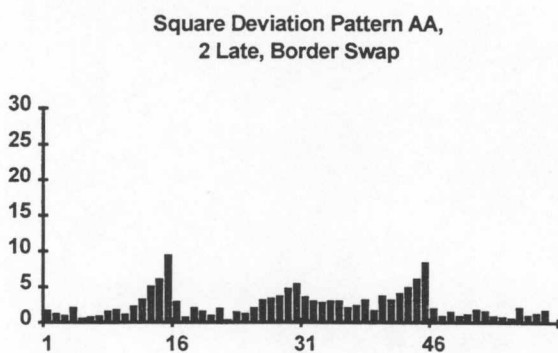


Figure 79: Border Swap, 2 Late.

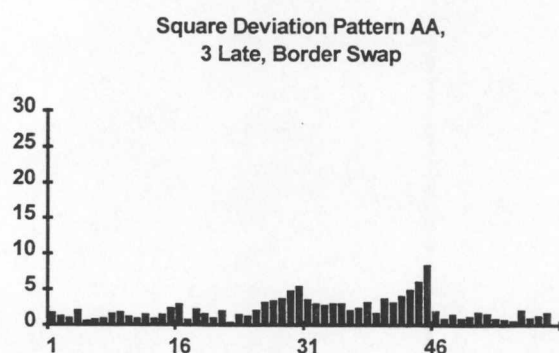


Figure 80: Border Swap, 3 Late.

Square Deviation Pattern AA,
4 Late, Border Swap

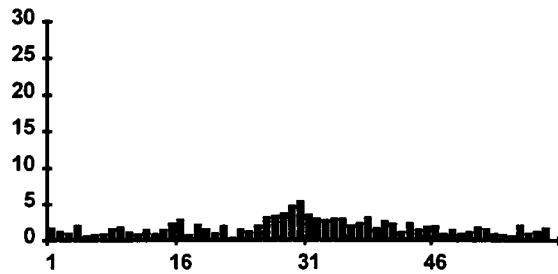


Figure 81: Border Swap, 4 Late.

Square Deviation Pattern AA,
3 Late, Border Swap

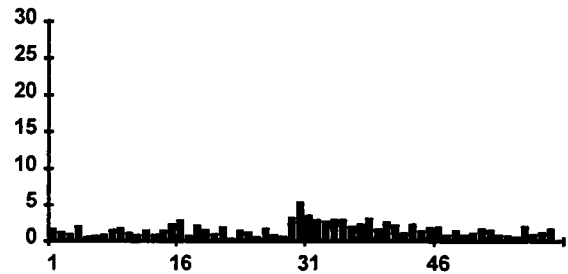


Figure 82: Border Swap, 3 Late Repeat.

Square Deviation Pattern AA,
5 Late, Border Swap

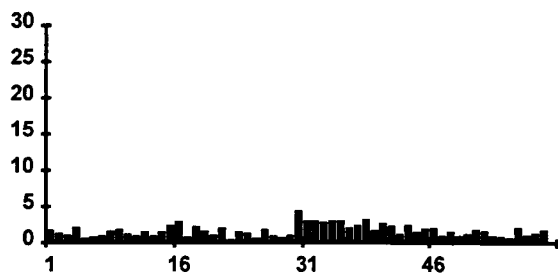


Figure 83: Border Swap, 5 Late.

Square Deviation Pattern AA,
5 Late, Border Swap

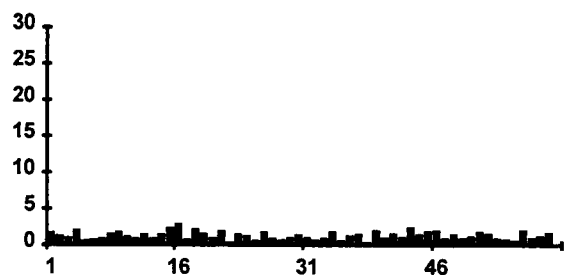


Figure 84: Border Swap, 5 Late Repeat.

Square Deviation Pattern AA,
6 Late, Border Swap

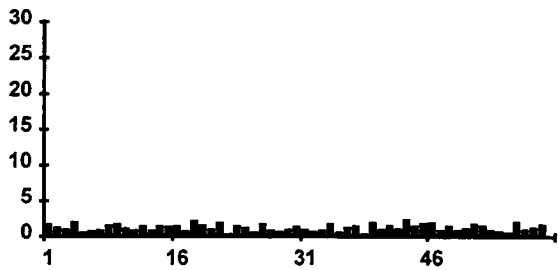


Figure 85: Border Swap, 6 Late.

Square Deviation Pattern AA,
Any Late

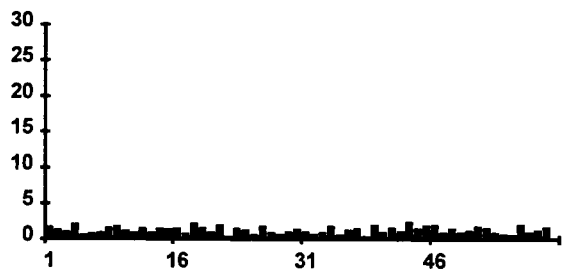


Figure 86: GC1-No DD Restrictions.

4. Evaluation of the Two Optimal Heuristic.

For some problems it is thought that the two-optimal procedure performs best when the initial starting solution is a randomly generated solution to the problem at hand. This hypothesis does not apply to a sequencing problem in which the number of late items is restricted. For the case where there are only a small number of items allowed to be late, just the opposite was found for the problems tested. For the data tested in this research, the comparison of the results of two different starting solutions shows that in almost every case, the better the starting solution, the better the results of the two optimal procedure.

The Goal Chasing 1 Look Ahead method was used to generate a solution to each problem for the case of zero items late. This solution was then used as the initial sequence for the two optimal algorithm. Various sequences were generated with zero items late, one item late, two items late, etc., up to a sequence with no restriction on the number of late items. The same procedure was performed with a sorted initial solution.

When zero items were allowed to be sequenced late, the two methods produced identical results for problems of size 20. For problems with 60 total items, the GC1-LA initial sequence produced better results for four of the demand patterns in terms of the overall goal, and equal results to the sorted sequence for the remaining six demand patterns. In terms of the single period goals, the two methods performed equally well. For almost all of the remaining problems with 60 items and various numbers of items allowed late, the GC1-LA initial sequence produced the best results for each demand pattern.

It can be seen in the following example why the GC1 initial sequence, seems to perform better than the sorted initial sequence when the number of items late is restricted. Tables 80, 81,

Table 80: Series of Sequences for Demand Pattern D Using the GC1 / 2opt Method with One Item Allowed Late.

# Swaps	1-5	6-10	11-15	16-20	Sq. Error	# Late
0	1, 3, 1, <u>3</u> , 3	4, 1, 1, 3, 1	5, 1, 3, 1, 3	5, <u>5</u> , 1, 2, 1	103.35	0
1	1, 3, 1, 5, 3	4, 1, 1, 3, 1	5, <u>1</u> , 3, 1, <u>3</u>	5, 3, 1, 2, 1	44.35	1
2	1, 3, 1, 5, 3	4, 1, <u>1</u> , <u>3</u> , 1	5, 3, 3, 1, 1	5, 3, 1, 2, 1	39.65	1
3	1, 3, 1, 5, 3	4, 1, 3, 1, 1	5, 3, 3, 1, 1	5, <u>3</u> , 1, <u>2</u> , 1	37.25	1
4	1, 3, 1, 5, 3	4, 1, 3, 1, 1	5, 3, 3, 1, 1	<u>5</u> , <u>2</u> , 1, 3, 1	35.25	1
5	1, 3, 1, 5, 3	<u>4</u> , <u>1</u> , 3, 1, 1	5, 3, 3, 1, 1	2, 5, 1, 3, 1	32.85	1
6	1, 3, 1, 5, 3	1, 4, 3, 1, 1	5, 3, <u>3</u> , <u>1</u> , 1	2, 5, 1, 3, 1	31.25	1
7	1, 3, 1, 5, 3	1, 4, 3, 1, 1	5, 3, 1, 3, 1	2, <u>5</u> , <u>1</u> , 3, 1	30.15	1
8	1, 3, <u>1</u> , <u>5</u> , 3	1, 4, 3, 1, 1	5, 3, 1, 3, 1	2, 1, 5, 3, 1	29.45	1
9	1, 3, 5, 1, 3	1, 4, 3, 1, 1	<u>5</u> , <u>3</u> , 1, 3, 1	2, 1, 5, 3, 1	28.75	1
10	1, 3, 5, 1, 3	1, 4, 3, 1, 1	3, 5, 1, 3, 1	2, 1, 5, 3, 1	28.35	1

Table 81: Series of Sequences for Demand Pattern D Using the GC1 / 2opt Method with Two Items Allowed Late.

# Swaps	1-5	6-10	11-15	16-20	Sq. Error	# Late
0	1, 3, 1, <u>3</u> , 3	4, 1, 1, 3, 1	5, 1, 3, 1, 3	5, <u>5</u> , 1, 2, 1	103.35	0
1	1, 3, 1, 5, 3	4, 1, <u>1</u> , 3, 1	5, 1, 3, 1, <u>3</u>	5, 3, 1, 2, 1	44.35	1
2	1, 3, 1, 5, 3	4, 1, 3, <u>3</u> , <u>1</u>	5, 1, 3, 1, 1	5, 3, 1, 2, 1	38.25	2
3	1, 3, 1, 5, 3	4, 1, 3, 1, 3	5, 1, 3, 1, 1	5, <u>3</u> , 1, <u>2</u> , 1	35.95	2
4	1, 3, 1, 5, 3	4, 1, 3, 1, 3	5, 1, 3, 1, 1	<u>5</u> , <u>2</u> , 1, 3, 1	33.95	2
5	1, 3, 1, 5, 3	<u>4</u> , <u>1</u> , 3, 1, 3	5, 1, 3, 1, 1	2, 5, 1, 3, 1	31.55	2
6	1, 3, 1, 5, 3	1, 4, 3, 1, 3	5, 1, <u>3</u> , <u>1</u> , 1	2, 5, 1, 3, 1	29.95	2
7	1, 3, 1, 5, 3	1, 4, 3, 1, 3	5, 1, 1, 3, 1	2, <u>5</u> , <u>1</u> , 3, 1	28.85	2
8	1, 3, <u>1</u> , <u>5</u> , 3	1, 4, 3, 1, 3	5, 1, 1, 3, 1	2, 1, 5, 3, 1	28.15	2
9	1, 3, 5, 1, 3	1, 4, 3, 1, 3	<u>5</u> , <u>1</u> , 1, 3, 1	2, 1, 5, 3, 1	27.45	2
10	1, 3, 5, 1, 3	1, 4, 3, 1, 3	1, 5, 1, 3, 1	2, 1, 5, 3, 1	27.35	2

82, and 83 show sequencing results for demand pattern D with twenty total items using the two optimal sequencing method. Tables 80 and 81 show the results when the Goal Chasing 1 method is used to generate the initial starting sequence. Tables 82 and 83 show the results when a sorted sequence that is random with respect to part usage is used to generate the initial starting sequence.

When one item is allowed to be sequenced late, as in Table 80, the due date borders do not affect the selection of the first two items to exchange. From the initial sequence, the fourth item, model 3 and the seventeenth item, model 5, were selected by the two optimal procedure to exchange. After this first swap, there is one item sequenced late and the total square error is reduced from 103.35 to 44.35, the largest single reduction in the whole series of swaps. The second step does not consider any exchanges that would lead to an additional late item. The second swap is between the twelfth and the fifteenth items, models 1 and 3. This exchange reduces the total square deviation from 44.35 to 39.65. From this point to the end, no exchanges are made across due date borders as was done in the first step. The borders imposed on the sequencing problem have greatly influenced the resulting solution.

In Table 81 when two items are allowed to be sequenced late. The first step is identical to Table 80, but the second step is now not restricted by the borders. The eighth and fifteenth items, models 1 and 3, are exchanged to reduce the total square deviation to 38.25. All of the remaining exchanges occur within due date borders.

Since the first swap is not restricted by the borders, it seems to have the largest potential for decreasing the total square error. If the initial sequence is structured to be a good starting sequence with respect to having a small square error, then the first swap is a 'well thought out'

Table 82: Series of Sequences for Demand Pattern D Using the Sort / 2opt Method with One Item Allowed Late.

# Swaps	1-5	6-10	11-15	16-20	Sq. Error	# Late
0	1, <u>1</u> , 3, 3, 3	1, 1, 1, 3, 4	1, 1, 3, 3, 5	1, 1, 2, 5, <u>5</u>		0
1	1, 5, 3, 3, 3	1, 1, 1, 3, 4	1, 1, <u>3</u> , 3, <u>5</u>	1, 1, 2, 5, 1	78.75	1
2	1, 5, 3, 3, 3	1, 1, 1, <u>3</u> , <u>4</u>	1, 1, 5, 3, 3	1, 1, 2, 5, 1	64.55	1
3	1, 5, 3, 3, 3	1, 1, 1, 4, 3	1, <u>1</u> , <u>5</u> , 3, 3	1, 1, 2, 5, 1	59.65	1
4	1, <u>5</u> , <u>3</u> , 3, 3	1, 1, 1, 4, 3	1, 5, 1, 3, 3	1, 1, 2, 5, 1	55.85	1
5	1, 3, 5, 3, 3	1, 1, <u>1</u> , <u>4</u> , 3	1, 5, 1, 3, 3	1, 1, 2, 5, 1	54.05	1
6	1, 3, 5, 3, 3	1, 1, 4, 1, 3	1, 5, 1, 3, 3	1, <u>1</u> , <u>2</u> , 5, 1	52.85	1
7	1, 3, 5, 3, 3	1, 1, 4, 1, 3	1, 5, 1, 3, 3	1, <u>2</u> , 1, <u>5</u> , 1	51.75	1
8	1, 3, 5, 3, 3	1, 1, 4, 1, 3	1, 5, 1, 3, 3	1, 5, 1, 2, 1	50.75	1

Table 83: Series of Sequences for Demand Pattern D Using the Sort / 2opt Method with Two Items Allowed Late.

# Swaps	1-5	6-10	11-15	16-20	Sq. Error	# Late
0	1, <u>1</u> , 3, 3, 3	1, 1, 1, 3, 4	1, 1, 3, 3, 5	1, 1, 2, 5, <u>5</u>		0
1	1, 5, 3, 3, <u>3</u>	1, 1, 1, 3, 4	1, 1, 3, 3, 5	1, 1, <u>2</u> , 5, 1	78.75	1
2	1, 5, 3, 3, 2	1, 1, 1, 3, 4	1, 1, <u>3</u> , 3, <u>5</u>	1, 1, 3, 5, 1	51.75	2
3	1, 5, 3, 3, 2	1, 1, 1, 3, 4	1, <u>1</u> , 5, <u>3</u> , 3	1, 1, 3, 5, 1	45.55	2
4	1, <u>5</u> , <u>3</u> , 3, 2	1, 1, 1, 3, 4	1, 3, 5, 1, 3	1, 1, 3, 5, 1	43.05	2
5	1, 3, 5, 3, 2	1, 1, 1, 3, 4	1, 3, 5, 1, 3	1, 1, <u>3</u> , <u>5</u> , 1	41.25	2
6	1, 3, 5, 3, 2	1, 1, 1, <u>3</u> , <u>4</u>	1, 3, 5, 1, 3	1, 1, 5, 3, 1	39.45	2
7	1, 3, 5, 3, 2	1, 1, 1, 4, 3	1, <u>3</u> , <u>5</u> , 1, 3	1, 1, 5, 3, 1	38.55	2
8	1, 3, 5, 3, 2	1, 1, 1, 4, 3	1, 5, 3, 1, 3	1, 1, 5, 3, 1	38.35	2

contributor to the reduction in square error. If, on the other hand, the initial sequence has a relatively large square error, as with the sorted solution, then situation is different. The first swap may reduce the square error, but it may not be the best choice in the long run for the exchange between borders, since the remaining swaps will more than likely not cross any of the border lines. Tables 82 and 83 show that the early exchange between borders can limit the problem with respect to the amount of possible reduction in the total square error.

5. Conclusion

Mixed-model sequencing is a method used on a final assembly line in place of batch production methods when a variety of items are to be produced. With several varieties of companies choosing to build-to-order rather than build-to-stock this topic is a very useful. It can be integrated into future and existing sequencing practices. Until now, there has not been published research combining two of the four main goals in sequencing: meet the customer due date, and level the pull of components through the system. There has been work done with respect to these goals individually and a third goal of leveling the workload at each workstation, but there has been no mention of the fourth main goal: minimize the production time spread of a customer's order, which this research introduced.

The focus of this research was to explore the combination of the first two goals. Through this exploration several contributions were made. A new definition of the goal used in a simple greedy algorithm was examined. Several low-order polynomial heuristics for solving a Mixed-Model sequencing problem were introduced. An evaluative methodology of the multi-objective environment was established. A branch-and-bound procedure was developed that exploits the embedded assignment problem constraints of assigning each item to a stage, and

assigning each stage an item.

From a practical standpoint, the dissertation provided an improvement to the well known Toyota sequencing method by Monden. Information was also provided concerning the tradeoffs between two of the four objectives of sequencing the final assembly line in the build-to-order environment.

6. Future Research

Research is needed which can give enlightenment to schedulers on some of these key areas of production decision making. This section contains two types of research topics. This first set of topics are direct extensions of this dissertation, while the second set of topics are broad extensions and general questions involving mixed-model sequencing.

1. Direct Extensions Of This Dissertation

1. Use simulation to evaluate the performance of the proposed heuristics with respect to the four sequencing goals.
2. Include a gradient check routine in the algorithms which will automatically determine when to stop the sequencing routine based on a decrease in the benefit of allowing one more late item.
3. Tighten the bounds in the branch-and-bound routine by considering the impact of multiple products of each product type.
4. Consider assigning weights to each model.
5. Update the goal at each stage of production based on a newly calculated goal which takes into consideration the sequence up to that point.

6. Use genetic algorithms to determine sequences based of the four main scheduling goals.

2. Broad Extensions And General Questions

1. How can the goal of minimizing the production time spread of an order be incorporated in a sequencing algorithm?
2. How can the tradeoffs between the four sequencing goals be evaluated?
3. When should a company use mixed-models?
4. How 'mixed' should the models be in the case where there is some advantage to batch production?
5. What are the advantages, disadvantages of using mixed-models in different assembly operations with various types of product lines?
6. How does a company get started changing from batch production to mixed-model production?
7. How often should a master schedule be revised in order to optimize the benefits of mixed-model sequencing?
8. Over what time span should the production schedule run to best optimize the sequencing goals?
9. What percentage of the due dates should be met?
10. What volume of demand will be considered "average" and what percentage above and below the average demand will the production line accommodate? That is, what will be the production capacity?
11. What is the break-even point for applying mixed-model scheduling in the build-to-order environment? If there is not a reasonable mix within a specific order, then a company could combine two or more orders and then define the mix. In this situation, the finished product

from the orders with the latter due date will have to be held in finished goods inventory.

How can a company tell how much finished goods inventory to store and what percent of the 'ideal' model-mix to meet?

12. Is it possible to directly apply some of the existing mixed-model sequencing strategies in the build-to-order environment?

13. Can a policy be formulated that describes how often a schedule should be updated? If a schedule is determined for a five day period, then what is the effect of updating the schedule two days into it? Are the benefits of mixed-model scheduling lost?

14. How can mixed-model scheduling be easily implemented in the build-to-order environment? Are expert systems a good choice?

BIBLIOGRAPHY

BIBLIOGRAPHY

- Bolat, A., "Generalized Mixed Model Assembly Line Sequencing Problem," Unpublished Ph.D. dissertation, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, MI, (1988).
- Dar-El, E. M., and Cothier, R. F., "Assembly Line Sequencing for Model Mixing," *International Journal of Production Research*, Vol. 13, No. 5, (1975) 463-477.
- Dar-El, E. M., and Cucuy, S., "Optimal Mixed-Model Sequencing for Balanced Assembly Lines," *OMEGA*, Vol. 5, No. 3, (1977) 333-342.
- Ding, F.-Y., and Cheng, L., "An Effective Mixed-Model Assembly Line Sequencing Heuristic for Just-In-Time Production Systems," *Journal of Operations Management*, Vol. 11, No. 1, (1993) 45-50.
- Gilbert, K., Holmes, D. D., and Rosenthal, R. E. A multiobjective discrete optimization model for land allocation. *Management Science*, Vol. 31, No. 12, (1985), 1509-1522.
- Hall, R. W., *Zero Inventories*, Dow Jones-Irwin, Homewood, Illinois (1983).
- Inman, R. R., and Bulfin, R. L., "Sequencing JIT Mixed-Model Assembly Lines," *Management Science*, Vol. 37, No. 7, (1991) 901-904.
- Japanese Management Association, *Kanban: Just-In-Time at Toyota*, Productivity Press, Stamford, Connecticut (1985).
- Kilbridge, M., and Webster, L., "The Assembly Line Model-Mix Sequencing Problem," *Proceedings of the Third International Conference on Operational Research*, Oslo, Paris: English Universities Press, (1963) 247.
- Kim, M. H., and Kim, Y. D., "Simulation-Based Real-Time Scheduling in a Flexible Manufacturing System," *Journal of Manufacturing Systems*, Vol. 13, No. 2, (1994) 85-93.
- Kubiak, W. and S. Sethi, "A Note on Level Schedules for Mixed-Model Assembly Lines in Just-In-Time Production Systems", *Management Science*, Vol. 37, No. 1, (1991).
- Macaskill, J. L., "Production-Line Balances for Mixed-Model Lines," *Management Science*, Vol. 19, No. 4, (1972) 423-433.
- Macaskill, J. L., "Computer Simulation for Mixed-Model Production Lines," *Management Science*, Vol. 20, (1973) 341.
- McCormick, S. T., Pinedo, M. L., Shenker, S., and Wolf, B., "Sequencing in an Assembly Line with Blocking to Minimize Cycle Time," *Operations Research*, Vol. 37, No. 6, (1989) 925-935.

- Miltenburg, G. J., "Level Schedules For Mixed-Model Assembly Lines in Just-In-Time Production Systems," *Management Science*, Vol. 35, No. 2, (1989) 192-207.
- Miltenburg, G. J., and Sinnamon, G., "Scheduling Mixed-Model Multi-Level Just-In-Time Production Systems," *International Journal of Production Research*, Vol. 27, No. 9, (1989) 1487-1509.
- Miltenburg, G. J., and Sinnamon, G., "Algorithms for Scheduling Multi-Level Just-In-Time Production Systems," *IIE Transactions*, Vol. 24, No. 2, (1992) 121.
- Monden, Y., *Toyota Production System*, Institute of Industrial Engineers Press, Norcross, Georgia, USA (1983).
- Okamura, K., and Yamashina, H., "A Heuristic Algorithm for the Assembly-Line Model-Mix Sequencing Problem to Minimize the Risk of Stopping the Conveyor," *International Journal of Production Research*, Vol. 17, No. 3, (1979) 233.
- Parrello, B., "Car Wars: The (Almost) Birth of an Expert System," *AI Expert*, (1988), 60-64.
- Raju, K. R., and Chetty, O. V. K., "Priority Nets for Scheduling Flexible Manufacturing Systems," *Journal of Manufacturing Systems*, Vol. 12, No. 4, (1993) 326-340.
- Rosenthal, R. E., "Principles of Multiobjective Optimization," *Decision Sci.*, Vol. 16, (1985), 133-152.
- Steiner George., and Yeomans, Scott, "Level Schedules for Mixed-Model Just-In-Time Processes," *Management Science*, Vol. 39, No. 6, (1993) 728-735.
- Sumichrast, R. T., Russell, R. S., and Taylor, B. W. "A Comparative Analysis of Sequencing Procedures for Mixed-Model Assembly Lines in a Just-In-Time Production System," *International Journal of Production Research*, Vol. 30, No. 1, (1992) 199-214.
- Sumichrast, R. T., and Russell, R. S, "Evaluating Mixed-Model Assembly Line Sequencing Heuristics for Just-in-Time Production Systems," *Journal of Operations Management*, Vol. 9, No. 3, (1990) 371-388.
- Thompoulos, N. T., "Mixed-Model Line Balancing with Smoothed Station Assignments," *Management Science*, Vol. 16, (1970) 593.
- Yano, C. A. and Rachamadugu, R., "Sequencing to Minimize Work Overload in Assembly Lines with Product Options," *Management Science*, Vol. 37, No. 5, (1991) 572-586.

APPENDICES

APPENDIX A

Steps of the Branch-And-Bound Algorithm

Step 1: Sort items in order by due date, then by model type within each due date period.

Step 2: Initialize the variables and set the total square error = 0, the total early and total late = 0, the best maximum square error = a very large number which is initially related to the incumbent, and the best square error total = the best square error from a previously administered heuristic algorithm. Set $k=1$ as the first item in the sorted list, and $t=1$ as the first stage in the sequence.

Step 3: Calculate the total rounding error for each stage and the cumulative rounding error at each stage. The rounding error at stage t is the minimum of (the fractional portion of the goal at stage t) and $(1 - \text{the fractional portion of the goal at stage } t)$. The cumulative rounding error at stage $t = \text{the cumulative rounding error at stage } (t-1) - \text{the rounding error of stage } (t-1)$.

Step 4: Temporarily sequence item k at stage t if one of the following is true.

- (1) There are zero items allowed late and the due date of item k matches the production date associated with stage t .
- (2) The number of items allowed late is $\leq X$ and the number of late items in the present sequence up to stage t is $< X$, and the number of early items is $\leq X$.

(3) The number of items allowed late is $\leq X$ and the number of late items in the present sequence up to stage t is $= X$, and the due date of item k is equal to the production date associated with stage t .

(4) There are any number of items allowed late in the sequence with no restrictions.

Step 5: Calculate the error associated with sequencing item k at stage t . If the error associated with k + the total error through stage $t - 1$ + the total remaining cumulative rounding error is $\leq$ the best maximum square error then sequence item k at stage t .

Step 6: Set $k = k + 1$. If item k has been sequenced then repeat this step. If $k \leq$ the total number of items then go to Step 4. If $k =$ total number of items + 1 then go to Step 8.

Step 7: Backtrack and remove the last item sequenced. Set $k = 1$.

Step 8: Stop, all items have been sequenced. The sequence with the best square error total is the optimal sequence with respect to the two goals of meeting the due date requirements and leveling component usage based on the overall period goal.

APPENDIX B

Example of the Border Swap Heuristic

Input File:

```
1 3 3 2 1 2 3 2 2 1
1 1 1 2 1 2 3 2 2 2 1 1 1 2 3 1 2 1 2 2
```

```
1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0
0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0
```

{NUMPARTS 19, NUMMODELS 10, TOTALITEMS 20, DAILY PRODUCTION 9}

Using the data presented above, the following sequence is produced from step 2:

0 Items Late

model	Period 1									Period 2									Period 3		
7	8	2	3	9	6	1	2	6	5	4	7	3	10	4	9	2	8	3	7		
1	2	2	3	3	3	4	6	15	10	7	5	4	2	3	3	3	3	1	0		

Squared Error

Total Squared Error: 89.3

Total Absolute Deviation: 134.4

Overall MSE: 0.236

Overall MAD: 0.354

The border candidates are models 6 and 5, and models 8 and 3. The 'highest MSE border pair' is 6 and 5 with a total MSE of 25 (15+10). The next schedule produced will allow model 6 to be produced in period 2 and model 5 to be produced in period 1.

1 Item Late (in bold)

model	Period 1										Period 2					Period 3				
7	8	2	3	9	6	5	1	2	4	10	7	3	8	2	4	9	6	3	7	
1	2	2	3	3	3	2	2	3	2	2	1	1	2	1	2	2	3	1	0	
Squared Error																				

Squared Error

Total Squared Error: 45.5

Total Absolute Deviation: 100.1

Overall MSE: 0.120

Overall MAD: 0.263

The border candidates are models 2 and 4, and models 6 and 3. The 'highest MSE border pair' is 2 and 4 with a total MSE of 5 (3+2). The next schedule produced will allow model 2 to be produced in period 2 and model 4 to be produced in period 1.

2 Items Late

model		Period 1										Period 2										Period 3		
7	8	2	3	4	9	6	5	1	2	10	7	3	8	2	4	9	6	3	7					
1	2	2	3	3	2	1	2	3	2	2	1	1	2	1	2	2	3	1	0					
Squared Error																								

Squared Error

Total Squared Error: 42.7

Total Absolute Deviation: 97.3

Overall MSE: 0.113

Overall MAD: 0.256

The border candidates are models 1 and 2, and 6 and 3. The 'highest MSE border pair' is 1 and 2 with a total MSE of 5 (3+2). The next schedule produced will allow model 1 to be produced in period 2 and model 2 to be produced in period 1. Since model 2 was originally from period 1, the number of late items does not increase, but remains at 2.

2 Items Late: Model 1 in period 2 and model 6 in period 2.

model	Period 1										Period 2					Period 3			
7	8	2	3	4	9	6	5	2	1	10	7	3	8	2	4	9	6	3	7
1	2	2	3	3	2	1	2	3	2	2	1	1	2	1	2	2	3	1	0
Squared Error																			

Squared Error

Total Squared Error: 42.9

Total Absolute Deviation: 97.5

Overall MSE: 0.113

Overall MAD: 0.257

There was no improvement in the overall MSE and an increase in the MAD value so according to step 6, we go back to the sequence which produced the lowest MSE and MAD

values. This sequence places models 2 and 1 in the periods where they were before the last swap.

2 Items Late

model	Period 1									Period 2									Period 3		
7	8	2	3	4	9	6	5	1	2	10	7	3	8	2	4	9	6	3	7		
1	2	2	3	3	2	1	2	3	2	2	1	1	2	1	2	2	3	1	0		

Squared Error

Total Squared Error: 42.7

Total Absolute Deviation 97.3

Overall MSE: 0.113

Overall MAD: 0.256

Now according to step 7 we check the other borders for possible improvements. The next highest border pair is models 6 and 3 with a combined MSE of 4. We swap the production dates of the two items to check for an improvement in the overall MSE.

2 Items Late

model	Period 1									Period 2									Period 3		
7	8	2	3	4	9	6	5	1	2	10	7	3	8	2	4	9	3	6	7		
1	2	2	3	3	2	1	2	3	2	2	1	1	2	1	2	2	3	1	0		

Squared Error

Total Squared Error: 41.9

Total Absolute Deviation 96.5

Overall MSE: 0.111

Overall MAD: 0.254

This sequence gives the lowest overall MSE and there are no more borders to check, so we conclude that the lowest possible MSE is 0.111 with 2 items late. Note that the algorithm does not keep track of whether an item is one period late or two periods late. Model 6 which is scheduled to be produced in period 3 has an actual due date in period 1.

NOTES ABOUT THE ALGORITHM:

This algorithm involving the heuristic procedure of picking items to be late according to the 'border sum of MSE's, gives better results than a previously run algorithm. Originally to achieve the lowest MSE, we sequenced the items with Monden's GC1 in a program that did not

restrict sequencing by due date. The result was a schedule that had 3 items late, but with the same 'best' MSE and MAD. The sequence follows.

3 Items Late

model			Period 1							Period 2							Period 3				
7	8	2	3	9	4	6	3	7	10	2	1	5	8	2	4	9	3	6	7		
1	2	2	3	3	2	1	2	3	2	2	1	1	2	1	2	2	3	1	0		
Squared Error																					

Squared Error

Overall MSE: 0.111 Overall MAD: 0.254

APPENDIX C

Input File Format and a Sample Input File

The first row indicates the number of items per model to be produced beginning with model type 1. The second row is the due date of each item. The following data shows this problem set contains 10 model types. There are 3 items to be produced of the first model type. The first is due in period 1, the second is due in period 3, and the third is due in period 1. The table that follows shows the parts list for each model type. In this problem there are 19 parts to be considered. Row 1 shows a part list for model type 1, row 2 shows a part list for model type 2, etc.. A '1' indicates that the part is included in the model, while a '0' indicates that it is not. Model 1 contains parts 1, 7, 12, and 15.

```
3 5 2 1 6 4 3 2 4 5
1 3 1 4 1 3 4 4 2 1 2 3 3 2 2 2 3 4 1 3 4 4 1 1 4 2 3 1 2 3 1 4 3 2 2

1 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0
0 0 1 0 0 0 0 1 0 1 0 1 0 0 1 0 0 0 0
0 1 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 1 0 1 1 0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0
0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 0
0 0 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 1 0
0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0
```

The constants required as input to the computer program follow: NUMPARTS 19, NUMMODELS 10, TOTALITEMS 35, and DAILY PRODUCTION 9. The daily production level must be established prior to solving the problem. In order for the computer program to properly calculate the schedule, the number of items produced each day should be set to one constant. It is not necessary to have full production of all items on the last day. In this problem,

the resulting sequence will be scheduled over a 4 day period with 9 products scheduled on days 1 through 3 and 8 items scheduled on day 4.

VITA

Robin Lovgren was born in Kendall, Florida on March 30, 1963. She attended schools in the public system of Clarksville, Tennessee, the Montgomery County Public School System, where she graduated from Montgomery Central High School in June, 1981. She entered Austin Peay State University, Clarksville, Tennessee, then in 1983 transferred to The Georgia Institute of Technology, Atlanta, Georgia where she was in the co-op program with Georgia Power Company, Atlanta, Georgia, and in May, 1986 she received the Bachelor of Industrial Engineering. After graduation she worked as an outside plant engineer with South Central Bell, Nashville, Tennessee.

In Fall, 1990 she entered The University of Tennessee, Knoxville, Tennessee to pursue a Master of Science degree in Statistics, and a Doctor of Philosophy in Management Science. These degrees will be awarded in August, 1996.

She is presently working as an Assistant Professor in the Industrial and Systems Engineering Program in the Civil Engineering Department at the University of Memphis, Memphis, Tennessee.

The author is a member of the Decision Sciences Institute (DSI) and the Institute for Operations Research and the Management Sciences (INFORMS).