

2014

Shop Scheduling In The Presence Of Batching, Sequence-dependent Setups And Incompatible Job Families Minimizing Earliness And Tardiness Penalties

Patricia Buchanan
University of Central Florida



Part of the [Industrial Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Buchanan, Patricia, "Shop Scheduling In The Presence Of Batching, Sequence-dependent Setups And Incompatible Job Families Minimizing Earliness And Tardiness Penalties" (2014). *Electronic Theses and Dissertations, 2004-2019*. 3018.

<https://stars.library.ucf.edu/etd/3018>



SHOP SCHEDULING IN THE PRESENCE OF BATCHING, SEQUENCE-DEPENDENT
SETUPS AND INCOMPATIBLE JOB FAMILIES MINIMIZING EARLINESS AND
TARDINESS PENALTIES

by

PATRICIA C. BUCHANAN

B.S. Industrial Engineering, University of Florida, 2003

M.S. Industrial Engineering, University of Florida, 2006

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2014

Major Professor: Christopher D. Geiger

ABSTRACT

The motivation of this research investigation stems from a particular job shop production environment at a large international communications and information technology company in which electro-mechanical assemblies (EMAs) are produced. The production environment of the EMAs includes the continuous arrivals of the EMAs (generally called jobs), with distinct due dates, degrees of importance and routing sequences through the production workstations, to the job shop. Jobs are processed in batches at the workstations, and there are incompatible families of jobs, where jobs from different product families cannot be processed together in the same batch. In addition, there are sequence-dependent setups between batches at the workstations. Most importantly, it is imperative that all product deliveries arrive on time to their customers (internal and external) within their respective delivery time windows. Delivery is allowed outside a time window, but at the expense of a penalty. Completing a job and delivering the job before the start of its respective time window results in a penalty, i.e., inventory holding cost. Delivering a job after its respective time window also results in a penalty, i.e., delay cost or emergency shipping cost. This presents a unique scheduling problem where an earliness-tardiness composite objective is considered.

This research approaches this scheduling problem by decomposing this complex job shop scheduling environment into bottleneck and non-bottleneck resources, with the primary focus on effectively scheduling the bottleneck resource. Specifically, the problem of scheduling jobs with unique due dates on a single workstation under the conditions of batching, sequence-dependent

setups, incompatible job families in order to minimize weighted earliness and tardiness is formulated as an integer linear program. This scheduling problem, even in its simplest form, is NP-Hard, where no polynomial-time algorithm exists to solve this problem to optimality, especially as the number of jobs increases. As a result, the computational time to arrive at optimal solutions is not of practical use in industrial settings, where production scheduling decisions need to be made quickly. Therefore, this research explores and proposes new heuristic algorithms to solve this unique scheduling problem. The heuristics use order review and release strategies in combination with priority dispatching rules, which is a popular and more commonly-used class of scheduling algorithms in real-world industrial settings. A computational study is conducted to assess the quality of the solutions generated by the proposed heuristics. The computational results show that, in general, the proposed heuristics produce solutions that are competitive to the optimal solutions, yet in a fraction of the time. The results also show that the proposed heuristics are superior in quality to a set of benchmark algorithms within this same class of heuristics.

To my loving husband

ACKNOWLEDGMENTS

I would first like to thank my advisor, Dr. Christopher D. Geiger, for his guidance and support during the development of my dissertation. He is a great teacher, mentor, and friend. His knowledge, patience, and dedication to guiding students helped to make this possible.

I would also like to thank Dr. Dima Nazzal who helped me develop my idea and the beginning stages of my research and phases. It was a pleasure working with her. I would also like to thank Dr. Mansooreh Mollaghasemi, Dr. Jennifer Pazour and Dr. Cheryl Xu for their insights, advice, and for serving on my doctoral dissertation committee.

I would also like to thank my former boss Phillip Burroughs, who constantly supported my continued education and was instrumental in helping me balance my work/life schedule. His constant encouragement and advice was one of the drivers that helped me succeed.

I would like to express my gratitude to all my friends and family for all their constant and loving support. They offered words of encouragement, guidance, a shoulder to lean on, babysitting, and reassurance in my ability to successfully finish this journey.

I especially thank my three children. Their smiles, hugs, and love helped me through my rough times. They were truly patient and understanding when mom had to study. They were my constant reminder why I was working so hard.

Most importantly, I would like to say thank you to my best friend and husband. He was incredibly supportive, encouraging, and was my sounding board when I needed to talk about ideas. This journey was as much his as it was mine.

TABLE OF CONTENTS

LIST OF FIGURES	x
LIST OF TABLES	xiii
CHAPTER 1 : INTRODUCTION	1
1.1. Background of the Industrial Setting Motivating This Research Investigation.....	1
1.1.1 The Importance of On-Time Delivery	2
1.1.2 The Coating Room.....	3
1.1.3 Process Characteristics within the Coating Room	5
1.2 Description of the General Problem	7
1.2.1 Job and Resource Characteristics.....	8
1.2.2 Processing Characteristics	9
1.2.3 The Performance Objective	12
1.3 Decomposition of the Problem – The Bottleneck.....	13
1.4 Objectives of This Research Investigation	14
1.5 Expected Contribution of This Research Investigation	15
1.6 Organization of This Document.....	15
CHAPTER 2 REVIEW OF THE EXISTING LITERATURE.....	17
2.1 Introduction.....	17
2.2 Job Shop Scheduling.....	17
2.2.1 Heuristic Approaches to the Job Shop Scheduling Problem	18
2.3 Batching and Sequence-Dependent Setups.....	19
2.3.1 Exact Approaches	20

2.3.2	Heuristic Approaches.....	21
2.3.3	Sequence-Dependent Setups.....	23
2.4	Unplanned Disruptions	24
2.5	Minimizing the Earliness and Tardiness Performance Objectives	26
2.5.1	Approaches for Minimizing Tardiness	26
2.5.2	Approaches for Minimizing Earliness-Tardiness	28
2.6	Summary.....	31
CHAPTER 3 : RESEARCH APPROACH.....		32
3.1	Introduction.....	32
3.2	Phase 1: Single Batch Workstation with Sequence-Independent Setups and Equal Job Weights	33
3.3	Phase 2: Single Batch Workstation with Sequence-Independent Setups and Unequal Weights	34
3.4	Phase 3: Single Batch Workstation with Sequence-Dependent Setups and Unequal Job Weights	36
CHAPTER 4 : MATHEMATICAL MODEL FORMULATION AND COMPUTATIONAL PERFORMANCE REQUIREMENTS		38
4.1	Phase 1: Single Batch Machine Scheduling with Sequence-Independent Setups and Equal Earliness and Tardiness Penalties.....	38
4.2	Computational Requirements of the Single Batch Machine Scheduling Problem with Sequence-Independent Setups and Equal Earliness and Tardiness Penalties	43
CHAPTER 5 : HEURISTIC SOLUTION PROCEDURES		45
5.1	Introduction.....	45
5.2	Proposed Solution Heuristics.....	45
5.2.1	Apparent Tardiness and Earliness Cost (ATEC) Heuristic.....	45

5.2.2	Batching Incompatible Families with Earliness Tardiness Cost (BIFET) Heuristic	47
5.2.2.1	The Family Batch Constraint (FBC) Parameter.....	47
5.3	Computational Study	49
5.3.1	Benchmark Heuristics.....	50
5.3.2	Experimental Design.....	52
5.4	Discussion of Computational Results	53
CHAPTER 6 : THE SINGLE BATCH WORKSTATION WITH UNEQUALLY-WEIGHTED PENALTIES AND VARYING BATCH SIZES AND INCOMPATIBLE FAMILIES AND SEQUENCE-INDEPENDENT SETUPS		76
6.1	Introduction.....	76
6.2	Phase 2: Single Batch Workstation with Sequence-Independent Setups and Unequal Job Weights.....	76
6.2.1	Experimental Design.....	77
6.2.2	Performance Results of Varying Maximum Batch Sizes	77
6.2.3	Unequal Weights.....	83
6.3	Integrating Order Review and Release Strategies	87
6.3.1	Experimental Design with ORR Strategies.....	89
6.4	Results with ORR Strategies.....	90
CHAPTER 7 : THE SINGLE BATCH WORKSTATION WITH UNEQUALLY-WEIGHTED PENALTIES AND VARYING BATCH SIZES AND INCOMPATIBLE FAMILIES AND SEQUENCE-DEPENDENT SETUPS		95
7.1	Introduction.....	95
7.2	Mathematical Model	95
7.2.1	Computational Time of the Mathematical Model vs the Proposed and Benchmark Heuristics	101

7.2.2	Experimental Design.....	103
7.3	Discussion of the Performance Results.....	104
7.3.1	Batch Size Impact	104
7.3.2	Effects of Weighted the Earliness and Tardiness Penalties	105
7.3.3	Overall Performance of Dispatching Rules and ORR versus Proposed Heuristics	109
CHAPTER 8 : SUMMARY AND FUTURE RESEARCH DIRECTIONS.....		119
8.1	Summary	119
8.2	Future Research Directions.....	120
8.2.1	Modeling Multiple Serial Machines in a Flow shop.....	120
8.2.2	Modeling Multiple Parallel Machines	121
8.2.3	Machine scheduling with Job Preemption	123
8.2.4	Due Date Time Windows.....	124
APPENDIX A: EQUAL WEIGHTS		126
APPENDIX B: EARLY PENALTY =10/TARDY PENALTY =1		157
APPENDIX C: TARDY PENALTY=10/EARLY PENALTY = 1		188
LIST OF REFERENCES		218

LIST OF FIGURES

Figure 1.1. General product flow through the Coating Room.	4
Figure 2.1. Summary of existing solution methodologies (modified from Mathirajan et al. 2006).	20
Figure 5.1. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and high traffic intensity.	65
Figure 5.2. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and medium traffic intensity.	66
Figure 5.3. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, low traffic intensity.	67
Figure 5.4. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and low traffic intensity.	68
Figure 5.5. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and low traffic intensity.	69
Figure 5.6. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and high traffic intensity.	70
Figure 5.7. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and all jobs arriving simultaneously.	71
Figure 5.8. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and all jobs arriving simultaneously.	72
Figure 5.9. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and medium traffic intensity.	73
Figure 5.10. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and medium traffic intensity.	74
Figure 6.1. 95% confidence interval on the performance results with varying batch sizes for tight due dates, all jobs arrive simultaneously, 5 families, and earliness penalty = 10 and tardiness penalty = 1.	81

Figure 6.2. 95% confidence interval on the performance results with varying batch sizes for tight due dates, all jobs arrive simultaneously, 5 families, and earliness penalty = 1 and tardiness penalty = 10.	81
Figure 6.3. 95% confidence interval on the performance results with varying batch sizes for tight due dates, medium traffic intensity, 5 families, earliness penalty = 10 and tardiness penalty = 1.	82
Figure 6.4. 95% confidence interval on the performance results for tight due dates, low traffic intensity, 5 families, and maximum batch size 10.	85
Figure 6.5. 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 1 family, and maximum batch 3.	85
Figure 6.6. 95% confidence interval on the performance results for Max batch size 1, tight due dates, medium traffic intensity, and 1 family.	86
Figure 6.7. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, all jobs arriving simultaneously, and 5 families.	91
Figure 6.8. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, all jobs arriving simultaneously, and 5 families.	92
Figure 6.9. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, high traffic intensity, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.	93
Figure 6.10. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, low traffic, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.	93
Figure 6.11. 95% confidence interval on the performance results for earliness penalty=10 and tardiness penalty=1, for tight due dates, high traffic intensity, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.	94
Figure 6.12. 95% confidence interval on the performance results for earliness penalty=1 and tardiness penalty=10, loose due dates, high traffic, 5 families, dispatching rules compared with ORR strategies combined with dispatching rules.	94
Figure 7.1. Computational time for CPLEX to arrive at a solution as problem complexity increases in terms of the number of jobs.	102

Figure 7.2. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, and high traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.	107
Figure 7.3. 95% confidence interval on the performance results for earliness penalty =10 and tardiness =1, tight due dates, and low traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.	108
Figure 7.4. 95% confidence interval on the performance results for tardiness penalty =10, and earliness penalty = 1, loose due dates, and high traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.	109
Figure 7.5. 95% confidence interval on the performance results for equal earliness and tardiness penalties, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.	110
Figure 7.6. 95% confidence interval on the performance results for tardiness penalty =1, and earliness penalty = 10, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.	111
Figure 7.7: 95% confidence interval on the performance results for earliness penalty =1 and tardiness =10, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.	112

LIST OF TABLES

Table 4.1. Summary of CPLEX solution times in order to reach an optimal solution.	44
Table 5.1. Experimental design for phase 1: single batch workstation with sequence-independent setups and equal job weights.....	53
Table 5.2. Performance results for 5 jobs with tight due dates results.	54
Table 5.3. Performance results for 10 jobs with tight due date results.	54
Table 5.4. Performance results for 15 jobs with tight due dates.....	55
Table 5.5. Performance results for 5 jobs with loose due dates.....	56
Table 5.6. Performance results for 10 jobs with loose due dates.....	56
Table 5.7. Performance results for 15 jobs with loose due dates.....	56
Table 5.8. Performance results for 5 jobs that arrive simultaneously.....	57
Table 5.9. Performance results for 10 jobs that arrive simultaneously.....	58
Table 5.10. Performance results for 15 jobs that arrive simultaneously.....	58
Table 5.11. Performance results for 5 jobs with high traffic intensity.	59
Table 5.12. Performance results for 10 jobs with high traffic intensity.	59
Table 5.13. Performance results for 15 jobs with high traffic intensity.	60
Table 5.14. Performance results for 5 jobs with medium traffic intensity.	61
Table 5.15. Performance results for 10 jobs with medium traffic intensity.	61
Table 5.16. Performance results for 15 jobs with medium traffic intensity.	62
Table 5.17. Performance results for 5 jobs with low traffic intensity.....	62
Table 5.18. Performance results for 10 jobs with low traffic intensity.....	63
Table 5.19. Performance results for 15 jobs with low traffic intensity.....	63

Table 6.1. Experimental design with emphasis on unequal job weights and varying batch sizes.	77
Table 6.2. Comparison of ATC and ATEC performance of percent error with varying batch sizes and equal earliness and tardiness weights.	79
Table 6.3. Comparison of ATC and BIFET performance of percent error with varying batch sizes and unequal earliness and tardiness weights where earliness is more important than tardiness.	79
Table 6.4. Comparison of ATC and ATEC performance of percent error with varying batch sizes and unequal earliness and tardiness weights where tardiness is more important than earliness.	80
Table 6.5. Comparison of FEDD and BIFET performance of percent error when varying penalties at a maximum batch size of 10.	84
Table 7.1. Experimental design with emphasis on unequal job weights and sequence-dependent setups.	103
Table 7.2. Comparison of MIL with ATC and BIFET performance of percent error when varying batch sizes.	105
Table 7.3. Comparison of MIL with ATC and BIFET performance of percent error when varying penalties.	106
Table 7.4. Average percent error for heuristics.	113
Table 7.5. Comparison of rule performance for ATC against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.	114
Table 7.6. Comparison of rule performance for ATEC against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.	115
Table 7.7. Comparison of rule performance for BIFET against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.	116
Table 7.8. Comparison of rule performance for MIL ATC against ATC for earliness penalty =1/tardiness penalty =10.	117
Table 7.9. Comparison of rule performance for ATEC against ATC for earliness penalty =1/tardiness penalty =10.	118

Table 7.10. Comparison of rule performance for BIFET against ATC for earliness penalty =1/tardiness penalty =10.	118
---	-----

CHAPTER 1 : INTRODUCTION

1.1. Background of the Industrial Setting Motivating This Research Investigation

The motivation of this research investigation stems from a particular production environment at large international communications and information technology company serving both the government and commercial markets. This company specializes in government contracts that provide funding to the company to produce various complex electro-mechanical assembly systems that support communications systems. These communication systems are typically installed on avionics, space, and ground systems such as radios, reflectors, ground system radars, etc. Often, the assemblies start with the production of circuit card assemblies (CCAs). These CCAs are later assembled into a chassis using mechanical subassemblies.

The typical customers for these products are the United States military, foreign militaries and industries, and commercial customers. The requirements of a government contract constitute what is called a manufacturing program, where the program has unique hardware requirements, quantities for the electro-mechanical assemblies (EMAs), and a unique customer. The work for a program includes the effort of designing and manufacturing a set of products or systems for that customer. Each EMA can contain one or multiple CCAs, and there can be multiple types (or families) of CCAs built for one program.

1.1.1 The Importance of On-Time Delivery

The Master Production Schedule (MPS) outlines the manufacturing start time (or release date) of each assembly within a program (or within multiple programs), expected manufacturing completion dates, and customer due dates. Strict adherence to the MPS for a contract is the primary objective, and the MPS is often fixed. The metric often used to measure on-time delivery performance of a program is the Schedule Performance Index (SPI). SPI is the ratio of what is completed versus what was scheduled to be completed. An SPI value equal to 1.0 means the program is on-schedule for delivery. An SPI value less than 1.0 means the program is behind schedule.

Strict adherence to the fixed schedule is necessary for several reasons. Firstly, some assemblies must be loaded onto cargo ships that arrive at local loading docks at specific times during the year, and typically, there is a one-week window during which these ships are docked. The assemblies must be delivered to and arrive at the loading dock during this one-week window in order to be loaded onto the cargo ship. Therefore, it is imperative that all deliveries arrive on time.

Secondly, missing a delivery date could result in a loss of an award fee to the company. Typically, programs are financially compensated when they reach certain performance goals. One of these performance goals is on-time delivery performance. The award fees are given when the program meets each delivery milestone. The award fee is another incentive for the manufacturing completion dates of programs to be aligned to the MPS.

Finally, finishing a program's production ahead of schedule incurs a penalty in the form of rework costs. If production jobs are finished ahead of schedule (or, early), and there is an engineering change order that requires a change to the design of the product, or if it is found that a component is not performing to its full capability and needs to be removed and replaced by a superior component, then the finished inventory will have to be reworked and retested.

1.1.2 The Coating Room

The particular area of interest within the subject manufacturing facility is the Coating Room, which is a key area in the production of circuit card assemblies. This area is classified as a low-volume, high product variety job shop. Virtually all programs completed in the facility require at least one or more process steps performed in the Coating Room, which results in a large variety of CCAs that visit the Room. The Coating Room has several functional areas such as Bonding, Staking and Coating (see Figure 1.1.).

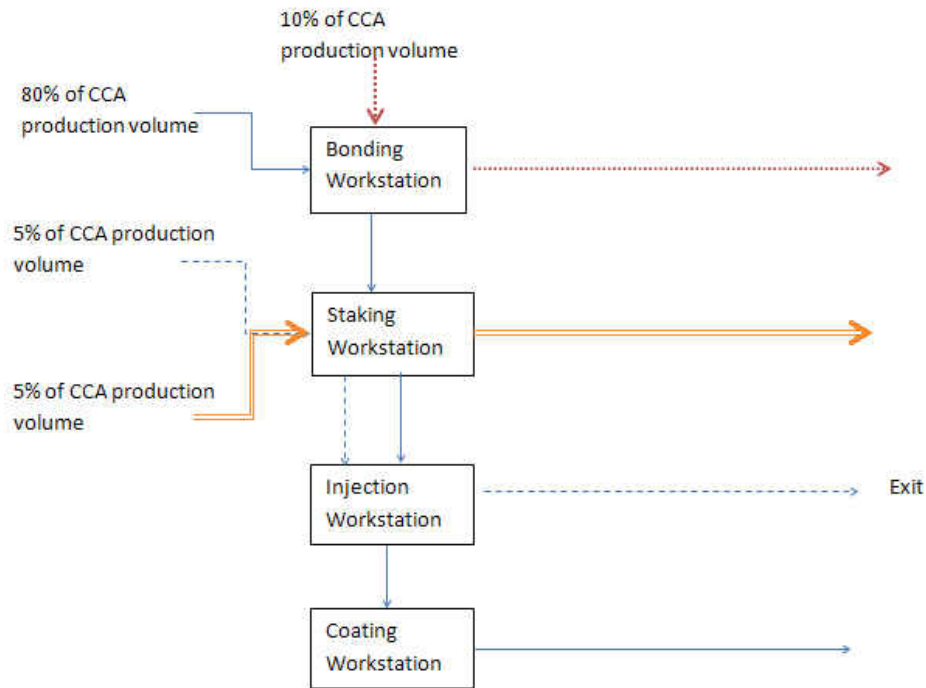


Figure 1.1. General product flow through the Coating Room.

Each process involves the application of chemicals and chemical adhesives to the assemblies. The assemblies require these additional processes for structural reinforcement, moisture protection, foreign object damage protection, and thermal protection to enhance heat transfer away from assembly sub-components, especially those that are critical to the functionality, durability and useful life to the CCAs. A single assembly can require one or multiple processes and can enter the Coating Room multiple times during the production of a circuit card. The type of processes and chemicals the assemblies require depend on the operating environment (i.e., ground, flight, and/or space) of the final assembly product.

The chemicals in the Coating Room are applied manually to the assemblies by Coating Room technicians and have long process times associated with them. More specifically, the chemicals and chemical adhesives must be given time not only to be applied to the assemblies but also to cure and harden after being applied.

1.1.3 Process Characteristics within the Coating Room

1.1.3.1 Batch Processing and Machine Changeovers

CCAs are typically processed in batches based on product type (or, family). A family of CCAs is a group of assemblies that share the same underlying architecture and are closely related in production and process requirements. CCAs of different families cannot be batched and processed together. The size of the batches varies depending upon processing requirements and CCA availability at the time of processing at a particular machine or set of machines.

Batching of assemblies is performed at different process steps and is due to the often long times that occur when changing a machine's current production setup to another. Machine changeover is needed when processing batches of jobs both within a CCA product family and between different CCA product families. As such, there are two types of machine changeovers – minor changeover and major changeover. As a result, there are two types of machine setup times associated with the changeover types. A minor changeover incurs a minor setup time, and a major changeover incurs a major setup time. A minor changeover at a machine takes place between two successive batches of jobs from the same family. Therefore, minor setup times tend

to be relatively short in length. A major changeover at a machine takes place between two successive batches of CCAs from different product families. The major setup time includes the time to clean the machine, to apply electronic devices to the batch of assemblies prior to the use of the new chemicals, and to mix and weigh those chemicals. Therefore, major setup times tend to be significantly longer than minor setup times.

1.1.3.2 Sequence-Dependent Setups

The length of the machine setup times between CCA product families (i.e., major setup times) varies depending upon the sequence in which the product families are processed at the machines. In other words, a machine's setup time for a batch from a particular CCA product family is determined not only by that batch's product family but also by the previous CCA product family for which the machine is currently setup.

1.1.3.3 Disruptions in the Production Schedule

Planned and unplanned disruptions in the MPS occur in the Coating Room and often threaten the on-time delivery of assemblies to the customer. By disruptions, it is meant that the scheduled completion of the CCAs within the Coating Room is delayed due to an expected or unexpected event. As a result, the delivery of the EMAs awaiting the installation of the CCAs is also delayed.

Unplanned disruptions are more commonplace in this particular job shop, and there are two main types of unplanned disruptions in the Coating Room. The first is the unexpected arrival of high priority (or “hot”) CCAs to the Room, or the unexpected elevation of priority of CCAs currently in process in the Room.

The second type of unplanned disruption is the availability of the chemicals and chemical adhesives, which have varying shelf lives and varying work lives. The shelf life of a chemical is the length of time that it can remain in its packaging or container before it expires prior to the container and packaging of the chemical being opened and the chemical being mixed. The work life is the amount of time the chemical is usable once it is mixed or once it is out of its container. Each chemical needs to be mixed prior to applying it to assemblies belonging to a family not currently being processed. When the chemicals expire according to the shelf life, the processing of the awaiting CCAs is delayed because the chemicals need to be recertified, in which the shelf life of the unusable chemicals is extended.

1.2 Description of the General Problem

In this section, the shop floor configuration and the general problem that underlie the industrial problem motivating this research investigation, as explained in Section 1.1, is described.

1.2.1 Job and Resource Characteristics

In the fundamental problem, there is an index set \mathbf{J} of n jobs (i.e., $i = \{1, \dots, n\}$) to be processed through an index set \mathbf{W} of m workstation resources (i.e., $j = \{1, \dots, m\}$) during a certain planning horizon T .

A job, in this context, refers to a single circuit card assembly unit and is generally referred to as job i . Jobs arrive on a continuous basis to the shop floor during planning horizon T , and each job i has a release date, r_i , at which the job is initially made available for processing on the shop floor. Each job i has a due date, d_i , by which the job is due to an internal or external customer and a weight, w_i , which indicates the priority of the job. Depending upon customer requirements, some jobs have a higher priority than others, and have to meet internal and/or external due dates. It is important to note that the due date d_i of each job i has a lower bound and upper bound, L_i and U_i , respectively. These bounds create a non-relaxable delivery time window for each job.

A workstation resource in this context consists of a single machine or a number of parallel identical machines. The m workstations are arranged by function in a low-volume, high-mix job shop. Job shops, by definition, are production systems that are configured by function to produce a number of different job types. Each type has a unique routing and a different processing time on each machine visited.

1.2.2 Processing Characteristics

1.2.2.1 Batching and Incompatible Job families

A machine in the production shop floor underlying the industrial problem processes a number of jobs simultaneously as a batch. Specifically, the jobs traverse the shop floor in transfer batches, and they are processed at the machine individually. A batch consists of a finite number of circuit card assembly units. A batch does not move on to the next station in its routing until all jobs in the batch have been processed.

Batch splitting, where a batch is divided into sub-batches and each sub-batch is process simultaneously across multiple resources, does not occur. In other words, once a batch is formed, it is permanent and is processed as a single unit. Batch preemption, however, does occur. Batch preemption occurs when a single batch is in process at a machine and another batch with higher priority arrives at that machine and interrupts the batch currently in process. After completing the higher priority batch that arrived, the preempted batch resumes processing until it is completed unless another higher priority batch arrives before it completes processing, and so on.

The set \mathbf{J} of n jobs awaiting processing can be partitioned into F different families and each job within a family f is denoted as job i^f . The number of jobs in family f is denoted by n^f . The processing times of all jobs belonging to family f are equal to a common processing time, p_j^f , for each machine j . It is important to note that the F product families are incompatible, in that jobs from different families cannot be batched together in the same batch. This is due to the

various chemical and chemical adhesive requirements. Therefore, each batch k is made up n_k^f jobs belonging to a single product family.

Batch sizes within a product family are not fixed and can vary across the different product families. In other words, a batch k of n_k^f jobs is processed of family f , and a batch l of n_l^g jobs of another family g can be processed next. Each machine is capable of processing up to B_{\max}^f jobs from the same family f simultaneously as a batch and each machine can begin processing a batch when there are at least B_{\min}^f jobs from the same family f available simultaneously at the machine.

The processing time of a batch k is equal to the sum of the processing times of its jobs. More succinctly, the processing time of a batch k is equal to the number of jobs in batch k times the common processing time for the family, i.e., $p_k^f = n_k^f \times p_j^f$.

1.2.2.2 Sequence-Dependent Setup Times

A common machine setup time precedes the processing of the first batch of family g after completing processing of the last batch of family f , and the setup times at the machines are influenced by the sequence in which product families are processed. In other words, the setup time to change over a machine j after processing the last in a sequence of batches of jobs belonging to family f in order to process the first in a sequence of batches of jobs belonging to family g is denoted by s_{jfg} . Therefore, the sequence-dependent setup times for the F families can be stored in an F -by- F matrix. Note that the setup time to change over a machine from family g

back to family f can be different. Also, it is important to note that a changeover is needed at a machine j between two batches of the same family, i.e., $\forall f: s_{jff} \neq 0$. In other words, the diagonal entries of the F -by- F matrix of setup times are non-zero.

1.2.2.3 Unplanned Disruptions

There are high priority, or “hot”, jobs that arrive unexpectedly to the shop floor. Also, due to the finite work life and finite shelf life of the chemicals and chemical adhesives, the perishability of materials is a prominent feature of the general problem.

First, consider the unexpected arrival of hot jobs during planning horizon T . Upon the arrival of a hot job i to the first machine in its routing, p represents the number of jobs impacted by job i 's arrival and need to be rescheduled and these p jobs make up the subset of jobs \mathbf{J}_i , where $\mathbf{J}_i \subset \mathbf{J}$. There is a subset of workstations that process that particular job along its routing and along the routing of the other p jobs that need to be rescheduled. Let q represents the number of workstations to be rescheduled in the subset of workstations \mathbf{W}_i for hot job i and the other p impacted jobs, where $\mathbf{W}_i \subset \mathbf{W}$.

Second, in addition to the unexpected arrival of hot jobs, jobs must arrive to workstations before the expiration of the shelf and work lives of the chemicals required for processing. This expiration time can be represented using a time window at the different workstations in each arriving job's process routing. Therefore, similar to that which is discussed in Section 1.2.3, delivery to each workstation is allowed outside the time window, but at the expense of an

associated penalty. When a job arrives to a workstation before the start of its respective time window for the chemicals results in a penalty, e.g., inventory holding/storage cost. If a job's estimated completion time at a workstation is after its respective time window for the chemicals, this also results in a penalty, e.g., delay cost due to chemical remixing and recertification. However, this penalty is much more severe than that incurred when a job arrives to a workstation before the start of its respective time window, and thus should be avoided.

1.2.3 The Performance Objective

Recall that strict adherence to the fixed Master Production Schedule is necessary for several reasons, and there is a one-week time window during which jobs must be delivered and loaded onto cargo ships. Therefore, it is imperative that all product deliveries arrive on time within the delivery time windows. Delivery is allowed outside the time window, but at the expense of an associated penalty. The process of manufacturing these electro-mechanical assemblies are very lengthy and complex. Completing a job and delivering the job before the start of its respective time window result in a penalty, i.e., inventory holding cost. Delivering a job after its respective time window also results in a penalty, i.e., delay cost or emergency shipping cost. In summary, completing shipments too early or too late is disadvantageous, and no penalty is incurred when a job is completed and delivered during its respective delivery time window. As a result, the performance objective for the general problem is a composition of performance measures E_i and T_i , where

$$E_i = \max \{L_i - c_i, 0\} \quad i = 1, \dots, n, \text{ and} \quad (1.1)$$

$$T_i = \max \{c_i - U_i, 0\} \quad i = 1, \dots, n. \quad (1.2)$$

Hence, the performance objective for the general problem is to minimize

$$Z = \sum_{i=1}^n (\theta_{1i} E_i + \theta_{2i} T_i), \quad (1.3)$$

where c_i is the actual completion/delivery time of job i , θ_{1i} is the cost per unit time of completing job i before the beginning of the delivery time window, and θ_{2i} is the cost per unit time of completing job i after the end of the delivery time window.

To summarize, the general problem considered in this research investigation is expressed using the widely adopted three-field scheduling notation $\alpha \mid \beta \mid \gamma$ (Blazewicz *et al.*, 1996) as $Jm \mid$

$$r_i, \text{ batch, prmp, } s_{ij} \mid \sum_{i=1}^n (\theta_{1i} E_i + \theta_{2i} T_i).$$

1.3 Decomposition of the Problem – The Bottleneck

In the industry problem, the overall performance of the Coating Room is greatly inhibited by the bottleneck – the Bonding Station, and approximately 80% of the jobs visit the Bonding Station. Therefore, this research can make inroads to the industry problem by focusing on the effective scheduling of this one workstation, which would, in turn, improve the overall performance of the entire production system.

1.4 Objectives of This Research Investigation

Hence, the primary objectives of this research investigation are to:

1. *Model the scheduling problem for one workstation in the presence of batching, incompatible job families and sequence-dependent setup times.* The definition and modeling of this scheduling problem, $1 \mid r_i, \text{batch}, s_{ij} \mid \sum_{i=1}^n (\theta_{1i}E_i + \theta_{2i}T_i)$, is a contribution in and of itself. The definition discusses the characteristics and challenges of this interesting problem and the subsequent model of the general problem presents valuable insights to solving it; and
2. *Develop effective heuristic solution approaches for the $1 \mid r_i, \text{batch}, s_{ij} \mid \sum_{i=1}^n (\theta_{1i}E_i + \theta_{2i}T_i)$, scheduling problem in the presence of batching, incompatible job families, and sequence-dependent setup times.* It is well-known that the single batch resource scheduling problem for all but very few special situations is NP-hard. In other words, a polynomial-time algorithm that solves this problem optimally does not exist. Therefore, researchers and practitioners seek and develop heuristic approaches to address this problem. The proposed solution approaches can serve as the initial effort to address this unique problem and other problems similar in structure.

1.5 Expected Contribution of This Research Investigation

The contribution of this research is two-fold. First, the general scheduling problem underlying the industrial problem motivating this research investigation is known to be NP-hard. Therefore, any effective solution approaches to this problem, not only for some simple, smaller special cases but also for cases that align with more realistically-sized job shop scheduling problems, will contribute significantly to the research body of knowledge in operations/production scheduling. Second, this investigation explores new approaches that should help shop floor production supervisors and managers effectively manage and schedule jobs in order to minimize the deviation of actual job delivery dates from scheduled delivery dates. Further, it should lead to the development of better proactive scheduling policies in job shops where sequence-dependent setups and batching are present.

1.6 Organization of This Document

The remainder of this document is organized as follows. Chapter 2 discusses the previous literature related to the primary areas of this research investigation, which include job shop scheduling, batching with sequence-dependent setups and incompatible job families. An overview of past solution approaches is also presented. Chapter 3 describes the research approach of this investigation in detail. Chapter 4 summarizes the proposed mathematical formulation of the scheduling problem, followed by Chapter 5, which presents the proposed heuristics and the benchmark heuristics to which the proposed heuristics are compared. Chapter 6 summarizes sensitivity analyses, where the weights of the earliness and tardiness measures and

limits of batch sizes are varied. The effects of these changes on the performance of the proposed heuristics and the benchmark heuristics are presented and discussed. This chapter also introduces the use of Order Review and Release strategies. Chapter 7 increases the complexity of the problem formulation by introducing sequence-dependent setups and updates the previous mathematical formulation to include this problem characteristic. Chapter 8 summarizes this research investigation as well as directions for future research.

CHAPTER 2

REVIEW OF THE EXISTING LITERATURE

2.1 Introduction

This chapter presents a review of the literature covers the job scheduling problem along with various approaches to solve this problem. The chapter is divided into several sections. The first major section summarizes existing literature on job shop scheduling problems. The second section discusses scheduling problems that include the batching of jobs and sequence-dependent setups. The third discusses existing methods for scheduling due to unplanned disturbances in the manufacturing flow. The chapter concludes with a review of the research gaps that are addressed during this investigation.

2.2 Job Shop Scheduling

A job shop is a facility that typically has many product families, but low volumes, and units visit multiple workstations during their production. The job shop scheduling problem is a well-researched problem and has been studied for many decades. One of the first to explore this problem is the research of Balas (1969). As a result, it is impractical to review all the research that addresses the general job shop scheduling problem and its variants. Therefore, the interested reader is encouraged to review the following works, which provide detailed background on this well-studied problem – Mellor (1966), Rodammer and White (1988), Blazewicz et al. (1996), Jain and Meeran (1999), Pinedo (2002), and Parveen and Ullah (2010). It has been concluded in

the open literature that the job shop scheduling problem, even in its simplest form, is proven to be NP-hard and no polynomial-time algorithm exists to solve this problem to optimality (see, for example, Muth and Thompson (1963), Garey et al. (1976), Kan (1976), Sotskov and Shakhlevich (1995)).

2.2.1 Heuristic Approaches to the Job Shop Scheduling Problem

Since the job shop scheduling problem is NP-hard, researchers typically pursue heuristic solution approaches instead of exact approaches. Other heuristic approaches that are applied to the job scheduling problems and perhaps have greater appeal, especially to practitioners, are order review and release (ORR) strategies and dispatching rules. These methodologies are shown to be effective in controlling the amount of work-in-process inventory on the floor as well as provide the best sequencing of jobs with respect to the desired performance criterion or set of criteria (e.g., Philipoom et al. 1993; Sabuncuoglu and Karapinar 2000; Chiang and Fu 2009; Lu et al. 2011).

Job shop scheduling problems involving batching and/or sequence-dependent setups are quite common in scheduling research literature (e.g., Gentile 2009; Vinod and Sridharan 2009; Zhang and Gu 2009). The next section discusses these problem features and methodologies proposed to solve scheduling problems with these features.

2.3 Batching and Sequence-Dependent Setups

There are different approaches in solving scheduling problems when batching and sequence-dependent setups are present. These solution approaches range from exact methodologies to heuristic methodologies. These various solution approaches can be used in order to improve performance metrics such as number of tardy jobs, maximum lateness and deviation from job due dates.

Batching is typically done when machine changeovers (or, setups) are required, in order to improve machine utilization, especially when the machine is a shared resource across product families. One industry in which the use of shared resources is common is the semiconductor manufacturing industry. This is due to the complex nature of the product being manufactured. This environment is similar to the industrial setting motivating this research investigation, with the exception that batch processors in a semiconductor manufacturing environment can process multiple units simultaneously. Scheduling of batch processors in a semiconductor industry has been studied for many years (e.g., Kim et al. 1998, Min and Yih 2003, Mathirajan et al. 2006, Lee et al. 2009, Senties et al. 2009). Mathirajan et al. (2006) identify three primary categories of methods to address these problems: mathematical programming models, exact approaches and heuristic approaches (see Figure 2.1).

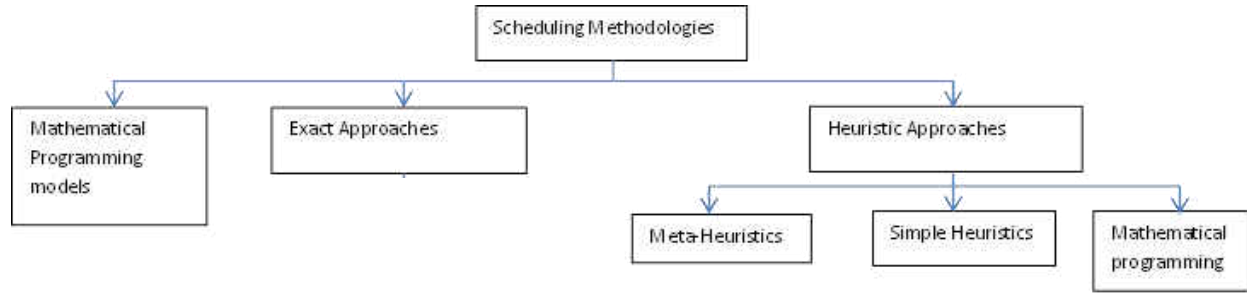


Figure 2.1. Summary of existing solution methodologies (modified from Mathirajan et al. 2006).

Many researchers address batch scheduling in other production environments, e.g., the automotive industry (e.g., Gokhale and Mathirajan 2010) and the fruit canning industry (e.g., Parthanadee and Buddhakulsomsiri 2010). These researchers explore different approaches, such as dispatching rules, to address their particular problem in their respective industries.

2.3.1 Exact Approaches

Pott et al. (2000) review batch scheduling and identify past research from Rinnooy Kan (1976) that considers the $1|s_{ij}|\sum C_i$ problem, which is the minimization of total job completion time on a single machine where there are job families and sequence-dependent family setup times. This problem is unary NP-hard for an arbitrary number of product families F . This research is interested in preventing late deliveries, and as a result, one of the objectives of interest is to minimize job maximum lateness, or L_{\max} .

Solving this problem for an arbitrary number of product families has been found to be NP-hard by Bruno and Downey (1978), and, as a response, Hariri and Potts (1997) develop a

branch and bound algorithm to solve such a problem using the Earliest Due Date (EDD) dispatching rule. A review of Pott et al. (2000) and Cheng et al. (2001) reveals that scheduling with batching is an NP-hard problem, and heuristics are better suited at solving a scheduling problem when batching is present. Cheng et al. (2001) show that single batch processor scheduling problems with multiple product families, setup times, and focusing on due date objectives, such as minimizing maximum lateness, is strongly NP-hard.

2.3.2 Heuristic Approaches

Given the computational complexity results of the batch scheduling problem, researchers, such as Mathirajan and Sivakkumar (2006), Geiger and Uzsoy (2008), Gokhale et al. (2010), explore the effects batching have on scheduling. Geiger and Uzsoy (2008) extend their previous work Geiger et al. (2006) of utilizing a bio-inspired genetic learning approach and apply it to more complex batch processing environments. They are able to evaluate this approach over a range of single batch processor scheduling problems under different production conditions to identify the robustness and performance of their proposed learning approach against other known heuristics such as Greedy Earliest Due Date (GREDD) (Uzsoy 1995) and Batch Apparent Tardiness Cost (BATC) (Mehta and Uzsoy 1998).

Gokhale et al. (2010) look at the problem of scheduling a batch processor in presence of unequal release times, incompatible job families and non-identical job sizes with job splitting to minimize the total weighted tardiness. The batch processor can process multiple compatible jobs at one time. The focus of their research is on the bottleneck of an automobile gear manufacturer.

In order to solve this problem, Gokhale et al. (2010) develop three heuristic algorithms (HAs): Weighted tardiness based HA, Weighted job score based HA, and Weighted family score and job score based HA. All three heuristics are tested on smaller-sized problems and then later on larger problems, and prove to be relatively effective.

Another approach to solving scheduling problems with batching is with the use of order review and release (ORR) strategies and dispatching rules. The combination of these two scheduling strategies have been shown to be effective in reducing work-in-process (WIP) inventory in order to improve the overall performance of the shop floor (e.g., Gronalt 2002 and Gentile 2009).

Batch preemption can occur when a machine is processing a job of an in-process batch and that processing is often interrupted by a batch (or single job) of higher priority. That machine then processes the interrupting higher priority batch (or single job) to completion, then resumes processing the previously interrupted job and completes the in-process batch. An assumption of many researchers used to solve scheduling problems is to not allow batch or machine preemptions (e.g., Mathirajan 2006, Chiang and Fu 2007, Geiger and Uzsoy 2008, Gokhale and Mathirajan 2010). Scheduling problems in which batch preemption occurs is NP-hard when the number of machines m is 2 (Monma and Potts 1993). Therefore, heuristic approaches are appropriate when solving batch scheduling problems with preemption.

2.3.3 Sequence-Dependent Setups

Zhu and Wilhelm (2005) summarize recent results in the existing literature for scheduling problems with sequence-dependent setups. Most research considers the reduction of setup costs and inventory holding costs. This research shows that any scheduling problem that has sequence-dependent setups any performance objective (e.g., minimize makespan, minimize total flow time, minimize maximum lateness, etc.) is NP-hard (see, for example, Monma and Potts 1989, Ghosh 1994, Pinedo 2002). As a result, even though some researchers have pursued exact approaches for simplified versions of scheduling problems with sequence-dependent setups, more commonly, heuristic solution approaches have been developed in order to address problems with sequence-dependent setups, including genetic algorithms, dispatching rules and simulation, to name a few.

2.3.3.1 Exact Approaches

A scheduling problem involving sequence-dependent setups is well-known to be similar to that of the Traveling Salesperson problem (TSP), which is known in literature to be NP-hard (see, for example, Kan 1976, Pinedo 2002, Zhu and Wilhelm 2006). The traveling salesperson problem is a widely-studied mathematical problem for which the ultimate goal is to find the optimal path given a set of distances. This problem has been applied to many other fields such as logistics, planning and manufacturing. A literature review by Allahverdi et al. (2008) shows that scheduling problems, with sequence-dependent setups, regardless if there is or isn't batching, are NP-hard.

2.3.3.2 Heuristic Approaches

Mathematical programming models, simulation models and metamodels, etc. all have been used to address scheduling problems in various environments (e.g., Ruiz 2008, Vinod and Sridharan 2009, Salmas et al. 2010, Fernandes et al. 2011).

Various heuristic scheduling approaches have been used and tested in order to improve shop floor performance. One of these approaches is the Similar Setup (SIMSET) rule (Vinod and Sridharan 2009, Fernandes et al. 2011), which has proven to be effective especially in the case of sequence-dependent setups.

Another popular strategy for solving scheduling problems with sequence-dependent setups is the integration of both order review and release (ORR) strategies and dispatching rules. Missbauer (1996), Lee (1997) and Gentile (2008) evaluate the performance of these strategies in a job shop and show the effectiveness of these strategies in reducing production costs and reducing work-in-process inventory. Missbauer (1996) create an analytical model as well as a simulation model. The author tests and analyzes the performance of these rules and determines the most suitable combination of ORR strategies and dispatching rules.

2.4 Unplanned Disruptions

In a manufacturing environment, it is often difficult to plan or predict the performance of the manufacturing system due to random disturbances, which can alter the originally intended schedule. Many theories have been suggested to identify the best planning strategies and

identify when there are dangers of missed deliveries. One of the planning strategies that can be utilized is dispatching rules.

Dispatching rules identify priorities of jobs waiting to be processed on a machine and can be based on various criteria. New research has suggested using simulation in order to select dispatching rules as the manufacturing conditions change (e.g., disruptions occur, unanticipated orders arrive, etc.). Interruptions such as machine failures, process yield and rework are some issues that impact the dispatching rules in the study performed by Min and Yih (2003). Their objective is to utilize scheduling rules to minimize work-in-process inventory, improve throughput. They researchers develop a simulation-based scheduler that provides real-time dispatching decisions.

A successful approach to dealing with unexpected changes in the system is to modify dispatching rules in real-time (e.g., Jeong and Kim 1998, Wu et al. 1999, Min and Yih 2003, Parthanadee and Buddhakulsomsiri 2010). Parthanadee and Buddhakulsomsiri (2010) successfully apply dispatching rules to a fruit canning facility's bottleneck. The desired outcomes of Parthanadee and Buddhakulsomsiri (2010)'s research are to:

1. identify the dispatching rules for scheduling, which are performed real-time;
2. use simulation modeling to analyze the dispatching rule performance.

Dynamically changing dispatching rules to account for these unforeseen changes, such as variable job arrival patterns, urgent jobs, or unplanned machine downtime, has been studied throughout the years (e.g., Jeong and Kim 1998, Min and Yih 2003, Upsani and Uzsoy 2008). For each significant disruption, the dispatching rules are analyzed and the best fit is selected.

Some authors' works involve testing different rules in order to determine the rules that provide the best performance (e.g., Joeng and Kim 1998). Other authors propose a more complex solution in which dispatching rules are adjusted in order to meet pre-determined acceptable levels in the performance metrics (e.g., Min and Yih 2003). Upsani and Uzsoy (2008) focus on how the dispatching rules impact the lateness of the product. Ishii and Talavage (1991) use simulation to analyze dispatching rules to identify the rule that provides the best solution to reduce work-in-process inventory, reduce costs and improve profit in a flexible manufacturing system. The authors assert that a dispatching rule selected at different periods in time greatly improves production performance. They assert that no one dispatching rule produces optimal performance in all cases.

2.5 Minimizing the Earliness and Tardiness Performance Objectives

2.5.1 Approaches for Minimizing Tardiness

It is reported that the single machine total tardiness problem is NP-hard in the ordinary sense, and its weighted counterpart is NP-hard in the strong sense (Pinedo 2002). As the tardiness problem increases in complexity, the tractability of the problem does not improve. Therefore, researchers and practitioners pursue approximate approaches to address this performance objective.

A successful and popular heuristic approach to addressing the tardiness objective is the Apparent Tardiness Cost (ATC) sequencing heuristic proposed by Rachamadugu and Morton

(1981). ATC calculates an index value for each job i in a set of jobs to be processed and arranges those jobs in decreasing order ATC index value. ATC works in a similar fashion as the weighted shortest processing time dispatching rule that processes a set of jobs in increasing order of the w_i / p_i index value, where w_i is the weight (or importance) of job i and p_i is the processing time of job i . However, ATC considers the “apparent cost” of the remaining jobs awaiting processing and scales the slack in accordance with the remaining number of competing jobs awaiting processing. The calculation of the ATC index value for job i is as follows:

$$ATC(i) = \frac{w_i}{p_i} \exp\left(-\left[\frac{[d_i - p_i - t]^+}{kp}\right]\right) \quad (2.1)$$

where:

p_i processing time for job i

d_i due date for job i

p average processing time of the waiting jobs

w_i weight of job i

k look-ahead parameter (user-defined for the problem at hand)

The k parameter is a look-ahead parameter related to the number of competing jobs, and k is a decision parameter whose specific value is determined by the user for the scheduling problem at hand. Appropriate values of the k -value range from $1.5 \leq k \leq 4.5$, according to the empirical analyses of Ow (1985) and Rachamadugu (1982).

Vepsalainen and Morton (1987) evaluate the ATC rule against several standard rules including First Come First Serve (FCFS), Earliest Due Date (EDD), Slack per Remaining

Processing Time (S/RPT), Weighted Shortest Processing Time (WSPT), and Weighted Cost over Time (COVERT), in a job shop environment with 10 machines and jobs arriving at a continuous Poisson arrival rate. ATC is shown to outperform the standard rules and the COVERT dispatching rule. In fact, Rachamadugu and Morton (1981) and Vepsalainen and Morton (1987) find that ATC outperforms several other heuristics for the tardiness and weighted tardiness objectives. It is found to generate near-optimal performance for single machine scheduling problems when tardiness and weighted tardiness are the performance objectives.

2.5.2 Approaches for Minimizing Earliness-Tardiness

Baker and Scudder (1990) discuss that there are several formulations of the earliness-tardiness scheduling problem. One formulation is that jobs have a common due date (see, for example, Gordon et al. 2002). Another formulation is that each job has a distinct due date, which is a generalization of the previous formulation. Yet, another formulation of the earliness-tardiness scheduling problem is to derive a due date that will ultimately help to optimize the sequence of jobs with respect to both objectives simultaneously. Each of these types of problems requires different solutions approaches in order to minimize earliness-tardiness.

For instance, Wan and Yen (2002) develop a tabu search method for the single machine problem with individual jobs having distinct due dates, and jobs are assumed to be all available at time $t = 0$. In order to consider the earliness portion of the objective, they allow for idle time to be inserted into the master production schedule. Their problem allows for time windows in which the job due dates fall. Batching, incompatible job families, sequence-dependent setups are

not considered. The authors' solution to this problem involves two algorithms: optimal timing and optimal sequencing. The optimal timing algorithm identifies the best completion time that corresponds to a particular job's due window. Forced idle time is inserted and jobs are shifted in order for their completion time to fall within their respective time windows.

Mazzini and Armentano (2001) approach the problem by proposing a heuristic in which idle times are inserted during the construction of the schedule rather than determining a schedule using a priority rule and then inserting idle times in order to optimize the schedule. The authors are able to compare the results of the heuristic against optimal solutions but only for 12 jobs or less. The authors show that it is computationally-expensive to reach a solution for more than 12 jobs. Therefore, their heuristic is compared to other known heuristic priority rules for problems greater than 12 jobs. Their results suggest that, if the job is going to be inevitably tardy, then the job would start at its ready time r_i ; however, if the job is early, the start would be the difference between the due date d_i and its processing time p_i . After the start time s_i is determined for each job i , the heuristic looks to see if any jobs have overlapping times, in which the feasibility procedure of their heuristic shifts jobs either to the left or to the right to ensure no overlapping and reduce overall earliness-tardiness. Their heuristic performs well when compared to the optimal solution and a benchmark set of standard heuristic priority rules.

Rabadi et al. (2004), Rabadi et al. (2006), and Sourd (2005) all address scheduling problems with sequence-dependent setups. Both the research of Rabadi et al. (2004) and Sourd (2005) focuses on a scheduling problem with one machine and no preemptions; however, Rabadi et al. (2004) assumes one common due date for all jobs and attempts to minimize the total

amount of earliness and tardiness, while Sourd (2005) assigns each job a due date and works to minimize earliness-tardiness and setup costs. Rabadi et al. (2004) attempts to minimize overall earliness and tardiness by scheduling jobs using the longest processing time first (to minimize earliness) until the due date is approached. Once jobs are considered tardy, shortest processing time is used to schedule the remaining jobs in order to minimize tardiness. In order to make the scheduling problem more manageable, Rabadi et al. (2004) sum the processing and setup times in a matrix as the adjusted processing time or AP_{ij} , where $AP_{ij} = S_{ij} + P_j$. The authors utilize a branch and bound algorithm and compare their solutions to those generated from a mixed integer programming (MIP) model by proposed by Coleman (1992). Sourd (2005) use a similar approach in that a MIP model is formulated, and then use a branch and bound algorithm approach to solve the MIP. Sourd (2005) also proposes a heuristic to solve the problem, and the solutions generated by the proposed heuristic are compared to those generated from the branch and bound algorithm. It is found that the branch and bound algorithm performs well; however, it is only suitable when the number of jobs is limited to less than 20. For larger-scale scheduling problems where the number of jobs is greater than or equal to 20, the heuristic approach is more suitable.

2.6 Summary

As concluded by most researchers in the existing literature, job shop scheduling problems in which batching and sequence-dependent setups are present are NP-hard. Therefore, in order to provide a solution to this problem, heuristic solution methods are pursued. There are many solution approaches to preventing late deliveries to customers considering some of these problem characteristics. However, currently there is no research regarding the scheduling problem encompasses all of these problem characteristics as described and defined in Section 1.2.

CHAPTER 3 : RESEARCH APPROACH

3.1 Introduction

Job shop scheduling with batching, incompatible job families, and sequence-dependent setups have all been addressed independently or in some combination. However, the open literature shows that there is no current work that identifies a methodology to solve the problem when all three features are present simultaneously and while minimizing both earliness and tardiness. As a result of the computational complexity results for this (and variants of this) problem, heuristics are the desired approach to solve this problem.

Recall that an objective of this research investigation is to formulate a special case of the general problem. Specifically, this research focuses on the effective scheduling of the bottleneck workstation, which would, in turn, improve the overall performance of the entire production system. In order to do this, the general multi-workstation problem is simplified through a series of assumptions that are systematically relaxed in a three-phased approach. The final phase involves modeling the problem $1 \mid r_i, \text{batch}, s_{ij} \mid \sum (\theta_1 E_i + \theta_2 T_i)$. The following three sections detail the relevant assumptions of each phase.

3.2 Phase 1: Single Batch Workstation with Sequence-Independent Setups and Equal Job Weights

Phase 1 of this research involves the following problem with the simplifying assumptions.

- The number of workstations m is 1, i.e., $|\mathbf{W}| = 1$;
- The number of machines in the workstation is 1;
- The number of jobs n arrives individually to the workstation and are placed in batches before being processed;
- There is no restriction on queue length for jobs and batches awaiting processing at the workstation;
- Jobs within a batch are processed one at a time at the workstation;
- Jobs depart the workstation in batches;
- No batch preemption. Once a batch has begun processing at a workstation, it cannot be interrupted due to the arrival of higher priority jobs (or batches of jobs) before the processing of that batch has been completed;
- No batch splitting. Once a batch has been created, it cannot be divided into smaller sub-batches of jobs. It is treated as a single entity;
- Incompatible job families. Jobs from different families cannot be processed together in the same batch;
- Machine setups occur between processing batches of jobs, regardless of the job family process prior; in other words, a machine's setup time for a batch from a family is

determined by that batch's product family not by the previous family for which the machine is currently setup, i.e., machine setups are batch sequence-independent; and

- The earliness penalty per unit time per job and the tardiness penalty per unit time per job are equal.

The assumptions for this phase characterize a scheduling problem similar to that addressed by Uzsoy (1995). One primary difference between the simplified problem of Phase 1 and that of Uzsoy (1995) is the scheduling performance criterion. While Uzsoy (1995) focuses on minimizing C_{\max} (or makespan) and weighted completion time $\sum w_i C_i$, the scheduling performance objective in this phase is to minimize $\sum (E_i + T_i)$, one that has not yet been addressed in the open literature for the batch scheduling problem in the presence of incompatible job families and dynamically arriving jobs. This simplified scheduling problem is expressed in the scheduling notation as follows: $1 \mid r_i, \text{batch} \mid \sum (E_i + T_i)$.

3.3 Phase 2: Single Batch Workstation with Sequence-Independent Setups and Unequal Weights

Phase 2 of this research involves the following problem and modeling assumptions. The relaxed assumption is italicized.

- The number of workstations m is 1, i.e., $|\mathbf{W}| = 1$;
- The number of machines in the workstation is 1; therefore, $m = 1$;

- The number of jobs n arrives individually to the workstation and are placed in batches before being processed;
- There is no restriction on queue length for jobs and batches awaiting processing at the workstation;
- Jobs within a batch are processed one at a time at the workstation;
- Jobs depart the workstation in batches;
- No batch preemption. Once a batch has begun processing at a workstation, it cannot be interrupted due to the arrival of higher priority jobs (or batches of jobs) before the processing of that batch has been completed;
- No batch splitting. Once a batch has been created, it cannot be divided into smaller sub-batches of jobs. It is treated as a single entity;
- Incompatible job families. Jobs from different families cannot be processed together in the same batch;
- Machine setups occur between processing batches of jobs, regardless of the job family process prior; in other words, a machine's setup time for a batch from a family is determined by that batch's product family not by the previous family for which the machine is currently setup, i.e., machine setups are batch sequence-independent; and
- *The earliness penalty per unit time per job and the tardiness penalty per unit time per job are unequal.*

This enhanced scheduling problem through the relaxation of the earliness/tardiness unit weighting is expressed in the scheduling notation as follows: $1 \mid r_i, \text{batch} \mid \sum (\theta_1 E_i + \theta_2 T_i)$. The rationale of experimenting with unequal weights is to understand if the proposed heuristics are robust under unequal weights.

3.4 Phase 3: Single Batch Workstation with Sequence-Dependent Setups and Unequal Job Weights

Phase 3 of this research involves the following problem and modeling assumptions. The relaxed assumptions are italicized.

- The number of workstations m is 1, i.e., $|\mathbf{W}| = 1$;
- The number of machines in the workstation is 1; therefore, $m = 1$;
- The number of jobs n arrives individually to the workstation and are placed in batches before being processed;
- There is no restriction on queue length for jobs and batches awaiting processing at the workstation;
- Jobs within a batch are processed one at a time at the workstation;
- Jobs depart the workstation in batches;
- No batch preemption. Once a batch has begun processing at a workstation, it cannot be interrupted due to the arrival of higher priority jobs (or batches of jobs) before the processing of that batch has been completed;

- No batch splitting. Once a batch has been created, it cannot be divided into smaller sub-batches of jobs. It is treated as a single entity;
- Incompatible job families. Jobs from different families cannot be processed together in the same batch;
- *A machine's setup time for a batch from a family is determined not only by that batch's product family but also by the previous family for which the machine is currently setup; in other words, machine setup are family sequence-dependent;*
- *Major machine setups occur between processing batches of jobs from different families;*
- *Minor machine setups occur between processing batches of jobs from the same family;*
- The earliness penalty per unit time per job and the tardiness penalty per unit time per job are unequal.

This even further relaxed scheduling problem is expressed in the scheduling notation as follows:

$1 \mid r_i, \text{batch}, s_{ij} \mid \sum (\theta_1 E_i + \theta_2 T_i)$. This problem now addresses sequence-dependent setups for the single batch processor scheduling problem minimizing the earliness/tardiness objective with unequal weights.

CHAPTER 4 : MATHEMATICAL MODEL FORMULATION AND COMPUTATIONAL PERFORMANCE REQUIREMENTS

4.1 Phase 1: Single Batch Machine Scheduling with Sequence-Independent Setups and Equal Earliness and Tardiness Penalties

Recall from CHAPTER 3 that Phase 1 involves a single batch workstation with sequence-independent setups and equal job weights and involves the following simplifying and modeling assumptions:

- The number of workstations m is 1, i.e., $|\mathbf{W}| = 1$;
- The number of machines in the workstation is 1;
- The number of jobs n arrives individually to the workstation and are placed in batches before being processed;
- There is no restriction on queue length for jobs and batches awaiting processing at the workstation;
- Jobs within a batch are processed one at a time at the workstation;
- Jobs depart the workstation in batches;
- No batch preemption. Once a batch has begun processing at a workstation, it cannot be interrupted due to the arrival of higher priority jobs (or batches of jobs) before the processing of that batch has been completed;
- No batch splitting. Once a batch has been created, it cannot be divided into smaller sub-batches of jobs. It is treated as a single entity;

- Incompatible job families. Jobs from different families cannot be processed together in the same batch;
- Machine setups occur between processing batches of jobs, regardless of the job family process prior; in other words, a machine's setup time for a batch from a family is determined by that batch's product family not by the previous family for which the machine is currently setup, i.e., machine setups are batch sequence-independent; and
- The earliness penalty per unit time per job and the tardiness penalty per unit time per job are equal.

Before presenting the general model formulation of this problem, the relevant notation and parameters are first presented.

Sets

J : Set of jobs to be processed

F : Set of product families

M : Set of machines

Parameters

n : Number of jobs, where $n = |\mathbf{J}|$

m : Number of machines, where $m = |\mathbf{M}|$

F : Number of families, where $F = |\mathbf{F}|$

- i : Index for jobs, where $i = 1, \dots, n$
 j : Index for machines, where $j = 1, \dots, m$
 f : Index for families, where $f = 1, \dots, F$
 r_i : Release time of job i
 d_i : Due date of job i
 w_i : Weight or importance of job i ; the jobs are equally-weighted, i.e., $w_i = 1$
 θ_{1i} : Earliness penalty of job i
 θ_{2i} : Tardiness penalty of job i
 p_f : Common processing time of a job of family f
 s_f : Setup time for family f
 B_{\min}^f : Minimum batch size of family f
 B_{\max}^f : Maximum batch size of family f
 L_{fi} : 1, if family f belongs to job i ; otherwise 0

Decision Variables

- X_{ik} : 1, if job i is processed in batch k ; otherwise 0.
 Y_{fk} : 1, if family f is processed in batch k ; otherwise 0

Dependent Variables

- C_i : Completion time of job i

- E_i : Earliness of job i
 T_i : Tardiness of job i
 t_k : Start time of batch k
 pb_k : Process time of batch k of family f
 Cb_k : Completion time of batch k

The following is the proposed mathematical model formulation:

Minimize

$$\sum_{i=1}^n (\theta_1 E_i + \theta_2 T_i) \quad (4.1)$$

s.t.

$$E_i \geq d_i - C_i \quad \forall i \quad (4.2)$$

$$T_i \geq C_i - d_i \quad \forall i \quad (4.3)$$

$$C_i + E_i - T_i = d_i \quad \forall i \quad (4.4)$$

$$\sum_{i=1}^n X_{ik} \geq \sum_{f=1}^F (Bmin_f \times Y_{fk}) \quad \forall f; \forall k \quad (4.5)$$

$$\sum_{i=1}^n X_{ik} \leq \sum_{f=1}^F (Bmax_f \times Y_{fk}) \quad \forall f; \forall k \quad (4.6)$$

$$pb_k = \sum_{i=1}^n (p_i \times X_{ik}) + \sum_{f=1}^F (s_f \times Y_{fk}) \quad \forall f; \forall k \quad (4.7)$$

$$\sum_{k=1}^n X_{ik} = 1 \quad \forall i; \quad \forall k \quad (4.8)$$

$$\sum_{f=1}^F Y_{fk} \leq 1 \quad \forall f; \quad \forall k \quad (4.9)$$

$$L_{fi} \times X_{ik} \leq Y_{fk} \quad \forall i; \quad \forall f; \quad \forall k \quad (4.10)$$

$$t_k \geq r_i X_{ik} \quad \forall i; \quad \forall k \quad (4.11)$$

$$t_k \geq Cb_{k-1} \quad \forall k \quad (4.12)$$

$$Cb_k = t_k + pb_k \quad \forall k \quad (4.13)$$

$$C_i = Cb_k X_{ik} \quad \forall i; \quad \forall k \quad (4.14)$$

$$E_i \geq 0 \quad \forall i \quad (4.15)$$

$$T_i \geq 0 \quad \forall i \quad (4.16)$$

$$r_i \geq 0 \quad \forall i \quad (4.17)$$

$$Cb_k \geq 0 \quad \forall k \quad (4.18)$$

$$C_i \geq 0 \quad \forall i \quad (4.19)$$

$$X_{ik} \in \{0,1\} \quad \forall i; \quad \forall k \quad (4.20)$$

$$Y_{fk} \in \{0,1\} \quad \forall f; \quad \forall k \quad (4.21)$$

Objective (4.1) minimizes the total tardiness and earliness over all jobs. Constraints (4.2) and (4.3) are the definitions of the earliness and tardiness measures of a job. Constraint (4.4) shows the definition of the due date for each job. Constraint (4.5) states that the sum of all jobs in a batch is greater than the minimum family batch size constraint for that batch. Constraint

(4.6) states that the sum of jobs in a batch is less than the maximum family batch size constraint for that batch. Constraint (4.7) defines the process time of a batch of jobs for the same family. This is the number in the batch times the individual processing time of each job plus the setup time associated for each family. Constraint (4.8) says that each job is assigned to a batch only once. Constraint (4.9) states that each family is assigned to exactly one batch. If there are no jobs in the batch then no families would be assigned to a batch. Constraint (4.10) ensures that if a job in a family is assigned to a batch, then that batch must be of the same family (i.e., if $X_{ik} = 1$, then $Y_{jk} = 1$). Constraint (4.11) ensures that the start time of a batch must be no less than the release time of any jobs forming that batch. Constraint (4.12) specifies that no job can start before the completion of the previous batch at a machine. Constraint (4.13) identifies that a job's start time must be early enough to allow for the processing time before it is completed. Constraint (4.14) states that the completion time of the job is equal to the completion time of the batch to which the job belongs. This is because jobs depart in batches. Constraints (4.15) through (4.19) are the non-negativity constraints. Constraints (4.20) and (4.21) are binary constraints for the decision variables.

4.2 Computational Requirements of the Single Batch Machine Scheduling Problem with Sequence-Independent Setups and Equal Earliness and Tardiness Penalties

The general integer programming model in Section 4.1 is solved using IBM ILOG CPLEX version 12.5 on a 2.13 GHz Intel Pentium CPU Processor with 4.0 GB RAM and Microsoft Windows 7 operating system. Table 4.1 summarizes the computational time of the

CPLEX solver. It can be seen from Table 4.1 that the solution time increases exponentially, even for the simpler single batch workstation scheduling problem under the various operating conditions, including tight and loose due dates and high and low incoming job traffic intensity at the workstation.

Table 4.1. Summary of CPLEX solution times in order to reach an optimal solution.

Problem Characteristic	Value	5 Jobs Instance		10 Jobs Instance		15 Jobs Instance	
		% Nodes Tested	Solution Time	% Nodes Tested	Solution Time	% Nodes Tested	Solution Time
# of Families	1	100%	< 2 Secs	100%	< 10 Secs	100%	< 10 Secs
Due Dates	Tight / Loose						
Traffic Intensity	High / Low						
# of Families	5	100%	< 2 Secs	100%	< 10 Secs	100%	0.42 Hrs - 18 Hrs
Due Dates	Tight / Loose						
Traffic Intensity	High / Low						
# of Families	1	100%	< 2 Secs	100%	1-5 Days	100%	8-10 Hrs
Due Dates	Tight / Loose						
Traffic Intensity	All Jobs Available Simultaneously						
# of Families	5	100%	< 2 Secs	100%	2-6 Days	4%	>12 Hrs
Due Dates	Tight / Loose						
Traffic Intensity	All Jobs Available Simultaneously						

CHAPTER 5 : HEURISTIC SOLUTION PROCEDURES

5.1 Introduction

One of the most effective heuristics in research and practice for addressing the tardiness problem (and its weighted counterpart) in several scheduling environments with various processing characteristics is the Apparent Tardiness Cost (ATC) heuristic (Vepsalainen and Morton, 1987). This heuristic works well for tardiness, but does not consider the earliness measure. In this research, the ATC heuristic is enhanced so that it considers both the tardiness and the earliness measure.

For the single batch machine scheduling problem with sequence-independent setups minimizing total earliness and tardiness objective with equal weights, $\sum(E_i + T_i)$, two solution heuristics are proposed, namely, (1) the Apparent Tardiness and Earliness Cost (ATEC) heuristic and (2) the Batching Incompatible Families with Earliness Tardiness Cost (BIFET) heuristic.

5.2 Proposed Solution Heuristics

5.2.1 Apparent Tardiness and Earliness Cost (ATEC) Heuristic

The steps of the proposed Apparent Tardiness and Earliness Cost (ATEC) heuristic are as follows.

1. Obtain a sequence for the available jobs awaiting processing at the batch machine using pure ATC heuristic (Vepsalainen and Morton 1987);
2. Construct sequential batches by adding awaiting jobs of the same family to a batch until the maximum batch size is reached;
3. When the maximum batch size is reached, then start a new batch with the remaining unbatched jobs;
4. Identify the start time of each job and set this time equal to s_i .
5. Create an initial schedule and calculate the completion time $C_i = p_i \times B_f + s_i$;
6. Calculate earliness-tardiness of each job i using

$$E_i = d_i - C_i \quad \forall i$$

$$T_i = C_i - d_i \quad \forall i$$

7. Count the number of early jobs, and set z equal to the number of early jobs; Set $j = 1$;
8. Start with the job with the highest earliness penalty;
9. Calculate the earliness, and adjust start time s_i by that amount $s_{i(\text{new})} = ra_i + d_i - C_i$;
Increment j ;
10. Recalculate the earliness/tardiness penalty for all batched jobs;
 - a. If it is improved, keep new start time; Go to Step 2; and
 - b. If it is not improved, set start time of the batch back to original start time; Go to Step 6, and identify the job with next highest earliness penalty;
11. When $j = z$, stop.

ATEC identifies the batch size in order to improve the overall objective function. All available jobs of the same family are batched together regardless if this degrades the total earliness and tardiness penalty. This does not align with the tardiness objective of the scheduling problem because it causes jobs to be batched, which causes tardy (or near tardy) jobs to be even tardier because those jobs are waiting for other jobs in the batch to be processed before they can be processed. To address this, the proposed ATEC heuristic is modified in order to improve how batches are formed. Batching Incompatible Families with Earliness Tardiness (BIFET) includes enhancements to ATEC.

5.2.2 Batching Incompatible Families with Earliness Tardiness Cost (BIFET) Heuristic

Before the steps of the Batching Incompatible Families with Earliness Tardiness Cost (BIFET) heuristic are presented, it is necessary to discuss the Family Batch Constraint parameter.

5.2.2.1 The Family Batch Constraint (FBC) Parameter

The FBC is a user-defined parameter. It allows the user to expand or constrain the number of jobs of the same family to be batched together. Variable j is assigned to the job with the highest ATC value awaiting processing. If another waiting job's ATC value of the same family falls within plus or minus j 's ATC value, then the job is batched with job j .

The steps of the proposed BIFET heuristic are as follows.

1. Obtain a sequence for the available jobs awaiting processing at the batch machine using pure ATC heuristic (Vepsalainen and Morton 1987);
2. Jobs are batched based on family and according to the FBC parameter;
3. Then, the start times of the jobs are identified and set equal to s ;
4. Construct sequential batches by adding jobs in the same family to a batch until the maximum batch size is reached;
5. When the maximum batch size is reached, then start a new batch with the remaining unbatched jobs;
6. Identify start time of each job i and set it equal to s_i ;
7. Create initial schedule and calculate completion time of each job i as $C_i = p_i \times B_f + s_i$;
8. Calculate earliness-tardiness penalties of each job i using

$$E_i = d_i - C_i \quad \forall i$$

$$T_i = C_i - d_i \quad \forall i$$

9. Count number of early jobs and set z equal to number of early jobs; Set $j = 1$;
10. Start with the job with the highest earliness penalty; Calculate earliness and adjust start time s_i by that amount $s_{i(\text{new})} = ra_i + d_i - C_i$; Increment j ;
11. Recalculate the earliness/tardiness penalty for all batched jobs;
 - a. If it is improved, keep the new start time; Go to Step 2; and

- b. If it is not improved, set start time of the batch back to original start time; Go to Step 8, and identify the job with next highest earliness penalty.

The results from the mathematical model in CHAPTER 4 using CPLEX are used as benchmark results for the heuristics in order to evaluate the quality of their performance. The percent error (or deviation) from the CPLEX solution results is calculated using Eq. 5.1.

$$\%Error = \left[\frac{\Phi(e) - \Phi(e^*)}{\Phi(e^*)} \right] \times 100 \quad (5.1)$$

where $\Phi(e)$ is the solution value of the heuristic tested, and $\Phi(e^*)$ denotes the value from CPLEX.

5.3 Computational Study

It is important to note that the value of the look-ahead parameter k for any ATC-based heuristic is somewhat problem-dependent. Past research suggests that the value of k should be set to a value from 1.5 to 4.5 (i.e., $1.5 \leq k \leq 4.5$). For the purpose of this research investigation, the most appropriate k -value for the problem characteristics being considered is determined using what can be considered a worst-case scheduling scenario, i.e., when all jobs to be processed arrive simultaneously from multiple job families and with tight due dates. This creates a high competition environment at the workstation. Thirty randomly-generated test problems with these conditions are run and the ATC look-ahead parameter k is varied from 1.5 to 4.5 at 0.5 increments, and the $\sum(E_i + T_i)$ objective value is recorded for each k -value and the k -value with

the best performance value is noted. The best performance results are achieved using a k -value of 4.5. Therefore, the look-ahead parameter k is set to 4.5 for all ATC-based heuristics, including the two proposed heuristics.

5.3.1 Benchmark Heuristics

The assumptions for this phase of the research characterize a scheduling problem similar to that addressed by Uzsoy (1995). Therefore, the proposed solution heuristics are benchmarked with heuristics proposed by Uzsoy (1995), which are shown to exhibit remarkable performance for single batch machine scheduling problems with incompatible families. One primary difference between the simplified problem of Phase 1 and that of Uzsoy (1995) is the scheduling performance objective. While Uzsoy (1995) focuses on minimizing C_{\max} (or makespan), maximum lateness (L_{\max}), and weighted completion time $\sum w_i C_i$, the scheduling performance objective in this phase is to minimize $\sum (E_i + T_i)$.

Uzsoy (1995) proposes various scheduling heuristics such as Family Earliest Due Date (FEDD) and Critical Job (CRIT) in order to minimize maximum lateness (L_{\max}). Since the problem is similar to the one addressed in this research, FEDD and CRIT are used for comparison against the two proposed heuristics – ATEC and BIFET. In addition, pure ATC (which is described in CHAPTER 2) is used as a benchmark heuristic. As FEDD and CRIT are briefly described here, the reader should refer to Uzsoy (1995) for the details of the two heuristic approaches.

5.3.1.1 Family Earliest Due date (FEDD)

FEDD begins when a machine becomes available and jobs are awaiting processing at the machine. When a machine becomes available and jobs are waiting, the job with the earliest due date is selected to be processed. If there are other jobs in the queue of the same family as the job selected, those jobs can be batched and processed with that job. B_{\max}^f is the maximum number of jobs of family f that can be batched together.

5.3.1.2 Critical Job (CRIT)

CRIT begins when a machine becomes available and jobs are awaiting processing at the available machine. The following are the steps of the CRIT heuristic, which begin with a schedule generated using the FEDD heuristic.

Step 1: Develop a schedule using the FEDD algorithm;

Step 2: Identify the critical job c and the job with L_{\max} , Job j ;

- If there is no critical job, stop.
- Otherwise, set $r_c = r_j$, and go to Step 1.

A critical job is defined as a job in the same busy period with a due date greater than the job with L_{\max}

All solutions generated from the benchmark heuristics are also compared to the solutions generated by CPLEX.

5.3.2 Experimental Design

The proposed and benchmark heuristics are compared in an empirical study using a set of randomly-generated problem instances. The number of jobs in the study is set to three levels – 5, 10, and 15. A family is randomly assigned to a job using an empirical probability distribution when each integer value from 1 to F , where F is the number of families, is equally-likely to occur. Job due dates to the customers are calculated using Eq. 5.2.

$$d_i = r_i + \beta(p_i) \quad (5.2)$$

The multiplicative factor β is used to control the tightness or looseness of the due dates. There are two cases that are considered in this computational study. Tight due dates are calculated setting β as a Uniform distribution $U(0, 2)$. Loose due dates are calculated setting β as a Uniform distribution $U(0, 4)$. A similar approach to Geiger and Uzsoy (2008) that uses a desired level of average machine utilization is used to assign release time to the jobs. These job release times dictate the level of traffic intensity (or competition) at the machine resource. The desired average utilization of the batch machine is set to three levels – 50%, 95% and 100%. The experimental design is summarized in Table 5.1.

Table 5.1. Experimental design for phase 1: single batch workstation with sequence-independent setups and equal job weights.

Factor	Values	Number of Levels
Number of Jobs, n	5, 10, 15	3
Number of Families, F	1, $U(1,5)$	2
Traffic Intensity	<ul style="list-style-type: none"> • Low: 50% Machine Utilization • Medium: 75% Machine Utilization • High: 95% Machine Utilization • Highest: All Jobs Arrive Simultaneously 	4
Due Date Tightness	Tight: $U(0,2)$, Loose: $U(0,4)$	2
Min Batch size	1	1
Max Batch Size	10	1
Replications per Problem		10

5.4 Discussion of Computational Results

Solving the problem instances using IBM ILOG CPLEX is run to completion (i.e., 100% of the branch and bound nodes tested for problems with 5 and 10 jobs). The time for CPLEX to run to completion for 10 jobs ranged between 30 seconds (for the smaller problems) and five days (for the larger problems). In order to effectively run all 240 problem instances for the 15-job problem, run time is kept to reasonable levels. All heuristics are coded in C++ and Bloodshed Dev-C++ compiler.

Table 5.2 through Table 5.4 summarizes the percent error results for the benchmark ATC, FEDD, CRIT heuristics and the proposed ATEC and BIFET heuristics for the various job sizes with tight due dates. Recall that the solutions produced by the heuristics are compared to those from CPLEX. It can be seen that the BIFET outperforms FEDD and CRIT in almost every category, with the exception of when all jobs are released simultaneously. As Table 5.2 through

Table 5.4 show, BIFET outperforms all other heuristics with respect to the deviation from the CPLEX solutions, regardless of traffic intensity when the due dates are tight.

Table 5.2. Performance results for 5 jobs with tight due dates results.

Tight Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	48.09%	11.40%	218.62%	24.28%	31.03%	68.92%	51.69%	219.38%
	Max	50.35%	18.64%	428.95%	87.01%	115.79%	307.14%	155.56%	1250.00%
CRIT	Average	48.09%	11.40%	218.62%	28.41%	31.03%	71.39%	51.69%	219.38%
	Max	50.35%	18.64%	428.95%	87.01%	115.79%	307.14%	155.56%	1250.00%
ATC	Average	48.09%	7.45%	218.62%	24.28%	31.03%	68.92%	51.69%	219.38%
	Max	50.35%	15.04%	428.95%	87.01%	115.79%	307.14%	155.56%	1250.00%
ATEC	Average	48.09%	6.80%	0.00%	2.46%	0.64%	5.86%	0.00%	0.00%
	Max	50.35%	15.04%	0.00%	10.00%	3.81%	39.28%	0.00%	0.00%
BIFET	Average	11.47%	2.53%	0.00%	2.46%	0.64%	5.86%	0.00%	0.00%
	Max	28.76%	5.73%	0.00%	10.00%	3.81%	39.29%	0.00%	0.00%

Table 5.3. Performance results for 10 jobs with tight due date results.

Tight Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	60.45%	15.07%	11.28%	15.24%	40.49%	34.50%	61.68%	37.73%
	Max	63.04%	36.36%	26.06%	54.32%	108.93%	88.19%	233.33%	91.19%
CRIT	Average	60.45%	15.07%	22.54%	24.99%	40.49%	39.39%	61.68%	37.73%
	Max	63.04%	36.36%	93.66%	57.92%	108.93%	88.19%	233.33%	91.19%
ATC	Average	60.45%	6.65%	11.28%	14.69%	40.49%	34.50%	61.68%	37.73%
	Max	63.04%	17.26%	26.06%	54.32%	108.93%	88.19%	233.33%	91.19%
ATEC	Average	60.45%	6.65%	3.02%	4.29%	1.03%	5.63%	0.00%	0.00%
	Max	63.04%	17.26%	12.72%	11.63%	7.14%	22.92%	0.00%	0.00%
BIFET	Average	11.41%	3.11%	3.02%	4.29%	1.03%	5.63%	0.00%	0.00%
	Max	25.17%	6.41%	12.72%	11.63%	7.14%	22.92%	0.00%	0.00%

Table 5.4. Performance results for 15 jobs with tight due dates.

Tight Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	43.28%	16.94%	10.79%	6.66%	43.69%	30.54%	55.66%	73.00%
	Max	46.77%	635.71%	30.24%	32.87%	86.63%	90.42%	144.12%	195.88%
CRIT	Average	43.28%	185.40%	16.07%	15.87%	43.69%	33.10%	55.66%	66.47%
	Max	46.77%	635.71%	61.95%	47.88%	86.63%	90.42%	144.12%	154.64%
ATC	Average	67.99%	185.40%	10.79%	6.66%	43.69%	30.54%	55.66%	73.00%
	Max	74.74%	635.71%	30.24%	32.87%	86.63%	90.42%	144.12%	195.88%
ATEC	Average	72.07%	3.83%	3.70%	1.80%	1.26%	2.65%	0.00%	13.60%
	Max	79.98%	16.81%	11.19%	5.84%	5.81%	6.59%	0.00%	32.99%
BIFET	Average	7.80%	3.83%	3.70%	1.80%	1.26%	2.65%	0.00%	11.79%
	Max	20.25%	16.81%	11.19%	5.84%	5.81%	6.59%	0.00%	32.99%

Table 5.5, Table 5.6 and Table 5.7 summarize the performance of the heuristics under loose due date conditions. The proposed ATEC and BIFET heuristics outperform the benchmark heuristics in every category. BIFET outperforms ATEC when there is high traffic intensity (i.e., jobs arrive simultaneously and high traffic). However, they perform the same when due dates are loose and traffic is low, which is intuitive since there is low competition among the jobs awaiting processing at the single batch workstation. The earliness portion of both heuristics helps improve the performance when jobs are early; however, it is difficult to effectively determine the best start time for the jobs that are early. For the high maximum error percentage in the low traffic, the majority of jobs tend to be early. However, there are some problem instances in which adjusting the start time of the jobs with the difference between the due date and completion time is excessive and causes the start time of the other jobs to be delayed, and, therefore, results in a larger percent error from the CPLEX solutions.

Table 5.5. Performance results for 5 jobs with loose due dates.

Loose Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	87.82%	20.77%	186.24%	106.49%	187.28%	119.42%	218.62%	192.61%
	Max	144.68%	54.60%	435.48%	326.09%	605.41%	330.43%	428.95%	480.77%
CRIT	Average	87.82%	20.77%	151.23%	100.27%	187.28%	120.53%	218.62%	193.99%
	Max	144.68%	54.60%	435.48%	326.09%	605.41%	330.43%	428.95%	480.77%
ATC	Average	87.82%	18.19%	186.24%	106.49%	187.28%	119.42%	218.62%	169.51%
	Max	144.68%	54.60%	435.48%	326.09%	605.41%	330.43%	428.95%	480.77%
ATEC	Average	87.82%	10.68%	32.02%	12.02%	17.46%	10.68%	0.00%	3.43%
	Max	144.68%	27.33%	89.74%	33.33%	42.31%	19.86%	0.00%	32.69%
BIFET	Average	7.09%	5.35%	23.24%	10.76%	15.03%	8.25%	0.00%	3.43%
	Max	36.17%	13.19%	74.36%	33.33%	37.97%	19.86%	0.00%	32.69%

Table 5.6. Performance results for 10 jobs with loose due dates.

Loose Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	71.01%	18.05%	127.88%	65.32%	160.80%	126.88%	345.96%	328.75%
	Max	88.78%	38.36%	242.00%	144.44%	347.17%	314.96%	1295.00%	994.23%
CRIT	Average	71.01%	85.22%	108.97%	69.62%	160.80%	128.53%	345.96%	328.75%
	Max	88.78%	994.23%	186.62%	120.29%	347.17%	314.96%	1295.00%	994.23%
ATC	Average	71.01%	13.89%	127.88%	65.32%	160.80%	126.88%	345.96%	328.75%
	Max	88.78%	38.36%	242.00%	144.44%	347.17%	314.96%	1295.00%	994.23%
ATEC	Average	71.01%	13.58%	35.17%	14.11%	27.81%	17.70%	1.02%	9.23%
	Max	88.78%	38.36%	90.00%	38.58%	48.60%	43.56%	7.63%	44.23%
BIFET	Average	5.05%	5.96%	26.97%	13.67%	26.76%	17.50%	1.02%	9.23%
	Max	19.13%	14.76%	66.00%	38.58%	48.60%	43.56%	7.63%	44.23%

Table 5.7. Performance results for 15 jobs with loose due dates.

Loose Due Dates		Jobs arrive Simultaneously		High Traffic		Medium Traffic		Low Traffic	
		1 family	5 families	1 family	5 families	1 family	5 families	1 family	5 families
FEDD	Average	43.27%	18.59%	95.02%	73.91%	189.75%	133.67%	235.60%	352.57%
	Max	46.77%	25.75%	194.04%	195.88%	397.60%	306.90%	446.81%	1563.79%
CRIT	Average	43.27%	18.59%	86.31%	67.21%	189.75%	132.14%	235.60%	352.57%
	Max	46.77%	25.75%	194.04%	154.64%	397.60%	306.90%	446.81%	1563.79%
ATC	Average	65.30%	13.58%	95.02%	73.91%	189.75%	133.67%	235.60%	352.57%
	Max	69.01%	24.98%	194.04%	195.88%	397.60%	306.90%	446.81%	1563.79%
ATEC	Average	68.73%	13.58%	24.29%	14.19%	37.23%	21.86%	2.29%	11.74%
	Max	72.61%	24.98%	39.72%	32.99%	62.42%	58.62%	10.34%	46.55%
BIFET	Average	7.79%	5.82%	20.33%	12.30%	37.72%	19.79%	2.29%	11.74%
	Max	20.25%	11.13%	39.57%	32.99%	62.42%	53.88%	10.34%	46.55%

Table 5.8 through Table 5.10 show how the algorithms perform when all the jobs arrive simultaneously. In this scenario, BIFET outperforms all other heuristics, including ATEC. All the heuristics perform better when there are multiple family types, which leads to the suspicion that having smaller batch sizes, might improve the performance of the benchmark heuristics. BIFET allows the use to control the number of jobs to be batched together by selecting an appropriate FBC parameter. However, this needs to be verified because Table 5.10 shows a large amount of variability.

Table 5.8. Performance results for 5 jobs that arrive simultaneously.

Jobs Arrive Simultaneously		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	48.09%	11.40%	87.82%	20.77%
	Max	50.35%	18.64%	144.68%	54.60%
CRIT	Average	48.09%	11.40%	87.82%	20.77%
	Max	50.35%	18.64%	144.68%	54.60%
ATC	Average	48.09%	7.45%	87.82%	18.19%
	Max	50.35%	15.04%	144.68%	54.60%
ATEC	Average	48.09%	6.80%	87.82%	10.68%
	Max	50.35%	15.04%	144.68%	27.33%
BIFET	Average	11.47%	2.53%	7.09%	5.35%
	Max	28.76%	5.73%	36.17%	13.19%

Table 5.9. Performance results for 10 jobs that arrive simultaneously.

Simultaneously		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	60.45%	15.07%	71.01%	18.05%
	Max	63.04%	36.36%	88.78%	38.36%
CRIT	Average	60.45%	15.07%	71.01%	85.22%
	Max	63.04%	36.36%	88.78%	994.23%
ATC	Average	60.45%	6.65%	71.01%	13.89%
	Max	63.04%	17.26%	88.78%	38.36%
ATEC	Average	60.45%	6.65%	71.01%	13.58%
	Max	63.04%	17.26%	88.78%	38.36%
BIFET	Average	11.41%	3.11%	5.05%	5.96%
	Max	25.17%	6.41%	19.13%	14.76%

Table 5.10. Performance results for 15 jobs that arrive simultaneously.

Simultaneously		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	43.28%	16.94%	43.27%	18.59%
	Max	46.77%	635.71%	46.77%	25.75%
CRIT	Average	43.28%	185.40%	43.27%	18.59%
	Max	46.77%	635.71%	46.77%	25.75%
ATC	Average	67.99%	185.40%	65.30%	13.58%
	Max	74.74%	635.71%	69.01%	24.98%
ATEC	Average	72.07%	3.83%	68.73%	13.58%
	Max	79.98%	16.81%	72.61%	24.98%
BIFET	Average	7.80%	3.83%	7.79%	5.82%
	Max	20.25%	16.81%	20.25%	11.13%

Table 5.11 through Table 5.13 show how the algorithms perform at high traffic intensity. The ATEC and BIFET proposed heuristics, again, outperform the benchmark heuristics under the varying scheduling conditions.

Table 5.11. Performance results for 5 jobs with high traffic intensity.

High Traffic		Tight Due Dates		Loose Due Dates	
		1 family	5 families	1 family	5 families
FEDD	Average	218.62%	24.28%	186.24%	106.49%
	Max	428.95%	87.01%	435.48%	326.09%
CRIT	Average	218.62%	28.41%	151.23%	100.27%
	Max	428.95%	87.01%	435.48%	326.09%
ATC	Average	218.62%	24.28%	186.24%	106.49%
	Max	428.95%	87.01%	435.48%	326.09%
ATEC	Average	0.00%	2.46%	32.02%	12.02%
	Max	0.00%	10.00%	89.74%	33.33%
BIFET	Average	0.00%	2.46%	23.24%	10.76%
	Max	0.00%	10.00%	74.36%	33.33%

Table 5.12. Performance results for 10 jobs with high traffic intensity.

High Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	11.28%	15.24%	127.88%	65.32%
	Max	26.06%	54.32%	242.00%	144.44%
CRIT	Average	22.54%	24.99%	108.97%	69.62%
	Max	93.66%	57.92%	186.62%	120.29%
ATC	Average	11.28%	14.69%	127.88%	65.32%
	Max	26.06%	54.32%	242.00%	144.44%
ATEC	Average	3.02%	4.29%	35.17%	14.11%
	Max	12.72%	11.63%	90.00%	38.58%
BIFET	Average	3.02%	4.29%	26.97%	13.67%
	Max	12.72%	11.63%	66.00%	38.58%

Table 5.13. Performance results for 15 jobs with high traffic intensity.

High Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	10.79%	6.66%	95.02%	73.91%
	Max	30.24%	32.87%	194.04%	195.88%
CRIT	Average	16.07%	15.87%	86.31%	67.21%
	Max	61.95%	47.88%	194.04%	154.64%
ATC	Average	10.79%	6.66%	95.02%	73.91%
	Max	30.24%	32.87%	194.04%	195.88%
ATEC	Average	3.70%	1.80%	24.29%	14.19%
	Max	11.19%	5.84%	39.72%	32.99%
BIFET	Average	3.70%	1.80%	20.33%	12.30%
	Max	11.19%	5.84%	39.57%	32.99%

Table 5.14 through Table 5.16 show the performance for the 5-, 10-, and 15-job problem instances with medium job traffic intensity. BIFET and ATEC continue to outperform the other heuristics, but especially more so when the due dates are tight. This is due to the fact that BIFET and ATEC can reschedule jobs that will be too early and allow for jobs that tighter due dates to be scheduled prior to those that have more slack.

Table 5.14. Performance results for 5 jobs with medium traffic intensity.

Medium Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	31.03%	68.92%	187.28%	119.42%
	Max	115.79%	307.14%	605.41%	330.43%
CRIT	Average	31.03%	71.39%	187.28%	120.53%
	Max	115.79%	307.14%	605.41%	330.43%
ATC	Average	31.03%	68.92%	187.28%	119.42%
	Max	115.79%	307.14%	605.41%	330.43%
ATEC	Average	0.64%	5.86%	17.46%	10.68%
	Max	3.81%	39.28%	42.31%	19.86%
BIFET	Average	0.64%	5.86%	15.03%	8.25%
	Max	3.81%	39.29%	37.97%	19.86%

Table 5.15. Performance results for 10 jobs with medium traffic intensity.

Medium Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	40.49%	34.50%	160.80%	126.88%
	Max	108.93%	88.19%	347.17%	314.96%
CRIT	Average	40.49%	39.39%	160.80%	128.53%
	Max	108.93%	88.19%	347.17%	314.96%
ATC	Average	40.49%	34.50%	160.80%	126.88%
	Max	108.93%	88.19%	347.17%	314.96%
ATEC	Average	1.03%	5.63%	27.81%	17.70%
	Max	7.14%	22.92%	48.60%	43.56%
BIFET	Average	1.03%	5.63%	26.76%	17.50%
	Max	7.14%	22.92%	48.60%	43.56%

Table 5.16. Performance results for 15 jobs with medium traffic intensity.

Medium Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	43.69%	30.54%	189.75%	133.67%
	Max	86.63%	90.42%	397.60%	306.90%
CRIT	Average	43.69%	33.10%	189.75%	132.14%
	Max	86.63%	90.42%	397.60%	306.90%
ATC	Average	43.69%	30.54%	189.75%	133.67%
	Max	86.63%	90.42%	397.60%	306.90%
ATEC	Average	1.26%	2.65%	37.23%	21.86%
	Max	5.81%	6.59%	62.42%	58.62%
BIFET	Average	1.26%	2.65%	37.72%	19.79%
	Max	5.81%	6.59%	62.42%	53.88%

Table 5.17 through Table 5.19 show how the proposed heuristics perform under low traffic intensity conditions. Both BIFET and ATEC outperform the other benchmark heuristics when there is low traffic. They are especially effective when due dates are tight.

Table 5.17. Performance results for 5 jobs with low traffic intensity.

Low Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	51.69%	219.38%	218.62%	192.61%
	Max	155.56%	1250.00%	428.95%	480.77%
CRIT	Average	51.69%	219.38%	218.62%	193.99%
	Max	155.56%	1250.00%	428.95%	480.77%
ATC	Average	51.69%	219.38%	218.62%	169.51%
	Max	155.56%	1250.00%	428.95%	480.77%
ATEC	Average	0.00%	0.00%	0.00%	3.43%
	Max	0.00%	0.00%	0.00%	32.69%
BIFET	Average	0.00%	0.00%	0.00%	3.43%
	Max	0.00%	0.00%	0.00%	32.69%

Table 5.18. Performance results for 10 jobs with low traffic intensity.

Low Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	61.68%	37.73%	345.96%	328.75%
	Max	233.33%	91.19%	1295.00%	994.23%
CRIT	Average	61.68%	37.73%	345.96%	328.75%
	Max	233.33%	91.19%	1295.00%	994.23%
ATC	Average	61.68%	37.73%	345.96%	328.75%
	Max	233.33%	91.19%	1295.00%	994.23%
ATEC	Average	0.00%	0.00%	1.02%	9.23%
	Max	0.00%	0.00%	7.63%	44.23%
BIFET	Average	0.00%	0.00%	1.02%	9.23%
	Max	0.00%	0.00%	7.63%	44.23%

Table 5.19. Performance results for 15 jobs with low traffic intensity.

Low Traffic		Tight Due Dates		Loose Due dates	
		1 family	5 families	1 family	5 families
FEDD	Average	55.66%	73.00%	235.60%	352.57%
	Max	144.12%	195.88%	446.81%	1563.79%
CRIT	Average	55.66%	66.47%	235.60%	352.57%
	Max	144.12%	154.64%	446.81%	1563.79%
ATC	Average	55.66%	73.00%	235.60%	352.57%
	Max	144.12%	195.88%	446.81%	1563.79%
ATEC	Average	0.00%	13.60%	2.29%	11.74%
	Max	0.00%	32.99%	10.34%	46.55%
BIFET	Average	0.00%	11.79%	2.29%	11.74%
	Max	0.00%	32.99%	10.34%	46.55%

The performance of the heuristics can be further analyzed using a confidence interval estimation approach to assess if the performance deltas of the proposed heuristics are statistically-significant. The confidence intervals of each scenario can be seen in Figure 5.1 through Figure 5.10. The figures show that the ATEC and BIFET proposed heuristics perform consistently better statistically than the benchmark heuristics. It is also evident that BIFET performs better than ATEC in some conditions such as when all jobs arrive simultaneously, i.e., during the highest level of traffic intensity, or job competition.

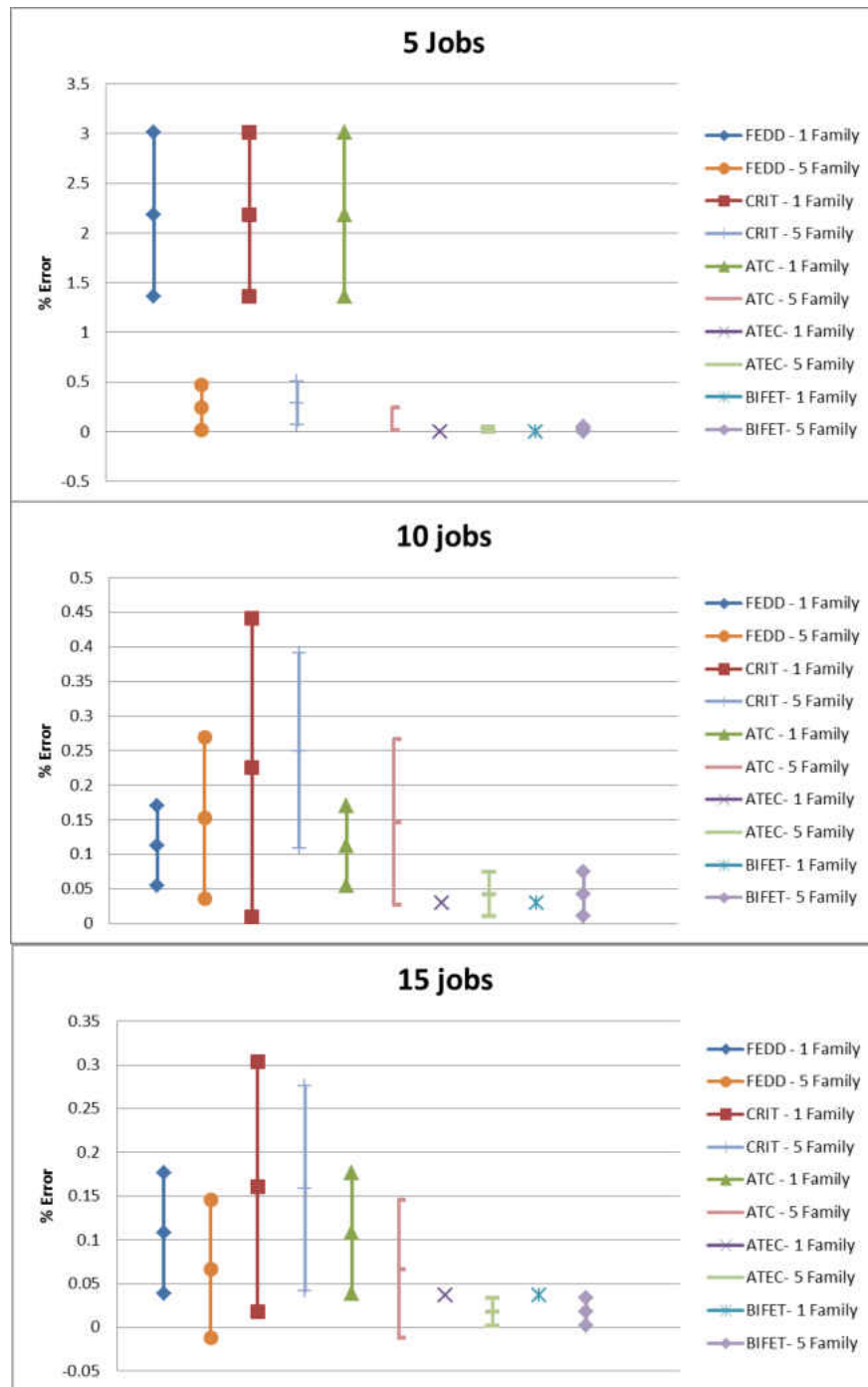


Figure 5.1. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and high traffic intensity.

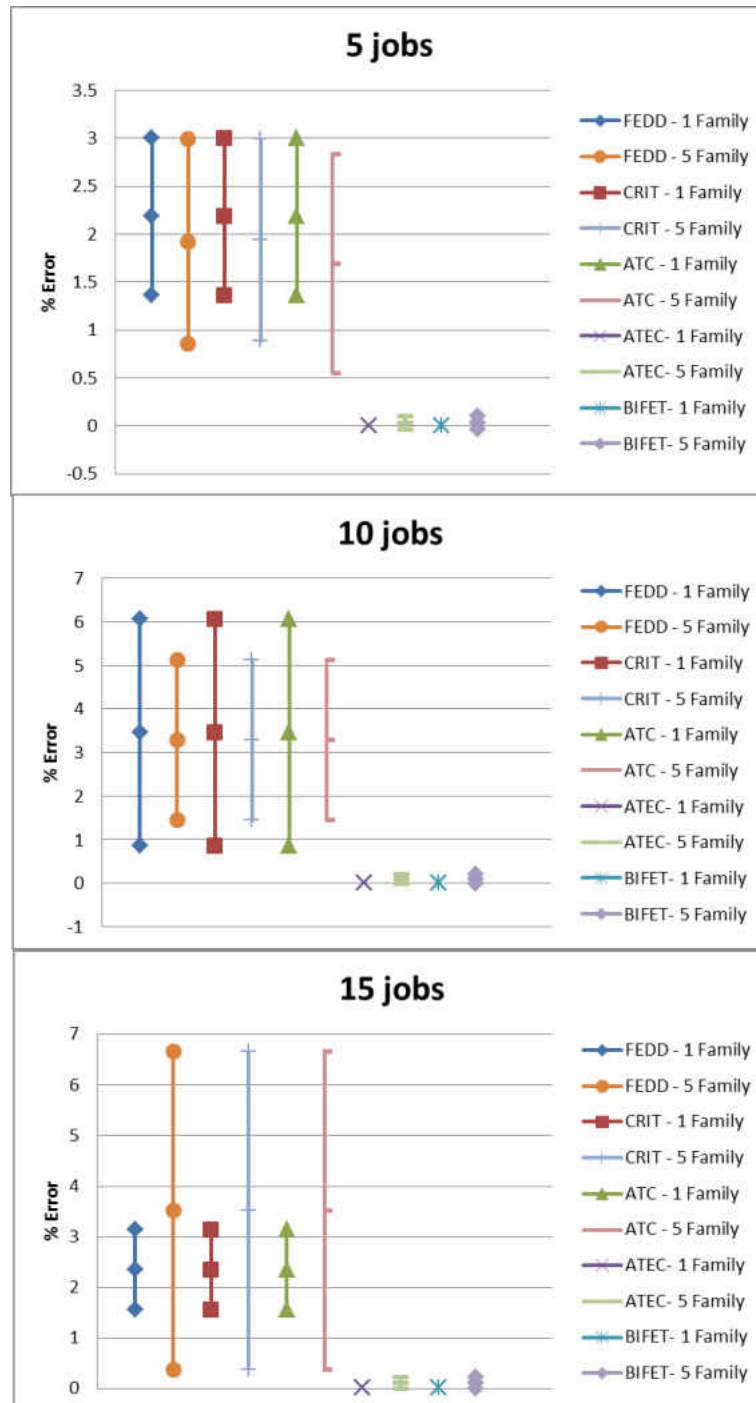


Figure 5.2. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and medium traffic intensity.

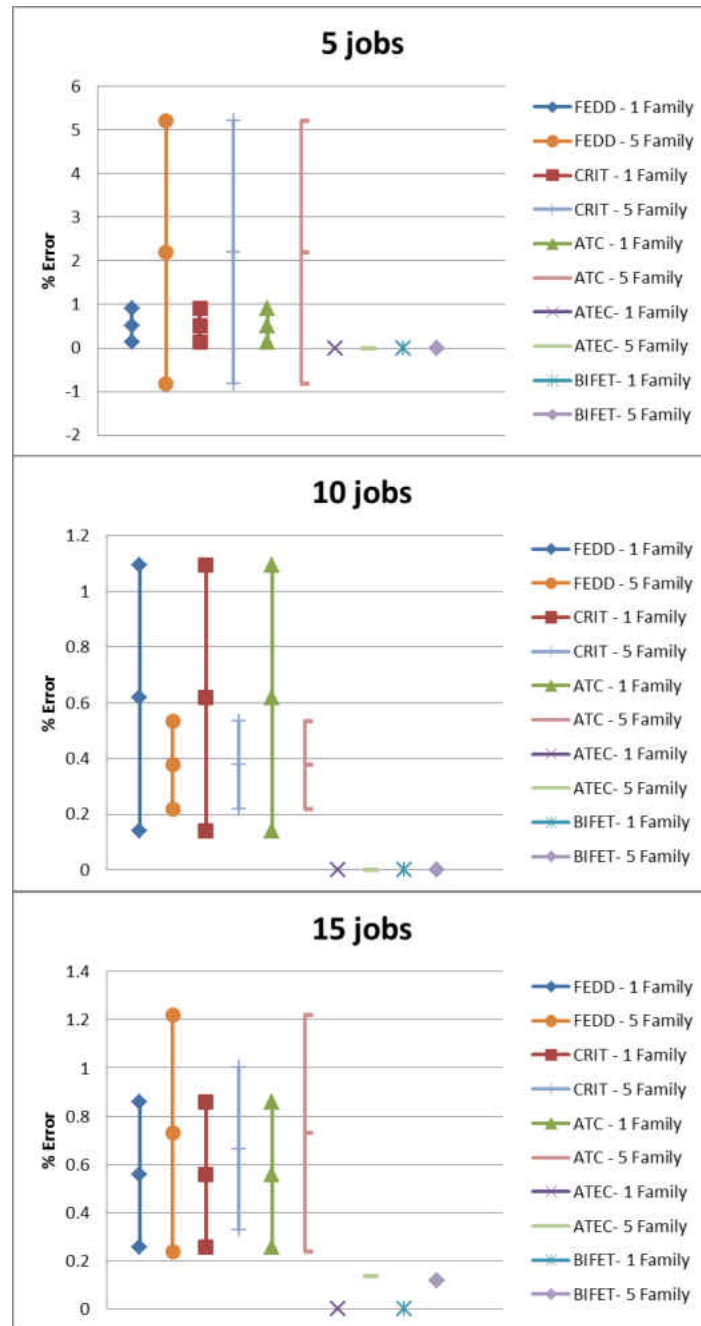


Figure 5.3. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, low traffic intensity.

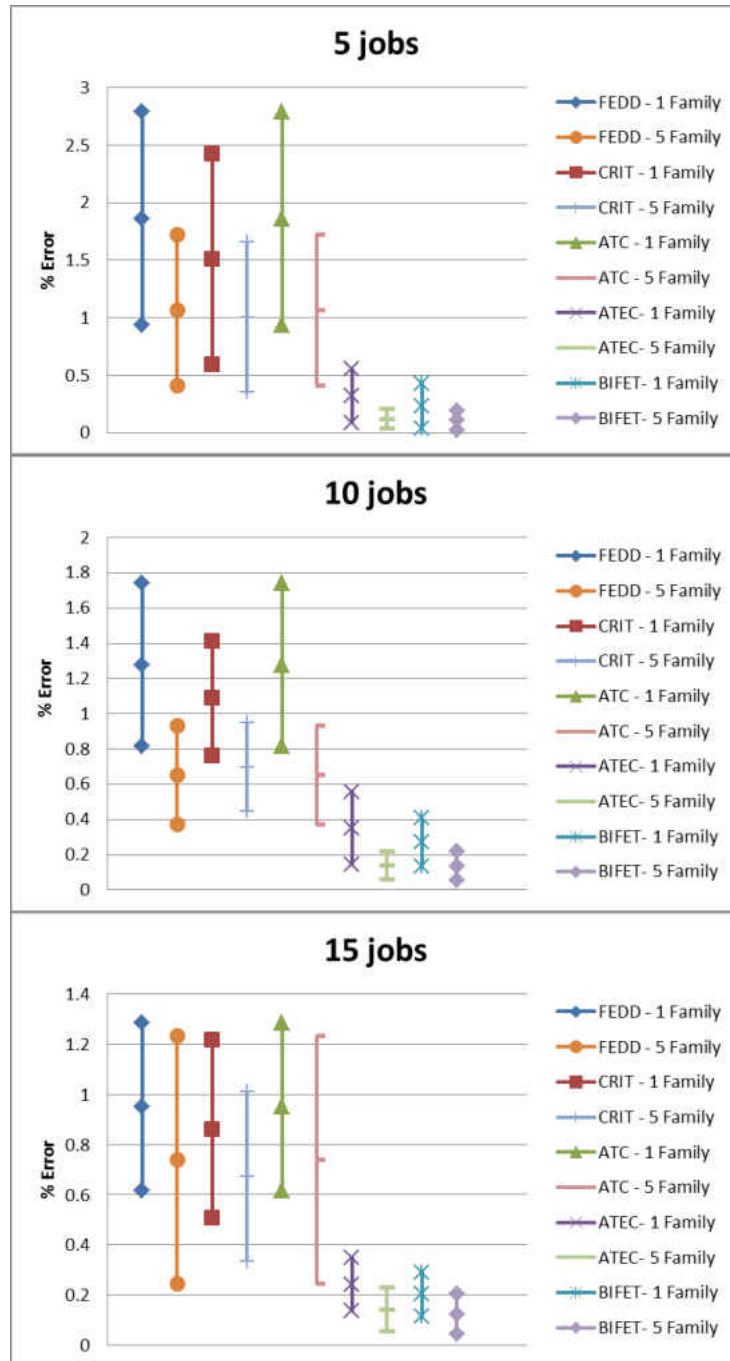


Figure 5.4. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and low traffic intensity.

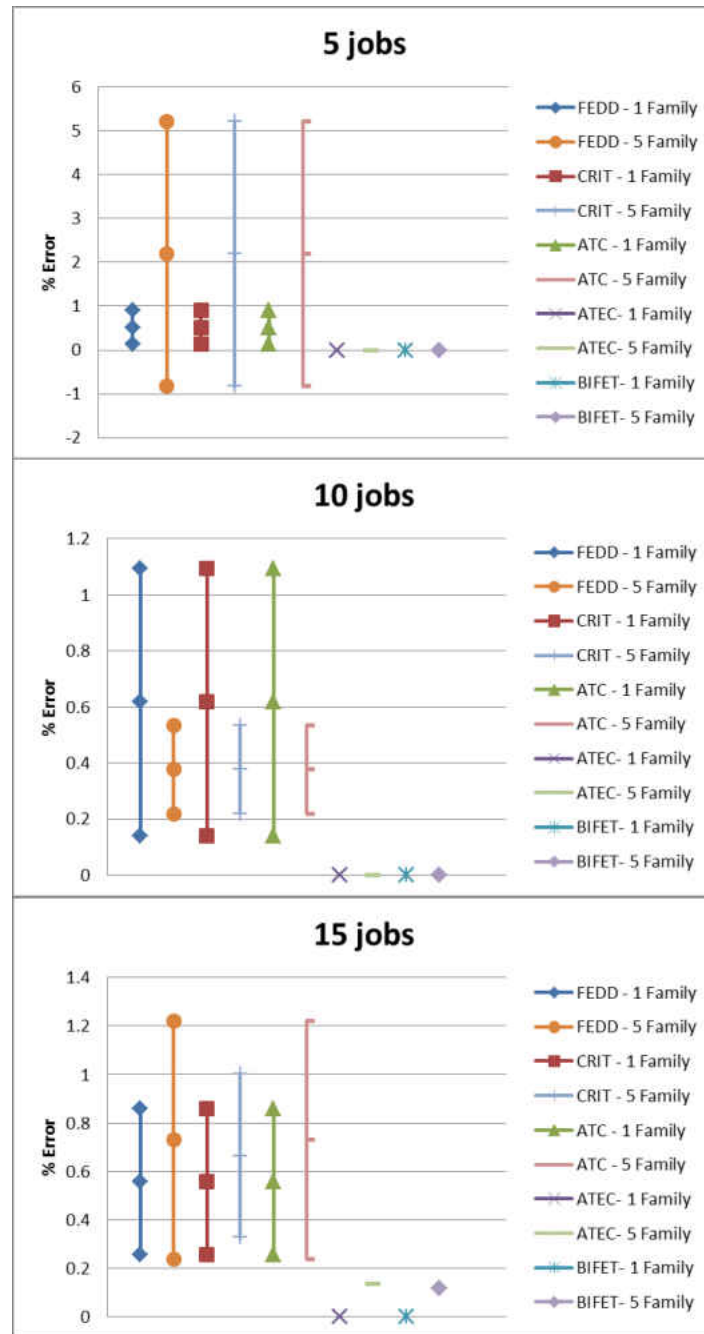


Figure 5.5. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and low traffic intensity.

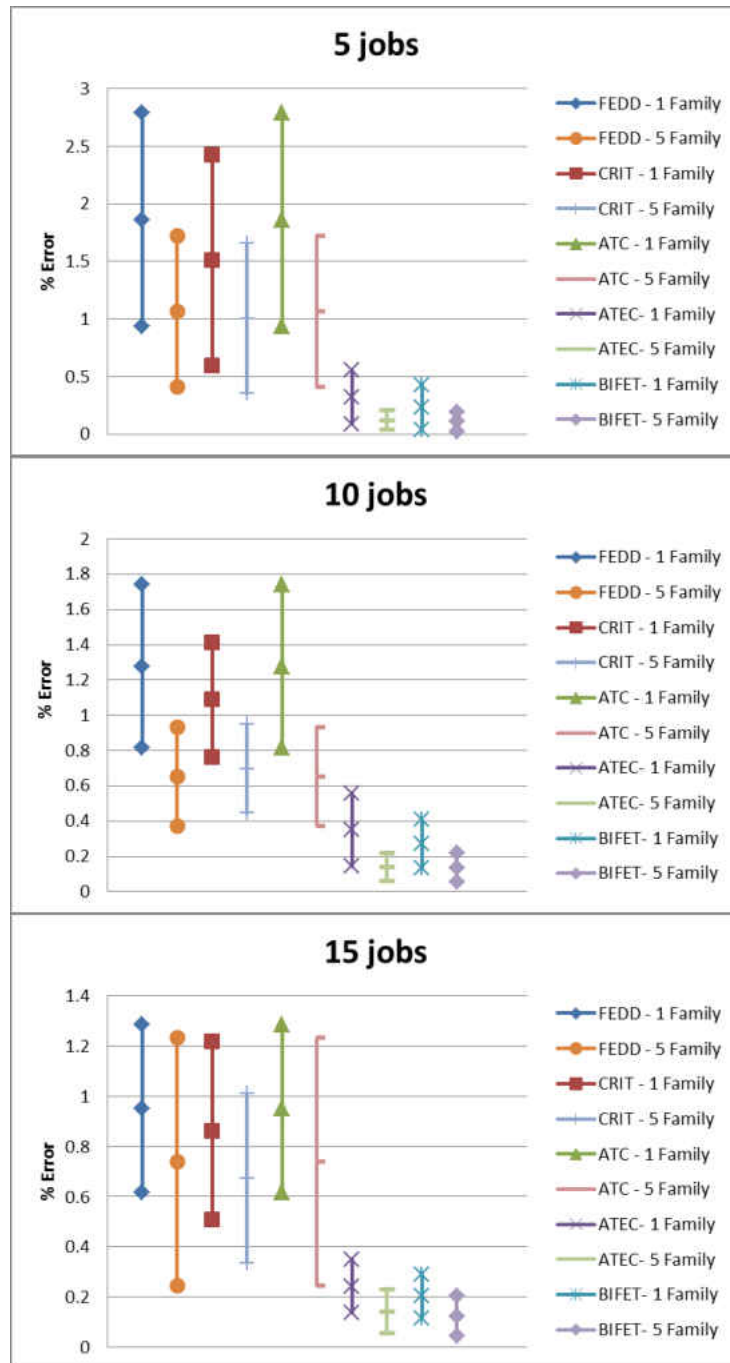


Figure 5.6. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and high traffic intensity.

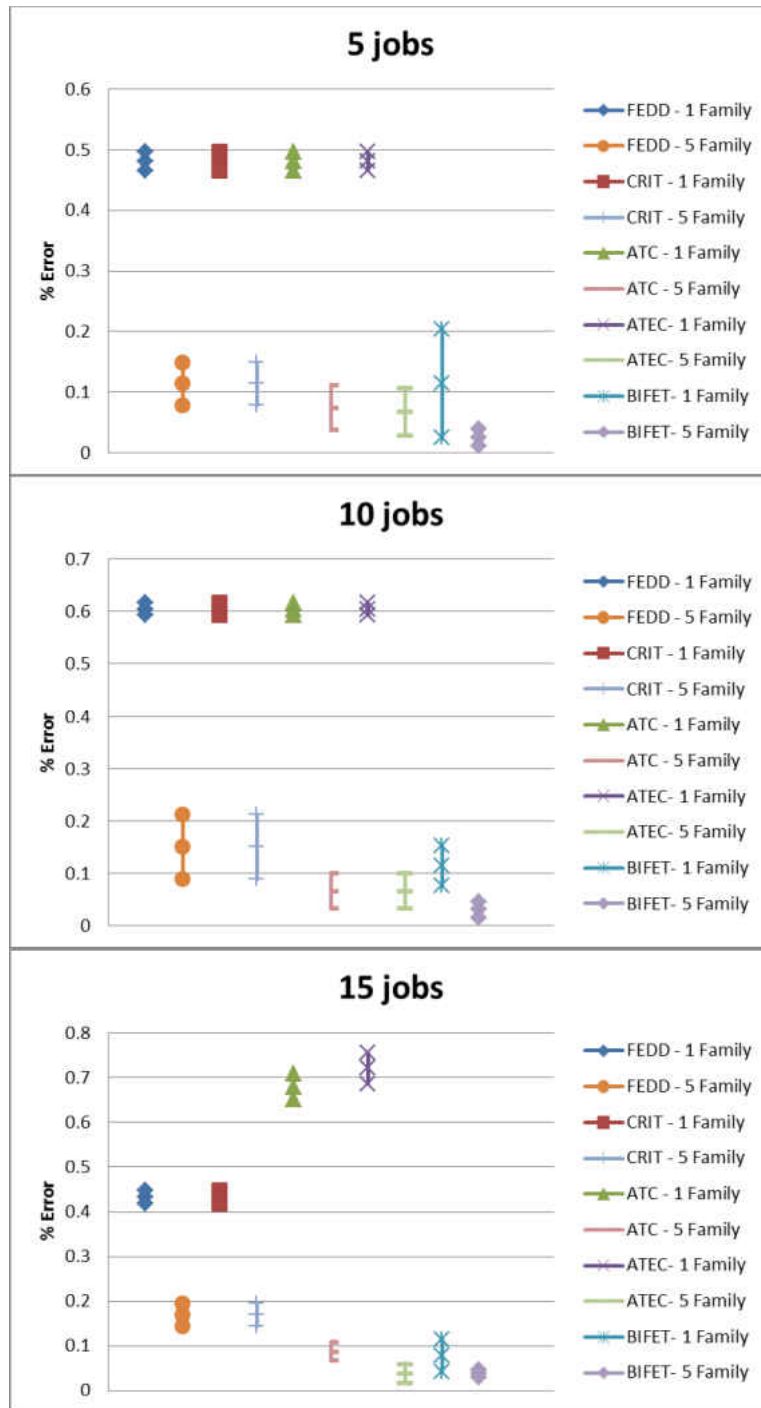


Figure 5.7. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and all jobs arriving simultaneously.

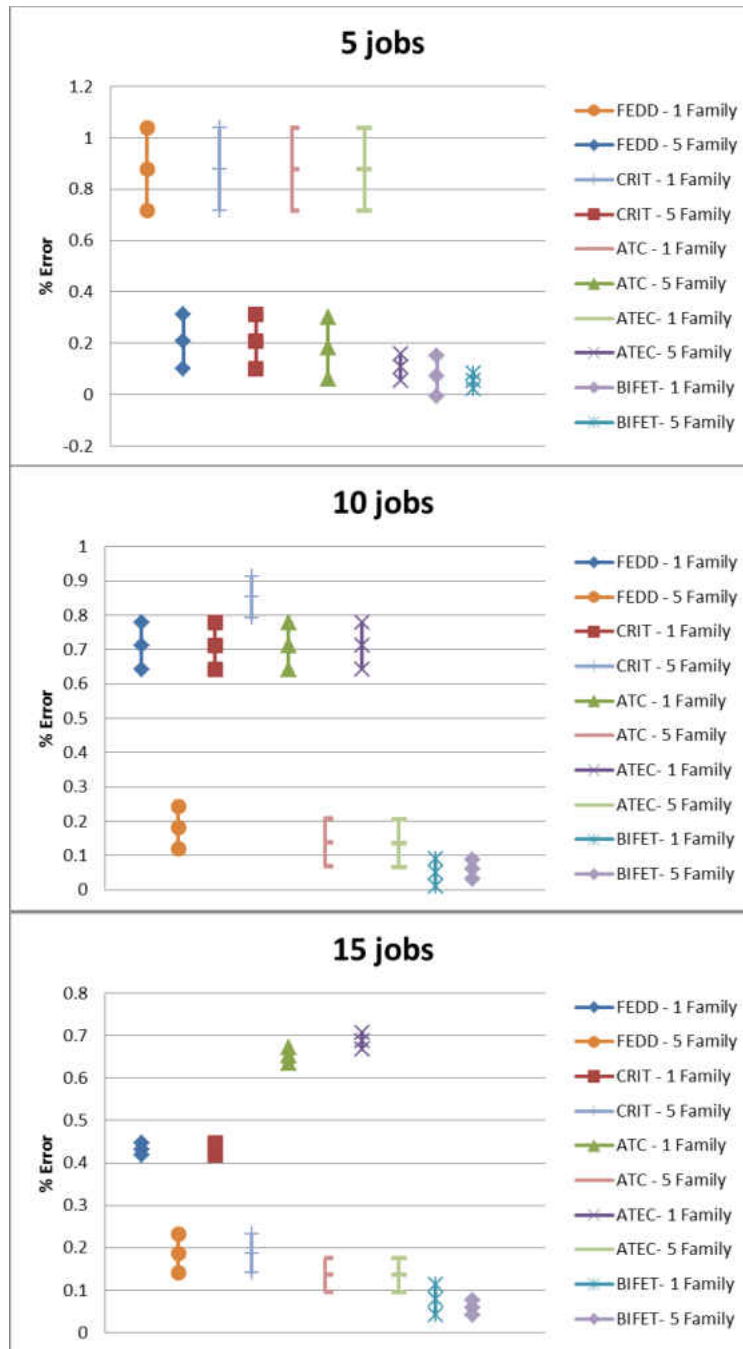


Figure 5.8. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and all jobs arriving simultaneously.

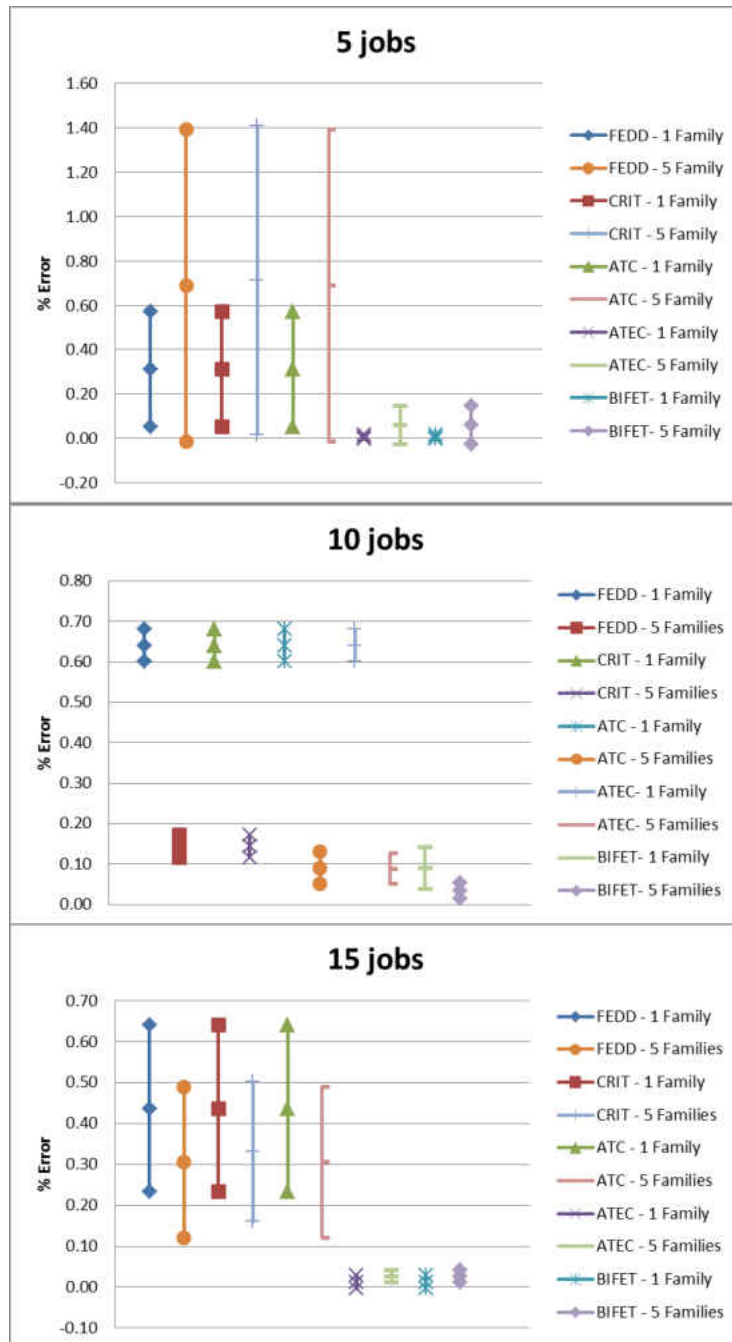


Figure 5.9. 95% confidence intervals on the performance results for 1 family and 5 families, tight due dates, and medium traffic intensity.

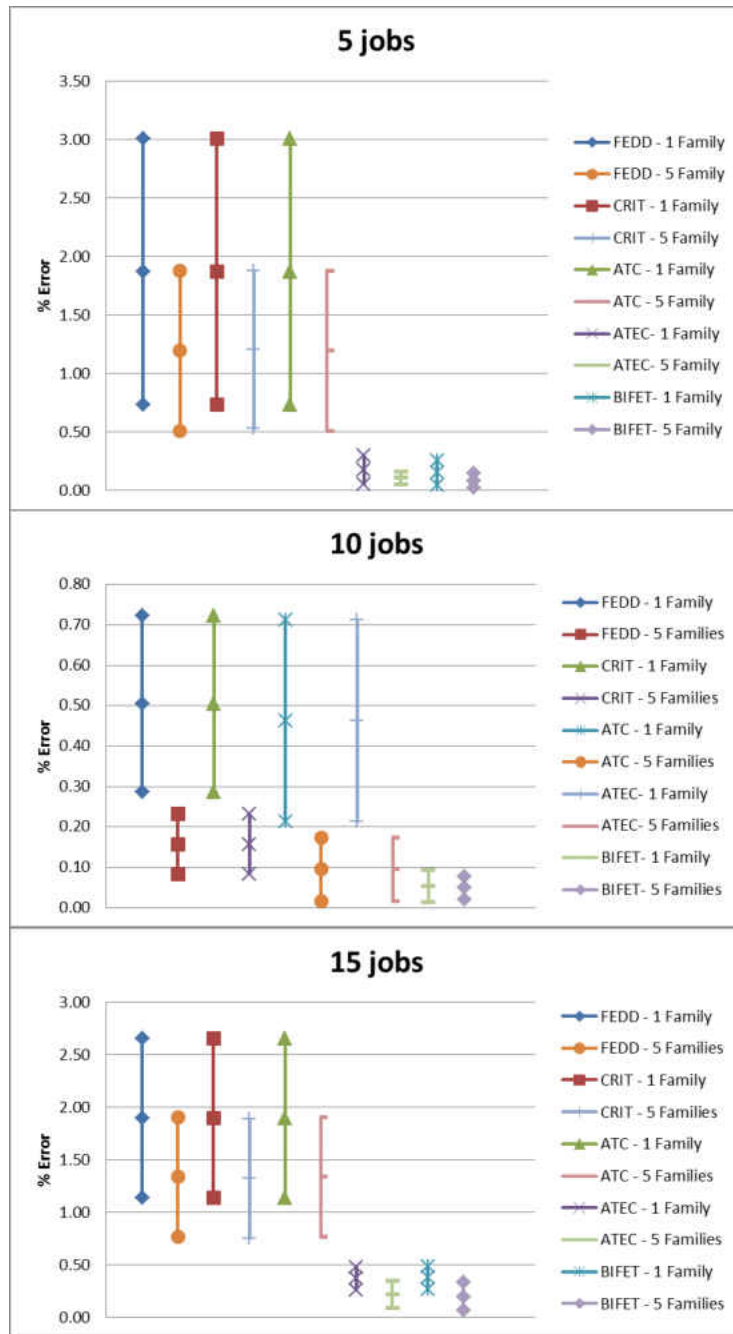


Figure 5.10. 95% confidence intervals on the performance results for 1 family and 5 families, loose due dates, and medium traffic intensity.

The ATEC and BIFET heuristics both delay the start time of jobs in order to prevent jobs from finishing unnecessarily early. This strategy has been very effective in minimizing overall earliness and tardiness. It also acts as an order review and release (ORR) strategy, in that, it delays the release time to the workstation. However, the general heuristics have several cases where the objective function can benefit from further improvement. ORR strategies have been found to be effective when integrated with priority dispatching rules. Therefore, blending these strategies with the benchmark heuristics may further improve the overall performance of the heuristics and therefore provide more competitive strategies against the proposed heuristics, ATEC and BIFET.

Another area that is explored is relaxing the constraint of equally-weighted tardiness and earliness penalties. Relaxing the constraint of equally-weighted tardiness and earliness penalties closer aligns the general problem described in Section 1.2. In addition, relaxation of the maximum batch size of ten is an interesting problem characteristic to explore. Removing the maximum batch size of 10 may reveal limitations in the performance of the heuristics at different batch sizes.

CHAPTER 6 : THE SINGLE BATCH WORKSTATION WITH UNEQUALLY-WEIGHTED PENALTIES AND VARYING BATCH SIZES AND INCOMPATIBLE FAMILIES AND SEQUENCE-INDEPENDENT SETUPS

6.1 Introduction

An insight gained from Phase 1 is that the size of the batch impacts the algorithm's performance. If several jobs are batched together, it causes jobs to be either very tardy or very early. Therefore, it is important to understand how the heuristics perform if the maximum batch size is varied. Another important aspect of the general problem is the weights placed on the earliness and tardiness measures. It is assumed in Phase 1 that the job are equally-weighted. In summary, process batch sizes, unequal tardiness and earliness penalties, and ORR strategies are investigated in this chapter.

6.2 Phase 2: Single Batch Workstation with Sequence-Independent Setups and Unequal Job Weights

In order to better assess the robustness of the proposed heuristics as well as the benchmark heuristics, they must be evaluated in varying production scheduling conditions including varying batch sizes and unequal earliness and tardiness penalties.

6.2.1 Experimental Design

Table 6.1 shows the experimental design which now includes unequally-weighted earliness and tardiness penalties as well as additional levels for the maximum batch size.

Table 6.1. Experimental design with emphasis on unequal job weights and varying batch sizes.

Factor	Values	Number of Values
Job Weight, θ_1, θ_2	$\theta_1 = \text{Earliness} = 1, \theta_2 = \text{Tardiness} = 1$ $\theta_1 = \text{Earliness} = 10, \theta_2 = \text{Tardiness} = 1$ $\theta_1 = \text{Earliness} = 1, \theta_2 = \text{Tardiness} = 10$	3
Number of Jobs, n	5, 10, 15	3
Number of Families, F	1, $U(1, 5)$	2
Traffic Intensity	<ul style="list-style-type: none"> • Low: 50% Machine Utilization • Medium: 75% Machine Utilization • High: 95% Machine Utilization • Highest: All Jobs Arrive Simultaneously 	4
Due Date Tightness	Tight: $U(0, 2)$, Loose: $U(0, 4)$	2
Max Batch Size	1, 3, 10	3
Replications per Problem		10
Total Number of Problem Instances		4,320

6.2.2 Performance Results of Varying Maximum Batch Sizes

In Phase 1, the maximum batch size is equal to 10. A feature of the BIFET heuristic which helps improve the overall early tardiness objective is that it constrains the maximum number of jobs in a batch. When all the jobs are batched according to maximum batch size, such as in FEDD and in CRIT, the heuristics do not consider and schedule the jobs in order to minimize both their earliness and tardiness measures. Instead all available jobs with the same family type are batched together, which, in turn, causes all those jobs in the batch either to be early or late. As shown in Table 6.1, there are two new levels added to the experimental design for maximum batch sizes. The scheduling scenarios are evaluated with a maximum batch size of

one and of three and then are compared to those same scenarios with a maximum batch size of 10.

Table 6.2 through Table 6.4 summarize the percent error between the different maximum batch sizes. For example, the percent error of pure ATC is shown with maximum batch size of 10 (MB10) minus the maximum batch size of 3 (MB3). If ATC has a higher percent error from the CPLEX solution, then the difference between MB10 and MB3 is a positive percent difference. If it is negative, then MB3 has a higher percent error, which means that the heuristic performs worse at a maximum batch size of 3.

As shown in Table 6.2, when batch sizes of ATC or ATEC vary, there is little impact on the overall performance of the heuristic in terms of percent error from the CPLEX optimal solutions, except in the cases when all jobs arrive simultaneously. There are also some differences when the traffic is high or medium and the due dates are loose. This difference can be seen when going from a maximum batch size of 10 to a maximum batch size of 1 or from a maximum batch size of 3 to a maximum batch size of 1. In these cases, the maximum batch size of 1 seems to perform better. This can be attributed to the fact that with a larger maximum batch size and loose due dates, the jobs available are processed together and cause the jobs to be early. This is shown in Table 6.4, where earliness is more important than tardiness. The percent changes are much higher than when the penalties are equally-weighted, or if tardiness is weighted higher than earliness.

Table 6.2. Comparison of ATC and ATEC performance of percent error with varying batch sizes and equal earliness and tardiness weights.

10 Jobs						
Equal Penalties	ATC					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%
<S, T, 5>	1.96%	3.73%	3.97%	4.57%	2.01%	1.85%
<H, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 1>	65.38%	5.97%	71.01%	9.47%	5.63%	3.95%
<S, L, 5>	2.91%	4.65%	8.79%	9.00%	5.88%	6.87%
<H, L, 1>	0.00%	0.00%	20.41%	16.73%	20.41%	16.73%
<H, L, 5>	0.00%	0.00%	4.66%	9.51%	4.66%	9.51%
<M, L, 1>	0.00%	0.00%	28.36%	28.24%	28.36%	28.24%
<M, L, 5>	0.00%	0.00%	2.26%	5.39%	2.26%	5.39%
<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

10 Jobs						
Equal Penalties	ATEC					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%
<S, T, 5>	1.96%	3.73%	3.97%	4.57%	2.01%	1.85%
<H, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 1>	66.26%	6.07%	71.01%	9.47%	4.75%	4.28%
<S, L, 5>	3.08%	4.49%	8.49%	8.94%	5.41%	6.44%
<H, L, 1>	0.00%	0.00%	12.40%	25.08%	12.40%	25.08%
<H, L, 5>	0.00%	0.00%	3.46%	7.94%	3.46%	7.94%
<M, L, 1>	0.00%	0.00%	16.15%	12.40%	16.15%	12.40%
<M, L, 5>	0.00%	0.00%	1.52%	3.45%	1.52%	3.45%
<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Table 6.3. Comparison of ATC and BIFET performance of percent error with varying batch sizes and unequal earliness and tardiness weights where earliness is more important than tardiness.

Early=10/ Tardy=1	ATC						Early=10/ Tardy=1	ATEC					
	MB10-MB3		MB10-MB1		MB3-MB1			MB10-MB3		MB10-MB1		MB3-MB1	
	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV		AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<Ti,DT,F>							<Ti,DT,F>						
<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%	<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%
<S, T, 5>	1.39%	3.19%	3.47%	4.05%	2.08%	1.82%	<S, T, 5>	1.39%	3.19%	3.47%	4.05%	2.08%	1.82%
<H, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<H, T, 1>	0.00%	0.00%	0.39%	1.24%	0.39%	1.24%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 1>	57.96%	7.95%	71.72%	8.16%	13.76%	13.75%	<S, L, 1>	62.91%	8.54%	71.72%	8.16%	8.82%	12.87%
<S, L, 5>	3.11%	4.72%	12.00%	13.85%	8.90%	13.50%	<S, L, 5>	3.27%	4.56%	8.62%	9.61%	5.35%	6.82%
<H, L, 1>	0.00%	0.00%	351.77%	328.21%	351.77%	328.21%	<H, L, 1>	0.00%	0.00%	74.62%	170.21%	74.62%	170.21%
<H, L, 5>	0.00%	0.00%	29.46%	72.22%	29.46%	72.22%	<H, L, 5>	-11.52%	39.18%	-7.10%	22.73%	4.42%	21.97%
<M, L, 1>	0.00%	0.00%	299.86%	400.33%	299.86%	400.33%	<M, L, 1>	0.00%	0.00%	35.91%	34.60%	35.91%	34.60%
<M, L, 5>	0.00%	0.00%	71.99%	165.99%	71.99%	165.99%	<M, L, 5>	0.00%	0.00%	5.18%	19.44%	5.18%	19.44%
<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	1.96%	6.20%	1.96%	6.20%	<LW, L, 5>	0.00%	0.00%	0.06%	0.20%	0.06%	0.20%

Table 6.4. Comparison of ATC and ATEC performance of percent error with varying batch sizes and unequal earliness and tardiness weights where tardiness is more important than earliness.

Early=1 /Tardy=10	ATC						Early=1 /Tardy=10	ATEC					
	MB10-MB3		MB10-MB1		MB3-MB1			MB10-MB3		MB10-MB1		MB3-MB1	
<TI,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV	<TI,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%	<S, T, 1>	56.29%	1.49%	60.45%	1.60%	4.17%	0.11%
<S, T, 5>	1.39%	3.19%	3.47%	4.05%	2.08%	1.82%	<S, T, 5>	1.39%	3.19%	3.47%	4.05%	2.08%	1.82%
<H, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<H, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<M, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 1>	65.95%	6.45%	71.72%	8.16%	5.77%	1.88%	<S, L, 1>	66.44%	6.67%	71.72%	8.16%	5.28%	1.75%
<S, L, 5>	2.95%	4.53%	8.63%	8.73%	5.68%	6.60%	<S, L, 5>	3.11%	4.38%	8.63%	8.73%	5.53%	6.48%
<H, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<H, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<H, L, 5>	-0.78%	2.46%	0.69%	4.13%	1.47%	2.91%	<H, L, 5>	-0.75%	2.37%	0.70%	4.04%	1.45%	2.88%
<M, L, 1>	0.00%	0.00%	0.32%	1.00%	0.32%	1.00%	<M, L, 1>	0.00%	0.00%	0.28%	0.89%	0.28%	0.89%
<M, L, 5>	0.00%	0.00%	0.04%	0.11%	0.04%	0.11%	<M, L, 5>	0.00%	0.00%	0.04%	0.11%	0.04%	0.11%
<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, L, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Figure 6.1 to Figure 6.3 show a closer look at the effects of batch sizes on the performance of the heuristics in percent error deviation from the optimal solution. As in the case when the jobs have tight due dates and arrive simultaneously, it is apparent from Figure 6.1 that the maximum batch size affects the overall performance of the heuristic. As the batch size decreases, the percent error as well as the variation of the percent error decreases. BIFET continuously performs better regardless of batch size in this scenario as well. Figure 6.1 shows the percent error steadily decreasing as the maximum batch size is reduced from 10 to 3 to 1. This is because all jobs are available at once and, therefore, FEDD, CRIT, and ATC batch all available jobs up to the maximum batch size. This causes many of the jobs to be either early or late. MB10 denotes a maximum batch size of 10, MB3 represents a maximum batch size of 3,

and MB1 represents a maximum batch size of 1. The graph depicting these results for one family can be found in Appendix A.

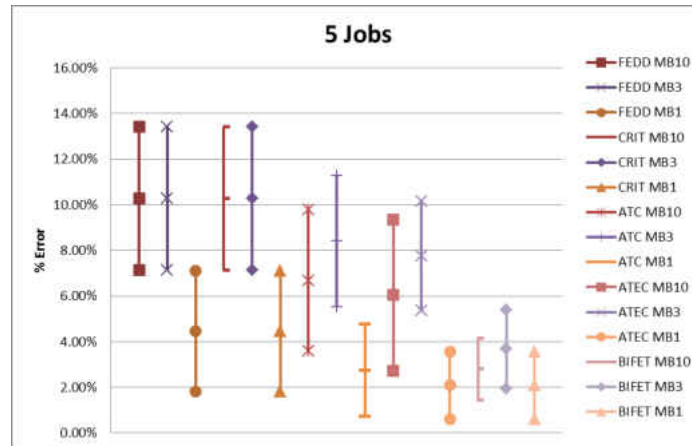


Figure 6.1. 95% confidence interval on the performance results with varying batch sizes for tight due dates, all jobs arrive simultaneously, 5 families, and earliness penalty = 10 and tardiness penalty = 1.

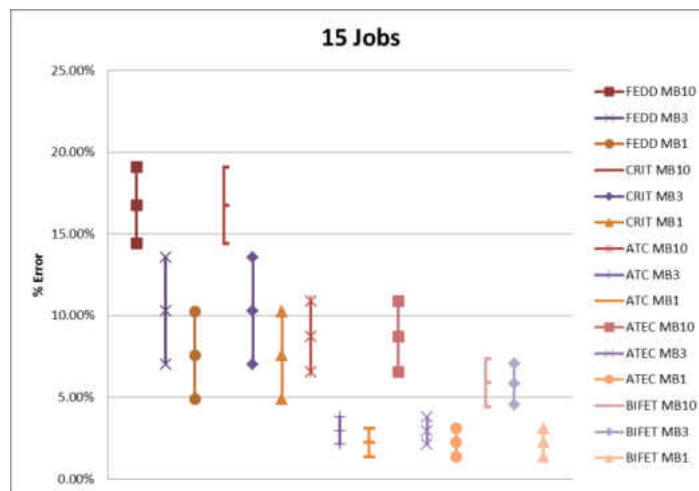


Figure 6.2. 95% confidence interval on the performance results with varying batch sizes for tight due dates, all jobs arrive simultaneously, 5 families, and earliness penalty = 1 and tardiness penalty = 10.

Problem instances where jobs arrive at high, medium and low traffic intensity with tight due dates show that maximum batch size does not impact the performance of the heuristic. This is because, if jobs arrive at varying times, the batch might not wait for a maximum batch size to be reached in order to be processed and therefore having a maximum batch size of 10 versus a maximum batch size of 3 does not impact the results since there may not be enough jobs waiting in the queue to meet the maximum batch size.

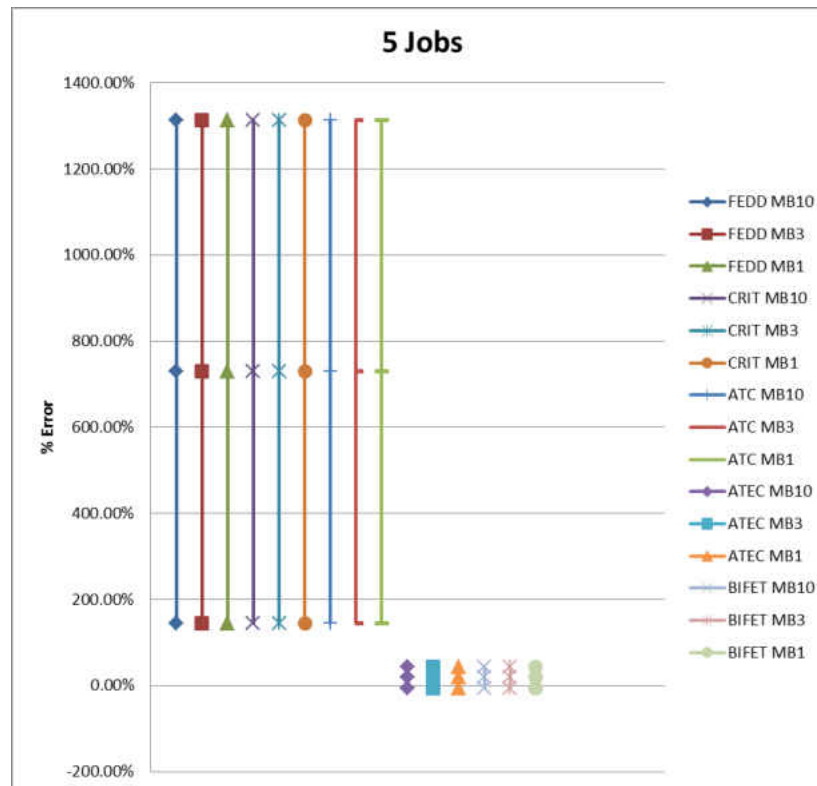


Figure 6.3. 95% confidence interval on the performance results with varying batch sizes for tight due dates, medium traffic intensity, 5 families, earliness penalty = 10 and tardiness penalty = 1.

ATEC and BIFET outperform all the other common heuristics in terms of overall percent error and variation. The graphs showing the results for high, and low traffic, as well as family size of one can all be found in appendix A. These graphs are similar to that of Figure 6.3.

6.2.3 Unequal Weights

In order to fully understand the robustness of the proposed algorithms, it is important to see their performance when one of the weights of earliness and tardiness is heavier than the other.

Table 6.2 shows the comparison of percent error between when job penalties are weighted differently. The letter ‘A’ is for both penalties being equally weighted. ‘E’ is when earliness is more heavily weighted and ‘T’ is when tardiness is more heavily weighted (also see Table 6.1 for the meaning of the acronyms). For example, one column shows the percent error difference of FEDD with equal weights (A) minus tardiness being heavily weighted (T). If the heuristic FEDD has a higher percent error off from the optimal solution in A, then the difference between A and T is a positive % difference and tardiness performs better, meaning there is less percent error in tardiness from the optimal solution. If it is negative, then T has a higher percent error, which means that the heuristic performs worse when tardiness is more heavily weighted.

Table 6.5, shows the impact of varying weights, when a batch size of 10 is held constant. It can be seen from this table that the performance of the heuristics, for the most part, relies heavily upon how the jobs are weighted, especially in the cases when the due dates are loose or when traffic intensity is low. FEDD’s percent error when earliness is more heavily weighted is

very high and this is due to the fact that FEDD does not consider earliness when scheduling jobs.

BIFET seems to perform better when the penalties are equally weighted as can be seen in Figure 6.5 for 5 jobs.

Table 6.5. Comparison of FEDD and BIFET performance of percent error when varying penalties at a maximum batch size of 10.

10 Jobs						
Max Batch 10	FEDD					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, T, 5>	0.55%	1.22%	0.55%	1.22%	0.00%	0.00%
<H, T, 1>	10.30%	7.28%	-244.50%	125.54%	-254.80%	131.77%
<H, T, 5>	11.95%	12.05%	-211.44%	187.20%	-223.39%	198.70%
<M, T, 1>	36.05%	30.98%	-444.09%	335.91%	-480.14%	366.38%
<M, T, 5>	30.43%	28.40%	-440.78%	332.06%	-471.21%	359.52%
<LW, T, 1>	55.51%	59.94%	-555.08%	599.43%	-610.59%	659.37%
<LW, T, 5>	33.90%	19.80%	-351.63%	201.58%	-385.53%	221.32%
<S, L, 1>	0.33%	8.24%	0.33%	8.24%	0.00%	0.00%
<S, L, 5>	-0.11%	0.70%	-1.26%	4.55%	-1.16%	5.12%
<H, L, 1>	117.77%	60.20%	-1378.31%	527.00%	-1496.08%	578.79%
<H, L, 5>	52.23%	36.79%	-781.48%	319.18%	-833.71%	350.18%
<M, L, 1>	144.05%	73.22%	-1639.71%	737.09%	-1783.76%	808.57%
<M, L, 5>	107.03%	67.78%	-1406.22%	703.75%	-1513.25%	769.99%
<LW, L, 1>	302.41%	326.91%	-3356.44%	3243.97%	-3658.85%	3570.27%
<LW, L, 5>	280.01%	224.08%	-3155.28%	2320.01%	-3435.30%	2542.49%

10 Jobs						
Max Batch 10	BIFET					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, T, 5>	-2.41%	2.82%	-2.41%	2.82%	0.00%	0.00%
<H, T, 1>	2.15%	3.91%	-13.39%	19.91%	-15.53%	21.86%
<H, T, 5>	2.12%	4.59%	-15.58%	21.55%	-17.70%	22.17%
<M, T, 1>	0.08%	1.60%	-6.11%	20.45%	-6.19%	21.99%
<M, T, 5>	2.71%	7.36%	-22.51%	27.35%	-25.22%	29.11%
<LW, T, 1>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	-0.51%	0.71%	0.00%	0.00%	0.51%	0.71%
<S, L, 1>	-0.09%	5.13%	-0.09%	5.13%	0.00%	0.00%
<S, L, 5>	-3.43%	2.51%	-3.07%	2.76%	0.36%	0.99%
<H, L, 1>	19.79%	13.23%	-58.69%	93.19%	-78.48%	84.12%
<H, L, 5>	6.76%	13.03%	-10.96%	25.66%	-17.72%	29.40%
<M, L, 1>	11.30%	19.69%	-30.32%	51.07%	-41.62%	53.10%
<M, L, 5>	4.50%	14.31%	-59.09%	70.72%	-63.60%	71.06%
<LW, L, 1>	-7.42%	8.30%	-8.71%	21.89%	-1.30%	26.44%
<LW, L, 5>	-12.75%	19.32%	-57.20%	130.43%	-44.45%	144.65%

Figure 6.5 and Figure 6.4 show how varying the earliness-tardiness penalties affect the performance of the heuristics.

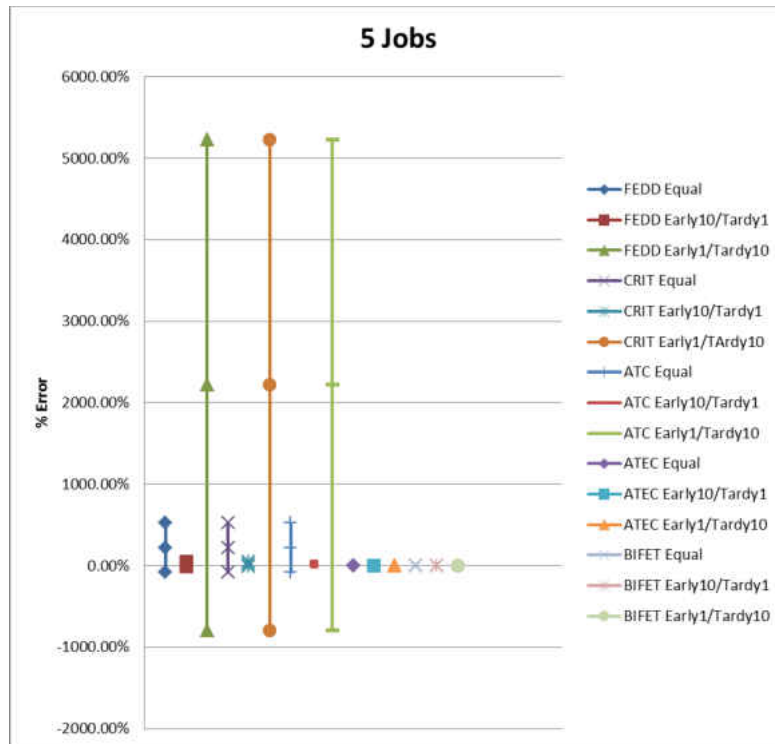


Figure 6.4. 95% confidence interval on the performance results for tight due dates, low traffic intensity, 5 families, and maximum batch size 10.

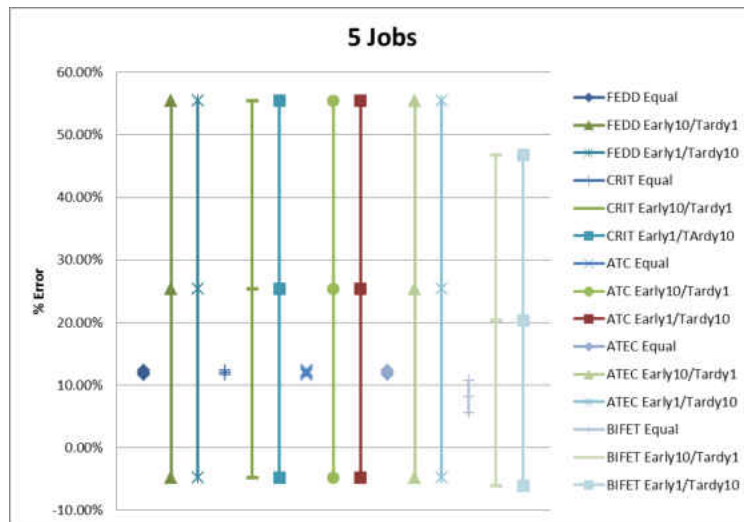


Figure 6.5. 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 1 family, and maximum batch size 3.

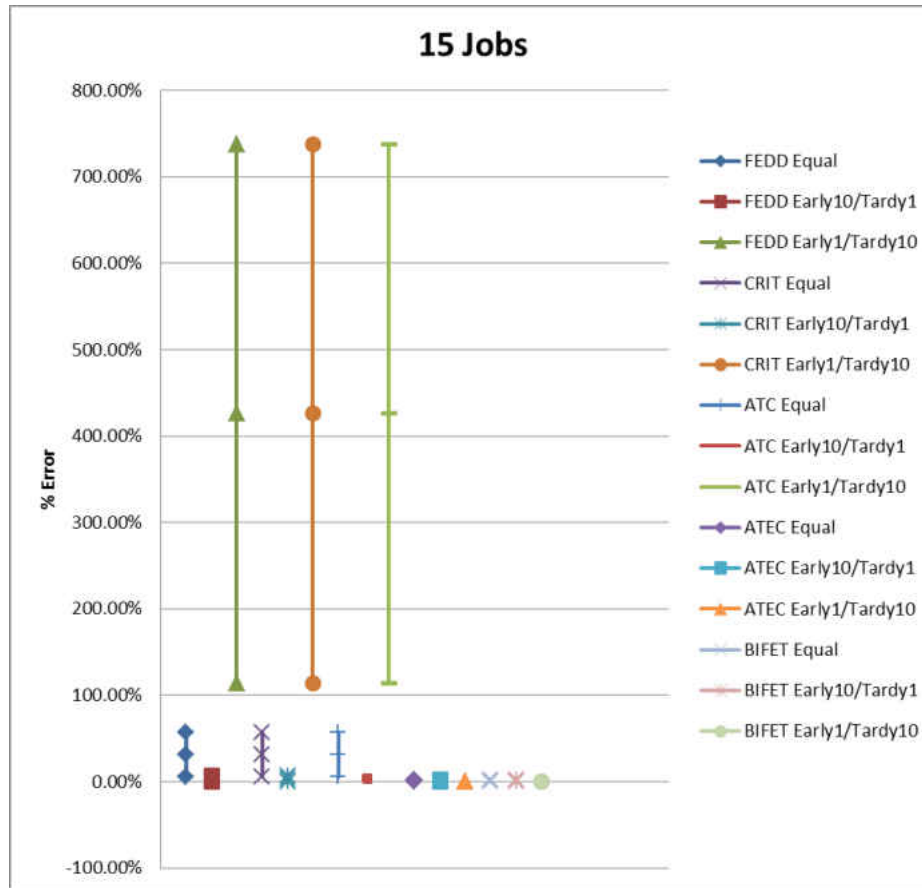


Figure 6.6. 95% confidence interval on the performance results for Max batch size 1, tight due dates, medium traffic intensity, and 1 family.

There is large variation in the performance of the benchmark heuristics and the proposed BIFET and ATEC outperform these heuristics in the majority of the scenarios. Again, this is due to the fact that ATEC and BIFET consider earliness and readjust the release time to the machine to minimize earliness and to ensure that the bottleneck is not being overrun. This, in turn, acts an ORR strategy in which it delays the release time in order to minimize work-in-process inventory, which can lead to machine congestion and tardiness as well as earliness, where jobs are being

sent to the machine prematurely. In an effort to reduce the percent error, effective ORR strategies are integrated with the general heuristics.

6.3 Integrating Order Review and Release Strategies

ATEC and BIFET heuristics have a clear advantage in their performance against the benchmark heuristics because, unlike FEDD and even ATC, they consider the earliness component of the composite objective function. Therefore, in order to compensate for this consideration, ORR strategies should be combined with these dispatching rules to determine if the ORR strategies will improve their performance especially under the conditions when the earliness penalty is weighted more heavily than tardiness.

ORR is the boundary between the manufacturing planning system and the production floor (Melnik et al, 1991) and controls the release of work to the floor. Melnik et al. (1991) use a combination of ORR and a planning system in conjunction with simulation and Design of Experiments to examine the following factors:

- Work load smoothing
- Order release mechanisms
- Dispatching rules used on the shop floor

The results from the experiment show that work load smoothing by the planning system improves system performance for tardiness related measures (mean tardiness and percent tardy) and flow time related measures. ORR strategies determine the timing in which orders are

released to the shop floor. Bertrand et al (2002) find that using ORR strategies while focusing on the due date performance of a shop can significantly decrease or eliminate assembly order lateness. Dispatching rules are often used for scheduling due to their ease of use, ability to react to disruptions without the need to reschedule, and ability to be used real time (Baykasoglu et al, 2009). Over the past several years, operations scheduling has been expanded to include various decision points (Lu et al., 2011). These include when to schedule product, dispatching rules, etc. It is evident, through works such as Min and Yih (2003) and Parthanadee et al (2010) to name a few, that analyzing the process flow and identifying the most appropriate dispatching rules can improve schedule performance. As a result, this research has been expanded to include decisions such as ORR strategies and dispatching rules to improve the earliness/tardiness objective function.

There are several ways in which ORR strategies can help reduce schedule deviation and improve on-time delivery performance. ORR strategies can be used to help control lead times by limiting the amount of WIP at each work center in order to prevent the manufacturing floor becoming too congested and exceeding capacity (Bergamaschi et al 1997). ORR strategies can also help moderate the utilization of machines by introducing product at balanced levels rather than having points of high volumes of WIP and then periods of starvation.

The interaction between ORR and dispatching rules has been shown to have a significant and positive impact on performance metrics such as MAD (Lu et al., 2011). MAD is a measurement that compares the actual delivery date with the original scheduled date (Lu et al., 2011). The mathematical model in this phase will reflect the model in Phase 1.

The following section will identify the ORR strategies selected in this research and identify the impact that these strategies have on the objective function of earliness and tardiness.

6.3.1 Experimental Design with ORR Strategies

Two ORR strategies have been selected to use in conjunction with the previous dispatching rules. These are IMM (Immediate Release) and Modified Infinite Loading strategies (MIL).

6.3.1.1 Immediate Release (IMM)

Immediate Release Strategy releases all jobs to the floor as they arrive, regardless of priority. Essentially, it is equal to not having an ORR strategy. Therefore, the dispatching rules remain the same and are compared with the dispatching rules combined with the MIL ORR strategy.

6.3.1.2 Modified Infinite Loading (MIL)

MIL is considered a “planning factor” method in which it considers information about the individual jobs (due date, number of ops) and about the current shop congestion.

MIL Modified Infinite Loading:

$$RD_i = d_i - k_1 n - k_2 Q_i \quad (6.1)$$

RD_i	New release time of job i
d_i	Due date of job i
k_1	Planning Factors
n	Number of operations for job i
k_2	Planning Factors
Q_i	Number of jobs in queue for same routing as job i 's

Planning Factors

The planning factor k_1 is multiplied by the number of operations in the job's routing. Since this research is examining only 1 machine, k_1 will be the average of the setup times for the job families. The planning factor k_2 is multiplied by the number of jobs waiting in the queue. This factor is derived from the average of the processing times for each of the families.

6.4 Results with ORR Strategies

The ORR strategy MIL is integrated with the benchmark dispatching rules (FEDD, CRIT, and ATC) and compared to the proposed heuristics ATEC and BIFET. Figure 6.7 shows all jobs arriving simultaneously. In this case, the ORR strategies do not appear to have a significant impact on the overall percent error between the heuristic and optimal solution, when compared against their respective batch sizes. This is due to all the jobs arriving simultaneously. The likelihood of the jobs being tardy is high and therefore a new release date is probably not calculated by the ORR strategy and therefore perform the same as when the dispatching rules are

used alone. BIFET and ATEC continue to perform better and all heuristic's performances improve as the batch sizes decrease, which was previously discussed in section 6.2.2.

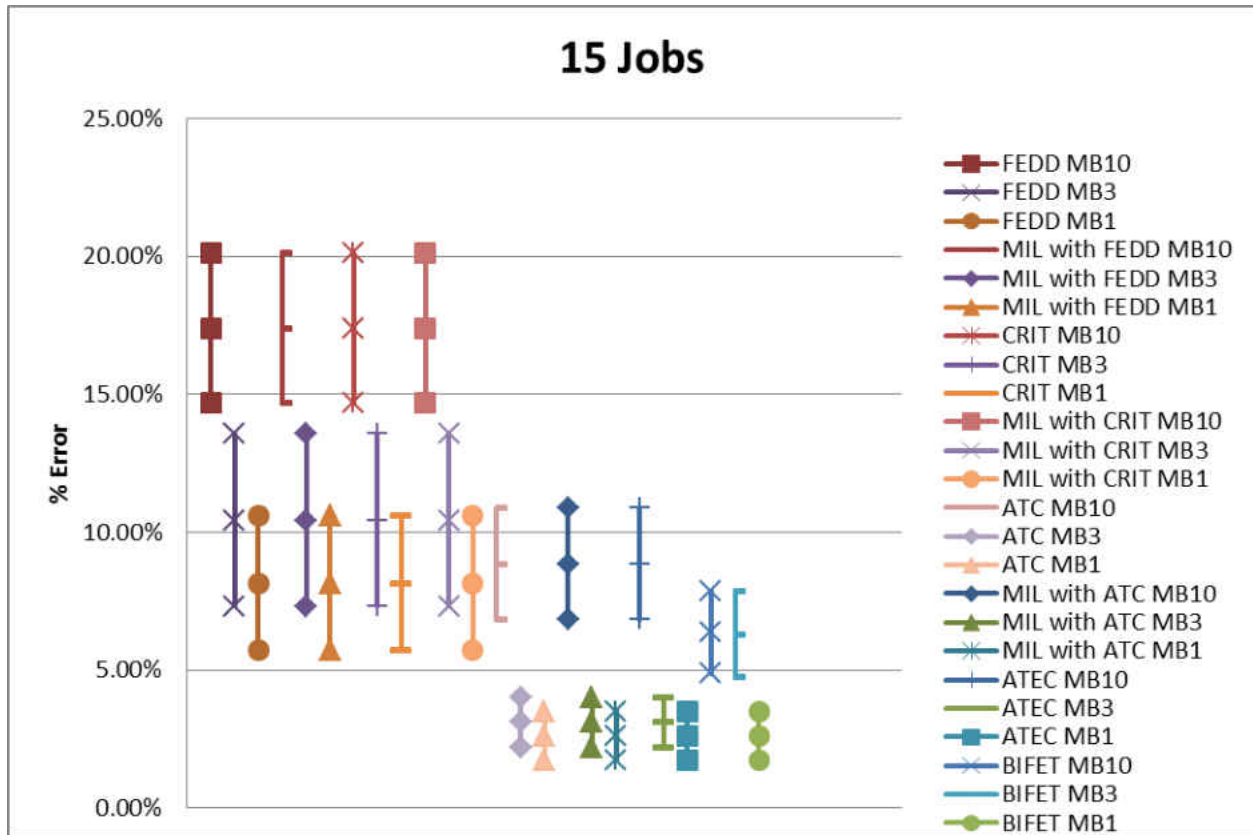


Figure 6.7. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, all jobs arriving simultaneously, and 5 families.

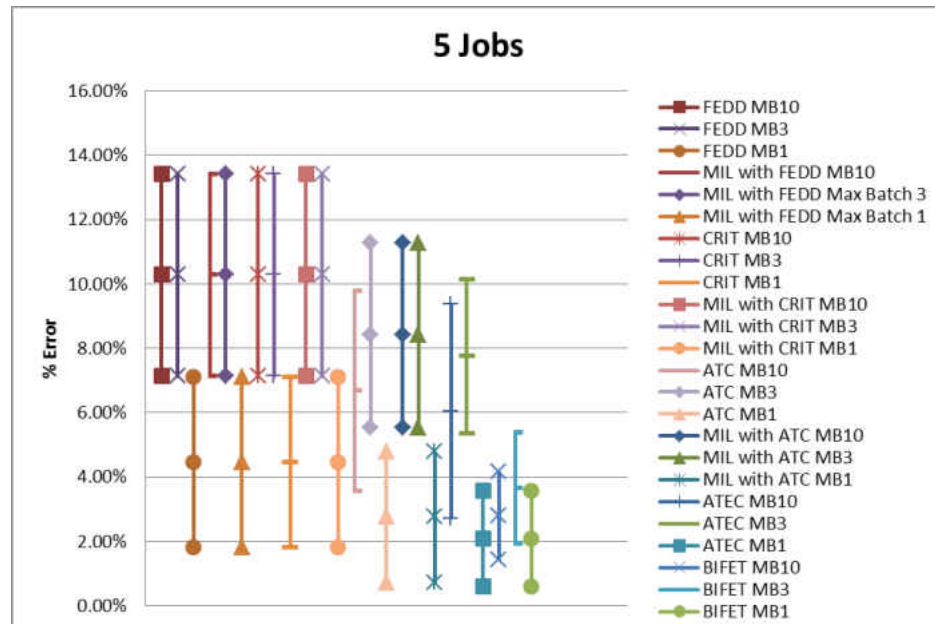


Figure 6.8. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, all jobs arriving simultaneously, and 5 families.

In the circumstances when jobs arrive at a high, medium, or low rate, the use of the MIL strategy has a large impact on the overall performance of the dispatching rule as well as reduces the variance of percent error (refer to Figure 6.9 through Figure 6.12). With the jobs flowing in at a slower rate, rather than being available all at the same time, there is a higher risk for the jobs to be completed earlier than the scheduled due date, thus impacting the overall performance objective. Controlling the release of the job to the machine ensures that the job will not be processed too early.

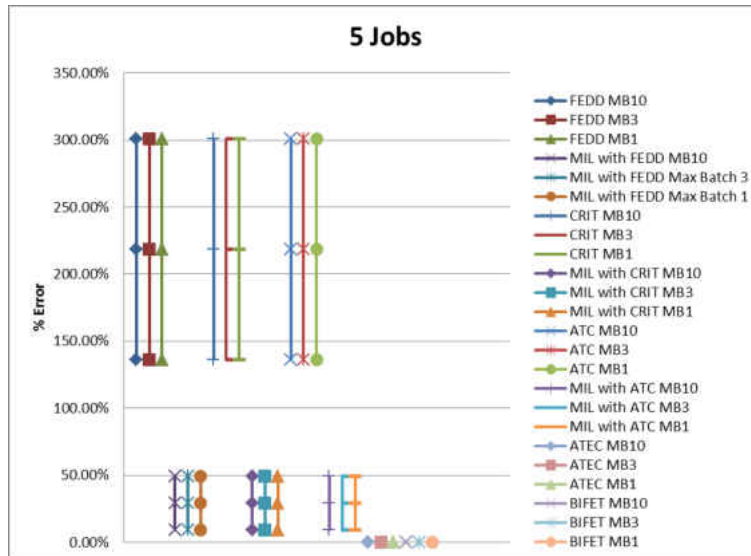


Figure 6.9. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, high traffic intensity, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.

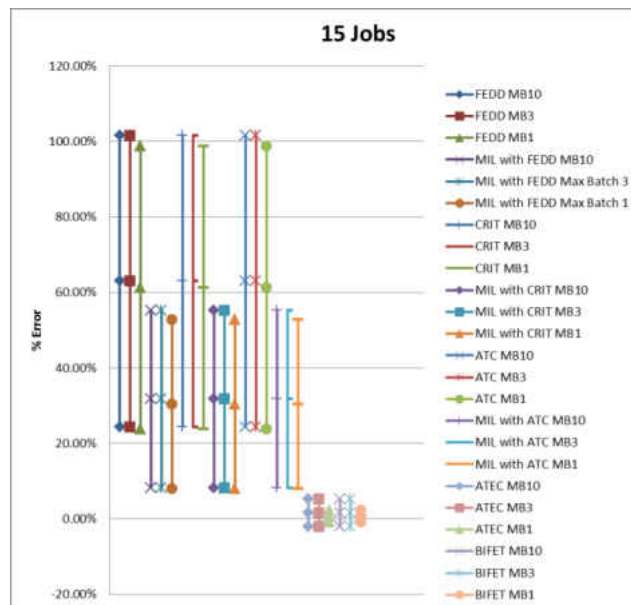


Figure 6.10. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, low traffic, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.

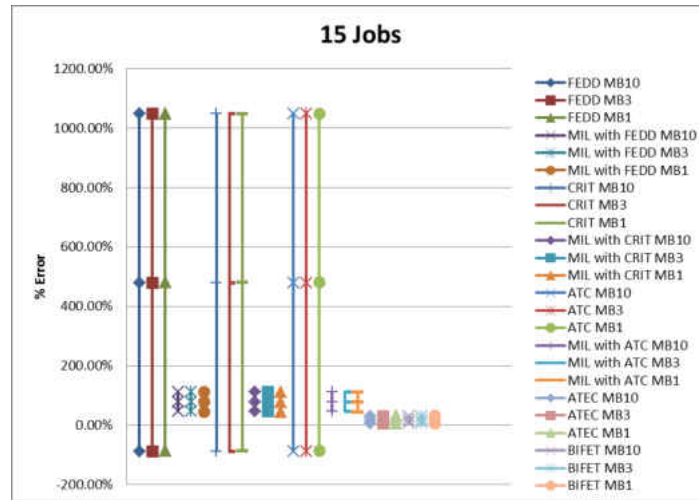


Figure 6.11. 95% confidence interval on the performance results for earliness penalty=10 and tardiness penalty=1, for tight due dates, high traffic intensity, 1 family, dispatching rules compared with ORR strategies combined with dispatching rules.

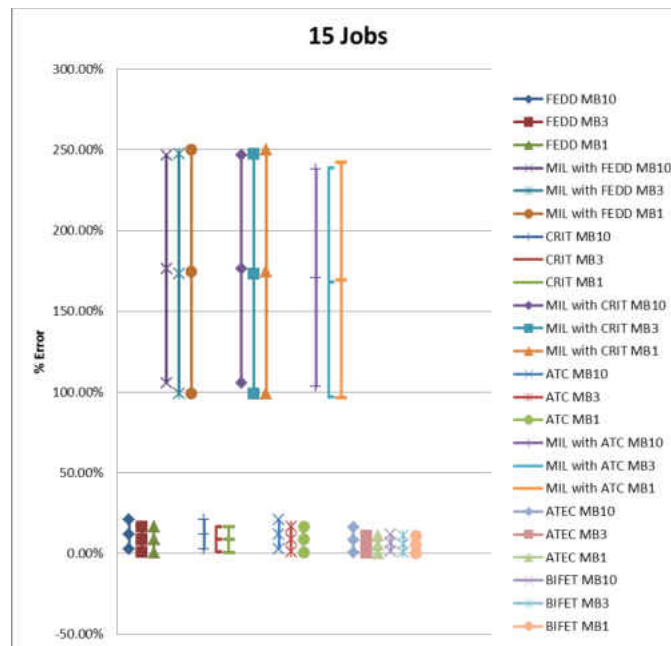


Figure 6.12. 95% confidence interval on the performance results for earliness penalty=1 and tardiness penalty=10, loose due dates, high traffic, 5 families, dispatching rules compared with ORR strategies combined with dispatching rules.

CHAPTER 7 : THE SINGLE BATCH WORKSTATION WITH UNEQUALLY-WEIGHTED PENALTIES AND VARYING BATCH SIZES AND INCOMPATIBLE FAMILIES AND SEQUENCE-DEPENDENT SETUPS

7.1 Introduction

In this final phase, the assumption that the setups are sequence-independent is relaxed. In this phase, minor and major setups are considered. This even further relaxed scheduling problem is expressed in the scheduling notation as follows: $1 \mid r_i, \text{batch}, s_{ij} \mid \sum(\theta_1 E_i + \theta_2 T_i)$. This problem now addresses sequence-dependent setups for the single batch processor scheduling problem minimizing the earliness-tardiness objective with unequal weights. In this phase, the proposed heuristics are integrated with ORR strategies in order to determine how robust these strategies are when introducing sequence-dependent setups.

7.2 Mathematical Model

Recall from 3.4 that Phase 3 of this research involves the following simplifying assumptions:

- The number of workstations m is 1, i.e., $|\mathbf{W}| = 1$;
- The number of machines in the workstation is 1; therefore, $m = 1$;
- The number of jobs n arrives individually to the workstation and are placed in batches before being processed;

- There is no restriction on queue length for jobs and batches awaiting processing at the workstation;
- Jobs within a batch are processed one at a time at the workstation;
- Jobs depart the workstation in batches;
- No batch preemption. Once a batch has begun processing at a workstation, it cannot be interrupted due to the arrival of higher priority jobs (or batches of jobs) before the processing of that batch has been completed;
- No batch splitting. Once a batch has been created, it cannot be divided into smaller sub-batches of jobs. It is treated as a single entity;
- Incompatible job families. Jobs from different families cannot be processed together in the same batch;
- *A machine's setup time for a batch from a family is determined not only by that batch's product family but also by the previous family for which the machine is currently setup; in other words, machine setup are family sequence-dependent;*
- *Major machine setups occur between processing batches of jobs from different families;*
- *Minor machine setups occur between processing batches of jobs from the same family;*
- The earliness penalty per unit time per job and the tardiness penalty per unit time per job are unequal.

This even further relaxed scheduling problem is expressed in the scheduling notation as follows: $1 \mid r_i, \text{batch}, s_{ij} \mid \sum (\theta_1 E_i + \theta_2 T_i)$. This problem now addresses sequence-dependent setups for the single batch processor scheduling problem minimizing the earliness/tardiness objective with unequal weights. In this phase, the proposed heuristics are integrated with ORR strategies in order to determine how robust these strategies are when introducing sequence-dependent setups.

Sets

J Set of jobs to be processed

F Set of product families

M Set of machines

Parameters

n Number of jobs, where $n = |\mathbf{J}|$

m Number of machines, where $m = |\mathbf{M}|$

F Number of families, where $F = |\mathbf{F}|$

i Index for jobs, where $i = 1, \dots, n$

j Index for machines, where $j = 1, \dots, m$

f Index for families, where $f = 1, \dots, F$

r_i Release time of job i

d_i Due date of job i

w_i Weight or importance of job i ; the jobs are equally-weighted, i.e., $w_i = 1$

θ_{1i}	Earliness penalty of job i
θ_{2i}	Tardiness penalty of job i
p_f	Common processing time of a job of family f
s_{f_{k-1},f_k}	Setup time for family f of batch k when it follows the family of the previous batch $k-1$
$s_{f_1f_1}$	Setup time for family f when it is the first batch to be processed
B_{\min}^f	Minimum batch size of family f
B_{\max}^f	Maximum batch size of family f
L_{fi}	1, if family f belongs to job i , otherwise 0

Decision Variables

X_{ik}	1, if job i is processed in batch k , otherwise 0.
Y_{fk}	1, if family f is processed in batch k , otherwise 0

Dependent Variables

E_i	Earliness of job i
T_i	Tardiness of job i
C_i	Completion time of job i
pb_k	Process time of batch k of family f
t_k	Start time of batch k

Cb_k Completion time of batch k

The following is the proposed mathematical model:

Minimize

$$\sum_{i=1}^n (\theta_1 E_i + \theta_2 T_i) \quad (7.1)$$

s.t.

$$E_i \geq d_i - C_i \quad \forall i \quad (7.2)$$

$$T_i \geq C_i - d_i \quad \forall i \quad (7.3)$$

$$C_i + E_i - T_i = d_i \quad \forall i \quad (7.4)$$

$$\sum_{i=1}^n X_{ik} \geq \sum_{f=1}^F (Bmin_f \times Y_{fk}) \quad \forall f; \forall k \quad (7.5)$$

$$\sum_{i=1}^n X_{ik} \leq \sum_{f=1}^F (Bmax_f \times Y_{fk}) \quad \forall f; \forall k \quad (7.6)$$

$$pb_1 = \sum_{i=1}^n (p_i \times X_{i1}) + \sum_{f=1}^F (s_{f(1),f(1)} \times Y_{f1}) \quad \forall f; k = 1 \quad (7.7)$$

$$pb_k = \sum_{i=1}^n (p_i \times X_{ik}) + \sum_{f=1}^F (s_{f(k-1),f(k)} \times Y_{fk}) \quad \forall f; k = 2 \dots n \quad (7.8)$$

$$\sum_{k=1}^n X_{ik} = 1 \quad \forall i; \forall k \quad (7.9)$$

$$\sum_{f=1}^F Y_{fk} \leq 1 \quad \forall f; \forall k \quad (7.10)$$

$$L_{fi} \times X_{ik} \leq Y_{fk} \quad \forall i; \forall f; \forall k \quad (7.11)$$

$$t_k \geq r_i X_{ik} \quad \forall i; \quad \forall k \quad (7.12)$$

$$t_k \geq Cb_{k-1} \quad \forall k \quad (7.13)$$

$$Cb_k = t_k + pb_k \quad \forall k \quad (7.14)$$

$$C_i = Cb_k X_{ik} \quad \forall i; \quad \forall k \quad (7.15)$$

$$E_i \geq 0 \quad \forall i \quad (7.16)$$

$$T_i \geq 0 \quad \forall i \quad (7.17)$$

$$r_i \geq 0 \quad \forall i \quad (7.18)$$

$$Cb_k \geq 0 \quad \forall k \quad (7.19)$$

$$C_i \geq 0 \quad \forall i \quad (7.20)$$

$$X_{ik} \in \{0,1\} \quad \forall i; \quad \forall k \quad (7.21)$$

$$Y_{fk} \in \{0,1\} \quad \forall f; \quad \forall k \quad (7.22)$$

Objective (7.1) minimizes the total tardiness and earliness over all jobs. Constraints (7.2) and (7.3) are the definitions of the earliness and tardiness of a job. Constraint (7.4) shows the definition of the due date for each job. Constraint (7.5) states that the sum of all jobs in a batch is greater than the minimum family batch size constraint for that batch. Constraint (7.6) states that the sum of jobs in a batch is less than the maximum family batch size constraint for that batch. Constraint (7.7) defines the process time for the first batch of jobs to be processed. Constraint (7.8) define the process time for the subsequent batches of jobs. Both constraints are the number in the batch times the individual processing time of each job plus the setup time associated for each family. The setup time is determined by the family of the batches that was

processed previously and the current family being processed. Constraint (7.9) says that each job is assigned to a batch only once. Constraint (7.10) states that each family is assigned to exactly one batch. If there are no jobs in the batch then no families would be assigned to a batch. Constraint (7.11) ensures that if a job in a family is assigned to a batch then that batch must be of the same family (i.e., if $X_{ik} = 1$, then $Y_{fk} = 1$). Constraint (7.12) ensures that the start time of a batch must be no less than the release time of any jobs forming that batch. Constraint (7.13) specifies that no job can start before the completion of the previous batch. Constraint (7.14) identifies that a job's start time must be early enough to allow for the processing time before it's completed. Constraint (7.15) states that the completion time of the job is equal to the completion time of the batch to which the job belongs. This is because jobs depart in batches. Constraints (7.16) through (7.20) are the negativity constraints. Constraints (7.21) through (7.22) are binary constraints.

The updated integer programming model, which now considers sequence-dependent setups, is also solved using IBM ILOG CPLEX version 12.5 on a 2.13 GHz Intel Pentium CPU Processor with 4.0 GB RAM and Microsoft Windows 7 operating system.

7.2.1 Computational Time of the Mathematical Model vs the Proposed and Benchmark Heuristics

The run time for each run scenario is recorded. The stopping time of 6 hours for Phase 2 and 3 is followed as it is in Phase 1. The percentage of nodes remaining to be tested is also recorded next to the completion time. For example, if the scenario is stopped after 6 hours and

50% of nodes remained to be tested, then both data are recorded along with the CPLEX solution at the stop time. As discussed earlier and as recognized by other researchers who have studied similar problems, the length of time to reach an optimal solution is impractical for real-world industrial settings and therefore heuristics should be pursued. In order to understand the impact of the increase in number of jobs, as well as adding complexity to the problem (such as sequence-dependent setups), the run time is graphed in Figure 7.1.

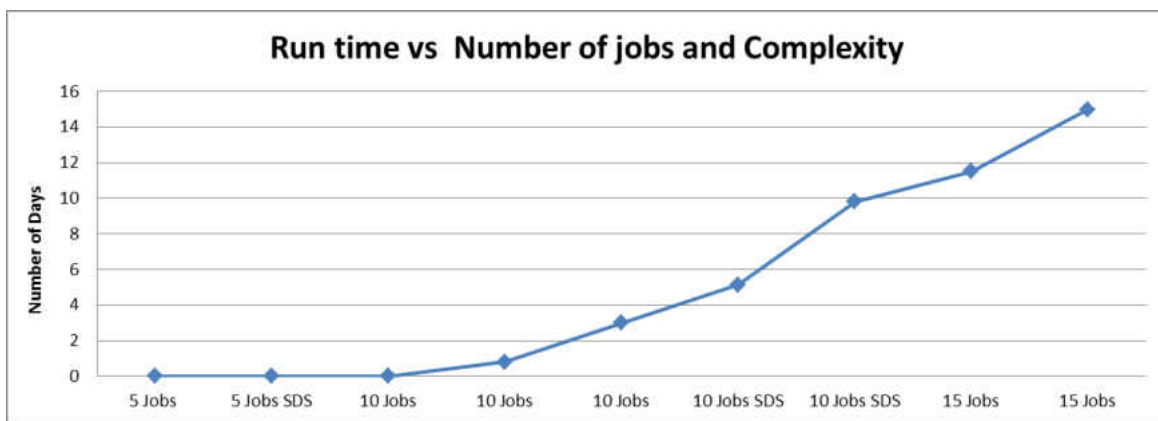


Figure 7.1. Computational time for CPLEX to arrive at a solution as problem complexity increases in terms of the number of jobs.

For each number of jobs (5, 10, 15), the number of days to completion for the various scenarios was averaged. If a particular run is stopped prematurely, the expected length of time is calculated by using the current percent complete, the run time, and extrapolating that in order to identify the run time at 100 percent completion. After reviewing the data, and by examining the slope in the Figure 7.1, it is infeasible to run the same data sets for 15 jobs with sequence-

dependent setups. Therefore, for $n = 15$, heuristic performance is compared against one another rather than to the optimal solution.

7.2.2 Experimental Design

Table 7.1 shows the experimental design for this phase. This is similar to the experimental design in CHAPTER 6, with one exception: the number of families. Since, setups are now dependent on which family has been processed previously, it is not necessary to look at scenarios with only one family type. Now, the run scenarios are set up to each be with 2 or more families.

Table 7.1. Experimental design with emphasis on unequal job weights and sequence-dependent setups

Factor	Values	Number of Values
Weight, θ_1, θ_2	$\theta_1 = \text{Earliness} = 1, \theta_2 = \text{Tardiness} = 1$ $\theta_1 = \text{Earliness} = 10, \theta_2 = \text{Tardiness} = 1$ $\theta_1 = \text{Earliness} = 1, \theta_2 = \text{Tardiness} = 10$	3
Number of Jobs, n	5, 10, 15	3
Number of Families, F	$U(2, 5)$	1
Traffic Intensity	<ul style="list-style-type: none"> • Low: 50% Machine Utilization • Medium: 75% Machine Utilization • High: 95% Machine Utilization • Highest: All Jobs Arrive Simultaneously 	4
Due Date Tightness	Tight: $U(0,2)$, Loose: $U(0,4)$	2
Replications per Problem		10
Total Number of Problem Instances		720

7.3 Discussion of the Performance Results

In the cases with sequence dependent setups, the proposed heuristics continued to perform superior to the general dispatching rules as well as the dispatching rules in combination with the ORR strategies.

7.3.1 Batch Size Impact

Table 7.2 shows the comparison of percent error between one batch size and another. For example, the percent error of MIL with ATC is shown with maximum batch size of 10 (MB10) minus the maximum batch size of 3 (MB3). If the heuristic MIL with ATC has a higher percent error from the optimal solution, then the difference between MB10 and MB3 is a positive % difference. If it is negative, then MB3 has a higher percent error, which means that the heuristic performs worse at a maximum batch size of 3.

As can be seen in Table 7.2, when batch sizes of BIFET vary, there is little impact on the overall performance of the heuristic in terms of percent error from the CPLEX solution. This is largely due to the fact that BIFET utilizes the Family Batch Constraint, which also regulates the number that can be grouped within in a batch. Often times this could even trump the maximum batch size entered by the user. This feature allows the heuristic BIFET to be more robust when batch sizes are changing which in turn also means the heuristic will maintain its robustness when the number of types of families change (the more different types of families for 5, 10, 15 jobs, the smaller the batch sizes will be). The table displaying the batch comparison for MIL with ATC is very similar to that of FEDD, CRIT and ATC.

Table 7.2. Comparison of MIL with ATC and BIFET performance of percent error when varying batch sizes.

5 Jobs with SDS						
Equal Penalties	MIL with ATC					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	0.00%	0.00%	-4.18%	7.78%	-4.18%	7.78%
<H, T, 5>	0.00%	0.00%	1.04%	3.28%	1.04%	3.28%
<M, T, 5>	0.00%	0.00%	2.59%	8.19%	2.59%	8.19%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	0.00%	0.00%	-3.37%	17.96%	-3.37%	17.96%
<H, L, 5>	0.00%	0.00%	5.86%	12.36%	5.86%	12.36%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

5 Jobs with SDS						
Equal Penalties	BIFET					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	0.00%	0.00%	0.53%	1.98%	0.53%	1.98%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	2.50%	7.91%	2.50%	7.91%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	0.00%	0.00%	8.26%	8.70%	8.26%	8.70%
<H, L, 5>	0.00%	0.00%	2.98%	9.43%	2.98%	9.43%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Early=10/Tardy=1	MIL with ATC					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-1.73%	3.84%	4.02%	5.61%	5.75%	4.98%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	2.53%	5.88%	29.51%	41.65%	26.98%	41.74%
<H, L, 5>	0.00%	0.00%	37.69%	119.20%	37.69%	119.20%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Early=10/Tardy=1	BIFET					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	0.00%	0.00%	2.09%	2.95%	2.09%	2.95%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	0.00%	0.00%	1.96%	4.34%	1.96%	4.34%
<H, L, 5>	0.00%	0.00%	3.02%	9.56%	3.02%	9.56%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Early=1/Tardy=10	MIL with ATC					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	0.00%	0.00%	6.69%	4.21%	6.69%	4.21%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	1.59%	5.04%	13.03%	19.94%	11.44%	19.56%
<H, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Early=1/Tardy=10	BIFET					
	MB10-MB3		MB10-MB1		MB3-MB1	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	0.00%	0.00%	2.09%	2.95%	2.09%	2.95%
<H, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, T, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<S, L, 5>	0.00%	0.00%	2.50%	6.94%	2.50%	6.94%
<H, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<M, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<LW, L, 5>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

7.3.2 Effects of Weighted the Earliness and Tardiness Penalties

The effects of varying weighted penalties can be seen in Table 7.3. It is apparent that the weights of the penalties affect the performance of the heuristics. BIFET continues to outperform the other dispatching rules and ORR strategies in the majority of the scenarios, with the

exception when all jobs arrive simultaneously and the due dates are tight. In those cases, BIFET performs just as well as the general dispatching rules paired with ORR strategies.

Table 7.3. Comparison of MIL with ATC and BIFET performance of percent error when varying penalties.

5 Jobs						
Max Batch 10	MIL with ATC					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	-21.13%	32.81%	6.39%	15.84%	27.53%	42.93%
<M, T, 5>	-61.75%	81.52%	-99.41%	145.01%	-37.66%	196.56%
<LW, T, 5>	36.46%	98.58%	-382.36%	977.13%	-418.82%	1075.63%
<S, L, 5>	3.73%	9.06%	-64.75%	108.09%	-68.47%	114.75%
<H, L, 5>	-83.36%	102.35%	-31.51%	116.47%	51.85%	174.82%
<M, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%
<LW, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%

5 Jobs						
Max Batch 10	BIFET					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	1.88%	3.06%	-8.43%	15.32%	-10.31%	15.83%
<M, T, 5>	1.37%	7.63%	-29.71%	67.02%	-31.07%	66.87%
<LW, T, 5>	-0.10%	0.33%	0.00%	0.00%	0.10%	0.33%
<S, L, 5>	-0.38%	1.20%	-15.24%	48.20%	-14.86%	47.00%
<H, L, 5>	-3.87%	30.79%	-60.17%	111.32%	-56.30%	125.87%
<M, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%
<LW, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%

5 Jobs						
Max Batch 3	MIL with ATC					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	-21.13%	32.81%	6.39%	15.84%	27.53%	42.93%
<M, T, 5>	-61.75%	81.52%	-99.41%	145.01%	-37.66%	196.56%
<LW, T, 5>	36.46%	98.58%	-382.36%	977.13%	-418.82%	1075.63%
<S, L, 5>	3.57%	9.12%	-64.75%	108.09%	-68.31%	114.79%
<H, L, 5>	-83.36%	102.35%	-31.51%	116.47%	51.85%	174.82%
<M, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%
<LW, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%

5 Jobs						
Max Batch 3	BIFET					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	1.88%	3.06%	-8.43%	15.32%	-10.31%	15.83%
<M, T, 5>	1.37%	7.63%	-29.71%	67.02%	-31.07%	66.87%
<LW, T, 5>	-0.10%	0.33%	0.00%	0.00%	0.10%	0.33%
<S, L, 5>	-0.38%	1.20%	-15.24%	48.20%	-14.86%	47.00%
<H, L, 5>	-3.87%	30.79%	-60.17%	111.32%	-56.30%	125.87%
<M, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%
<LW, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%

5 Jobs						
Max Batch 1	MIL with ATC					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	-21.01%	32.85%	6.22%	16.21%	27.23%	43.24%
<M, T, 5>	-64.35%	80.40%	-97.23%	145.01%	-32.89%	195.71%
<LW, T, 5>	36.46%	98.58%	-382.36%	977.13%	-418.82%	1075.63%
<S, L, 5>	-2.09%	5.49%	-24.22%	57.85%	-22.12%	59.25%
<H, L, 5>	-84.41%	99.06%	-31.51%	116.47%	52.90%	173.13%
<M, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%
<LW, L, 5>	-11835.45%	26758.43%	-3133.88%	6182.21%	8701.57%	21255.83%

5 Jobs						
Max Batch 1	BIFET					
	A-T		A-E		T-E	
<Ti,DT,F>	AVE	STD DEV	AVE	STD DEV	AVE	STD DEV
<S, T, 5>	-0.04%	0.12%	-2.62%	8.27%	-2.58%	8.16%
<H, T, 5>	1.88%	3.06%	-8.43%	15.32%	-10.31%	15.83%
<M, T, 5>	-1.13%	7.42%	-29.71%	67.02%	-28.57%	68.07%
<LW, T, 5>	-0.10%	0.33%	0.00%	0.00%	0.10%	0.33%
<S, L, 5>	-2.92%	4.37%	-10.23%	29.16%	-7.31%	29.10%
<H, L, 5>	-6.85%	29.13%	-54.00%	99.59%	-47.15%	108.94%
<M, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%
<LW, L, 5>	-90.71%	284.63%	-88.51%	285.52%	2.19%	424.42%

Figure 7.2 and Figure 7.3, take a closer look at how ATEC and BIFET perform when the job penalties are equal or the earliness penalty is more heavily weighted than the tardiness penalty. As stated previously, the dispatching rules with ORR strategies and the proposed

heuristics perform better, which appear that the ORR strategies handle the scheduling of jobs better when tardiness is not more heavily weighted. The ORR strategy might be delaying the release to the machine more so than needed which in turn creates more tardy jobs. When the tardiness is heavier weighted, this negatively impacts the overall objective function. This is displayed in Figure 7.4.



Figure 7.2. 95% confidence interval on the performance results for equal earliness and tardiness penalties, tight due dates, and high traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.

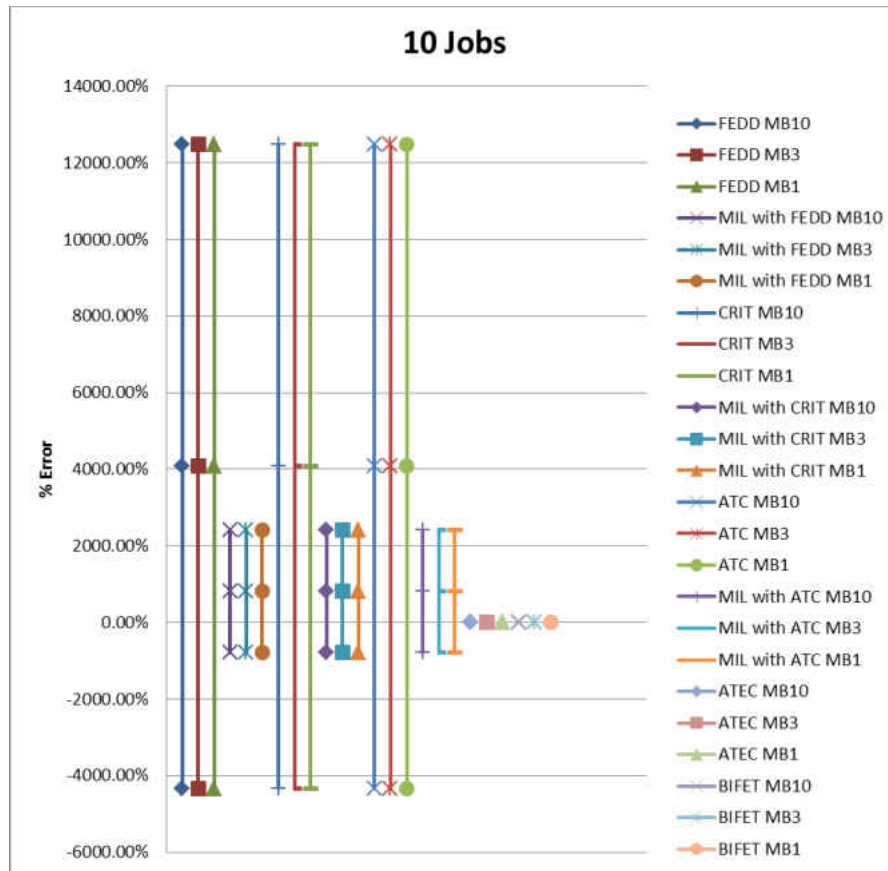


Figure 7.3. 95% confidence interval on the performance results for earliness penalty = 10 and tardiness = 1, tight due dates, and low traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.



Figure 7.4. 95% confidence interval on the performance results for tardiness penalty = 10, and earliness penalty = 1, loose due dates, and high traffic intensity, dispatching rules compared with ORR strategies combined with dispatching rules.

7.3.3 Overall Performance of Dispatching Rules and ORR versus Proposed Heuristics

The proposed heuristics in this research, ATEC and BIFET continue to outperform the general heuristics even when used in conjunction with ORR strategies.

Figure 7.5 to Figure 7.7 show the confidence intervals for 10 Jobs with sequence dependent setups. In Figure 7.5 (equal earliness and tardiness penalties) and Figure 7.6 (higher earliness penalties), it can be seen that the dispatching rules perform better with the ORR strategy MIL rather than having jobs immediately released to the machine according to their

arrival time. However, in Figure 7.7, where the tardiness penalty is weighted higher than earliness, the dispatching rules perform worse when used in conjunction with the ORR strategies. This is evident that delaying the release time of the jobs in order to avoid earliness, creates more tardy jobs. Since, in this scenario, tardiness is weighted higher, the overall performance objective worsens.

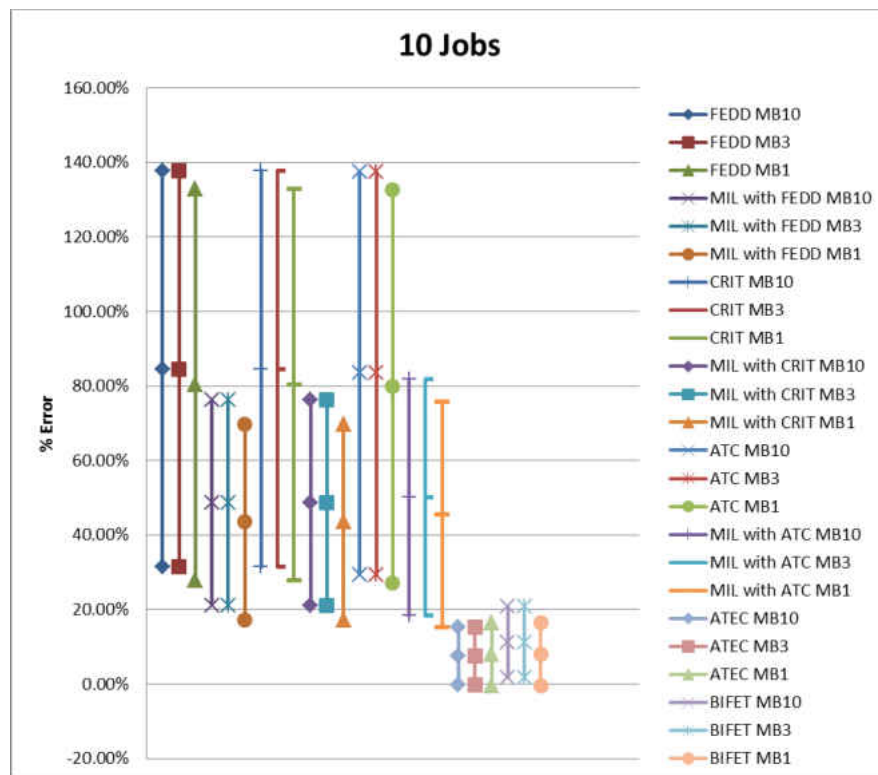


Figure 7.5. 95% confidence interval on the performance results for equal earliness and tardiness penalties, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.



Figure 7.6. 95% confidence interval on the performance results for tardiness penalty = 1, and earliness penalty = 10, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.

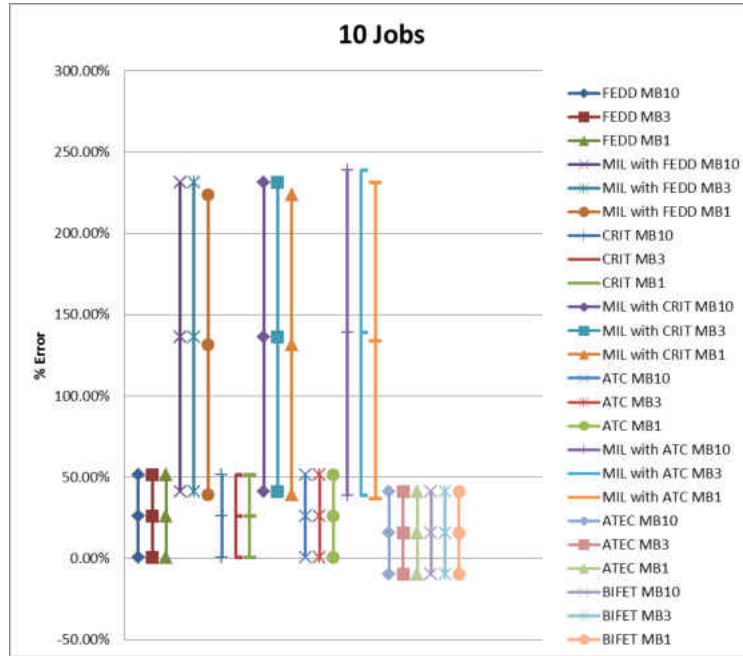


Figure 7.7: 95% confidence interval on the performance results for earliness penalty =1 and tardiness =10, loose due dates, high traffic, dispatching rules compared with ORR strategies combined with dispatching rules.

7.3.3.1 Performance of Proposed Heuristics Versus the Benchmark Heuristic for 15 Jobs

Since the optimal solution for 15 jobs could not be found in a reasonable amount of time, the proposed heuristics are compared against the best performing general heuristic. After examining the results, it is determined that ATC with MIL works best over FEDD and CRIT (with or without MIL) when the penalties are either equal or earliness is rated higher. In the case where tardiness is more heavily weighted, ATC works best on its own.

In order to get an overall idea how the general heuristics perform compared to one another, the Table 7.4 is constructed to compare the overall percent error average. As shown in Table 7.4, the ATC rule with MIL has the lowest average percent error in the case where the

penalties are equally weighted and when earliness is weighted heavier. ATC alone has a lower overall percent error when tardiness is weighted heavier.

Table 7.4. Average percent error for heuristics.

	10 Jobs						5 Jobs					
	Equal		Early=10/Tardy=1		Early=1/Tardy=10		Equal		Early=10/Tardy=1		Early=1/Tardy=10	
	Average	Std Dev	Average	Std Dev	Average	Std Dev	Average	Std Dev	Average	Std Dev	Average	Std Dev
FEDD Max Batch 10	883%	1591%	544%	604%	10%	6%	1118%	1861%	10731%	17863%	964%	1751%
FEDD Max Batch 1	888%	1588%	542%	600%	15%	11%	1118%	1861%	10725%	17867%	965%	1750%
FEDD Max Batch 3	881%	1592%	543%	605%	10%	6%	1118%	1861%	10731%	17863%	964%	1751%
MIL with FEDD Max Batch 10	323%	533%	59%	44%	71%	53%	514%	784%	1335%	2154%	3463%	6197%
MIL with FEDD Max Batch 1	329%	529%	60%	38%	79%	50%	514%	783%	1336%	2154%	3464%	6196%
MIL with FEDD Max Batch 3	321%	534%	58%	45%	70%	54%	514%	784%	1335%	2154%	3463%	6197%
CRIT Max Batch10	883%	1591%	544%	604%	10%	6%	1118%	1861%	10731%	17863%	964%	1751%
CRIT Max Batch1	888%	1588%	542%	600%	15%	11%	1118%	1861%	10725%	17867%	965%	1750%
CRIT Max Batch3	881%	1592%	543%	605%	10%	6%	1118%	1861%	10731%	17863%	964%	1751%
MIL with CRIT Max Batch10	339%	541%	59%	44%	71%	53%	514%	784%	1335%	2154%	3463%	6197%
MIL with CRIT Max Batch1	329%	529%	60%	38%	79%	50%	514%	783%	1336%	2154%	3464%	6196%
MIL with CRIT Max Batch3	321%	534%	58%	45%	70%	54%	514%	784%	1335%	2154%	3463%	6197%
ATC Max Batch 10	883%	1592%	545%	604%	9%	6%	1118%	1861%	10738%	17858%	963%	1751%
ATC Max Batch 1	887%	1589%	541%	601%	14%	9%	1118%	1861%	10727%	17865%	965%	1750%
ATC Max Batch 3	881%	1592%	544%	605%	9%	6%	1118%	1861%	10738%	17858%	963%	1751%
MIL with ATC Max Batch 10	322%	533%	57%	44%	71%	57%	514%	783%	1345%	2148%	3464%	6196%
MIL with ATC Max Batch 1	329%	529%	59%	39%	79%	55%	514%	783%	1339%	2152%	3465%	6196%
MIL with ATC Max Batch 3	320%	534%	56%	45%	71%	57%	514%	783%	1345%	2148%	3464%	6196%

Therefore, MIL with ATC is used as a benchmark heuristic to compare the performance of the proposed heuristics. Table 7.5 to

Table 7.10 summarize the results of the proposed heuristics and the two benchmark heuristics when earliness and tardiness penalties are equal, when earliness is weighted heavier, and when tardiness is weighted heavier. A negative average indicates that the benchmark heuristic performs worse than the heuristic being compared against. In Table 7.5, it is interesting to see that ATC performs just as well or even better when the traffic intensity is simultaneous or high, with the exception when the due tightness is loose. When earliness is more heavily weighted, the use of the ORR strategy, MIL, in combination with ATC proves to be much more effective in minimizing the performance objective.

Table 7.5. Comparison of rule performance for ATC against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.

Performance relative to MIL with ATC							
<Ti,DT,F>		ATC MB10		ATC MB3		ATC MB1	
		Average % Error	Std Deviation	Average % Error	Std Deviation	Average % Error	Std Deviation
Equal	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	-5.6%	17.2%	-7.7%	18.6%	-5.6%	17.2%
	<M, T, 5>	4.2%	33.5%	3.9%	34.1%	4.2%	33.5%
	<LW, T, 5>	-3.7%	19.4%	-4.8%	21.6%	-3.7%	19.4%
	<S, L, 5>	-1.6%	5.1%	-2.6%	8.4%	-1.6%	5.1%
	<H, L, 5>	-16.3%	15.7%	-27.1%	16.5%	-16.3%	15.7%
	<M, L, 5>	52.8%	48.1%	52.8%	48.1%	52.8%	48.1%
	<LW, L, 5>	45.3%	51.2%	45.3%	51.2%	45.3%	51.2%
Early	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	0.6%	8.6%	-3.2%	8.9%	0.6%	8.6%
	<M, T, 5>	407.5%	194.5%	431.6%	227.3%	407.5%	194.5%
	<LW, T, 5>	143.2%	159.0%	143.2%	159.0%	143.2%	159.0%
	<S, L, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, L, 5>	128.5%	98.8%	97.1%	65.2%	128.5%	98.8%
	<M, L, 5>	596.7%	261.5%	597.3%	261.3%	596.7%	261.5%
	<LW, L, 5>	596.7%	261.5%	597.3%	261.3%	596.7%	261.5%

Table 7.6 and Table 7.7 show how the proposed heuristics perform against MIL with ATC. Both BIFET and ATEC outperform MIL with ATC, with the exception that BIFET does not perform as well in the cases where jobs arrive simultaneously and the maximum batch sizes are set to the extremes (a maximum batch size of 10 and a maximum batch size of 1).

Table 7.6. Comparison of rule performance for ATEC against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.

<TI,DT,F>		ATEC MB10		ATEC MB3		ATEC MB1	
		Average % Error	Std Deviation	Average % Error	Std Deviation	Average % Error	Std Deviation
Equal	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	-13.6%	11.6%	-14.3%	11.9%	-13.6%	11.6%
	<M, T, 5>	-31.9%	12.7%	-32.5%	12.7%	-31.9%	12.7%
	<LW, T, 5>	-20.3%	11.3%	-21.5%	13.3%	-20.3%	11.3%
	<S, L, 5>	-3.2%	9.0%	-3.7%	11.9%	-3.0%	9.0%
	<H, L, 5>	-22.0%	17.1%	-28.3%	18.2%	-22.0%	17.1%
	<M, L, 5>	-32.6%	6.2%	-32.6%	6.2%	-32.6%	6.2%
	<LW, L, 5>	-31.6%	6.6%	-31.6%	6.6%	-31.6%	6.6%
Early	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	-7.5%	7.2%	-8.8%	7.3%	-7.5%	7.2%
	<M, T, 5>	-17.1%	24.4%	-14.5%	21.7%	-17.1%	24.4%
	<LW, T, 5>	-26.0%	10.8%	-26.0%	10.8%	-26.0%	10.8%
	<S, L, 5>	-2.4%	4.9%	0.0%	0.0%	-2.3%	4.7%
	<H, L, 5>	9.5%	37.2%	-6.6%	8.9%	9.5%	37.2%
	<M, L, 5>	-41.4%	22.5%	-41.4%	22.5%	-41.4%	22.5%
	<LW, L, 5>	-41.4%	22.5%	-41.4%	22.5%	-41.4%	22.5%

Table 7.7. Comparison of rule performance for BIFET against MIL with ATC for equal penalties and earliness penalty=10/tardiness penalty =1.

Performance relative to MIL with ATC							
<TI,DT,F>		BIFET MB10		BIFET MB3		BIFET MB1	
		Average % Error	Std Deviation	Average % Error	Std Deviation	Average % Error	Std Deviation
Equal	<S, T, 5>	12.6%	9.4%	0.0%	0.0%	18.3%	9.4%
	<H, T, 5>	-3.7%	16.6%	-14.3%	11.9%	-3.7%	16.6%
	<M, T, 5>	-32.2%	12.3%	-32.5%	12.7%	-32.2%	12.3%
	<LW, T, 5>	-20.3%	11.3%	-21.5%	13.3%	-20.3%	11.3%
	<S, L, 5>	10.7%	16.6%	-3.7%	11.9%	16.9%	17.5%
	<H, L, 5>	-15.1%	19.0%	-28.3%	18.2%	-13.4%	21.8%
	<M, L, 5>	-32.6%	6.2%	-32.6%	6.2%	-32.6%	6.2%
	<LW, L, 5>	-31.6%	6.6%	-31.6%	6.6%	-31.6%	6.6%
Early	<S, T, 5>	11.5%	9.3%	0.0%	0.0%	17.2%	7.6%
	<H, T, 5>	2.9%	11.9%	-8.8%	7.3%	2.9%	11.9%
	<M, T, 5>	-16.7%	22.7%	-14.5%	21.7%	-16.7%	22.7%
	<LW, T, 5>	-26.0%	10.8%	-26.0%	10.8%	-26.0%	10.8%
	<S, L, 5>	11.8%	9.2%	0.0%	0.0%	18.2%	10.0%
	<H, L, 5>	2.3%	23.4%	-6.6%	8.9%	2.3%	23.4%
	<M, L, 5>	-41.4%	22.5%	-41.4%	22.5%	-41.4%	22.5%
	<LW, L, 5>	-41.4%	22.5%	-41.4%	22.5%	-41.4%	22.5%

Since ATC seems to perform better than MIL with ATC in the cases of 5 and 10 jobs when the tardiness penalty is higher, BIFET and ATEC are compared to ATC in Table 7.9 and

Table 7.10. ATC is compared to MIL with ATC in Table 7.8 to ensure that its performance continues to be superior with 15 jobs. After examining the data in the table, it's continues to be the case. ATEC continues to perform stronger overall than ATC. BIFET performs better when the traffic intensity is medium or low. ATEC performs better in the case where jobs arrive simultaneously than does BIFET. In the scenarios where the jobs arrive simultaneously or traffic

is high, BIFET's family batch constraint parameter might be restricting the heuristic too much, in which that it is not batching effectively enough. It is recommended that in cases where the traffic is high that the user experiments with the family batch constraint in order to find the best performance.

Table 7.8. Comparison of rule performance for MIL ATC against ATC for earliness penalty =1/tardiness penalty =10.

Performance relative to ATC							
<TI,DT,F>		MIL with ATC MB10		MIL with ATC MB3		MIL with ATC MB1	
		Average % Error	Std Dev	Average % Error	Std Dev	Average % Error	Std Dev
Tardy	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	13.0%	17.6%	16.3%	18.1%	13.0%	17.6%
	<M, T, 5>	134.6%	74.9%	141.6%	78.2%	134.6%	74.9%
	<LW, T, 5>	24.3%	16.4%	24.3%	16.4%	24.3%	16.4%
	<S, L, 5>	0%	0%	0%	0%	0%	0%
	<H, L, 5>	70.9%	44.8%	97.5%	58.8%	70.9%	44.8%
	<M, L, 5>	81.7%	32.4%	81.5%	32.5%	81.7%	32.4%
	<LW, L, 5>	81.7%	32.4%	81.5%	32.5%	81.7%	32.4%

Table 7.9. Comparison of rule performance for ATEC against ATC for earliness penalty =1/tardiness penalty =10.

Performance relative to ATC							
<TI,DT,F>		ATEC MB10		ATEC MB3		ATEC MB1	
		Average % Error	Std Dev	Average % Error	Std Dev	Average % Error	Std Dev
Tardy	<S, T, 5>	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	<H, T, 5>	-1.5%	4.1%	0.0%	0.1%	-1.5%	4.1%
	<M, T, 5>	-1.2%	2.7%	-1.2%	2.7%	-1.2%	2.7%
	<LW, T, 5>	-1.9%	2.1%	-1.9%	2.1%	-1.9%	2.1%
	<S, L, 5>	0%	0%	0%	0%	0%	0%
	<H, L, 5>	-1.6%	2.8%	-0.2%	0.6%	-1.6%	2.8%
	<M, L, 5>	-9.6%	9.9%	-9.6%	9.9%	-9.6%	9.9%
	<LW, L, 5>	-9.6%	9.9%	-9.6%	9.9%	-9.6%	9.9%

Table 7.10. Comparison of rule performance for BIFET against ATC for earliness penalty =1/tardiness penalty =10.

Performance relative to ATC							
<TI,DT,F>		BIFET MB10		BIFET MB3		BIFET MB1	
		Average % Error	Std Dev	Average % Error	Std Dev	Average % Error	Std Dev
Tardy	<S, T, 5>	11.5%	9.3%	0.0%	0.0%	17.2%	7.6%
	<H, T, 5>	9.8%	8.8%	0.0%	0.1%	9.8%	8.8%
	<M, T, 5>	-1.2%	2.7%	-1.2%	2.7%	-1.2%	2.7%
	<LW, T, 5>	-1.9%	2.1%	-1.9%	2.1%	-1.9%	2.1%
	<S, L, 5>	16%	10%	0%	0%	24%	8%
	<H, L, 5>	6.5%	8.1%	-0.2%	0.6%	6.5%	8.1%
	<M, L, 5>	-9.6%	9.9%	-9.6%	9.9%	-9.6%	9.9%
	<LW, L, 5>	-9.6%	9.9%	-9.6%	9.9%	-9.6%	9.9%

CHAPTER 8 : SUMMARY AND FUTURE RESEARCH DIRECTIONS

8.1 Summary

The objective of this research is to understand the effects of batching, traffic intensity, sequence-dependent setups, incompatible job families, and penalties on the objective function to minimize earliness and tardiness penalties. CHAPTER 1 discusses the case study after which the problem has been modeled. CHAPTER 2 summarizes the relevant literature in which researchers conclude that the problem, in its simplest form, is NP-Hard. As a result, many researchers choose to utilize heuristics, since arriving at an optimal solution in a reasonable time is unrealistic especially in industry, where decisions need to be made quickly. CHAPTER 3 describes how the problem is dissected into more manageable phases in order to understand the gap in performance with general heuristics. This provides a framework to develop a new heuristic to improve the minimization of the objective function. The mathematical model of the phase 1 approach and the computational time required by CPLEX to solve optimally the model is provided in CHAPTER 4. A comparison of the new proposed heuristics against certain benchmark heuristics is developed in CHAPTER 5. During the development and analysis of the new heuristics it was evident that the current benchmark heuristics do not consider the earliness portion of the objective function. CHAPTER 6 looks at how the heuristics perform when the batch sizes are varied and unequal penalties in the objective function are introduced. The ORR strategies IMM and MIL are discussed and incorporated with the general dispatching rules. This

provides a more even comparison against the proposed heuristics because it looks at releasing the jobs based off the due date rather than immediately releasing them in queue when available. CHAPTER 7 relaxes the sequence independent setups assumption and introduces minor and major changeovers. Using the ORR strategies introduced in CHAPTER 6, the proposed heuristics are compared against the general dispatching rules with the added complexity of sequence-dependent setups. With this added complexity, the time to solve becomes exponentially long and unrealistic to solve optimally for 15 jobs.

8.2 Future Research Directions

The research studied in this dissertation and its conclusions provide a solid foundation for further development and exploration.

8.2.1 Modeling Multiple Serial Machines in a Flow shop

Recall in Chapter 1 the case study, which this research was modeled after, contained multiple, serial, work stations. As a result of the complexity of the initial problem, the assumption of a single machine was made in order to first study the main bottleneck to achieve performance improvements. Now that the bottleneck has been studied and an effective solution has been found, the work could be extended to schedule the work for the remainder of the flow in the shop. It is important to analyze the remaining workstations within the flow of the shop in order to ensure that the bottleneck is not being starved or overrun.

The ORR strategy, MIL, considers multiple work stations in its calculation.

$$RD_i = d_i - k_1 n - k_2 Q_i$$

The planning factor k_1 is the number of operations in the job's routing.

Lu et al. (2011) and Baykasoglu et al. (2009) studied the performance that varying ORR strategies and dispatching rules had in a multiple serial machine flow shop. Lu et al. (2011) looks at how well ORR strategies minimized Mean Absolute Deviation (MAD) at three varying machine utilization levels, while Baykasoglu et al. (2009) focuses on tardiness-related performance metrics. As a result, job earliness is not considered. It would be beneficial to understand how it impacts earliness especially when looking at machine utilization. If the machine shop is producing too many jobs unnecessarily early, that will drive up the utilization of the machines and create bottlenecks. These papers also did not consider setup times, batching or incompatible job families.

Currently ATEC and BIFET do not account for multiple workstations that a job would visit in its routing. In order to accommodate for this, the methodology in both heuristics will need to be adjusted. The wait time at each station and the number of stations in a job's routing should be considered when improving both heuristics in order to be able to extend their use to a multiple machine case study.

8.2.2 Modeling Multiple Parallel Machines

Balanced scheduling in a job shop environment is an effective way to ensure jobs are finished on time and resources are being used to their full ability. However, in cases where the

traffic intensity is high and the amount of work in process inhibits on-time delivery, the use of multiple, identical, parallel machines at the bottleneck workstation could be considered. The proposed heuristics are a good way to handle the scheduling of work, but would need to be modified to accommodate multiple parallel machines. The number of machines at the workstation would need to be incorporated into the heuristic. Another concept that could be explored is identifying intelligently which machine should be used for a particular job. In other words, should the jobs with longer processing times be sent to one particular machine, or should all families with large setup times get processed on a particular machine. Uzsoy (1995) explores this problem with incompatible job families and utilizes list scheduling as a technique in order to minimize maximum completion (C_{\max}). He proposes the algorithm Batch Longest Processing Time (BLPT) which orders the batches in decreasing order of their processing times, then sends the batch to the machine that is available. This idea is to reduce idle time at the machine. While this works well to reduce C_{\max} , jobs could be processed on machines too early. Incorporating ORR strategies in this problem could be an effective solution to minimize earliness as well as tardiness.

In a more recent study, Malve and Uzsoy (2005) implement a genetic algorithm to reduce maximum lateness (L_{\max}) on multiple parallel machines with dynamic job arrivals and incompatible job families. In this paper they look at the effects of varying batch sizes, number of machines and traffic intensity of job arrivals. They also extend Uzsoy's previous algorithm RDU to account for multiple parallel machines. In addition they develop two new heuristics which were found to be effective overall, but less effective when the traffic intensity is high or

there is a large congestion of work. Again, in this paper as well as the previous, the author's main performance metric is aligned with minimizes tardiness.

8.2.3 Machine scheduling with Job Preemption

Another assumption that is made in Chapter 3 is that job preemption is not allowed. In other words, jobs with a higher priority could not interrupt the work that is currently being processed. This problem has been studied by researchers such as Jeong and Kim(1998), Min and Yih(2003), Upsani and Uzsoy(2008), to name a few. They focused on using dispatching rules real time in order to make scheduling changes as 'hot' jobs arrive and interrupt current work.

The experiment would need to examine the number of hot job interruptions, or how frequently they occur in the system. The performance of the ORR strategies and dispatching rules can be measured at various intervals of interruptions. For example, if there is a low frequency of interruption, versus medium, or a high number of interruptions.

Sequence-dependent setups might also play an important role in job shop scheduling when there is a requirement to stop current production of a job and accommodate a new job that may be of a different family. It would be interesting to see if the proposed heuristics would be robust when batches are split due to interruptions or if they will need to be modified to be more robust to the additional setups incurred as a result of preemption.

8.2.4 Due Date Time Windows

Another aspect of the case study is the idea of using due date time windows in which an upper and lower bound are present for the due dates. For the purposes of this research, each job had a unique hard due date. Rather than having a singular due date for each job, it would be interesting to explore if those dates had some flexibility and how the proposed heuristics would perform against this change. Koulamas (1996) and Wan and Yen (2002) have researched heuristics and algorithms to effectively schedule jobs in order to minimize earliness and tardiness. Koulamas (1996) shows that the scheduling problem, with due date time windows, is NP Hard for a single machine. He also proposes several heuristics that are tested at various ranges of due dates as well as an average tardiness factor. He found that his proposed heuristics work well when sequencing jobs in order to minimize tardiness and then show how to optimally insert idleness time in order to minimize earliness. This was especially true when altering the start time of the job was the main focus when scheduling rather than when the completion time of the job will be.

Wan and Yen(2002) also looks at a single machine and sets the experiments at varying job sizes and ranges of due dates as well as a tardiness factor. They use a tabu search method in conjunction with an optimal timing algorithm in order to create a schedule. In their research, tardiness was also more heavily weighted than earliness. They assumed that tardiness would be less desirable than earliness. The tabu search method used in conjunction with the timing algorithm was compared to optimal solutions found by the branch and bound techniques. The proposed methodology proved to be effective in the tested problem instances. None of the above

authors consider, batching, incompatible job families, sequence-dependent setups, or varying job arrival rates.

The proposed heuristics in this research would need to be modified in order to be more flexible and to be able to recognize and work with the upper and lower bounds of the due dates, rather than focusing on a singular date. It would also be beneficial to see whether it would be more advantageous to shoot for the lower, middle or upper bound of that due date window. Scheduling jobs in order to meet the lower bound of the due date window could result in jobs being processed too early, and creating unnecessary bottlenecks in the workstation. However, if the jobs are being scheduled with the upper bound in mind, it could create a situation where the machines are being under-utilized for a portion of the time, and then over-utilized as the upper bound time due date approaches, causing late shipments and a bottleneck later in the delivery schedule. The other thought is to take the approach Koulamas had and focus on the start date of the job rather than when the job will be completed in the due date window.

APPENDIX A: EQUAL WEIGHTS

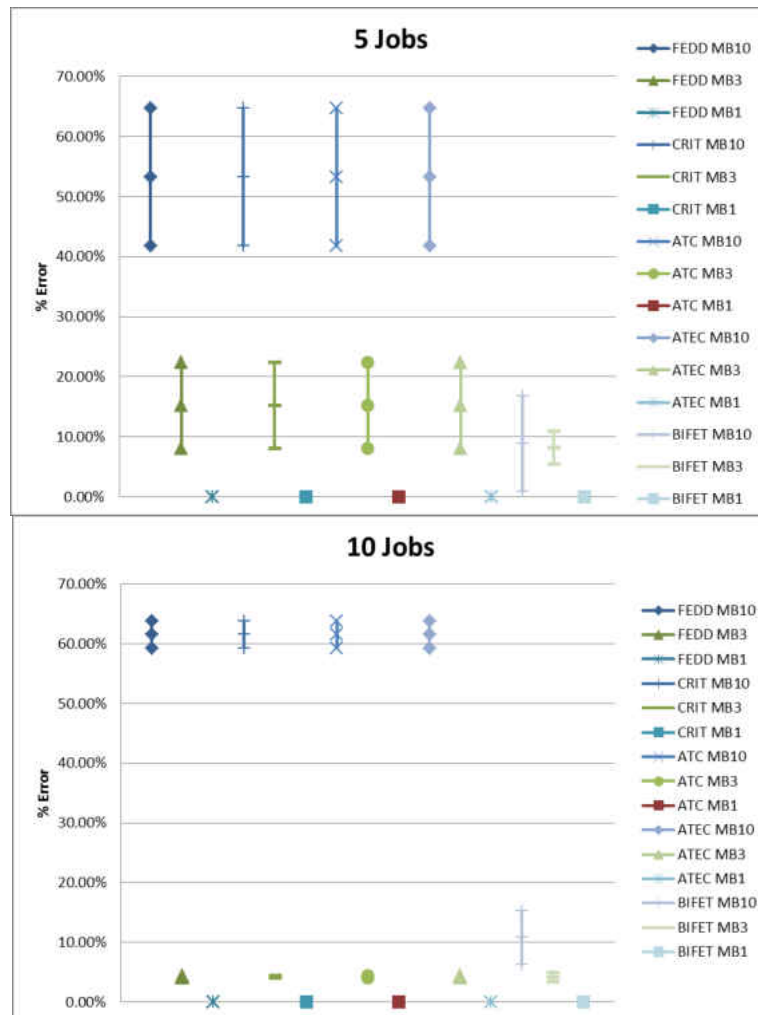


Figure A 1: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 1 family.

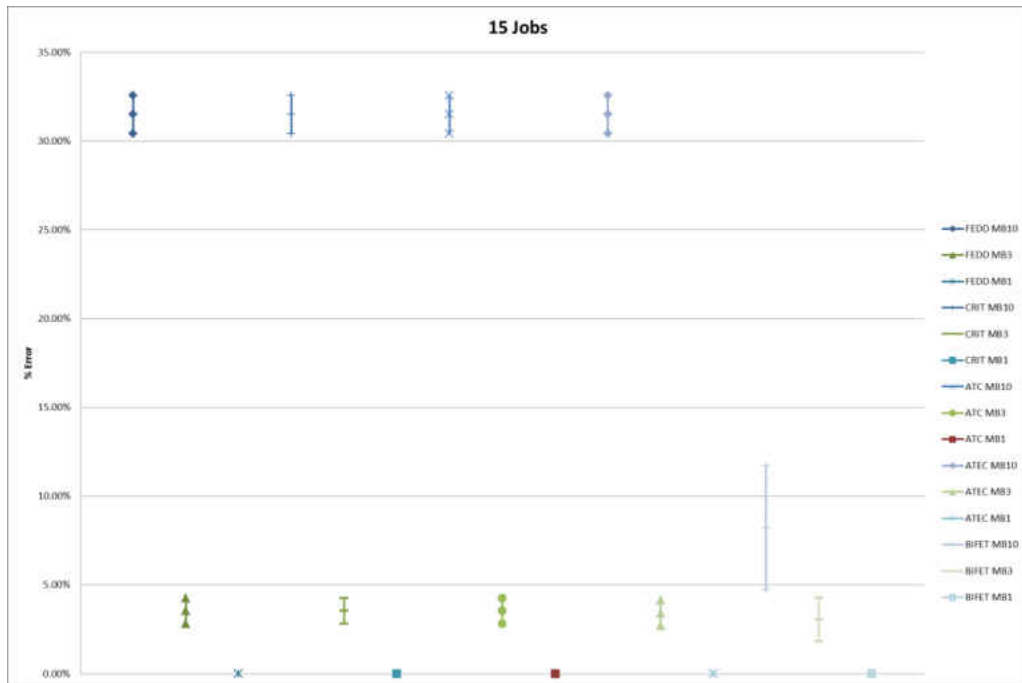


Figure A 2: 95% confidence interval on the performance results tight due dates, jobs arrive simultaneously, 1 family.

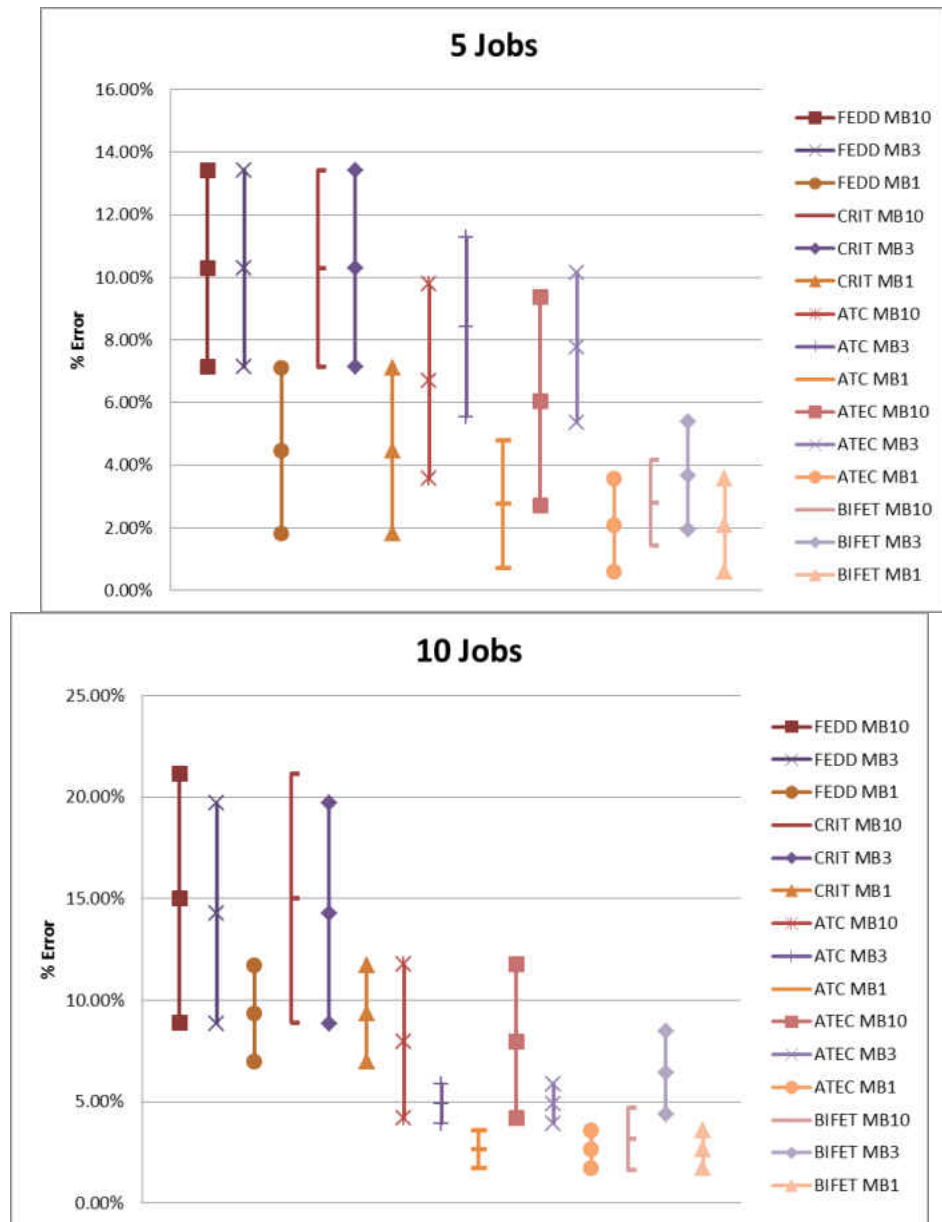


Figure A 3: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 5 families.

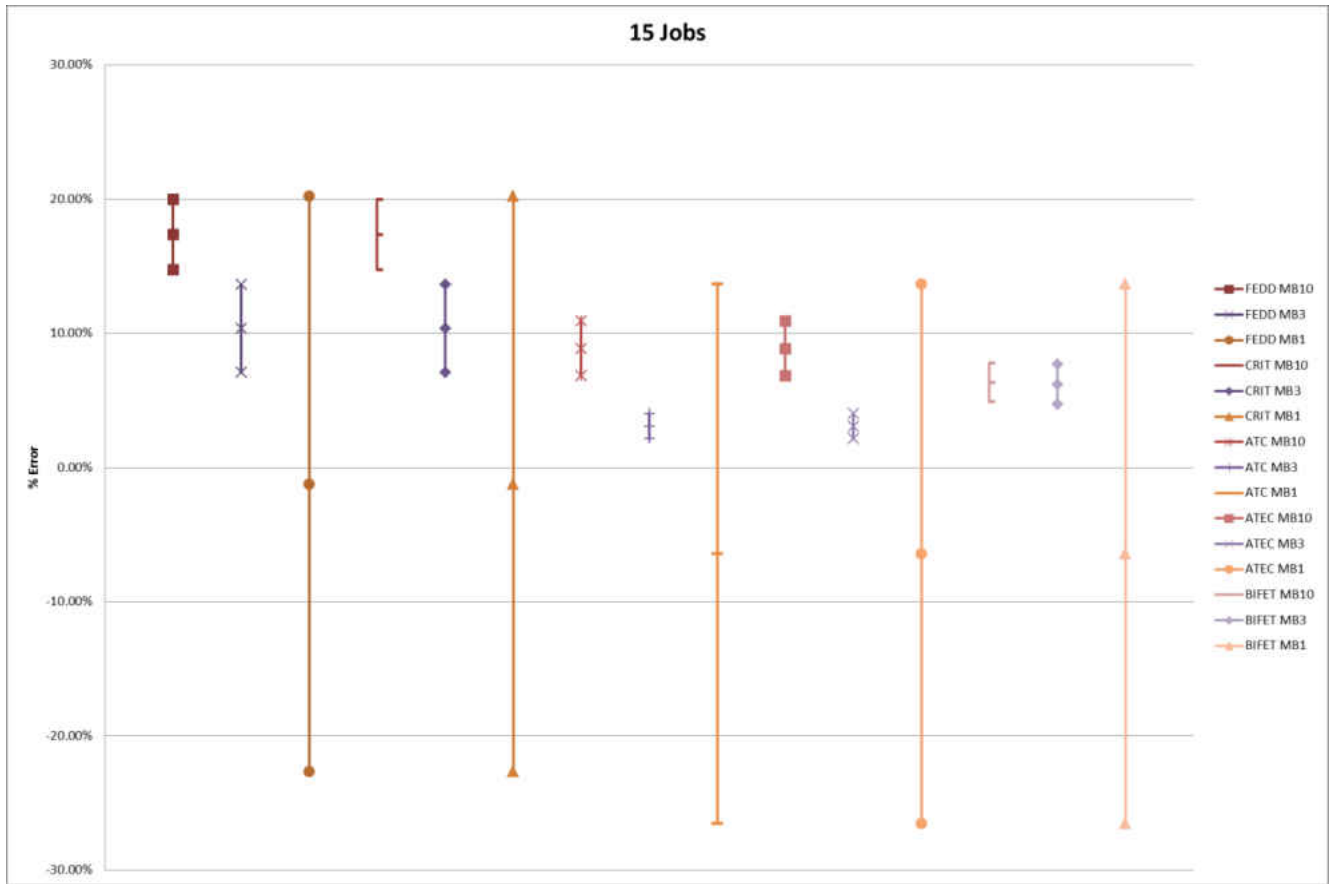


Figure A 4: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 5 families.

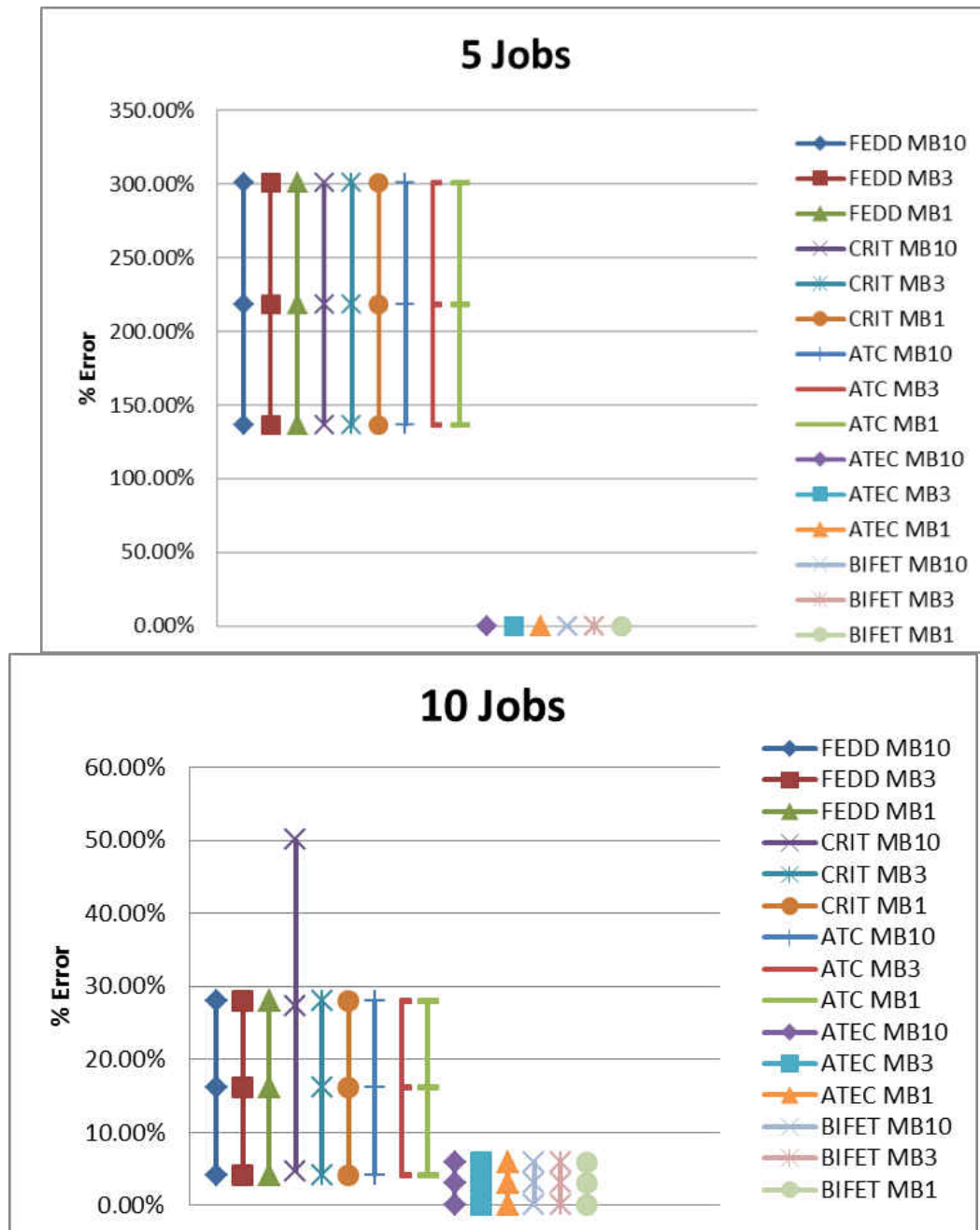


Figure A 5: 95% confidence interval on the performance results for tight due dates, high traffic, 1 family.

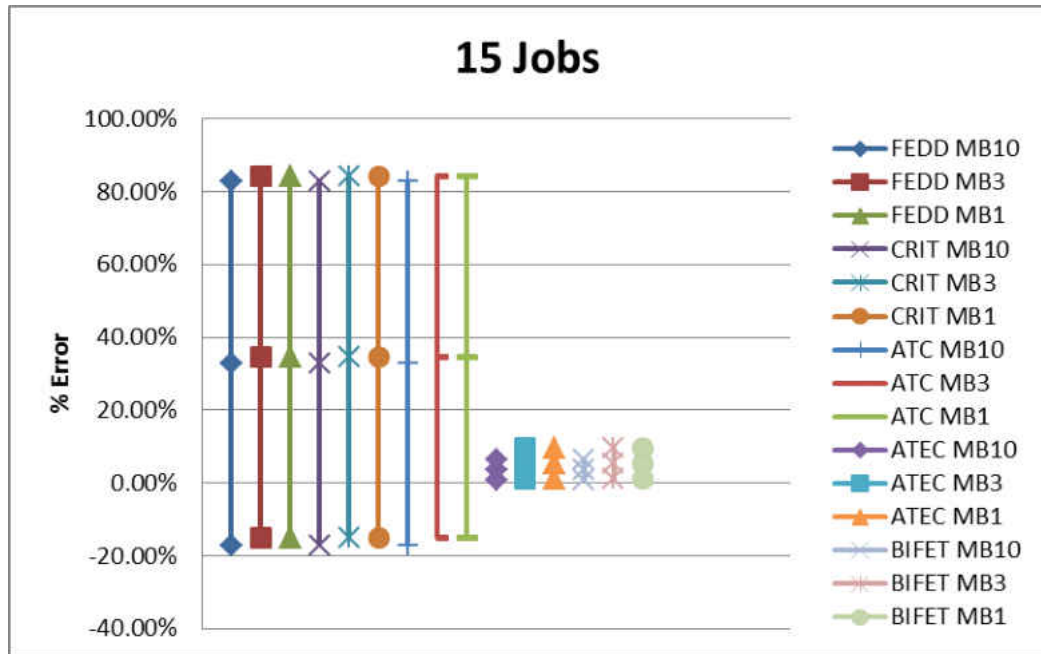


Figure A 6: 95% confidence interval on the performance results for tight due dates, high traffic, 1 family.

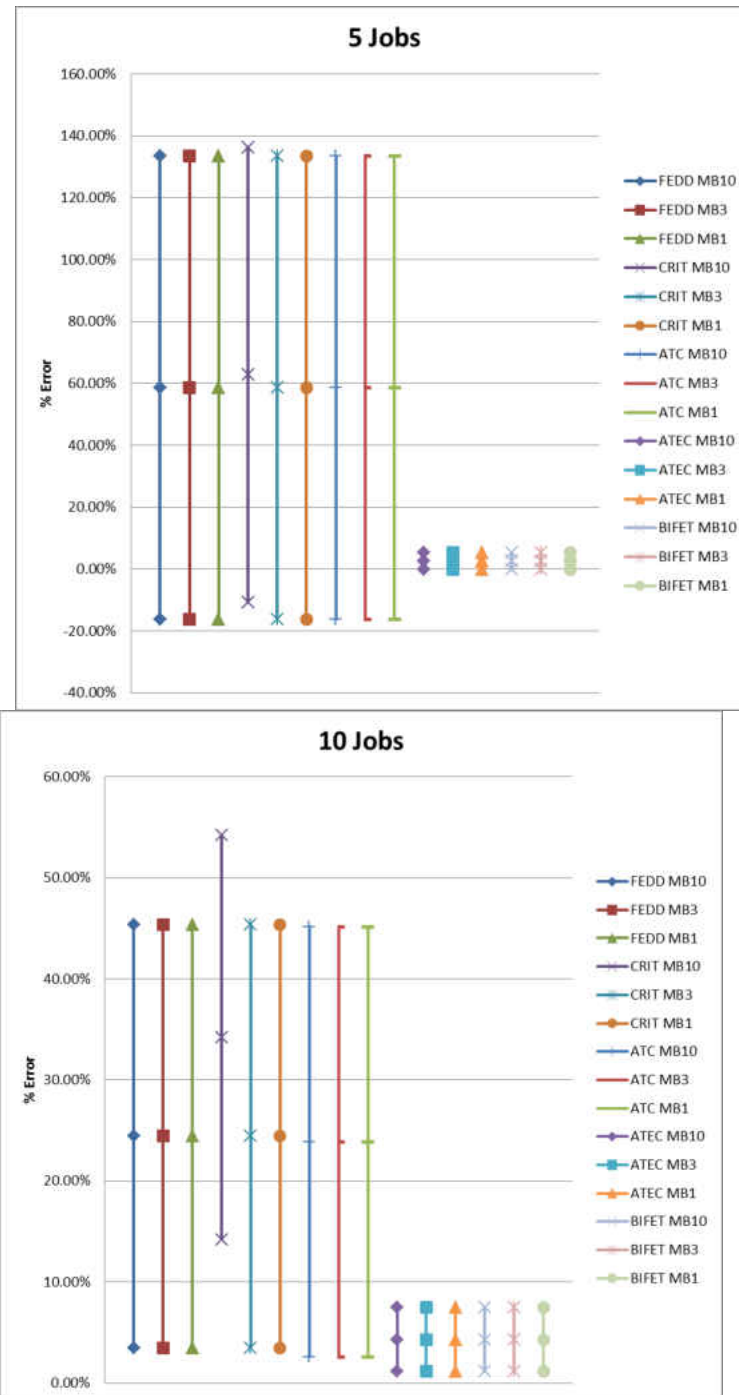


Figure A 7: 95% confidence interval on the performance results for tight due dates, high traffic, 5 families.

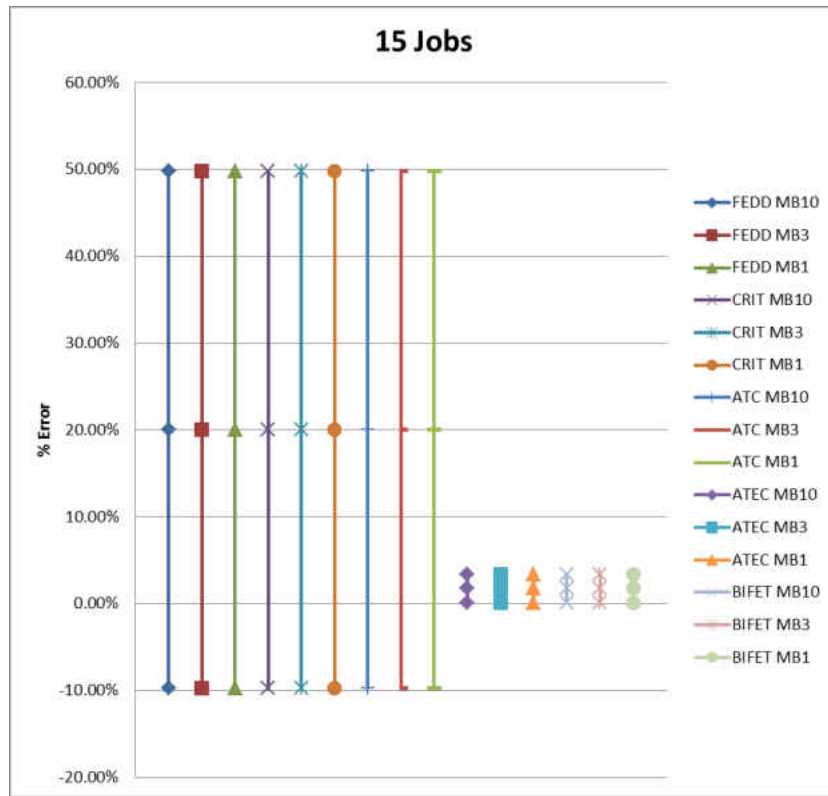


Figure A 8: 95% confidence interval on the performance results for tight due dates, high traffic, 5 families.

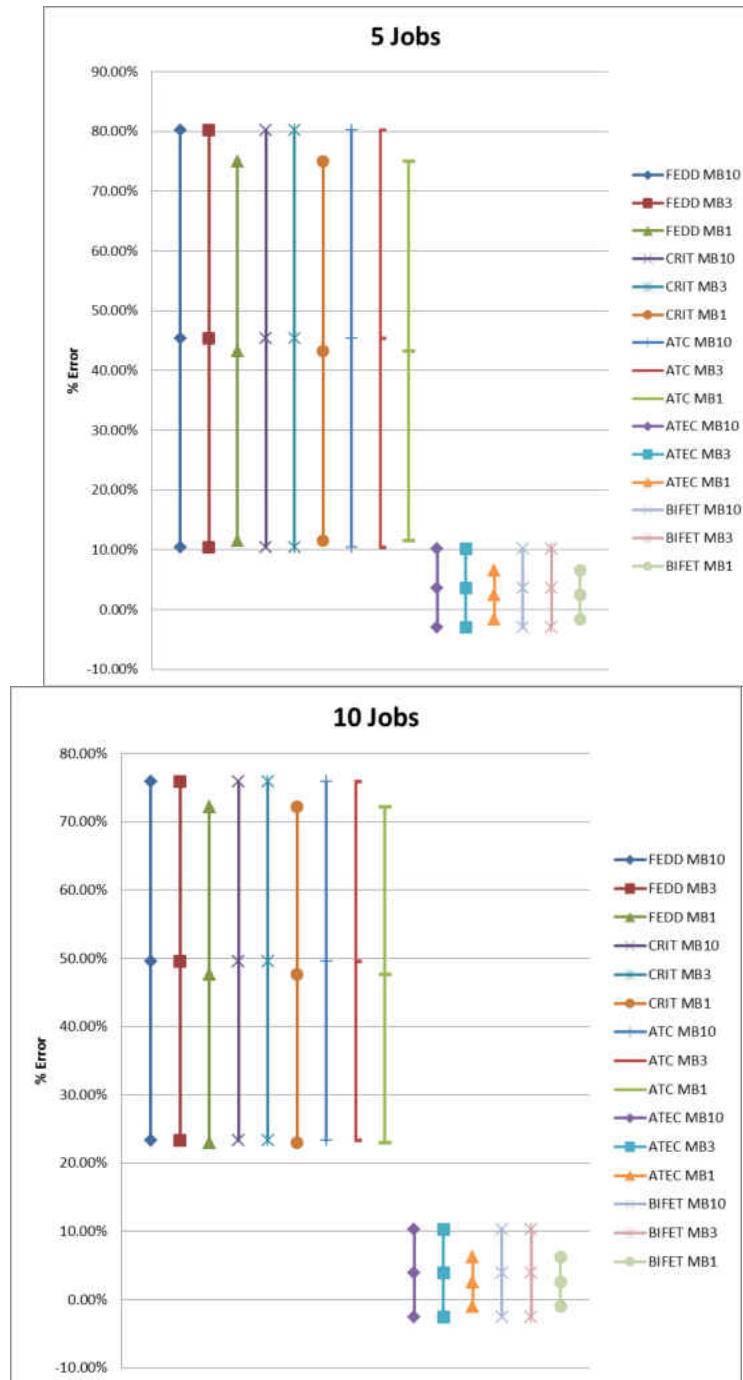


Figure A 9: 95% confidence interval on the performance results tight due dates, medium traffic, 1 family.

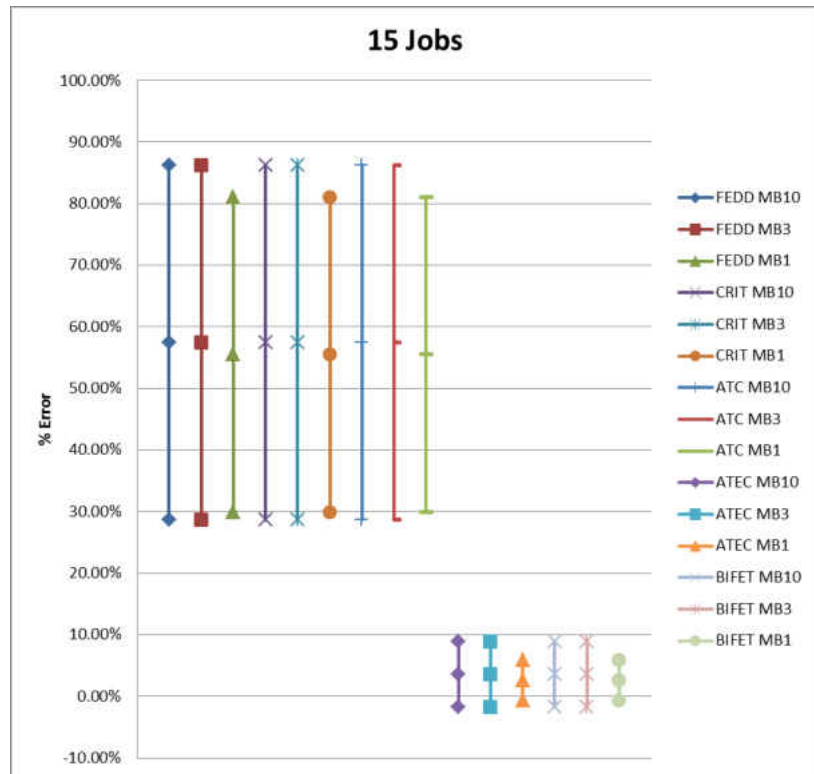


Figure A 10. 95% confidence interval on the performance results for tight due dates, medium traffic intensity, 1 family.

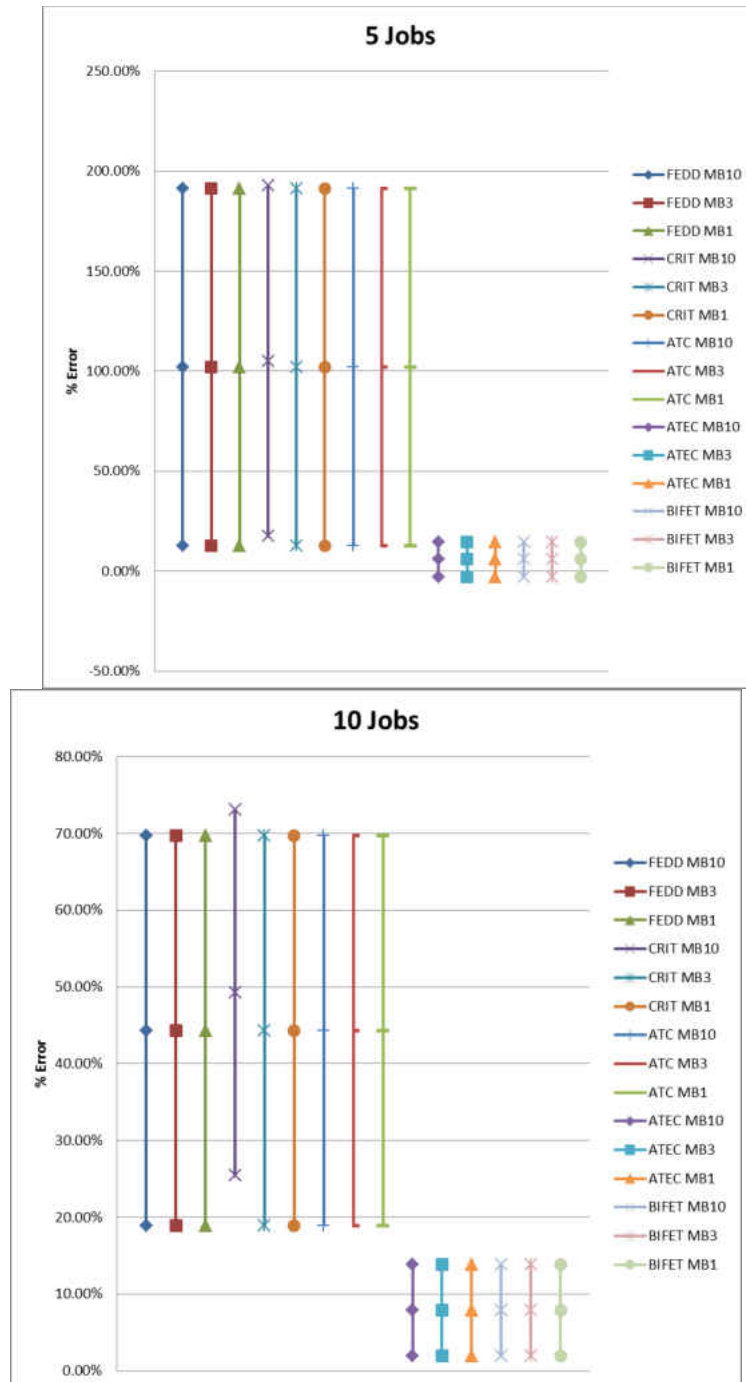


Figure A 11: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

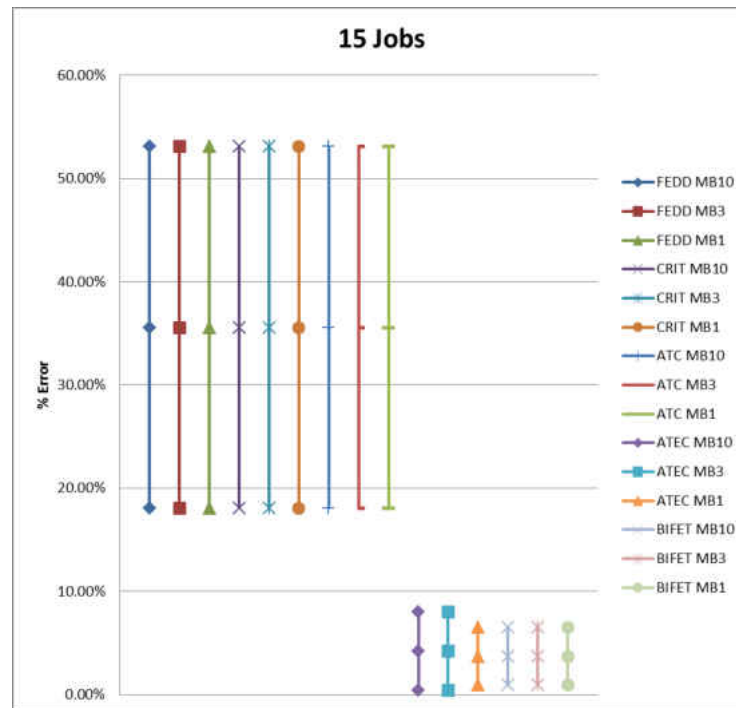


Figure A 12: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

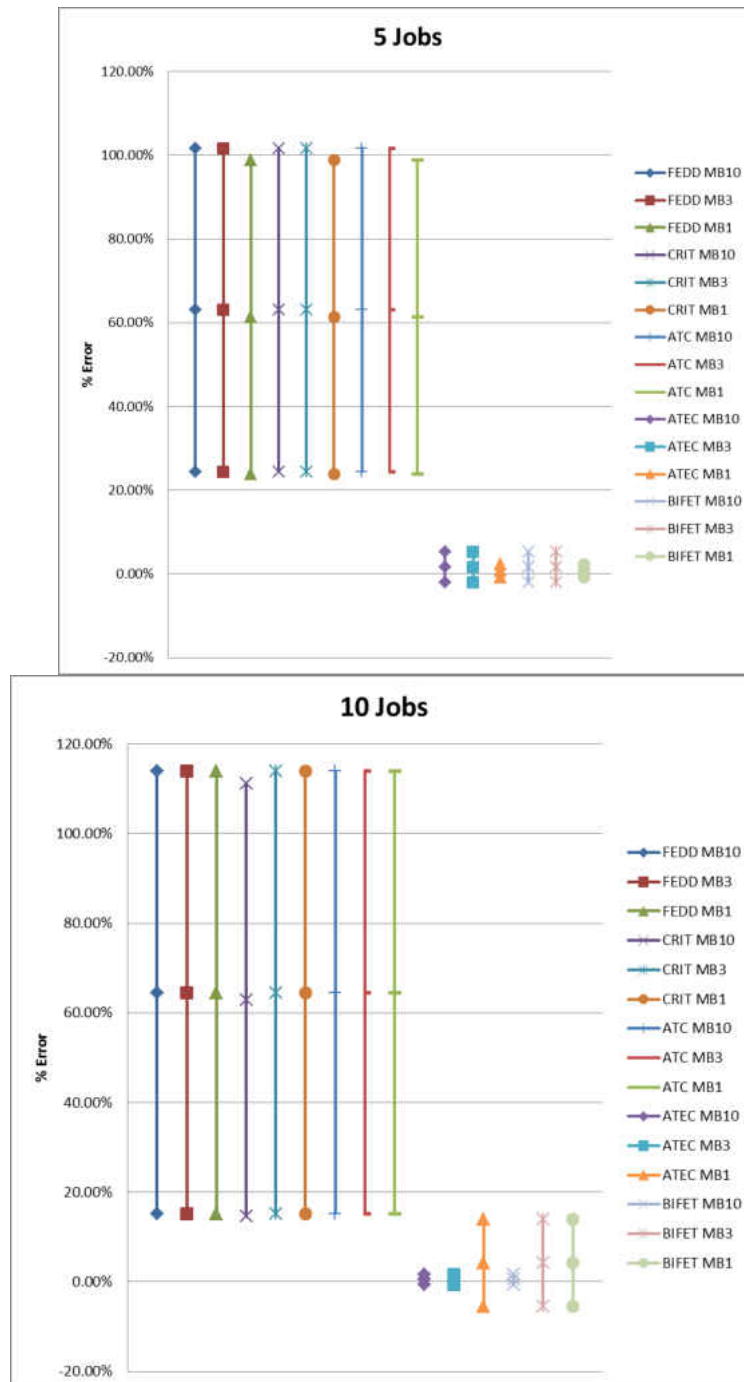


Figure A 13: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

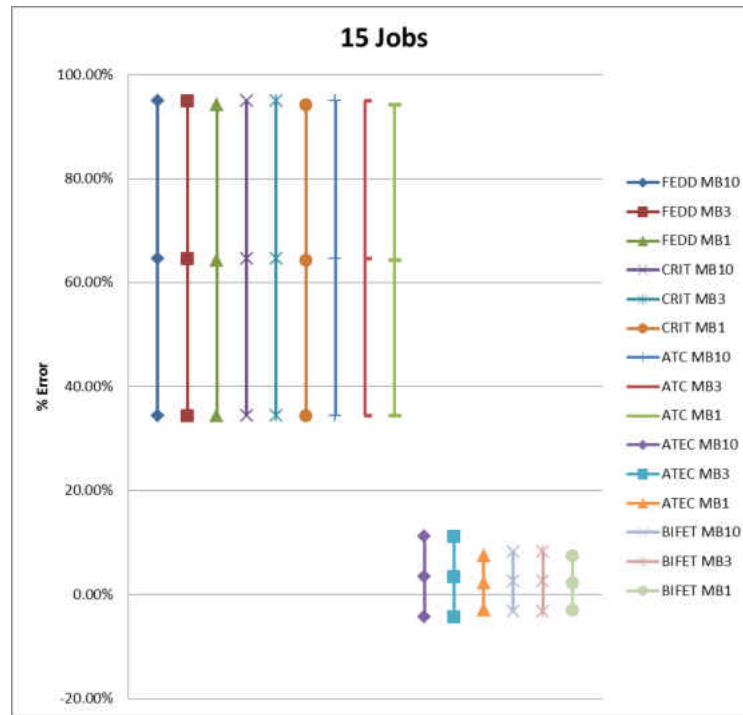


Figure A 14: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

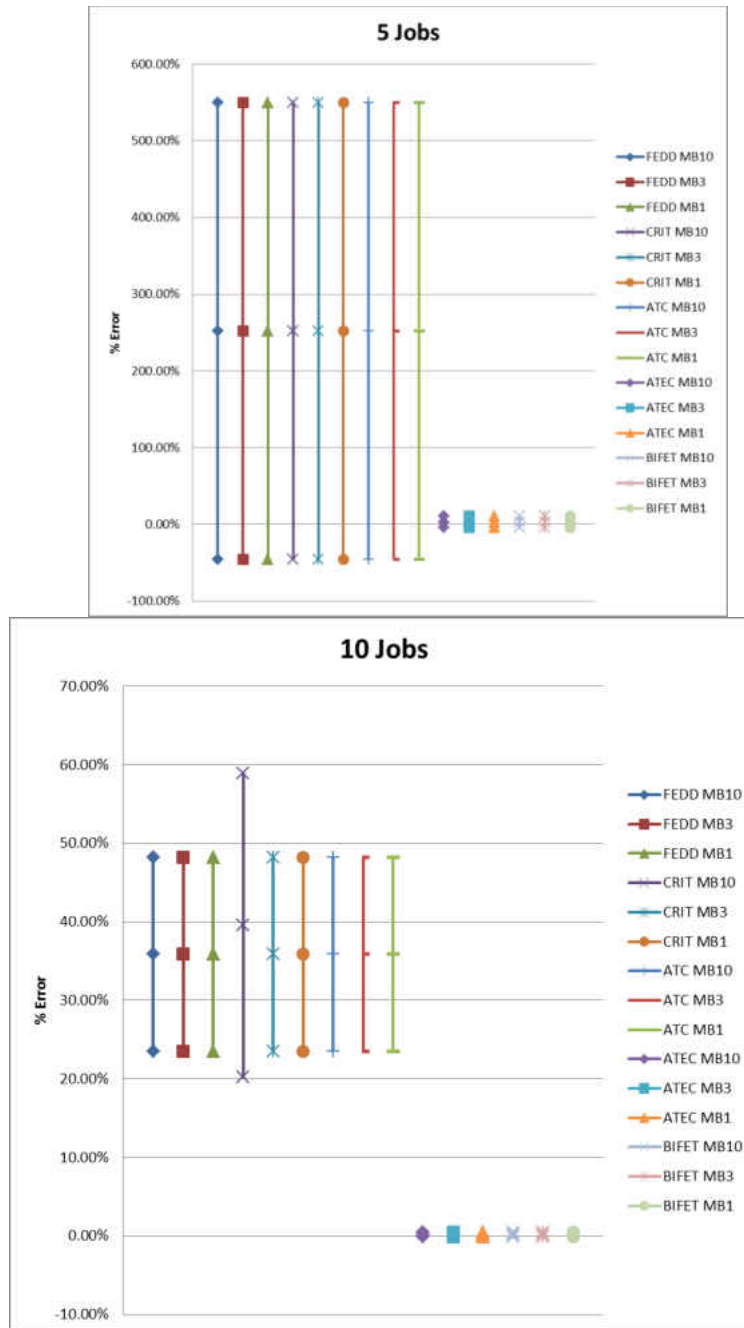


Figure A 15: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families.

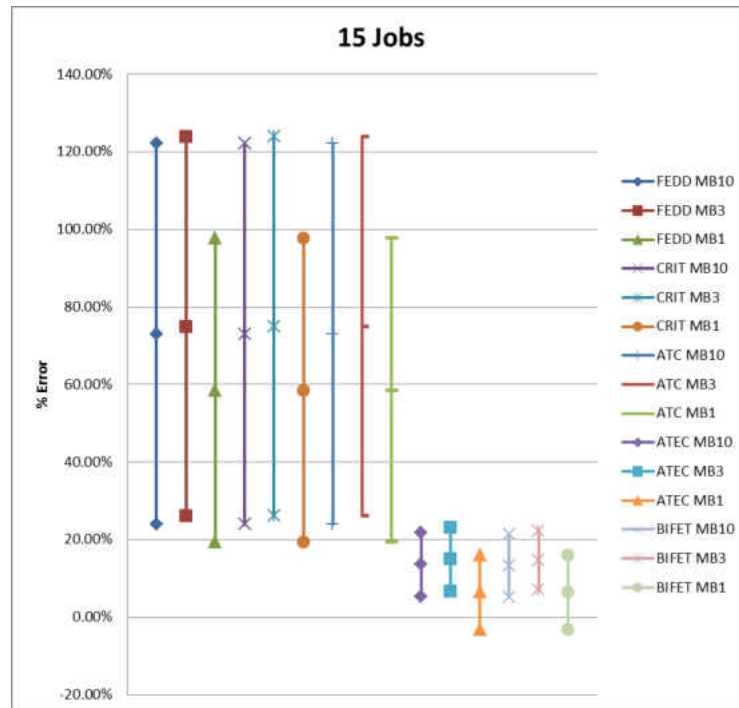


Figure A 16: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families

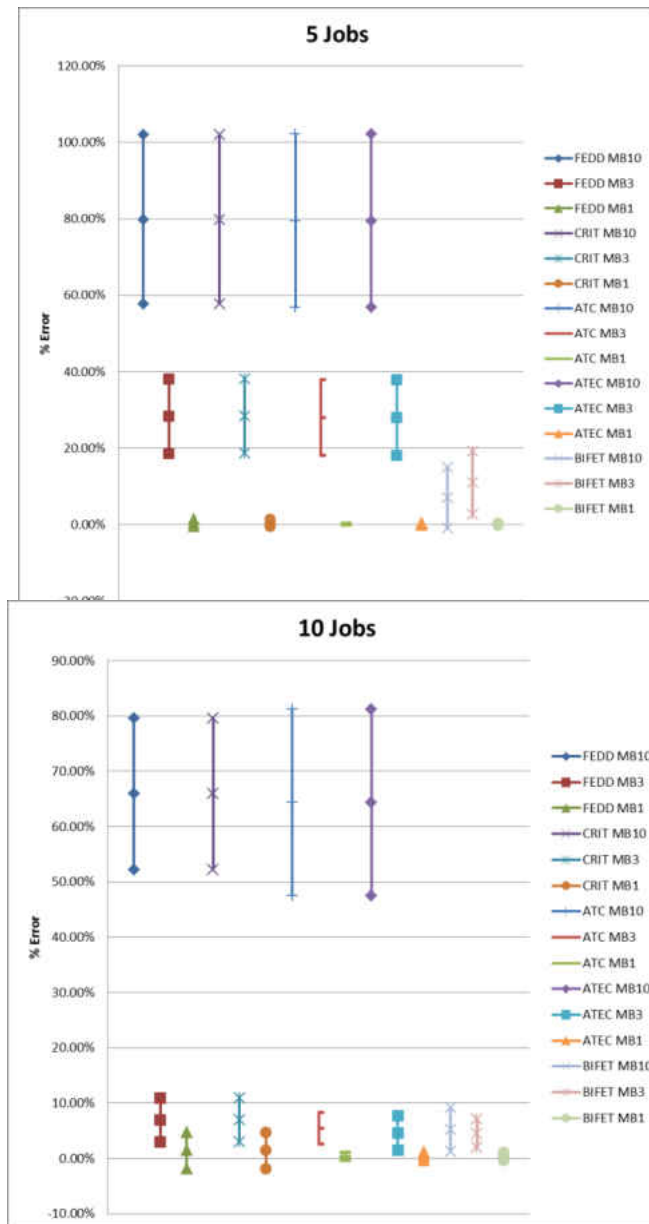


Figure A 17: 95% confidence interval on the performance results for loose due dates, jobs arrive simultaneously, 1 family.

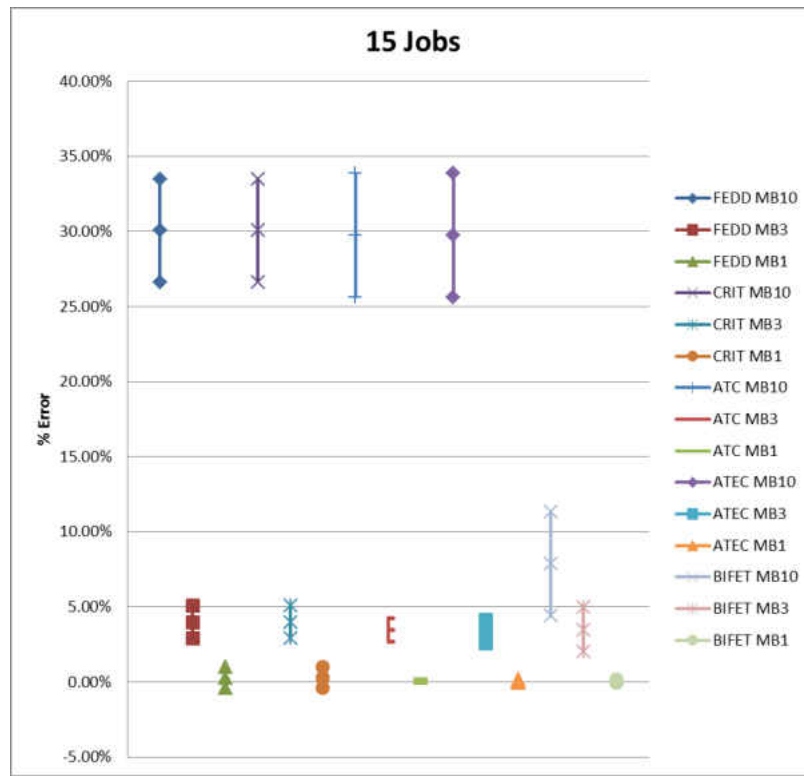


Figure A 18: 95% confidence interval on the performance results for loose due dates, jobs arrive simultaneously, 1 family.

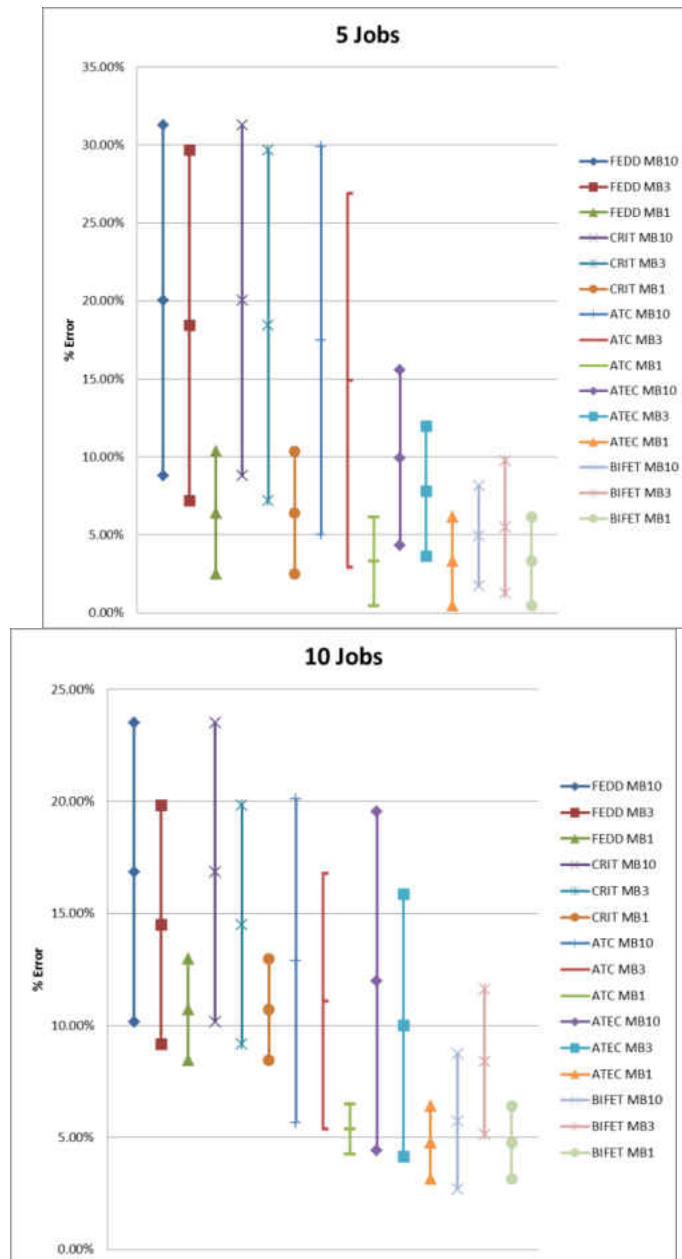


Figure A 19: 95% confidence interval on the performance results for loose due dates, jobs arrive simultaneously, 5 families.

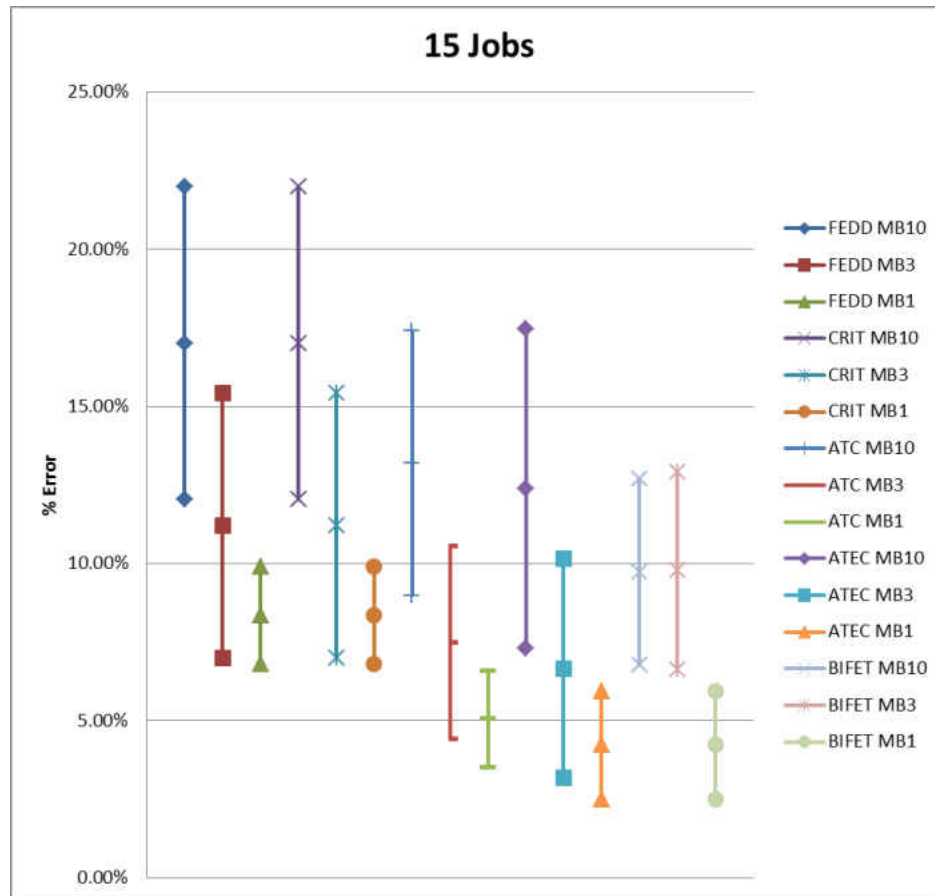


Figure A 20: 95% confidence interval on the performance results for loose due dates, jobs arrive simultaneously, 5 families.

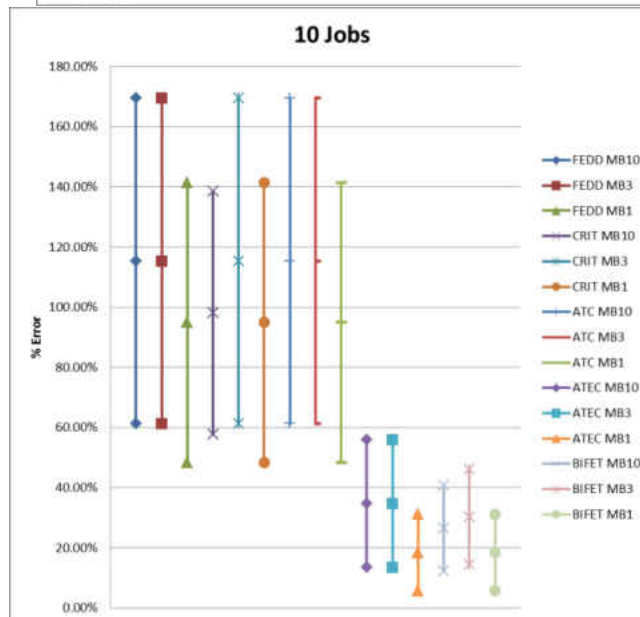
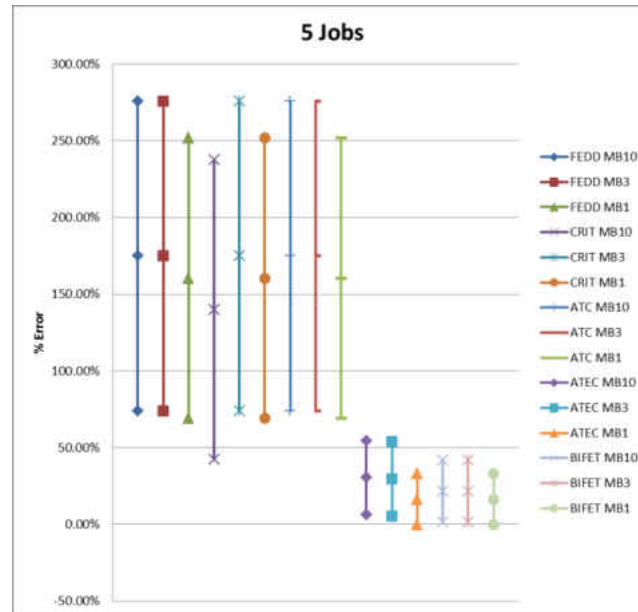


Figure A 21: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

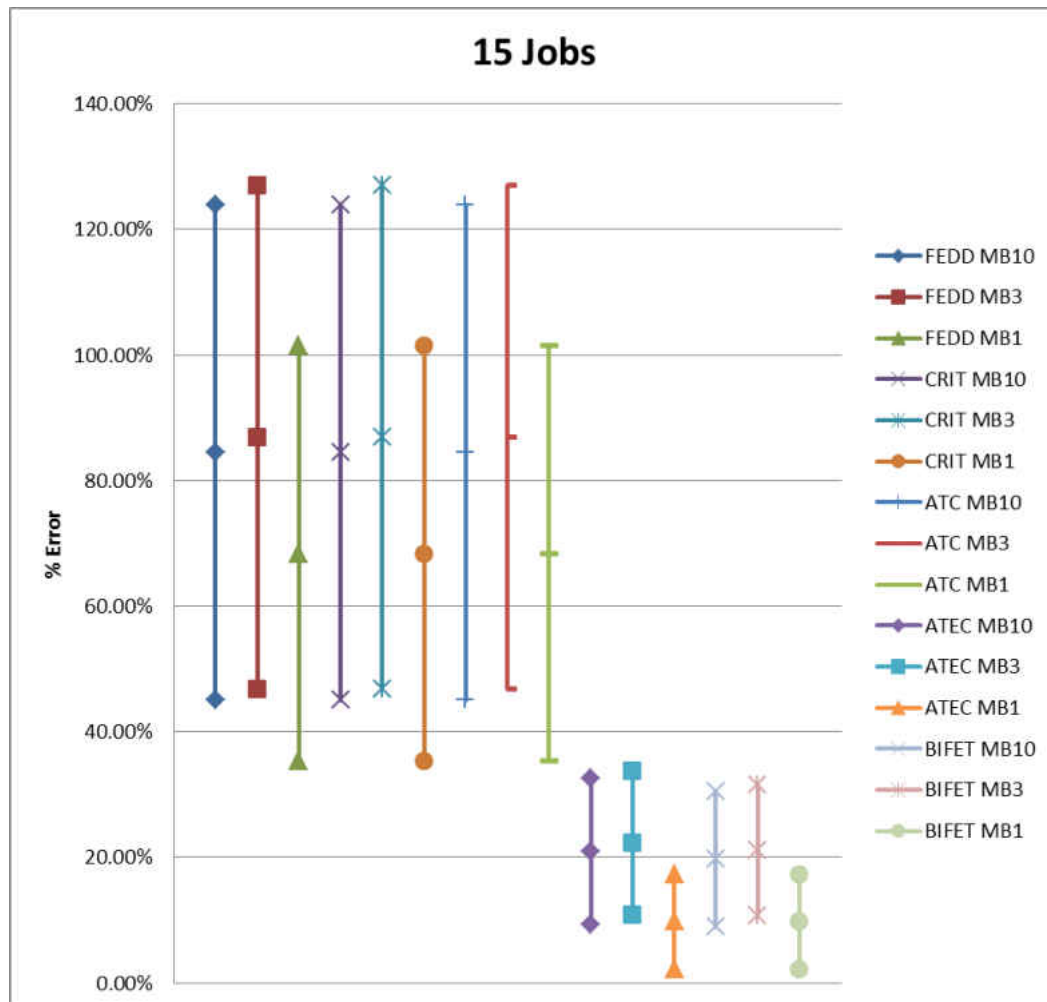


Figure A 22: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

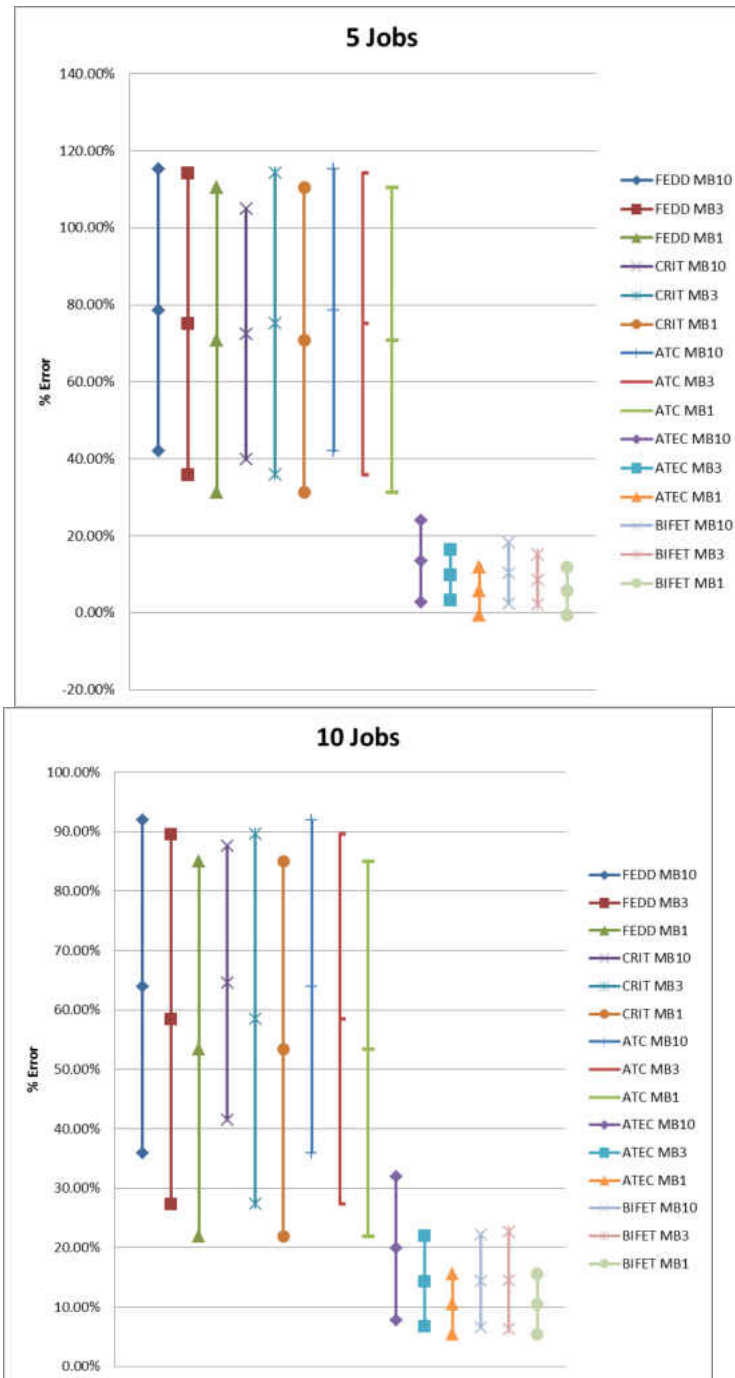


Figure A 23: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

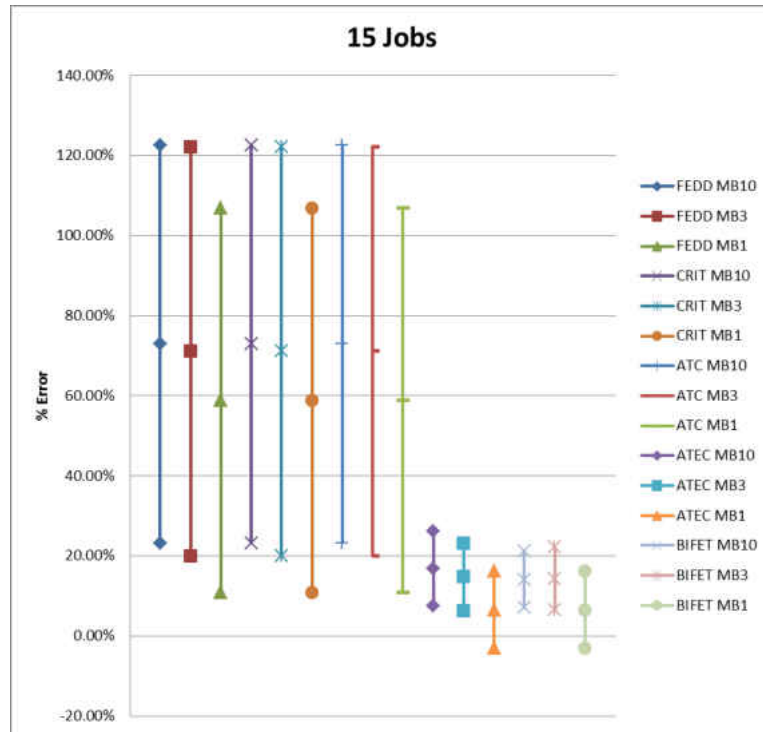


Figure A 24: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

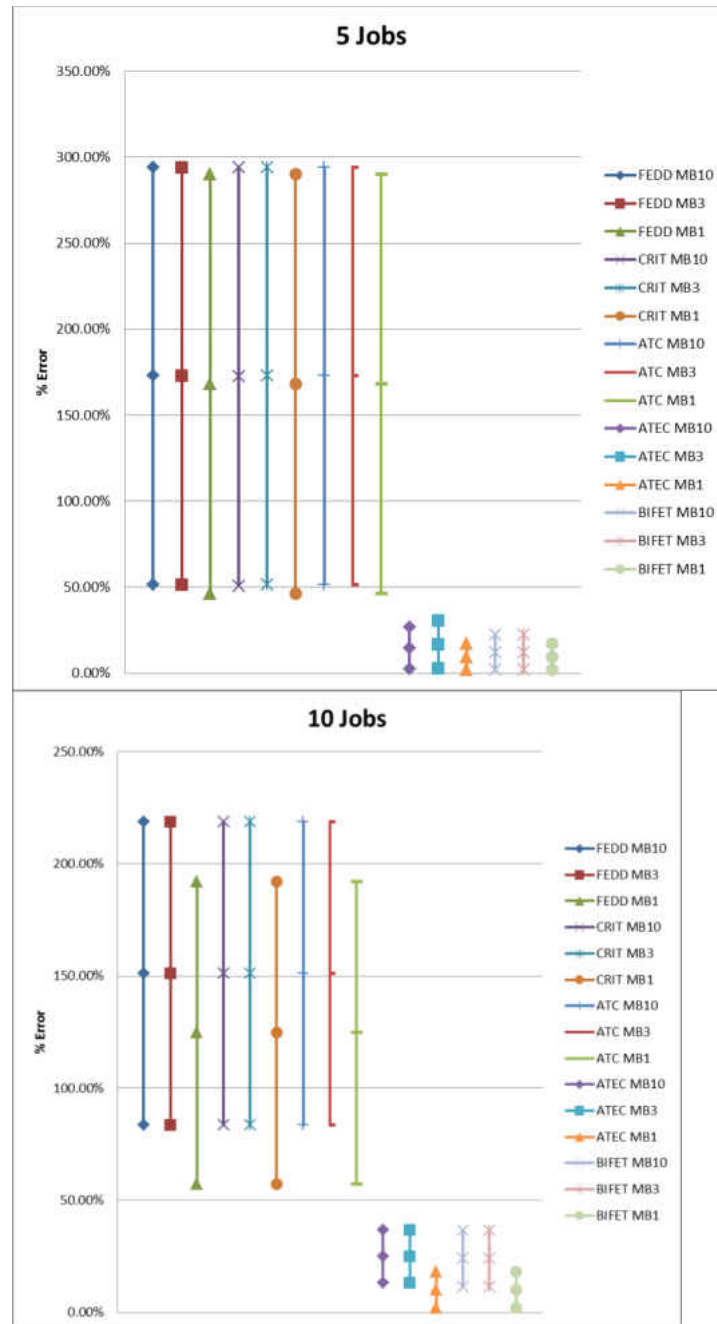


Figure A 25: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

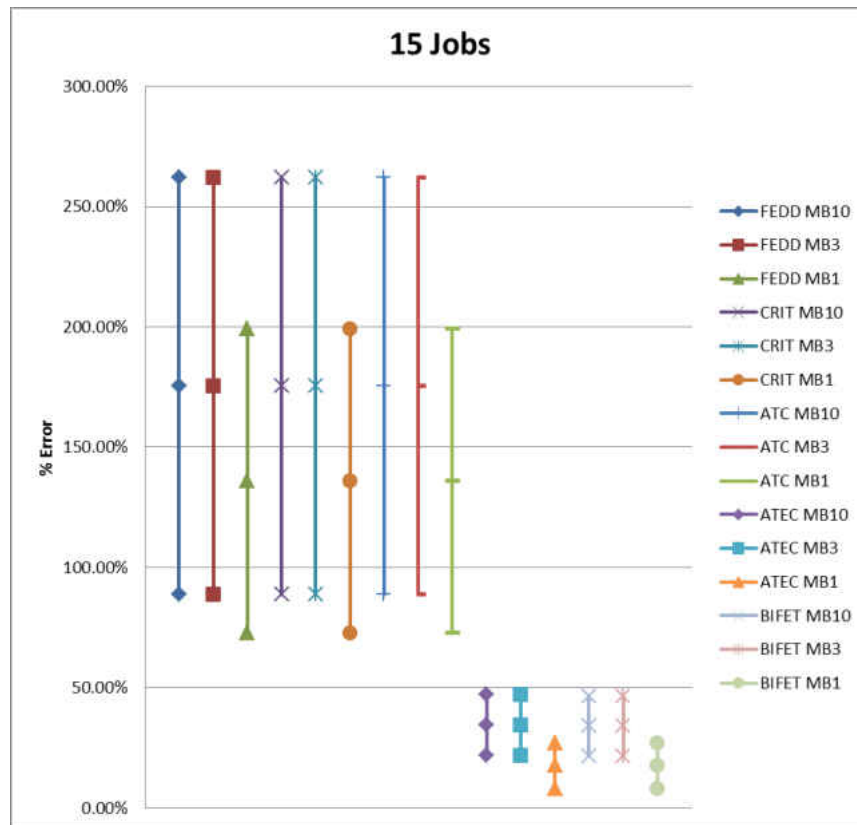


Figure A 26: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

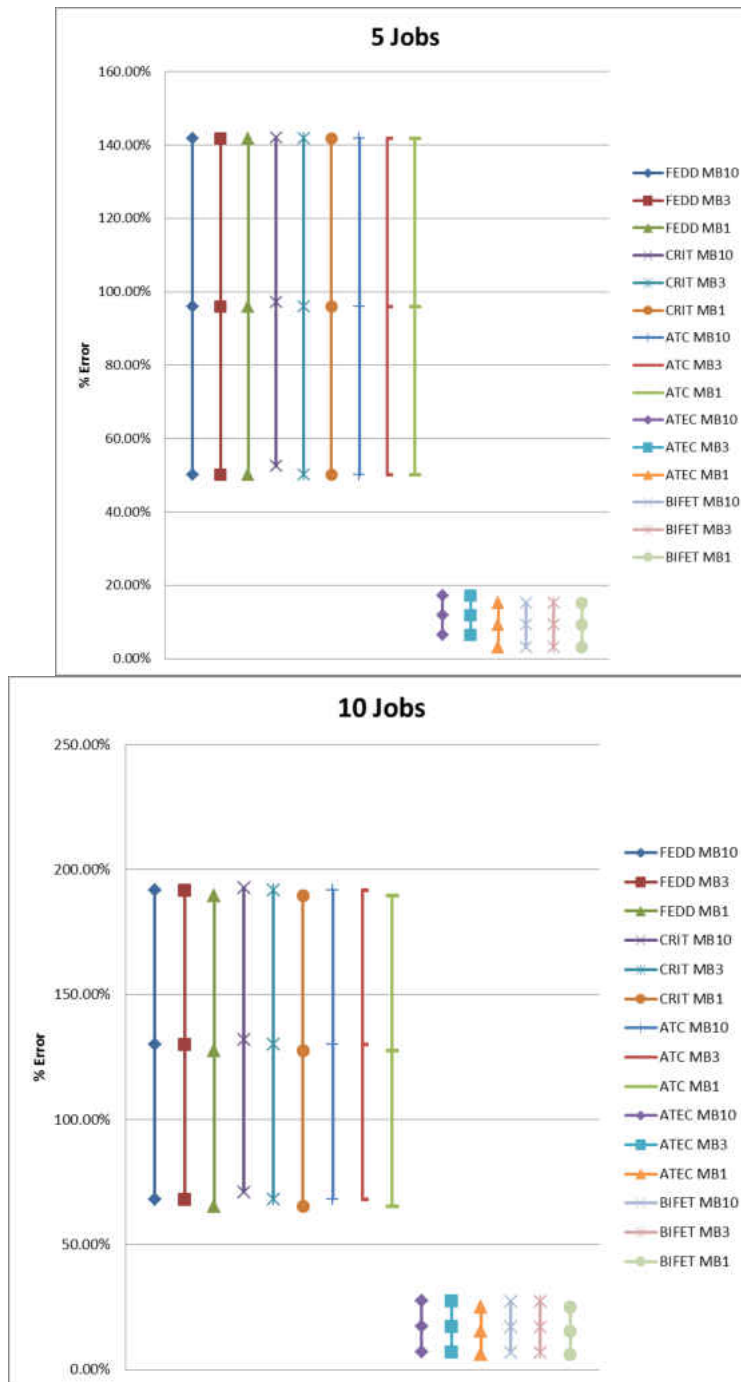


Figure A 27: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families.

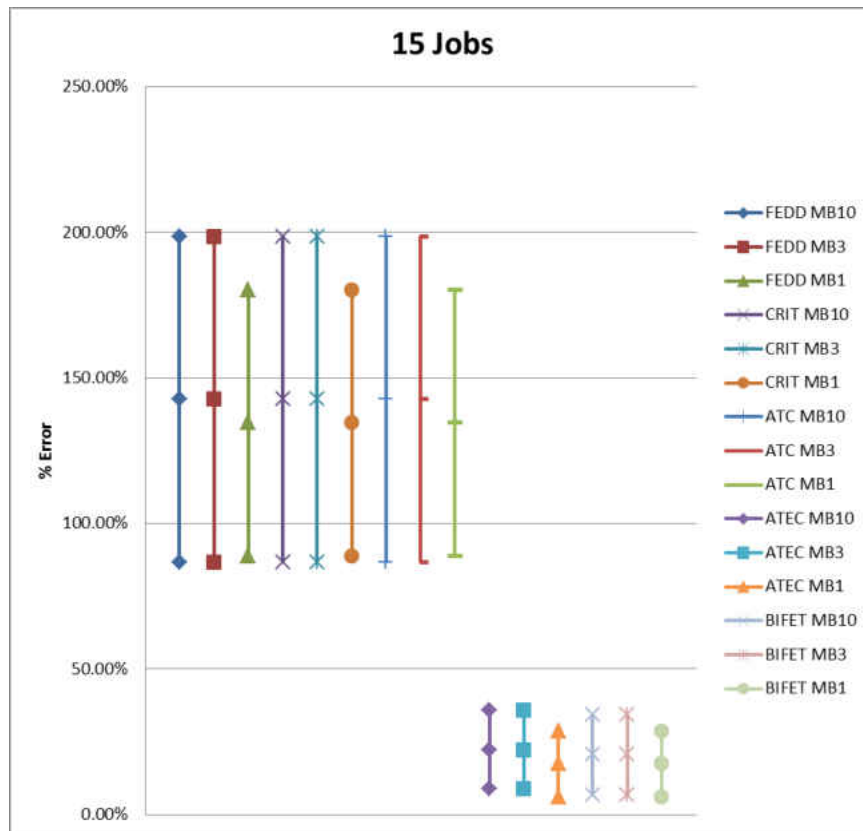


Figure A 28: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families.

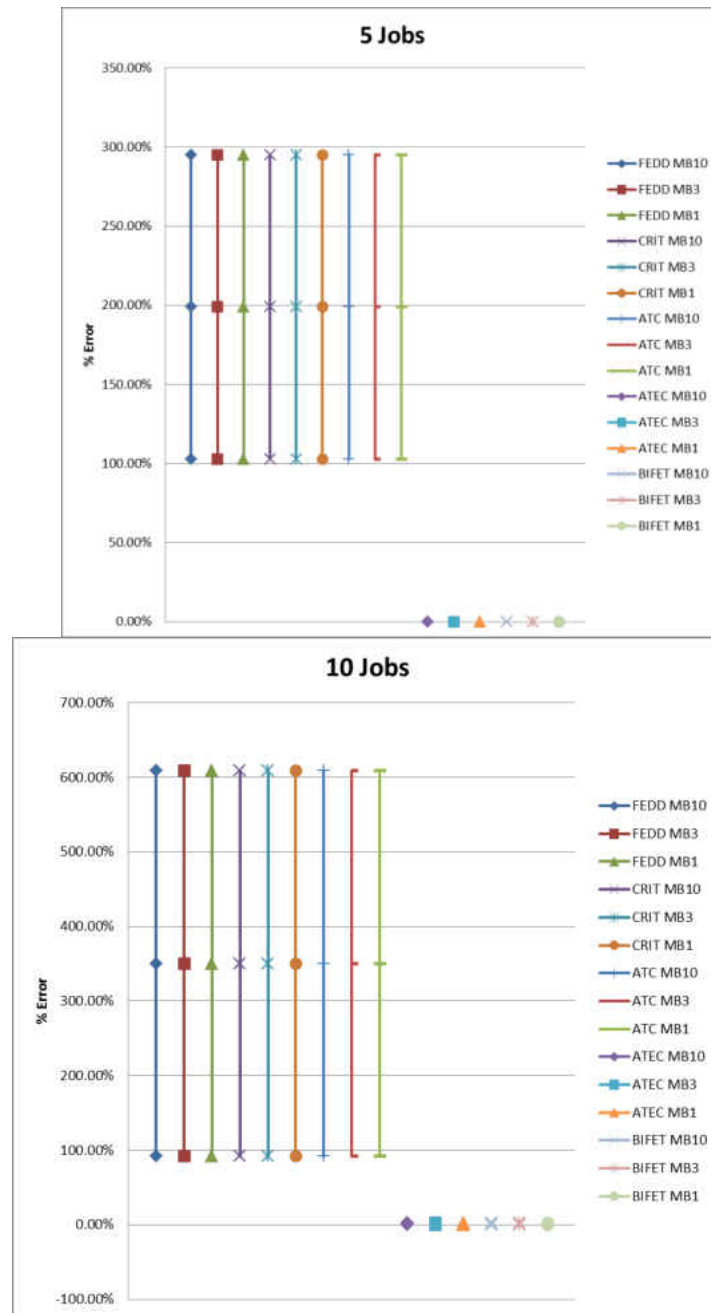


Figure A 29: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

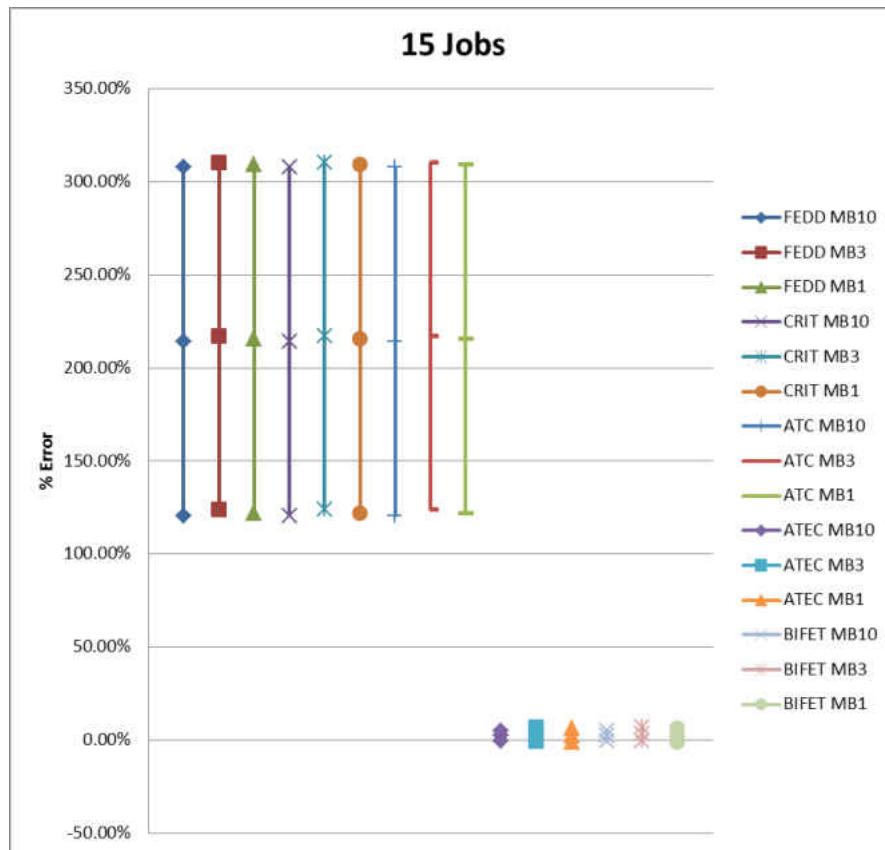


Figure A 30: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

APPENDIX B: EARLY PENALTY =10/TARDY PENALTY =1

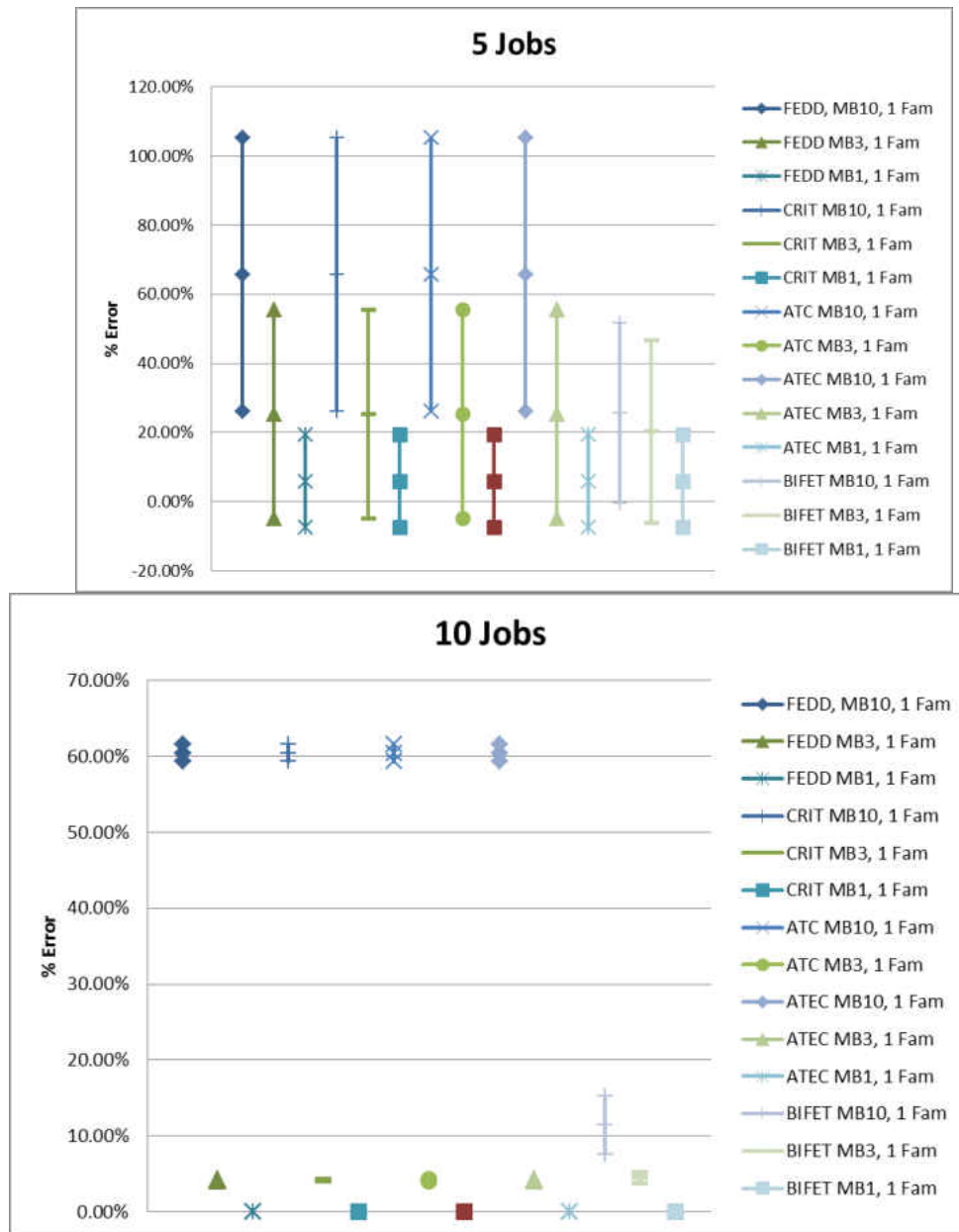


Figure B 1: 95% confidence interval on the performance results for jobs arrive simultaneously, tight due dates, 1 family.

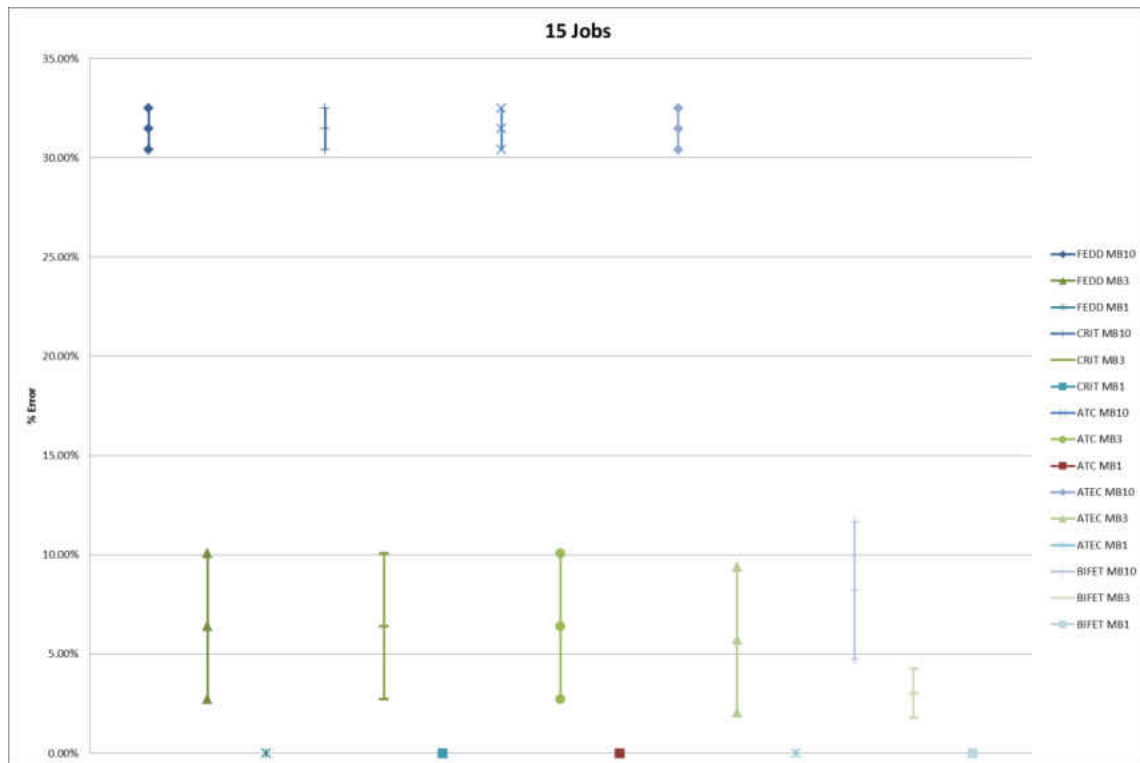


Figure B 2: 95% confidence interval on the performance results for jobs arrive simultaneously, tight due dates, 1 family.

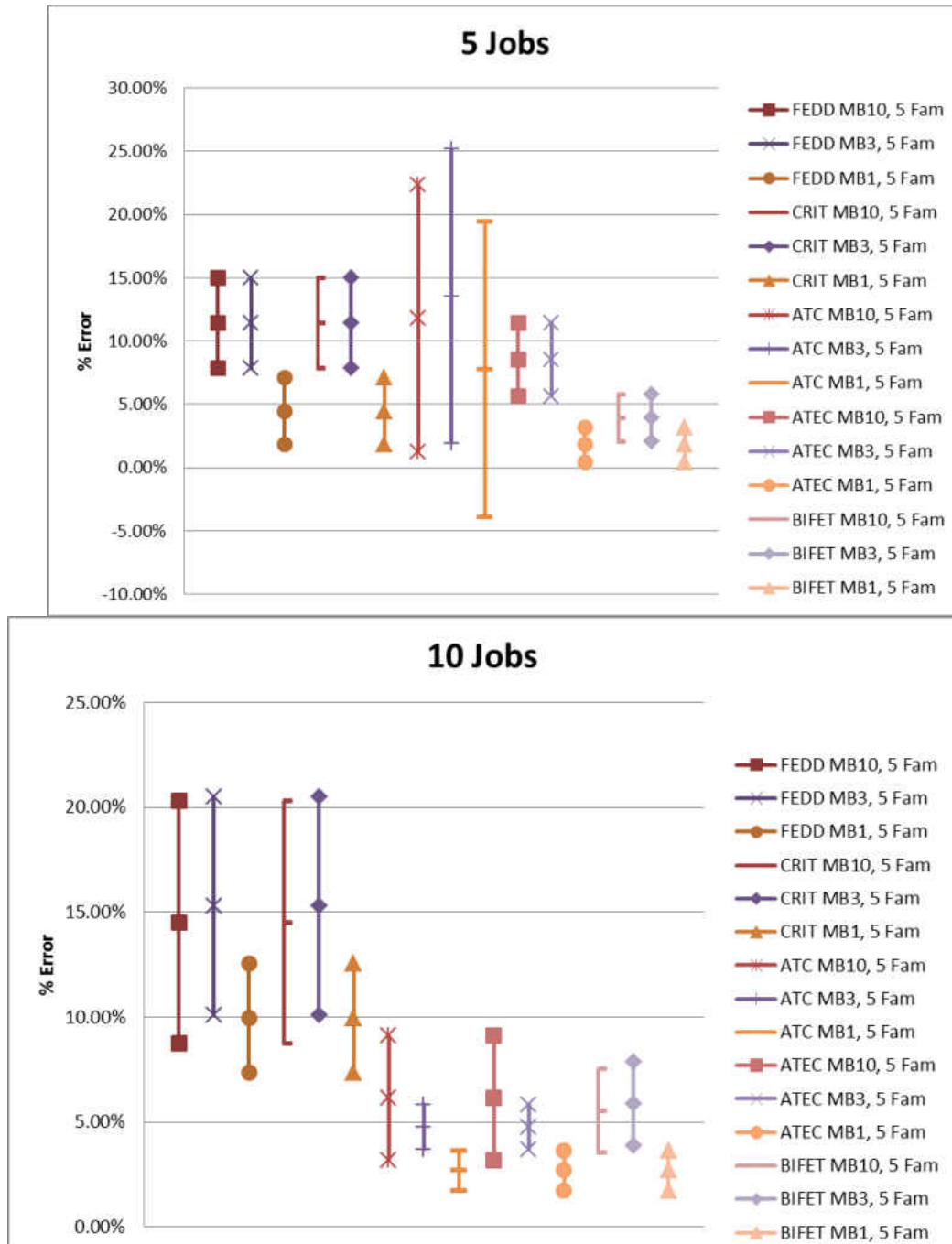


Figure B 3: 95% confidence interval on the performance results for jobs arrive simultaneously, tight due dates, 5 families.

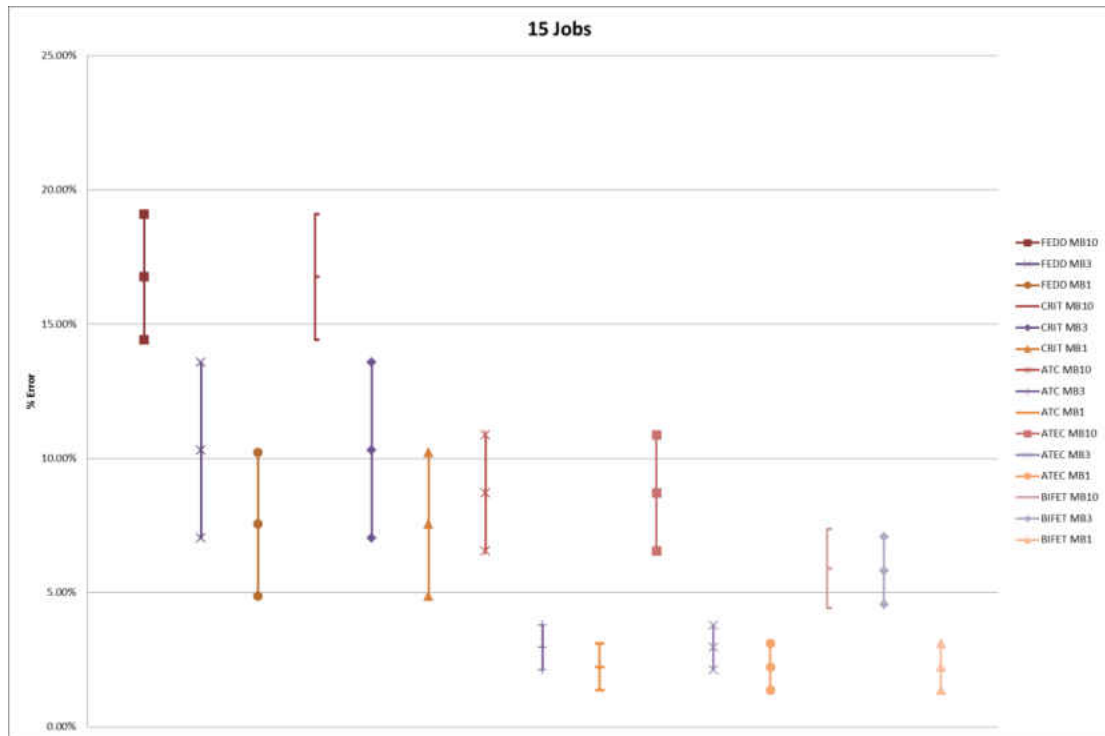


Figure B 4: 95% confidence interval on the performance results for jobs arrive simultaneously, tight due dates, 5 Families.

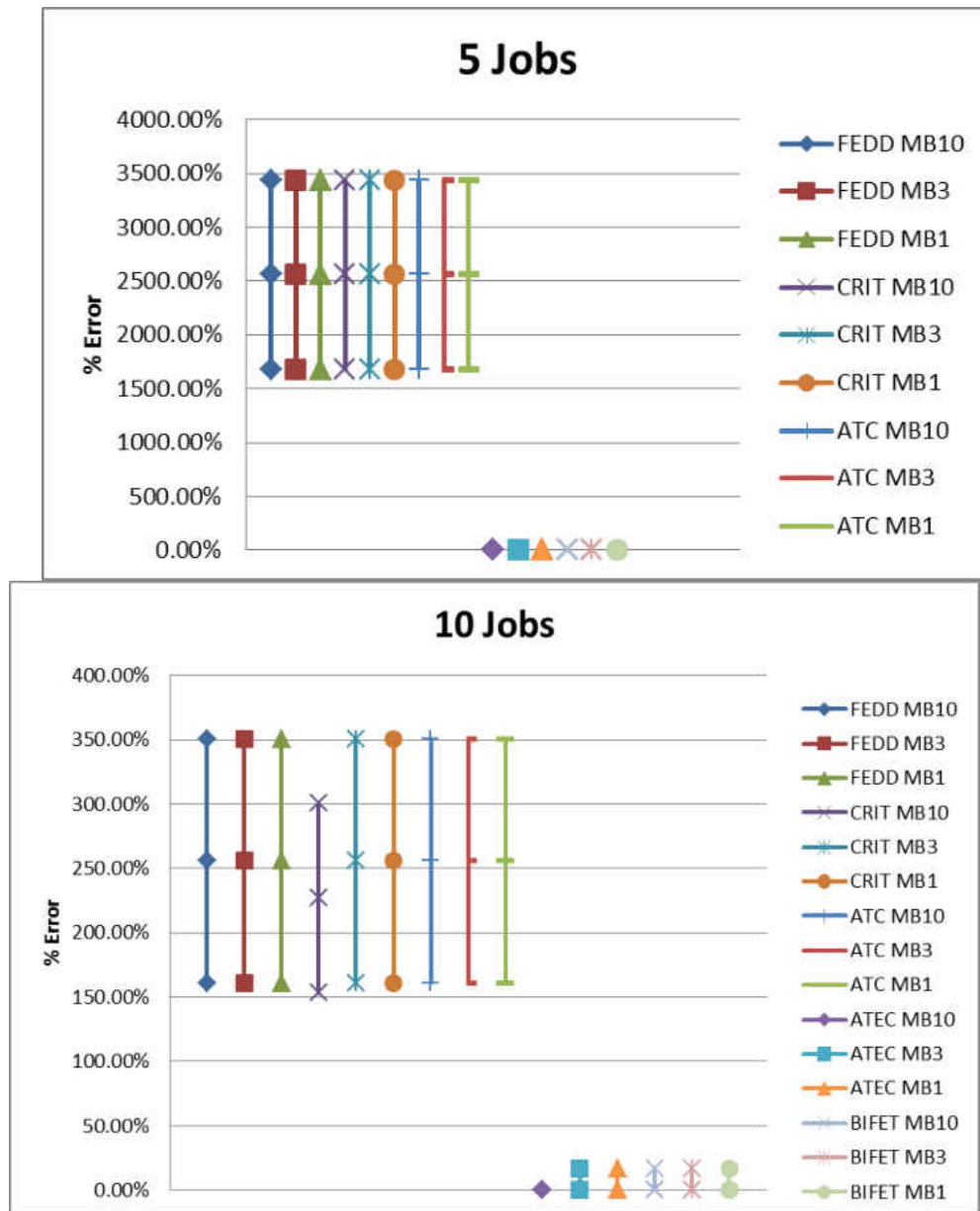


Figure B 5: 95% confidence interval on the performance results for tight due dates, high traffic, 1 family.

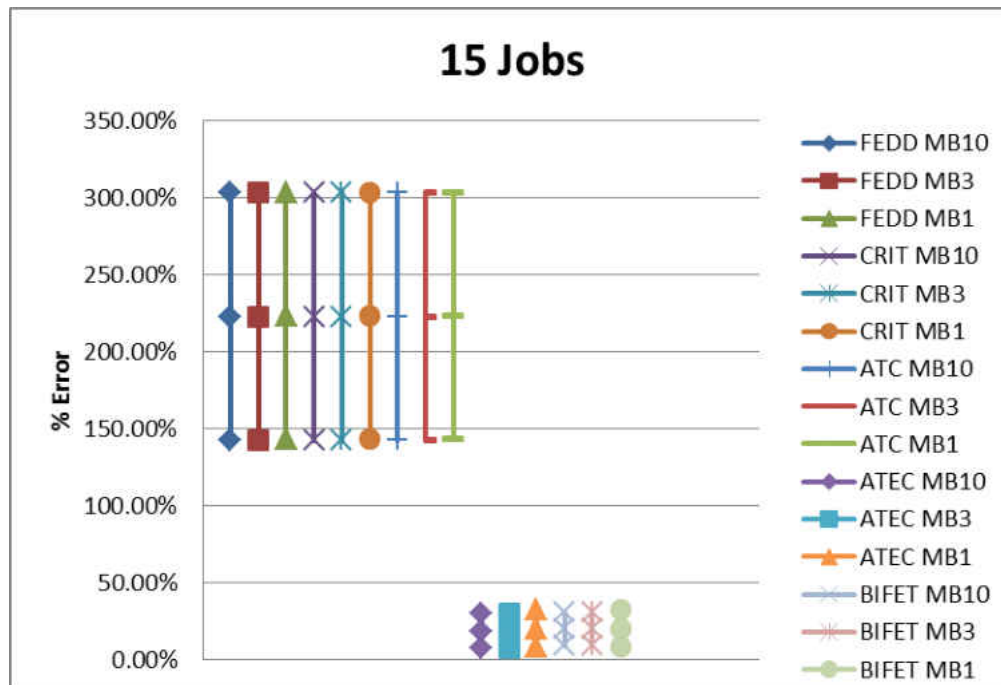


Figure B 6: 95% confidence interval on the performance results for tight due dates, high traffic, 1 family.

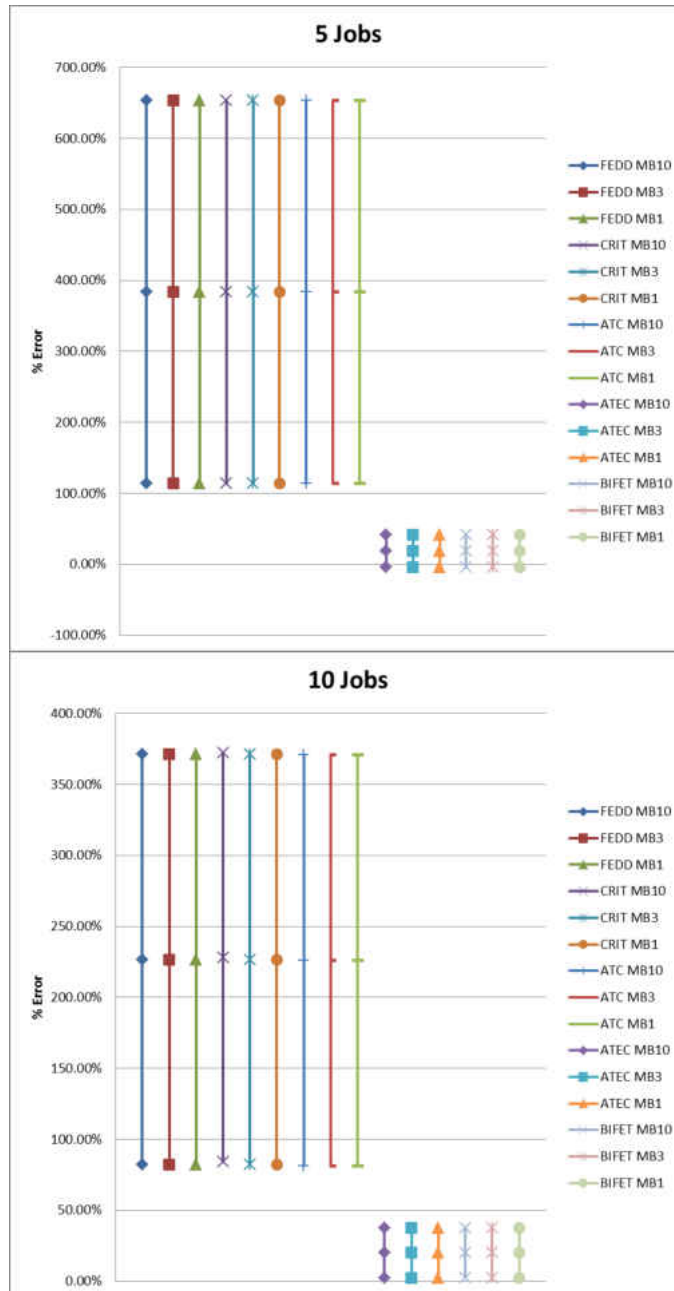


Figure B 7: 95% confidence interval on the performance results for tight due dates, high traffic, 5 families.

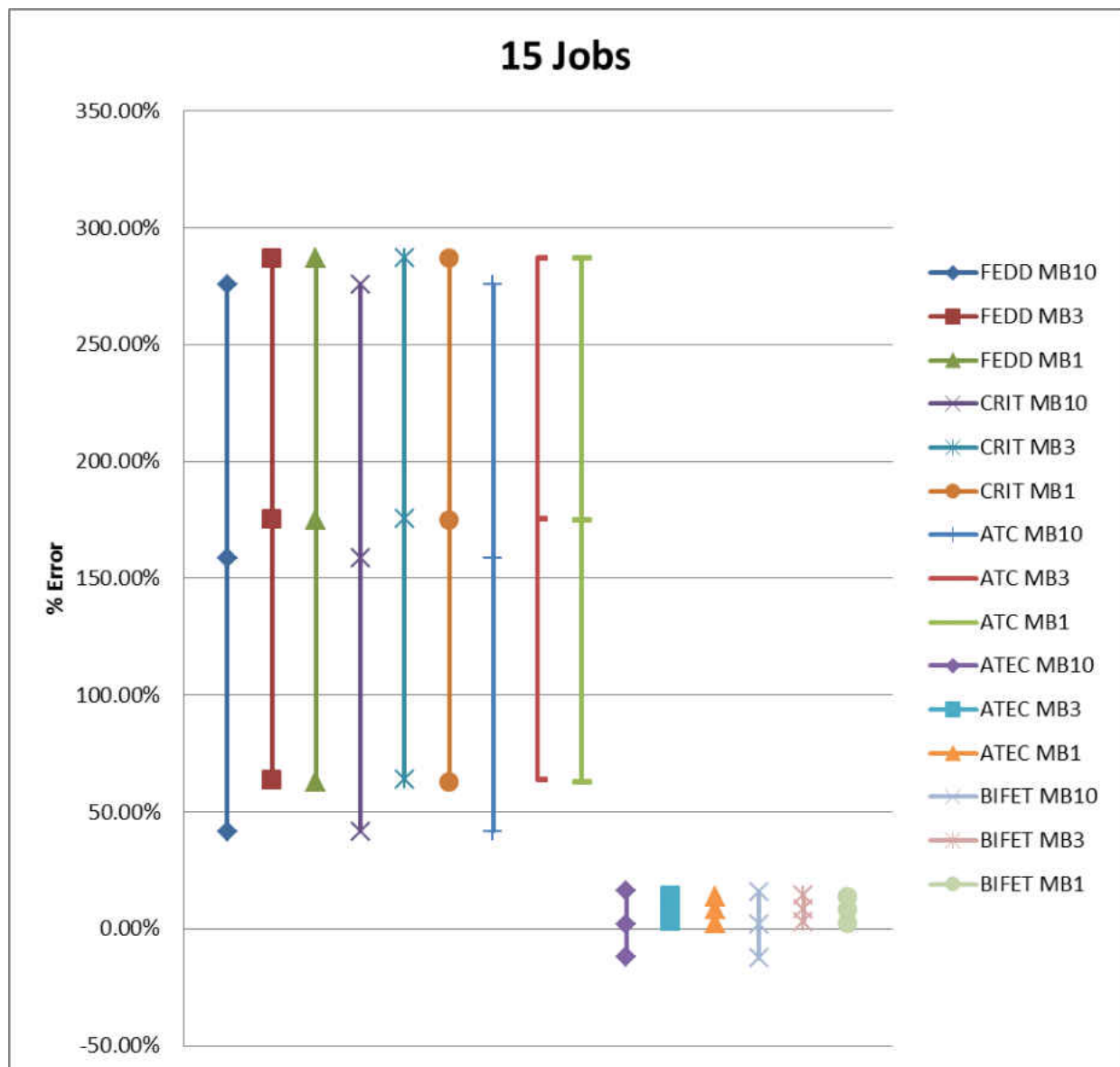


Figure B 8: 95% confidence interval on the performance results for tight due dates, high traffic, 5 families.

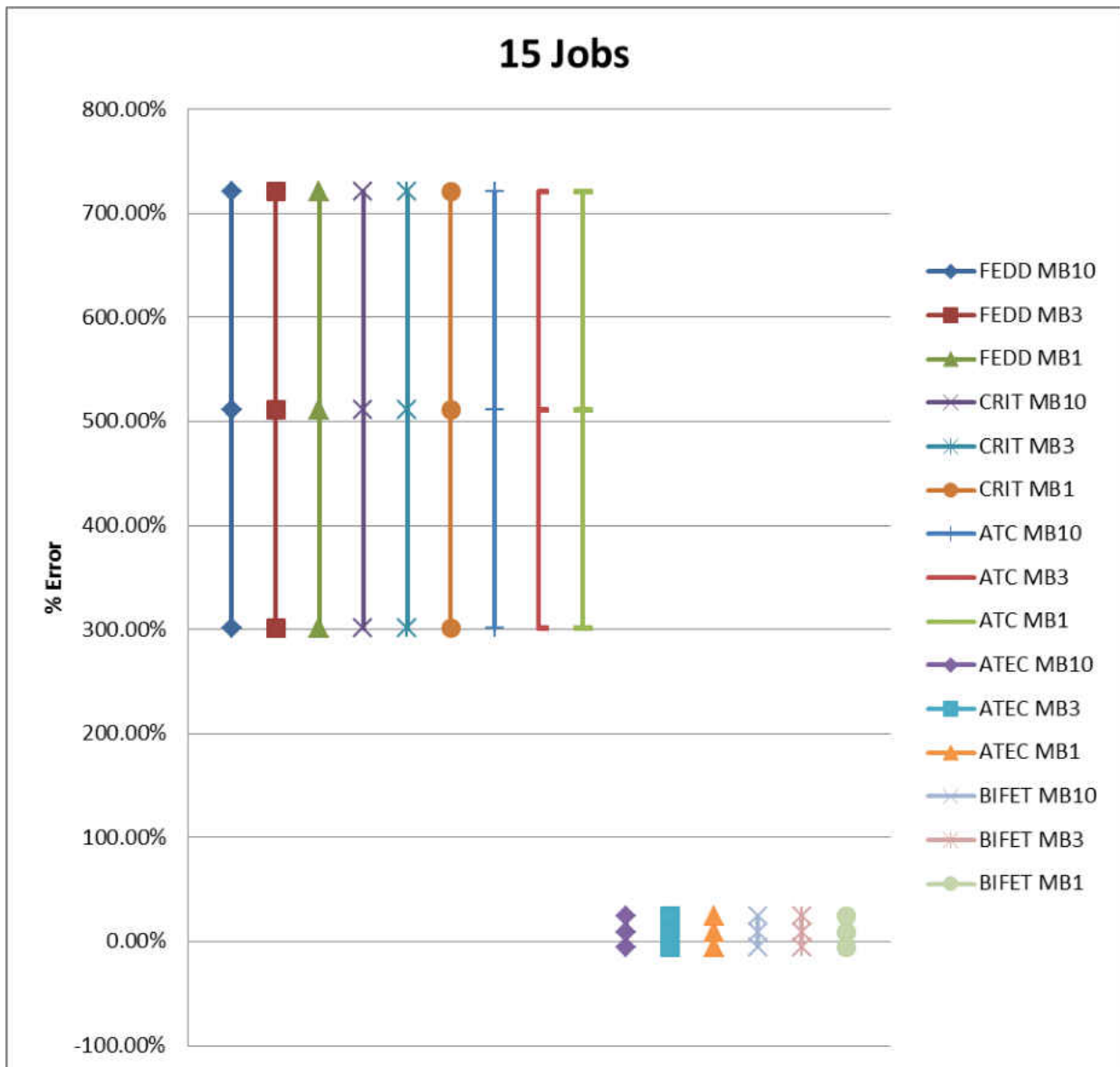


Figure B 10: 95% confidence interval on the performance results for tight due dates, medium traffic, 1 family.

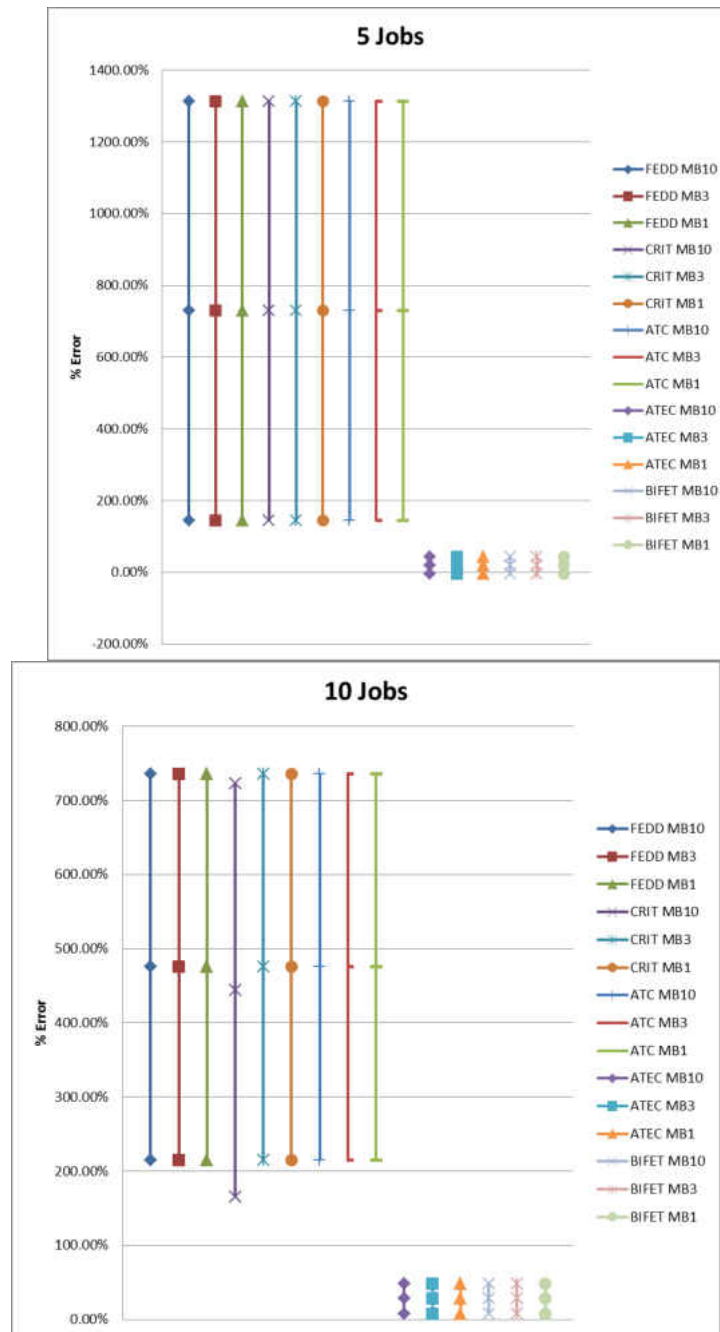


Figure B 11: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

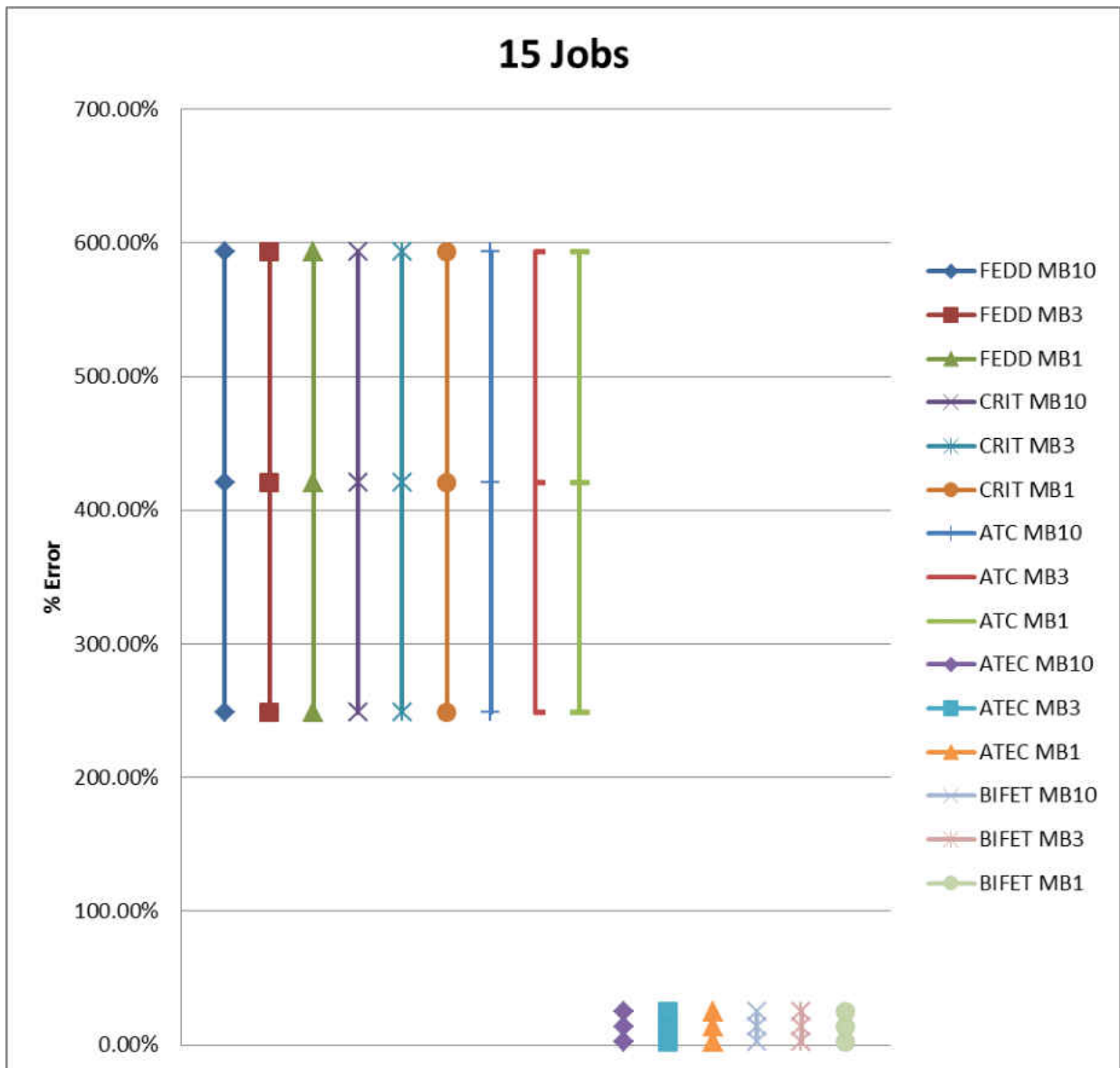


Figure B 12: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

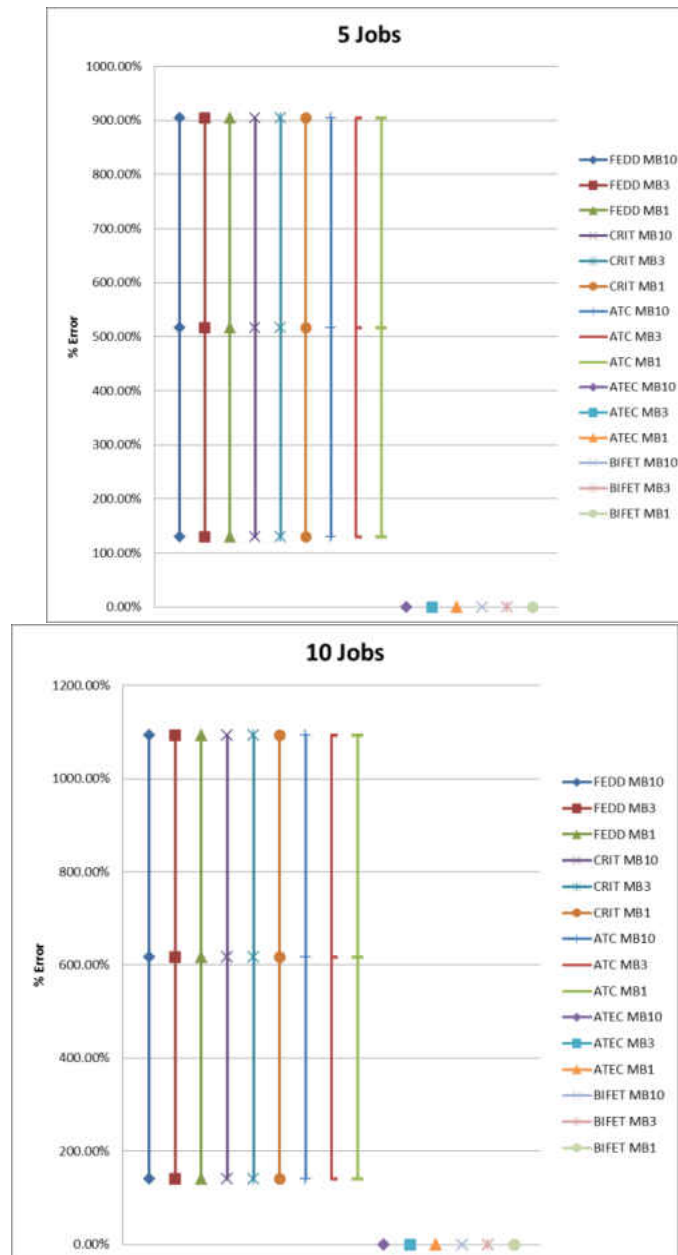


Figure B 13: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

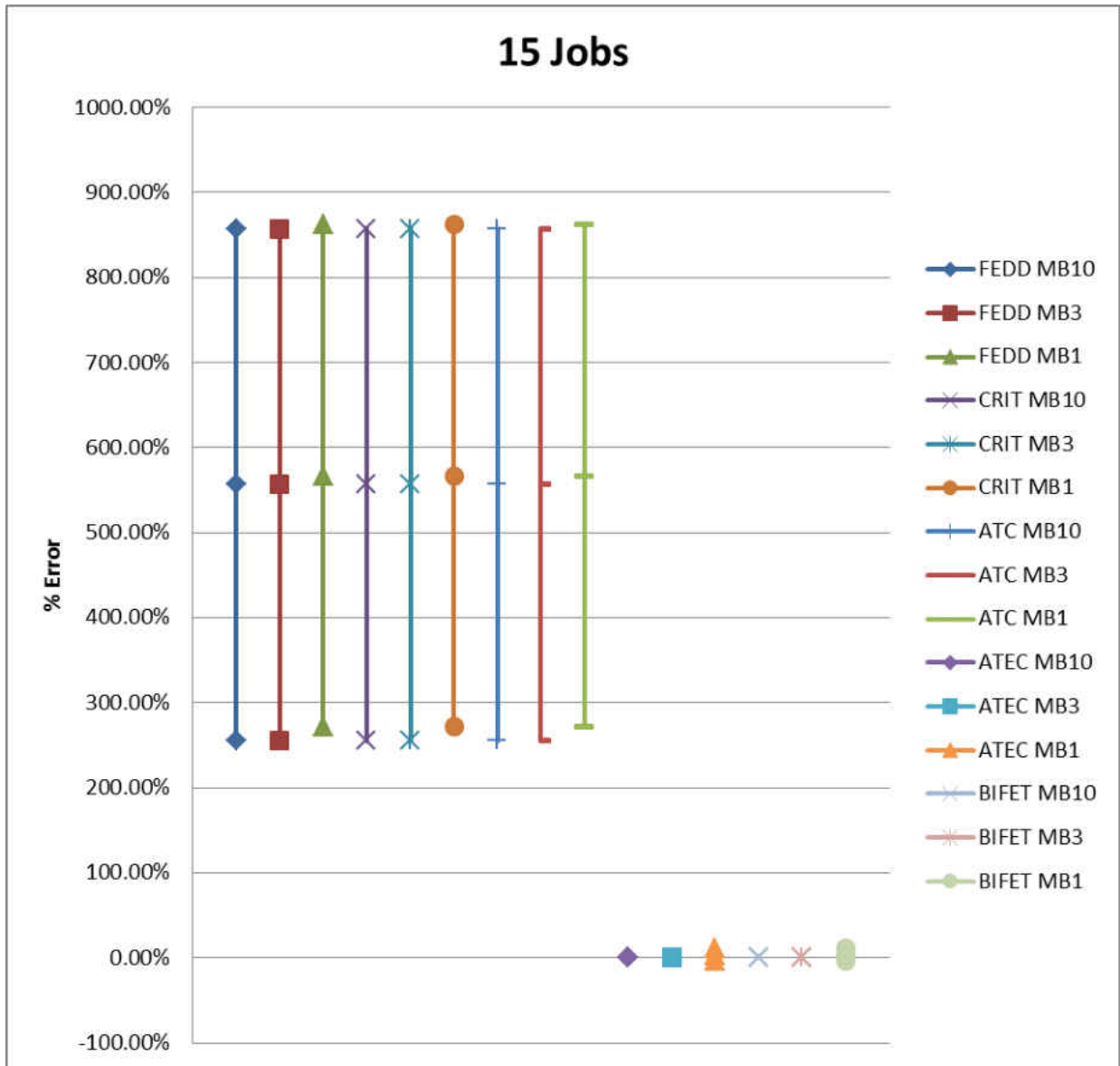


Figure B 14: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

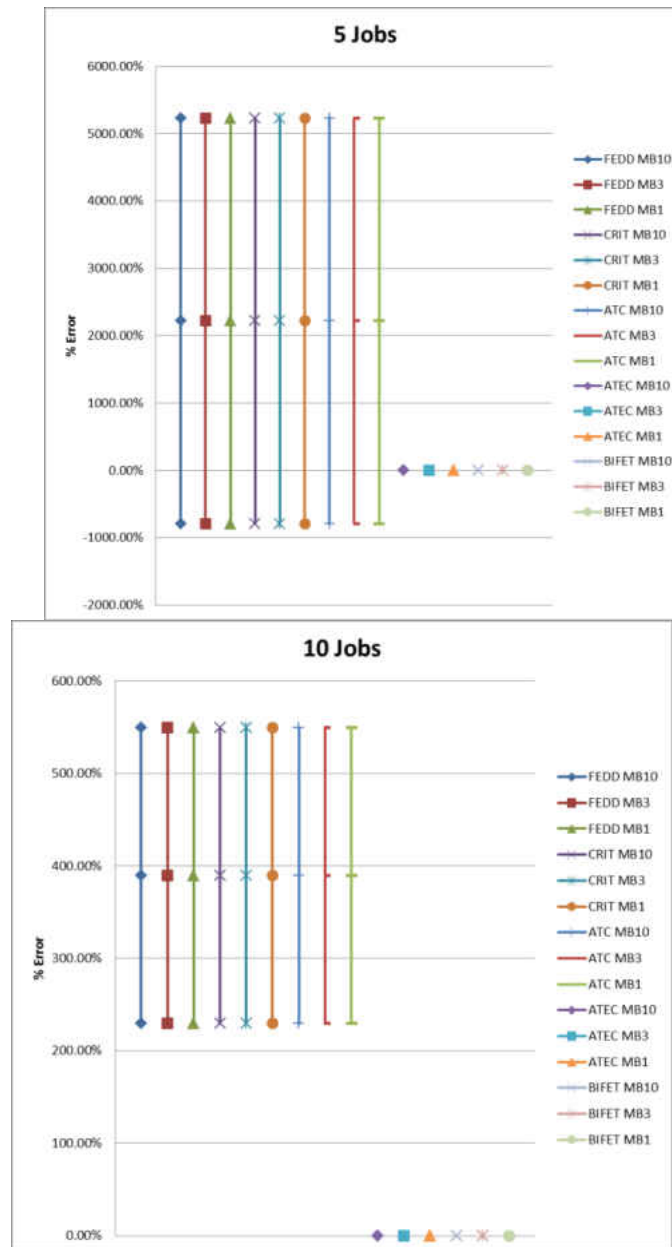


Figure B 15: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families.

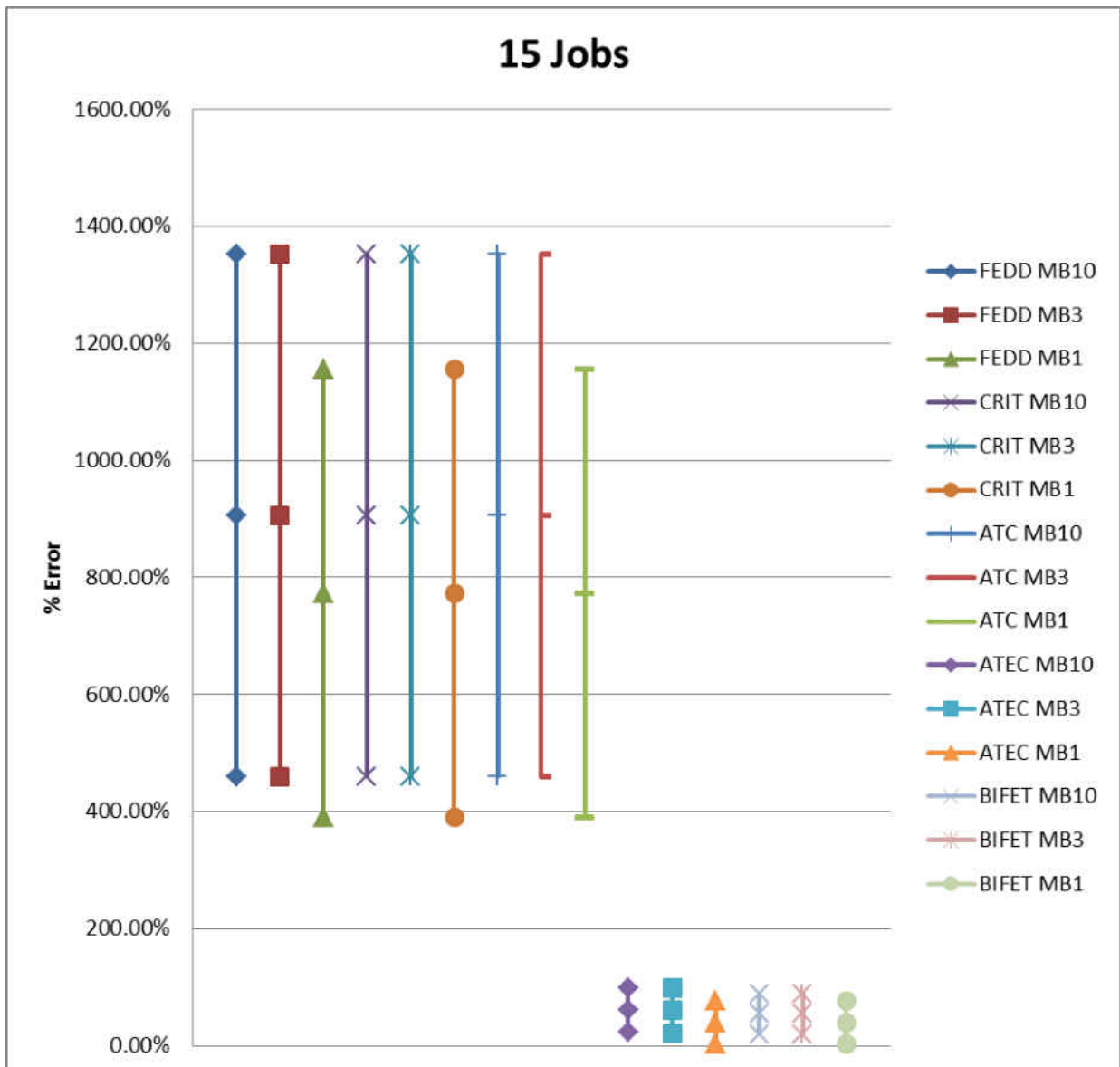


Figure B 16: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families.

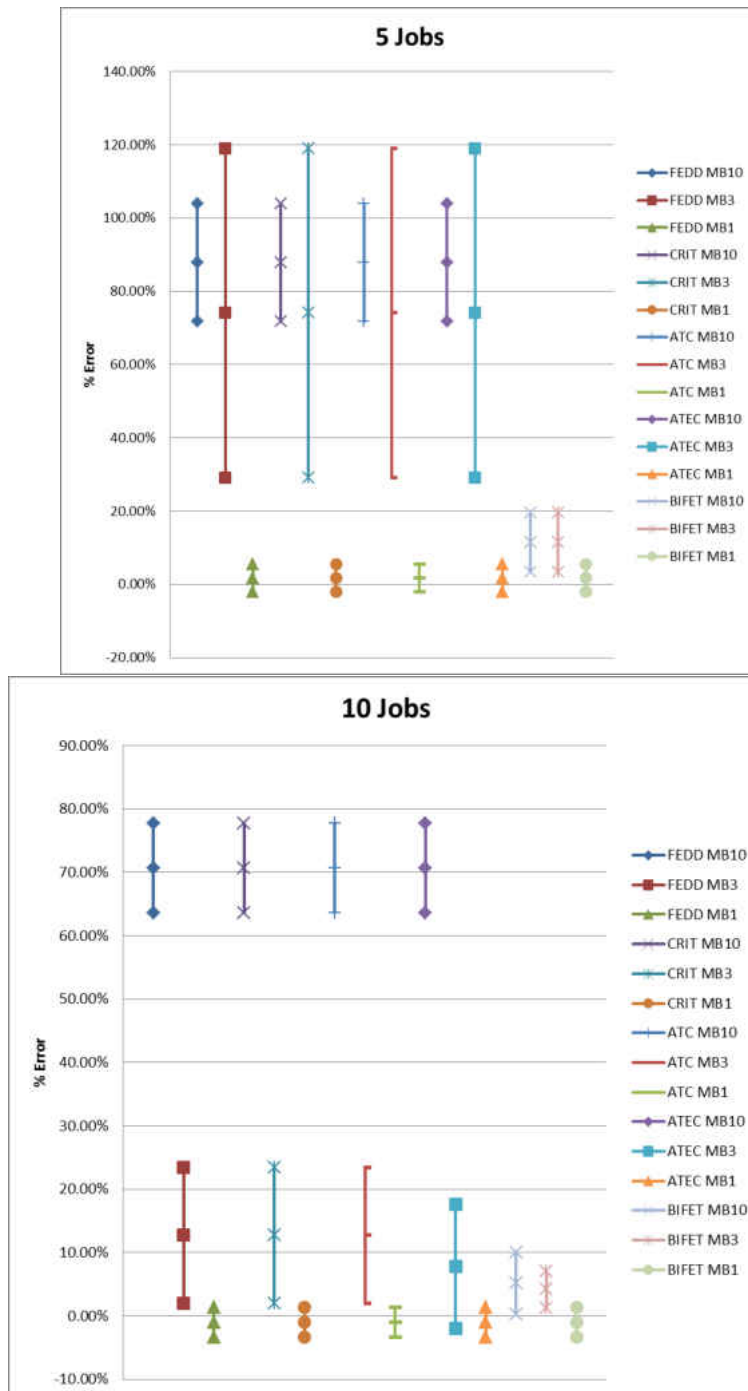


Figure B 17: 95% confidence interval on the performance results loose due dates, simultaneously, 1 family.

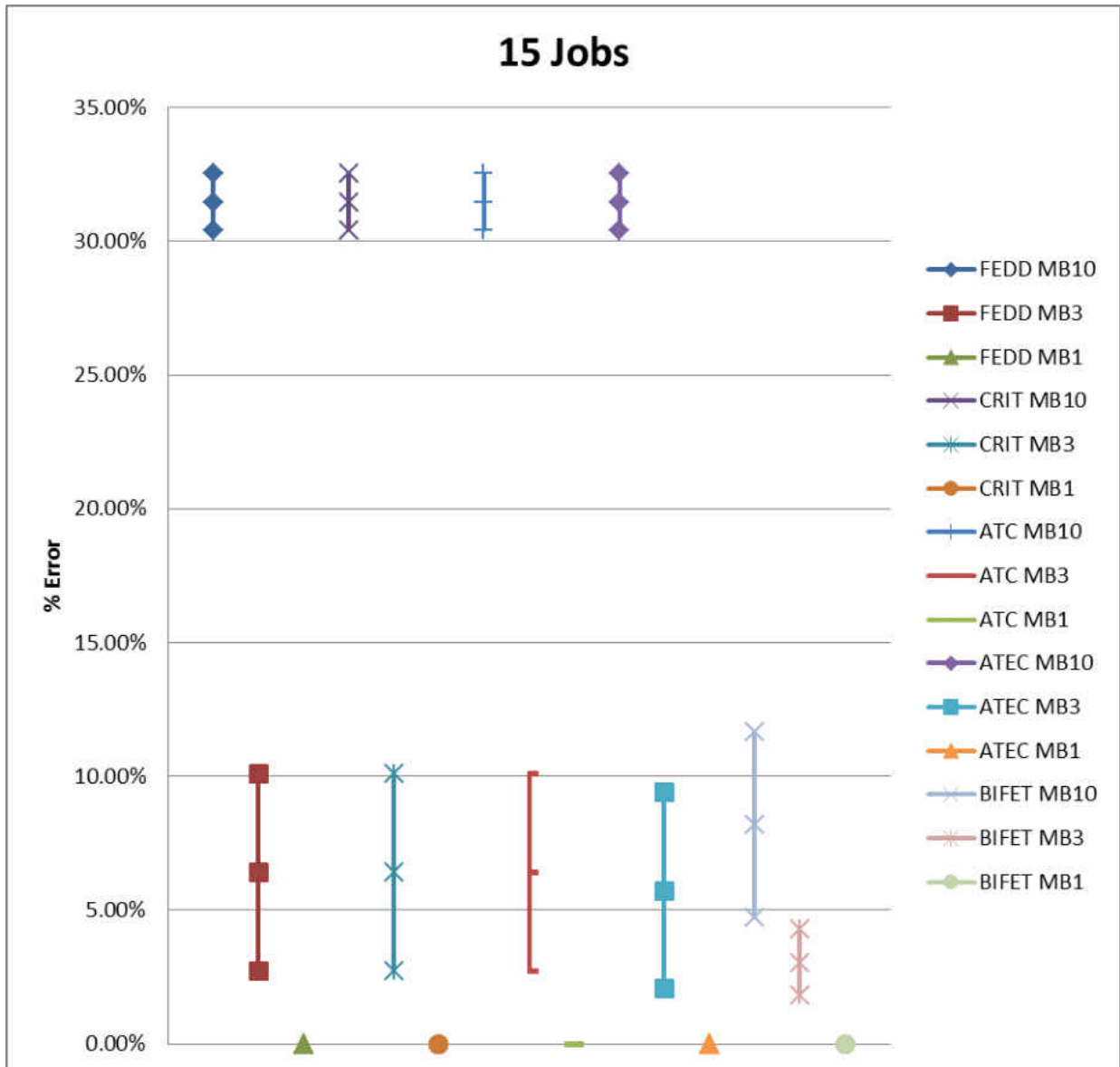


Figure B 18: 95% confidence interval on the performance results for loose due dates, simultaneously, 1 family.

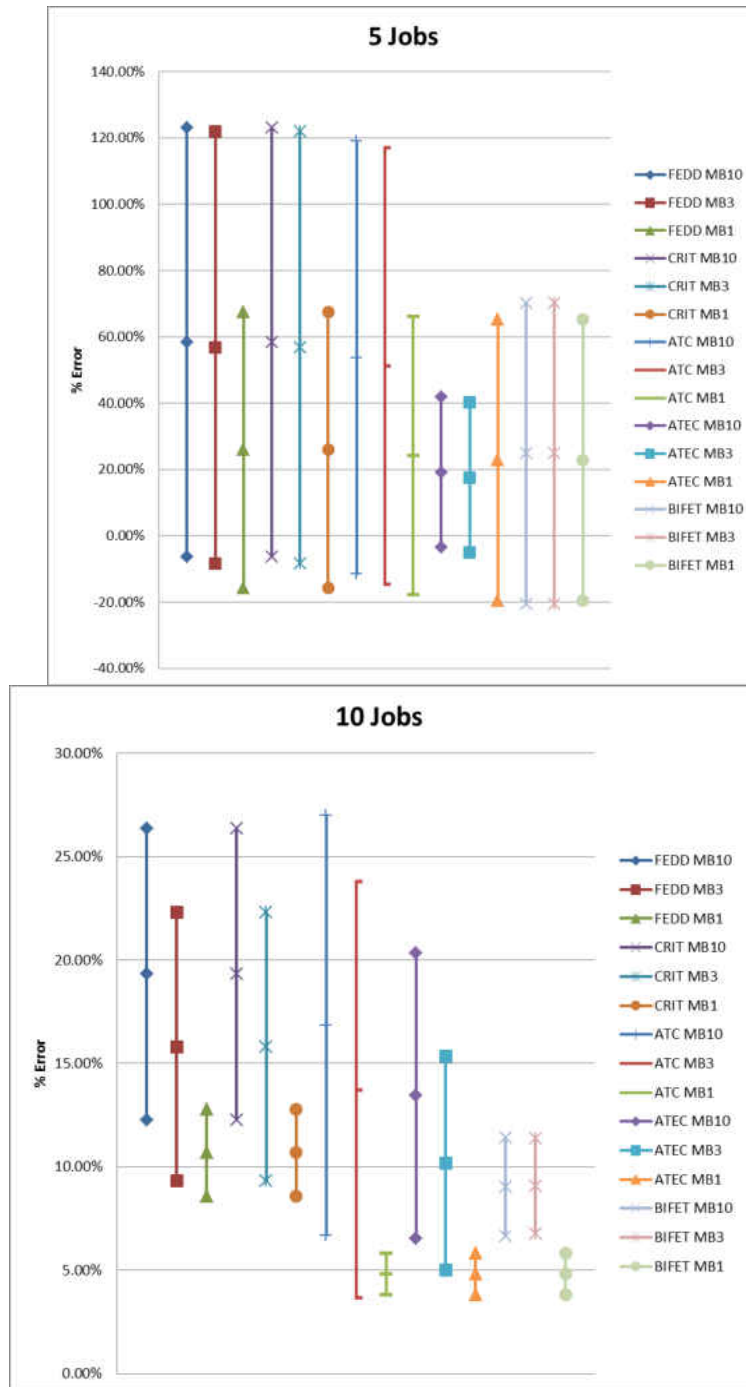


Figure B 19: 95% confidence interval on the performance results for loose due dates, simultaneously, 5 families.

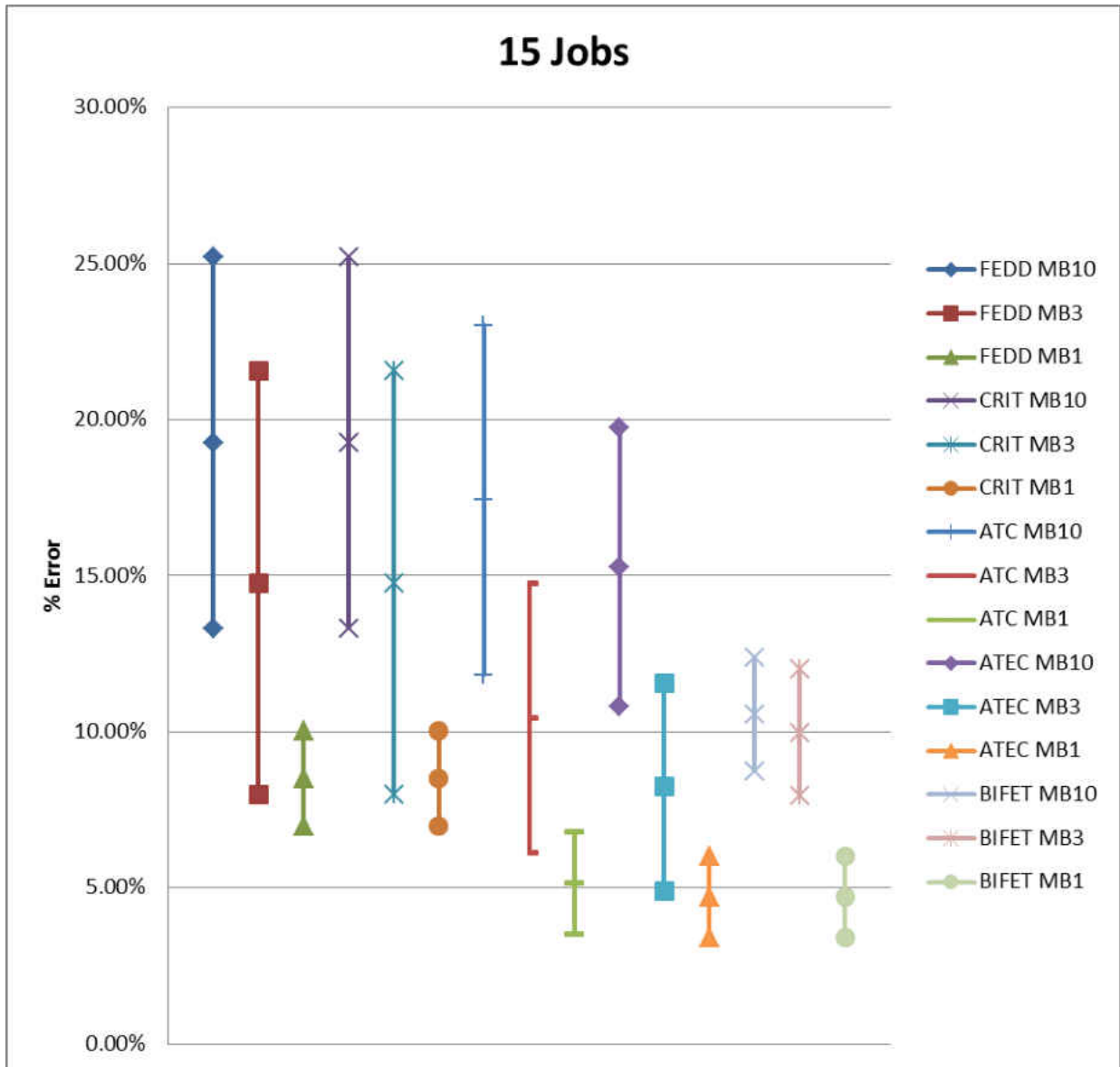


Figure B 20: 95% confidence interval on the performance results for loose due dates, simultaneously, 5 families.

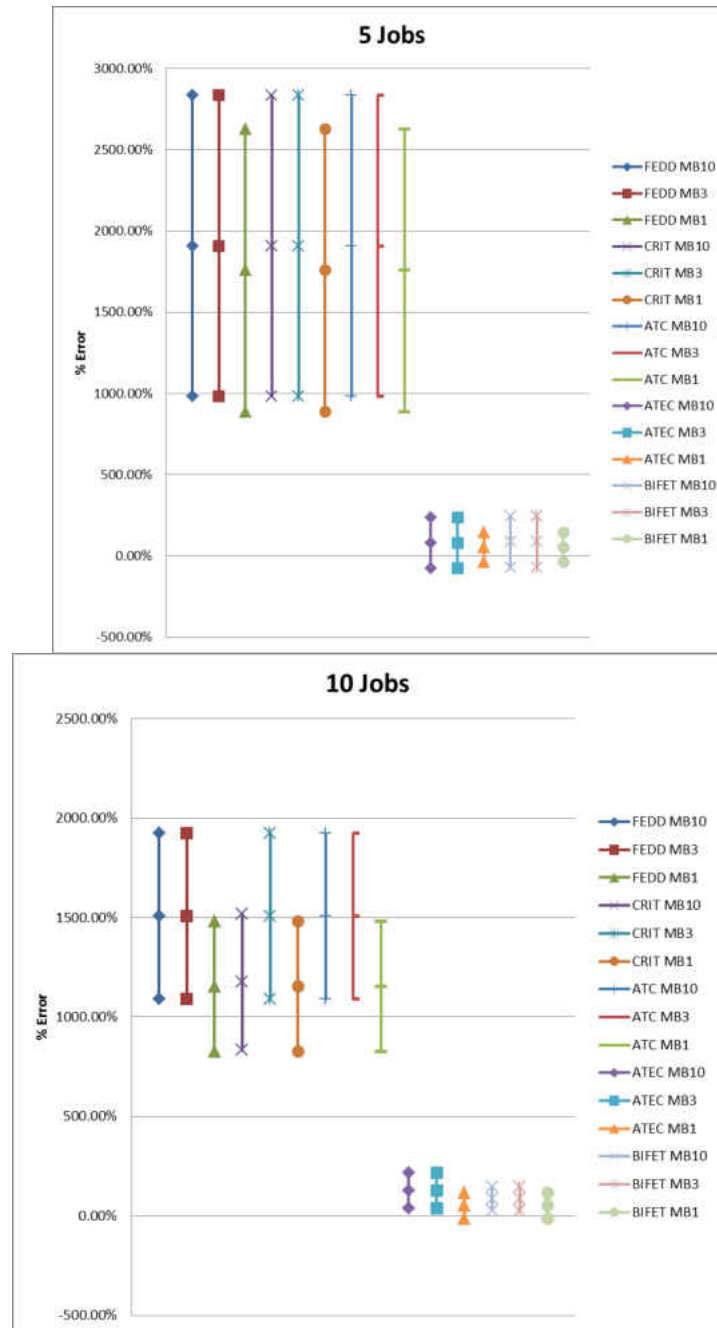


Figure B 21: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

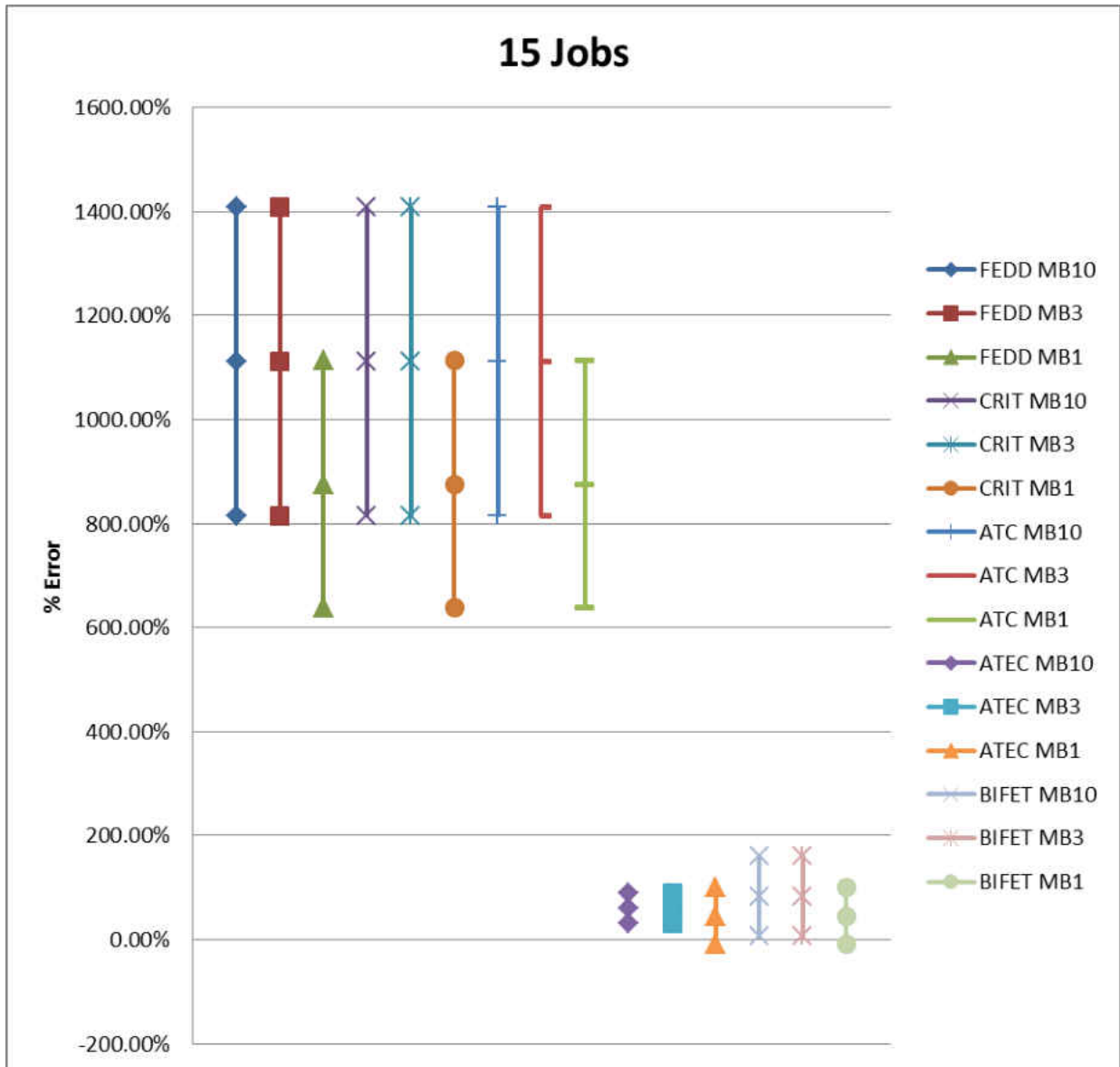


Figure B 22: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

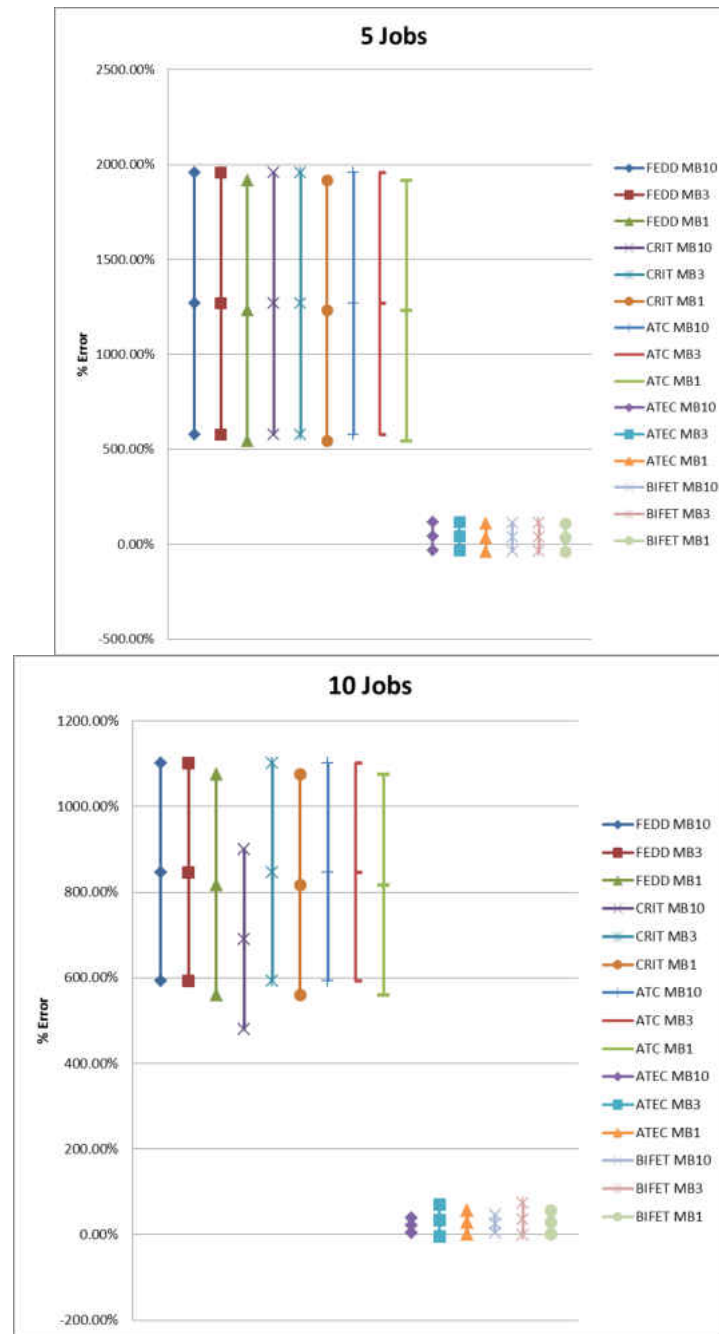


Figure B 23: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

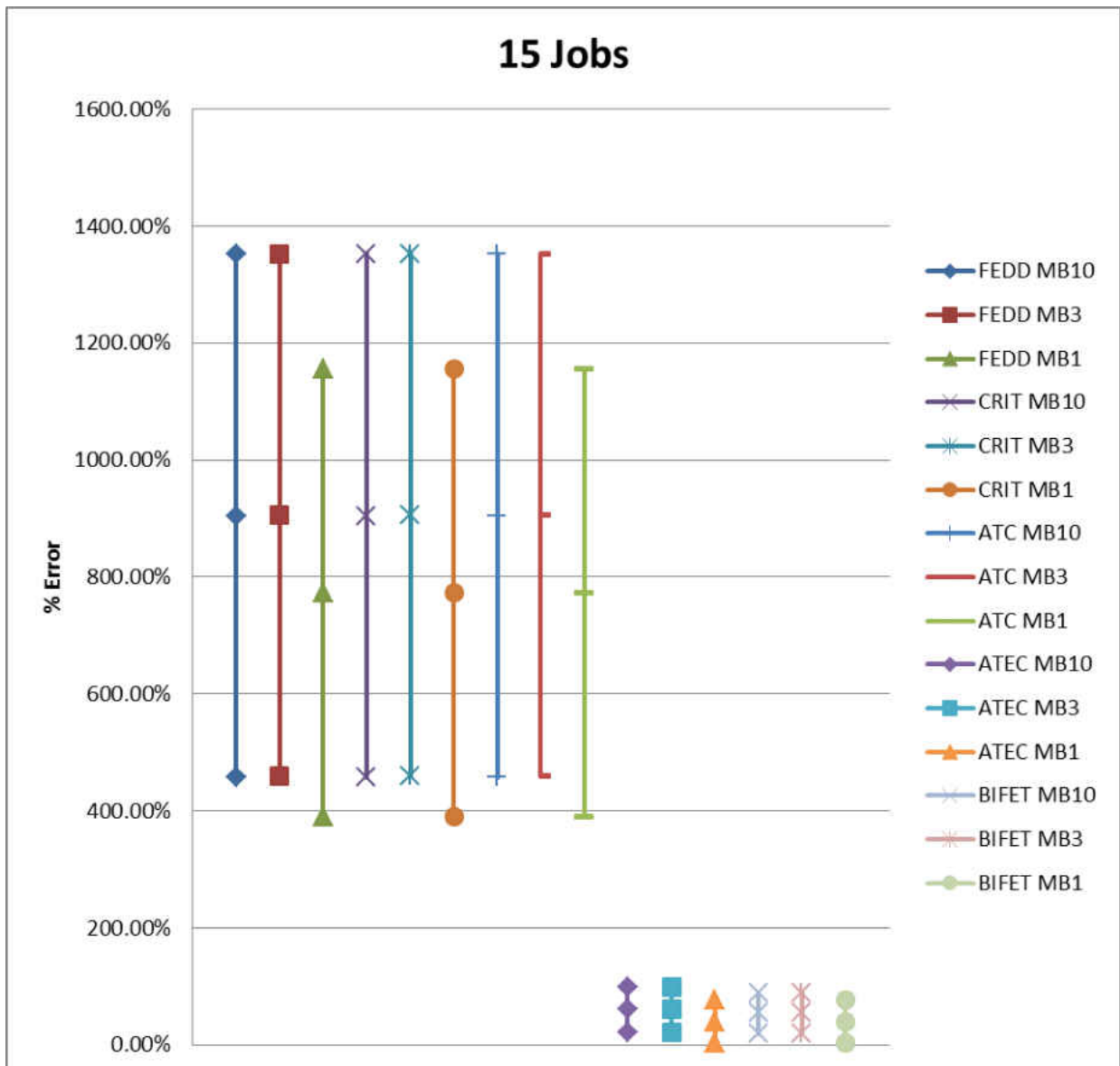


Figure B 24: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

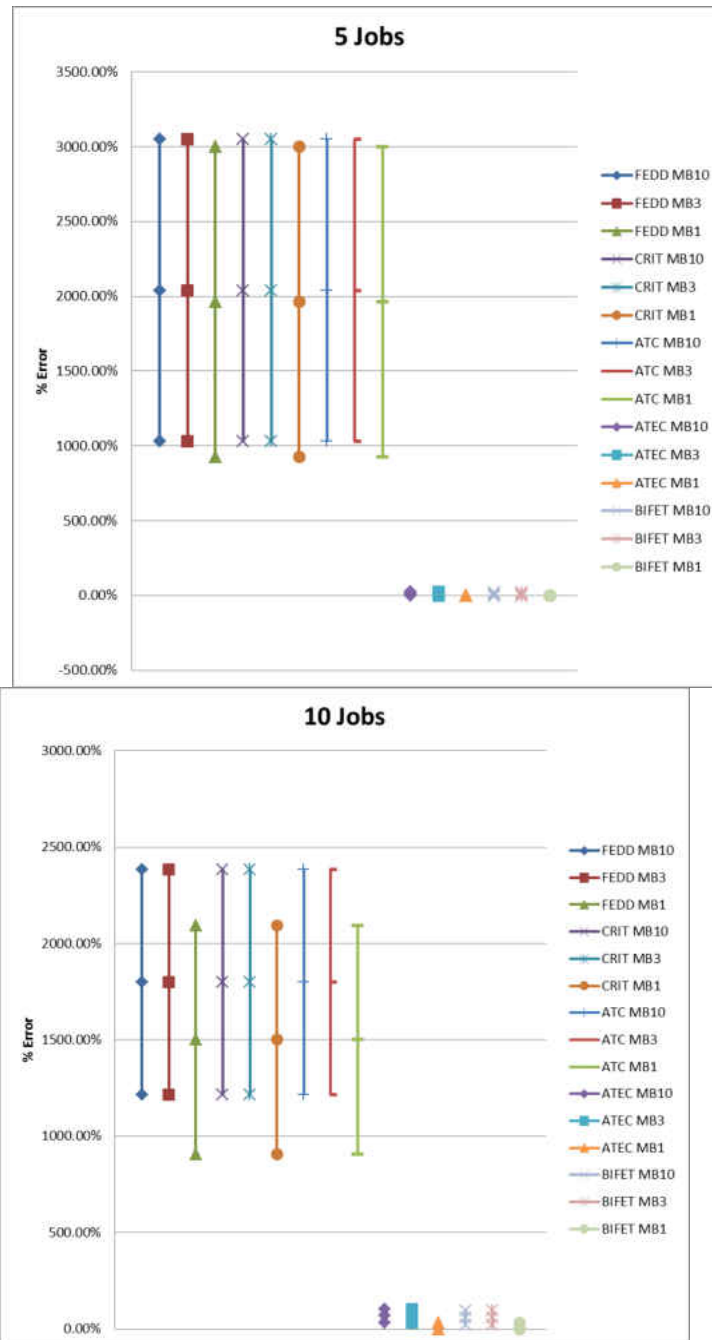


Figure B 25: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

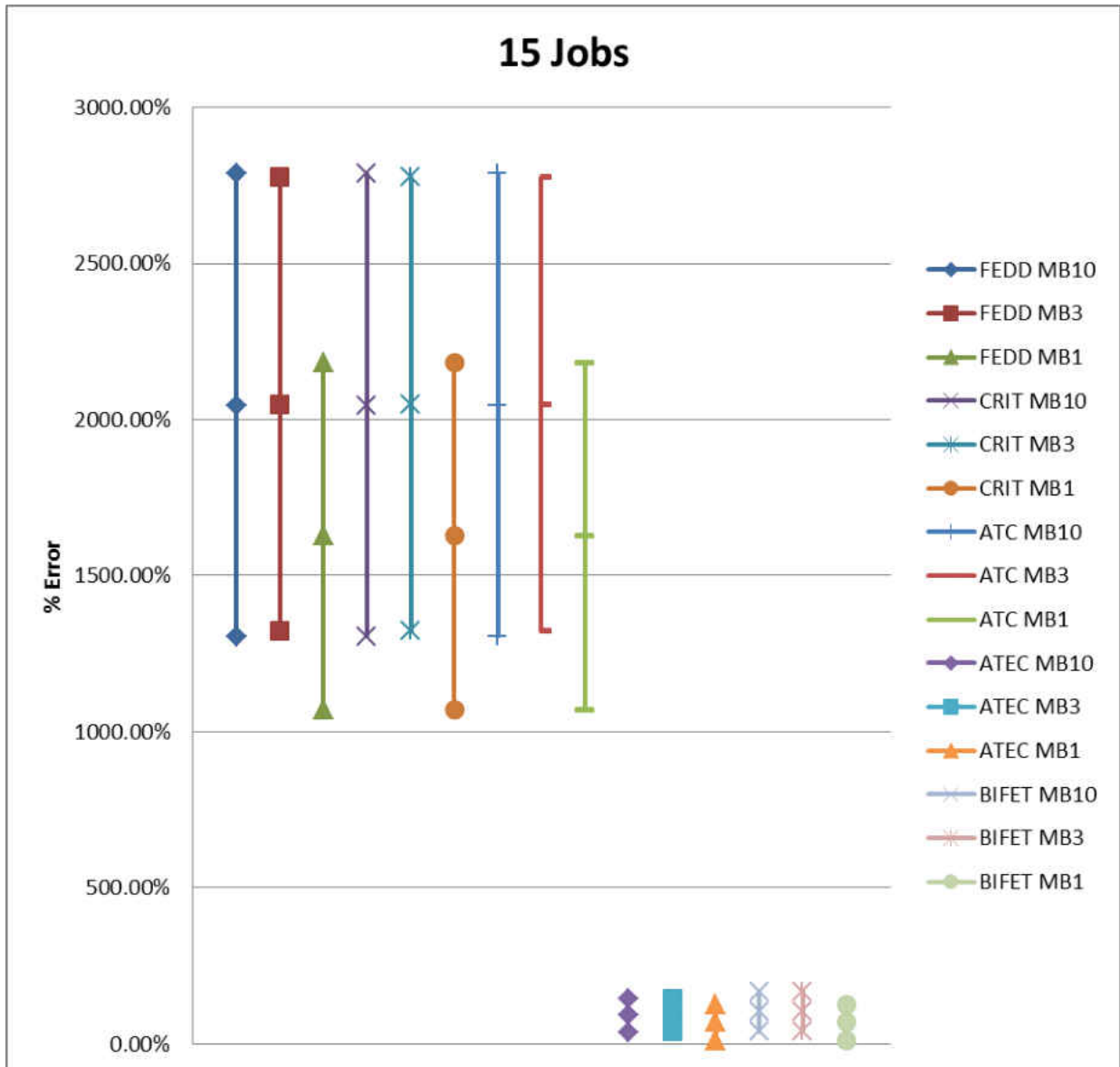


Figure B 26: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

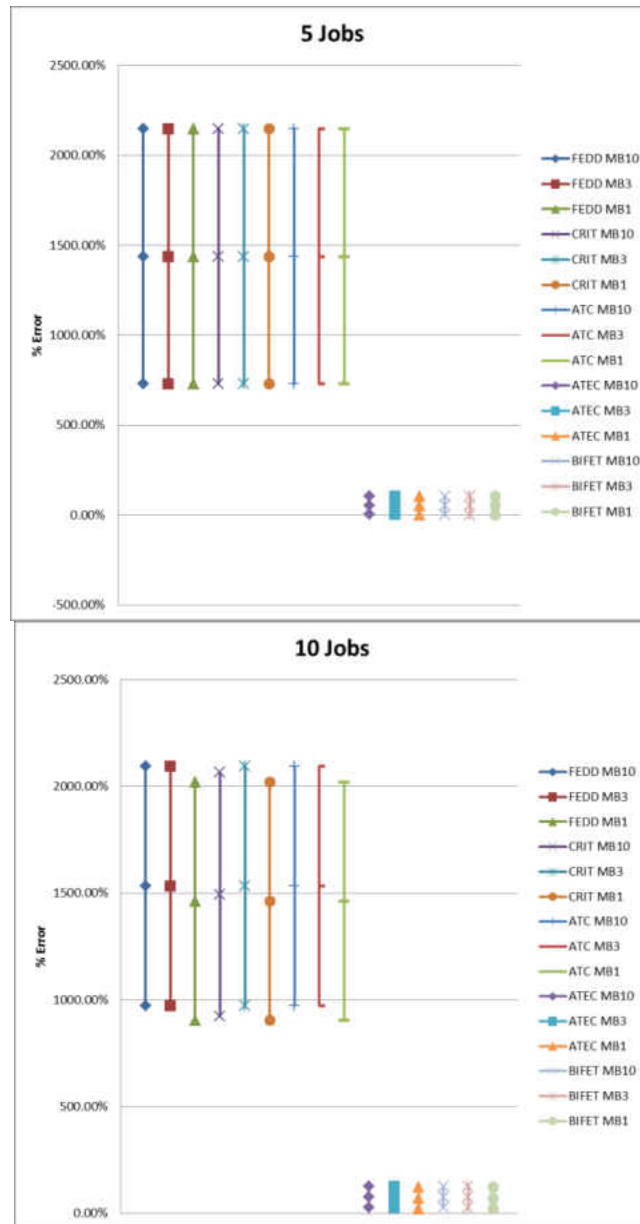


Figure B 27: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families.

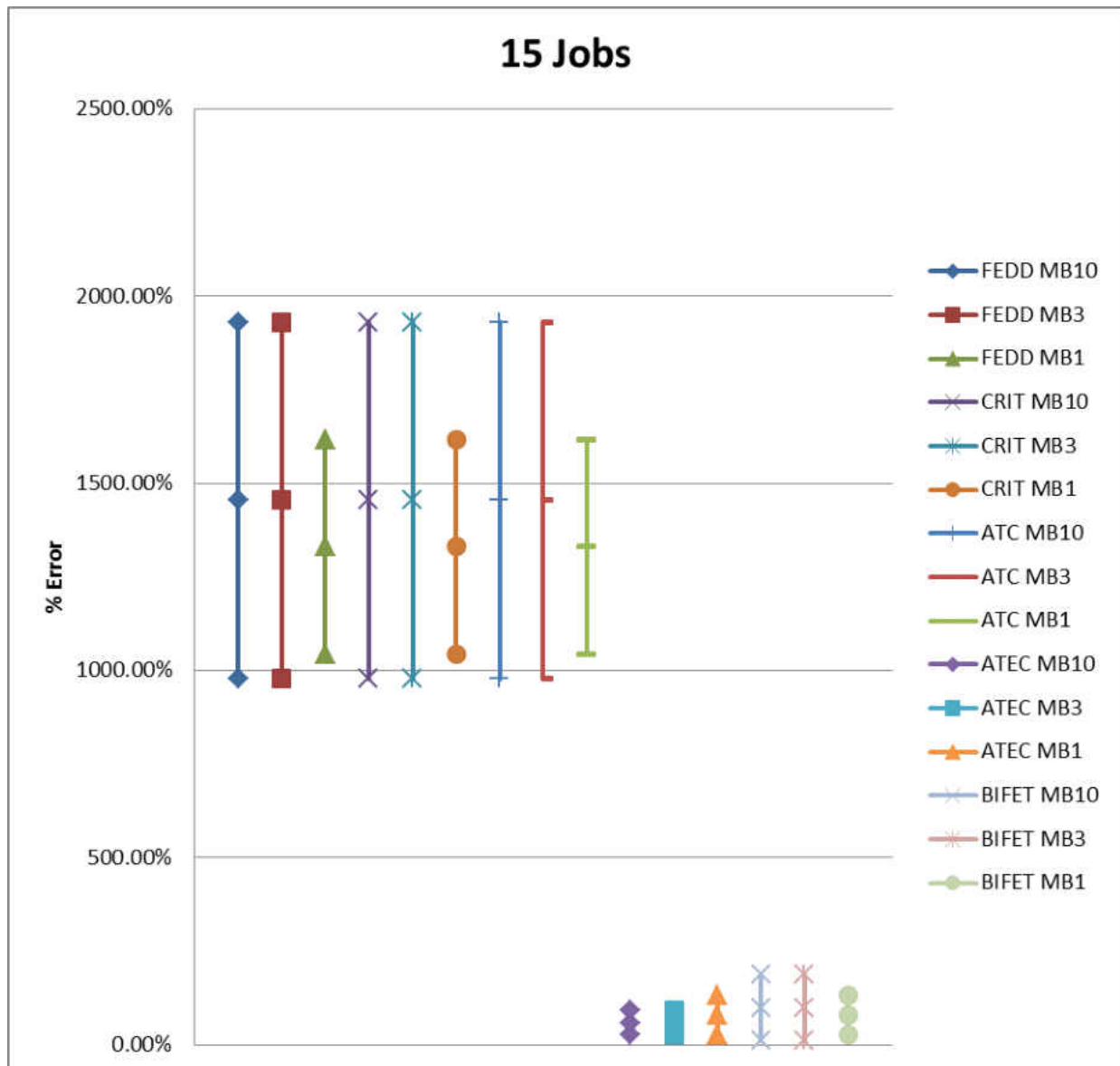


Figure B 28: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families.

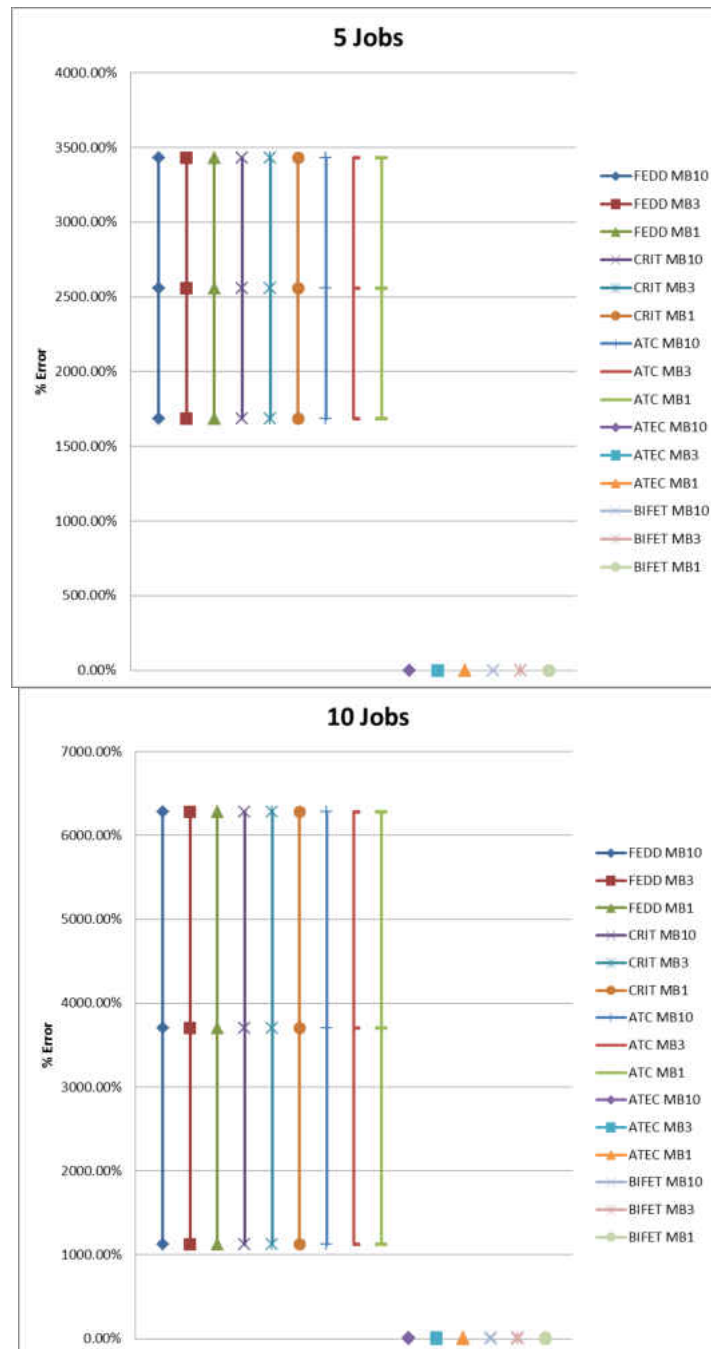


Figure B 29: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

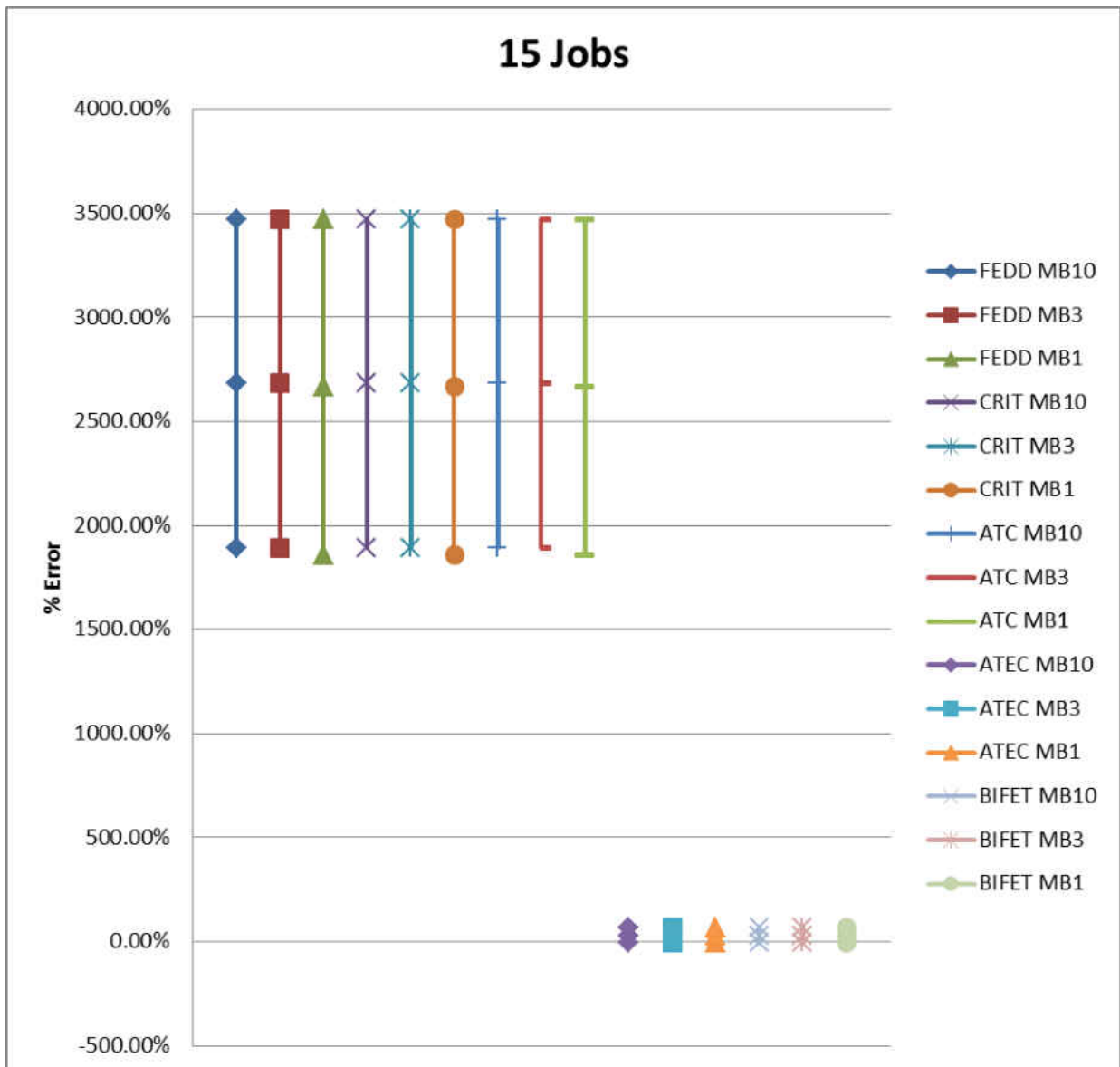


Figure B 30: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

APPENDIX C: TARDY PENALTY=10/EARLY PENALTY = 1

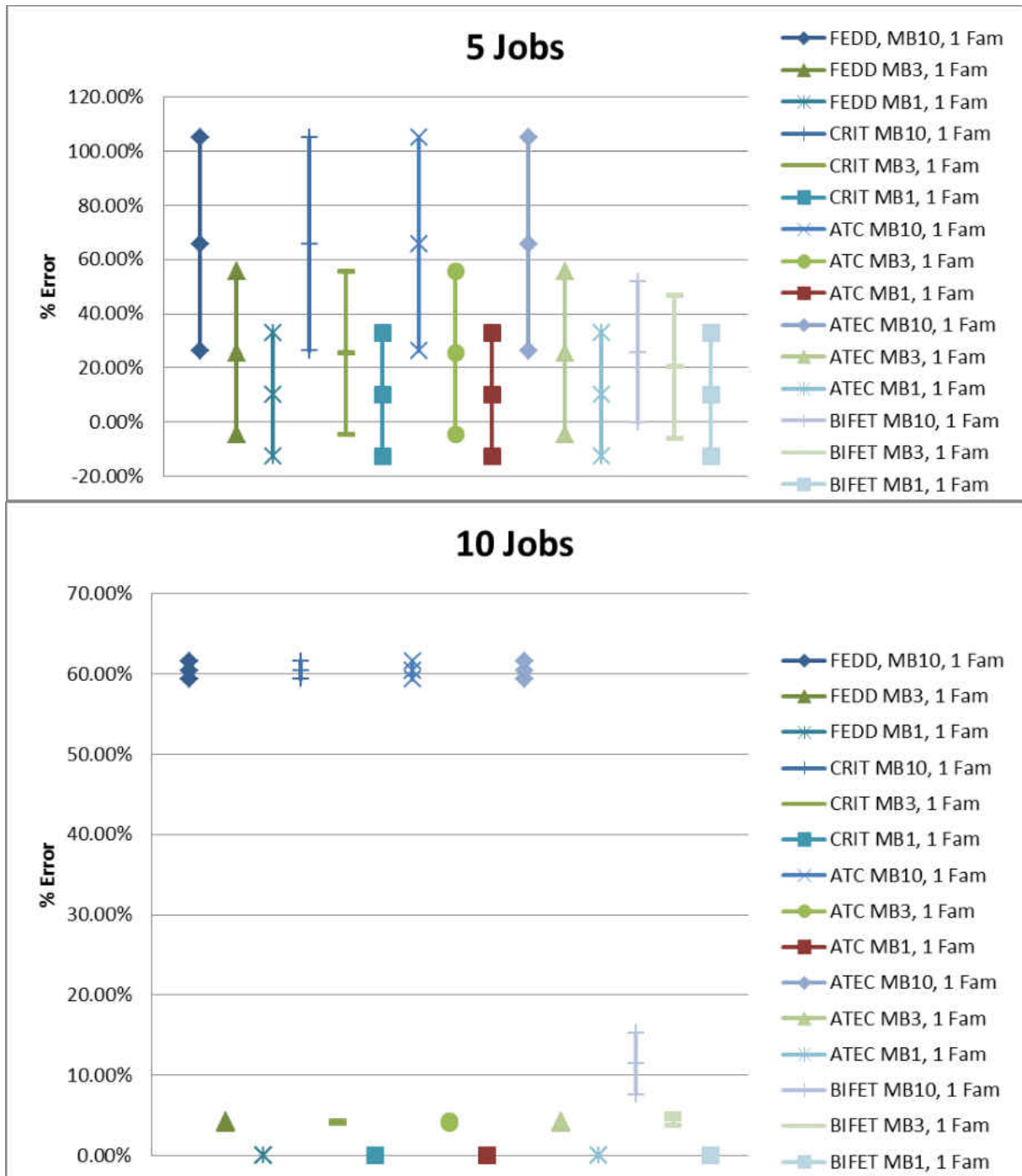


Figure C 1: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 1 family.

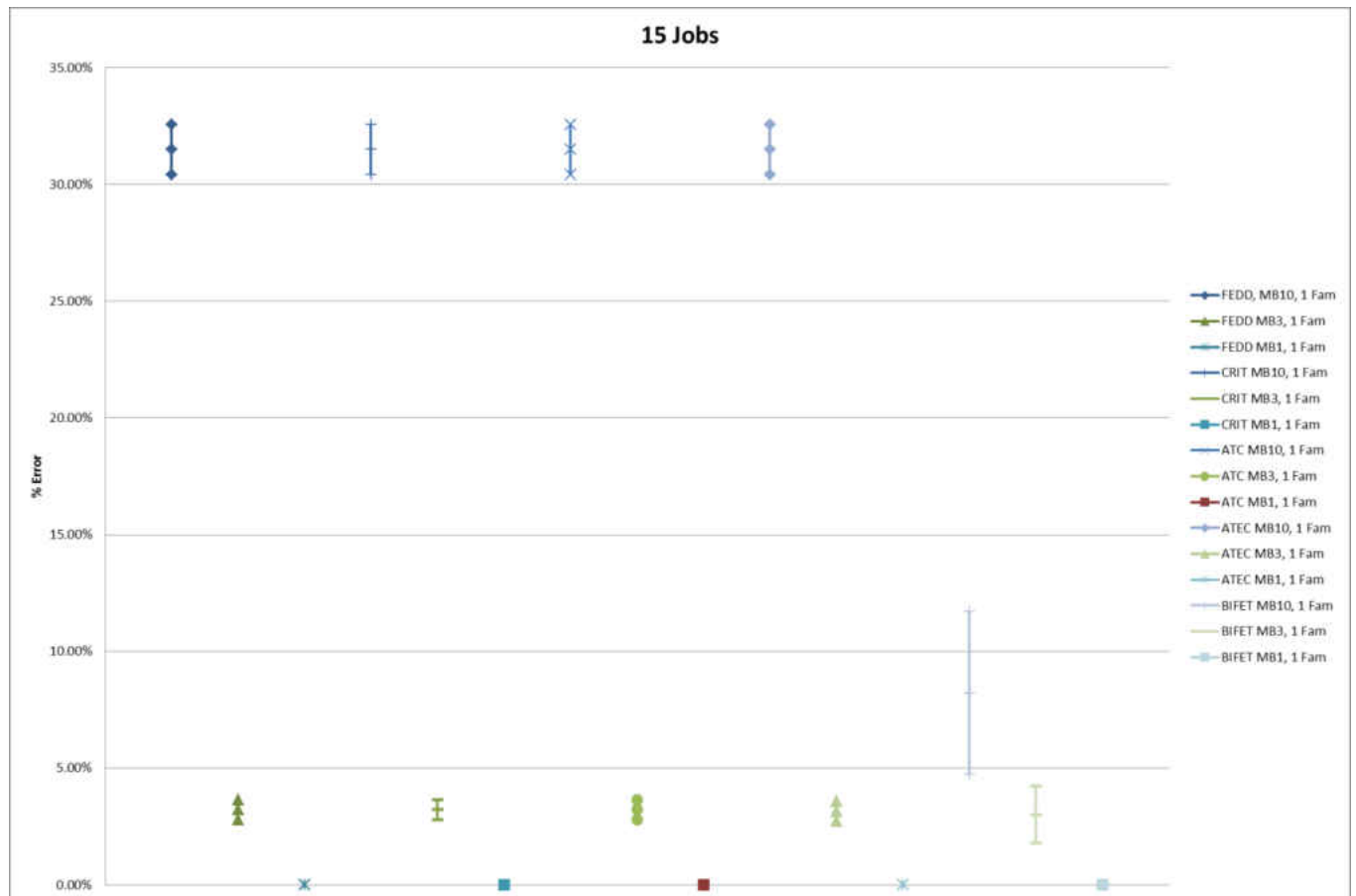


Figure C 2: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 1 family.

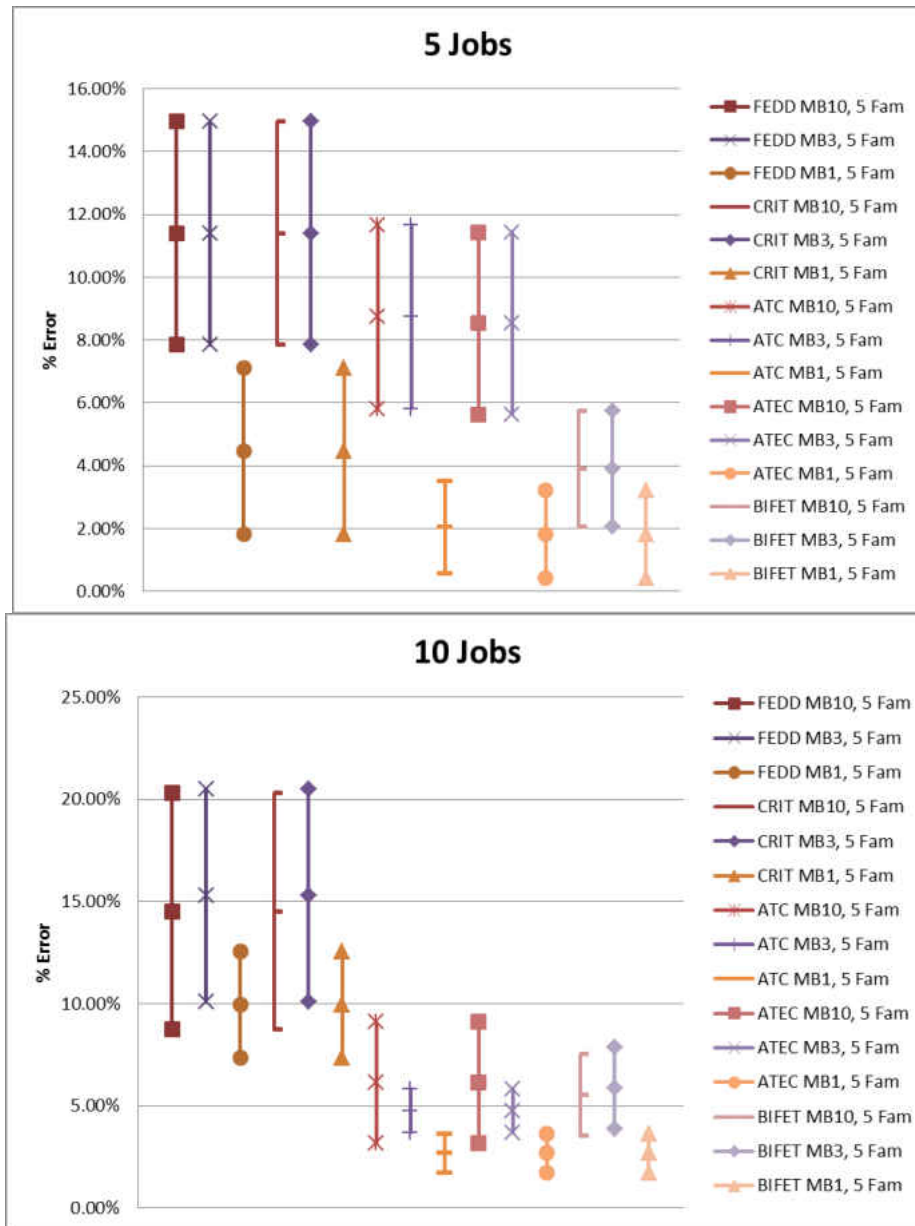


Figure C 3: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 5 families.

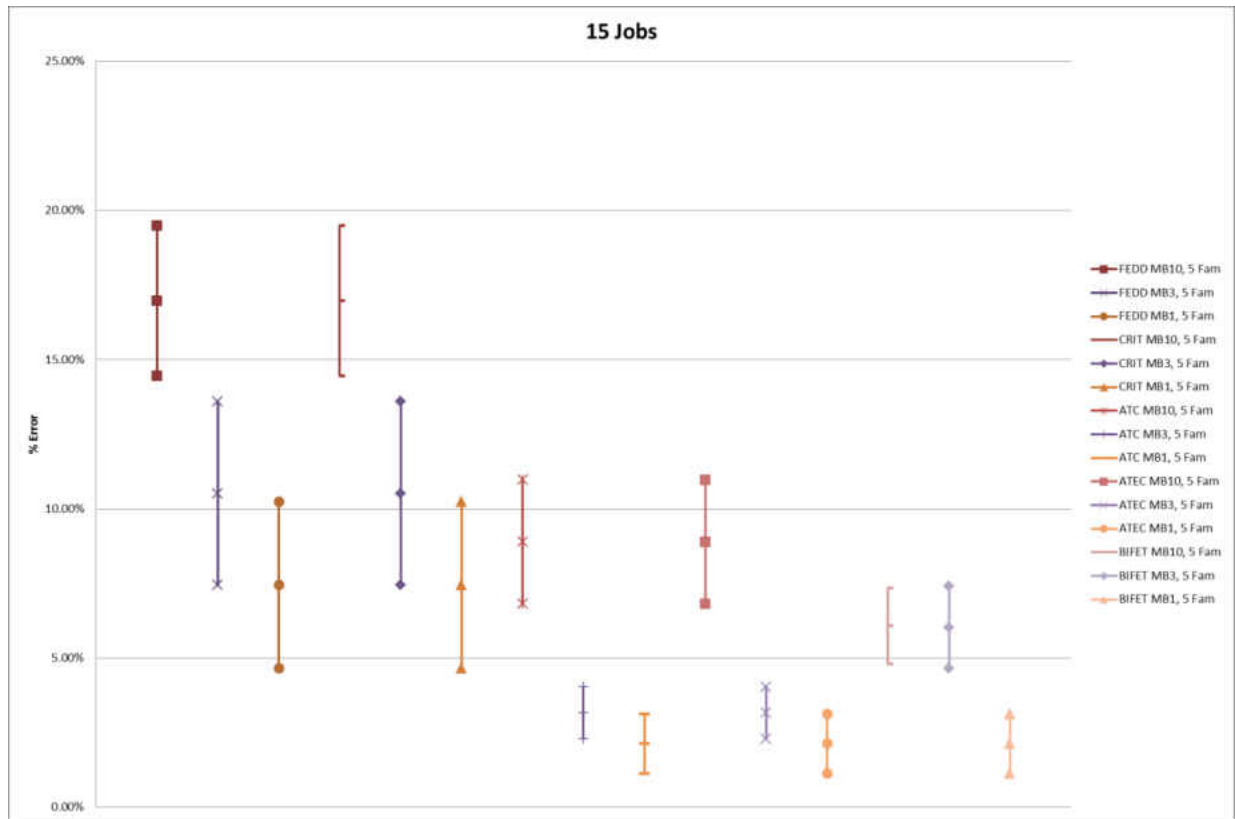


Figure C 4: 95% confidence interval on the performance results for tight due dates, jobs arrive simultaneously, 5 families.

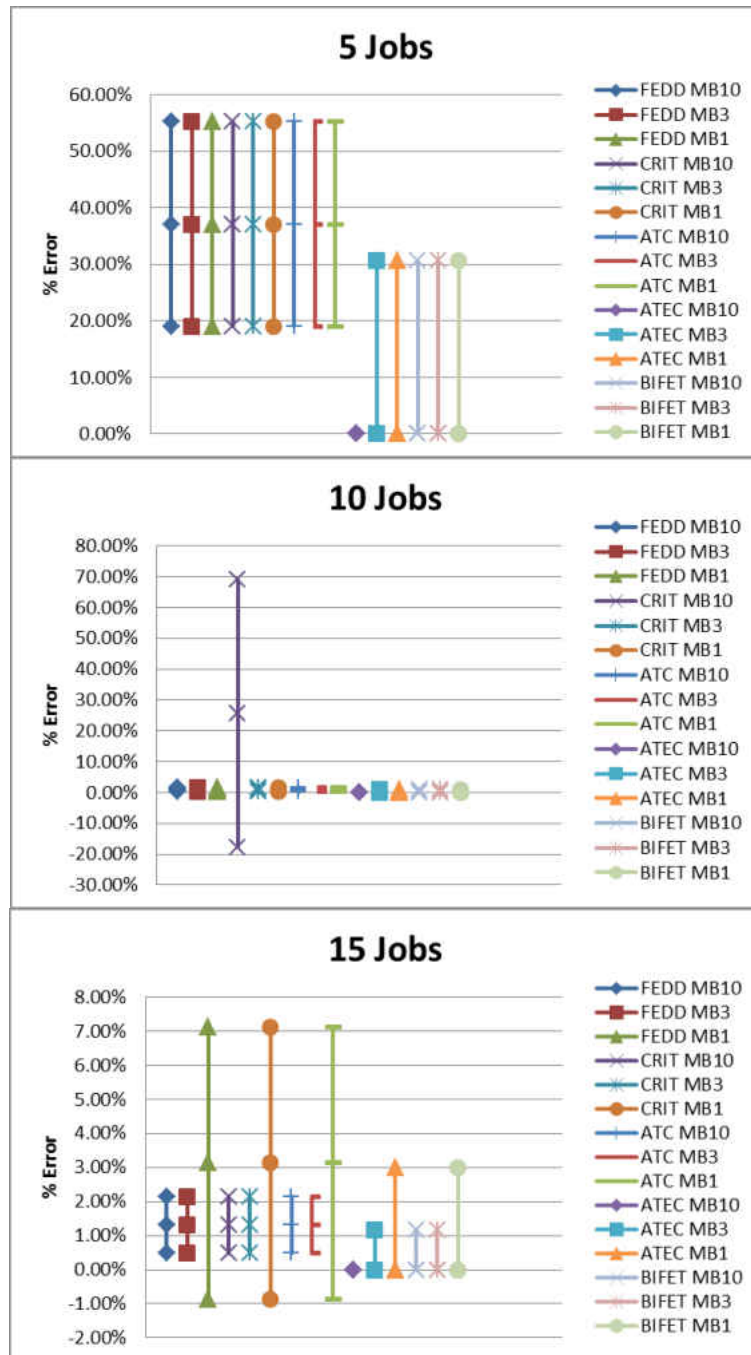


Figure C 5: 95% confidence interval on the performance results for tight due dates, high traffic, 1 family.

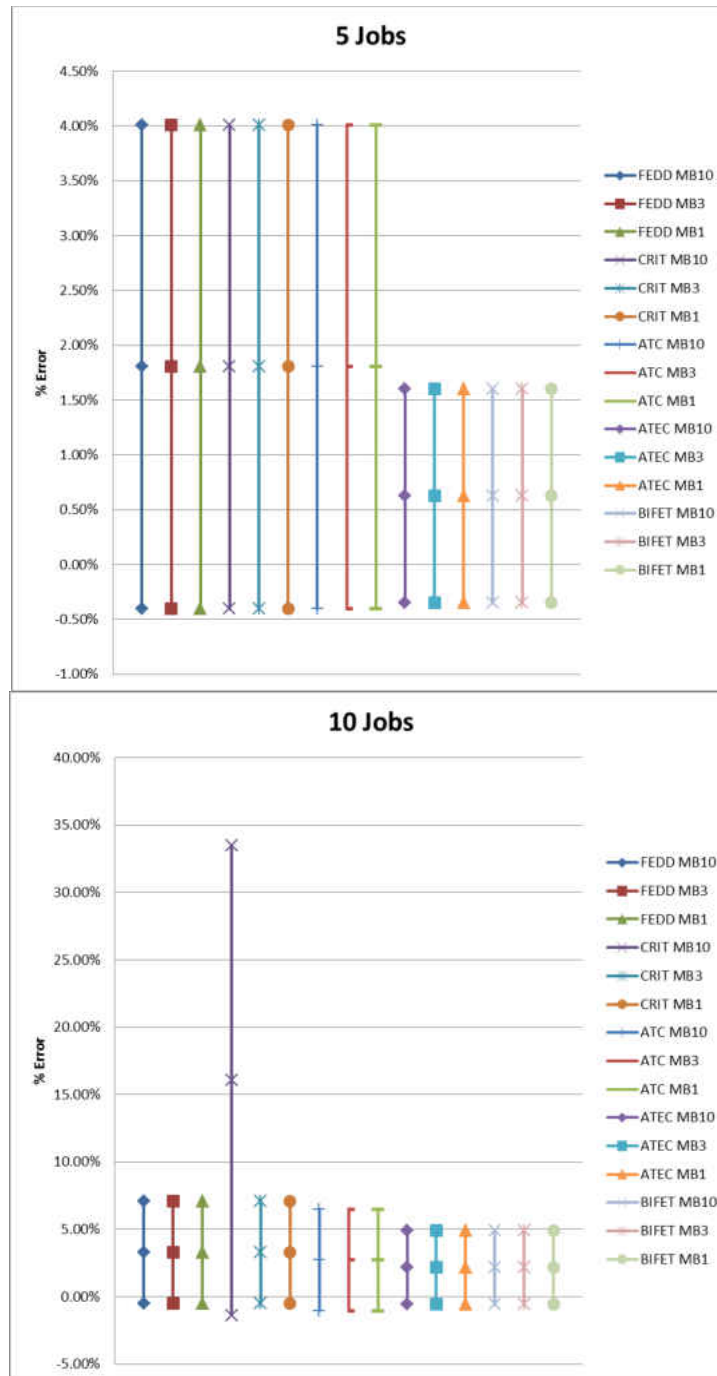


Figure C 6: Tight Due Dates, High Traffic, 5 Families

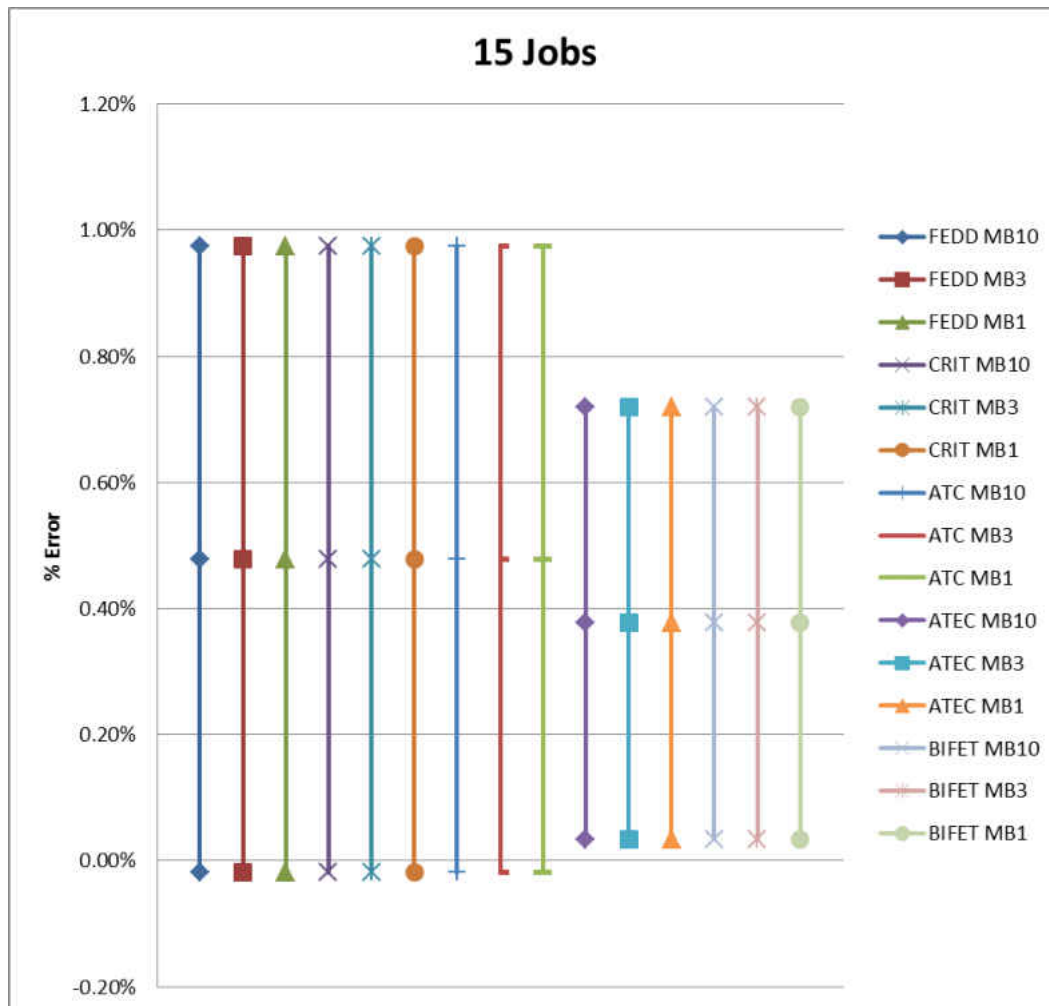


Figure C 7: 95% confidence interval on the performance results for tight due dates, high traffic, 5 families.

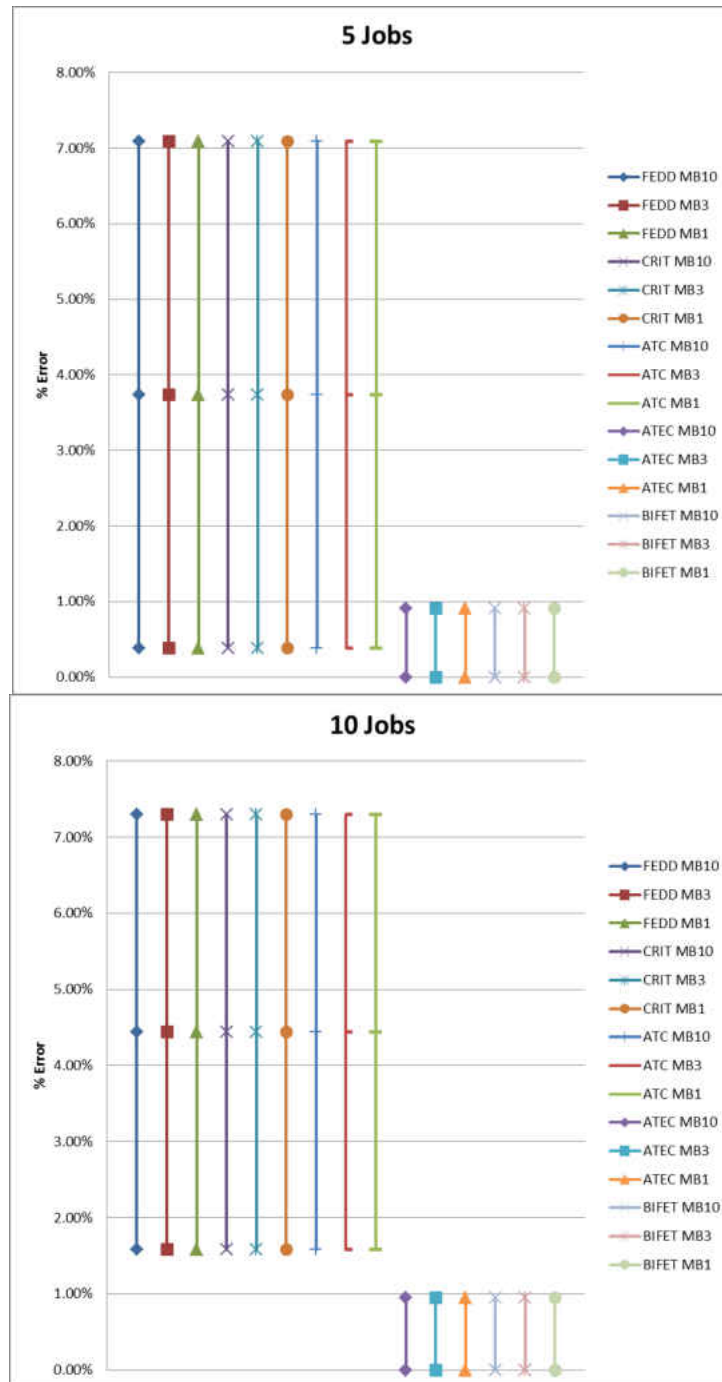


Figure C 8: 95% confidence interval on the performance results for tight due dates, medium traffic, 1 family.

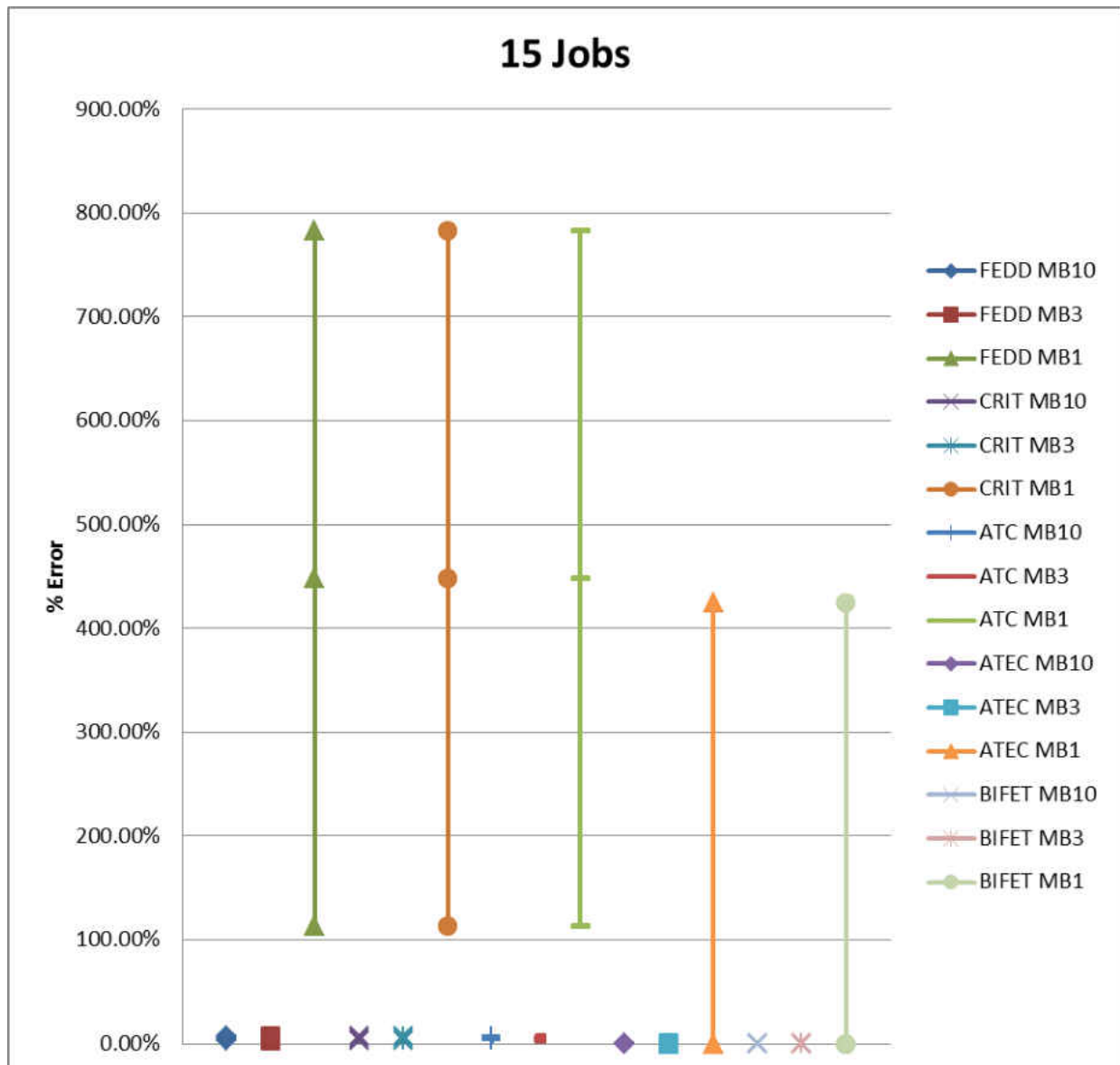


Figure C 9: 95% confidence interval on the performance results for tight due dates, medium traffic, 1 family.

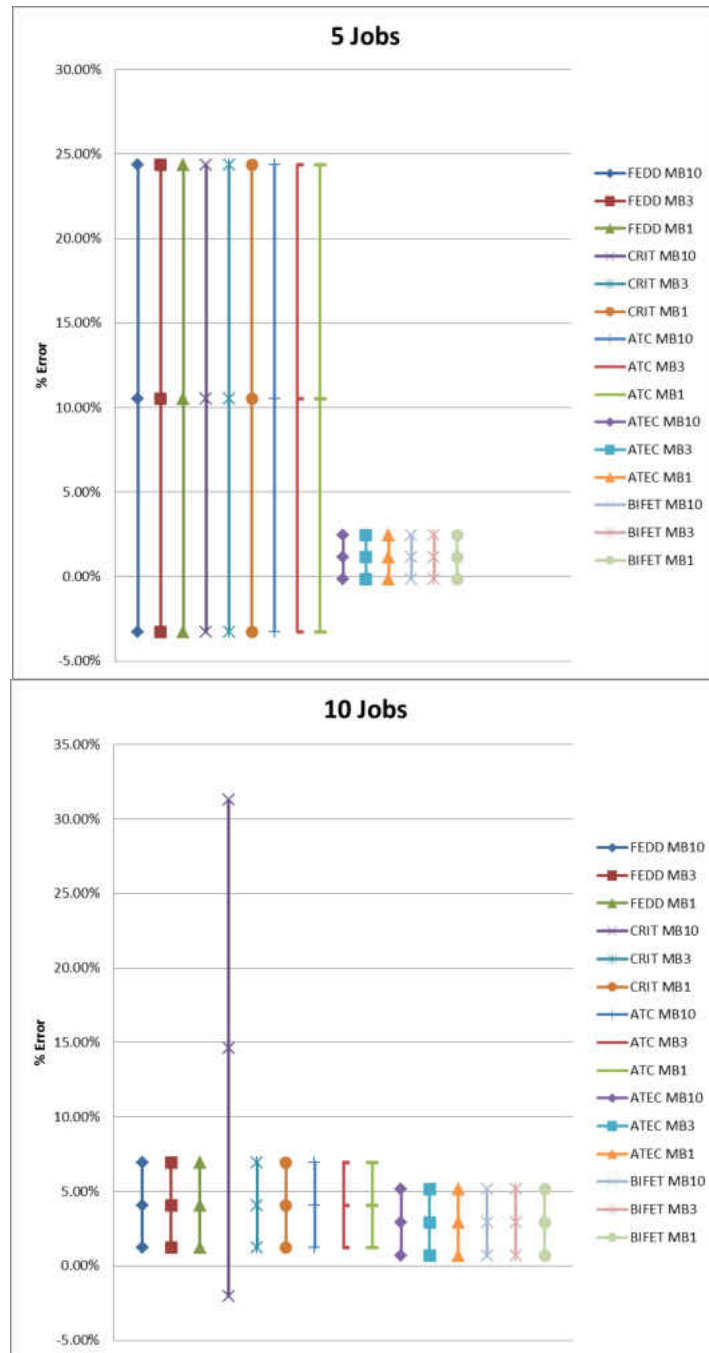


Figure C 10: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

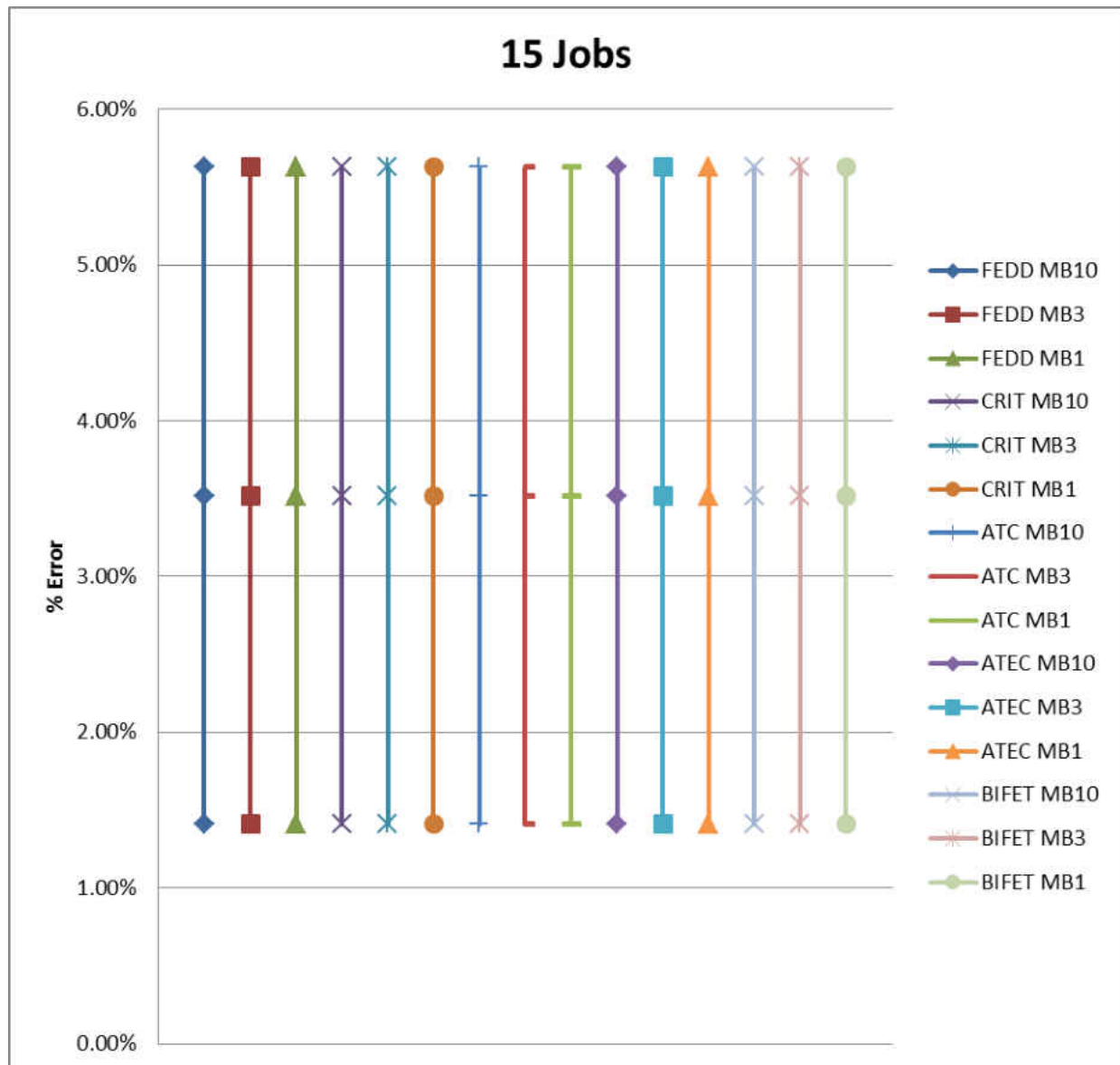


Figure C 11: 95% confidence interval on the performance results for tight due dates, medium traffic, 5 families.

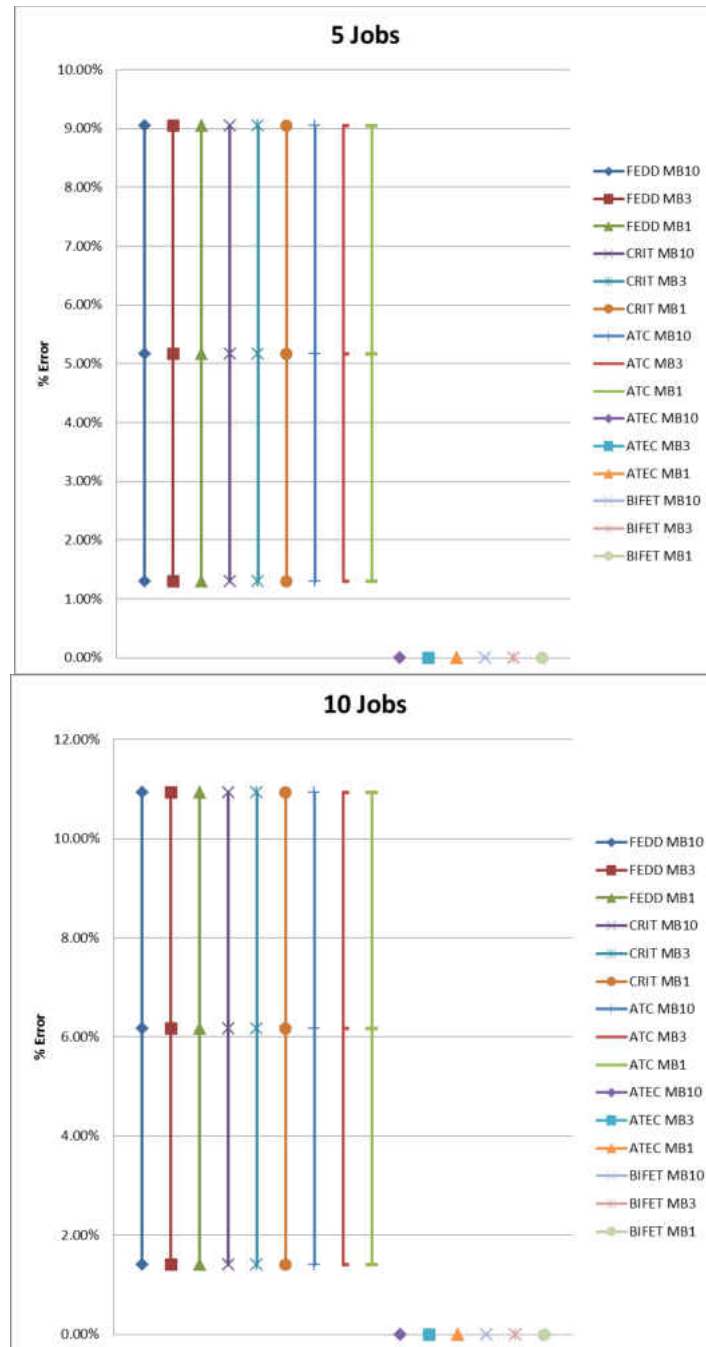


Figure C 12: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

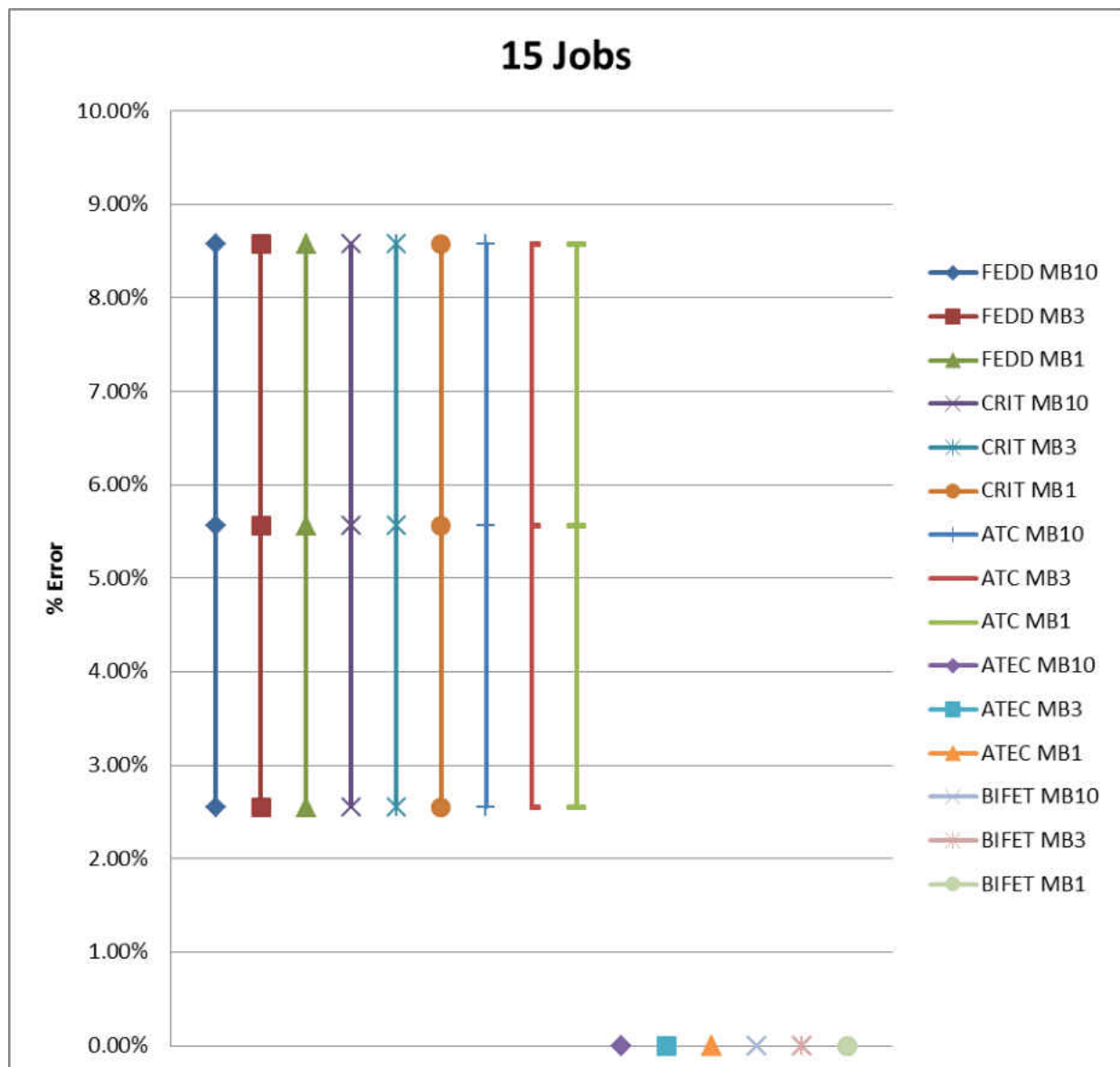


Figure C 13: 95% confidence interval on the performance results for tight due dates, low traffic, 1 family.

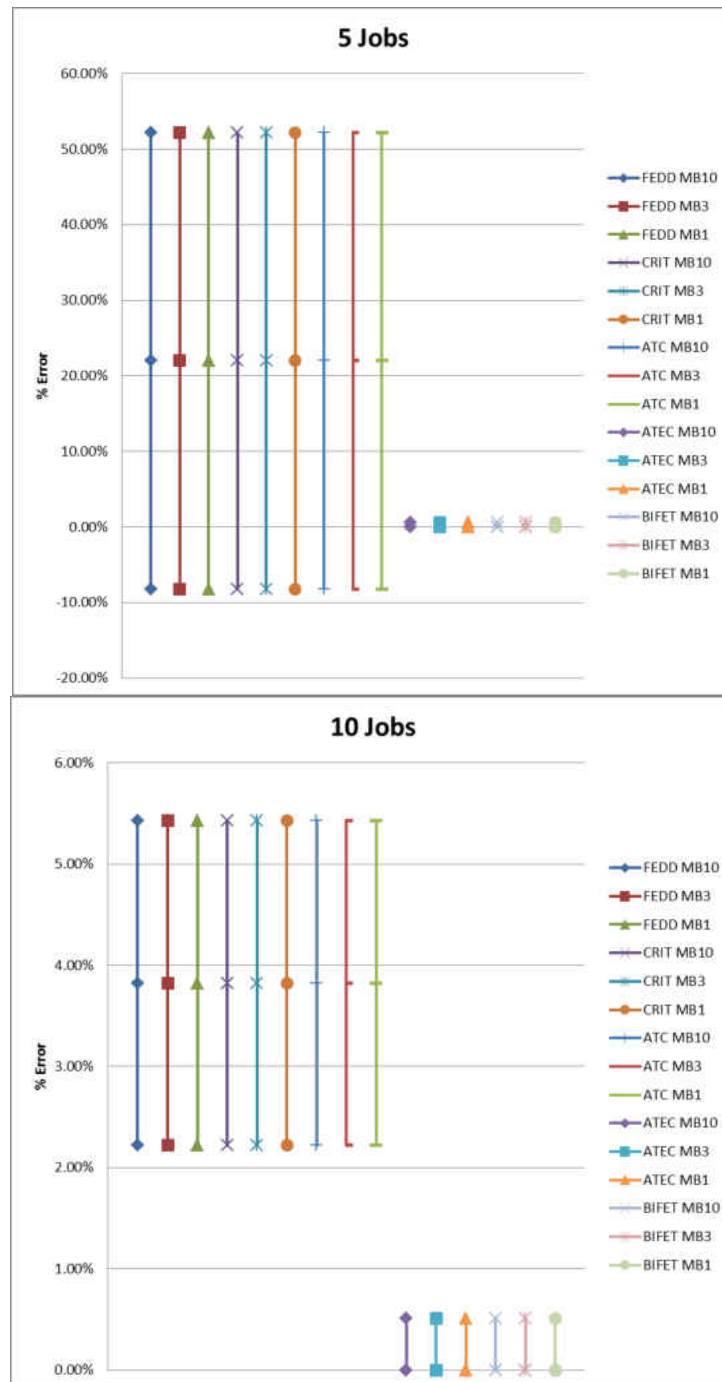


Figure C 14: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families.

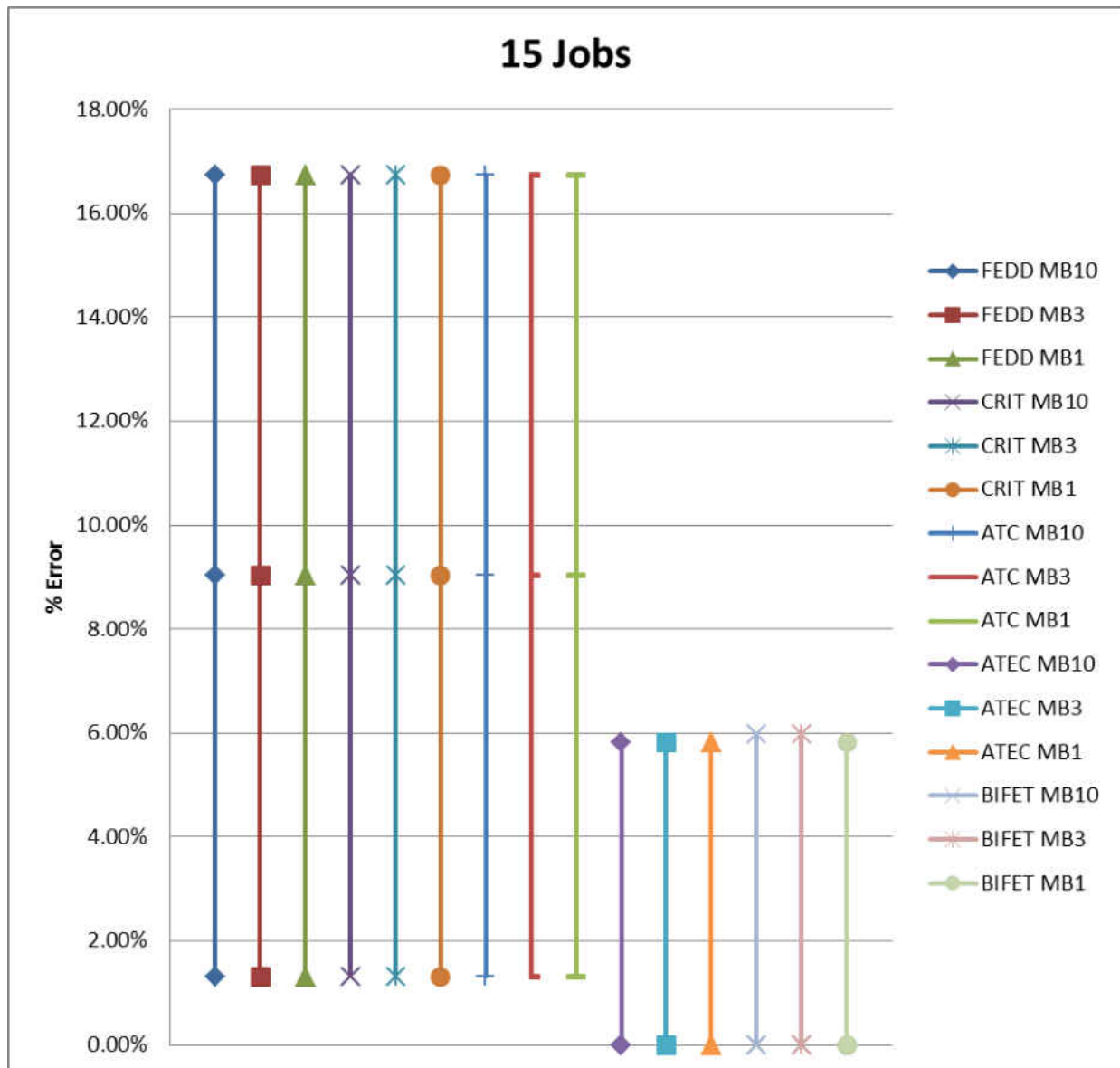


Figure C 15: 95% confidence interval on the performance results for tight due dates, low traffic, 5 families.

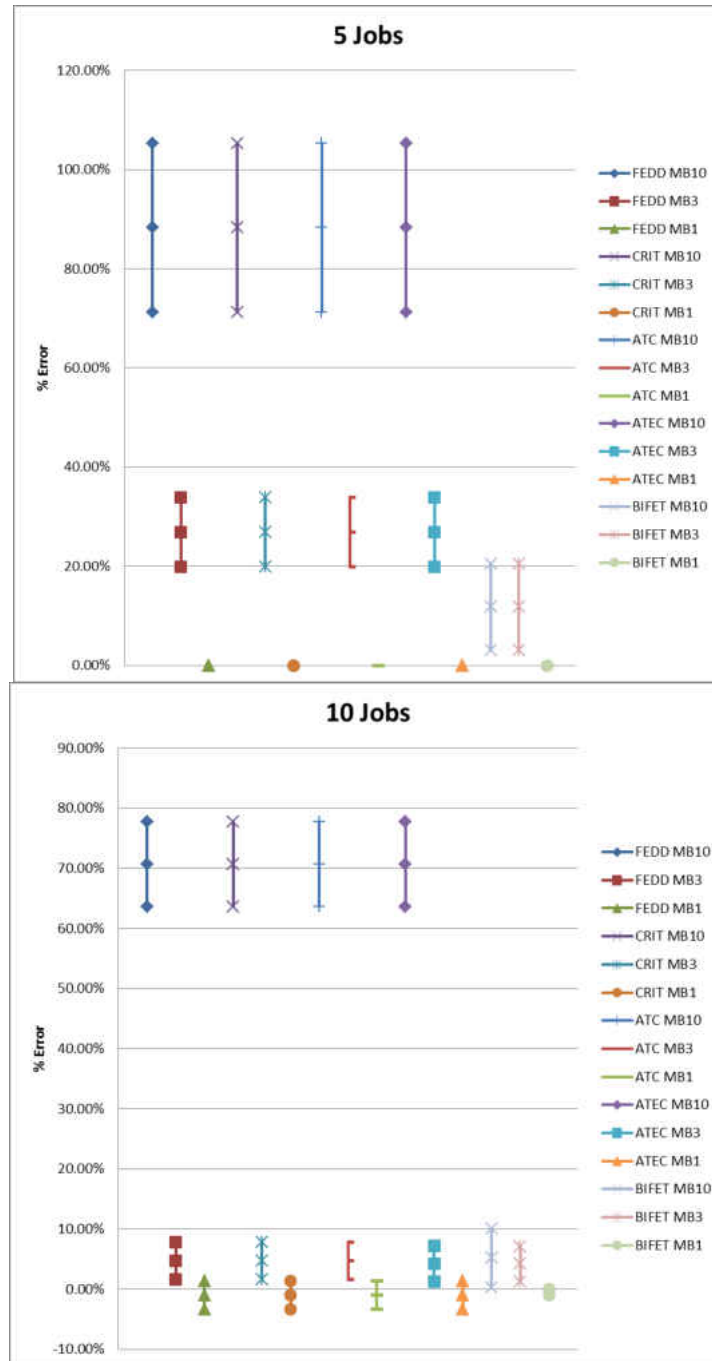


Figure C 16: 95% confidence interval on the performance results for loose due dates, simultaneously, 1 family.

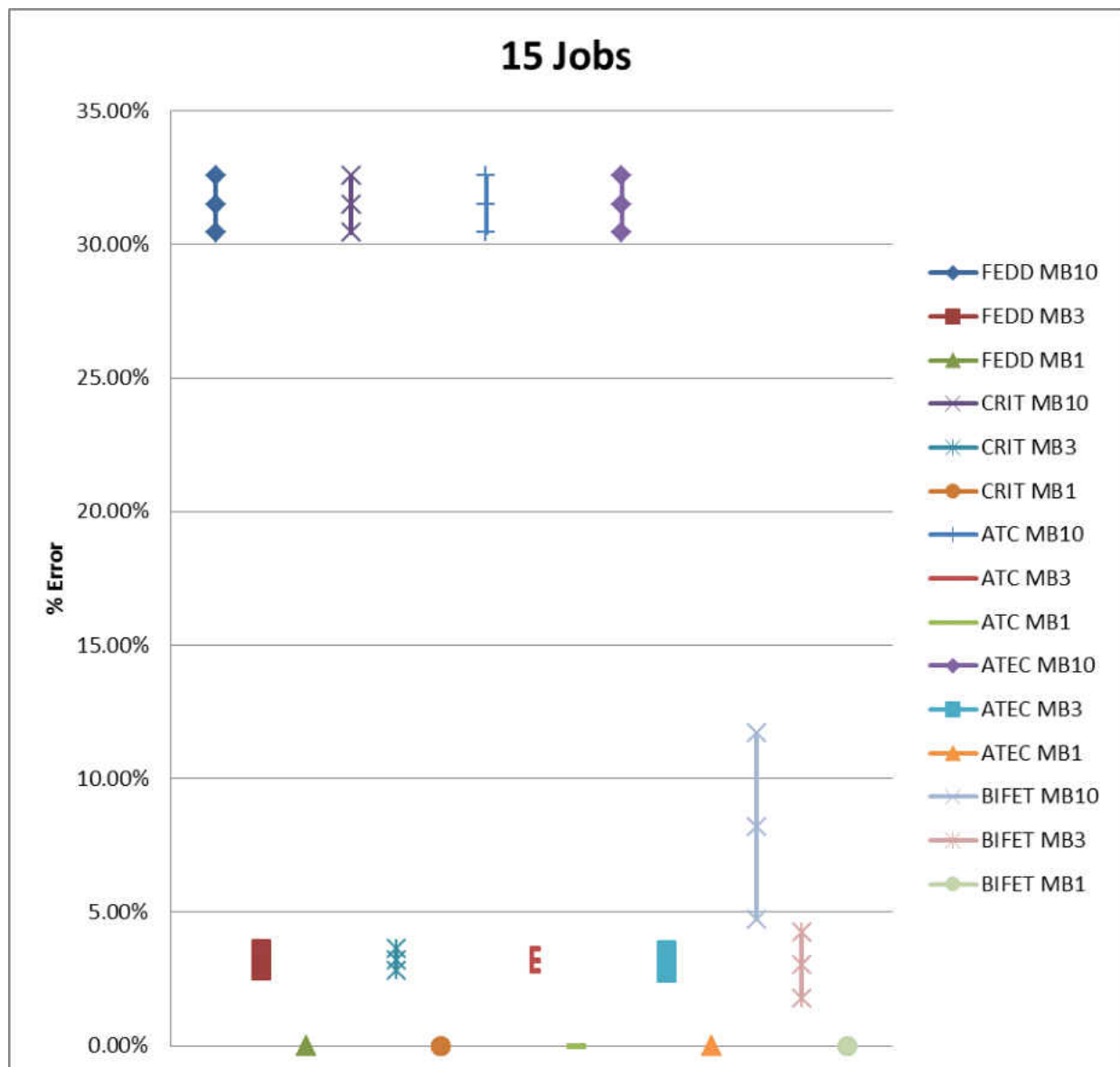


Figure C 17: 95% confidence interval on the performance results for loose due dates, simultaneously, 1 family.

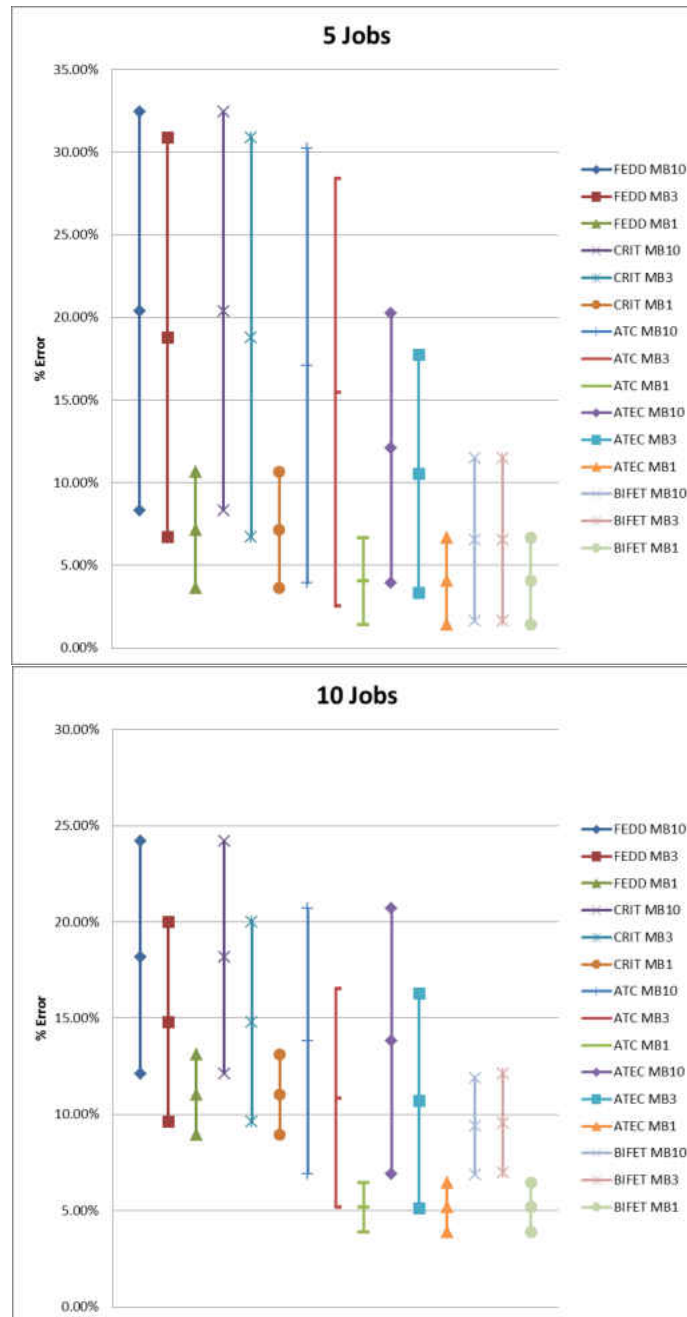


Figure C 18: 95% confidence interval on the performance results for loose due dates, simultaneously, 5 families.

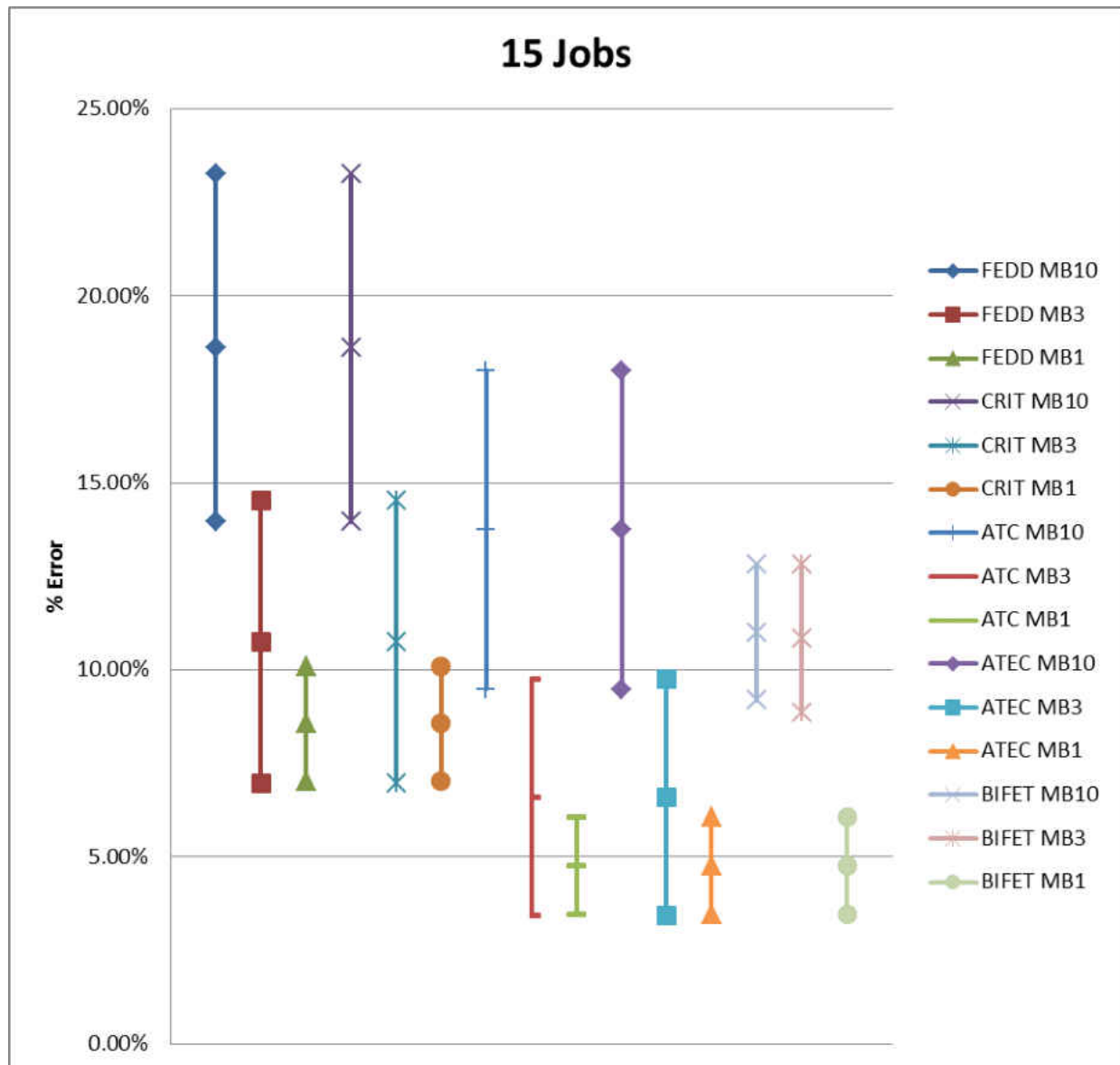


Figure C 19: 95% confidence interval on the performance results for loose due dates, simultaneously, 5 families.

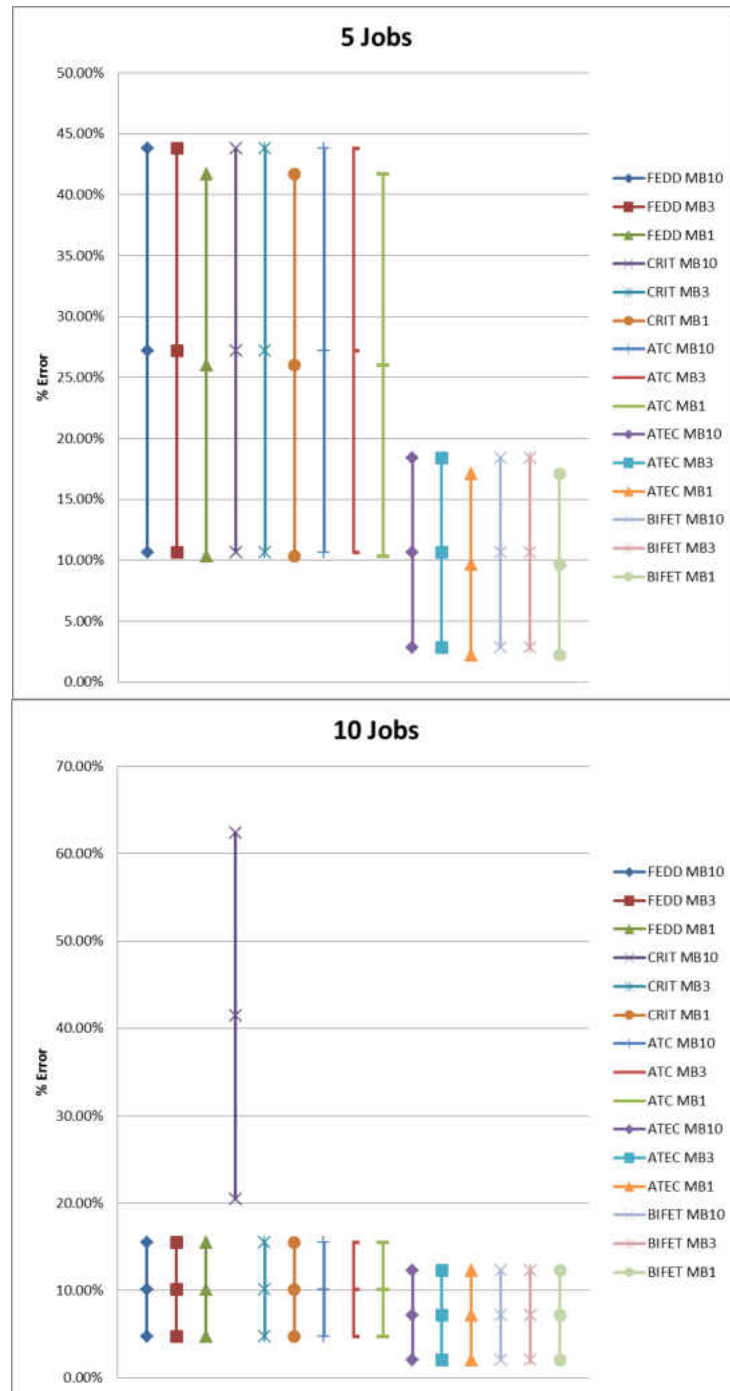


Figure C 20: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

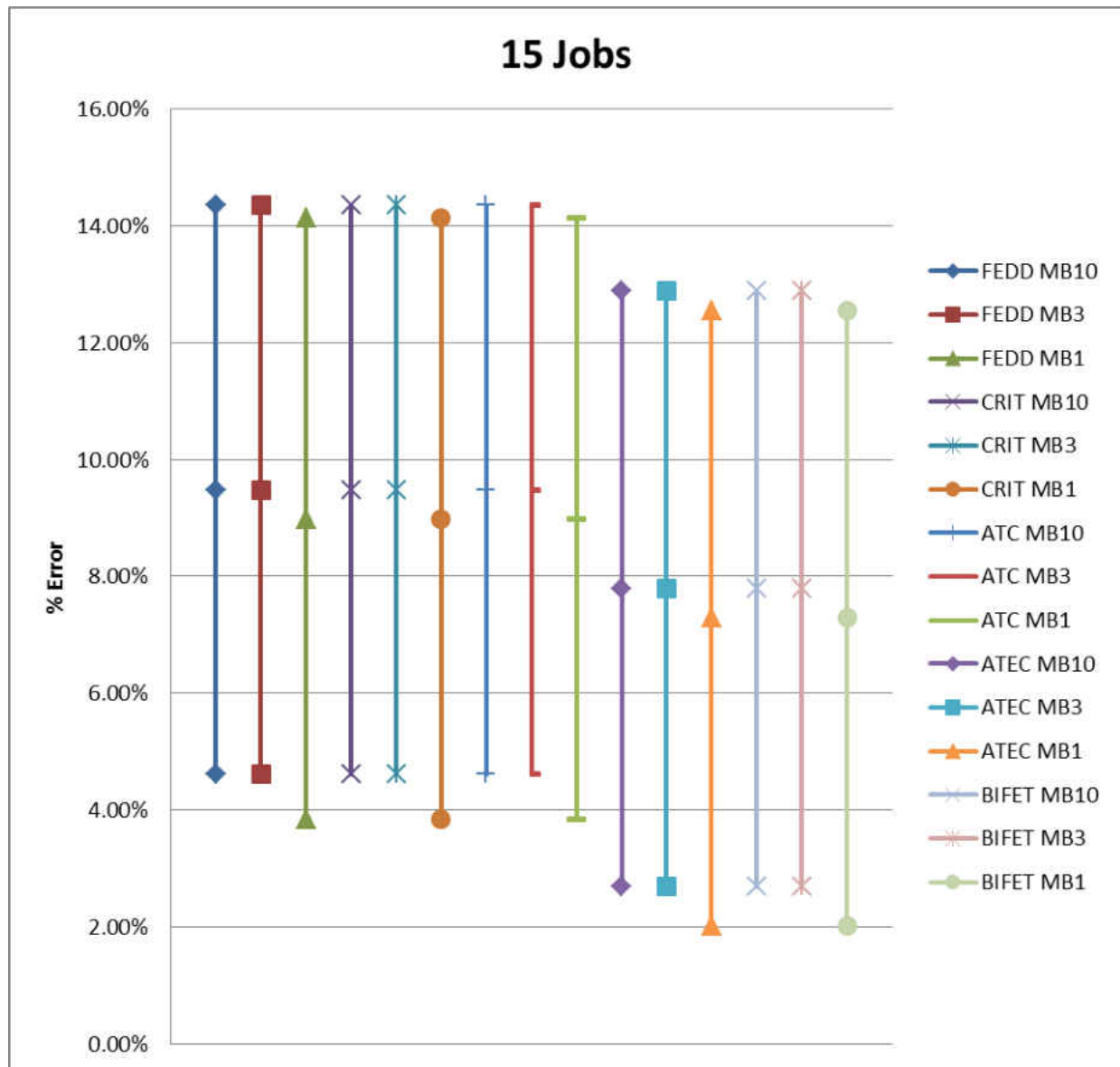


Figure C 21: 95% confidence interval on the performance results for loose due dates, high traffic, 1 family.

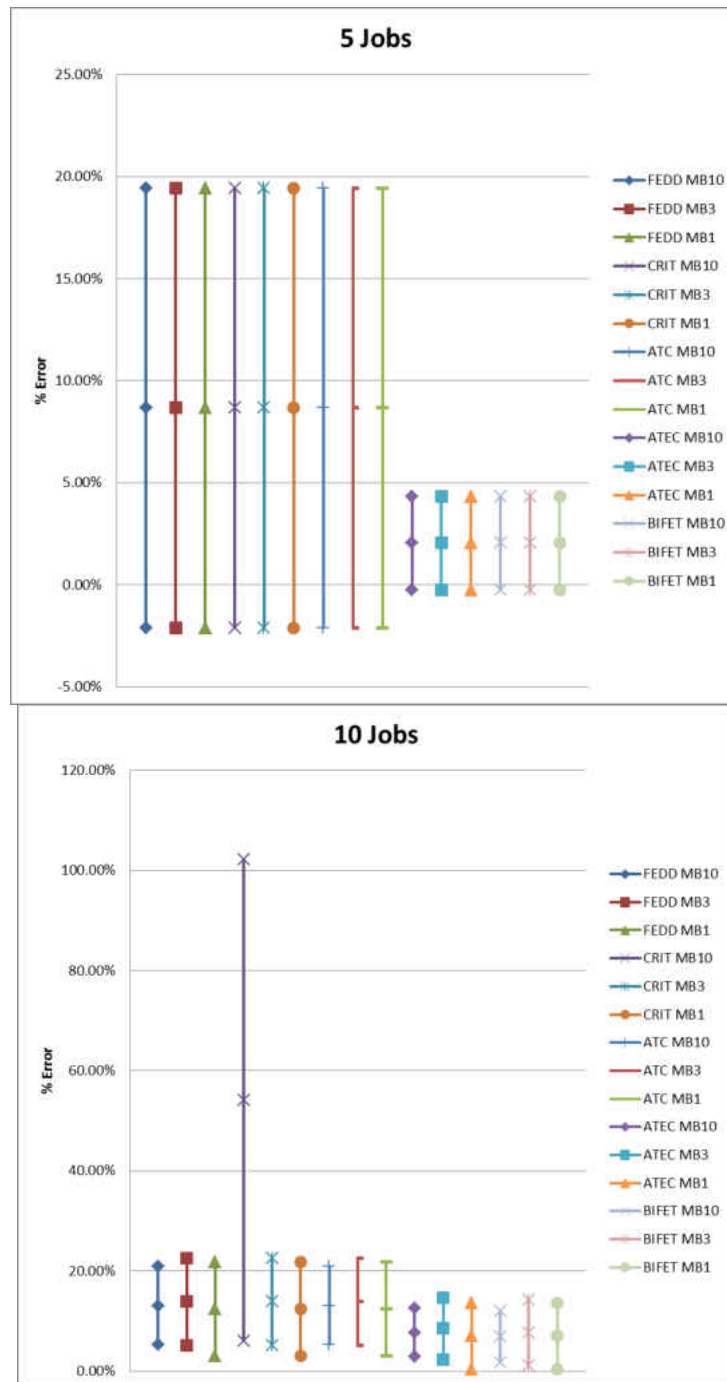


Figure C 22: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

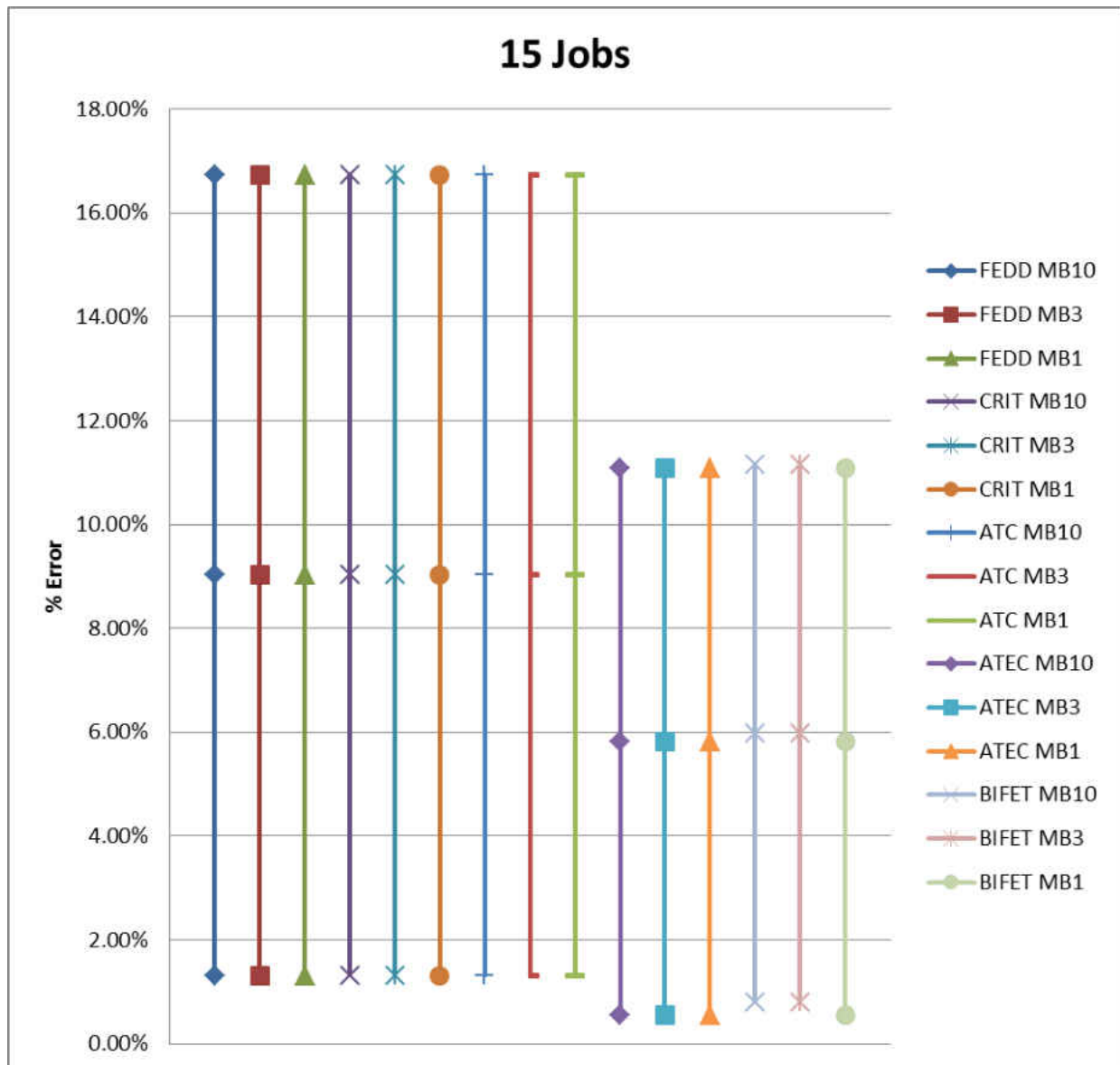


Figure C 23: 95% confidence interval on the performance results for loose due dates, high traffic, 5 families.

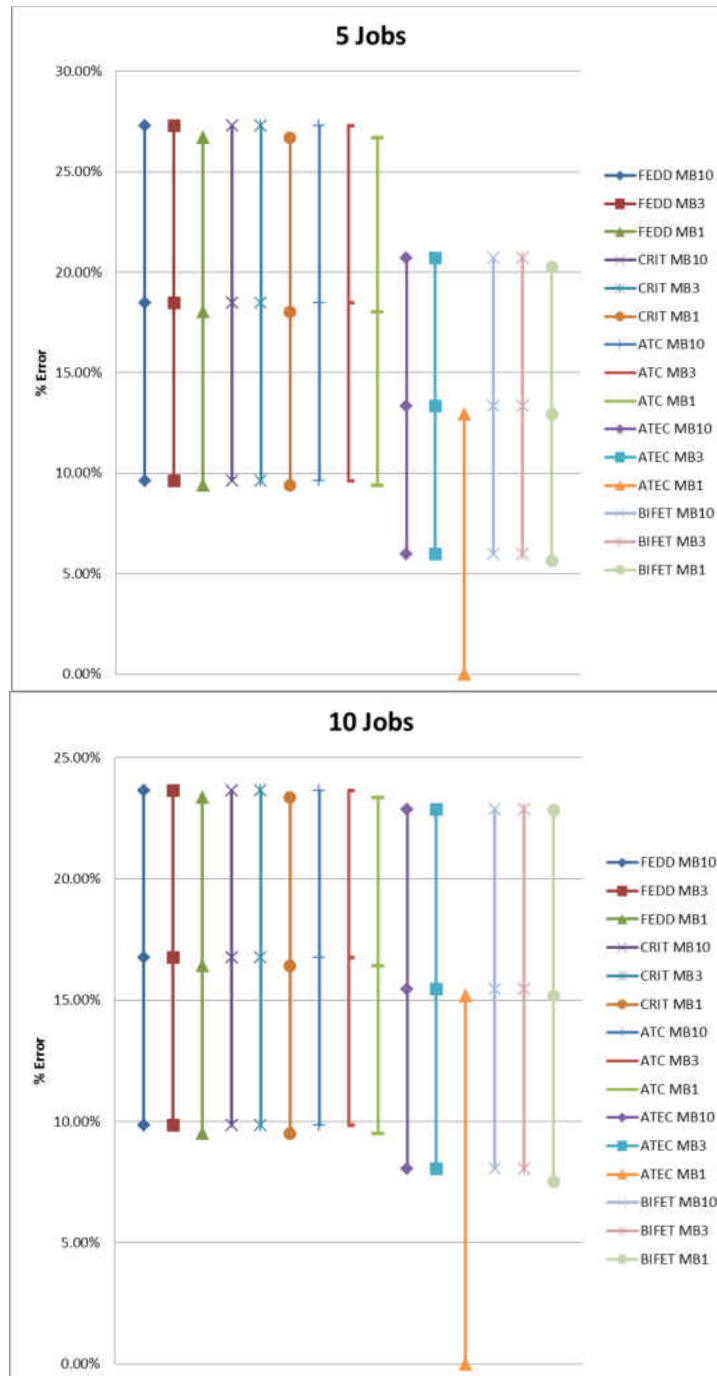


Figure C 24: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

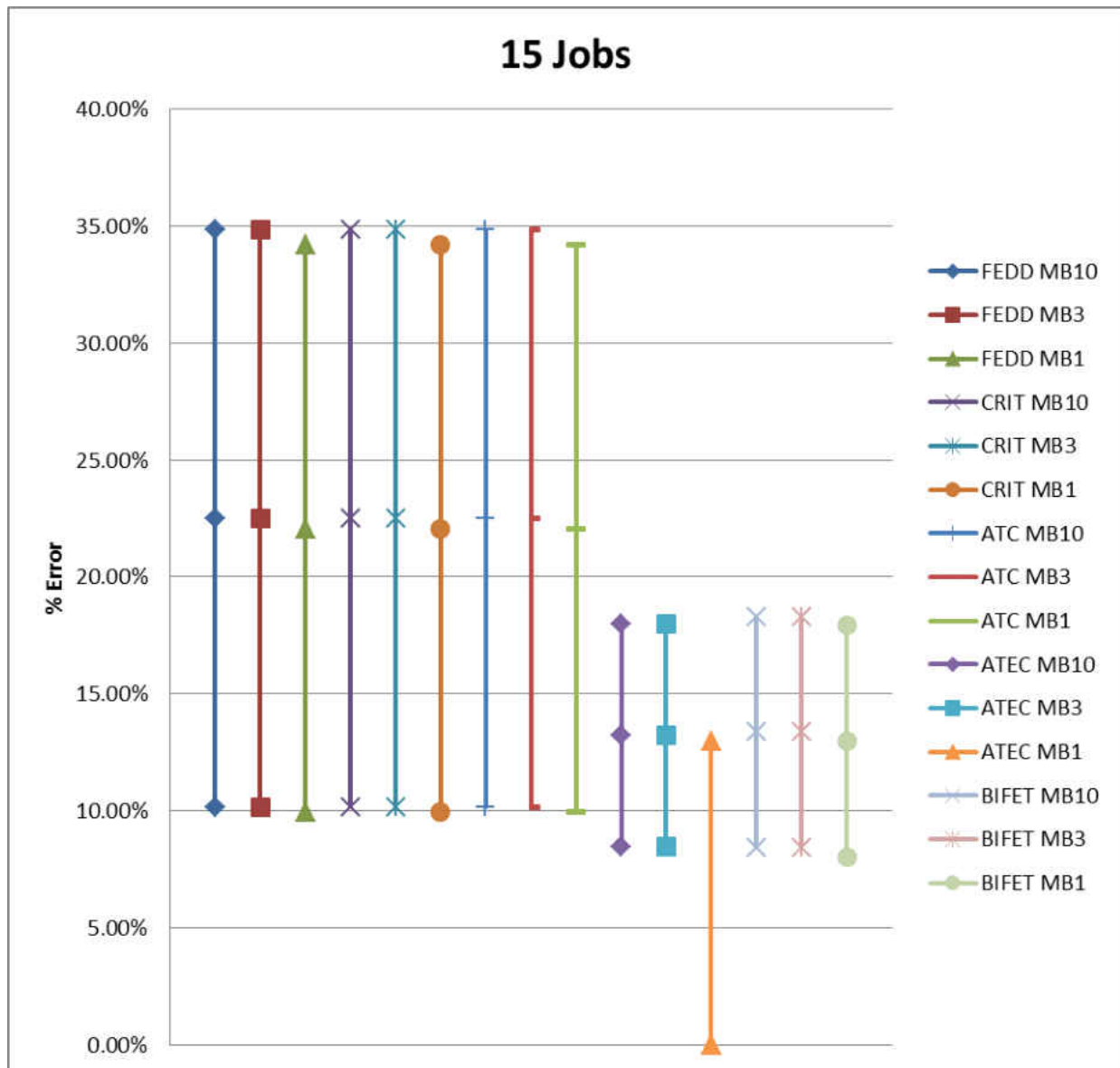


Figure C 25: 95% confidence interval on the performance results for loose due dates, medium traffic, 1 family.

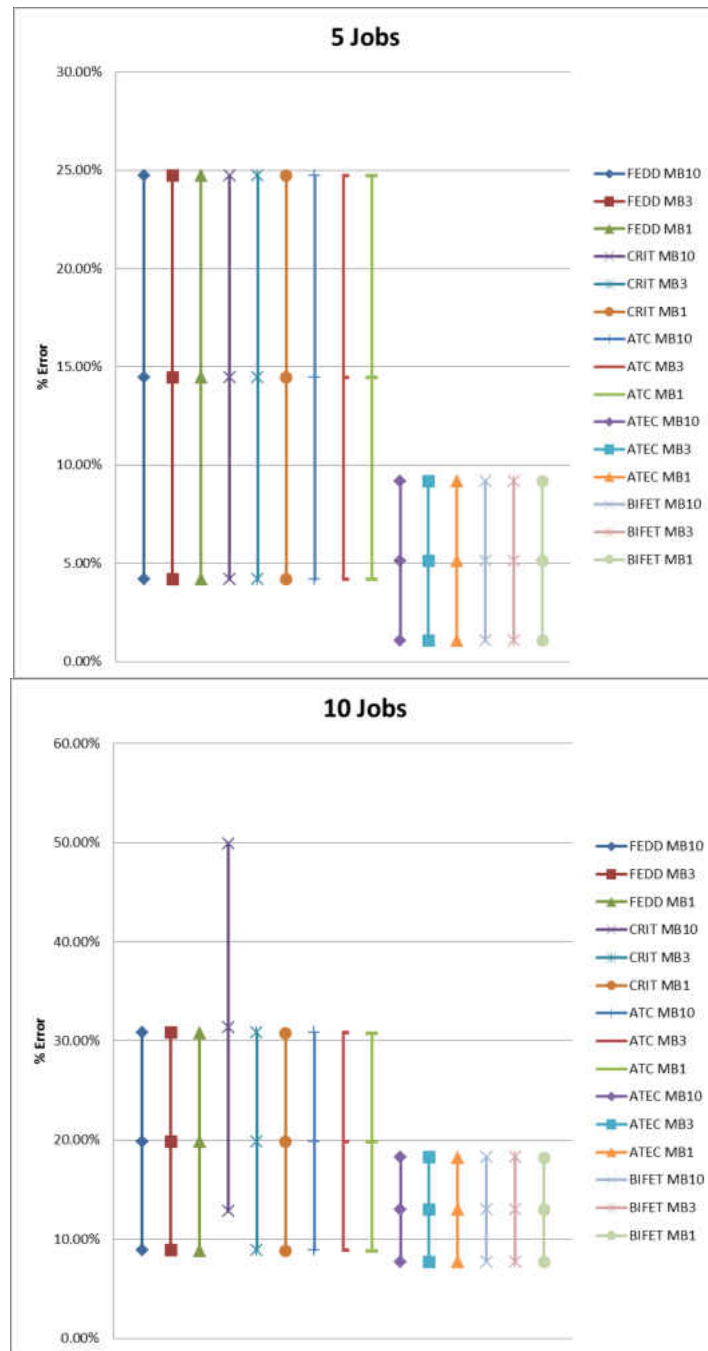


Figure C 26: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families

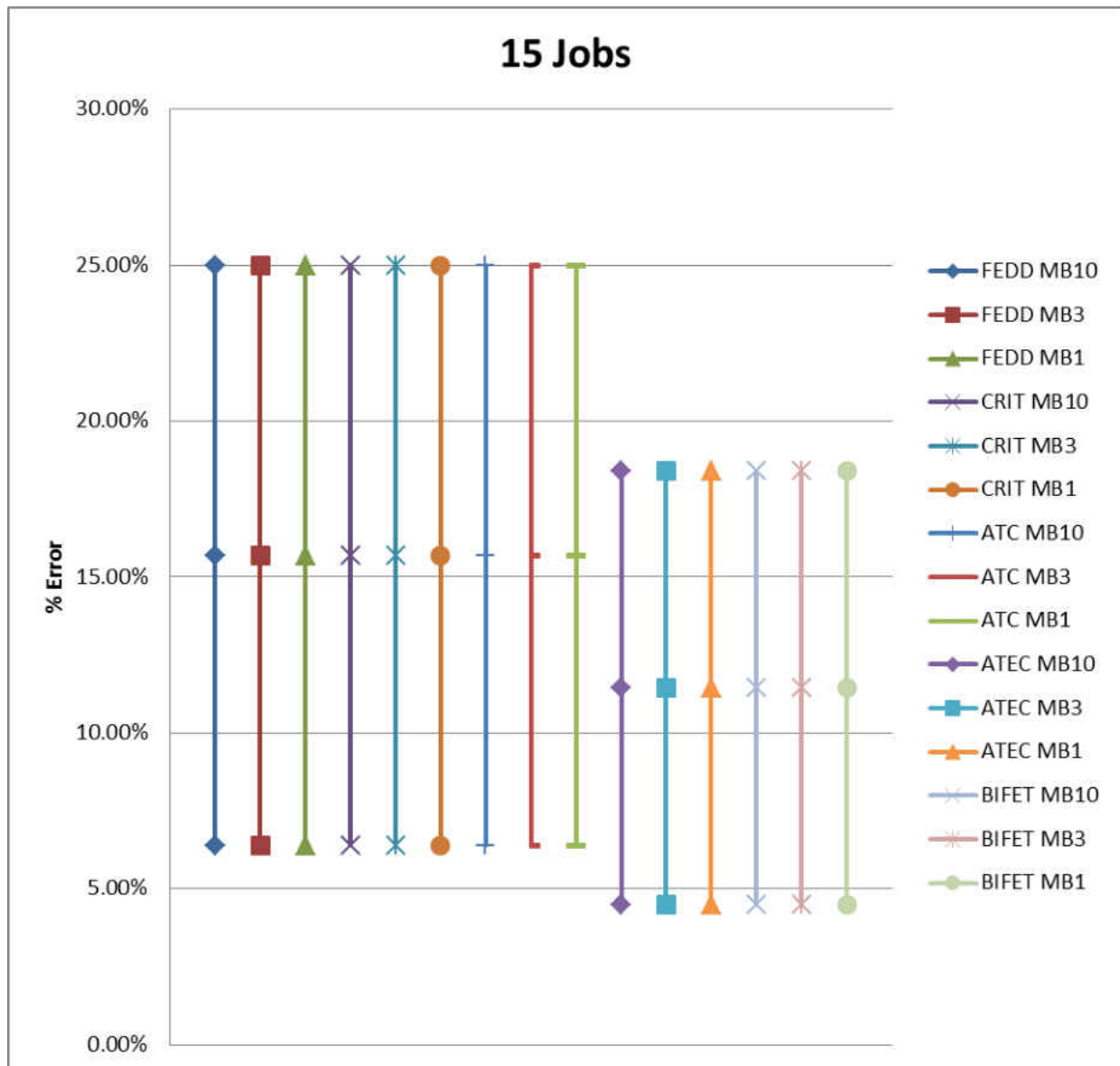


Figure C 27: 95% confidence interval on the performance results for loose due dates, medium traffic, 5 families.

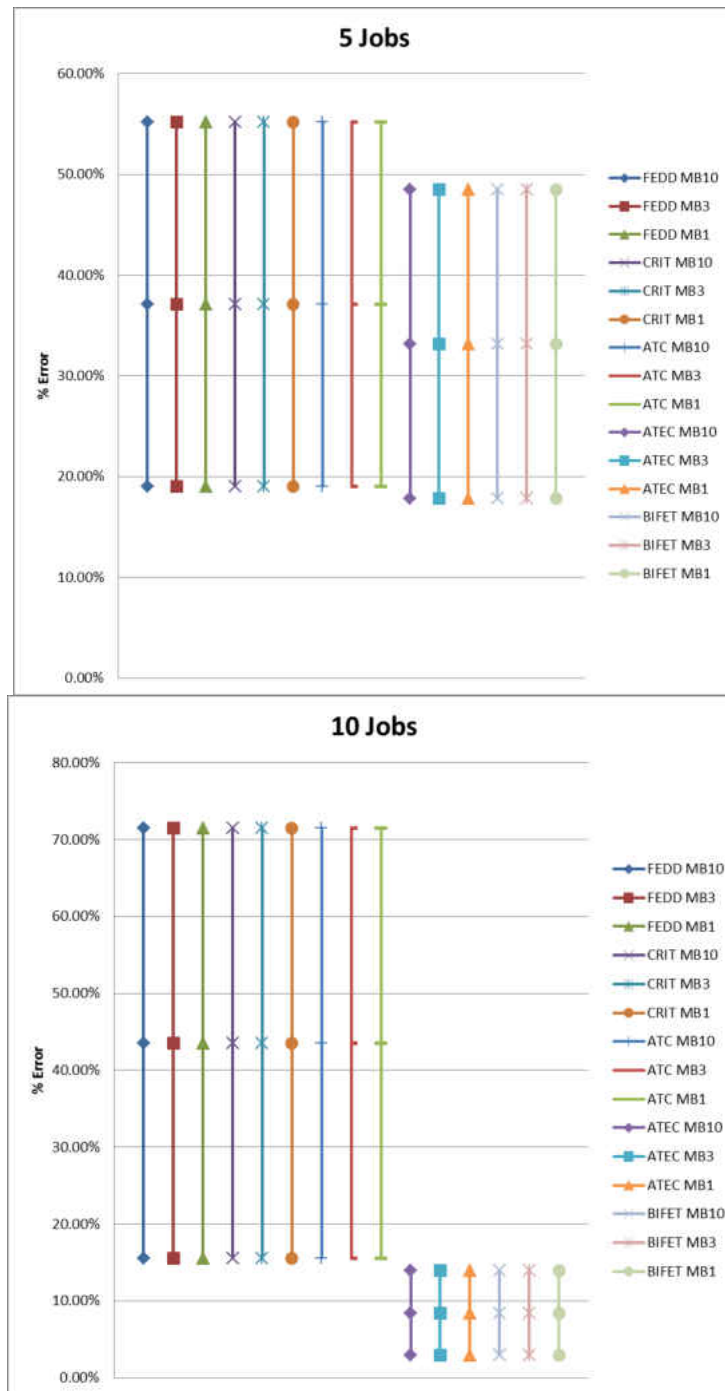


Figure C 28: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

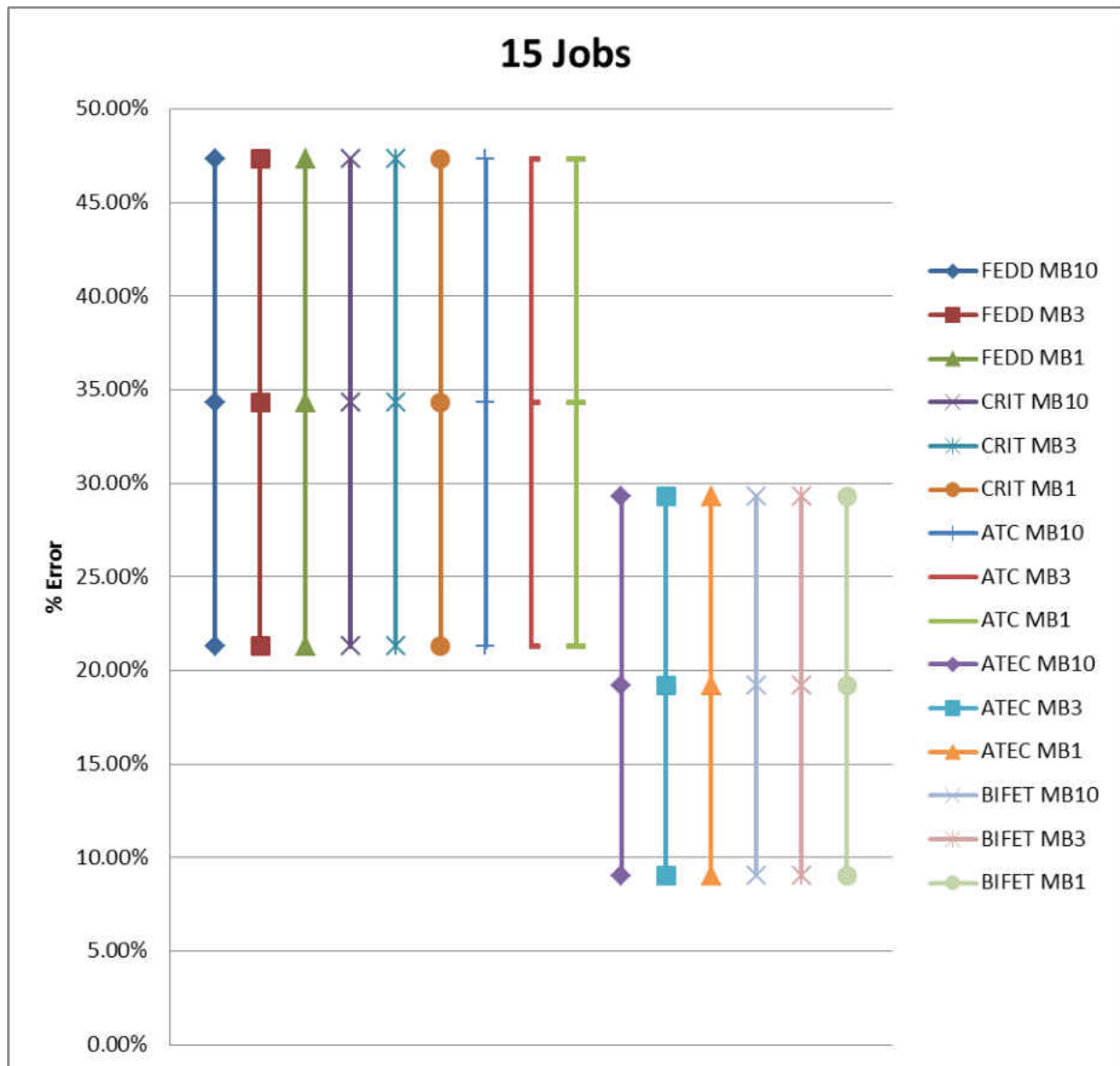


Figure C 29: 95% confidence interval on the performance results for loose due dates, low traffic, 1 family.

LIST OF REFERENCES

1. Ahmed, I. and Fisher, W.W. (1992). Due date assignment, job order release, and sequencing interaction in job shop scheduling. *Decision Sciences* 23, 633-647.
2. Ahmed, I. and Fisher, W.W. (1992). Due date assignment, job order release, and sequencing interaction in job shop scheduling. *Decision Sciences* 23, 633-647.
3. Baek, D.H., Yoon, W.C. and Park, S.C. (1998). A spatial rule adaptation procedure for reliable production control in a wafer fabrication system. *International Journal of Production Research* 36(6), 1475-1491.
4. Baker, Kenneth R. and Scudder, Gary D. (1990). Sequencing with Earliness and Tardiness Penalties: A Review. *Operations Research*, 38(1), 22-36.
5. Balas, E. (1969). Machine scheduling via disjunctive graphs: An implicit enumeration algorithm. *Operations Research* 17, 941-957.
6. Baykasoglu, Adil and Gocken, Mustafa. (2011). A simulation based approach to analyse the effects of job release on the performance of a multi-stage job-shop with processing flexibility. *International Journal of Production Research* 49(2), 585-610.
7. Bean, J.C., Birge, J.R., Mittenthal, J. and Noon, C.E. (1990). Matchup scheduling with multiple resources, release dates and disruptions. *Operations Research*, 39(3), 470-484.
8. Bechte, W. (1994). Load-oriented manufacturing control just-in-time production for job shops. *Production Planning and Control* 5, 292-307
9. Bergamaschi, D., Cigolini, R., Perona, M., Portioli, A. (1997). Order review and release strategies in a job shop environment: a review and a classification. *International Journal of Production Research*, 35(2), 399-420.
10. Bertrand, J.W.M. and Van De Wakker, A. M. (2002). An investigation of order release and flow time allowance policies for assembly job shops. *Production Planning and Control* 13(7), 639-648.
11. Blazewicz, J., Domschke, W., & Pesch, E. (1996). The job shop scheduling problem: Conventional and new solution techniques. *European Journal of Operational Research*, 93, 1-33

12. Bruno, J. and Downey, P. (1978). Complexity of task sequencing with deadlines, set-up times and changeover costs, *SIAM Journal on Computing* 7, 393-404.
13. Cheng, T.C.E, NG, C.T., and Yuan, J.J. (2003). The single machine batching problem with family setup times to minimize maximum lateness is strongly NP-Hard. *Journal of Scheduling* 6, 483-490
14. Cheng, T.C.E., Liu, Z. and Yu, W. (2001). Scheduling jobs with release dates and deadlines on a batch processing machine. *IIE Transactions* 33, 685-690.
15. Chiang, T.C. and Fu, L.C. (2007). Using dispatching rule for job shop scheduling with due date-based objectives. *International Journal of Production Research*, 45(14), 3245-3262.
16. Garey, M.R., Johnson, D.S., and Sethi, R.(1976). The complexity of flowshop and jobshop scheduling. *Mathematics of Operations Research*, 1, 117–129
17. Geiger, C.D. and Uzsoy, R. (2008). Learning effective dispatching rules for batch processor scheduling. *International Journal of Production Research* 46(6), 1431-1454.
18. Gentile, F. and Rogers, K.J. (2009). Order release and dispatching in a sequence dependent job shop. *Management of Engineering & Technology* PICMET.
19. Ghosh, J.B. (1994). Batch scheduling to minimize total completion time. *Operations Research Letters* 16, 271±275
20. Gokhale, R. and Mathirajan, M. (2010). Heuristic algorithms for scheduling of a batch processor in automobile gear manufacturing. *International Journal of Production Research* 29(10), 2705-2728.
21. Gordon, Valery., Proth, Jean-Marie, Chu, Chengbin (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research* 139, 1-25.
22. Gronalt, M. (2002). Work order release and input sequencing for a printed circuit board assembly cell. *Production Planning and Control* 13(7), 591-601.
23. Hariri, A.M.A. and Potts, C.N. (1997). Single machine scheduling with batch set-up times to minimize maximum lateness. *Annals of Operations Research* 70, 75±92.
24. Ishii, N. and Talavage, J.J. (1991). A transient-based real-time scheduling algorithm in FMS. *International Journal of Production Research* 29(12), 2501-2520.

25. Jain, A.S. and Meeran, S.(1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research* 113, 390-434.
26. Jeong, K.C. and Kim Y.D. (1998). A real-time scheduling mechanism for a flexible manufacturing system: using simulation and dispatching rules. *International Journal of Production Research* 36(9), 2609-2626.
27. Kan, A.H.G. Rinnooy. (1976) Machine Scheduling Problems: classification, complexity and computations. Nijhoff, The Hague.
28. Kim, M.H. and Kim, Y.D. (1994). Simulation-based real-time scheduling mechanism in a flexible manufacturing system. *Journal of Manufacturing Systems* 13, 85-93.
29. Koulamas, Christos (1996). Single-machine scheduling with time windows and earliness/tardiness penalties. *European Journal of Operational Research* 91, 190-202.
30. Lee, Y.F., Bhaskaran, K., and Pinedo, M. (1997) A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Transactions* 29, 45-52.
31. Lee, Y.F., Jiang, Z.B., and Liu, H.R. (2009) Multiple-objective scheduling and real-time dispatching for the semiconductor manufacturing system. *Computers and Operations Research* 36, 866-884.
32. Lu, H.L., Huang, G.Q., and Yang, H.D. (2011). Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach. *International Journal of Production Research* 49(3) 647-669.
33. Mathirajan, M. and Sivakumar, A.I. (2006). A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor, *International Journal of Advanced Manufacturing Technology* 29, 990-1001
34. Mazzini, R. and Armentano, Vinicius A. (2001). A heuristic for single machine scheduling with early and tardy costs, *International Journal of Operational Research*, 128, 129-146.
35. Mehta, S.V. and Uzsoy, R. (1998). Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE Transactions* 30, 165-178.
36. Mellor, P. (1966). A review of job shop scheduling. *Operational Research Society* 17(2), 161-171.

37. Melnyk, S., Ragatz, G. and Fredendall, L. (1991). Load smoothing by the planning and order review/release systems: A simulation experiment. *Journal of Operations Management* 10(4), 512-523.
38. Melnyk, S.A. and Ragatz, G.L. (1989). Order Review/Release: Research Issues and Perspectives.” *International Journal of Production Research* 27(7), 1081-1096.
39. Min, H. S. and Yih, Y. (2003). Selection of dispatching rules on multiple dispatching decision points in real-time scheduling of a semiconductor wafer fabrication system, *International Journal of Production Research* 41(16), 3921-3941.
40. Missbauer, H. (1997). Order Release and Sequence-Dependent Setup Times. *International Journal Production Economics* 49 131-143.
41. Monma, C.L. and Potts, C.N. (1989). On the complexity of scheduling with batch setup times. *Operations Research*, 37(5), 798–804.
42. Monma, C.L. and Potts, C.N. (1993). Analysis of heuristics for preemptive parallel machine scheduling with batch setup times. *Operations Research*, 41(5), 981–993.
43. Muth, J. and Thompson, G. (Eds) (1963). *Industrial Scheduling*. Prentice Hall, Englewood Cliffs, NJ.
44. Ow, P. S. (1985). Focused Scheduling in Proportionate Flowshops. *Management Science* 31(7), 852-869.
45. Parthanadee, P. and Buddhakulsomsiri, J. (2010). Simulation modeling and analysis for production scheduling using real-time dispatching rules: A case study in canned fruit industry. *Computers and Electronics in Agriculture* 70, 245–255.
46. Parveen, S. and Ullah, H. (2010). Review on job-shop scheduling using multi criteria decision making. *Journal of Mechanical Engineering*, 41(2), 130-146.
47. Philipoom, P.R., Malhotra, M.K., and Jensen, J.B. (1993) An evaluation of capacity sensitive order review and release procedures in job shops. *Decision Sciences* 24, 1109-1133.
48. Pinedo, M. (2002). *Scheduling: Theory, Algorithms and Systems*, 2nd ed, Prentice Hall, Englewood Cliffs, NJ.
49. Potts, C.N. and Kovalyov, M.Y. (2000). Scheduling with batching: A review. *European Journal of Operational Research* 120, 228-249.

50. Rabadi, G., Mollaghasemi, M., Anagnostopoulos, G. (2004). A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time. *Computers & Operations Research* 31 (1727-1751).
51. Rachamadugu, R. V. (1982). Myopic Heuristics in Job Shop Scheduling, Ph.D. thesis, Graduate School of Industrial Administration, Carnegie-Mellon University.
52. Ragatz, G.L. and Mabert, V.A. (1988). An evaluation of order release mechanisms in a job shop environment. *Decision Sciences* 19, 167-189.
53. Rangsaritratsamee, R., Ferrell, W.G. and Kurz, M.B. (2004). Dynamic rescheduling that simultaneously considers efficiency and stability. *Computers and Industrial Engineering* 46, 1-15.
54. Rodammer, F.A. and White, K.P. Jr. (1988). A recent survey of production scheduling, *IEEE Transactions of Systems, Man and Cybernetics* 18 (6), 841-851.
55. Ruiz, R. and Stutzle, T. (2008). An iterated greed heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research* 187,1143-1159.
56. Sabuncuoglu, I. and Karapinar, H.Y. (2000). A load-based and due-date-oriented approach to order review and release in job shops. *Decision Sciences* 31(2) 413-447.
57. Senties, O. Baez, Azzaro-Pantel, C., Pibouleau, L., and Domenech, S. (2009). A Neural network and genetic algorithm for multiobjective scheduling of semiconductor manufacturing plants, *Industrial and Engineering Chemistry* 48, 9546-9555.
58. Sotskov, Y.N., and Shakhlevich, N.V. (1995). NP-hardness of shop scheduling problems with three jobs. *Discrete Applied Mathematics* 59(3), 237-266.
59. Upasan,i A. and Uzsoy, R. (2008). Integrating a decomposition procedure with problem reduction for factory scheduling with disruptions: a simulation study. *International Journal of Production Research* 46(21), 5883–5905.
60. Uzsoy, R. (1995). Scheduling batch processing machines with incompatible job families. *International Journal of Production Research* 33, 2685-2708.
61. Vepsalainen, ARI P. J. and Morton, Thomas E. (1987). Priority Rules for Job Shops with Weighted Tardiness Costs. *Management Science* 33, No. 8, 1035-1047.

62. Viera, Guilherme E., Herrman Jeffrey, W., Lin, Edward. (2003) Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling* 6: 39-62.
63. Vinod, V. and Sridharan, R. (2009). Simulation-based metamodels for scheduling a dynamic job shop with sequence-dependent setup times. *International Journal of Production Research* 47(6), 1425–1447.
64. Wan, Guohua and Yen, Benjamin P.C. (2002). Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties, *International Journal of Production Research* 142, 271-281.
65. Wu, S.D., Byeon, E. S., and Storer, R.H. (1999). A graph-theoretic decomposition of the job shop scheduling problem to scheduling robustness, *Operations Research* 47(1), 113-124.
66. Zhang, H. and Gu, M. (2009). Modeling job shop scheduling with batches and setup times by timed Petri nets. *Mathematical and Computer Modeling* 49(1-2), 286-294.
67. Zhu, X. and Wilhelm, W.E. (2006). Scheduling and lot sizing with sequence-dependent setup: A literature review, *IIE Transactions* 38, 987-1007.