DYNAMIC CONTROL OF SERIAL-BATCH PROCESSING SYSTEMS

A Dissertation

by

ABDULLAH CEREKCI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2008

Major Subject: Industrial Engineering

DYNAMIC CONTROL OF SERIAL-BATCH PROCESSING SYSTEMS

A Dissertation

by

ABDULLAH CEREKCI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Amarnath Banerjee
Committee Members,	Brett A. Peters
	Don T. Phillips
	Ricardo Gutierrez-Osuna
Head of Department,	Brett A. Peters

December 2008

Major Subject: Industrial Engineering

ABSTRACT

Dynamic Control of Serial-Batch Processing Systems. (December 2008)

Abdullah Cerekci, B.S., Bilkent University

Chair of Advisory Committee: Dr. Amarnath Banerjee

This research explores how near-future information can be used to strategically control a batch processor in a serial-batch processor system setting. Specifically, improved control is attempted by using the upstream serial processor to provide near-future arrival information to the batch processor and further meet the re-sequencing requests to shorten critical products' arrival times to the batch processor. The objective of the research is to reduce mean cycle time and mean tardiness of the products being processed by the serial-batch processor system.

This research first examines how mean cycle time performance of the batch processor can be improved by an upstream re-sequencing approach. A control strategy is developed by combining a look-ahead control approach with an upstream re-sequencing approach and is then compared with benchmark strategies through simulation. The experimental results indicate that the new control strategy effectively improves mean cycle time performance of the serial-batch processor system, especially when the number of product types is large and batch processor traffic intensity is low or medium. These conditions are often observed in typical semiconductor manufacturing environments.

Next, the use of near-future information and an upstream re-sequencing approach is investigated for improving the mean tardiness performance of the serial-batch processor system. Two control strategies are devised and compared with the benchmark strategies through simulation. The experimental results show that the proposed control strategies improve the mean tardiness performance of the serial-batch processor system.

Finally, the look-ahead control approaches that focus on mean cycle time and mean tardiness performances of the serial-batch processor system are embedded under a new control strategy that focuses on both performance measures simultaneously. It is demonstrated that look-ahead batching can be effectively used as a tool for controlling batch processors when multiple performance measures exist.

ACKNOWLEDGEMENTS

This dissertation could not have been completed without the guidance and encouragement of Dr. Amarnath Banerjee, Dr. Brett Peters, Dr. Don Phillips and Dr. Ricardo Gutierrez-Osuna. I would like to thank all members of my committee for their assistance. I specially wish to thank my mentor, Dr. Banerjee, for keeping me focused and for his kindness throughout my graduate studies. I also would like to thank Dr. Richard Feldman for inspiring me with his simulation teaching.

There are many other people at Texas A&M University who have helped me stay motivated. I would like to thank Dr. Bikram Sharda, Dr. Reha Botsali, Dr. Bahadir Aral, Dr. Onur Bakir and Yasin Alan for their friendship and help over the years. I also would like to thank Dr. Salih Yurttas, Dr. Tahir Cagin and Dr. Ercan Sirakaya for their valuable advises throughout my graduate studies.

On a more personal note, I would like to thank my parents, Ayten and Bayram Cerekci, and my sisters, Rana Cerekci and Sule Ozevren, for keeping me always positive and supporting my career decisions. Finally, a special thanks to my fiancée Hande Kocak for her patience and faith in me. Without her support, understanding and love, this dissertation would not have been completed.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

CHAPTER I

INTRODUCTION

The semiconductor industry has grown tremendously in recent years due to the increasing number of products, ranging from personal computers to cellular phones, where integrated circuits (IC) are used. Parallel to this market growth, interest in IC fabrication technology and methods has steadily increased among researchers and practitioners. There are five steps in semiconductor manufacturing: wafer fabrication, wafer probe, device assembly, class test, and then a final test. Wafer fabrication is the most capital intensive step and control of wafer production is a well-known complex problem. The complexity is due to several factors: wafers go through a large number of processing steps in production routes; the process itself has complexities such as re-entrant flow structure and sequence-dependent setup times; the product-mix is highly diverse; and the production flow is highly variable.

Batch processors, where a number of products can be processed simultaneously in a batch, are encountered in semiconductor front-end (wafer fabrication) and semiconductor back-end (final testing). By inducing large WIP increases and decreases, these processors are major sources of variation because of resulting non-smoothness in the production flow. In many cases, the processing times of these processors are quite long compared to those of serial processors where products are processed one at a time.

This dissertation follows the style of *IIE Transactions*.

The long processing times have profound effects on overall production performance. This makes effective control of these processors an important management concern. Some examples of such processors are diffusion furnaces in wafer fabrication and burn-in ovens in the final testing stage.

The purpose of this dissertation is to study the long-run control of a batch processor that is operating in front-end semiconductor manufacturing. Specifically, this research explores the use of near-future arrival information available at the batch processor's upstream station in batch process decision making. The focus is on a set of performance measures that have high priority from management's perspective. The objective is to improve the quality of decision making in the batch processor station by developing and testing a set of control strategies that incorporate product information and participate in the sequence decisions at the upstream station.

1.1 Overview of the problem

The problem addressed in this research is strongly motivated from the diffusion furnaces that are present in front-end semiconductor manufacturing. In the diffusion operation, wafers are processed in standard lots (products) and it is possible to process a number of lots together as a batch. Once processing of a batch has been initiated, no products can be removed or added to the batch and the process is uninterruptable. Due to the chemical nature of the process, it is impossible to batch products with different recipes together. Products with the same recipe require the same processing times, and can be viewed as a product type. Consequently, these product types are incompatible since different product types cannot be processed together. Batch sizes are limited by the capacity of the furnace, typically varying between 6 products and 8 products, depending on the product type. The recipes are controlled by a computer program at the furnace, and require constant processing times which are independent of the number of products in the batches. Furnace process times are typically about 5 to 10 times longer than the process times of the serial (discrete) operations. Therefore, effective control of these operations is critical for the overall performance of the wafer fabrication facility.

If a batch has as many products as the capacity of the batch processor, the batch is called a “full batch”. On the other hand, if the number of products in the batch is less than the batch processor capacity, the batch is called a “partial batch”. Whenever the batch processor becomes available, if there is a full batch product type, there will be no benefit in waiting to start the batch process. In this case, the only decision making is the selection between full batch product types. However, the problem lies in the control of the batch processors, in that whenever a batch processor becomes available and there are only partial batches, a non-trivial decision must be made to either process one of the partial batches or wait for additional products to arrive. Since the batch processor serves incompatible product types, the decision alternatives include selecting the most appropriate product type to load now or waiting for a particular product type. Decision making becomes very complicated when the number of product types is large.

The rule sets used in this decision making process are referred to as dynamic control strategies. A dynamic control strategy reviews the batch processor at decision points and makes a decision based on the underlying rules. A decision point is defined as the time that the batch processor becomes available or the time an arrival occurs while the batch processor is idle. Most wafer fabrication facilities have sophisticated shop floor control systems, which provide high visibility of events occurring at each processing station. The availability of these systems provides accurate near-future information for the batch processing stations. A number of dynamic control strategies make use of this near-future information in the control of batch processors. Under these strategies, each alternative decision is evaluated for the performance measure of interest, using future information that lies within the time horizon affected by the decision. The time horizon of a decision alternative is the time-window starting at the current decision point and ending at the time that the batch processor becomes available after executing the considered decision alternative. This evaluation process is referred to as “look-ahead batching” and is the main focus of this dissertation.

There are mainly two inter-related groups of objectives that receive high attention from the management perspective in semiconductor manufacturing. The first group of objectives is related to the cycle time of products, which represents the degree of manufacturer’s performance. Most industries are driven by the cycle time consideration of their products, since short cycle times give a competitive advantage in the form of delivering products in shorter times, as well as quickly responding to changes in market demand. In addition to this, specifically in semiconductor manufacturing, long cycle times are highly undesirable because the process yields are inversely proportional to the amount of time wafers spend in production. This is due to the fact that the more time wafers spend in fabrication; the more likely they are to become contaminated. The second group of objectives is related to the on-time delivery of products, which measures the degree of customer satisfaction. Companies that ensure on-time delivery generally have a better chance of retaining customers and receiving subsequent orders based on their previous performance.

In this dissertation, the long-run control of a batch processor is explored with cycle time and due-date related objectives accounting for concerns from the managerial perspective. Since the focus is on long-run performance, “mean cycle time” and “mean tardiness” of the products that are served by the batch processor are considered. These performance measures are selected over other possible measures due to their prevalent use by semiconductor manufacturing management as production performance indicators in the long-run (month/quarter) (Pfund et al. (2006)).

The research domain of this dissertation assumes that the near-future information coming from the upstream stations of the batch processor station is accurate and available for batch process decision making. In order to model the near-future information, an upstream serial processor station is attached to the batch processor station. In the serial-batch processor system, the serial processor serves for the control of the batch processor with two main contributions. First, the serial processor station provides the serial process time, batch process due-date and the sequence position information of the products that are waiting in its queue. Second, the serial processor station allows a batch control strategy that incorporates the re-sequencing activities in the serial processor’s queue to improve the performance of the batch processor.

The assumptions relating to the serial-batch processor system are as follows: There are N incompatible product types being processed by the serial-batch processor system. Batch process times are constant, specific to product types and independent of the number of products in the batches. The capacity of the batch processor is specific to product types. There is a single overall inter-arrival time distribution for the serial processor and the product type of a given arrival is assigned probabilistically depending on the product-mix values. The arrival rate at the serial processor and the arrival rate at the batch processor are the same. The service rate of the serial processor is determined by a fixed utilization level chosen for the serial processor. The serial process times are determined stochastically by the service time distribution of the serial processor. The serial process times and batch process due-dates of the products and the sequence information in the serial processor's queue are available to a controller attached to the serial-batch processor system.

There are several process, product and processor characteristics that may influence the way that a control strategy operates on the serial-batch processor system. These include the number of product types, the product-mix, the traffic intensity of the batch processor, the capacity of the batch processor for product types, and the batch process times for product types. The influence of these characteristics on the performance of the batch process control is also a part of this dissertation's focus.

1.2 Research objectives

There are 3 major objectives of this research, and they are as follows:

i) Develop a control strategy for the serial-batch processor system that will effectively utilize information from the serial processor station and incorporate the sequence decisions in the serial processor's queue in minimizing the time that products spend after they enter the serial processor station until they are loaded to the batch processor. Compare the control strategy's performance with benchmarks in the look-ahead batch control literature.

ii) Develop two control strategies for the serial-batch processor system where:

- The first control strategy will utilize the information on the serial processor station in batching decisions to minimize the tardiness of the products at the end of the batch process.
- The second control strategy will further incorporate the sequence decisions in the serial processor's queue to minimize the tardiness of the products at the end of the batch process.

Compare the performances of the control strategies with benchmarks in the look-ahead batch control literature.

iii) Combine the control strategies proposed for the single criterion control problems in (*i*) and (*ii*) above under a new control strategy that will utilize the information on the serial processor station in minimizing simultaneously the waiting times of the products in the batch processor's buffers and the tardiness of the products at the end of the batch process. Compare the performance of the control strategy with the modified well-known control policies for the bi-criteria problem.

1.3 Significance of the research

The batch process control problem is often converted to a machine scheduling problem by assuming that the long-run future data is fully available and deterministic. The scheduling domain focuses mainly on the static version of this problem rather than the dynamic case where future arrivals are allowed. This is due to additional complexity coming with the dynamic case, since the static problem is itself very complex because of the constraints of incompatible product types and different batch process times. The static problem has been shown NP-hard for total completion time (Chandru et al. (1993a)), makespan (Uzsoy (1994)) and total tardiness (Mehta and Uzsoy (1998)) criteria. The dynamic problem has been shown NP-hard for makespan (Liu and Yu (2000)), maximum tardiness (Li and Lee (1997)) and total tardiness (Tangudu and Kurz (2006)) criteria. Due to the high complexity of these scheduling problems, optimal solution procedures are not much better than complete enumeration and therefore suffer from a computational burden. These solutions are not practical for use in real-time or near real-time settings. Therefore, the main focus in literature is on finding heuristic solution procedures.

However, little information on future arrivals can be obtained accurately from shop floors because the level of stochasticity increases in the problem data with the length of the scheduling horizon. Therefore, in this dissertation, the problem is considered from the dynamic control point of view and the decisions are limited to whether to start the batch process with one of the product types or keep the processor idle until the next decision point. This dynamic decision making utilizes near-future information for evaluating alternative decisions. This problem domain is known as look-ahead batch control, and the future information required for the look-ahead control strategies is bounded by two times the length of the batch process time. This is from the fact that only those future arrivals expected to occur within a batch process time-window can influence the batch processor to make a wait decision. Additionally, the evaluation of starting the batch process at a particular arrival point needs a future time-window with a length equal to the batch process time.

Although look-ahead batch control has been extensively studied in the literature for cycle time related performance measures, there are only a few studies that explore its use for due-date related performance measures. Also, none of these look-ahead control strategies embed upstream control (i.e., re-sequencing decisions on the upstream station) with the control of batch processors.

In this dissertation, new control strategies that address these issues to extend the usability and the effectiveness of look-ahead batch control are proposed. These proposed control strategies effectively use the near-future information and improve the performance measures of interest. The algorithms run in O(N) and O(N^2) complexity with the number of product types, N, and can be implemented easily in a wafer fabrication facility for on-line control of batch processors.

1.4 Organization of the dissertation

The remainder of this dissertation is organized into five chapters. Chapter II summarizes the literature that is relevant to the control of batch processors. Chapter III discusses the combination of look-ahead batching and upstream re-sequencing in the control of batch processors with mean cycle time performance measure. The use of look-ahead batching and upstream re-sequencing in the control of batch processors with mean tardiness performance measure is presented in Chapter IV. Chapter V demonstrates the extension of look-ahead control strategies developed in Chapters III and IV for the bi-criteria control of batch processors where mean cycle time and mean tardiness performance measures are considered together. The contributions of the dissertation and future research directions are summarized in Chapter VI.

CHAPTER II

LITERATURE REVIEW

An extensive literature review for the batch process control problem is provided in this chapter. The literature is grouped under three sections with respect to problem objectives relating to this dissertation. The first and the second sections present the previous research focusing on cycle time related objectives and due-date related objectives respectively. The third section summarizes the previous research addressing multiple criteria batch process control. The literature is grouped further with respect to the nature of the product flow (whether dynamic or static) and the availability of the problem data (whether deterministic or stochastic) in each section. In static problems, all products are ready at time zero while dynamic problems consider an arrival process in which the ready times of products are different. In dynamic problems, three cases are considered in the literature: full knowledge on future arrivals (full deterministic), availability of near-future arrival information (stochastic + deterministic) and no future arrival information (full stochastic).

Early research on the control of batch processors can be found mainly in queueing theory but most of these papers focus on performance evaluation rather than control of batch processors. A recent paper by Mathirajan and Sivakumar (2006a) provides a detailed review of the literature.

2.1 Batch process control with cycle time related objectives

Table 2.1 provides a matrix of the literature according to the availability of future information and the nature of the product flow.

Table 2.1. List of literature on cycle time related objectives

Availability of the future info Nature of the product flow	No future arrival information	Full knowledge on future arrivals	Near-future arrival information is available
Dynamic	Neuts (1967), Deb and Serfozo (1973), Gurnani et al. (1992) Duenyas and Neale (1997), Avramidis et al. (1998), Akcali et al. (2000), Neale and Duenyas (2000), (2003)	Uzsoy (1995), Lee and Uzsoy (1999), Liu and Yu (2000), Sung et al. (2002), Cheraghi et al. (2003)	Glassey and Weng (1991), Fowler et al. (1992), (2000) Weng and Leachman (1993), Robinson et al. (1995), Van Der Zee et al. (1997), (2001), (2002), (2007) Solomon et al. (2002), Cigolini et al. (2002)
Static	Ahmadi et al. (1992), Chandru et al. (1993a), (1993b), Uzsoy (1994), Hochbaum and Landy (1997), Ghazvini and Dupont (1998) Dobson and Nambimadom (2001), Uzsoy and Yaoyu (1997), Azizoglu and Webster (2001), Kim and Kim (2002), Dupont and Dhaenens-Flipo (2002)		

2.1.1 Static problem domain with cycle time related objectives

In the static domain, full problem data is assumed to be available and therefore problem relates to deterministic machine scheduling. The batch processor follows a no-idling schedule since any delay between two batch processes worsens the objective of the problem. Consequently, the tasks of the problem are limited to how to form the batches and how to sequence already formed batches.

Chandru et al. (1993b) examine the problem of minimizing total completion time on a batch processor with compatible product types. They propose a branch and bound algorithm that eliminates an important percent of the batching alternatives. However, the computational complexity of the algorithm limits the range of the problems that are solvable. Therefore, they present two heuristic procedures for practical purposes. In an extension to their research, Chandru et al. (1993a) show that if products can be partitioned into categories such that process times of the products in the same category are the same, the problem of minimizing total completion time can be solved in polynomial time. Their solution procedure is based on dynamic programming, and its complexity is in the form of $O(N^3B^{N+1})$ where N is the number of product categories and B is the batch processor capacity. For the same problem, Hochbaum and Landy (1997) provide a more efficient heuristic solution that has a complexity in the form of $O(N^2 3^N)$. Uzsoy (1994) extends the problem for the case of products having different sizes and accordingly different capacity requirements. He proves that the problems of minimizing total completion time and makespan are both NP-hard. Consequently he proposes heuristic solution procedures for both problems. Ghazvini and Dupont (1998) study the problem of minimizing total completion time in the case of compatible products and non-identical product sizes. They propose new heuristic approaches and compare their performances with the heuristics developed by Uzsoy (1994).

Dupont and Dhaenens-Flipo (2002) extend the results of Ghazvini and Dupont (1998) for the problem where the objective is to minimize makespan. They provide a branch and bound solution algorithm, which is able to find optimal solution in a better computation time than the previous enumeration methods if the number of products and the product sizes are small. Non-identical product size case is also studied by Dobson and Nambimadom (2001) with additional incompatibility constraint between product types. They prove that the problem is NP-hard for this setting and total completion time criteria. Consequently, they propose an iterative batching-sequencing solution procedure which can lead to a local optimum. Also, they provide a polynomial time optimal solution procedure for a special case of the problem. For the general problem, they develop heuristic solutions and discuss the solution qualities with problem parameters.

Uzsoy and Yaoyu (1997) address the problem with identical product sizes, priority weights assigned to products and a total weighted completion time criteria. They provide a number of efficient heuristics and a composite heuristic which has an embedded local search. Considering the same objective, Azizoglu and Webster (2001) focus on the problem with incompatible product types and non-identical product sizes. They propose a branch and bound procedure which solves the problem optimally for up to 25 products.

Research focusing on multi-station systems containing a batch processor also exists in static problem domain. Ahmadi et al. (1992) study two-station systems with compatible product types, constant batch process times and a total completion time criteria. Focusing on a serial-batch processor setting, they provide a dynamic programming method that has a complexity in the form of $O(n^3)$ where n is the number of products to be processed by the serial-batch processor system. They also show that the batch-serial processor setting is NP-complete for total completion time criteria. For this problem, Kim and Kim (2002) propose a genetic algorithm (GA) based approach and discuss its performance with the heuristics developed by Ahmadi et al. (1992).

2.1.2 Dynamic problem domain with cycle time related objectives

The dynamic domain has more relevance to real world situations. This domain allows an arrival process for the products to become available for the batch processor. The arrival times are usually referred to as “release times” or “ready times” depending on the position of the batch processor in the wafer production line. The literature is grouped into three categories based on the availability of the future arrival information of the products. In the first group, all future arrival data is available to the decision maker at the beginning of the decision process. The problem becomes fully deterministic, and machine scheduling approaches are utilized to provide solutions. Uzsoy (1995) focuses on minimizing the makespan on a batch processor with incompatible product types. He provides a time-symmetric solution procedure which follows a full batch policy. Lee and Uzsoy (1999) consider the same problem assuming that product types are compatible. They perform analysis on the special cases of the problem such as the case of agreeable arrival and process times, and the case of two distinct arrival times. They propose polynomial solution methods for these special cases and a few heuristic methods for the general problem. Liu and Yu (2000) study the problem with compatible product types and makespan criteria. They show that the problem has NP-hard complexity even in the case of fixed number of distinct arrival times. Consequently, they propose a greedy heuristic method which has an approximation level of two. Sung et al. (2002) focus on the same setting of the problem assuming products can be grouped by fixed number of distinct batch process times. Their dynamic programming approach has polynomial complexity with the number of products in each group and exponential complexity with the number of groups. Cheraghi et al. (2003) examine the restricted version of the problem with makespan criteria. The restrictions come from the assumptions that batch processing times are the same for all product types and products have due-dates which must be met in a schedule. They develop a GA based heuristic method for the problem.

The deterministic scheduling domain suffers from two main problems in practice. Especially in the dynamic domain, only little information on future arrivals can be obtained in shop floors. The level of stochasticity increases in the problem data with the length of the scheduling horizon. Therefore, dynamic updates on the scheduling decisions must be made. The second and most important issue is the computational burden in the optimal solution procedures developed for the problems. Even in the static problem domain, the optimal solution procedures are not much better than complete enumeration because of the complexity of the problems. Therefore literature mainly focuses on heuristic methods.

In the second group, no future arrival information is available to the decision maker and problem remains in a full stochastic framework. Literature is limited to control limit policies which are based upon the information about the current state of the batch processor. These policies suggest the start of the batch process when the number of products waiting in the batch processor’s queue exceeds the control limit. Neuts (1967) focuses on controlling a batch service queue with Poisson arrivals of a single product type and comes up with the frequently used Minimum Batch Size (MBS) rule. According to this rule, a batch starts when the number of products in the queue exceeds the MBS level. Deb and Serfozo (1973) provide a dynamic programming formulation to choose the MBS level in order to minimize the expected discounted cost over an infinite horizon. They claim that if the optimal MBS value is used, a better control policy cannot be found using only information about the current state of the batch processor. Later work by Glassey and Weng (1991) shows that using a non-optimal MBS value can result in significant deviations from optimality. Using semi-markov decision model and dynamic programming, Duenyas and Neale (1997) provide an optimal control limit policy for a single batch processor where the number of product types is limited to two. Adapting this optimum control policy, they propose heuristic control policies for larger number of product types. Neale and Duenyas (2003) also study the compatible product type case in which batch process times of each product type is coming from a separate distribution and products of different types can be batched together. They develop a semi-markov decision model for two product type case. The state space of this dynamic programming method increases non-polynomially with the number of product types. Hence, they propose a heuristic approach for problems with more than two product types. Avramidis et al. (1998) develop an optimal batch control policy to minimize expected long-run average number of products in the batch processor's queue for the case of single product type. Their main contribution is the extension to the work of Deb and Serfozo (1973) for the case where the batch process time is defined by a general distribution. Akcali et al. (2000) discuss the application of control limit approaches in a real wafer fabrication. They use two stage (loading-dispatching) methods for batch processors with incompatible product types. In the first stage, loading problem focuses on the decision of whether to start a batch or to wait for future arrivals. They use threshold approaches for this stage. The second stage focuses on the selection of the product type. They use a number of priority metrics to choose the winner product type.

There are a few studies that address the use of control limit policies on two-stage processor systems that contain a batch processor. Gurnani et al. (1992) consider a serial-batch processor system where there are multiple serial processors feeding a batch processor. In their model, the serial processors are subject to random failures which make the arrival rate to the batch processor change over time. They propose a control-limit policy to minimize the costs associated with the control of the batch processor where the control limits are approximately found using stochastic dynamic programming with a renewal approximation method. However, their model does not include any cost item related to serial processors and does not utilize the current state of the serial processors. Neale and Duenyas (2000) focus on different two-stage processor systems where there is a single product type and the objective is to minimize the average number of products in the whole system. For a serial-batch processor sequence, they use a stochastic dynamic programming formulation with three-dimensional state space. The dimensions include the number of products at the serial processor, in the batch processor’s queue and being served in the current batch. They use value iteration algorithm and show that a control limit policy is optimal. However, the complexity of this value iteration algorithm is sensitive to the size of the batch processor's capacity and the upper limits on the queue size.

A common drawback in the control limit policies is that it is not possible to find analytical optimum control limits for the case of multiple product types. Therefore the concentration in the literature is limited to single product type case. In real world situations, a single batch processor can process hundreds of different product types. Although control limits (threshold methods) are commonly used in practice, there is more information available in today's shop-floors than full stochastic product flows. The third group addresses this issue.

In the third group, near-future arrival information is assumed to be available to the decision maker. A detailed discussion of this group is provided due to its relevance to the research domain in this dissertation. Look-ahead batch control strategies utilize the near-future information in a specified time-window to choose the best point to start the batch process. A decision point is defined by distinct points in time that the batch processor becomes available or an arrival occurs while the batch processor is in waiting mode. Glassey and Weng (1991) propose the first look-ahead batch control policy, Dynamic Batching Heuristic (DBH), for the case of single product type. The performance measure of the control task is mean waiting time of the products in the batch processor's queue. DBH evaluates the future arrival points existing in a batch process time-window which make the candidate set of batch process start times. The number of candidate points is determined by the minimum of the remaining space in the current batch and the number of arrivals expected in a batch process time-window. If the outcome of the evaluation favors a future arrival point to start the batch process, the batch processor waits for that arrival to start the batch process; otherwise the batch process starts immediately at the current decision point. DBH shows better results as compared to the MBS rule. However on a wait decision, instead of postponing a decision to the next arrival point, DBH aims to jump ahead. This way, a possible improvement on the accuracy of the decision is avoided since updating the decision at intermediate arrival points is skipped.

Fowler et al. (1992) address this issue by integrating a rolling horizon approach, and propose a new control strategy called Next Arrival Control Heuristic (NACH). NACH takes only the next arrival time into account, and if it is more beneficial to start the batch process at the next arrival time, the decision making process is repeated once this arrival occurs. The results show that rolling horizon approach improves the robustness of the decisions. Another contribution of NACH is its extension to the case of multiple product types. At a decision point, if full batches are available, the batch process starts straight away. In that case, the product type to be loaded is chosen using a Weighted Shortest Processing Time (WSPT) dispatching rule. If no full batches are available, each product type present in the batch processor’s queue is evaluated by NACH heuristic proposed for the single product type case, ignoring the other product types. This way a decision is determined for each product type. If all product types have start decisions, then WSPT is again used to choose the winner product type. If all product types have a wait decision, then the decision is updated at the next arrival point. If some product types have start, some others have wait decisions; total waiting times corresponding to decision alternatives are evaluated in the time-windows that are affected by the execution of the decision alternatives. The minimizing decision alternative is selected as the winner. In their later work, they extend NACH approach for multiple processor case (Fowler et al. (2000)). Although NACH improves the quality of the batching decisions, there remain issues worth exploring. The most important issue is that the evaluation considers only next arrival times of product types leaving out the information on other future arrivals within the decision horizon. Weng and Leachman (1993) include this point in a new control strategy called Minimum Cost Rate (MCR), which aims to minimize the average queue length of the batch processor. The cost index associated with each decision alternative is the total waiting time of products during the execution of the decision alternative divided by the length of the time-window affected by the decision alternative. In the case of partial batches, MCR considers a number of future arrival points limited by the capacity of the batch processor. The arrival point that minimizes the expected cost and the associated product type are chosen as the output of the decision process. Similar to DBH, MCR forces jumping to the winner arrival point without refreshing the decision process at intermediate arrival points. Compared to MBS, DBH and NACH, MCR performs better in situations where near-future information is accurately available. This is due to the amount of information used by MCR on all product arrival times in the decision horizon. However NACH shows more robustness with prediction errors.

Robinson et al. (1995) extends MCR by adding a rolling horizon approach in a new control strategy called Rolling Horizon Cost Rate (RHCR) heuristic. RHCR follows the same cost indexing of MCR. If the cost rate index of a future arrival point is the minimum of the alternatives, instead of directly jumping to that arrival point, RHCR decides to repeat the decision making process at the next arrival point. RHCR performs identical to MCR for both single product and multiple product case in the case of no prediction errors. Results favor RHCR compared to MCR when prediction errors are injected to the future arrival information but show no improvements compared to NACH.

Van Der Zee et al. (1997) integrate the strong points of NACH and MCR in a new control strategy called Dynamic Job Assignment Heuristic (DJAH). According to DJAH, next arrival times of product types are considered as the alternative batch process starting points similar to NACH. Similar to MCR, DJAH uses a cost rate method to evaluate the effect of batching decisions in the look-ahead windows. On the other hand, similar to NACH, DJAH adopts a rolling horizon decision making mechanism in which if starting the batch process at a future arrival point is more beneficial, then DJAH repeats the algorithm at the next arrival point. Results indicate that DJAH outperforms NACH and MCR. In their later work, Van Der Zee et al. (2001) extend DJAH to the case of multiple batch processors working in parallel. They also propose a similar control strategy for the case of compatible product types (Van Der Zee (2007)). Cigolini et al. (2002) incorporate the “Wait No Longer Than Time” (WNLTT) concept from semiconductor manufacturing. WNLTT for a particular product type is the maximum time in which another arrival of the product type reduces the total waiting time of the products in the time-window. They also consider set-up times between the batch processes of different product types. A specific WNLTT value for each product type is calculated. The minimum of these values is chosen as the global WNLTT.

There are a few studies that apply look-ahead control strategies for multiple stage processor systems that include a batch processor. Robinson et al. (1995) propose an extension of RHCR for a batch-serial processor system. In the extended strategy called RHCR-S, starvation time for the downstream serial processor is additionally included as near-future information. They provide a comparison of RHCR and RHCR-S and show that RHCR-S reduces the cycle time of the batch-serial processor system. Solomon et al. (2002) investigate a version of NACH strategy (named as NACH-setup) for the same batch-serial processor system where a setup is required in the serial station when two consecutive products are from different types. They discuss the influence of downstream setup times on the batching decisions. Van Der Zee (2002) focuses on a similar batch-serial processor system in which the serial station has multiple parallel processors. He presents an extension of DJAH strategy for this system called DJAH-F and provides comparative study with the RHCR-S strategy.

Look-ahead control approach has been extensively studied in the literature for cycle time related performance criteria assuming that near-future information for the batch processors can be predictable in wafer fabrication through monitoring systems. Although future arrivals for the batch processors are determined by upstream process completions (if zero transfer times between stations are assumed), up to date literature does not discuss look-ahead control approach in a two station setting where the downstream station is the batch processor. And also, upstream control is not embedded in any of the studies. Chapter III addresses the use of upstream control and look-ahead batching in minimizing cycle time of a serial-batch processor system.

2.2 Batch process control with due-date related objectives

Due-date related performance criteria have received more attention recently in the control of batch processors compared to those related to cycle time. Table 2.2 provides a matrix of the literature in this group according to the availability of future information and nature of the product flow.

Table 2.2. List of literature on due-date related objectives

Availability of the future info Nature of the product flow	Full knowledge on future arrivals	Near-future arrival information is available
Dynamic	Li and Lee (1997), Mason et al. (2002), Mathirajan and Sivakumar (2006), Tangudu and Kurz (2006)	Kim et al. (2001), Monch et al. (2005), Habenicht and Monch (2005),
Static	Mehta and Uzsoy (1998), Balasubramanian, et al. (2004), Perez et al. (2005), Jolai (2005)	

2.2.1 Static problem domain with due-date related objectives

In the static domain, the problem relates to deterministic machine scheduling and deals with two subtasks: how to form the batches and how to sequence them. Mehta and Uzsoy (1998) discuss the use of dynamic programming in scheduling a single batch processor where there are multiple product types. The objective of the problem is to minimize total tardiness and they show that this problem is NP-hard. They develop a dynamic programming method that has polynomial complexity with the number of products. In order to provide less complex solutions, they propose a heuristic batch prioritization method called BATC, which is the batch version of the ATC rule developed by Vepsalainen (1987). According to this indexing method, batches are formed for each product type in the order of increasing due-dates. Then a BATC index is assigned to each batch, and batches are sequenced according to their priority indices. Perez et al. (2005) includes the product priority weights to the problem. For total weighted tardiness criteria, they provide experimental study that combines and tests different heuristics in a two stage (batching-sequencing) solution framework. Best performance is obtained when ATC rule is used to assign products to batches, BATC rule is used to determine the initial sequence of the batches and a heuristic search method is applied to this initial batch sequence. The search method simply divides the whole sequence into subsequences with length λ and starting from the first subsequence finds the optimal sequence of the subsequence by complete enumeration. Parallel batch processor version of the same problem is studied by Balasubramanian et al. (2004) with total weighted tardiness criteria. They propose three-stage decomposition algorithms. In the first algorithm, products are assigned to batches, batches are assigned to processors and then sequence of the batches is determined for each individual processor. In the second algorithm, products are assigned to processors, batches are formed for each processor with the assigned products and the sequence of the batches for each processor is determined. For each product type, products are prioritized using ATC rule and batches are formed with this priority rule. Then for each batch, BATC indexing is used to determine the priority of the batch. Genetic algorithm plays the role on assigning batches and products to the processors in the first and second algorithms respectively. Jolai (2005) proposes a dynamic programming method to minimize number of tardy jobs on a batch processor station where incompatible product types exist. The complexity of the method is exponential with number of product types. A polynomial solution is discussed for a special case of the problem in which products of the same type have common due-dates.

2.2.1 Dynamic problem domain with due-date related objectives

In the dynamic problem domain, the literature is categorized into two groups: problems with full knowledge on future arrival information and problems with near-future arrival information are available. In the first group, the domain becomes deterministic machine scheduling where arrival times and due-dates of the all products are known perfectly. Li and Lee (1997) focus on minimizing maximum tardiness on a single burn-in oven where products are compatible. They provide a proof for the NP-hard complexity of the problem, and propose a dynamic programming algorithm for the special case of agreeable ready times and due-dates. Tangudu and Kurz (2006) study the problem with incompatible product types and total tardiness criteria, and provide a branch and bound procedure which shows better complexity than complete enumeration. Mason et al. (2002) explore the prioritization method of Mehta and Uzsoy (1998), namely BATC, for dynamic problem including sequence-dependent batch processing steps. In this new prioritization method called BATCS, batch ready times and sequence-dependent setup times are included in the formulation. Mathirajan and Sivakumar (2006b) discuss a three-step heuristic algorithm for scheduling parallel non-identical batch processors where non-identical product sizes exist. In the first step, algorithms select the batch processor that will be scheduled next according to the availability times of the processors. In the second step, the product type that will be loaded to the batch processor is selected using priority indices driven by the processing times and the cumulative due-dates of product types. In the third step, a batch to the full extent from the winning product type is selected using alternative priority rules. Then the availability time of the batch processor is changed to the completion time of the selected batch. These three steps are repeated till all the products are scheduled. The relevance of models relying on full knowledge of future arrivals is quite limited because in practice only little information is available on future arrivals.

The second group assumes that limited future arrival information is available to the decision maker on the decision points. In this case, literature focuses on developing heuristic procedures in which the batching decisions are based on the information lying in a pre-specified time-window. Kim et al. (2001) provide a modification of DBH strategy suggested by Glassey and Weng (1991), to minimize the total tardiness. According to the new control strategy, namely MDBH, product types are prioritized based on the average due-date slack time of the products waiting in the batch processor’s queue. Starting from the highest priority product type, two decision alternatives, whether to wait for another arrival or start the batch process at current time, are compared. In the comparison, the total weighted waiting times that are caused by these alternatives are determined using the reciprocals of due-date slacks as product weights. Once a start decision is found, the batch process starts with the product type. On the other hand if all product types return a wait decision, then the decision making process is postponed to the next arrival point. Habenicht and Monch (2005) attach a time-window based batch composition to the prioritization method developed by Mason et al. (2002). According to the new prioritization rule, namely DBDH, all possible batch compositions in a specified time-window are determined and prioritized for each product type. Consequently, the final decision is made using the priority indices of the alternative batch compositions. Monch et al. (2005) propose three time-window based priority indexing methods extending DBDH. The priority indices, namely BATC-I and BATC-II show very good performance compared to alternative heuristics. They also discuss the use of decision theory in prioritizing batches. According to this approach, total weighted tardiness of alternative batch decisions is estimated in the time-window and the decision alternative with the minimum estimate is selected. This approach is advantageous in the sense that the effect of a batching decision on other product types is accounted in the decision process. Monch et al. (2006) address the use of neural network for selecting the best performing parameters to improve the effectiveness of time-window based prioritization rules. They provide analysis of the factors that have influence on the performance of these parameters.

There is a potential in extending the look-ahead batch control idea for due-date related objectives. None of the existing control strategies evaluate the decision alternatives in the look-ahead window with a mean tardiness metric. Chapter IV addresses the use of look-ahead batch control on mean tardiness performance of a serial-batch processor system. Additionally, Chapter IV demonstrates how an upstream station can be controlled to improve the batch process decision making with mean tardiness performance measure.

2.3 Batch process control with multiple objectives

In semiconductor manufacturing, management usually deals with multiple performance criteria simultaneously. Such cases require a strategy that will result in target performance levels in all criteria. However, frequently observed conflicts between criteria of interest make it very difficult to find out effective strategies. Tabucanon (1988) describes general solution techniques for problems with multiple objectives. A detailed survey on the evolutionary multi-criteria optimization techniques can be found in Coello (1999). Although multi-criteria analysis is a mature research area in scheduling (see T'kindt et al. (2006)), there is a limited amount of research that has been specialized in the area of batch process control.

Controlling batch processors with multiple performance criteria is a relatively new research area. Ganesan et al. (2004) propose a concept called schedule control for the batch processors to minimize mean cycle time and maximum tardiness simultaneously. According to this concept, each decision alternative is simulated within short-term future and the outcomes with respect to the criteria of interest are estimated. This way, Pareto optimal decision alternatives are found and given to the decision maker as the Pareto-optimal boundary. Reichelt and Monch (2006) focus on minimizing makespan and total weighted tardiness on multiple batch processors in a dynamic problem setting. Their approach is the adaptation of the three-stage (batching, assignment and sequencing) algorithm demonstrated by Monch et al. (2005) to multiple objective situations. The adaptation takes place in the batch assignment stage in which a genetic algorithm based method (NSGA-II) is used to find the Pareto-front solutions and a local search method is used to improve the Pareto-front solutions. Gupta and Sivakumar (2007) focus on minimizing earliness and tardiness on a batch processor. They use a look-ahead batching method to evaluate different batch scenarios and compromise programming to find the Pareto-optimal boundary.

Although there are a few studies exploring the use of look-ahead batch control within problems where there are multiple criteria, none of these studies attempts to simultaneously control mean tardiness and mean cycle time performances of the batch processors. Chapter V addresses this issue by extending the results of Chapter III and Chapter IV for the batch process control problem where the objective is to minimize both mean cycle time and mean tardiness performances.

CHAPTER III

EFFECT OF UPSTREAM RE-SEQUENCING IN CONTROLLING CYCLE TIME PERFORMANCE OF BATCH PROCESSORS

This chapter discusses the effect of controlling upstream processors to improve the cycle time performance of batch processors. The focus is on the problem domain in which near-future arrival information for the batch processors is available by predicting the upstream process times. The objective is to minimize the mean cycle time of the products that visit the batch processor. The sequence information at the upstream station is used while evaluating the decision alternatives of either starting the batch process at the current decision point or waiting for future arrivals. A new control strategy that involves a re-sequencing procedure for the upstream station is proposed in this chapter.

3.1 Introduction

In today's semiconductor manufacturing, management still considers cycle time related performance measures to be among the most important performance indicators from the capacity planning perspective to the manufacturing perspective. Therefore, maintaining short cycle times is one of the major objectives in wafer production. However, the complicated production specifics of wafer fabrication discussed in Chapter I make this objective quite challenging. One major complication is the presence of batch processors in the production system. Control of batch processors is often a very critical task and receives priority from the management perspective. In this chapter, the focus is on controlling a single batch processor that is described in Chapter I. The performance measure of the control task is mean cycle time of the products.

In dynamic systems, on-line control reviews the state of the production system at specific decision points to find the product type to be processed next and the time that the process will start. A decision point is defined as the time that the batch processor becomes available or the time an arrival occurs while the batch processor is idle. At a particular decision point, if the size of a particular batch is equal to the capacity of the batch processor (i.e. full batch of any product type), there is no benefit in delaying the start of the batch process with the cycle time performance measure. On the other hand, if all batch sizes are smaller than the capacity of the batch processor, then a non-trivial decision must be made whether to start one of the partial batches or to wait for future arrivals to occur. In the literature, this typical decision making process is referred to as "batch process control".

In a typical production system, a few upstream production steps carry the most reliable future information for the step being considered. The future arrival horizon can be limited by focusing on only one upstream process if the upstream station holds enough information. In such a two-step production unit, the product sequence of the upstream process determines the future arrival times and the future arrival pattern for the downstream process. In the semiconductor manufacturing environment, it is common practice for a batch process operator to communicate with the operator at the upstream station to obtain information on the sequence of products and also to be involved in the upstream sequence decision to receive the desired products in a shorter time. Akcali et al. (2000) discuss the importance of these operator communications and local decisions in the batch loading problem, based on their experiences in wafer fabrication. Improving this common practice is the main motivation of the research in this chapter.

No mathematical formalism exists in practice for use in such re-sequencing decisions. In this chapter, a control strategy called Next Arrival Re-sequencing based Control Heuristic (NARCH) is proposed to combine re-sequencing with a look-ahead batching framework. The rest of the chapter is organized as follows. In Section 3.2, definition of the problem is given with the notation used in later sections. The control strategy proposed for the problem is described in Section 3.3. A simulation study is presented to compare the proposed strategy with the benchmark approaches in Section 3.4. The contribution of the chapter is discussed in Section 3.5.

3.2 Problem definition and notation

A production unit, composed of a perfectly reliable serial processor followed by a perfectly reliable batch processor, is considered here. The objective is to minimize mean cycle time of the products visiting this production unit. A serial processor handles one product at a time while a batch processor can process products in batches with a capacity limitation on the number of products in the batch. There are N product types visiting the serial-batch processor system which are incompatible in the batch process. Once a product arrives at the serial process station, its serial process time is assumed to be predicted by the controller attached to the system. This way, its arrival time at the batch processor, t_{jn} (for n^{th} upcoming product of type j), is determined. Each product may have a different serial processing time. If the serial processor is available, it is immediately loaded with the arriving product. On the other hand, if the serial processor is busy, the arriving product takes the last position in the queue and waits until it is loaded on the processor. Unless changed by a re-sequencing decision, products are processed in the serial processor by following a first-in-first-out rule. Once a product's serial process is complete, it arrives at the batch process station and waits in the buffer reserved for its product type until loaded to the batch processor. For a particular product type j, the batch process takes a constant T_j time units independent of the number of products in the batch with a capacity limit B_j. The serial processor's queue and batch processor's buffers are assumed to have infinite storage capacities. Figure 3.1 illustrates the serial-batch processor system for two product types. The future information beyond the serial processor queue is stochastic and unknown. On the other hand the near-future information in the serial processor station is available to the controller at the decision points. The following notations are used in the rest of the chapter:

N = number of product types
T_j = batch processing time for product type j
B_j = batch processor capacity for product type j
P_j = ratio of product type j in the mix
t_0 = current decision point
t_{jn} = time that n^{th} upcoming product of type j arrives at the batch processor's buffers
q_j = number of products of type j in the batch processor's buffers at t_0
D_{j0} = mean waiting time metric value for starting the batch process of type j at t_0
D_{jn} = mean waiting time metric value for starting the batch process of type j at t_{jn}

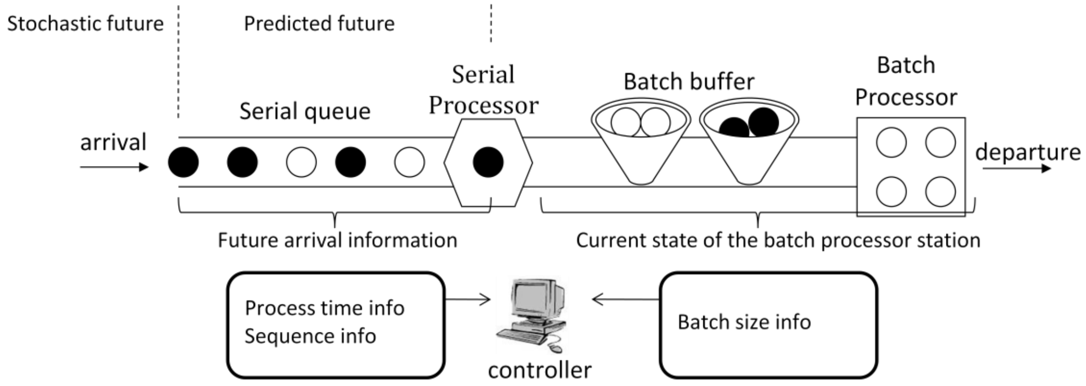

Fig. 3.1. Serial-batch processor system for Chapter III

3.3 Next arrival re-sequencing based control heuristic (NARCH)

NARCH is a rolling horizon look-ahead control approach which combines the strongest components of look-ahead batch control strategies with an upstream re-sequencing method. There are three components of this approach: a mean waiting time metric that uses a look-ahead window, rolling-horizon decision making and re-sequencing at the upstream station (see Figure 3.2). Given the constant batch process times, the minimization of the cycle time at the batch processor station is in fact equivalent to the minimization of the mean waiting time in the batch processor’s buffers. Therefore, a mean waiting time metric is used similar to those found in look-ahead batching literature. The mean waiting time metric evaluates the alternative decisions using the time-windows (look-ahead window) that are affected by the execution of the decision alternatives. The purpose of re-sequencing at the upstream station is to shorten the next arrival time of product types in the evaluation of the wait decisions. Rolling horizon decision making adds the benefit of updating a decision at intermediate arrival points if the decision requires an additional arrival of a product type to start the batch process. This update allows refreshing the decision with additional future information at a future decision point.

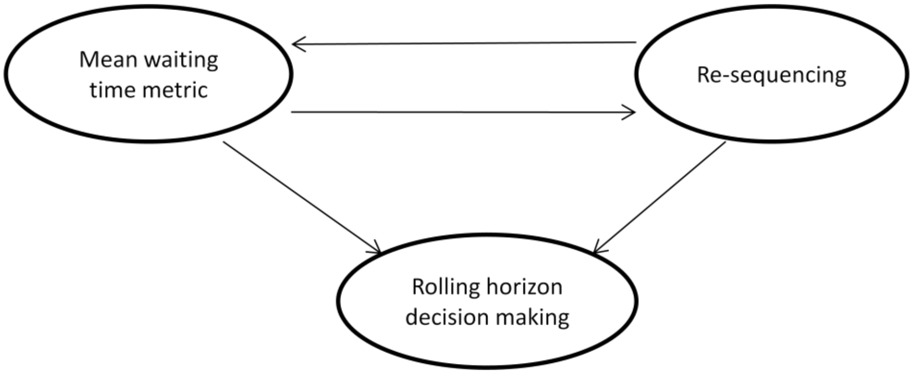

Fig. 3.2. Components of NARCH algorithm

At a particular decision point t_0, the controller starts the review of the serial-batch processor system by checking for any full batch product types. If there is at least one full batch product type, i.e. $\exists j, j \in \{1,2, \dots N\}, q_j \geq B_j$, then the trivial decision is to start the batch process with the product type, the loading of which minimizes the mean waiting time metric in the decision horizon. For product type *j*, the time-window that is considered in a start decision is (t_0, t_0+T_j). Equation (3.1) determines the value for the mean waiting time metric that is caused by starting the full batch of product type *j*.

$$D_{j0} = \left(\left(\sum_{k=1}^{N} q_k - B_j\right) \times T_j + \sum_{k=1}^{N} \sum\nolimits_{t_{km} < t_0 + T_j} \left(t_0 + T_j - t_{km}\right)\right) \Big/ B_j \tag{3.1}$$

D_{j0} finds the total waiting time of those products that spend time in the batch processor's buffers during the horizon when the batch processor is processing the full batch of product type j. The total waiting time is divided by the throughput of the decision alternative, B_j. The products that are either available in the batch processor's buffers at t_0 or expected to arrive at the buffers within T_j time units are considered. The first portion of D_{j0} calculates the total waiting time of the products that are available in the batch processor's buffers, but are not loaded on the batch processor at t_0. The second portion calculates the total waiting time of the products that are expected to arrive within the time-window of the decision alternative. After calculating D_{j0} values for each full batch product type, the product type j^* with the minimum D_{j0} value is loaded on the batch processor at t_0. j^* is found by equation (3.2).

$$j^* = \operatorname{argmin}(D_{j0} | j \in \text{set of full batch product types}) \tag{3.2}$$

If there is no full batch of any product type at t_0, then further analysis is required to reach a decision. The analysis has two stages. In the first stage, each product type is individually evaluated, excluding the effects of other product types. At the end of the first stage, individual decisions are suggested for product types that are available in the batch processor's buffers at t_0. In the second stage, the decision alternatives coming from the first stage (maximum of N alternatives) are evaluated by determining the values of the mean waiting time metric in their decision horizons. The effects of all product types are included in this evaluation.

In the first stage, the following analysis is employed for each product type that is available in the batch processor's buffers at t_0. For a particular product type j, assume there are R_j products of type j that are available in the serial processor station. Then each of these R_j products is a candidate for being the next job in the serial processor through re-sequencing. The total waiting time gain/loss of waiting for the first arrival of product type j is calculated by trying each of these R_j products as the assumed next job in the serial processor. Consider the n^{th} arrival of product type j where $n \in \{1,2,..R_j\}$. Equation (3.3) determines the total waiting time gain/loss value, M_{jn}, of waiting for this product after pulling it to the front of the serial processor's queue.

$$M_{jn} = \left(t_0 + T_j - t_{jn}\right) - q_j \times (t_{jn} - t_0) \tag{3.3}$$

To find M_{jn}, the arrival time t_{jn} is updated as equal to the sum of the product's predicted processing time and the remaining time of the serial processor's current job. The first portion of equation (3.3) calculates the waiting time gain of the arriving product at updated t_{jn}, and the second portion calculates the total waiting time increase of the q_j products that are already in the batch processor's buffers at t_0. M_{jn} values are calculated for each $n \in \{1,2,..R_j\}$, and n^* is found by equation (3.4).

$$n^* = \text{argmax}(M_{jn} \,|n \in \left\{1,2,..R_j\right\}) \tag{3.4}$$

It should be noted that n^*=1 if the serial processor is processing a product of type j at t_0. Otherwise, out of R_j products, the one with the shortest serial processing time gives the M_{jn*} value. If M_{jn*} is positive, then the suggested decision for product type j is to wait for the n^{*th} product after re-sequencing the serial processor's queue, as the n^{*th} product of type j becomes the next job for the serial processor. On the other hand, if M_{jn*} is negative or equal to 0, then the decision suggested for product type j is to start the batch process at t_0 with q_j products.

The first stage is completed by employing the same analysis for each product type. The set of decision alternatives (each decision alternative corresponds to one product type) moves to the second stage, which evaluates the decision alternatives by including these decisions' effects on the other product types. The following method is applied for each decision alternative. If the decision suggested for product type j is to start the batch process at t_0, the value of the mean waiting time metric (D_{j0}) caused by this decision is calculated by equation (3.5). The only difference between (3.1) and (3.5) is the number of products being processed in the batch (B_j and q_j respectively).

$$D_{j0} = \left(\left(\sum_{k=1}^{N} q_k - q_j \right) \times T_j + \sum_{k=1}^{N} \sum\nolimits_{t_{km} < t_0 + T_j} \left(t_0 + T_j - t_{km} \right) \right) \Big/ q_j \tag{3.5}$$

However, if the decision suggested for product type j is to re-sequence the serial processor's queue as the n^{*th} arriving product of type j becomes the next job for the serial processor, and to wait for this arrival before starting the batch process, then equation (3.6) determines the value of the mean waiting time metric that is caused by this decision. It should be noted that the updated arrival time of the n^{*th} product, t_{jn*} and updated arrival times of the other products t_{km} are found by different methods in the following two cases:

i) If the serial processor is currently processing a product of type j, then this product is the one for which the batch processor is waiting. In this case, the values of t_{jn*} and t_{km} are not changed since there is no re-sequencing involved.

ii) If the serial processor is currently processing another product type, then the value of t_{jn*} is the sum of n^{*th} product's serial processing time and the remaining time of the serial processor's current job. For any other product initially having an earlier sequence position than the n^{*th} product of type j, the new t_{km} values become the sum of the old t_{km} values and the serial processing time of the n^{*th} product. The arrival time of the current product being processed by the serial processor remains the same, as well as the products that initially have a later position in the sequence than the n^{*th} product.

$$D_{jn^*} = \left(\left(\sum_{k=1}^{N} q_k \right) \times (t_{jn^*} - t_0) + \left(\sum_{k=1 \,\&\, k \neq j}^{N} q_k \right) \times T_j + \sum_{t_{km} < t_{jn^*} + T_j \,\&\, t_{km} \neq t_{jn^*}} (t_{jn^*} + T_j - t_{km}) \right) / (q_j + 1) \tag{3.6}$$

D_{jn^*} determines the total waiting time that will occur within the interval (t_0, $t_{jn^*}+T_j$) and divides this value by the size of the batch that will start at t_{jn^*}. The first portion of the equation calculates the total waiting time coming from the additional $t_{jn^*}-t_0$ delay of the products that are in the batch processor's buffers at t_0. The second portion accounts for the T_j delay of the products that are in the batch processor's buffers at t_0 except those of product type j. The last portion calculates the delay of the products that arrive at the batch processor's buffers in the interval (t_0, $t_{jn^*}+T_j$).

The same calculations are completed for each decision alternative. For each product type j, the mean waiting time metric value of its associated decision alternative (after calculating by either (3.5) or (3.6)) is recorded as D_j. The final decision of the algorithm at the decision point t_0 is the suggested decision of the product type $j^* = \text{argmin}(D_j)$. If the suggested decision for j^* is to start the batch process at t_0, then the batch process starts with available products of type j^* immediately.

```
If there are full batch product types (i.e. $q_j \geq B_j$) at $t_0$
  Start the batch process with product type $j^* = \text{argmin}(D_{j0} | q_j \geq B_j)$ where $D_{j0}$ is found by equation (3.1)
Else
    For all product type $j$ available in the batch processor's buffer at $t_0$
        If number of products of type $j$ in the serial processor station, $R_j$, is positive
            For each $n$=1,…..,$R_j$
                Calculate $M_{jn}$ by re-sequencing using equation (3.3)
            End For
            Find $n^* = \text{argmax}(M_{jn} | n = \{1,2,\ldots R_j\})$
            If $M_{jn^*} > 0$ then
                The decision suggested for product type $j$ is to wait for its next arriving product after re-sequencing the serial processor's queue as its $n^{*th}$ arriving product becomes the next job on the serial processor
            Else
                The decision suggested for product type $j$ is to start the batch process with $q_j$ products at $t_0$
            End If
        Else
            The decision suggested for product type $j$ is to start the batch process with $q_j$ products at $t_0$
        End If
    End For
    For each product type $j$
        If the suggested decision is start the batch process at $t_0$
            $D_j = D_{j0}$ ($D_{j0}$ is calculated by equation (3.5))
        Else
            $D_j = D_{jn^*}$ ($D_{jn^*}$ is calculated by equation (3.6))
        End If
    End For
    Find $j^* = \text{argmin}(D_j)$
    If the suggested decision for $j^*$ is to start at $t_0$
        Start the batch process with $q_{j^*}$ products of type $j^*$ at $t_0$
    Else
        Re-sequence the serial processor queue with the underlying re-sequence decision of product type $j^*$ and wait for the next arrival point
    End If
  End If
```

Fig. 3.3. Pseudo-code of the NARCH algorithm

On the other hand, if the suggested decision for j^* is to wait for a future arrival, then the underlying re-sequencing action is taken on the serial processor queue for product type j^* and the batch processor stays idle until the next arrival point. The batching decision is reviewed at the next arrival point by the algorithm. Figure 3.3 presents the pseudo-code of the NARCH algorithm.

3.4 Simulation study

Benchmark control strategies, the simulation model that is used for comparing the strategies and the results of the simulations are discussed in this section.

3.4.1 Benchmark control approaches

A detailed discussion of the look-ahead policies was provided in Chapter II. Three look-ahead control strategies that focus on mean cycle time performance of batch processors are considered as benchmarks. In addition to the look-ahead control strategies, a control limit approach is also utilized as a benchmark. Table 3.1 summarizes the attributes of these benchmarks, together with NARCH.

The first benchmark strategy is NACH (Next Arrival Control Heuristic) proposed by Fowler, et al. (1992). NACH is the first look-ahead control strategy that involves a rolling-horizon decision making that improves the quality of batching decisions by updating the near-future information at intermediate arrival points. The mean cycle time metric used in NACH is the total waiting time of the products within the decision horizon. The second benchmark strategy is RHCR (Rolling Horizon Cost Rate) developed by Robinson, et al. (1995). Similar to NACH, RHCR uses a rolling horizon decision making and look-ahead batching framework. The main difference is that RHCR considers all arrival points within the decision horizon as candidate batch starting points whereas NACH considers only the next arrival points of each product type. RHCR uses a mean queue length metric, which is found by dividing the total waiting time of the products within the decision horizon by the length of the decision horizon.

Table 3.1. Summary of the benchmark control approaches

Algorithms Features	MBS	NACH	RHCR	DJAH	NARCH
Near-future knowledge	X	√	√	√	√
Rolling horizon decision making	X	√	√	√	√
All points in the decision horizon are candidate start points	X	X	√	X	√
Metric: total waiting time	X	√	X	X	X
Metric: total waiting time / length of the decision horizon	X	X	√	X	X
Metric: total waiting time / number of products in the batch	X	X	X	√	√
Re-sequencing on the upstream station	X	X	X	X	√

The third benchmark strategy is DJAH (Dynamic Job Assignment Heuristic) proposed by Van Der Zee, et al. (1997). DJAH involves a rolling-horizon decision making as other benchmarks. Similar to NACH, DJAH considers next arrival points of each product type as the candidate points for delaying the start of the batch process. Also similar to RHCR, DJAH considers the effect of a decision alternative on all arrival points within the decision horizon. The mean waiting time metric used in DJAH is found by dividing the total waiting time caused by a decision in the decision horizon by the number of products that will be produced by the execution of that decision, similar to

NARCH. The last benchmark approach is the MBS (Minimum Batch Size) policy which relies on the current state of the batch processor's buffers without using any future information. According to MBS, the batch process starts if the number of products in the buffers exceeds a minimum number. MBS serves as the best benchmark that does not use future arrival information. The purpose in using MBS is to demonstrate the benefit of near-future arrival information in the control of batch processors.

3.4.2 Simulation experiments

The flow of products that occurs in the simulation model of the serial-batch processor system is described here. Products arrive at the serial processor station one by one following a stochastic arrival process. Once the product arrives at the serial processor station, its serial process time is assigned immediately and it takes the last position in the serial processor's queue. The serial processing time and the sequence position information for all products in the serial station is available to the controller, to be used in the batch process decision making. Unless there is any change in the sequence, products follow a first-in first-out rule in the serial processor. After completing the serial process, products enter the buffer allocated for their product type and wait until they are loaded on the batch processor. The cycle time of a product on this serial-batch processor system is the time between its batch process completion and its arrival at the serial processor station.

There are several product, processor and process characteristics that may affect the performance of such a serial-batch processor system. The performance of a control strategy is evaluated under scenarios driven by these characteristics. Table 3.2 gives an overview of the simulation experiments that are commonly investigated by the look-ahead batch control literature (Fowler, et al. (1992), Van Der Zee, et al. (2001), etc.). Each simulation scenario reflects an alternative system configuration that is defined by a particular setting of the product, process and processor characteristics. The performances of the benchmark control strategies are tested on each of the simulation scenarios.

Table 3.2. Configuration of the simulation study for Chapter III

No	Factor	Levels			
1	Control Strategy	MBS			
		NACH			
		RHCR			
		DJAH			
		NARCH			
2	Number of Products	2			
		5			
		10			
3	Interarrival Distribution	Exponential			
			2 Products	5 Products	10 Products
4	Product Mix	Equal	(0.5, 0.5)	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different	(0.7, 0.3)	(0.35, 0.35, 0.1, 0.1, 0.1)	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
5	Capacity By Product	Equal	(5, 5)	(5, 5, 5, 5, 5)	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different	(7,2)	(7, 6, 5, 4, 3)	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
6	Batch Processing Time By Product	Equal	(25, 25)	(25, 25, 25, 25, 25)	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different	(40, 10)	(40, 30, 25, 20, 10)	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
7	Traffic Intensity	0.4			
		0.6			
		0.8			

In order to observe performance change with product diversity, three different settings, low (2), medium (5) and high (10), are considered for the number of product types being processed by the serial-batch processor system. Two different settings are used for the product-mix values, equal and different, to investigate the behavior of the control heuristics in unbalanced product-mix situations. Similarly, two different settings for batch process times and two different settings for batch processor capacity are considered. Since workload has a profound effect on the performance of batch processors, each system configuration is analyzed with low (0.4), moderate (0.6) and high (0.8) traffic intensities. The batch traffic intensity (ρ) is defined by Chaudry and Templeton (1983) as the mean arrival rate of products divided by the maximum batch processor service rate when operating at the maximum capacity. From a particular combination of batch processor traffic intensity (ρ), product mix (P_j), batch process time (T_j) and batch processor capacity (B_j), one can find the mean arrival rate (λ) at the batch processor's buffers using equation (3.7).

$$\lambda = \rho \Big/ \sum_{j=1}^{N} \frac{P_j T_j}{B_j} \tag{3.7}$$

The arrival rate at the serial processor is the same as the arrival rate at the batch processor, λ. The service rate of the serial processor is chosen as $\mu=\lambda/0.8$ in this study, which satisfies a reasonable utilization level for the serial processor as well as a reasonable steady state queue length. Exponential distribution with parameters λ and μ is used for the inter-arrival and service time distributions, respectively. All settings of the system characteristics combine to create the different simulation scenarios. A combination of all settings gives a total of 72 (3^2x2^3) scenarios on which the control strategies are tested. Each scenario is separately simulated with each of the control strategies: each simulation experiment has a duration of 100,000 units, a warm-up of 5,000 time units and 10 replications. The simulation code is developed using VB.net and scenarios are simulated on a Pentium Dual Core 1.73 GHz. processor with 2GB RAM. The mean of the time that products spend after their arrival at the serial processor station until they are loaded on the batch processor is reported in the form of an X-factor, which is the actual time normalized by the batch process time.

3.4.3 Simulation results

In this section, the performance of NARCH is compared with the four benchmark control strategies. Since there is not an analytical method to determine the best MBS levels for a multiple product type problem, all possible combinations of MBS levels for a particular scenario are simulated, and the minimum normalized waiting time achieved from the combinations is reported. In Table 3.3, the mean of the replications are averaged over different settings of the product, processor and process characteristics to illustrate how the performance improvement obtained by NARCH is affected by different system settings. A paired-*t* approach is used with a 95% confidence interval to test the statistical validity of the performance improvements gained by NARCH, compared to the benchmarks (Law and Kelton (1991)). For a particular scenario, the values obtained by the replications are compared pair-wise with a paired-*t* test, and if a significant difference is obtained between strategies, the actual difference as well as the percentage difference is reported. If there is not a significant difference, then zero value is reported for both actual difference and percentage difference (see appendix-A for detailed analysis).

The mean and the half-width of the 95% confidence interval of the normalized waiting times obtained by the control strategies are presented in Table 3.3 in the third through the twelfth columns. The mean normalized waiting times are the average over

Table 3.3. Summary of the simulation results: NARCH is compared with the benchmark control strategies, the last four columns show the percentage improvements obtained by NARCH

		MBS		NACH		RHCR		DJAH		NARCH					
No	Average at:	mean	hmax	mean	hmax	mean	hmax	mean	hmax	mean	hmax	Δ1	Δ2	Δ3	Δ4
1	Number of Products = 2	3.3315	0.0403	3.1060	0.0398	3.1086	0.0398	3.0952	0.0399	3.0475	0.0399	8.7%	2.2%	2.3%	1.8%
2	Number of Products = 5	4.0701	0.0240	3.7963	0.0243	3.7923	0.0239	3.7591	0.0238	3.5698	0.0229	12.1%	5.9%	5.8%	4.9%
3	Number of Products = 10	5.7132	0.0191	5.3008	0.0200	5.3350	0.0200	5.2617	0.0198	4.9582	0.0180	13.3%	6.5%	7.2%	5.8%
4	Traffic Intensity = 0.4	4.7601	0.0403	4.4126	0.0398	4.4434	0.0398	4.4158	0.0399	4.1633	0.0399	12.3%	5.3%	6.0%	5.4%
5	Traffic Intensity = 0.6	4.1500	0.0185	3.8899	0.0179	3.9193	0.0178	3.8724	0.0178	3.6916	0.0182	11.1%	4.9%	5.5%	4.4%
6	Traffic Intensity = 0.8	4.2046	0.0220	3.9005	0.0207	3.8732	0.0212	3.8278	0.0208	3.7206	0.0203	10.7%	4.3%	3.8%	2.7%
7	Capacity = Equal	4.2491	0.0240	4.0186	0.0243	4.0361	0.0239	3.9986	0.0238	3.8242	0.0229	9.6%	4.6%	4.9%	4.1%
8	Capacity = Different	4.4941	0.0403	4.1168	0.0398	4.1212	0.0398	4.0787	0.0399	3.8928	0.0399	13.1%	5.2%	5.3%	4.3%
9	Product Mix = Equal	4.5950	0.0403	4.3531	0.0398	4.3384	0.0398	4.3077	0.0399	4.1391	0.0399	9.8%	4.7%	4.4%	3.7%
10	Product Mix = Different	4.1482	0.0259	3.7823	0.0251	3.8189	0.0250	3.7697	0.0251	3.5779	0.0253	13.0%	5.0%	5.8%	4.7%
11	Process Times = Equal	4.4862	0.0403	4.2125	0.0398	4.2300	0.0398	4.2257	0.0399	4.0184	0.0399	9.8%	4.4%	4.7%	4.6%
12	Process Times = Different	4.2570	0.0240	3.9228	0.0243	3.9273	0.0239	3.8516	0.0238	3.6986	0.0229	12.9%	5.3%	5.5%	3.8%
13	Overall Avg.	4.3716	0.0403	4.0677	0.0398	4.0786	0.0398	4.0387	0.0399	3.8585	0.0399	11.4%	4.9%	5.1%	4.2%

Δ1 = 100*(MBS-NARCH)/MBS
Δ2 = 100*(NACH-NARCH)/NACH
Δ3 = 100*(RHCR-NARCH)/RHCR
Δ4 = 100*(DJAH-NARCH)/DJAH

the scenarios determined by the setting in the second column. The half-width confidence interval (hmax) is the maximum half-width of the 95% confidence interval that is observed in the scenarios determined by the setting in the second column. The last four columns present the percentage improvements obtained by NARCH as compared to MBS, NACH, RHCR and DJAH, respectively.

The overall performance comparison shows that NARCH is the best performing strategy among the benchmarks. Overall performance improvements gained by NARCH are 4.2%, 5.1%, 4.9% and 11.4% when compared to DJAH, RHCR, NACH and MBS, respectively. It should be noted that the best performing benchmark is DJAH, followed by NACH and RHCR, and the performance differences between these three control strategies are very small. These results are consistent with the results obtained by Van Der Zee et al. (1997). The half-width of the 95% confidence interval values indicate that re-sequencing doesn’t increase the variance on the normalized waiting time values whereas the mean waiting time values decrease significantly.

In practice, a batch processor can process many different recipes (referred to as product types) in the same planning horizon. Results indicate that the waiting time values increase when there are more product types. This is mainly due to the fact that the product types compete with each other at the decision points to become the next load on the batch processor. However, the performance improvement gained by NARCH increases when the number of product types increases. Figure 3.4 shows the trend in the percentage improvements and actual improvements obtained by NARCH for different number of product types. The interpretation behind this result is related to the product type inter-arrival times, which are the times between two arrivals of the same product type. Product type inter-arrival times become longer with more product types, since the ratio of the product type in the product-mix decreases. As the next arrival time for a product type increases, the total waiting time of the products in the batch processor’s buffers increases drastically if a wait decision is considered. Therefore, wait decisions are very rare outcomes of control strategies under such circumstances. However, re-sequencing offers an advantage in overcoming this problem by shortening the next arrival time of the desired product type.

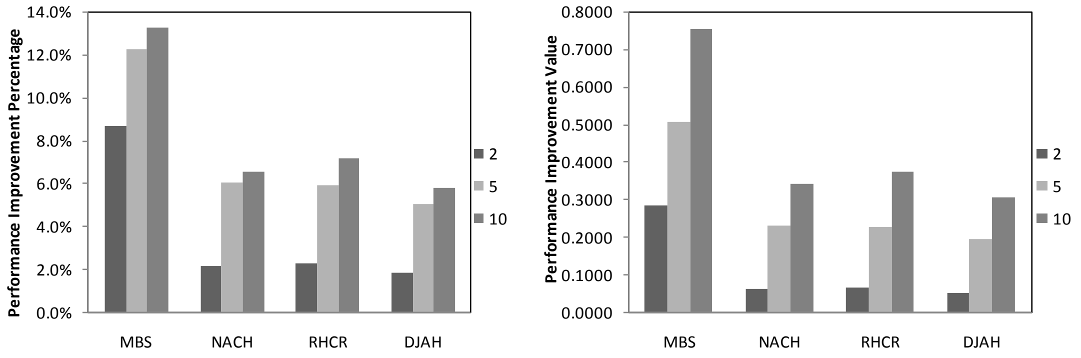

Fig. 3.4. Performance improvement over benchmark strategies with the number of product types

The performance improvement obtained by NARCH is affected negatively while the traffic intensity of the batch processor increases (see Figure 3.5). With higher traffic intensities, the number of products waiting in the batch processor’s buffers at decision points becomes larger and full batch situations are observed often. This leads to more start decisions with each benchmark strategy since all of the control heuristics give priority to full batch product types. Even in partial batch situations, start decisions are suggested more often by the first stage of the NARCH algorithm due to the higher total delay caused by the products available in the buffers. Since the current job on the serial processor cannot be changed, re-sequencing does not help in shortening the next arrival time enough to make wait decisions preferable in the first stage.

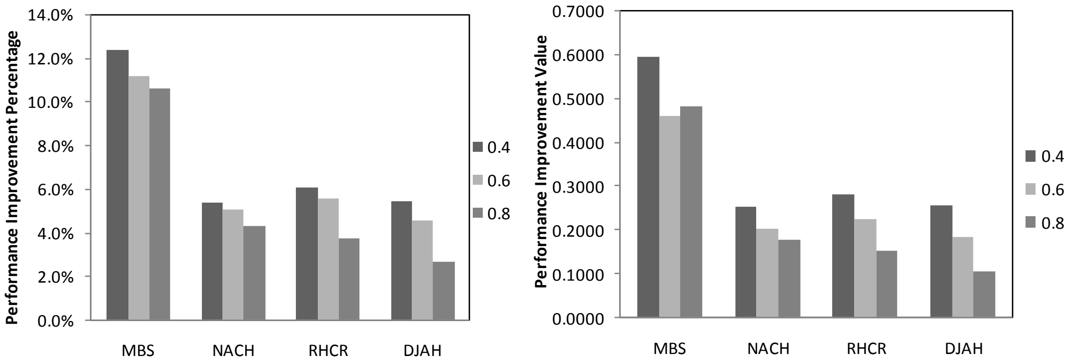

Fig. 3.5. Performance improvement over benchmark strategies with the batch processor traffic intensity

Batch processor capacity requirements typically differ between product types. This case is considered to be an alternative capacity setting and its impact on the control of batch processors is observed. Simulation results indicate that normalized waiting times increase when the batch processor capacity is different for product types. However, the performance improvement obtained by NARCH is not affected by this situation (see Figure 3.6). In fact, improvements over MBS significantly increase with the unbalanced capacity setting. This is due to the fact that control limit approaches work better when maximum batch sizes are equal for different product types.

Results also indicate that unbalanced product-mix values reduce the normalized waiting time values. Similar arguments used in the discussion of number of product types are applicable in product-mix settings. As some product types become more prevalent than other product types, the system behaves as if there are fewer product types. This situation results in better normalized cycle time values. Process times also have similar effect when the process times are different for product types. The improvement obtained by NARCH is not affected negatively in unbalanced product-mix and process time settings (see Figure 3.7 and Figure 3.8).

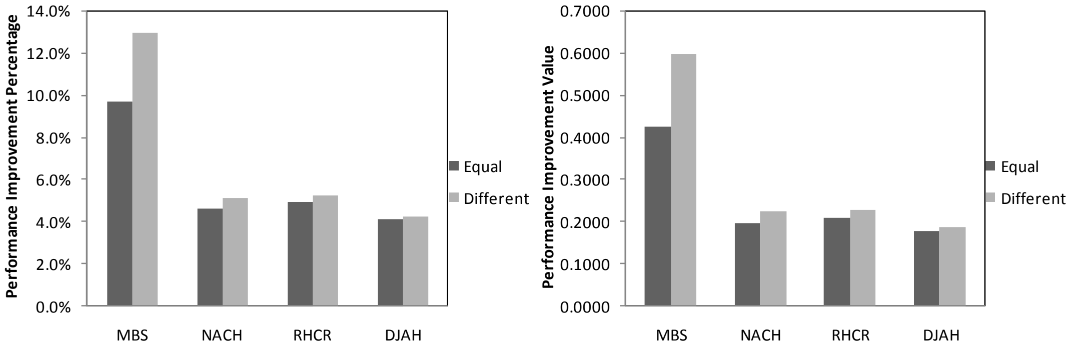

Fig. 3.6. Performance improvement over benchmark strategies with the batch processor capacity settings

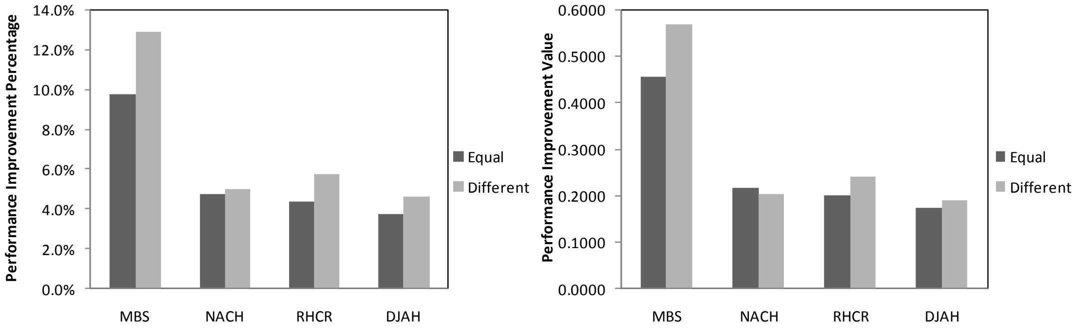

Fig. 3.7. Performance improvement over benchmark strategies with the product-mix settings

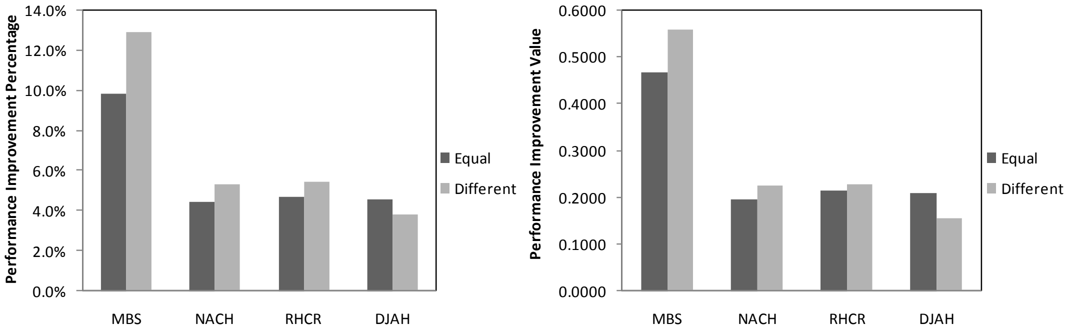

Fig. 3.8. Performance improvement over benchmark strategies with the batch process time settings

3.4.3 Complexity of the NARCH algorithm

The complexity of NARCH needs to be studied to see if the algorithm is efficient for on-line control. The algorithm starts by checking for available full batches. A maximum of N (the number of product types) comparisons are performed to check for full batch product types. For each full batch product type, the arrival points within the decision horizon and the products waiting in the buffers are used to determine the mean waiting time metric. Assume M is the upper bound on the number of arrival points in the evaluation time-window. At a decision point, assuming the batch processor's buffers are steady, there is an upper bound for the number of available products in the buffers. K is the upper bound for the number of product types waiting in the buffers. Then a maximum of N(M+K) operations is required to calculate the metric values, and N-1 comparisons are run to find the minimum metric value. Hence, full batch condition is bounded by N(M+K+2) operations.

In case of partial batches, the following operations are performed to reach a decision alternative and its corresponding mean waiting time metric value for each product type. M is again the upper bound for the number of arrivals to be considered for re-sequencing. For each re-sequencing decision, M operations are required to update the arrival times. If B is the maximum value in product type capacities of the batch processor (i.e. $B_j \leq B$ for all product type j), then a maximum of B operations is required for the total gain/loss calculation of the wait decisions. Then, a maximum of M comparisons are performed to find the maximum total gain/loss value and its comparison with 0. Combining all steps of the partial batch case, $M^2+B+M = M(M+1)+B$ is the upper bound for the number of operations required to suggest a decision alternative for a product type. K is again the upper bound for the number of available products in the buffers. The evaluation in the second stage requires M operations for the future arriving products and K operations for the available products to determine the metric value of a decision alternative. Combining with the first stage analysis, a total of M(M+2)+B+K operations is enough to reach D_j value of product type j. Since there are a maximum of N alternative decisions and the final comparison of the decision alternatives requires N-1 operations, the upper bound of the required operations becomes N(M(M+2)+B+K+1). Therefore, the complexity of the NARCH algorithm is in the form of O(N). Here, B, K and M are upper bounds for batch capacity, number of available products and number of arrivals in the decision time-window, respectively, and are independent of N.

Table 3.4 summarizes the CPU time of the simulations in seconds. Simulation time of the strategies that use future arrival information increases linearly with the number of product types, whereas MBS shows an exponential increase. Searching the best MBS level requires testing all combinations of possible minimum batch size levels with the simulation. A product type can have an MBS level from 1 to B. Then, the best

MBS level can be searched with $O(B^N)$ complexity. Since analytic models do not exist for determining the best MBS levels, MBS policy is not practical for multiple product type cases.

Table 3.4. Simulation times (in seconds) with respect to number of product types

Simulation Time

Number of Products	MBS	NACH	RHCR	DJAH	NARCH
2	4.64	0.31	0.41	0.37	0.45
5	56.44	0.54	0.80	0.74	0.83
10	19074.34	1.24	1.76	1.70	1.72

3.5 Contribution of the chapter

The research presented in this chapter contributes to the control of batch processors in the following manner:

i) It is experimentally shown that there is a potential benefit in controlling upstream stations to improve the cycle time performance of the batch processors. Upstream stations make two important contributions to batch process decision making: providing future arrival information for the batch processor, and incorporating the batch processor’s benefit in determining the sequence of the products. The first contribution is explored extensively by look-ahead batching literature. However, this is the first research combining the second contribution of upstream stations with the look-ahead batching framework using the re-sequencing method described in this chapter.

ii) In order to demonstrate the effect of upstream re-sequencing, a serial-batch processor system is modeled and a control strategy, namely NARCH, is devised specifically for this system. Although it is not possible to re-sequence the product being processed by the serial processor, results show a significant performance improvement with NARCH. Performance improvements gained by NARCH increase with the number of product types, which highlights the applicability of the re-sequencing approach considering the high diversity of product types in wafer fabrication. NARCH also obtains better performance improvement with moderate and low batch processor traffic intensity. This result is important in the sense that NARCH can be effectively used in continuous control of batch processors where there are high variations in product flow. Since the batch processors aim to follow full batch policy with high traffic intensities, a look-ahead based control strategy is more active when the traffic intensity is low or moderate since non-trivial decision making is performed more often in these cases.

iii) The NARCH algorithm runs with O(N) complexity (N being the number of product types), which is efficient to handle large number of product types. With such complexity, NARCH can be implemented for real-time control of batch processors in wafer fabrication.

CHAPTER IV

CONTROLLING DELIVERY PERFORMANCE OF BATCH PROCESSORS USING LOOK-AHEAD BATCHING

This chapter discusses the use of future arrival information in the long-run control of batch processors with due-date related objectives. The objective is to minimize the mean tardiness of a single batch processor in the long-run. Two on-line control strategies are proposed for the problem. These control strategies are the first control approaches that combine look-ahead batching with a due-date related metric. The upstream station of the batch processor is incorporated into the decision making process. The first strategy uses product sequence information at the upstream station to incorporate the arrival time and due-date information of the upcoming products in batching decisions. The second strategy extends the first strategy through a re-sequencing approach that takes place at the upstream station when there is a benefit in shortening the arrival time of a critical product.

4.1 Introduction

The competitive behavior of the semiconductor market increases the importance of customer related performance measures, in management's perspective. It is a very challenging task to meet customers' due-date expectations in the wafer fabrication industry. In order to sustain an important market share, companies need to manage their production in the best way to meet product due-dates. However, the complicated properties of wafer production make this task very challenging. The control of batch processors is one of the most important tasks among the complications described in Chapter I.

Early research in the control of batch processors attempts to fulfill internal manufacturing objectives such as minimizing cycle time to reduce manufacturing costs. Chapter II provided a review of these studies and Chapter III studied the dynamic control of batch processors from this perspective with cycle time performance measure. However, recent research directions in the literature address customer related objectives such as on-time delivery of the final products. In this chapter, the focus of dynamic control is on due-date related performance of batch processors. The choice of performance measure of the control task is mean tardiness of the products which is a good indicator of the on-time delivery performance in the long-run. Once a due-date is determined for a customer order, the due-date for each intermediate processing step can be determined using the routing information of the products that are ordered. This way, on-time delivery of an intermediate processing step can be studied by the production planners. For a long-run (monthly/quarterly) evaluation of customer satisfaction, mean tardiness of products is a reasonable indicator for overall production performance as well as for the performance of an intermediate processing step, such as a batch processing step in the front end of wafer production.

In this chapter, a look-ahead batching framework is exploited for mean tardiness performance. Similar to Chapter III, future information about upcoming products is provided by the serial processor at the upstream processing step. The arrival and due-date information of the products waiting in the serial processor station makes up the future information used in the decision making. Two look-ahead batching strategies are proposed to control the batch processor. The objective is to minimize mean tardiness of the products from their batch process due-dates. The first control strategy relies on the use of available future information at the upstream serial processor station, whereas the second control strategy further incorporates the upstream control through a re-sequencing approach. The purpose of re-sequencing is to shorten the arrival time of an urgent product by pulling it to the front of the queue. This chapter contributes by providing a mathematical endeavor in the use and change of upstream sequence for controlling the serial-batch processor system with respect to mean tardiness performance. The rest of the chapter is organized as follows. The problem definition is given in Section 4.2. Section 4.3 presents the modification of popular time-window approaches for the serial-batch setting. The proposed control strategies are described in Sections 4.4 and 4.5. A simulation based comparison of the proposed strategies with the benchmarks is provided in Section 4.6. The contribution of the chapter is discussed in Section 4.7.

4.2 Problem definition and notation

The properties of the serial-batch processor described in Chapter III also apply in this chapter. Figure 4.1 illustrates the serial-batch processor system for two product types. Each product carries a due-date to meet at the end of its batch process. In addition to the serial process time and serial process sequence information, the due-date information of the products is also available to the controller attached to the serial-batch processor system. There are N incompatible product types that visit the serial-batch processor system. Each product has an independent serial process time that is driven by a stochastic process. The serial processor's queue sets a limit on future arrival information for the batch processor.

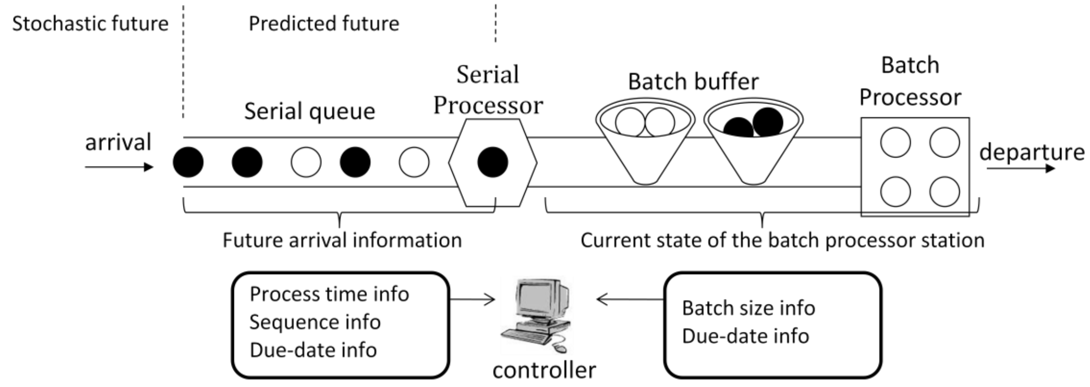

Fig. 4.1. Serial-batch processor system for Chapter IV

The objective is to minimize the mean tardiness of the products processed in the serial-batch processor system. The tardiness value of a product is given by $\max(0, C_{ij} - dd_{ij})$. Here, C_{ij} is the batch process completion time of the i^{th} product of type j. Similarly, dd_{ij} is the batch process due-date assigned to i^{th} product of type j. Products are assumed to be equally important, so there are no priority weights. The following notations are used in the rest of the chapter:

N = number of product types
T_j = batch processing time for product type j
B_j = batch processor capacity for product type j
P_j = ratio of product type j in the mix

d_{ij} = due-date of the product i of type j
r_{ij} = batch process ready time of the product i of type j
a_{ij} = time that the product i of type j arrives at the serial processor station
t_0 = current decision point
t_{jn} = time that n^{th} upcoming product of type j will arrive at the batch processor's buffers
K_{j0} = the union of the products that are in the batch processor's buffers at t_0 and the products that will arrive within a time-window of length T_j
K_{jn} = the union of the products that are in the batch processor's buffers at t_0 and the products that will arrive within a time-window of length $t_{jn}+T_j-t_0$
τ_{j0} = average time spent in the batch processor station by the remaining batches in K_{j0} after loading product type j at t_0
τ_{jn} = average time spent in the batch processor station by the remaining batches in K_{jn} after loading product type j at t_{jn}
q_j = number of products of type j in the batch processor's buffers at t_0
q_{ij0} = number of products of type i in the set K_{j0}
q_{ijn} = number of products of type i in the set K_{jn}
n_b = number of products in batch b
MT_{j0} = mean tardiness metric value caused by starting batch β_{j0} at t_0
MT_{jn} = mean tardiness metric value caused by starting batch β_{jn} at t_{jn}

4.3 Modification of time-window based prioritization rules

In this section, BATC-I and BATC-II batch priority indexing rules, developed by Monch, et al. (2005), are modified for the serial-batch processor system setting. These two rules are selected first, due to their effective performance in minimizing total tardiness and second, due to their time-window approach. The time-windows used in these rules require local near-future information instead of full deterministic future information. Besides, they are known as the best performing heuristic methods for the control of batch processors with incompatible product types and dynamic arrivals. Detailed discussion on these rules can be found in Chapter II. The reason for the modification is to add a dynamic control framework. The modification involves limiting future arrival information with the upstream serial processor and adding a rolling horizon decision making mechanism.

For a particular product type j, there is no benefit to wait for the arrivals that are expected to occur outside T_j time-window, since a batch process can be completed and the processor becomes available again within that time period. Therefore, time-windows used in BATC-I and BATC-II algorithms are determined by the batch process times of the product types. All products are equally important, so priority weights of the products are all set to 1. At every decision point, one batch from each available product type is chosen; of all the considered batches, one is chosen to be the next job of the batch processor. For a particular product type, the best batch is formed using the priority indices of the products determined by a dynamic version of the ATC rule developed by Vepsalainen (1987).

i) Modified BATC-I:

The detailed modification of the BATC-I algorithm is the following. At a decision point t_0, assuming q_j products of type j are available in the batch processor's buffers, one of the alternative decisions is to start the batch process with these q_j products without waiting for any future arrivals. In this alternative decision, if $q_j > B_j$, the controller ranks the q_j products by ATC priority index given by equation (4.1) and forms the full batch β_{j0} using the first B_j products. In ATC indexing, the highest priority is given to the minimum slack product by $(d_{ij} - T_j - t_0)^+$. The parameters k and p are described later.

$$I_{ij,ATC} = \left(\frac{1}{T_j}\right) exp\left(-\frac{\left(d_{ij} - T_j - t_0\right)^+}{kp}\right) \quad (4.1)$$

There is no need to prioritize the products if $q_j \leq B_j$. β_{j0} is formed with all q_j products in this case. The priority index of β_{j0} is determined by equation (4.2). In the equation, the common due-date of the batch β_{j0}, $d_{\beta_{j0}}$, is the minimum due-date of the q_j products in the batch (i.e. $d_{\beta_{j0}} = \min(d_{ij} \,|ij \in \beta_{j0})$). In this indexing method, the priority of a batch is mainly dependent on its slack value. In addition to the slack value, the fullness of a batch is also accepted as an important factor for the priority of the batch. k is a look-ahead parameter used for scaling purposes. p is the average batch processing time of the products that do not go into β_{j0} that are either waiting in the batch processor's buffers or expected to arrive before the batch processor's next available point, t_0+T_j. According to this indexing, the most prior batch alternative is selected for each product type available in the batch processor's buffers at t_0.

$$I_{\beta_{j0}} = \left(\frac{1}{T_j}\right) exp\left(-\frac{\left(d_{\beta_{j0}} - T_j - t_0\right)^+}{kp}\right)\left(\frac{n_{\beta_{j0}}}{B_j}\right) \quad (4.2)$$

In addition to t_0, the arrival points of product type j within T_j time-window define the set of alternative batch start points for product type j. Assume that there are R_j products of type j expected to arrive within the time-window of length T_j. For a particular future arrival point of product type j, the products of type j expected to be available on or before this arrival point are included to determine the highest priority batch composition. For example, consider the case that the n^{th} upcoming product of type j is expected to arrive within T_j time units, i.e. $t_{jn} - t_0 \leq T_j$. If $q_j + n \leq B_j$, batch β_{jn} is formed with the q_j products available at t_0 and the n arrivals expected to occur within (t_{jn}-t_0) time units. On the other hand, if $q_j + n > B_j$, then for each product i in these q_j+n products, index I_{ij} is calculated using equation (4.3). This prioritization index is the modified version of the static ATC index given by equation (4.1), and briefly discussed by Monch, et al. (2005). It should be noted that the additional amount $(r_{ij}\text{-}t_0)^+$ in the exponent reduces the priority of a product if the product has a ready time (arrival time) that is later than t_0. The bigger the amount the less priority is assigned to the product.

$$I_{ij} = \left(\frac{1}{T_j}\right) exp\left(-\frac{\left(d_{ij} - T_j - t_0 + (r_{ij} - t_0)^+\right)^+}{kp}\right) \tag{4.3}$$

I_{ij} indices are used to select the highest priority B_j products out of q_j+n products to form the batch β_{jn}. β_{jn} represents the best batch alternative of product j for the n^{th} arrival point. The priority of the batch, $I_{\beta_{jn}}$, is determined by equation (4.4). The term $(r_{\beta_{jn}} - t_0)^+$ takes the ready time of the batch into consideration. $r_{\beta_{jn}}$ is the ready time of the batch β_{jn}, and equals to the largest ready time of the products in the batch (i.e. $r_{\beta_{jn}} = \max(r_{ij} \,|ij \in \beta_{jn}))$. Considering the arrival points within the time-window (t_0,t_0+T_j), a maximum of R_j+1 batch alternatives can be formed for product type j. Out of these alternative batch formations, the one with the highest priority is selected as the best batch formation of product type j at t_0. The priority index of the best batch formation is found by $I_j = I_{\beta_{jn^*}}$ where $n^* = \text{argmax}\left(I_{\beta_{jn}} \middle| n = 0,1,\dots,R_j\right)$. The I_j value is saved to compare with other product types' priority indices. If $n^* > 0$ then the suggested decision alternative for product type j is to wait for the next decision point to repeat the analysis. On the other hand, if $n^* = 0$ then the suggested decision alternative is to start the batch process with β_{j0} at t_0. The same analysis is completed for each product type and I_j values and corresponding decisions are saved. The winner product type $j^* = \text{argmax}(I_j \,|\, j = 1,2, \ldots N)$ is found from all the saved I_j values. If the corresponding n^* value for the winner product type j^* is greater than 0, the decision is to keep the batch processor in the waiting mode until the next arrival point to repeat the procedure. On the other hand, if n^* is equal to 0, the decision is to start the batch process with β_{j^*0} at t_0. This adds a rolling horizon behavior to the algorithm.

$$I_{\beta_{jn}} = \left(\frac{1}{T_j}\right) exp\left(-\frac{\left(d_{\beta_{jn}} - T_j - t_0 + (r_{\beta_{jn}} - t_0)^+\right)^+}{kp}\right)\left(\frac{n_{\beta_{jn}}}{B_j}\right) \tag{4.4}$$

ii) Modified BATC-II:

The modified BATC-II algorithm follows the same steps of the modified BATC-I except the indexing equations (4.2) and (4.4) are replaced with equations (4.5) and (4.6) respectively. BATC-I uses a united priority index for a batch while in BATC-II, the priority index of a batch is determined by the sum of the priority indices of the products composing the batch.

$$I_{\beta_{j0}} = \sum_{i=1}^{n_{\beta_{j0}}} \left(\left(\frac{1}{T_j}\right) \exp\left(-\frac{\left(d_{ij} - T_j - t_0\right)^+}{kp}\right)\left(\frac{n_{\beta_{j0}}}{B_j}\right)\right) \tag{4.5}$$

$$I_{\beta_{jn}} = \sum_{i=1}^{n_{\beta_{jn}}} \left(\left(\frac{1}{T_j}\right) \exp\left(-\frac{\left(d_{ij} - T_j - t_0 + (r_{\beta_{jn}} - t_0)^+\right)^+}{kp}\right)\left(\frac{n_{\beta_{jn}}}{B_j}\right)\right) \tag{4.6}$$

4.4 Next arrival control heuristic with tardiness measure (NACH-T)

In this section, a new control strategy is proposed for the serial-batch processor system. The proposed approach, namely NACH-T, is a dynamic control strategy that combines and modifies the components of DJAH strategy developed by Van Der Zee, et al. (1997) and mean tardiness metric described by Monch, et al. (2005). NACH-T is the first look-ahead batch control approach that uses a due-date related performance metric to evaluate alternative batching decisions. The arrival times and the due-dates of the upcoming products are utilized in the algorithm using a look-ahead framework. A rolling horizon approach is followed to improve the quality of the decisions. NACH-T follows a 3-step algorithm.

Step 1: *Suggest individual decisions for each product type excluding the effect of these decisions on other product types, then go to Step 2.*

In this step, each product type that is available in the batch processor's buffers is reviewed individually. A decision alternative is suggested for each product type avoiding other product types in the batch processor's buffers and their future arrivals. This way, a set of alternative decisions is composed for Step 2 analysis.

At a particular decision point t_0, NACH-T starts by checking the batch processor's buffers for full batches. For the full batch product types, the suggested decision is to start the batch process. On the other hand, further analysis is required to suggest decision alternatives for the partial batch product types. The following procedure is followed for each partial batch product type. For a particular partial batch product type j, the controller checks the time-window of length T_j for any expected arrivals of product type j. Consider the case that product type j has R_j future arrivals that are expected to occur in the time interval (t_0, t_0+T_j). Then $\min(B_j - q_j, R_j)$ of these arrivals compose the set of alternative points to which the batch process of product type j can be postponed. In the algorithm, these $\min(B_j - q_j, R_j)$ arrival points are called feasible arrival points since each of them provides an alternative batch formation. For a particular feasible arrival point, say n^{th} arrival that is expected to occur at t_{jn}, where $t_{jn} \leq t_0 + T_j$ and $n \leq \min(B_j - q_j, R_j)$, the total tardiness gain/loss (M_{jn}) of waiting for this arrival is calculated by equation (4.7).

$$M_{jn} = \left[\sum_{i=1}^{q_j}(t_0 + T_j - d_{ij})^+ \quad + \sum_{i=q_j+1}^{q_j+n}(t_0 + 2 \times T_j - d_{ij})^+\right] - \left[\sum_{i=1}^{q_j+n}(t_{jn} + T_j - d_{ij})^+\right] \tag{4.7}$$

M_{jn} is the difference between the total tardiness of q_j+n products in two cases: starting the batch process at t_0 and starting the batch process at t_{jn}. The first portion of the formulation determines the total tardiness of q_j+n products if the batch process starts at t_0 with q_j products, while the second portion determines the total tardiness of q_j+n products if the batch process starts at t_{jn} with q_j+n products. The value returned indicates whether it is worthwhile to wait for the n^{th} arrival of the product type j or is better to start a batch of this type at t_0. In the presence of positive M_{jn} values, the arrival with the maximum positive value, $n^* = \text{argmax}(M_{jn} \,|\, n = 1,2, \dots min(B_j - q_j, R_j))$, is selected. In this case, the suggested decision for product type j is to wait for its n^{*th} arrival and then start the batch process with q_j+n^* products at t_{jn^*}. In case all M_{jn} values are negative or there is no future arrival of product type *j* in the time-window of length T_j, the suggested decision is to start the batch process with q_j products at t_0.

After following this procedure for each partial batch product type, each product type remains with a single decision alternative and these decision alternatives (maximum of N) are evaluated in Step 2 to find the best alternative.

<u>Step 2</u>: *Evaluate each alternative decision by including its effects on all product types, then go to Step 3.*

The evaluation of a particular decision alternative is essentially calculating the value of the mean tardiness metric that will result in by executing the decision alternative. Future information in the decision alternative's look-ahead window is used for the evaluation. In the case that the suggested decision for product type *j* is to start the batch process with min(B_j,q_j) products at t_0, the value of the mean tardiness metric (MT_{j0}) caused by this decision is calculated using equation (4.8).

$$MT_{j0} = \left[\sum_{ij \in \beta_{j0}} (t_0 + T_j - d_{ij})^+ + \sum_{mk \in (K_{j0} - \beta_{j0})} (t_0 + T_j + \tau_{j0} - d_{mk})^+ \right] \Big/ \min(B_j, q_j) \tag{4.8}$$

$$\text{where} \quad \tau_{j0} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kj0}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jj0} - \min(B_j, q_j)}{B_j} \right\rceil . T_j \right] \Big/ 2$$

The products that are either available in the batch processor's buffers at t_0 or will arrive within T_j time units (represented by the set K_{j0}) are included in the calculation of MT_{j0}. The first portion of the equation determines the total tardiness of the products that form batch β_{j0}, while the second portion calculates the total tardiness of the remaining products in the set K_{j0} (represented by the set K_{j0} - β_{j0}). The tardiness calculation for the products in batch β_{j0} is straightforward. On the other hand, for the products in set K_{j0} - β_{j0}, the total tardiness estimate is found using the batch process time of product type j (T_j) and the estimate of the time (τ_{j0}) that the remaining batches spend in the batch processor station after the batch process of β_{j0} is completed.

In the calculation of τ_{j0}, the minimum number of batches required for each product type is found first. The total time required to process these batches is the product of the number of batches with their batch processing times. This total value is divided by two to find the mean estimate of the time that is spent in the batch processor station by the batches of set K_{j0} - β_{j0}. The mean tardiness metric is found by dividing the total tardiness value by the throughput of the batch process, min(B_j,q_j).

On the other hand, if the suggested decision for product type j is to wait for the n^{th} arrival before starting the batch process, equation (4.9) is used to calculate the value of the mean tardiness metric caused by this decision alternative. In this decision alternative, the batch process will start with β_{jn} which has q_j+n products. The decision alternative's look-ahead window becomes (t_0, $t_{jn}+T_j$), where the batch process completion time is $t_{jn}+T_j$. While calculating τ_{jn}, the number of products of type k at $t_{jn}+T_j$ is represented by q_{kjn} and the set of products that will be in the batch processor's buffers at $t_{jn}+T_j$ is represented by K_{jn} - β_{jn} in this case.

$$MT_{jn} = \Bigg[\sum_{ij \in \beta_{jn}} \left(t_{jn} + T_j - d_{ij}\right)^+ + \sum_{mk \in \left(K_{jn} - \beta_{jn}\right)} \left(t_{jn} + T_j + \tau_{jn} - d_{mk}\right)^+\Bigg] / (q_j + n) \tag{4.9}$$

$$\text{where } \tau_{jn} = \left[\sum_{k=1,k\neq j}^{N} \left(\left\lceil \frac{q_{kjn}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jjn} - (q_j + n)}{B_j} \right\rceil . T_j \right] \Big/ 2$$

Employing the same analysis for each product type, mean tardiness metric values are obtained for the set of decision alternatives. For a particular product type j, the metric value of its suggested decision is recorded as MT_j. The suggested decision for product type $j^* = \text{argmin}(MT_j)$ becomes the best decision alternative for decision point t_0.

Step 3: Take the action corresponding to the best alternative decision and exit.

If the suggested decision for j^* is to start the batch process at t_0, then the output of the NACH-T algorithm is to load batch β_{j^*0} on the batch processor with min(B_{j^*},q_{j^*}) products. On the other hand, if the suggested decision for j^* is to wait for the n^{*th} arrival to start the batch process, then the decision is postponed to the next decision point to review the system with the NACH-T algorithm again. In this case, the batch processor stays idle and the algorithm is repeated at the next decision point (arrival point). Figure 4.2 illustrates the flow of the NACH-T algorithm.

4.5 Next arrival re-sequencing based control heuristic with tardiness measure (NARCH-T)

In this section, an improved version of the NACH-T algorithm is proposed. The new look-ahead control strategy, namely NARCH-T, includes an additional control on the product sequence of the upstream serial processor. NARCH-T incorporates re-sequencing of products at the upstream serial processor to improve the batching decisions. Re-sequencing is considered only for wait decision alternatives. The purpose of the re-sequencing is to shorten an urgent (in terms of due-dates) product's arrival time at the batch processor. For a particular product type *j*, the algorithm considers the products in the serial processor station, and checks if changing the sequence position of a product improves the quality of the wait decision.

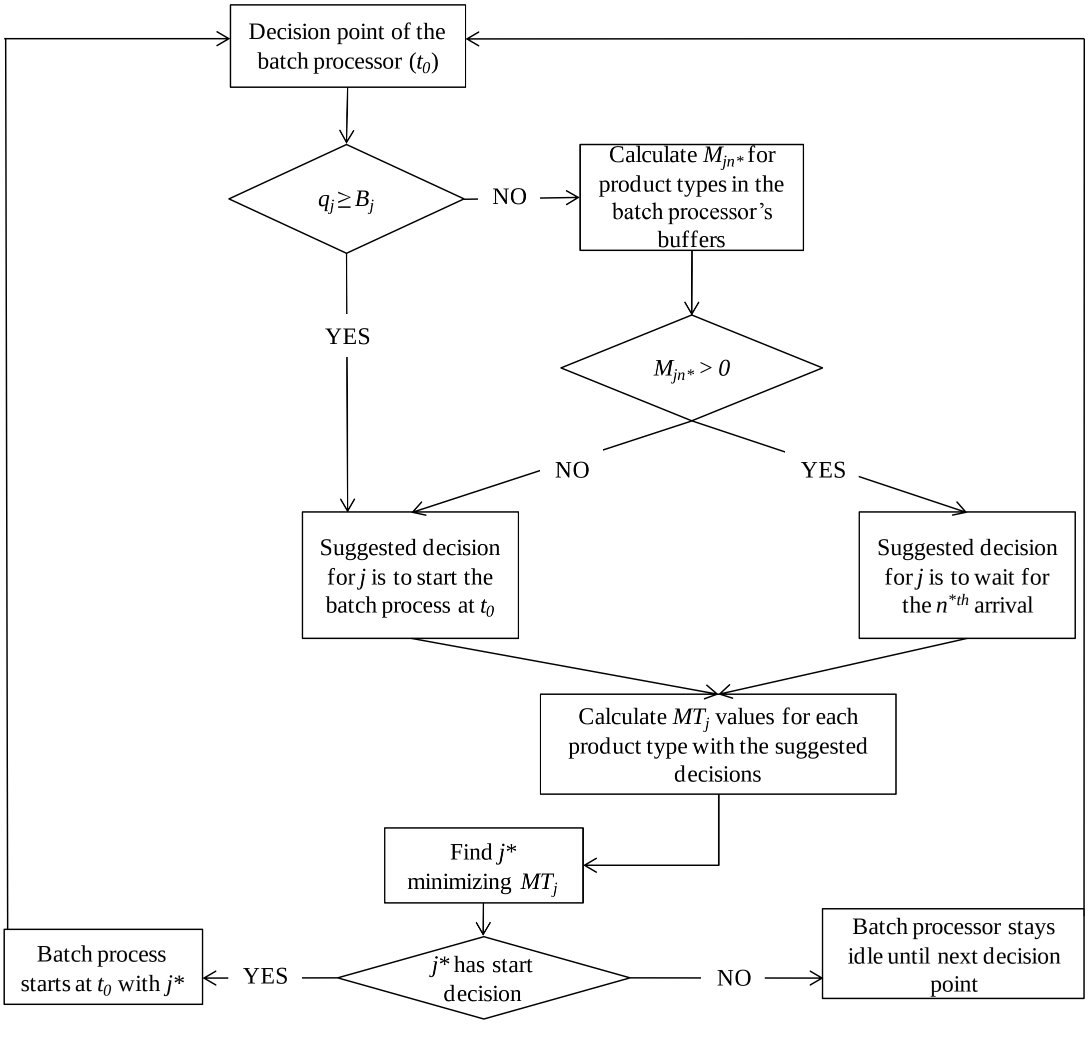

Fig. 4.2. Flowchart of the NACH-T algorithm

Step 1*: Suggest individual decisions for each product type excluding the effect of these decisions on other product types, then go to Step 2.*

At a particular decision point t_0, similar to NACH-T, NARCH-T starts by suggesting individual decision alternatives for the product types in the batch processor's buffers. In this step, the effects of the decision on other product types are excluded while assigning product type decision alternatives. The following procedure is employed to find the suggested decisions for each product type.

For a particular product type j available in the batch processor's buffers at t_0, the algorithm checks if the product type has a full batch. In case product type j has a full batch, the suggested decision for product type j is to start the batch process at t_0. On the other hand, if product type j has a partial batch, further analysis is performed as the followings. The algorithm reviews the serial processor station for any expected arrivals of the same type. Assuming there are R_j products of type j available in the serial processor station, each of these R_j products is a possible candidate to be the next job on the serial processor through re-sequencing. Each of these R_j products are assumed to be the first product in the sequence through re-sequencing and the total tardiness gain/loss values of taking these actions are calculated to find the upcoming product that is more beneficial to wait for.

By indexing in the increasing order of the products' original arrival times, the algorithm employs the following steps to reach a decision for product type j. For the n^{th} upcoming product of type j, consider M_{jn} value as the total tardiness gain/loss for pulling this product to the front of the sequence and waiting for its arrival. To find M_{jn}, the arrival time t_{jn} is updated as equal to the sum of the product's expected processing time and the remaining time of the serial processor's current job. M_{jn} is calculated using equation (4.10). The main difference between equations (4.7) and (4.10) is the number of additional arrivals of product type *j* that is considered in the calculations. Since NACH-T does not involve a re-sequencing approach for the upstream serial processor, it includes the effect of all *n* future arrivals for the evaluation of the n^{th} arrival point, while NARCH-T considers only one arrival by pulling the n^{th} product to the front of the serial processor's queue. The first portion of equation (4.10) calculates the total tardiness value for q_j+1 products assuming the batch process starts with q_j products at t_0. The second portion assumes that the batch process starts with q_j+1 products at updated (re-sequenced) t_{jn}. Same steps are repeated for each of the R_j products, and M_{jn} values are found.

$$M_{jn} = \left[\sum_{i=1}^{q_j}\left(t_0 + T_j - d_{ij}\right)^+ \quad + \left(t_0 + 2 \times T_j - d_{ij}\right)^+\right] - \left[\sum_{i=1}^{q_j+1}\left(t_{jn} + T_j - d_{ij}\right)^+\right] \tag{4.10}$$

In the case of at least one positive M_{jn} value, the suggested decision for product type *j* is to re-sequence the serial processor's queue by pulling the n^{*th} product of type *j* to the front of the queue where $n^* = \text{argmax}\left(M_{jn} \,\middle|\, n = 1,2,..R_j\right)$ and then to wait for its arrival before starting the batch process. If all M_{jn} values returned are less than or equal to 0 or there are no products of type *j* available in the serial processor station, then the suggested decision for product type *j* is to start the batch processor with q_j products at t_0.

The same procedure is repeated for all product types available in the batch processor's buffers. This way, a particular decision alternative is suggested for each product type to be evaluated in Step 2.

<u>Step 2</u>*: Evaluate each alternative decision by including its effects on all product types, then go to Step 3.*

In this step, the decision alternatives suggested for each product type are evaluated by including the decisions' effects on all the other product types. The following strategy is employed for each decision alternative. If the decision alternative related to product type j is to start the batch process at t_0, then the value of the mean tardiness metric (MT_{j0}) caused by starting the batch process with $\min(B_j,q_j)$ products is determined by equation (4.8) given in the previous section. On the other hand, if the suggested decision is to re-sequence the products in the serial processor's queue to make n^{*th} product of type j be the next job on the serial processor and wait for its arrival at the batch processor, then t_{jn*} is updated as the sum of the product's serial process time and the remaining time of the current job on the serial processor. The arrival times of the products whose positions are initially in front of the product being pulled are updated accordingly while the arrival times of products whose positions are initially behind remain the same. After updating the arrival times in the look-ahead window, the value of the mean tardiness metric (MT_{jn*}) caused by starting the batch process at updated t_{jn*} with q_j+1 products (q_j products available in the batch processor's queue and the product that will arrive at updated t_{jn*}) is determined using equation (4.11).

$$MT_{jn^*} = \left(\sum_{ij \in \beta_{jn^*}} \left(t_{jn^*} + T_j - d_{ij}\right)^+ + \sum_{mk \in \left(K_{jn^*} - \beta_{jn^*}\right)} \left(t_{jn^*} + T_j + \tau_{jn^*} - d_{mk}\right)^+ \right) / \left(q_j + 1\right) \tag{4.11}$$

$$where \ \ \tau_{jn^*} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kjn^*}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jjn^*} - (q_j + 1)}{B_j} \right\rceil . T_j \right] \Big/ 2$$

After employing the same analysis for each product type, mean tardiness metric values are obtained for all decision alternatives. For a particular product type j, the metric value of its suggested decision is recorded as MT_j. The suggested decision for the product type $j^* = \text{argmin}(MT_j)$ becomes the best decision for decision point t_0.

<u>Step 3</u>: Take the action corresponding to the best alternative decision and exit.

If the suggested decision for j^* is to start the batch process at t_0, then the output of the NARCH-T algorithm is to load β_{j*0} on the batch processor with min(B_{j*},q_{j*}) products. On the other hand, if the suggested decision for j^* is to wait for a future arrival, then the serial processor's queue is re-sequenced with the underlying decision suggested for $j*$ and the batch processor stays in the waiting mode until the next decision point. The batch decision is reviewed at the next decision point in this case. Figure 4.3 illustrates the flow of the NARCH-T algorithm.

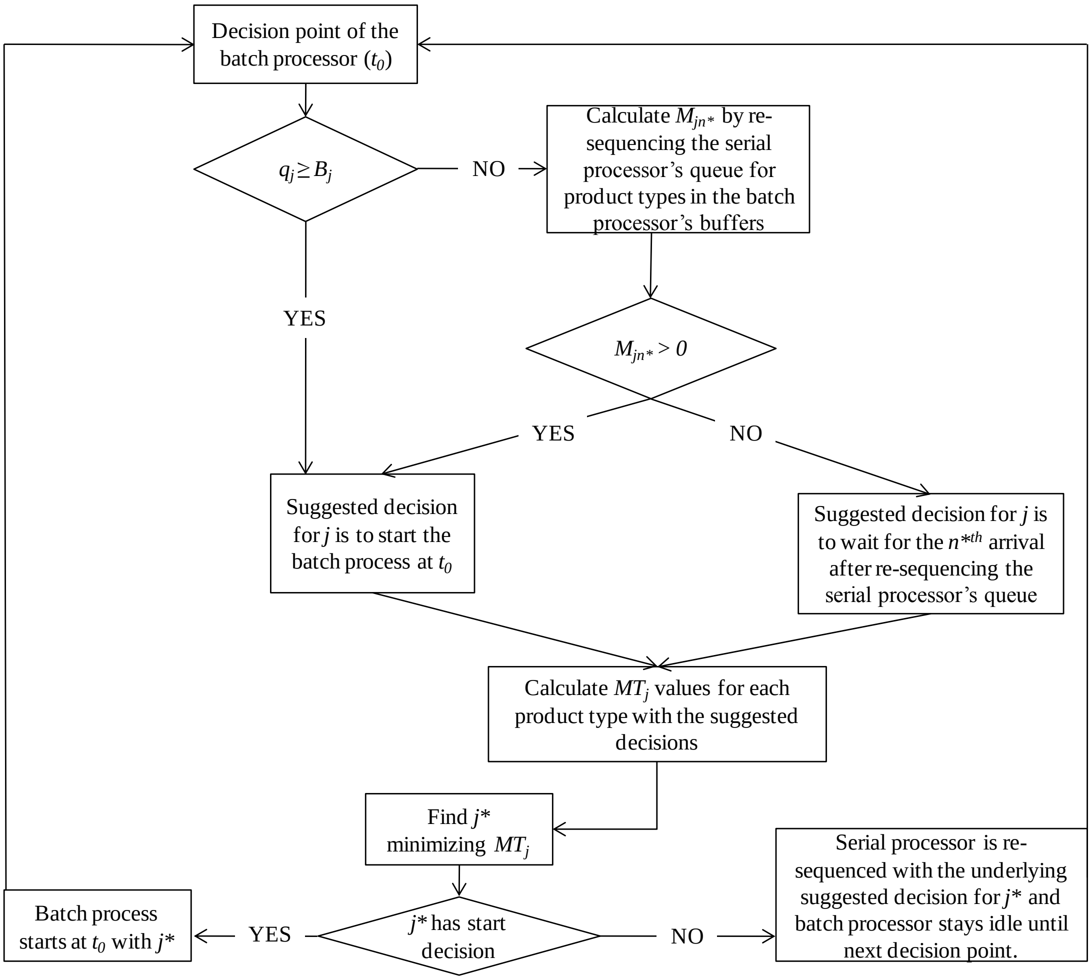

Fig. 4.3. Flowchart of the NARCH-T algorithm

4.6 Simulation study

In this section, a simulation based comparison of the proposed batch control strategies with the benchmarks is provided. There are 4 benchmark control approaches that are specifically developed for the tardiness related criteria. Two of the benchmarks, BATC-I and BATC-II, were discussed in detail in Section 4.3. The selection of the *k*-value is very important for the performance of BATC-I and BATC-II. 10 different *k*-

values from 0.5 to 5 in increments of 0.5 are used in the experiments similar to Mehta and Uzsoy (1998) and Monch, et al. (2005). For every test instance, each *k* value is tested and the one with the best performance is selected. In addition to BATC-I and BATC-II rules, two more benchmark control strategies, namely MDBH and EDD (Earliest Due-date) are included in the simulation study. MDBH strategy, which is proposed by Kim, et al. (2001), is an adaptation of the DBH heuristic developed by Glassey and Weng (1991) for due-date related objectives. The details of this approach were presented in Chapter II. EDD is also a common rule used to prioritize different batch formations. According to EDD strategy, a batch process starts immediately if there is any available product in the batch processor's buffers. The batch with the minimum average due-date is selected to be loaded on the batch processor.

The simulation model of the serial-batch processor system has the same attributes described in Chapter III. There is an additional due-date assignment of the products which is not required by the control strategies in Chapter III. Once a product arrives at the serial processor station, its serial process time and its batch process due-date are assigned immediately. Serial processing time, due-date and sequence information of the products waiting in the serial processor's queue are available to the controller attached to this system.

Simulation scenarios are created by the combinations of product, process and processor characteristics that have been identified to have profound effects on the quality of the batch process control. These characteristics and their corresponding settings used in this chapter are listed in Table 4.1. Most of these production characteristics were described in Chapter III, except the ratio of urgent products. Products are divided into two categories based on their due-date assignments: urgent and normal products. The reason for this categorization is to control the ratio of tight due-date products. Urgent products have tighter due-dates than normal products. This is obtained by equation (4.14), which assigns the batch process due-date of the products. This due-date assignment rule is similar to those in Akcali, et al. (2000) and Gupta, et al. (2004). Once a particular product arrives at the serial processor station, its due-date is assigned by adding the randomly generated term to its arrival time a_{ij}. If the product is categorized as an urgent (or hotline) product, a Uniform distribution with a smaller mean, as compared to normal products, is used. The choice of means in the Uniform distributions is derived from the simulation results in Chapter III. The ranges of both distributions are the same. A 20% setting is used for the ratio of urgent products.

$$d_{ij} = a_{ij} + T_j \times \begin{cases} Uniform(-2,4) & \text{if the product is urgent} \\ Uniform(2,8) & \text{if the product is not urgent} \end{cases} \tag{4.14}$$

For a particular scenario, the arrival rate for the batch processing station, λ, is found by equation (3.7), as discussed in Chapter III. The arrival rate for the serial processor is the same as the arrival rate for the batch processor, λ. The service rate of the serial processor, μ, is selected as $\mu = \lambda/0.8$ similar to Chapter III. Exponential distribution with parameters λ and μ is used for inter-arrival and service time distributions respectively.

Table 4.1. Configuration of the simulation study for Chapter IV

No	Factor	Levels			
1	Control Strategy	BATC-I			
		BATC-II			
		MDBH			
		EDD			
		NACH-T			
		NARCH-T			
2	Number of Products	2			
		5			
		10			
3	Interarrival Distribution	Exponential			
4	Ratio of Urgent Products	0.2			
			2 Products	5 Products	10 Products
5	Product Mix	Equal	(0.5, 0.5)	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different	(0.7, 0.3)	(0.35, 0.35, 0.1, 0.1, 0.1)	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
6	Capacity By Product	Equal	(5, 5)	(5, 5, 5, 5, 5)	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different	(7,2)	(7, 6, 5, 4, 3)	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
7	Batch Processing Time By Product	Equal	(25, 25)	(25, 25, 25, 25, 25)	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different	(40, 10)	(40, 30, 25, 20, 10)	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
8	Traffic Intensity	0.4			
		0.6			
		0.8			

All settings are combined to create different problem instances (simulation scenarios). A combination of the settings gives a total of 72 (3^2x2^3) scenarios on which the control strategies are tested. For a particular strategy, the control of each scenario is simulated with a duration of 100,000 time units, a warm-up of 5,000 time units and 10 replications. The simulation code is developed using VB.net and the scenarios are simulated on a Pentium Dual Core 1.73 GHz. processor with 2GB RAM. Since NARCH-T is an extension of NACH-T strategy, with an additional re-sequencing feature, the comparison of NACH-T with benchmark strategies is presented first, followed by the comparison of NARCH-T with NACH-T.

4.6.1 Comparison of NACH-T with the benchmark strategies

In this section, the performance of NACH-T is compared with the four benchmark strategies described earlier. An overview of the simulation results is provided in Table 4.2. The first ten columns report the average and the half-width of the 95% confidence interval of the normalized tardiness values which were obtained by controlling the batch processor with the strategy specified in the top row. Normalized tardiness of a product is its tardiness time divided by its batch process time. The half-width confidence interval (hmax) is the maximum half-width of the 95% confidence intervals observed in the simulation scenarios defined by the second column.

A paired-*t* approach is used with a 95% confidence interval to test the statistical validity of the performance improvements gained by NACH-T, compared to the benchmarks (see appendix-B for detailed analysis). The last four columns in Table 4.2 provide the percentage improvements. Overall results indicate that the best performing model out of the five alternatives is the NACH-T approach. This is mainly due to the mean tardiness metric used in NACH-T. MDBH is the closest performing strategy since it is the only benchmark that evaluates decision alternatives by considering their effects on all product types. EDD is a no-idling rule which does not allow waiting for additional arrivals and due-date related priorities are used only for choosing the starting batch. On the other hand, BATC-I and BATC-II priority rules use due-date related measures in the decision making, but their drawback is that the effects of other product types are not included when a priority value is given to a particular batch. Results also show that BATC-II outperforms BATC-I marginally, and both BATC-I and BATC-II outperform EDD by about 5%. These results are consistent with the results presented by Monch, et al. (2005). The results also indicate that there is not an increase in the variance of the normalized tardiness values when NACH-T is used since the half-width 95% confidence

Table 4.2. Summary of the simulation results: NACH-T is compared with the benchmark control strategies, the last four columns show the percentage improvements obtained by NACH-T

		EDD		BATC-I		BATC-II		MDBH		NACH-T					
No	Average at:	mean	hmax	mean	hmax	mean	hmax	mean	hmax	mean	hmax	Δ1	Δ2	Δ3	Δ4
1	Number of Products = 2	1.5606	0.0240	1.5966	0.0215	1.5536	0.0231	1.4317	0.0281	1.3956	0.0359	10.4%	12.0%	9.2%	2.8%
2	Number of Products = 5	2.0943	0.0272	2.0344	0.0203	1.9842	0.0203	1.8130	0.0223	1.7203	0.0210	17.1%	15.5%	13.3%	5.0%
3	Number of Products = 10	3.9577	0.0404	3.3823	0.0232	3.3260	0.0275	3.2687	0.0187	2.9598	0.0162	22.8%	12.3%	10.8%	8.9%
4	Traffic Intensity = 0.4	2.5455	0.0301	2.5989	0.0215	2.5689	0.0231	2.4463	0.0281	2.3039	0.0359	8.6%	11.4%	10.5%	4.9%
5	Traffic Intensity = 0.6	2.2917	0.0404	2.2091	0.0169	2.1627	0.0175	2.0600	0.0154	1.9074	0.0172	14.0%	13.1%	11.2%	5.9%
6	Traffic Intensity = 0.8	2.7753	0.0207	2.2053	0.0232	2.1322	0.0275	2.0072	0.0175	1.8645	0.0171	27.6%	15.3%	11.7%	5.8%
7	Capacity = Equal	2.4528	0.0301	2.2852	0.0186	2.2521	0.0185	2.1225	0.0223	1.9683	0.0210	16.3%	14.1%	12.5%	6.1%
8	Capacity = Different	2.6222	0.0404	2.3903	0.0232	2.3238	0.0275	2.2198	0.0281	2.0822	0.0359	17.2%	12.4%	9.8%	5.0%
9	Product Mix = Equal	2.7204	0.0240	2.4767	0.0203	2.4290	0.0231	2.3693	0.0281	2.1509	0.0359	18.5%	13.1%	11.2%	7.5%
10	Product Mix = Different	2.3546	0.0404	2.1988	0.0232	2.1469	0.0275	1.9730	0.0223	1.8996	0.0215	15.0%	13.4%	11.0%	3.6%
11	Process Times = Equal	2.6498	0.0240	2.4389	0.0215	2.3799	0.0231	2.2233	0.0281	2.0469	0.0359	20.1%	16.6%	13.9%	6.7%
12	Process Times = Different	2.4252	0.0404	2.2366	0.0232	2.1960	0.0275	2.1191	0.0223	2.0036	0.0210	13.4%	10.0%	8.4%	4.4%
13	Overall Avg.	2.5375	0.0404	2.3378	0.0232	2.2879	0.0275	2.1712	0.0281	2.0252	0.0359	16.8%	13.3%	11.1%	5.5%

Δ1 = 100*(EDD-NACH-T)/EDD

Δ2 = 100*(BATC-I-NACH-T)/BATC-I

Δ3 = 100*(BATC-II-NACH-T)/BATC-II

Δ4 = 100*(MDBH-NACH-T)/MDBH

interval values obtained by NACH-T are similar to those obtained by benchmark control strategies.

Figure 4.4 presents the trend of performance improvements gained by NACH-T with an increasing number of product types. Although improvement percentages do not show an increasing trend, the actual improvement values clearly increase when the number of product types increases. This is due to the fact that NACH-T handles the incompatibility issue in the batch process decision making better than the benchmarks since NACH-T considers the effect of a decision alternative on all product types while selecting the best alternative decision. It should also be noted that the performance of EDD strategy worsens considerably when there are 10 product types.

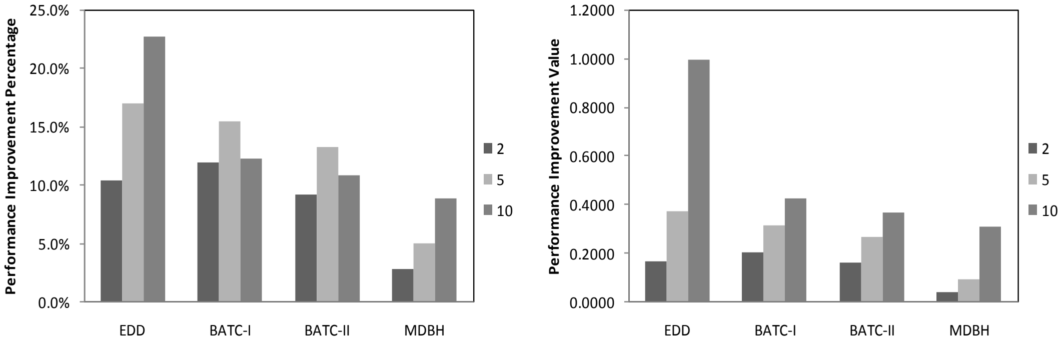

Fig. 4.4. Performance improvement of NACH-T with the number of product types

The performance improvements gained by NACH-T show a steady trend with increasing traffic intensity (see Figure 4.5). As an exception, the improvement over EDD strategy shows a significant increase with increasing traffic intensity. This result is expected since the EDD approach does not account for batch sizes when selecting the winner batch. Therefore, the fullness of a batch does not warrant priority. Instead, its main focus is the average due-date of the products in a batch.

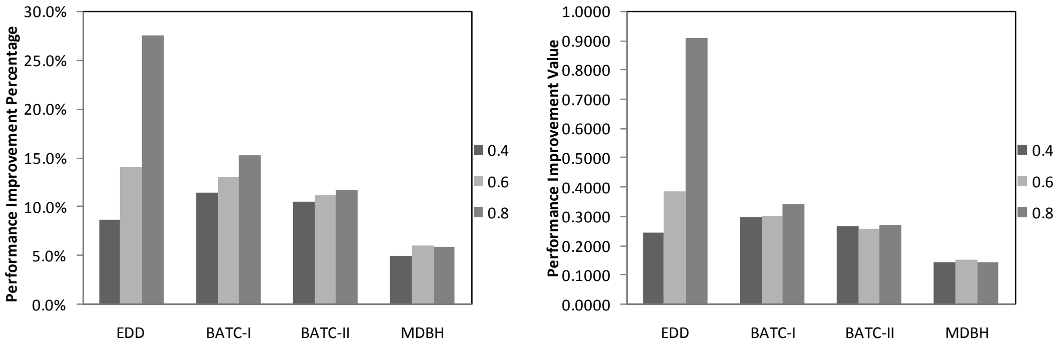

Fig. 4.5. Performance improvement of NACH-T with the batch processor traffic intensity

Percentage improvement gained by NACH-T does not show significant change between the cases where the batch processor capacities are equal and different for product types (see Figure 4.6). It is very common in semiconductor manufacturing that the batch processor capacity differs for different product types. Therefore it is crucial to have significant performance improvement in different capacity cases as well as in equal capacity cases.

The performance improvements gained by NACH-T also show a steady trend with equal and different product-mix settings (see Figure 4.7). However, the performance improvement over MDBH reduces significantly when the product-mix is unbalanced between product types. The main interpretation behind this result is similar to the case where the number of product types is small. In the different product-mix setting, the product-mix is dominated by a few product types. Consequently, control is focused on the dominating product types and the processor pretends as if there are fewer product types than the original. It should be noted that the results for the case where the number of product types is equal to 2 and the case where the product-mix is dominated by a few product types show a great deal of similarity when MDBH and NACH-T are compared.

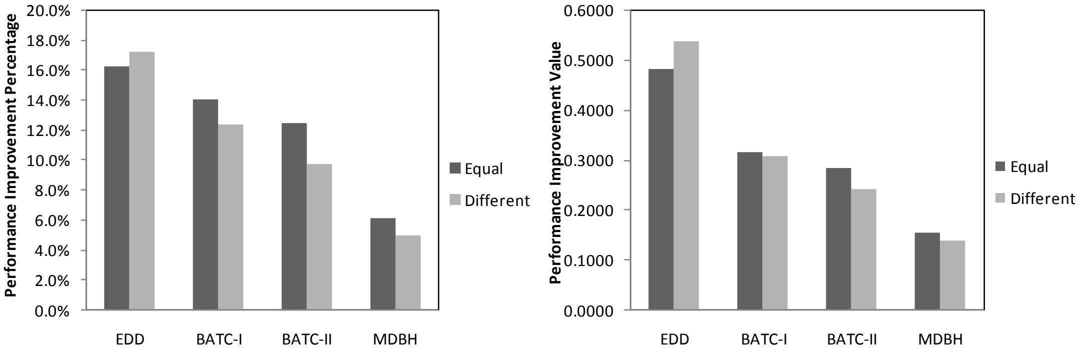

Fig. 4.6. Performance improvement of NACH-T with the batch processor capacity settings

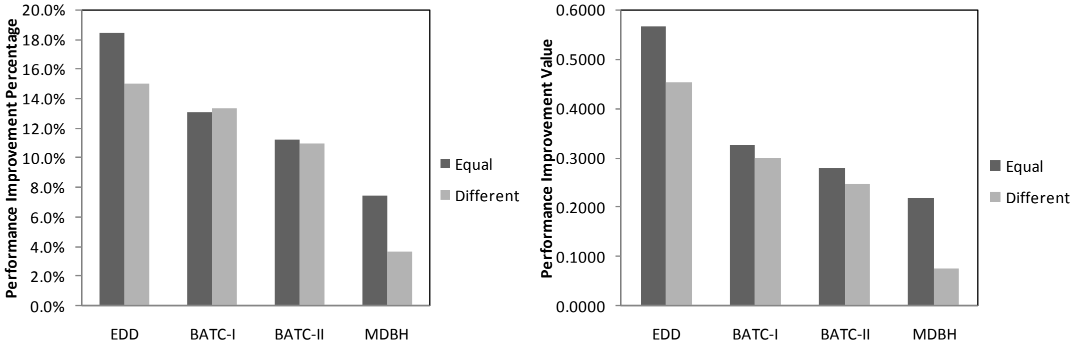

Fig. 4.7. Performance improvement of NACH-T with the product-mix settings

Similar results are observed when the batch process times are different for product types (see Figure 4.8). The performance improvement gained by NACH-T significantly decreases when the process times are unequal. This is due to the fact that product types with longer batch process times dominate the overall performance of the batch processor. In this case, waiting decisions are mainly driven by longer processing time product types.

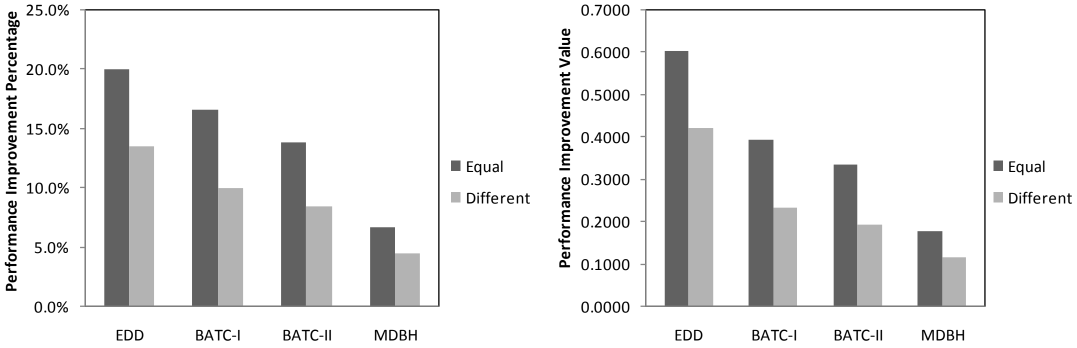

Fig. 4.8. Performance improvement of NACH-T with the batch process time settings

4.6.2 Comparison of NARCH-T with NACH-T and the benchmark strategies

The performance comparison of NARCH-T with the benchmark strategies and NACH-T is discussed in this section. Table 4.3 provides the performance improvements gained by NARCH-T when compared to the other strategies. It should be noted that the closest performing strategy is NACH-T since NARCH-T is based on NACH-T with the additional upstream re-sequencing approach.

Overall performance improvements are over 8% when compared to the benchmarks, excluding NACH-T. NARCH-T outperforms NACH-T by 3.5% overall. As expected, the performance improvements gained by NARCH-T and NACH-T show great similarity when compared to the benchmarks. It is more important to analyze the contribution of the re-sequencing used in NARCH-T. Therefore, the discussion is focused more on the comparison of NARCH-T with NACH-T. It should be also noted that the half-width of 95% confidence interval values obtained by NARCH-T are smaller

Table 4.3. Summary of the simulation results: NARCH-T is compared with the benchmark control strategies, the last five columns show the percentage improvements obtained by NARCH-T

		EDD		BATC-I		BATC-II		MDBH		NACH-T		NARCH-T						
No	Average at:	mean	hmax	mean	hmax	mean	hmax	mean	hmax	mean	hmax	mean	hmax	Δ1	Δ2	Δ3	Δ4	Δ5
1	Number of Products = 2	1.5606	0.0240	1.5966	0.0215	1.5536	0.0231	1.4317	0.0281	1.3956	0.0359	1.3683	0.0196	12.4%	13.9%	11.2%	4.9%	2.2%
2	Number of Products = 5	2.0943	0.0272	2.0344	0.0203	1.9842	0.0203	1.8130	0.0223	1.7203	0.0210	1.6316	0.0207	19.9%	19.0%	17.1%	9.5%	4.9%
3	Number of Products = 10	3.9577	0.0404	3.3823	0.0232	3.3260	0.0275	3.2687	0.0187	2.9598	0.0162	2.8599	0.0134	25.5%	15.3%	13.8%	11.9%	3.3%
4	Traffic Intensity = 0.4	2.5455	0.0301	2.5989	0.0215	2.5689	0.0231	2.4463	0.0281	2.3039	0.0359	2.1908	0.0207	12.8%	15.6%	14.7%	9.3%	4.6%
5	Traffic Intensity = 0.6	2.2917	0.0404	2.2091	0.0169	2.1627	0.0175	2.0600	0.0154	1.9074	0.0172	1.8438	0.0152	17.1%	16.1%	14.3%	9.2%	3.6%
6	Traffic Intensity = 0.8	2.7753	0.0207	2.2053	0.0232	2.1322	0.0275	2.0072	0.0175	1.8645	0.0171	1.8251	0.0174	27.9%	16.5%	13.1%	7.7%	2.2%
7	Capacity = Equal	2.4528	0.0301	2.2852	0.0186	2.2521	0.0185	2.1225	0.0223	1.9683	0.0210	1.8910	0.0207	19.6%	17.5%	15.9%	9.7%	3.9%
8	Capacity = Different	2.6222	0.0404	2.3903	0.0232	2.3238	0.0275	2.2198	0.0281	2.0822	0.0359	2.0155	0.0196	18.9%	14.7%	12.2%	7.8%	3.1%
9	Product Mix = Equal	2.7204	0.0240	2.4767	0.0203	2.4290	0.0231	2.3693	0.0281	2.1509	0.0359	2.0663	0.0196	21.6%	16.5%	14.6%	11.0%	3.8%
10	Product Mix = Different	2.3546	0.0404	2.1988	0.0232	2.1469	0.0275	1.9730	0.0223	1.8996	0.0215	1.8402	0.0207	16.9%	15.7%	13.5%	6.5%	3.1%
11	Process Times = Equal	2.6498	0.0240	2.4389	0.0215	2.3799	0.0231	2.2233	0.0281	2.0469	0.0359	1.9757	0.0196	21.9%	19.0%	16.4%	9.7%	3.4%
12	Process Times = Different	2.4252	0.0404	2.2366	0.0232	2.1960	0.0275	2.1191	0.0223	2.0036	0.0210	1.9308	0.0207	16.6%	13.2%	11.7%	7.8%	3.6%
13	Overall Avg.	2.5375	0.0404	2.3378	0.0232	2.2879	0.0275	2.1712	0.0281	2.0252	0.0359	1.9532	0.0207	19.3%	16.1%	14.0%	8.8%	3.5%

Δ1 = 100*(EDD-NARCH-T)/EDD

Δ2 = 100*(BATC-I-NARCH-T)/BATC-I

Δ3 = 100*(BATC-II-NARCH-T)/BATC-II

Δ4 = 100*(MDBH-NARCH-T)/MDBH

Δ5 = 100*(NACH-T-NARCH-T)/NACH-T

than those obtained by NACH-T which shows that re-sequencing does not increase the variance in the normalized tardiness values.

Figure 4.9 illustrates the increasing trend of performance improvements gained by NARCH-T with an increasing number of product types. The interpretation of this result is due to the fact that the actual inter-arrival time of a specific product type increases with the increase in the number of product types since the product-mix ratios reduce. However, the next arrival time of a specific product type can be reduced by re-sequencing the serial processor's queue. This may result in a better batching decision for the product type at its next arrival point.

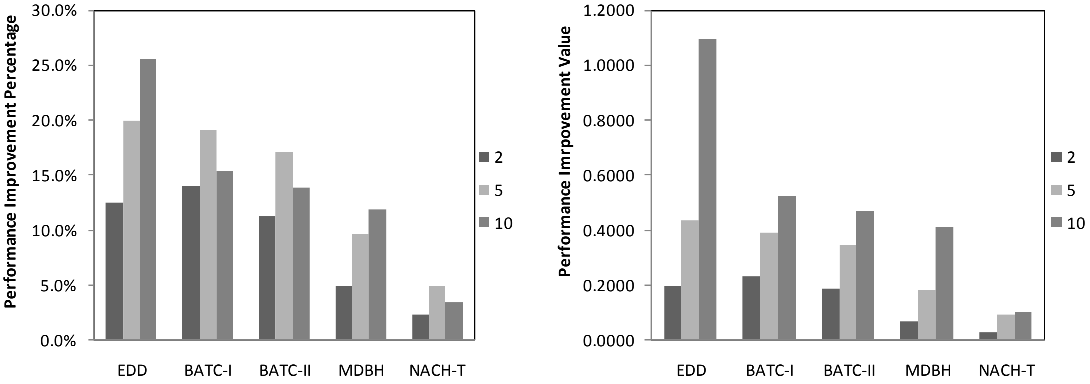

Fig. 4.9. Performance improvement of NARCH-T with the number of product types

On the other hand, when traffic intensity increases, the performance improvement of NARCH-T decreases (see Figure 4.10), except for the EDD strategy. This is due to the fact that the number of products available at the batch processor's buffers increases when the traffic is high. This makes it very rare to realize a benefit in re-sequencing the serial processor's queue and waiting for the desired product. Since re-sequencing only takes place when waiting decisions are considered, the improvement over NACH-T decreases clearly with high traffic intensity levels. The performance improvement shows an increase when NARCH-T is compared to EDD and this is again due to the fact that EDD does not account for fullness of a batch as a priority feature.

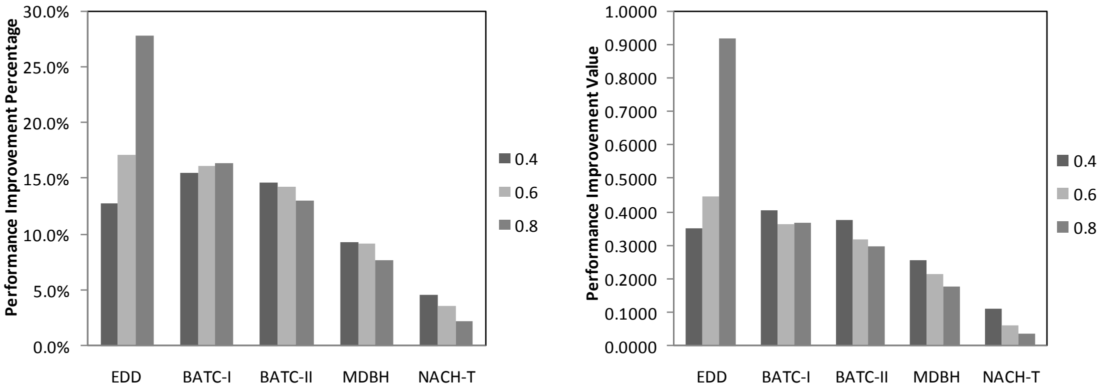

Fig. 4.10. Performance improvement of NARCH-T with the batch processor traffic intensity

Although batch processors have high overall utilization levels in practice, the actual traffic levels change over time during production which makes it possible to observe medium and low traffic. When traffic is high, control strategies aim to start the batch processor with one of the available product types. However, batch process control becomes more important in situations of low or medium traffic. It should be noted that if the control of batch processors are not effective during low or medium traffic states, then it is possible to get into a high traffic state very quickly. Very often, a lack of proper control strategy may lead the batch processors to bottleneck situations.

4.6.3 Complexity of the NACH-T and NARCH-T algorithms

Since the control strategies discussed in this chapter are proposed for on-line control of batch processors, their computational complexities have to be low enough to be implemented in real applications. In both NACH-T and NARCH-T, the number of decision alternatives that are evaluated to find the best decision depends on the number of product types, N.

The analysis in Step 1 for both NACH-T and NARCH-T considers at most B future arrivals (B denotes the maximum of the batch capacity values for product types) for the tardiness gain/loss determination of individual product types. For each arrival point a maximum of B operations are required to find the total tardiness gain/loss of waiting for the arrival point and B comparisons to find the suggested decision. NACH-T reaches a suggested decision for a particular product type at maximum B^2+B operations. Say M and K denote the upper bounds for the number of arrivals within a decision horizon and the number of products waiting in the batch processor's buffers. Since NARCH-T requires additional M arrival time updates when a re-sequencing is considered, the maximum number of operations is B^2+BM+1 for NARCH-T. Therefore, in order to find the suggested decisions in Step 1, at most N(B^2+B) and N(B^2+BM+1) operations are required for NACH-T and NARCH-T respectively.

In Step 2, there are a maximum of N decision alternatives (since at most one decision alternative is suggested per product type) to evaluate considering the upper bound M future arrivals. The tardiness metric value corresponding to decision alternatives require at most 2N+M+K operations to determine τ_{j0} and at most 3(M+K) operations to find the sum of the individual tardiness values of the M+K products. Therefore the evaluation of a decision alternative is bounded by 2N+4(M+K)+1 operations. Since a maximum of N decision alternatives is considered, total evaluation is bounded by N(2N+4(M+K)+1) operations. Additional N-1 comparisons are required to find the best decision alternative. Together with the maximum number of operations required for Step 1, the complexity of NACH-T and NARCH-T is in the form of $O(N^2)$ to reach a final decision at a particular decision point.

4.7 Contribution of the chapter

The research presented in this chapter contributes to the control of batch processors in the following manner:

i) A mathematical endeavor is provided to use available upstream information and possibly change the upstream sequence for batch process decision making with respect to mean tardiness performance.

ii) Two new control strategies, namely NACH-T and NARCH-T, are developed for the long-run control of batch processors to minimize mean tardiness of products. The look-ahead batching framework and the mean tardiness metric used in these strategies make them the first dynamic control approaches that combine near-future information with due-date related evaluation in batching decisions. The experimental study shows that mean tardiness values are reduced with NACH-T and NARCH-T significantly, when compared to the benchmark approaches.

iii) Overall performance improvements observed over the best performing benchmark strategy are about 5.5% and 8.8% with NACH-T and NARCH-T respectively. An increasing trend is observed in performance improvements when the number of product types increases. This result is very promising when the diverse product portfolio of the semiconductor industry is considered.

iv) Results obtained by NARCH-T indicate that upstream re-sequencing can be used effectively to improve the delivery performance of batch processors. As compared to NACH-T, NARCH-T reduces mean tardiness measure, especially when the traffic intensity is in the medium or low levels and the number of products is large.

v) The complexity of the algorithms is in the form of $O(N^2)$, with N being the number of product types. This provides an advantage for implementing these control strategies as real-time decision making tools for wafer fabrication facilities where the product-mix has high diversity.

CHAPTER V

BI-CRITERIA CONTROL OF BATCH PROCESSORS USING LOOK-AHEAD BATCHING

The purpose of this chapter is to demonstrate the use of look-ahead batching in on-line control of batch processors when there is more than one performance criteria. Specifically, simultaneous minimization of mean cycle time and mean tardiness performances is addressed as the bi-criteria version of the control problem. Look-ahead batch control approaches for these two criteria are combined with a weighted global criterion method, assuming the decision maker has appropriate choices of criteria weights.

5.1 Introduction

Real life production control problems require the decision maker to consider a number of criteria before arriving at any decision. A solution that shows very good performance with respect to one criterion might show very poor performance with respect to another criterion. The decision maker needs to evaluate the trade-off between the criteria in the presence of conflicting objectives. For example, in semiconductor production management, cycle time related criteria and due-date related criteria usually conflict when the arrival times and due-dates of the products are not agreeable.

In this chapter the objective of the dynamic control is to minimize mean cycle time and mean tardiness simultaneously on a batch processor. The significance of both criteria from the management perspective was extensively discussed in earlier chapters. From the management point of view, cycle time is a measure for production performance, since cycle time reduction leads to cost savings related to work-in-process inventory. On the other hand, due-date related objectives are considered performance measures for customer satisfaction. Tardiness is one of the commonly used measures to evaluate on-time delivery of products. Large tardiness levels lead to loss of good-will and subsequently loss of new orders. The motivation of this chapter is the necessity of incorporating the manufacturer’s concerns as well as the customer’s concern in the control of batch processors.

A bi-criteria look-ahead batch control strategy is proposed in this chapter. The main contribution is the successful adaptation of the look-ahead control approaches described in previous chapters for single criterion problems to the bi-criteria problem. The rest of the chapter is organized as follows. In Section 5.2, a brief background on multi-objective optimization is provided. The problem is defined briefly and the notations are presented in Section 5.3. The look-ahead batch control policy proposed for the bi-criteria problem is introduced in Section 5.4. In Section 5.5, well-known benchmark batch control policies are adapted for the bi-criteria problem. Discrete event simulation is used to compare the proposed approach with the benchmark policies, and simulation results are discussed in Section 5.6. The contributions of the chapter to the control of batch processors are summarized in Section 5.7.

5.2 Background on multi-objective optimization

The general approach to multi-objective optimization problems is to combine the multiple objectives into one scalar objective, whose solution is a Pareto optimal point for the original multi-objective problem. Most of these combinations are expressed through a linear function or distance derivatives. Among the methods often used are *Weighted Aggregation*, *Global Criterion*, *Minimum Fractional Deviation* and *Compromise Programming* (see Tabucanon (1988) and Gupta and Sivakumar (2002)).

Weighted Aggregation: In this method, the objective is to minimize a positively weighted convex sum of the objectives. The solution's quality is dependent on the decision maker's choice of appropriate weights. The form of the problem is given by (5.1).

$$Min\ F = \sum_{j=1}^{m} w_j f_j\ (x) \tag{5.1}$$

Here, w_j represents the non-negative weights assigned to the objectives f_j with $\sum_{j=1}^{m} w_j = 1$. Although weighted aggregation is one of the simplest ways to characterize the multi-objective problem by a single objective problem, the frequently used approach is to generate multiple points in the Pareto set by using different settings of the convex weights. However, this method is open to the domination of one objective if there are magnitude differences. In these situations, this method becomes misleading, always deciding in favor of a particular objective, unless normalization is performed.

Global Criterion: In this method, a single objective function is formed by summing the relative distances of individual objectives from their known minimal values. This way, from the original *m* objective functions, a single function is formulated and the problem becomes a single objective optimization problem. The form of the modified global problem is given by (5.2).

$$Min\ F = \sum_{j=1}^{m} \left[\frac{f_j(x^*) - f_j(x)}{f_j(x^*)} \right] \tag{5.2}$$

Here, $f_j(x^*)$ is the optimum value of a single objective function j at its optima point $x*$ and $f_j(x)$ is the function value itself. The major negative effects of magnitude differences are removed by using the ratios.

Minimum Fractional Deviation: In this method, a single objective function is formulated by the sum of the fractional deviation of all objectives. The fractional deviation of each objective is expressed as a fraction of its maximum deviation. The form of the problem is given by (5.3)

$$Min\ F = \sum_{j=1}^{m} \left[\frac{f_j(x^*) - f_j(x)}{f_j(x^*) - f_j(x^0)} \right] \tag{5.3}$$

Here, $f_j(x^0)$ is the least desirable value of $f_j(x)$. The normalization of the deviations eliminates the effect of the magnitude differences.

Compromise Programming: In this method, a single objective function is formulated by attaching a distance metric to the weighted sum of the fractional deviation of all objectives. The point of interest is the comparison of distances of different efficient points from the point of reference. The form of the problem is given by (5.4)

$$Min\ F = \left[\sum_{j=1}^{m}\left[w_j \frac{f_j(x^*) - f_j(x)}{f_j(x^*)}\right]^r\right]^{1/r} \tag{5.4}$$

When the value of r is 1, the compromise programming technique becomes a weighted global criterion. Here, a double weighting exists where the r value reflects the importance of the deviation from the single criterion optimal values and the w_j values reflect the relative importance of each criterion.

5.3 Problem definition and notation

There are N incompatible product types being processed by the batch processor. A serial processor is used as an upstream station to provide near-future information for the batch processor. The attributes of the serial and batch processors described in Chapters III and IV are the same for this chapter. The only purpose of using the serial processor is to provide near-future information for the batch processor. The re-sequencing method described in previous chapters is not considered in this chapter. The objective of the control is to simultaneously minimize mean waiting time in the batch processor's buffers and mean tardiness at the end of the batch process. Criteria weight choices of the decision maker and near-future information (including due-dates and arrival times of the products) are assumed to be available for the decision making process. The following notation is used in the algorithm described in the next section:

N = number of product types
T_j = batch processing time for product type j
B_j = batch processor capacity for product type j
P_j = ratio of product type j in the mix
d_{ij} = due-date of the product i of type j
t_0 = current decision point

t_{jn} = time that n^{th} upcoming product of type j will arrive at the batch processor's buffers
K_{j0} = the union of the products that are in the batch processor's buffers at t_0 and the products that will arrive within a time-window of length T_j
K_{jn} = the union of the products that are in the batch processor's buffers at t_0 and the products that will arrive within a time-window of length $t_{jn}+T_j-t_0$
τ_{j0} = average time spent in the batch processor station by the remaining batches in K_{j0} after loading product type j at t_0
τ_{jn} = average time spent in the batch processor station by the remaining batches in K_{jn} after loading product type j at t_{jn}
q_j = number of products of type j in the batch processor's buffers at t_0
q_{ij0} = number of products of type i in the set K_{j0}
q_{ijn} = number of products of type i in the set K_{jn}
MT_{j0} = mean tardiness metric value caused by starting batch β_{j0} at t_0
MT_{jn} = mean tardiness metric value caused by starting batch β_{jn} at t_{jn}
D_{j0} = mean waiting time metric value caused by starting batch β_{j0} at t
D_{jn} = mean waiting time metric value caused by starting batch β_{jn} at t_{jn}
S = set of alternative decisions at decision point t_0

5.4 Next arrival control heuristic for bi-criteria batch processing (NACH-II)

NACH-II is the extension of the look-ahead batch control approaches described in previous chapters for single criterion problem settings. In the bi-criteria extension, near-future upcoming product information is utilized to evaluate the decision alternatives with respect to the two criteria, mean waiting time and mean tardiness. The metrics described in the previous chapters for mean waiting time and mean tardiness are used in the evaluation process. Again, a rolling horizon approach is followed to postpone the final batching decisions to future decision points when it is preferable to wait for a future arrival. The weighted global criterion method is adapted to the dynamic control framework and applied to handle the bi-criteria problem, assuming the decision maker has distinct criteria weights. The weighted global criterion method accounts for the fractional deviation of each decision alternative's criteria values from the best possible criteria values in the set of decision alternatives. The weighted sum of these fraction values is counted as a single criterion for the decision alternative. The weighted global criterion is a special form of compromise programming when r value is 1 and takes care of the magnitude differences between criteria values.

The NACH-II algorithm is composed of two stages that are completed to reach a final decision at a decision point. In the first stage, the controller reviews the state of the batch processor with the near-future arrival information to find the decision alternatives. For each decision alternative, the values of the mean waiting time and mean tardiness metrics that are caused by executing the decision alternative are determined and reported to the second stage. In the second stage, the best decision is selected by applying the weighted global criterion method to the alternative decisions. The details of the first and second stages are as follows.

First Stage: At a particular decision point t_0, the algorithm evaluates product types to compose the set of alternative decisions. For a particular product type j, the controller starts by checking if the product type has a full batch available. If product type j has a full batch, the only decision alternative for product type j is to start the batch process at t_0. In order to use in the second stage analysis, the values for mean waiting time metric and mean tardiness metric of starting the batch process with product type j at t_0 are determined by using the near-future time window. The controller first ranks the products of type j in the batch processor's buffers by Earliest Due Date (EDD) priority index and forms the full batch β_{j0} using the first B_j products. Future information in a time window of length T_j is utilized to determine the outcome of starting the batch process with β_{j0}. Equation (5.4) is used to calculate the mean tardiness metric (MT_{j0}) of this start decision and equation (5.5) is used to calculate the mean waiting time metric (D_{j0}) (see Chapters III and IV for a detailed discussion of these equations).

$$MT_{j0} = \left[\sum_{ij \in \beta_{j0}} (t_0 + T_j - d_{ij})^+ + \sum_{mk \in (K_{j0} - \beta_{j0})} (t_0 + T_j + \tau_{j0} - d_{mk})^+ \right] \Big/ B_j \tag{5.4}$$

$$where \quad \tau_{j0} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kj0}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jj0} - B_j}{B_j} \right\rceil . T_j \right] \Big/ 2$$

$$D_{j0} = \left(\left(\sum_{k=1}^{N} q_k - B_j \right) \times T_j + \sum_{k=1}^{N} \sum\nolimits_{t_{km} < t_0 + T_j} \left(t_0 + T_j - t_{km} \right) \right) \Big/ B_j \tag{5.5}$$

On the other hand, if product type *j* has a partial batch, further analysis is required to find the decision alternatives. In this case, additional arrival points of product type *j* are considered by the algorithm. The controller includes these future arrival points in the set of alternative batch start times for product type *j*. The number of arrival points considered in the set is bounded by the number of required additional products to comprise a full batch. There are two categories of decision alternatives for product type *j*:

i) Start the batch process with product type *j* at t_0

ii) Wait for additional future arrivals that will be realized within the time window of length T_j.

The evaluation details of these alternative decisions are as follows.

i) Start the batch process with product type *j* at t_0:

A partial batch β_{j0} is constructed with the available q_j products of type j and loaded on the batch processor at t_0. The time window to be used in the evaluation of this decision alternative is the interval (t_0, t_0+T_j). Equations (5.6) and (5.7) are used to determine the values of mean tardiness and mean waiting time metrics respectively. These equations are similar to equations (5.4) and (5.5), except the fact that the throughput of the decision is q_j this time.

$$MT_{j0} = \left[\sum_{ij \in \beta_{j0}} \left(t_0 + T_j - d_{ij}\right)^+ + \sum_{mk \in \left(K_{j0} - \beta_{j0}\right)} \left(t_0 + T_j + \tau_{j0} - d_{mk}\right)^+ \right] \Big/ q_j \tag{5.6}$$

$$\text{where } \tau_{j0} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kj0}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jj0} - q_j}{B_j} \right\rceil . T_j \right] \Big/ 2$$

$$D_{j0} = \left(\left(\sum_{k=1}^{N} q_k - q_j \right) \times T_j + \sum_{k=1}^{N} \sum\nolimits_{t_{km} < t_0 + T_j} \left(t_0 + T_j - t_{km}\right) \right) \Big/ q_j \tag{5.7}$$

ii) Wait for a future arrival of product type j to start the batch process:

Assume that product type j has R_j future arrivals that are expected to occur in the time interval (t_0, t_0+T_j), then $\min\left(B_j - q_j, R_j\right)$ arrivals define the alternative points to start the batch process with product type j since further arrivals is not considered if a full batch condition is satisfied. These arrival points can be referred to as "feasible arrival points" because each of them provides an alternative decision for the current decision point t_0. At each feasible arrival point, an alternative batch can be formed including the arriving product. Consider a particular future arrival point for product type j, n^{th} arrival that is expected to occur at t_{jn}, where $t_{jn} \leq t_0 + T_j$ and $n \leq \min(B_j - q_j, R_j)$. With these conditions, the arrival time t_{jn} is considered a feasible arrival point for product type j to start the batch process. Future arrival information in the time interval (t_0, $t_{jn}+T_j$) is used to determine the values of the two metrics that are caused by loading this batch alternative. Equations (5.8) and (5.9) are used for mean tardiness and mean waiting time metrics respectively. Similar calculations are performed for each feasible arrival point of product type j (see Chapters III and IV for a detailed discussion of these equations).

$$
\begin{aligned}
MT_{jn} = \Bigg(& \sum_{ij \in \beta_{jn}} \left(t_{jn} + T_j - d_{ij}\right)^{+} \\
& + \sum_{mk \in \left(K_{jn} - \beta_{jn}\right)} \left(t_{jn} + T_j + \tau_{jn} - d_{mk}\right)^{+} \Bigg) \Big/ (q_j + n)
\end{aligned}
\tag{5.8}
$$

$$
\text{where} \quad \tau_{jn} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kjn}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jjn} - (q_j + n)}{B_j} \right\rceil . T_j \right] \Big/ 2
$$

$$
\begin{aligned}
D_{jn} = \Bigg(& \sum_{k=1, k \neq j}^{N} \left(q_k \times \left(t_{jn} + T_j - t_0\right)\right) + \sum_{t_{km} < t_{jn} + T_j \,\&\, k \neq j} \left(t_{jn} + T_j - t_{km}\right) \\
& + q_j \times \left(t_{jn} - t_0\right) + \sum_{p<n} \left(t_{jn} - t_{jp}\right) + \sum_{n<p \,\&\, t_{jp} < t_{jn} + T_j} \left(t_{jn} + T_j - t_{jp}\right) \Bigg) \Big/ (q_j + n)
\end{aligned}
\tag{5.9}
$$

The set of alternative decisions, S, is completed by repeating the same analysis for each product type that is available in the batch processor's buffers at t_0. Set S and the metric values of the alternatives in the set are moved to the second stage for the bi-criteria analysis.

Second Stage: Set S includes the decision alternatives and their corresponding single criterion metric values. In this stage, the weighted global criterion method is applied to find the best decision alternative according to the criteria weights, w_1 and w_2, selected by the decision maker. The values of mean waiting time and mean tardiness metrics for alternative i in the set S are represented by f_{1i} and f_{2i} respectively. The best (minimum) metric values in S are denoted by f_1^* and f_2^* for mean waiting time and mean tardiness respectively. Using this notation, for a particular decision alternative i in set S, the fractional single criteria values γ_{1i} and γ_{2i} are determined by equation (5.10).

$$\gamma_{1i} = \frac{f_{1i} - f_1^*}{f_1^*} \quad \text{and} \quad \gamma_{2i} = \frac{f_{2i} - f_2^*}{f_2^*} \tag{5.10}$$

Fractional criteria values are combined in one scalar value using the criteria weights of the decision maker. The combined single criterion value for decision alternative i is denoted by c_i where $c_i = w_1.\gamma_{1i} + w_2.\gamma_{2i}$. Then the decision alternative $i^* = \text{argmin}(c_i | i \in S)$ is selected as the winner decision. If i^* is a start decision, the batch process starts with the product type corresponding to i^*. On the other hand if i^* suggests waiting for a future arrival to start the batch process, then the batch processor is kept in waiting mode (i.e. idle) until the next arrival point. In this case, the algorithm is repeated at the next arrival point. The flowchart of the NACH-II algorithm is provided in Figure 5.1.

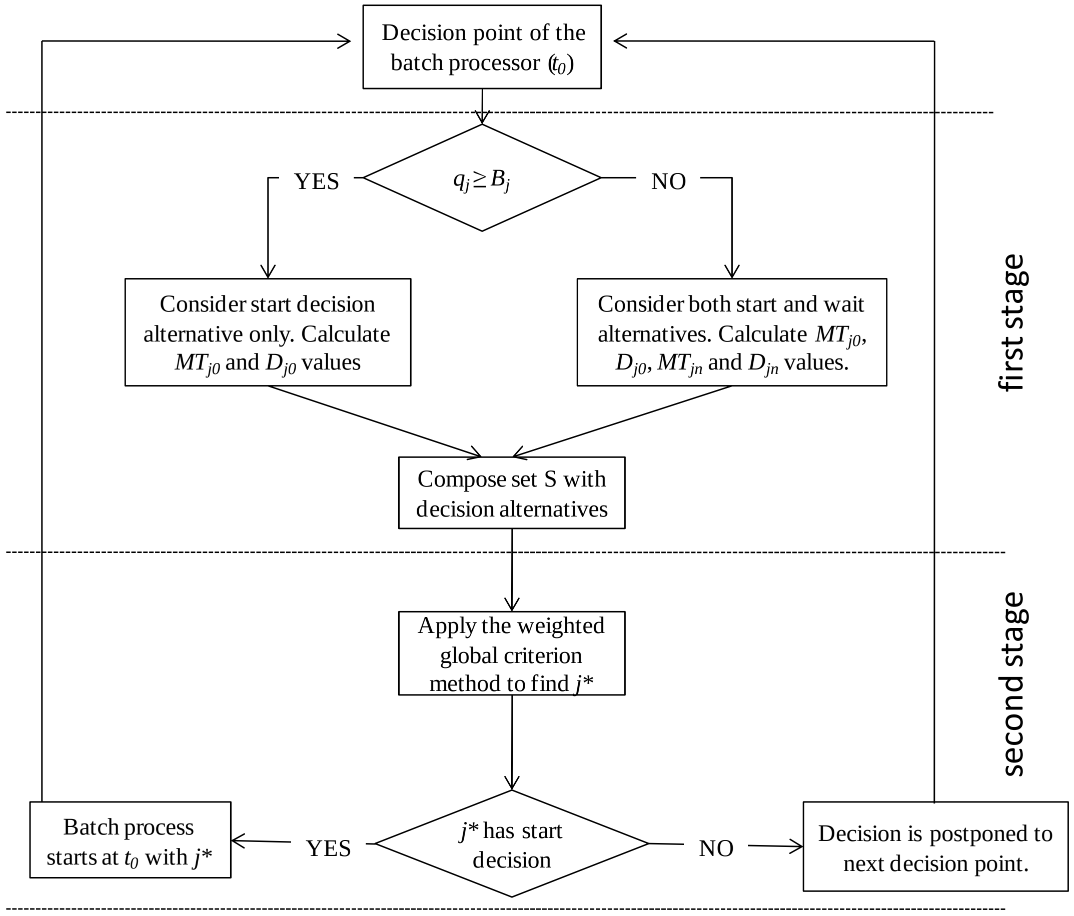

Fig. 5.1. Flowchart of the NACH-II algorithm

5.5 Modification of the benchmark control strategies for the bi-criteria problem

Although multi-criteria batch process control is becoming a popular research direction, a limited amount of research actually aims to minimize due-date and cycle time related performance criteria simultaneously. Therefore, well-known control strategies are adapted to benchmark the NACH-II strategy. Three control strategies are selected due to their prevalent use in the semiconductor industry: *full batch policy*, *minimum batch size (MBS) policy* and *no-idling policy*. Without using any future arrival information, these control strategies rely only on the batch processor's buffer information to make start and wait decisions. However, in this research, a bi-criteria modification with near-future arrival information is added to these strategies to make them suitable for bi-criteria decision making.

i) Full Batch Policy: At a particular decision point t_0, if there is at least one available full batch, the batch process starts with one of the full batch product types. The selection of the winner product type out of the full batch product types follows exactly the same steps as the full batch case of the NACH-II algorithm. For a particular full batch product type j, if $q_j > B_j$ then first B_j products of type j are selected using EDD index and batch β_{j0} is formed with these products. Then, a look-ahead window (t_0, t_0+T_j) is used to estimate the outcome of starting the batch process with β_{j0} at t_0. The values of mean tardiness and mean waiting time metrics for this decision alternative are determined using equations (5.4) and (5.5) respectively. After completing the MT_{j0} and D_{j0} calculations for the full batch product types, set S is formed and treated with the second stage analysis described in the NACH-II algorithm. In the case of partial batches, the batch processor is kept idle until the next decision point.

ii) Minimum Batch Size (MBS) Policy: According to this policy, the batch process starts if there is at least one product type satisfying the minimum batch size (MBS) requirements with its available products in the batch processor's buffers. At a particular decision point t_0, the product types with more products ready than their MBS requirements are considered as batch alternatives. For a particular product type j

satisfying this condition, batch β_{j0} is formed with min(q_j,B_j) products following an EDD priority. Again, a look-ahead window (t_0, t_0+T_j) is used to determine the metric values caused by starting the batch process with β_{j0} at t_0. Equations (5.11) and (5.12) are used to find the values of mean tardiness and mean waiting time metrics. This way, set S is formed with the products satisfying the MBS rule, and the second stage analysis described in the NACH-II algorithm is performed to find the winner product type. If none of the product types satisfies the MBS requirements, then the batch processor is kept idle until the next decision point.

$$\begin{aligned} MT_{j0} = \Bigg(& \sum_{ij \in \beta_{j0}} \left(t_0 + T_j - d_{ij}\right)^+ \\ & + \sum_{mk \in \left(K_{j0} - \beta_{j0}\right)} \left(t_0 + T_j + \tau_{j0} - d_{mk}\right)^+ \Bigg) / \min(q_j, B_j) \end{aligned} \tag{5.11}$$

$$\text{where} \quad \tau_{j0} = \left[\sum_{k=1, k \neq j}^{N} \left(\left\lceil \frac{q_{kj0}}{B_k} \right\rceil . T_k \right) + \left\lceil \frac{q_{jj0} - \min(q_j, B_j)}{B_j} \right\rceil . T_j \right] \Big/ 2$$

$$\begin{aligned} D_{j0} = \Bigg(& \left(\sum_{k=1}^{N} q_k - \min(q_j, B_j) \right) \times T_j \\ & + \sum_{k=1}^{N} \sum\nolimits_{t_{km} < t_0 + T_j} \left(t_0 + T_j - t_{km}\right) \Bigg) / \min(q_j, B_j) \end{aligned} \tag{5.12}$$

iii) No-idling Policy: No-idling policy aims to keep the batch processor running as long as there are available products in the batch processor's buffers. At a particular decision epoch t_0, if there is at least one available full batch, the no-idling policy follows the steps described in the full batch policy. On the other hand, if there are only partial batches, a special MBS rule with MBS levels equal to one is followed. The equations to calculate MT_{j0} and D_{j0} are provided in the descriptions of full batch and MBS policies. After the determination of the alternative decisions and the corresponding metric values, the bi-criteria approach proposed in the second stage of NACH-II is applied to find the winning product type.

5.6 Simulation study

The NACH-II algorithm is extensively tested by conducting a series of simulation experiments. The simulation model of the serial-batch processor system has the same attributes described in Chapters III and IV. Once products arrive at the serial station, their due-dates for the batch process and their serial processing times are assigned and this information is available to the controller attached to this system. Each simulation scenario is a particular combination of the settings for the product, process and processor characteristics. The characteristics and their different settings are listed in Table 5.1. For a particular scenario, the arrival rate for the batch processing station, λ, is found by equation (3.7), as explained in Chapter III. The arrival rate for the serial processor is same as the arrival rate for the batch processor, λ. The service rate of the serial processor, μ, is selected as $\mu=\lambda/0.8$ in this study.

A combination of the settings gives a total of 48 ($3x2^4$) different scenarios to investigate the performance of the control strategies. Since the proposed bi-criteria approach relies on the assumption that the decision maker has distinct criteria weights, the following weight vectors in the form of (w_1,w_2) are considered: tardiness dominated (0.1,0.9), equally important (0.5,0.5) and cycle time dominated (0.9,0.1). Here, w_1 is the weight for the mean waiting time criterion and w_2 is the weight for the mean tardiness criterion. Each control approach is tested on a problem instance that is composed of a particular simulation scenario and a particular vector of criteria weights. Each simulation experiment is replicated 10 times on a Pentium Dual Core 1.73 GHz. processor with 2GB RAM. Each replication has a duration of 100,000 time units and a warm-up of 5,000 time units.

Table 5.1. Configuration of the simulation study for Chapter V

No	Factor	Levels			
1	Control Strategy	Full Batch			
		MBS			
		No-idle			
		NACH-II			
2	Number of Products	2			
		5			
		10			
3	Interarrival Distribution	Exponential			
4	Ratio of Urgent Products	0.2			
			2 Products	5 Products	10 Products
5	Product Mix	Equal	(0.5, 0.5)	(0.2, 0.2, 0.2, 0.2, 0.2)	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different	(0.7, 0.3)	(0.35, 0.35, 0.1, 0.1, 0.1)	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
6	Capacity By Product	Equal	(5, 5)	(5, 5, 5, 5, 5)	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different	(7,2)	(7, 6, 5, 4, 3)	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
7	Batch Processing Time By Product	Equal	(25, 25)	(25, 25, 25, 25, 25)	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different	(40, 10)	(40, 30, 25, 20, 10)	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
8	Traffic Intensity	0.5			
		0.8			

Tables 5.2, 5.3 and 5.4 provide an overview of the simulation results for the cases where criteria weights are tardiness dominated (0.1,0.9), equally important (0.5,0.5) and waiting time dominated (0.9,0.1) respectively. In the tables, the simulation results are averaged over different settings of the product, processor and process characteristics specified in the second columns. The next 12 columns report the mean normalized waiting time in the batch processor's buffers, the mean normalized tardiness

from the batch process due-dates and the half-width of the 95% confidence interval for the weighted sum of these two performance measures.

The waiting time of a product is the time it enters the batch processor's buffers until it is loaded to the batch processor. The tardiness of a product is the positive value between its batch process completion time and its batch process due-date. The values are normalized by the batch process times as described in Chapters III and IV. The half-width confidence interval value is the maximum value observed in the scenarios defined by the second column. The values for MBS policy are obtained by simulating all MBS levels and selecting the best performing level. The last three columns summarize the performance improvement percentages obtained by NACH-II when compared to full batch, minimum batch size and no-idling policies respectively. The percentages are obtained by the weighted combination of the individual criterion improvement percentages. A positive percentage value indicates that NACH-II performs better. In order to test the statistical validity of the differences, a paired-*t* approach is used with a 95% confidence interval (see appendix-C for detailed analysis).

The criteria weights in the experiments determine the relative importance of the two criteria in batch process decision making. For all strategies, the trend of the overall results from Tables 5.2 through 5.4 indicates that the second stage of the NACH-II algorithm successfully tunes the objective of the problem from tardiness dominated to waiting time dominated cases except for the full batch policy. In other words, the mean tardiness values tend to increase when the relative criteria weights increase in favor of mean waiting time, and an opposite trend is observed in the mean waiting time values.

Table 5.2. Summary of the simulation results for w_1=0.1 w_2=0.9: NACH-II is compared with the benchmark control strategies, the last three columns show the percentage improvements obtained by NACH-II

No	Average at:	Full Batch			MBS			No-idling			NACH-II					
		CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	Δ1	Δ2	Δ3
1	Number of Products = 2	1.6577	1.7403	0.0230	1.1971	1.3762	0.0202	1.4529	1.5885	0.0208	1.1004	1.2825	0.0194	27.2%	7.1%	17.2%
2	Number of Products = 5	3.6393	3.0687	0.0244	2.1382	1.7745	0.0194	2.2607	1.9060	0.0194	2.0164	1.6861	0.0199	41.8%	5.0%	8.5%
3	Number of Products = 10	6.8990	6.1288	0.0220	3.7214	3.0737	0.0214	4.0909	3.4444	0.0832	3.5446	2.9435	0.0206	47.1%	4.2%	9.2%
4	Traffic Intensity = 0.5	4.6929	4.6070	0.0244	1.9941	2.1590	0.0214	2.0109	2.1872	0.0214	1.8906	2.0718	0.0206	53.3%	5.4%	6.5%
5	Traffic Intensity = 0.8	3.4378	2.6848	0.0177	2.7104	1.9905	0.0183	3.1921	2.4387	0.0832	2.5503	1.8696	0.0177	24.2%	5.4%	16.8%
6	Capacity = Equal	4.1136	3.6850	0.0223	2.2011	1.9519	0.0214	2.2134	1.9647	0.0214	2.1394	1.8929	0.0206	41.9%	3.3%	3.9%
7	Capacity = Different	4.0171	3.6069	0.0244	2.5034	2.1977	0.0202	2.9896	2.6613	0.0832	2.3015	2.0486	0.0194	35.5%	7.6%	19.3%
8	Product Mix = Equal	4.2402	3.8202	0.0244	2.4950	2.2140	0.0214	2.7422	2.4689	0.0832	2.3373	2.0955	0.0206	38.7%	6.0%	11.9%
9	Product Mix = Different	3.8905	3.4716	0.0168	2.2095	1.9355	0.0152	2.4608	2.1571	0.0152	2.1036	1.8459	0.0154	38.8%	4.9%	11.4%
10	Process Times = Equal	4.1951	3.7409	0.0230	2.4535	2.1520	0.0198	2.8529	2.5412	0.0832	2.2900	2.0352	0.0194	38.7%	5.7%	13.7%
11	Process Times = Different	3.9356	3.5510	0.0244	2.2510	1.9975	0.0214	2.3501	2.0848	0.0214	2.1509	1.9062	0.0206	38.7%	5.2%	9.5%
12	Overall Avg.	4.0653	3.6459	0.0244	2.3522	2.0748	0.0214	2.6015	2.3130	0.0832	2.2204	1.9707	0.0206	38.7%	5.4%	11.6%

Δ1 = Weighted percentage improvement of NACH-II over Full Batch

Δ2 = Weighted percentage improvement of NACH-II over MBS

Δ3 = Weighted percentage improvement of NACH-II over No-idling

Table 5.3. Summary of the simulation results for w_1=0.5 w_2=0.5: NACH-II is compared with the benchmark control strategies, the last three columns show the percentage improvements obtained by NACH-II

No	Average at:	Full Batch			MBS			No-idling			NACH-II					
		CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	Δ1	Δ2	Δ3
1	Number of Products = 2	1.6577	1.7403	0.0204	1.1740	1.3984	0.0200	1.4378	1.6048	0.0204	1.0387	1.3155	0.0192	28.3%	8.7%	18.8%
2	Number of Products = 5	3.6365	3.0556	0.0207	2.1043	1.7944	0.0206	2.2298	1.9274	0.0196	1.9853	1.7018	0.0173	41.7%	5.1%	8.6%
3	Number of Products = 10	6.8934	6.1295	0.0178	3.6545	3.0961	0.0185	4.0199	3.4668	0.0711	3.4935	2.9737	0.0187	47.2%	3.9%	8.9%
4	Traffic Intensity = 0.5	4.6926	4.6075	0.0207	1.9672	2.1797	0.0200	1.9904	2.2013	0.0204	1.8402	2.0979	0.0187	53.8%	6.4%	7.5%
5	Traffic Intensity = 0.8	3.4325	2.6761	0.0157	2.6547	2.0129	0.0206	3.1346	2.4647	0.0711	2.5047	1.8961	0.0192	24.2%	5.4%	16.7%
6	Capacity = Equal	4.1135	3.6850	0.0180	2.1786	1.9661	0.0185	2.1922	1.9772	0.0185	2.1008	1.9160	0.0187	42.3%	3.8%	4.5%
7	Capacity = Different	4.0115	3.5986	0.0207	2.4433	2.2265	0.0206	2.9327	2.6889	0.0711	2.2441	2.0780	0.0192	35.8%	8.0%	19.8%
8	Product Mix = Equal	4.2376	3.8205	0.0207	2.4514	2.2345	0.0206	2.7063	2.4857	0.0711	2.3035	2.1186	0.0192	38.9%	6.1%	12.1%
9	Product Mix = Different	3.8874	3.4631	0.0150	2.1705	1.9581	0.0143	2.4186	2.1803	0.0143	2.0414	1.8754	0.0129	39.1%	5.7%	12.2%
10	Process Times = Equal	4.1950	3.7409	0.0204	2.4443	2.1593	0.0176	2.8461	2.5502	0.0711	2.2715	2.0474	0.0192	38.9%	5.9%	14.0%
11	Process Times = Different	3.9300	3.5427	0.0207	2.1776	2.0333	0.0206	2.2789	2.1158	0.0204	2.0735	1.9466	0.0187	39.2%	5.9%	10.3%
12	Overall Avg.	4.0625	3.6418	0.0207	2.3109	2.0963	0.0206	2.5625	2.3330	0.0711	2.1725	1.9970	0.0192	39.0%	5.9%	12.1%

Δ1 = Weighted percentage improvement of NACH-II over Full Batch

Δ2 = Weighted percentage improvement of NACH-II over MBS

Δ3 = Weighted percentage improvement of NACH-II over No-idling

Table 5.4. Summary of the simulation results for w_1=0.9 w_2=0.1: NACH-II is compared with the benchmark control strategies, the last three columns show the percentage improvements obtained by NACH-II

No	Average at:	Full Batch			MBS			No-idling			NACH-II					
		CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	CT	Tard.	hmax	Δ1	Δ2	Δ3
1	Number of Products = 2	1.6577	1.7403	0.0201	1.1546	1.4100	0.0228	1.3844	1.6209	0.0198	1.0120	1.3282	0.0289	35.4%	12.4%	21.6%
2	Number of Products = 5	3.6245	3.0765	0.0123	2.0600	1.8129	0.0247	2.1884	1.9407	0.0106	1.9532	1.7151	0.0153	43.4%	4.7%	8.0%
3	Number of Products = 10	6.8764	6.1291	0.0251	3.5887	3.1293	0.0182	3.9565	3.5121	0.0376	3.4180	3.0017	0.0208	47.5%	4.4%	9.1%
4	Traffic Intensity = 0.5	4.6893	4.6093	0.0251	1.9330	2.1961	0.0228	1.9451	2.2157	0.0183	1.8100	2.1081	0.0094	58.8%	8.4%	9.1%
5	Traffic Intensity = 0.8	3.4164	2.6880	0.0201	2.6026	2.0387	0.0247	3.0743	2.5002	0.0376	2.4455	1.9219	0.0289	25.3%	5.9%	16.7%
6	Capacity = Equal	4.1122	3.6862	0.0251	2.1525	1.9857	0.0182	2.1669	1.9898	0.0183	2.0692	1.9271	0.0208	44.8%	4.9%	5.7%
7	Capacity = Different	3.9935	3.6111	0.0226	2.3831	2.2491	0.0247	2.8525	2.7261	0.0376	2.1863	2.1030	0.0289	39.4%	9.4%	20.1%
8	Product Mix = Equal	4.2253	3.8283	0.0226	2.4144	2.2552	0.0247	2.6546	2.5064	0.0376	2.2603	2.1310	0.0289	41.7%	7.4%	12.5%
9	Product Mix = Different	3.8804	3.4690	0.0251	2.1212	1.9796	0.0182	2.3649	2.2094	0.0183	1.9952	1.8991	0.0145	42.5%	7.0%	13.3%
10	Process Times = Equal	4.1950	3.7409	0.0251	2.4268	2.1681	0.0201	2.8259	2.5697	0.0376	2.2563	2.0573	0.0289	41.0%	7.1%	14.8%
11	Process Times = Different	3.9107	3.5564	0.0209	2.1087	2.0667	0.0247	2.1936	2.1462	0.0129	1.9992	1.9727	0.0208	43.1%	7.3%	11.0%
12	Overall Avg.	4.0529	3.6487	0.0251	2.2678	2.1174	0.0247	2.5097	2.3579	0.0376	2.1277	2.0150	0.0289	42.1%	7.2%	12.9%

Δ1 = Weighted percentage improvement of NACH-II over Full Batch

Δ2 = Weighted percentage improvement of NACH-II over MBS

Δ3 = Weighted percentage improvement of NACH-II over No-idling

Usually, there is only one decision alternative at the decision points when the full batch policy is used. Therefore the performance values are not affected by the criteria weights since it is not possible to choose between multiple alternatives. In all three settings of the criteria weights, significant improvements are observed with NACH-II as compared to the benchmark policies. It appears that there is a benefit in considering future arrival points as alternative batch start points when near-future arrival information is available. On the average, the weighted sum of the performance measures improves by 5.4-7.2% when NACH-II is compared to the best MBS levels. The improvement percentages are 11.6-12.9% and 38.7-42.1% when NACH-II is compared to no-idling and full batch policies respectively. The results also indicate that the half-width of the 95% confidence interval values obtained by NACH-II are similar to those obtained by the benchmark control strategies and many times smaller. This shows that NACH-II does not have a negative effect on the variance of the performance measures.

When the number of products increases, no matter which control strategy is used, the outcomes of both criteria tend to be larger due to the incompatibility of the products. Full batch policy is very sensitive to the traffic intensity level of the batch processor. The products wait extremely long to fulfill the full batch requirements in the case of lower traffic levels. On the other hand, the performance of the no-idling policy decreases with increasing traffic intensity since the inter-arrival times decrease, and waiting for another product becomes very beneficial. It should be noted that the percentage improvement gained by NACH-II is affected significantly with change in traffic intensity levels, due to the benchmark policies' performance instability.

The performance of MBS policy is not significantly affected with changing traffic intensity levels because the values reported in the tables belong to the best performing MBS levels in each problem setting. Best performing MBS levels tend to be high when the traffic intensity level is high and tend to be low when the traffic intensity level is low. However, finding the best performing MBS levels requires very long simulation times. Minimum batch size policy is practically inefficient in a system that has a significant variation in the traffic intensity of the batch processor because tuning the MBS levels to the dynamic changes of the system is a very difficult task.

The batch processor capacity setting has an impact on the performance of the control strategies. When the capacity requirement is equal for each product type, control strategies perform better as compared to the case where capacities are different except for the full batch policy. This result is mainly due to the fact that different maximum batch sizes lead to full batch situations more often than equal maximum batch sizes. Therefore the waiting time for full batch situations is not affected negatively in the case of different capacity levels. The performance improvement gained by NACH-II as compared to MBS and no-idling policies increases when the capacity by product is different. This result is very important, due to its relevance for real-life situations in which the capacity of the batch processor is typically different for different product types.

The product-mix settings also affect the performance of the control strategies. When the product-mix is dominated by some product types, the attention is given to fewer product types than the original number of product types, since they have more influence on the performance measures. The values for both criteria tend to decrease with different product-mix settings. The intuition of this result is related to the effect of having fewer product types on the performance measures. Results indicate that the performance improvement gained by NACH-II is not affected when the product-mix is set to different values for product types. The process time settings also have a similar impact on the performance of the control strategies as the product-mix levels do. This result also originates from the same explanation above for different product-mix settings.

The complexity of the NACH-II algorithm is driven by the evaluation of the decision alternatives with mean tardiness metric. It was shown in Chapter IV that reaching a final decision has a complexity in the form of $O(N^2)$ when the proposed mean tardiness metric is used. On the other hand, evaluating all decision alternatives with mean waiting time metric has a complexity in the form of O(N) as addressed in Chapter III. Since these evaluations are decomposed and performed separately, the complexity of the NACH-II algorithm is also in the form of $O(N^2)$. It should also be noted that the complexity of searching for the best performing MBS level is in the form of $O(B^N)$ where B is the upper bound for the batch processor capacity.

5.7 Contribution of the chapter

The research presented in this chapter contributes to the control of batch processors in the following manner:

i) This chapter demonstrates a method for on-line control of batch processors when there are multiple performance measures of interest. Look-ahead batch control approaches developed for mean tardiness and mean cycle time measures are extended to the case where the control task is to minimize both measures simultaneously. This extension can be generalized to *m* different performance measures as long as their representative metric values can be determined using look-ahead windows.

ii) A control strategy, namely NACH-II, is developed for the extension, and significant performance improvements are observed when compared to the benchmark strategies. The simulation results also indicate that the performance improvements are very steady in the group of scenarios that reflect real-life situations (e.g. unequal product-mix, different capacity by product).

iii) The complexity of NACH-II is in the form of $O(N^2)$ where N is the number of product types. Therefore NACH-II can be efficiently implemented in wafer fabrication facilities for on-line control of batch processors.

iv) The look-ahead evaluation mechanism and the second stage of the NACH-II algorithm are also applied to control policies frequently used in wafer fabrication facilities. The quality of the batch process decision making can be improved in real-life wafer fabrication by combining these two components of the NACH-II algorithm with other control strategies.

CHAPTER VI

SUMMARY, CONTRIBUTIONS AND FUTURE RESEARCH DIRECTIONS

6.1 Contributions of the dissertation

This research indicates that there is potential benefit in utilizing queue information and further incorporating the sequencing decisions of the upstream station in the control of the batch processor. A mathematical endeavor has been developed to use upstream information and the re-sequencing approach within new control strategies. These proposed control strategies have been experimentally shown to improve mean cycle time and mean tardiness of the serial-batch processor system on which the control strategies are developed.

In Chapter III, a new control strategy, namely NARCH, that combines look-ahead batching with a re-sequencing approach on the upstream serial station was proposed and compared with the benchmark control strategies. The proposed control strategy shows promising results by improving the mean waiting time performance of the products as compared to the benchmark control strategies. Performance improvement increases when the number of product types is large and when the traffic intensity of the batch processor is moderate or low. This result is important in the sense that there is a high diversity of product types in a typical wafer fabrication process and the traffic intensity of the batch processor stations is highly variable.

In Chapter IV, two heuristic control strategies, namely NACH-T and NARCH-T, were proposed for controlling the batch processor with mean tardiness performance criteria. NACH-T effectively utilizes near-future information coming from the upstream serial processor to reduce the mean tardiness of the products being processed by the serial-batch processor system. Experimental results indicate that NACH-T improves the mean tardiness performance measure as compared to benchmark control strategies. The improvement increases with an increasing number of product types. In addition, NARCH-T improves the mean tardiness performance further by applying a re-sequencing approach at the decision points. The comparison of NARCH-T with NACH-T shows that the improvement gained by NARCH-T increases especially when the number of product types is large and when the traffic intensity of the batch processor is moderate or low.

Chapter V demonstrated how look-ahead control strategies developed for mean tardiness and mean cycle time performance measures can be effectively extended to the case where both performance measures are present in the objective of the control task. The proposed control strategy, NACH-II, shows that weighted combination methods can be utilized by a convex combination of the metrics developed for individual performance measures. Experimental results indicate that look-ahead batch control improves the weighted performance measure as compared to the benchmarks. Chapter V also shows that well known control strategies such as full batch, MBS and no-idling can be adapted to the bi-criteria control of batch processors by applying the look-ahead evaluation mechanism and the second stage of the NACH-II algorithm. The extension of the approach is possible to *m* different performance measures as long as the values of these measures can be determined by a look-ahead framework.

The control strategies developed in this research have complexities in the form of O(N) and $O(N^2)$ where N is the number of product types. This is a major advantage, considering the presence of high product-mix diversity in semiconductor manufacturing. Supported by automated shop-floor controllers, these control strategies can be efficiently implemented as a near real-time decision making tool for the control of batch processors in wafer fabrication facilities.

Finally, it should also be noted that the control strategies developed in this dissertation can be applied to more than semiconductor wafer fabrication. Any uninterruptable manufacturing unit that produces incompatible product types batch-wise is a candidate for implementing these control strategies. For instance, this work might be applicable to oven systems that are used in the aircraft industry to harden synthetic parts.

6.2 Future research directions

While this dissertation demonstrates the potential for utilizing near-future information and incorporating upstream re-sequencing in the dynamic control of batch processors, there remain issues worth exploring. This section discusses the areas for further research.

The first area for further research concerns the modification of the proposed control strategies for multiple processor scenarios. It should be noted that all of the proposed control strategies have been developed on a serial-batch processor system setting. It would be possible to extend these strategies to the case of multiple serial processors feeding a batch processor. For the control strategies that utilize near-future information without any re-sequencing approach, near-future information coming from multiple upstream stations can be combined into a single list of future events, and decisions can be based on this information. For the control strategies that use a re-sequencing approach, the re-sequencing scenarios for each upstream station need to be considered separately. This might require an additional stage in the algorithms since a best decision should be determined for each upstream station and the best of these must be selected.

Another area for further research addresses due-date information of the products. In Chapters IV and V, product due-dates are assumed to be strict points in time. Although wafer lots are released in wafer fabrication based on the final products' due-dates that are typically agreed upon with the customer, there is always an uncertainty in determining product due-dates for an intermediate processing step. Therefore, the applicability of the proposed approaches would be broadened by considering the batch process due-dates in an uncertainty interval. This way, earliness and tardiness of the products can be studied by penalizing the products that are completed before or after their due-date interval.

Chapter V demonstrated that look-ahead control algorithms developed for single performance measures can be merged effectively for the case where batching decisions need to consider both performance measures. A weighted global criterion method is followed, assuming the decision maker has made an appropriate choice of relative weights for the two performance measures. Further research can be directed to the methods in which there is no need for criteria weights such as fractional deviation methods. Also, the use of re-sequencing in the bi-criteria problem setting can be studied as an extension of this work.

Finally, there are numerous other production characteristics relating to the serial-batch processor system that are not included in this research. Some of these additional productions characteristics are as follows:

- Setup times when the batch processor switches between product types
- Yield issues on both processors
- Transportation times between serial and batch processors

Since these characteristics may influence the effectiveness of the control strategies developed in this research, future research that considers these characteristics might lead to control strategies which are more applicable in wafer fabrication.

REFERENCES

Ahmadi, J.H., Ahmadi, R.H., Dasu, S. and Tang, C.S. (1992) Batching and scheduling jobs on batch and discrete processors. *Operations Research*, **40**, 750-63.

Akcali, E., Uzsoy, R., Hiscock, D.G., Moser, A.L. and Teyner, T.J. (2000) Alternative loading and dispatching policies for furnace operations in semiconductor manufacturing: a comparison by simulation, in *Proceedings of the 32nd Conference on Winter Simulation*, Orlando, FL, 1428-35.

Avramidis, A.N., Healy, K.J. and Uzsoy, R. (1998) Control of a batch-processing machine: a computational approach. *International Journal of Production Research*, **36**, 3167-3181.

Azizoglu, M. and Webster, S. (2001) Scheduling a batch processing machine with incompatible job families. *Computers and Industrial Engineering*, **39**, 325-35.

Balasubramanian, H., Monch, L., Fowler, J. and Pfund, M. (2004) Genetic algorithm based scheduling of parallel batch machines with incompatible job families to minimize total weighted tardiness. *International Journal of Production Research*, **42**, 1621-1638.

Chandru, V., Lee, C.Y. and Uzsoy, R. (1993a) Minimizing total completion time on a batch processing machine with job families. *Operations Research Letters*, **13**, 61-5.

Chandru, V., Lee, C.Y. and Uzsoy, R. (1993b) Minimizing total completion time on batch processing machines. *International Journal of Production Research*, **31**, 2097-2121.

Chaudhry, M.L. and Templeton, J.G.C. (1983) *A First Course in Bulk Queues*, Wiley, New York

Cheraghi, S.H., Vishwaram, V. and Krishnan, K.K. (2003) Scheduling a single batch-processing machine with disagreeable ready times and due dates. *International Journal of Industrial Engineering : Theory Applications and Practice*, **10**, 175-187.

Cigolini, R., Perona, M., Portioli, A. and Zambelli, T. (2002) A new dynamic look-ahead scheduling procedure for batching machines. *Journal of Scheduling*, **5**, 185-204.

Coello, C.A.C. (1999) An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends, in *Proceedings of the Congress on Evolutionary Computation*, Washington, DC, 3-13.

Deb, R.K. and Serfozo, R.F. (1973) Optimal control of batch service queues. *Advances in Applied Probability*, **5**, 340-361.

Dobson, G. and Nambimadom, R.S. (2001) The batch loading and scheduling problem. *Operations Research*, **49**, 52-65.

Duenyas, I. and Neale, J.J. (1997) Stochastic scheduling of a batch processing machine with incompatible job families. *Annals of Operations Research*, **70**, 191-220.

Dupont, L. and Dhaenens-Flipo, C. (2002) Minimizing the makespan on a batch machine with non-identical job sizes: an exact procedure. *Computers & Operations Research*, **29**, 807-19.

Fowler, J.W., Hogg, G.L. and Phillips, D.T. (2000) Control of multiproduct bulk server diffusion/oxidation processes. 2. Multiple servers. *IIE Transactions*, **32**, 167-76.

Fowler, J.W., Phillips, D.T. and Hogg, G.L. (1992) Real-time control of multiproduct bulk-service semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, **5**, 158-63.

Ganesan, V.K., Gupta, A.K. and Iyer, S.A. (2004) Bi-objective schedule control of batch processes in semiconductor manufacturing, in *IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, **2**, 1077-1082.

Ghazvini, F.J. and Dupont, L. (1998) Minimizing mean flow times criteria on a single batch processing machine with non-identical jobs sizes. *International Journal of Production Economics*, **55**, 273-80.

Glassey, C.R. and Weng, W.W. (1991) Dynamic batching heuristic for simultaneous processing. *IEEE Transactions on Semiconductor Manufacturing*, **4**, 77-82.

Gupta, A.K. and Sivakumar, A.I. (2007) Controlling delivery performance in semiconductor manufacturing using Look Ahead Batching. *International Journal of Production Research*, **45**, 591-613.

Gurnani, H., Anupindi, R. and Akella, R. (1992) Control of batch processing systems in semiconductor wafer fabrication facilities. *IEEE Transactions on Semiconductor Manufacturing*, **5**, 319-28.

Habenicht, I. and Monch, L. (2005) Simulation-based assessment of batching strategies in a multi product semiconductor wafer fabrication facility. *International Journal of Simulation: Systems, Science & Technology*, **6**, 3-13.

Hochbaum, D.S. and Landy, D. (1997) Scheduling semiconductor burn-in operations to minimize total flowtime. *Operations Research*, **45**, 874-85.

Jolai, F. (2005) Minimizing number of tardy jobs on a batch processing machine with incompatible job families. *European Journal of Operational Research*, **162**, 184-190.

Kim, B. and Kim, S. (2002) Application of genetic algorithms for scheduling batch-discrete production system. *Production Planning and Control*, **13**, 155-65.

Kim, Y.D., Kim, J.G., Choi, B. and Kim, H.U. (2001) Production scheduling in a semiconductor wafer fabrication facility producing multiple product types with distinct due dates. *IEEE Transactions on Robotics and Automation*, **17**, 589-598.

Law, A.M., Kelton, W.D. (1991) *Simulation Modeling and Analysis*, 2nd ed. McGraw-Hill, New York.

Lee, C.Y. and Uzsoy, R. (1999) Minimizing makespan on a single batch processing machine with dynamic job arrivals. *International Journal of Production Research*, **37**, 219-36.

Li, C.-L. and Lee, C.-Y. (1997) Scheduling with agreeable release times and due dates on a batch processing machine. *European Journal of Operational Research*, **96**, 564-569.

Liu, Z. and Yu, W. (2000) Scheduling one batch processor subject to job release dates. *Discrete Applied Mathematics*, **105**, 129-36.

Mason, S.J., Fowler, J.W. and Carlyle, W.M. (2002) A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *Journal of Scheduling*, **5**, 247-262.

Mathirajan, M. and Sivakumar, A.I. (2006a) A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *International Journal of Advanced Manufacturing Technology*, **29**, 990-1001.

Mathirajan, M. and Sivakumar, A.I. (2006b) Minimizing total weighted tardiness on heterogeneous batch processing machines with incompatible job families. *International Journal of Advanced Manufacturing Technology*, **28**, 1038-47.

Mehta, S.V. and Uzsoy, R. (1998) Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE Transactions*, **30**, 165-178.

Monch, L., Balasubramanian, H., Fowler, J. and Pfund, M. (2005) Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. *Computers & Operations Research*, **32**, 2731-50.

Neale, J.J. and Duenyas, I. (2000) Control of manufacturing networks which contain a batch processing machine. *IIE Transactions*, **32**, 1027-41.

Neale, J.J. and Duenyas, I. (2003) Control of a batch processing machine serving compatible job families, *IIE Transactions*, **35**, 699 - 710.

Neuts, M.F. (1967) A general class of bulk queues with Poisson input. *The Annals of Mathematical Statistics*, **38**, 759-770.

Perez, I.C., Fowler, J.W. and Carlyle, W.M. (2005) Minimizing total weighted tardiness on a single batch process machine with incompatible job families. *Computers & Operations Research*, **32**, 327-341.

Pfund, M.E., Mason, S.J. and Fowler, J.W. (2006) Semiconductor Manufacturing Scheduling and Dispatching, Chapter 9 in *Handbook of Production Scheduling*, Herrmann, J.W. (ed.), Springer, New York, 213-241.

Reichelt, D. and Monch, L. (2006) *Multiobjective Scheduling of Jobs with Incompatible Families on Parallel Batch Machines*, Lecture Notes in Computer Science, Springer-Verlag, Berlin, 209-21.

Robinson, J.K., Fowler, J.W. and Bard, J.F. (1995) Use of upstream and downstream information in scheduling semiconductor batch operations. *International Journal of Production Research*, **33**, 1849-1869.

Solomon, L., Fowler, J.W., Pfund, M. and Jensen, P.H. (2002) The inclusion of future arrivals and downstream setups into wafer fabrication batch processing decisions. *Journal of Electronics Manufacturing*, **11**, 149-159.

Sung, C.S., Choung, Y.I., Hong, J.M. and Kim, Y.H. (2002) Minimizing makespan on a single burn-in oven with job families and dynamic job arrivals. *Computers & Operations Research*, **29**, 995-1007.

T'kindt, V., Billaut, J.-C. and Scott, H. (2006) *Multicriteria Scheduling: Theory, Models and Algorithms*, Springer-Verlag, New York

Tabucanon, M.T. (1988) *Multiple Criteria Decision Making in Industry,* Elsevier Science Publishers, Amsterdam, Netherlands.

Tangudu, S.K. and Kurz, M.E. (2006) A branch and bound algorithm to minimize total weighted tardiness on a single batch processing machine with ready times and incompatible job families. *Production Planning and Control*, **17**, 728-41.

Uzsoy, R. (1994) Scheduling a single batch processing machine with non-identical job sizes. *International Journal of Production Research*, **32**, 1615-1635.

Uzsoy, R. (1995) Scheduling batch processing machines with incompatible job families. *International Journal of Production Research*, **33**, 2685-708.

Uzsoy, R. and Yaoyu, Y. (1997) Minimizing total weighted completion time on a single batch processing machine. *Production and Operations Management*, **6**, 57-73.

Van Der Zee, D.J. (2002) Adaptive scheduling of batch servers in flow shops. *International Journal of Production Research*, **40**, 2811-33.

Van Der Zee, D.J. (2007) Dynamic scheduling of batch-processing machines with non-identical product sizes. *International Journal of Production Research*, **45**, 2327-2349.

Van Der Zee, D.J., Van Harten, A. and Schuur, P. (2001) On-line scheduling of multi-server batch operations. *IIE Transactions*, **33**, 569-586.

Van Der Zee, D.J., van Harten, A. and Schuur, P.C. (1997) Dynamic job assignment heuristics for multi-server batch operations-a cost based approach. *International Journal of Production Research*, **35**, 3063-93.

Vepsalainen, A.P.J. and Morton, T.E. (1987) Priority rules for job shops with weighted tardiness costs. *Management Science,* **33**, 1035-47.

Weng, W.W. and Leachman, R.C. (1993) An improved methodology for real-time production decisions at batch-process work stations. *IEEE Transactions on Semiconductor Manufacturing*, **6**, 219-25.

APPENDIX A

PAIRED-T TEST RESULTS FOR CHAPTER III

The short codes used in the tables for the production characteristics and their settings are as follows:

No	Factor/ Codes	Settings/Codes		
1	Number of Products - PN	2		
		5		
		10		
2	Product-mix - PM	Equal - E	2 products	(0.5, 0.5)
			5 products	(0.2, 0.2, 0.2, 0.2, 0.2)
			10 products	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different - D	2 products	(0.7, 0.3)
			5 products	(0.35, 0.35, 0.1, 0.1, 0.1)
			10 products	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
3	Batch Processor Capacity by Products - C	Equal - E	2 products	(5, 5)
			5 products	(5, 5, 5, 5, 5)
			10 products	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different - D	2 products	(7,2)
			5 products	(7, 6, 5, 4, 3)
			10 products	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
4	Batch Processing Time By Product - PT	Equal - E	2 products	(25, 25)
			5 products	(25, 25, 25, 25, 25)
			10 products	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different - D	2 products	(40, 10)
			5 products	(40, 30, 25, 20, 10)
			10 products	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
5	Traffic Intensity - TI	0.4		
		0.6		
		0.8		

Δ = The mean of the normalized waiting time differences obtained by 10 replications of NARCH and the benchmark strategy that the column belongs to. For example, if X~NACH and Y~NARCH then,

$$\Delta = \frac{1}{10}\sum_{i=1}^{10}(X_i - Y_i)$$

σ = The standard deviation of the replication differences

conf = The half-width of the 95% confidence interval of the replication differences

sign = + if there is a significant performance difference observed with NARCH

- if there is not a significant performance difference observed with NARCH

						MBS				NACH			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.2095	0.0037	0.0023	+	0.0641	0.0044	0.0027	+
2	2	0.4	E	E	D	0.3805	0.0051	0.0032	+	0.0302	0.0043	0.0027	+
3	2	0.4	E	D	E	0.0965	0.0047	0.0029	+	-0.0686	0.0075	0.0047	-
4	2	0.4	E	D	D	0.4410	0.0035	0.0022	+	0.0108	0.0105	0.0065	+
5	2	0.4	D	E	E	0.2377	0.0038	0.0024	+	0.0430	0.0056	0.0034	+
6	2	0.4	D	E	D	0.1664	0.0034	0.0021	+	-0.0179	0.0034	0.0021	-
7	2	0.4	D	D	E	0.2979	0.0045	0.0028	+	0.0699	0.0033	0.0020	+
8	2	0.4	D	D	D	0.5131	0.0036	0.0022	+	0.0580	0.0049	0.0030	+
9	2	0.6	E	E	E	0.1960	0.0022	0.0014	+	0.0973	0.0029	0.0018	+
10	2	0.6	E	E	D	0.3208	0.0021	0.0013	+	0.0731	0.0020	0.0012	+
11	2	0.6	E	D	E	0.2565	0.0041	0.0025	+	0.0779	0.0066	0.0041	+
12	2	0.6	E	D	D	0.4552	0.0025	0.0015	+	0.0854	0.0045	0.0028	+
13	2	0.6	D	E	E	0.2315	0.0021	0.0013	+	0.1103	0.0028	0.0017	+
14	2	0.6	D	E	D	0.3309	0.0018	0.0011	+	0.0745	0.0027	0.0017	+
15	2	0.6	D	D	E	0.2358	0.0025	0.0015	+	0.0590	0.0054	0.0034	+
16	2	0.6	D	D	D	0.3949	0.0022	0.0014	+	0.0685	0.0009	0.0006	+
17	2	0.8	E	E	E	0.1299	0.0008	0.0005	+	0.0881	0.0031	0.0019	+
18	2	0.8	E	E	D	0.2604	0.0025	0.0016	+	0.0932	0.0042	0.0026	+
19	2	0.8	E	D	E	0.5062	0.0108	0.0067	+	0.0899	0.0078	0.0048	+
20	2	0.8	E	D	D	0.3021	0.0018	0.0011	+	0.0759	0.0034	0.0021	+
21	2	0.8	D	E	E	0.1418	0.0007	0.0005	+	0.0860	0.0029	0.0018	+
22	2	0.8	D	E	D	0.2624	0.0022	0.0014	+	0.0596	0.0033	0.0021	+
23	2	0.8	D	D	E	0.2157	0.0021	0.0013	+	0.0241	0.0036	0.0022	+
24	2	0.8	D	D	D	0.2339	0.0016	0.0010	+	0.0513	0.0025	0.0016	+
25	5	0.4	E	E	E	0.4805	0.0074	0.0046	+	0.2756	0.0114	0.0071	+
26	5	0.4	E	E	D	0.6586	0.0065	0.0040	+	0.2948	0.0126	0.0078	+
27	5	0.4	E	D	E	0.5547	0.0089	0.0055	+	0.3083	0.0117	0.0073	+
28	5	0.4	E	D	D	0.6685	0.0075	0.0047	+	0.2976	0.0156	0.0097	+
29	5	0.4	D	E	E	0.6215	0.0074	0.0046	+	0.2718	0.0110	0.0068	+
30	5	0.4	D	E	D	0.8009	0.0080	0.0050	+	0.2461	0.0084	0.0052	+
31	5	0.4	D	D	E	0.6871	0.0067	0.0041	+	0.3141	0.0084	0.0052	+
32	5	0.4	D	D	D	0.9766	0.0072	0.0045	+	0.2654	0.0072	0.0045	+
33	5	0.6	E	E	E	0.2969	0.0023	0.0014	+	0.1801	0.0070	0.0043	+
34	5	0.6	E	E	D	0.3190	0.0036	0.0022	+	0.2094	0.0087	0.0054	+
35	5	0.6	E	D	E	0.5831	0.0032	0.0020	+	0.3279	0.0091	0.0056	+
36	5	0.6	E	D	D	0.6943	0.0026	0.0016	+	0.4040	0.0129	0.0080	+

37	5	0.6	D	E	E	0.3786	0.0025	0.0015	+	0.2048	0.0083	0.0051	+
38	5	0.6	D	E	D	0.4421	0.0039	0.0024	+	0.1625	0.0079	0.0049	+
39	5	0.6	D	D	E	0.4152	0.0029	0.0018	+	0.1949	0.0048	0.0029	+
40	5	0.6	D	D	D	0.5827	0.0022	0.0013	+	0.2069	0.0058	0.0036	+
41	5	0.8	E	E	E	0.1366	0.0011	0.0007	+	0.0928	0.0035	0.0021	+
42	5	0.8	E	E	D	0.1886	0.0012	0.0008	+	0.1233	0.0050	0.0031	+
43	5	0.8	E	D	E	0.4415	0.0029	0.0018	+	0.1847	0.0070	0.0044	+
44	5	0.8	E	D	D	0.7332	0.0021	0.0013	+	0.3663	0.0070	0.0043	+
45	5	0.8	D	E	E	0.2428	0.0010	0.0007	+	0.1838	0.0050	0.0031	+
46	5	0.8	D	E	D	0.4015	0.0019	0.0012	+	0.2503	0.0045	0.0028	+
47	5	0.8	D	D	E	0.3976	0.0017	0.0010	+	0.0784	0.0053	0.0033	+
48	5	0.8	D	D	D	0.4029	0.0015	0.0009	+	0.0900	0.0060	0.0037	+
49	10	0.4	E	E	E	0.7376	0.0077	0.0047	+	0.4731	0.0122	0.0076	+
50	10	0.4	E	E	D	0.3460	0.0090	0.0056	+	0.4610	0.0239	0.0148	+
51	10	0.4	E	D	E	0.6379	0.0082	0.0051	+	0.3648	0.0137	0.0085	+
52	10	0.4	E	D	D	0.2927	0.0103	0.0064	+	0.4889	0.0178	0.0110	+
53	10	0.4	D	E	E	1.0654	0.0164	0.0102	+	0.5785	0.0160	0.0099	+
54	10	0.4	D	E	D	1.3939	0.0102	0.0063	+	0.3642	0.0135	0.0084	+
55	10	0.4	D	D	E	0.9444	0.0052	0.0032	+	0.5107	0.0153	0.0095	+
56	10	0.4	D	D	D	1.1149	0.0075	0.0047	+	0.2784	0.0077	0.0048	+
57	10	0.6	E	E	E	0.3525	0.0030	0.0019	+	0.2301	0.0099	0.0061	+
58	10	0.6	E	E	D	0.4088	0.0034	0.0021	+	0.2785	0.0139	0.0086	+
59	10	0.6	E	D	E	0.6087	0.0028	0.0017	+	0.3285	0.0097	0.0060	+
60	10	0.6	E	D	D	0.7386	0.0032	0.0020	+	0.3688	0.0180	0.0112	+
61	10	0.6	D	E	E	0.5311	0.0019	0.0012	+	0.1940	0.0120	0.0075	+
62	10	0.6	D	E	D	0.6418	0.0036	0.0022	+	0.2133	0.0130	0.0081	+
63	10	0.6	D	D	E	0.6905	0.0039	0.0024	+	0.3008	0.0099	0.0062	+
64	10	0.6	D	D	D	0.9972	0.0064	0.0039	+	0.4091	0.0165	0.0102	+
65	10	0.8	E	E	E	0.2955	0.0013	0.0008	+	0.1233	0.0072	0.0045	+
66	10	0.8	E	E	D	0.6764	0.0020	0.0012	+	0.1530	0.0114	0.0070	+
67	10	0.8	E	D	E	1.4290	0.0046	0.0028	+	0.3070	0.0133	0.0082	+
68	10	0.8	E	D	D	0.6761	0.0019	0.0012	+	0.4424	0.0130	0.0081	+
69	10	0.8	D	E	E	0.4192	0.0012	0.0007	+	0.1054	0.0101	0.0063	+
70	10	0.8	D	E	D	0.6917	0.0042	0.0026	+	0.6242	0.0158	0.0098	+
71	10	0.8	D	D	E	1.1388	0.0044	0.0028	+	0.1127	0.0126	0.0078	+
72	10	0.8	D	D	D	1.2910	0.0058	0.0036	+	0.5110	0.0094	0.0058	+

						RHCR				DJAH			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.0514	0.0050	0.0031	+	0.0496	0.0047	0.0029	+
2	2	0.4	E	E	D	0.0753	0.0058	0.0036	+	0.0206	0.0042	0.0026	+
3	2	0.4	E	D	E	-0.0852	0.0065	0.0040	-	-0.0854	0.0066	0.0041	-
4	2	0.4	E	D	D	0.0516	0.0113	0.0070	+	0.0041	0.0094	0.0058	-
5	2	0.4	D	E	E	0.0367	0.0046	0.0028	+	0.0362	0.0043	0.0027	+
6	2	0.4	D	E	D	0.0379	0.0030	0.0019	+	-0.0177	0.0035	0.0021	-
7	2	0.4	D	D	E	0.0627	0.0037	0.0023	+	0.0672	0.0032	0.0020	+
8	2	0.4	D	D	D	0.0579	0.0063	0.0039	+	0.0555	0.0047	0.0029	+
9	2	0.6	E	E	E	0.0770	0.0028	0.0018	+	0.0748	0.0025	0.0015	+
10	2	0.6	E	E	D	0.0928	0.0061	0.0038	+	0.0618	0.0037	0.0023	+
11	2	0.6	E	D	E	0.0484	0.0067	0.0042	+	0.0488	0.0055	0.0034	+
12	2	0.6	E	D	D	0.0925	0.0056	0.0035	+	0.0953	0.0046	0.0029	+
13	2	0.6	D	E	E	0.0951	0.0042	0.0026	+	0.0934	0.0038	0.0024	+
14	2	0.6	D	E	D	0.0915	0.0047	0.0029	+	0.0772	0.0025	0.0016	+
15	2	0.6	D	D	E	0.0506	0.0052	0.0032	+	0.0547	0.0051	0.0031	+
16	2	0.6	D	D	D	0.0903	0.0016	0.0010	+	0.0612	0.0016	0.0010	+
17	2	0.8	E	E	E	0.0576	0.0041	0.0025	+	0.0558	0.0041	0.0025	+
18	2	0.8	E	E	D	0.1179	0.0069	0.0043	+	0.0742	0.0037	0.0023	+
19	2	0.8	E	D	E	0.0576	0.0075	0.0046	+	0.0578	0.0056	0.0035	+
20	2	0.8	E	D	D	0.0954	0.0041	0.0025	+	0.0743	0.0045	0.0028	+
21	2	0.8	D	E	E	0.0568	0.0034	0.0021	+	0.0555	0.0034	0.0021	+
22	2	0.8	D	E	D	0.0658	0.0048	0.0030	+	0.0599	0.0041	0.0025	+
23	2	0.8	D	D	E	0.0249	0.0041	0.0025	+	0.0194	0.0033	0.0020	+
24	2	0.8	D	D	D	0.0646	0.0024	0.0015	+	0.0501	0.0032	0.0020	+
25	5	0.4	E	E	E	0.3032	0.0147	0.0091	+	0.2946	0.0133	0.0082	+
26	5	0.4	E	E	D	0.2746	0.0124	0.0077	+	0.2082	0.0102	0.0063	+
27	5	0.4	E	D	E	0.3319	0.0112	0.0069	+	0.3295	0.0100	0.0062	+
28	5	0.4	E	D	D	0.2739	0.0154	0.0096	+	0.2708	0.0151	0.0093	+
29	5	0.4	D	E	E	0.2730	0.0116	0.0072	+	0.2792	0.0131	0.0081	+
30	5	0.4	D	E	D	0.3393	0.0085	0.0052	+	0.2664	0.0089	0.0055	+
31	5	0.4	D	D	E	0.2818	0.0105	0.0065	+	0.2859	0.0114	0.0071	+
32	5	0.4	D	D	D	0.3336	0.0101	0.0063	+	0.2972	0.0079	0.0049	+
33	5	0.6	E	E	E	0.1933	0.0077	0.0048	+	0.1747	0.0088	0.0055	+
34	5	0.6	E	E	D	0.1666	0.0130	0.0081	+	0.1525	0.0090	0.0056	+
35	5	0.6	E	D	E	0.3219	0.0043	0.0026	+	0.3133	0.0081	0.0051	+
36	5	0.6	E	D	D	0.2517	0.0134	0.0083	+	0.1694	0.0110	0.0068	+
37	5	0.6	D	E	E	0.2461	0.0099	0.0061	+	0.2347	0.0098	0.0061	+
38	5	0.6	D	E	D	0.2297	0.0112	0.0070	+	0.1712	0.0092	0.0057	+

39	5	0.6	D	D	E	0.1972	0.0054	0.0034	+	0.1877	0.0061	0.0038	+
40	5	0.6	D	D	D	0.2911	0.0113	0.0070	+	0.1806	0.0082	0.0051	+
41	5	0.8	E	E	E	0.0761	0.0038	0.0024	+	0.0592	0.0039	0.0024	+
42	5	0.8	E	E	D	0.1390	0.0109	0.0067	+	0.1005	0.0042	0.0026	+
43	5	0.8	E	D	E	0.1758	0.0045	0.0028	+	0.1656	0.0072	0.0044	+
44	5	0.8	E	D	D	0.1735	0.0140	0.0087	+	0.0630	0.0096	0.0059	+
45	5	0.8	D	E	E	0.1858	0.0052	0.0032	+	0.1696	0.0030	0.0019	+
46	5	0.8	D	E	D	0.2061	0.0094	0.0058	+	0.1231	0.0050	0.0031	+
47	5	0.8	D	D	E	0.0991	0.0048	0.0030	+	0.0911	0.0060	0.0037	+
48	5	0.8	D	D	D	0.0749	0.0104	0.0064	+	0.0539	0.0075	0.0046	+
49	10	0.4	E	E	E	0.4629	0.0199	0.0123	+	0.4670	0.0202	0.0125	+
50	10	0.4	E	E	D	0.3388	0.0314	0.0194	+	0.3377	0.0325	0.0201	+
51	10	0.4	E	D	E	0.4576	0.0193	0.0120	+	0.4710	0.0185	0.0115	+
52	10	0.4	E	D	D	0.4065	0.0138	0.0086	+	0.3150	0.0135	0.0084	+
53	10	0.4	D	E	E	0.5963	0.0163	0.0101	+	0.6492	0.0158	0.0098	+
54	10	0.4	D	E	D	0.5796	0.0227	0.0141	+	0.4254	0.0180	0.0111	+
55	10	0.4	D	D	E	0.5211	0.0172	0.0107	+	0.5592	0.0148	0.0092	+
56	10	0.4	D	D	D	0.6586	0.0186	0.0115	+	0.4723	0.0139	0.0086	+
57	10	0.6	E	E	E	0.2970	0.0110	0.0068	+	0.2669	0.0129	0.0080	+
58	10	0.6	E	E	D	0.3352	0.0035	0.0022	+	0.2234	0.0197	0.0122	+
59	10	0.6	E	D	E	0.3817	0.0087	0.0054	+	0.3635	0.0097	0.0060	+
60	10	0.6	E	D	D	0.3061	0.0286	0.0178	+	0.2085	0.0215	0.0133	+
61	10	0.6	D	E	E	0.3528	0.0125	0.0078	+	0.3426	0.0132	0.0082	+
62	10	0.6	D	E	D	0.3729	0.0123	0.0076	+	0.1578	0.0109	0.0068	+
63	10	0.6	D	D	E	0.4746	0.0111	0.0069	+	0.4607	0.0144	0.0089	+
64	10	0.6	D	D	D	0.5106	0.0211	0.0131	+	0.2651	0.0160	0.0099	+
65	10	0.8	E	E	E	0.1396	0.0089	0.0055	+	0.1106	0.0080	0.0050	+
66	10	0.8	E	E	D	0.1844	0.0114	0.0070	+	0.1528	0.0130	0.0080	+
67	10	0.8	E	D	E	0.3214	0.0069	0.0043	+	0.3050	0.0134	0.0083	+
68	10	0.8	E	D	D	0.2280	0.0234	0.0145	+	0.1120	0.0137	0.0085	+
69	10	0.8	D	E	E	0.1846	0.0143	0.0089	+	0.1653	0.0122	0.0076	+
70	10	0.8	D	E	D	0.3934	0.0213	0.0132	+	0.2031	0.0116	0.0072	+
71	10	0.8	D	D	E	0.2084	0.0090	0.0056	+	0.1877	0.0091	0.0056	+
72	10	0.8	D	D	D	0.3311	0.0199	0.0123	+	0.0626	0.0121	0.0075	+

APPENDIX B

PAIRED-T TEST RESULTS FOR CHAPTER IV

The short codes used in the tables for the production characteristics and their settings are as follows:

No	Factor/ Codes	Settings/Codes		
1	Number of Products - PN	2		
		5		
		10		
2	Product-mix - PM	Equal - E	2 products	(0.5, 0.5)
			5 products	(0.2, 0.2, 0.2, 0.2, 0.2)
			10 products	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different - D	2 products	(0.7, 0.3)
			5 products	(0.35, 0.35, 0.1, 0.1, 0.1)
			10 products	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
3	Batch Processor Capacity by Products - C	Equal - E	2 products	(5, 5)
			5 products	(5, 5, 5, 5, 5)
			10 products	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different - D	2 products	(7,2)
			5 products	(7, 6, 5, 4, 3)
			10 products	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
4	Batch Processing Time By Product - PT	Equal - E	2 products	(25, 25)
			5 products	(25, 25, 25, 25, 25)
			10 products	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different - D	2 products	(40, 10)
			5 products	(40, 30, 25, 20, 10)
			10 products	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
5	Traffic Intensity - TI	0.4		
		0.6		
		0.8		

Δ = The mean of the normalized waiting time differences obtained by 10 replications of NACH-T and the benchmark strategy that the column belongs to. For example, if X~BATC-I and Y~NACH-T then,

$$\Delta = \frac{1}{10}\sum_{i=1}^{10}(X_i - Y_i)$$

σ = The standard deviation of the replication differences

conf = The half-width of the 95% confidence interval of the replication differences

sign = + if there is a significant performance difference observed with NACH-T

- if there is not a significant performance difference observed with NACH-T

a) Comparison of NACH-T with the Benchmarks:

						BATC-I				BATC-II			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.2910	0.0054	0.0034	+	0.2681	0.0050	0.0031	+
2	2	0.4	E	E	D	0.1843	0.0063	0.0039	+	0.2190	0.0062	0.0038	+
3	2	0.4	E	D	E	0.6376	0.0049	0.0030	+	0.5079	0.0014	0.0009	+
4	2	0.4	E	D	D	0.1687	0.0041	0.0026	+	0.1869	0.0044	0.0027	+
5	2	0.4	D	E	E	0.2987	0.0028	0.0018	+	0.3155	0.0029	0.0018	+
6	2	0.4	D	E	D	0.2521	0.0083	0.0051	+	0.2704	0.0098	0.0061	+
7	2	0.4	D	D	E	0.4032	0.0036	0.0023	+	0.4244	0.0041	0.0026	+
8	2	0.4	D	D	D	0.1732	0.0017	0.0010	+	0.1163	0.0013	0.0008	+
9	2	0.6	E	E	E	0.1448	0.0036	0.0022	+	0.2156	0.0059	0.0037	+
10	2	0.6	E	E	D	0.0373	0.0058	0.0036	+	0.0428	0.0052	0.0032	+
11	2	0.6	E	D	E	0.4155	0.0053	0.0033	+	0.2736	0.0035	0.0021	+
12	2	0.6	E	D	D	0.0287	0.0049	0.0030	+	0.0452	0.0020	0.0012	+
13	2	0.6	D	E	E	0.2180	0.0066	0.0041	+	0.1622	0.0033	0.0020	+
14	2	0.6	D	E	D	0.0682	0.0013	0.0008	+	0.0686	0.0040	0.0025	+
15	2	0.6	D	D	E	0.1999	0.0034	0.0021	+	0.1071	0.0035	0.0022	+
16	2	0.6	D	D	D	0.0731	0.0043	0.0027	+	0.0134	0.0040	0.0025	+
17	2	0.8	E	E	E	0.2946	0.0057	0.0035	+	0.1007	0.0055	0.0034	+
18	2	0.8	E	E	D	0.0898	0.0022	0.0013	+	0.0591	0.0019	0.0012	+
19	2	0.8	E	D	E	0.3112	0.0392	0.0243	+	0.1820	0.0294	0.0182	+
20	2	0.8	E	D	D	0.0641	0.0017	0.0010	+	0.0224	0.0027	0.0017	+
21	2	0.8	D	E	E	0.2920	0.0089	0.0055	+	0.1139	0.0037	0.0023	+
22	2	0.8	D	E	D	0.0795	0.0022	0.0014	+	0.0558	0.0005	0.0003	+
23	2	0.8	D	D	E	0.1567	0.0114	0.0071	+	0.0726	0.0154	0.0096	+
24	2	0.8	D	D	D	0.0053	0.0016	0.0010	+	0.0107	0.0040	0.0025	+
25	5	0.4	E	E	E	0.2681	0.0056	0.0035	+	0.2906	0.0088	0.0054	+
26	5	0.4	E	E	D	0.1617	0.0042	0.0026	+	0.1923	0.0041	0.0025	+
27	5	0.4	E	D	E	0.3392	0.0085	0.0053	+	0.3216	0.0127	0.0079	+
28	5	0.4	E	D	D	0.2259	0.0079	0.0049	+	0.2194	0.0081	0.0050	+
29	5	0.4	D	E	E	0.3212	0.0076	0.0047	+	0.3393	0.0065	0.0040	+
30	5	0.4	D	E	D	0.2779	0.0037	0.0023	+	0.2202	0.0049	0.0031	+
31	5	0.4	D	D	E	0.3175	0.0036	0.0022	+	0.2767	0.0032	0.0020	+
32	5	0.4	D	D	D	0.1725	0.0020	0.0012	+	0.0930	0.0011	0.0007	+
33	5	0.6	E	E	E	0.3617	0.0194	0.0120	+	0.2763	0.0046	0.0029	+
34	5	0.6	E	E	D	0.2286	0.0048	0.0030	+	0.2198	0.0038	0.0024	+
35	5	0.6	E	D	E	0.4597	0.0129	0.0080	+	0.3652	0.0067	0.0041	+

36	5	0.6	E	D	D	0.2815	0.0054	0.0034	+	0.1690	0.0045	0.0028	+
37	5	0.6	D	E	E	0.3910	0.0083	0.0051	+	0.3237	0.0074	0.0046	+
38	5	0.6	D	E	D	0.2144	0.0071	0.0044	+	0.2498	0.0082	0.0051	+
39	5	0.6	D	D	E	0.4397	0.0167	0.0104	+	0.2528	0.0078	0.0048	+
40	5	0.6	D	D	D	0.1292	0.0014	0.0009	+	0.1348	0.0025	0.0015	+
41	5	0.8	E	E	E	0.3208	0.0214	0.0133	+	0.2617	0.0034	0.0021	+
42	5	0.8	E	E	D	0.4586	0.0026	0.0016	+	0.3917	0.0022	0.0014	+
43	5	0.8	E	D	E	0.4168	0.0040	0.0025	+	0.3180	0.0047	0.0029	+
44	5	0.8	E	D	D	0.3228	0.0014	0.0009	+	0.2569	0.0009	0.0005	+
45	5	0.8	D	E	E	0.3157	0.0152	0.0094	+	0.3083	0.0054	0.0034	+
46	5	0.8	D	E	D	0.6091	0.0021	0.0013	+	0.4683	0.0029	0.0018	+
47	5	0.8	D	D	E	0.2652	0.0092	0.0057	+	0.1958	0.0067	0.0042	+
48	5	0.8	D	D	D	0.2381	0.0015	0.0009	+	0.1889	0.0020	0.0012	+
49	10	0.4	E	E	E	0.3698	0.0040	0.0025	+	0.3494	0.0046	0.0028	+
50	10	0.4	E	E	D	0.1688	0.0053	0.0033	+	0.1529	0.0039	0.0024	+
51	10	0.4	E	D	E	0.4626	0.0042	0.0026	+	0.4036	0.0053	0.0033	+
52	10	0.4	E	D	D	0.2421	0.0045	0.0028	+	0.1779	0.0055	0.0034	+
53	10	0.4	D	E	E	0.4903	0.0048	0.0029	+	0.3816	0.0071	0.0044	+
54	10	0.4	D	E	D	0.2694	0.0047	0.0029	+	0.1891	0.0065	0.0040	+
55	10	0.4	D	D	E	0.3690	0.0018	0.0011	+	0.3883	0.0014	0.0008	+
56	10	0.4	D	D	D	0.2785	0.0044	0.0027	+	0.1196	0.0040	0.0025	+
57	10	0.6	E	E	E	0.4911	0.0009	0.0006	+	0.5463	0.0025	0.0016	+
58	10	0.6	E	E	D	0.3283	0.0017	0.0010	+	0.2940	0.0011	0.0007	+
59	10	0.6	E	D	E	0.6297	0.0012	0.0007	+	0.6097	0.0025	0.0016	+
60	10	0.6	E	D	D	0.4777	0.0022	0.0013	+	0.2351	0.0014	0.0008	+
61	10	0.6	D	E	E	0.6187	0.0143	0.0089	+	0.6348	0.0206	0.0128	+
62	10	0.6	D	E	D	0.1573	0.0020	0.0012	+	0.2340	0.0022	0.0014	+
63	10	0.6	D	D	E	0.5839	0.0163	0.0101	+	0.5054	0.0116	0.0072	+
64	10	0.6	D	D	D	0.2641	0.0022	0.0013	+	0.1488	0.0023	0.0014	+
65	10	0.8	E	E	E	0.5518	0.0034	0.0021	+	0.5049	0.0033	0.0021	+
66	10	0.8	E	E	D	0.6167	0.0074	0.0046	+	0.5333	0.0071	0.0044	+
67	10	0.8	E	D	E	0.4617	0.0065	0.0040	+	0.4064	0.0070	0.0043	+
68	10	0.8	E	D	D	0.4784	0.0047	0.0029	+	0.4547	0.0025	0.0016	+
69	10	0.8	D	E	E	0.7800	0.0333	0.0206	+	0.5967	0.0105	0.0065	+
70	10	0.8	D	E	D	0.4380	0.0038	0.0024	+	0.4165	0.0043	0.0026	+
71	10	0.8	D	D	E	0.4371	0.0142	0.0088	+	0.4016	0.0142	0.0088	+
72	10	0.8	D	D	D	0.1753	0.0101	0.0062	+	0.1052	0.0119	0.0074	+

						MDBH				**EDD**			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.0555	0.0022	0.0014	+	0.1236	0.0056	0.0034	+
2	2	0.4	E	E	D	0.0733	0.0041	0.0026	+	0.1632	0.0073	0.0045	+
3	2	0.4	E	D	E	0.0622	0.0105	0.0065	+	0.1696	0.0109	0.0068	+
4	2	0.4	E	D	D	0.0123	0.0048	0.0029	+	0.1417	0.0038	0.0023	+
5	2	0.4	D	E	E	0.0532	0.0035	0.0021	+	0.1224	0.0035	0.0021	+
6	2	0.4	D	E	D	0.0325	0.0091	0.0057	+	0.1451	0.0090	0.0056	+
7	2	0.4	D	D	E	0.0433	0.0035	0.0021	+	0.1484	0.0040	0.0025	+
8	2	0.4	D	D	D	0.0361	0.0033	0.0021	+	0.0730	0.0018	0.0011	+
9	2	0.6	E	E	E	0.0548	0.0016	0.0010	+	0.1117	0.0176	0.0109	+
10	2	0.6	E	E	D	0.0168	0.0008	0.0005	+	0.0831	0.0029	0.0018	+
11	2	0.6	E	D	E	0.0477	0.0027	0.0017	+	0.2516	0.0032	0.0020	+
12	2	0.6	E	D	D	0.0198	0.0035	0.0022	+	0.1627	0.0012	0.0007	+
13	2	0.6	D	E	E	0.0421	0.0033	0.0020	+	0.0943	0.0040	0.0025	+
14	2	0.6	D	E	D	0.0530	0.0003	0.0002	+	0.1144	0.0018	0.0011	+
15	2	0.6	D	D	E	0.0462	0.0033	0.0020	+	0.1937	0.0058	0.0036	+
16	2	0.6	D	D	D	0.0278	0.0033	0.0020	+	0.0703	0.0029	0.0018	+
17	2	0.8	E	E	E	0.0400	0.0022	0.0013	+	0.1646	0.0175	0.0108	+
18	2	0.8	E	E	D	0.0269	0.0033	0.0021	+	0.0569	0.0039	0.0024	+
19	2	0.8	E	D	E	0.0340	0.0323	0.0200	+	0.7024	0.0708	0.0439	+
20	2	0.8	E	D	D	-0.0007	0.0040	0.0025	-	0.2277	0.0027	0.0017	+
21	2	0.8	D	E	E	0.0292	0.0025	0.0015	+	0.1718	0.0149	0.0092	+
22	2	0.8	D	E	D	0.0586	0.0003	0.0002	+	0.0745	0.0019	0.0012	+
23	2	0.8	D	D	E	0.0577	0.0071	0.0044	+	0.4280	0.0307	0.0190	+
24	2	0.8	D	D	D	0.0083	0.0032	0.0020	+	0.0273	0.0013	0.0008	+
25	5	0.4	E	E	E	0.1694	0.0179	0.0111	+	0.2602	0.0344	0.0213	+
26	5	0.4	E	E	D	0.0616	0.0041	0.0025	+	0.1577	0.0047	0.0029	+
27	5	0.4	E	D	E	0.1540	0.0090	0.0056	+	0.3008	0.0352	0.0218	+
28	5	0.4	E	D	D	0.1207	0.0095	0.0059	+	0.2460	0.0091	0.0056	+
29	5	0.4	D	E	E	0.0756	0.0168	0.0104	+	0.2074	0.0305	0.0189	+
30	5	0.4	D	E	D	0.0814	0.0023	0.0014	+	0.0472	0.0055	0.0034	+
31	5	0.4	D	D	E	0.0865	0.0043	0.0027	+	0.1914	0.0034	0.0021	+
32	5	0.4	D	D	D	0.0424	0.0025	0.0015	+	0.0277	0.0126	0.0078	+
33	5	0.6	E	E	E	0.1862	0.0216	0.0134	+	0.3749	0.0186	0.0115	+
34	5	0.6	E	E	D	0.0851	0.0022	0.0014	+	0.2550	0.0041	0.0026	+
35	5	0.6	E	D	E	0.1953	0.0119	0.0073	+	0.4793	0.0221	0.0137	+
36	5	0.6	E	D	D	0.0481	0.0043	0.0027	+	0.3450	0.0071	0.0044	+
37	5	0.6	D	E	E	0.0626	0.0127	0.0079	+	0.3132	0.0136	0.0084	+
38	5	0.6	D	E	D	0.0801	0.0022	0.0014	+	0.0451	0.0039	0.0024	+

39	5	0.6	D	D	E	0.0299	0.0094	0.0058	+	0.2414	0.0180	0.0112	+
40	5	0.6	D	D	D	0.0294	0.0019	0.0012	+	0.0008	0.0011	0.0007	+
41	5	0.8	E	E	E	0.1565	0.0145	0.0090	+	0.6973	0.0125	0.0077	+
42	5	0.8	E	E	D	0.1056	0.0026	0.0016	+	0.5955	0.0055	0.0034	+
43	5	0.8	E	D	E	0.1820	0.0037	0.0023	+	0.8679	0.0048	0.0030	+
44	5	0.8	E	D	D	0.0023	0.0016	0.0010	+	0.6448	0.0035	0.0021	+
45	5	0.8	D	E	E	0.0744	0.0079	0.0049	+	1.0803	0.0255	0.0158	+
46	5	0.8	D	E	D	0.0896	0.0046	0.0028	+	0.5657	0.0031	0.0019	+
47	5	0.8	D	D	E	0.0439	0.0061	0.0038	+	0.7168	0.0126	0.0078	+
48	5	0.8	D	D	D	0.0622	0.0012	0.0007	+	0.3129	0.0020	0.0012	+
49	10	0.4	E	E	E	0.6140	0.0048	0.0030	+	0.7277	0.0044	0.0028	+
50	10	0.4	E	E	D	0.3836	0.0036	0.0022	+	0.5106	0.0035	0.0021	+
51	10	0.4	E	D	E	0.5877	0.0070	0.0044	+	0.7722	0.0065	0.0040	+
52	10	0.4	E	D	D	0.3837	0.0046	0.0029	+	0.5845	0.0052	0.0032	+
53	10	0.4	D	E	E	0.1045	0.0056	0.0034	+	0.2859	0.0077	0.0048	+
54	10	0.4	D	E	D	0.0593	0.0054	0.0034	+	0.0429	0.0141	0.0087	+
55	10	0.4	D	D	E	0.0837	0.0018	0.0011	+	0.1886	0.0024	0.0015	+
56	10	0.4	D	D	D	0.1063	0.0034	0.0021	+	0.2251	0.0050	0.0031	+
57	10	0.6	E	E	E	0.6846	0.0017	0.0010	+	1.0406	0.0041	0.0026	+
58	10	0.6	E	E	D	0.4771	0.0014	0.0009	+	0.9576	0.0053	0.0033	+
59	10	0.6	E	D	E	0.6954	0.0025	0.0015	+	1.1744	0.0094	0.0058	+
60	10	0.6	E	D	D	0.4059	0.0043	0.0027	+	0.8618	0.0015	0.0009	+
61	10	0.6	D	E	E	0.1273	0.0274	0.0170	+	1.1097	0.0279	0.0173	+
62	10	0.6	D	E	D	0.0729	0.0020	0.0013	+	0.1691	0.0064	0.0039	+
63	10	0.6	D	D	E	0.0559	0.0168	0.0104	+	0.7508	0.0166	0.0103	+
64	10	0.6	D	D	D	0.1173	0.0041	0.0026	+	0.0234	0.0163	0.0101	+
65	10	0.8	E	E	E	0.5357	0.0025	0.0015	+	0.9488	0.0052	0.0032	+
66	10	0.8	E	E	D	0.4287	0.0041	0.0026	+	1.5168	0.0043	0.0026	+
67	10	0.8	E	D	E	0.5359	0.0010	0.0006	+	2.1198	0.0057	0.0035	+
68	10	0.8	E	D	D	0.2854	0.0082	0.0051	+	1.7659	0.0067	0.0042	+
69	10	0.8	D	E	E	0.2562	0.0079	0.0049	+	2.2647	0.0309	0.0192	+
70	10	0.8	D	E	D	0.0952	0.0017	0.0011	+	1.8928	0.0037	0.0023	+
71	10	0.8	D	D	E	0.0993	0.0083	0.0051	+	1.9213	0.0178	0.0110	+
72	10	0.8	D	D	D	0.2185	0.0015	0.0009	+	2.0955	0.0115	0.0071	+

b) Comparison of NARCH-T with the Benchmarks:

						NACH-T				**BATC-I**			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.0432	0.0021	0.0013	+	0.3343	0.0064	0.0040	+
2	2	0.4	E	E	D	0.0500	0.0003	0.0002	+	0.1843	0.0063	0.0039	+
3	2	0.4	E	D	E	0.0279	0.0434	0.0269	+	0.6516	0.0094	0.0058	+
4	2	0.4	E	D	D	0.0158	0.0005	0.0003	+	0.1844	0.0039	0.0024	+
5	2	0.4	D	E	E	0.0349	0.0003	0.0002	+	0.3336	0.0027	0.0017	+
6	2	0.4	D	E	D	0.0163	0.0042	0.0026	+	0.2684	0.0066	0.0041	+
7	2	0.4	D	D	E	0.0134	0.0099	0.0061	+	0.4166	0.0095	0.0059	+
8	2	0.4	D	D	D	0.0088	0.0004	0.0003	+	0.1821	0.0013	0.0008	+
9	2	0.6	E	E	E	0.0394	0.0019	0.0012	+	0.1842	0.0043	0.0026	+
10	2	0.6	E	E	D	0.0439	0.0004	0.0002	+	0.0811	0.0058	0.0036	+
11	2	0.6	E	D	E	0.0239	0.0028	0.0018	+	0.4394	0.0028	0.0017	+
12	2	0.6	E	D	D	0.0278	0.0031	0.0019	+	0.0565	0.0018	0.0011	+
13	2	0.6	D	E	E	0.0352	0.0027	0.0017	+	0.2532	0.0061	0.0038	+
14	2	0.6	D	E	D	0.0444	0.0002	0.0001	+	0.1126	0.0014	0.0009	+
15	2	0.6	D	D	E	0.0248	0.0002	0.0001	+	0.2247	0.0034	0.0021	+
16	2	0.6	D	D	D	0.0348	0.0002	0.0001	+	0.1079	0.0043	0.0027	+
17	2	0.8	E	E	E	0.0198	0.0012	0.0008	+	0.3144	0.0061	0.0038	+
18	2	0.8	E	E	D	0.0268	0.0002	0.0001	+	0.1166	0.0022	0.0014	+
19	2	0.8	E	D	E	0.0119	0.0321	0.0199	-	0.3231	0.0199	0.0123	+
20	2	0.8	E	D	D	0.0271	0.0002	0.0001	+	0.0912	0.0017	0.0010	+
21	2	0.8	D	E	E	0.0191	0.0019	0.0011	+	0.3110	0.0097	0.0060	+
22	2	0.8	D	E	D	0.0276	0.0003	0.0002	+	0.1071	0.0021	0.0013	+
23	2	0.8	D	D	E	0.0117	0.0034	0.0021	+	0.1684	0.0100	0.0062	+
24	2	0.8	D	D	D	0.0248	0.0002	0.0001	+	0.0301	0.0016	0.0010	+
25	5	0.4	E	E	E	0.1684	0.0084	0.0052	+	0.4366	0.0118	0.0073	+
26	5	0.4	E	E	D	0.1659	0.0026	0.0016	+	0.3276	0.0026	0.0016	+
27	5	0.4	E	D	E	0.1640	0.0066	0.0041	+	0.5032	0.0126	0.0078	+
28	5	0.4	E	D	D	0.0856	0.0082	0.0051	+	0.3115	0.0057	0.0035	+
29	5	0.4	D	E	E	0.1266	0.0063	0.0039	+	0.4479	0.0121	0.0075	+
30	5	0.4	D	E	D	0.1296	0.0030	0.0018	+	0.4075	0.0040	0.0025	+
31	5	0.4	D	D	E	0.1235	0.0035	0.0021	+	0.4410	0.0048	0.0030	+
32	5	0.4	D	D	D	0.1314	0.0009	0.0005	+	0.3039	0.0026	0.0016	+
33	5	0.6	E	E	E	0.0972	0.0048	0.0030	+	0.4589	0.0211	0.0131	+
34	5	0.6	E	E	D	0.1020	0.0026	0.0016	+	0.3306	0.0023	0.0014	+
35	5	0.6	E	D	E	0.0971	0.0069	0.0043	+	0.5568	0.0154	0.0095	+

36	5	0.6	E	D	D	0.0852	0.0031	0.0019	+	0.3667	0.0026	0.0016	+
37	5	0.6	D	E	E	0.0761	0.0066	0.0041	+	0.4671	0.0105	0.0065	+
38	5	0.6	D	E	D	0.0890	0.0006	0.0004	+	0.3034	0.0074	0.0046	+
39	5	0.6	D	D	E	0.0588	0.0053	0.0033	+	0.4985	0.0166	0.0103	+
40	5	0.6	D	D	D	0.0735	0.0006	0.0004	+	0.2027	0.0014	0.0009	+
41	5	0.8	E	E	E	0.0565	0.0035	0.0022	+	0.3772	0.0194	0.0121	+
42	5	0.8	E	E	D	0.0685	0.0005	0.0003	+	0.5271	0.0026	0.0016	+
43	5	0.8	E	D	E	0.0383	0.0006	0.0004	+	0.4551	0.0039	0.0024	+
44	5	0.8	E	D	D	0.0487	0.0006	0.0004	+	0.3715	0.0015	0.0010	+
45	5	0.8	D	E	E	0.0379	0.0034	0.0021	+	0.3535	0.0149	0.0092	+
46	5	0.8	D	E	D	0.0511	0.0006	0.0004	+	0.6602	0.0016	0.0010	+
47	5	0.8	D	D	E	0.0252	0.0034	0.0021	+	0.2904	0.0087	0.0054	+
48	5	0.8	D	D	D	0.0306	0.0004	0.0003	+	0.2687	0.0016	0.0010	+
49	10	0.4	E	E	E	0.2394	0.0046	0.0028	+	0.6091	0.0046	0.0029	+
50	10	0.4	E	E	D	0.2543	0.0041	0.0025	+	0.4230	0.0057	0.0036	+
51	10	0.4	E	D	E	0.1873	0.0037	0.0023	+	0.6499	0.0052	0.0032	+
52	10	0.4	E	D	D	0.2067	0.0035	0.0022	+	0.4488	0.0035	0.0022	+
53	10	0.4	D	E	E	0.1062	0.0036	0.0022	+	0.5965	0.0023	0.0014	+
54	10	0.4	D	E	D	0.1387	0.0027	0.0017	+	0.4081	0.0036	0.0022	+
55	10	0.4	D	D	E	0.1694	0.0029	0.0018	+	0.5384	0.0037	0.0023	+
56	10	0.4	D	D	D	0.1034	0.0031	0.0019	+	0.3819	0.0031	0.0019	+
57	10	0.6	E	E	E	0.0938	0.0009	0.0006	+	0.5849	0.0011	0.0007	+
58	10	0.6	E	E	D	0.1249	0.0011	0.0007	+	0.4532	0.0022	0.0014	+
59	10	0.6	E	D	E	0.0970	0.0008	0.0005	+	0.7267	0.0015	0.0009	+
60	10	0.6	E	D	D	0.1063	0.0009	0.0005	+	0.5840	0.0023	0.0014	+
61	10	0.6	D	E	E	0.0277	0.0115	0.0071	+	0.6464	0.0156	0.0097	+
62	10	0.6	D	E	D	0.0407	0.0019	0.0012	+	0.1980	0.0025	0.0016	+
63	10	0.6	D	D	E	0.0559	0.0092	0.0057	+	0.6398	0.0182	0.0113	+
64	10	0.6	D	D	D	0.0264	0.0015	0.0009	+	0.2905	0.0026	0.0016	+
65	10	0.8	E	E	E	0.0506	0.0007	0.0004	+	0.6024	0.0035	0.0021	+
66	10	0.8	E	E	D	0.0654	0.0008	0.0005	+	0.6821	0.0077	0.0048	+
67	10	0.8	E	D	E	0.0355	0.0010	0.0006	+	0.4972	0.0064	0.0039	+
68	10	0.8	E	D	D	0.0485	0.0009	0.0006	+	0.5269	0.0045	0.0028	+
69	10	0.8	D	E	E	0.0271	0.0114	0.0070	+	0.8071	0.0285	0.0177	+
70	10	0.8	D	E	D	0.0437	0.0011	0.0007	+	0.4817	0.0033	0.0021	+
71	10	0.8	D	D	E	0.0972	0.0074	0.0046	+	0.5343	0.0152	0.0094	+
72	10	0.8	D	D	D	0.0518	0.0016	0.0010	+	0.2271	0.0093	0.0058	+

						BATC-II				MDBH			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.3113	0.0063	0.0039	+	0.0987	0.0040	0.0025	+
2	2	0.4	E	E	D	0.2190	0.0062	0.0038	+	0.0733	0.0041	0.0026	+
3	2	0.4	E	D	E	0.5219	0.0086	0.0053	+	0.0761	0.0108	0.0067	+
4	2	0.4	E	D	D	0.2026	0.0041	0.0026	+	0.0281	0.0048	0.0030	+
5	2	0.4	D	E	E	0.3504	0.0027	0.0017	+	0.0881	0.0033	0.0021	+
6	2	0.4	D	E	D	0.2867	0.0073	0.0045	+	0.0488	0.0071	0.0044	+
7	2	0.4	D	D	E	0.4377	0.0095	0.0059	+	0.0567	0.0093	0.0058	+
8	2	0.4	D	D	D	0.1251	0.0011	0.0007	+	0.0449	0.0031	0.0019	+
9	2	0.6	E	E	E	0.2550	0.0056	0.0035	+	0.0942	0.0027	0.0017	+
10	2	0.6	E	E	D	0.0867	0.0053	0.0033	+	0.0607	0.0007	0.0004	+
11	2	0.6	E	D	E	0.2975	0.0011	0.0007	+	0.0716	0.0007	0.0004	+
12	2	0.6	E	D	D	0.0730	0.0011	0.0007	+	0.0476	0.0006	0.0004	+
13	2	0.6	D	E	E	0.1974	0.0037	0.0023	+	0.0772	0.0021	0.0013	+
14	2	0.6	D	E	D	0.1131	0.0041	0.0026	+	0.0974	0.0004	0.0002	+
15	2	0.6	D	D	E	0.1319	0.0036	0.0022	+	0.0710	0.0032	0.0020	+
16	2	0.6	D	D	D	0.0482	0.0040	0.0025	+	0.0626	0.0033	0.0020	+
17	2	0.8	E	E	E	0.1204	0.0062	0.0038	+	0.0597	0.0023	0.0014	+
18	2	0.8	E	E	D	0.0859	0.0020	0.0012	+	0.0537	0.0034	0.0021	+
19	2	0.8	E	D	E	0.1938	0.0112	0.0070	+	0.0459	0.0629	0.0390	+
20	2	0.8	E	D	D	0.0495	0.0026	0.0016	+	0.0264	0.0039	0.0024	+
21	2	0.8	D	E	E	0.1329	0.0037	0.0023	+	0.0483	0.0025	0.0016	+
22	2	0.8	D	E	D	0.0834	0.0005	0.0003	+	0.0862	0.0003	0.0002	+
23	2	0.8	D	D	E	0.0843	0.0144	0.0089	+	0.0694	0.0088	0.0055	+
24	2	0.8	D	D	D	0.0355	0.0039	0.0024	+	0.0331	0.0032	0.0020	+
25	5	0.4	E	E	E	0.4590	0.0123	0.0076	+	0.3378	0.0231	0.0143	+
26	5	0.4	E	E	D	0.3582	0.0025	0.0016	+	0.2276	0.0025	0.0015	+
27	5	0.4	E	D	E	0.4856	0.0150	0.0093	+	0.3180	0.0108	0.0067	+
28	5	0.4	E	D	D	0.3050	0.0054	0.0034	+	0.2063	0.0021	0.0013	+
29	5	0.4	D	E	E	0.4659	0.0112	0.0069	+	0.2023	0.0189	0.0117	+
30	5	0.4	D	E	D	0.3498	0.0036	0.0022	+	0.2110	0.0015	0.0009	+
31	5	0.4	D	D	E	0.4002	0.0008	0.0005	+	0.2100	0.0052	0.0033	+
32	5	0.4	D	D	D	0.2244	0.0018	0.0011	+	0.1738	0.0030	0.0019	+
33	5	0.6	E	E	E	0.3735	0.0072	0.0045	+	0.2834	0.0210	0.0130	+
34	5	0.6	E	E	D	0.3218	0.0015	0.0009	+	0.1871	0.0010	0.0006	+
35	5	0.6	E	D	E	0.4623	0.0112	0.0069	+	0.2924	0.0126	0.0078	+
36	5	0.6	E	D	D	0.2542	0.0016	0.0010	+	0.1333	0.0015	0.0009	+
37	5	0.6	D	E	E	0.3998	0.0117	0.0073	+	0.1387	0.0136	0.0084	+
38	5	0.6	D	E	D	0.3388	0.0085	0.0053	+	0.1691	0.0025	0.0016	+

39	5	0.6	D	D	E	0.3116	0.0067	0.0041	+	0.0887	0.0064	0.0040	+
40	5	0.6	D	D	D	0.2083	0.0024	0.0015	+	0.1029	0.0018	0.0011	+
41	5	0.8	E	E	E	0.3181	0.0043	0.0027	+	0.2130	0.0165	0.0102	+
42	5	0.8	E	E	D	0.4602	0.0020	0.0013	+	0.1741	0.0030	0.0018	+
43	5	0.8	E	D	E	0.3563	0.0045	0.0028	+	0.2203	0.0035	0.0022	+
44	5	0.8	E	D	D	0.3055	0.0012	0.0007	+	0.0509	0.0018	0.0011	+
45	5	0.8	D	E	E	0.3462	0.0060	0.0037	+	0.1122	0.0086	0.0053	+
46	5	0.8	D	E	D	0.5194	0.0025	0.0015	+	0.1408	0.0041	0.0026	+
47	5	0.8	D	D	E	0.2210	0.0050	0.0031	+	0.0691	0.0066	0.0041	+
48	5	0.8	D	D	D	0.2195	0.0023	0.0014	+	0.0928	0.0015	0.0009	+
49	10	0.4	E	E	E	0.5887	0.0028	0.0017	+	0.8534	0.0064	0.0040	+
50	10	0.4	E	E	D	0.4072	0.0032	0.0020	+	0.6379	0.0040	0.0025	+
51	10	0.4	E	D	E	0.5909	0.0063	0.0039	+	0.7750	0.0065	0.0041	+
52	10	0.4	E	D	D	0.3846	0.0039	0.0024	+	0.5904	0.0043	0.0027	+
53	10	0.4	D	E	E	0.4878	0.0051	0.0032	+	0.2107	0.0029	0.0018	+
54	10	0.4	D	E	D	0.3278	0.0045	0.0028	+	0.1980	0.0041	0.0026	+
55	10	0.4	D	D	E	0.5577	0.0038	0.0024	+	0.2531	0.0042	0.0026	+
56	10	0.4	D	D	D	0.2229	0.0031	0.0019	+	0.2097	0.0023	0.0014	+
57	10	0.6	E	E	E	0.6401	0.0027	0.0017	+	0.7784	0.0015	0.0009	+
58	10	0.6	E	E	D	0.4189	0.0016	0.0010	+	0.6020	0.0016	0.0010	+
59	10	0.6	E	D	E	0.7067	0.0023	0.0014	+	0.7924	0.0030	0.0019	+
60	10	0.6	E	D	D	0.3414	0.0017	0.0011	+	0.5122	0.0041	0.0026	+
61	10	0.6	D	E	E	0.6624	0.0147	0.0091	+	0.1550	0.0207	0.0129	+
62	10	0.6	D	E	D	0.2747	0.0032	0.0020	+	0.1136	0.0025	0.0015	+
63	10	0.6	D	D	E	0.5613	0.0155	0.0096	+	0.1118	0.0164	0.0102	+
64	10	0.6	D	D	D	0.1752	0.0028	0.0017	+	0.1437	0.0046	0.0029	+
65	10	0.8	E	E	E	0.5555	0.0036	0.0022	+	0.5863	0.0024	0.0015	+
66	10	0.8	E	E	D	0.5988	0.0074	0.0046	+	0.4941	0.0045	0.0028	+
67	10	0.8	E	D	E	0.4419	0.0072	0.0045	+	0.5714	0.0014	0.0009	+
68	10	0.8	E	D	D	0.5032	0.0025	0.0016	+	0.3339	0.0080	0.0049	+
69	10	0.8	D	E	E	0.6238	0.0096	0.0060	+	0.2833	0.0100	0.0062	+
70	10	0.8	D	E	D	0.4602	0.0039	0.0024	+	0.1389	0.0013	0.0008	+
71	10	0.8	D	D	E	0.4988	0.0121	0.0075	+	0.1965	0.0109	0.0068	+
72	10	0.8	D	D	D	0.1571	0.0115	0.0071	+	0.2703	0.0022	0.0014	+

						EDD			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**
1	2	0.4	E	E	E	0.1668	0.0066	0.0041	+
2	2	0.4	E	E	D	0.1632	0.0073	0.0045	+
3	2	0.4	E	D	E	0.1836	0.0115	0.0071	+
4	2	0.4	E	D	D	0.1575	0.0035	0.0022	+
5	2	0.4	D	E	E	0.1573	0.0033	0.0021	+
6	2	0.4	D	E	D	0.1614	0.0071	0.0044	+
7	2	0.4	D	D	E	0.1618	0.0096	0.0059	+
8	2	0.4	D	D	D	0.0819	0.0016	0.0010	+
9	2	0.6	E	E	E	0.1511	0.0184	0.0114	+
10	2	0.6	E	E	D	0.1270	0.0030	0.0019	+
11	2	0.6	E	D	E	0.2755	0.0059	0.0037	+
12	2	0.6	E	D	D	0.1905	0.0024	0.0015	+
13	2	0.6	D	E	E	0.1295	0.0040	0.0025	+
14	2	0.6	D	E	D	0.1588	0.0019	0.0012	+
15	2	0.6	D	D	E	0.2185	0.0058	0.0036	+
16	2	0.6	D	D	D	0.1051	0.0029	0.0018	+
17	2	0.8	E	E	E	0.1843	0.0183	0.0113	+
18	2	0.8	E	E	D	0.0837	0.0040	0.0025	+
19	2	0.8	E	D	E	0.7143	0.0634	0.0393	+
20	2	0.8	E	D	D	0.2548	0.0027	0.0017	+
21	2	0.8	D	E	E	0.1909	0.0153	0.0095	+
22	2	0.8	D	E	D	0.1021	0.0020	0.0013	+
23	2	0.8	D	D	E	0.4397	0.0323	0.0200	+
24	2	0.8	D	D	D	0.0522	0.0013	0.0008	+
25	5	0.4	E	E	E	0.4286	0.0347	0.0215	+
26	5	0.4	E	E	D	0.3237	0.0029	0.0018	+
27	5	0.4	E	D	E	0.4647	0.0365	0.0226	+
28	5	0.4	E	D	D	0.3316	0.0032	0.0020	+
29	5	0.4	D	E	E	0.3340	0.0323	0.0200	+
30	5	0.4	D	E	D	0.1768	0.0029	0.0018	+
31	5	0.4	D	D	E	0.3149	0.0048	0.0030	+
32	5	0.4	D	D	D	0.1591	0.0131	0.0081	+
33	5	0.6	E	E	E	0.4720	0.0204	0.0126	+
34	5	0.6	E	E	D	0.3571	0.0017	0.0011	+
35	5	0.6	E	D	E	0.5764	0.0219	0.0136	+
36	5	0.6	E	D	D	0.4302	0.0064	0.0040	+
37	5	0.6	D	E	E	0.3893	0.0175	0.0109	+
38	5	0.6	D	E	D	0.1340	0.0042	0.0026	+

39	5	0.6	D	D	E	0.3002	0.0151	0.0093	+
40	5	0.6	D	D	D	0.0742	0.0012	0.0007	+
41	5	0.8	E	E	E	0.7538	0.0146	0.0091	+
42	5	0.8	E	E	D	0.6640	0.0055	0.0034	+
43	5	0.8	E	D	E	0.9061	0.0049	0.0031	+
44	5	0.8	E	D	D	0.6935	0.0035	0.0022	+
45	5	0.8	D	E	E	1.1181	0.0255	0.0158	+
46	5	0.8	D	E	D	0.6168	0.0027	0.0017	+
47	5	0.8	D	D	E	0.7420	0.0134	0.0083	+
48	5	0.8	D	D	D	0.3435	0.0022	0.0013	+
49	10	0.4	E	E	E	0.9670	0.0054	0.0034	+
50	10	0.4	E	E	D	0.7649	0.0054	0.0033	+
51	10	0.4	E	D	E	0.9595	0.0064	0.0039	+
52	10	0.4	E	D	D	0.7912	0.0049	0.0030	+
53	10	0.4	D	E	E	0.3921	0.0054	0.0033	+
54	10	0.4	D	E	D	0.1816	0.0121	0.0075	+
55	10	0.4	D	D	E	0.3580	0.0045	0.0028	+
56	10	0.4	D	D	D	0.3285	0.0030	0.0018	+
57	10	0.6	E	E	E	1.1344	0.0042	0.0026	+
58	10	0.6	E	E	D	1.0825	0.0048	0.0030	+
59	10	0.6	E	D	E	1.2714	0.0099	0.0061	+
60	10	0.6	E	D	D	0.9681	0.0016	0.0010	+
61	10	0.6	D	E	E	1.1374	0.0236	0.0146	+
62	10	0.6	D	E	D	0.2098	0.0064	0.0040	+
63	10	0.6	D	D	E	0.8067	0.0196	0.0121	+
64	10	0.6	D	D	D	0.0498	0.0168	0.0104	+
65	10	0.8	E	E	E	0.9994	0.0057	0.0035	+
66	10	0.8	E	E	D	1.5822	0.0049	0.0030	+
67	10	0.8	E	D	E	2.1553	0.0059	0.0036	+
68	10	0.8	E	D	D	1.8144	0.0067	0.0041	+
69	10	0.8	D	E	E	2.2918	0.0253	0.0157	+
70	10	0.8	D	E	D	1.9365	0.0039	0.0024	+
71	10	0.8	D	D	E	2.0185	0.0193	0.0120	+
72	10	0.8	D	D	D	2.1473	0.0119	0.0074	+

APPENDIX C

PAIRED-T TEST RESULTS FOR CHAPTER V

The short codes used in the tables for the production characteristics and their settings are as follows:

No	Factor/ Codes	Settings/Codes		
1	Number of Products - PN	2		
		5		
		10		
2	Product-mix - PM	Equal - E	2 products	(0.5, 0.5)
			5 products	(0.2, 0.2, 0.2, 0.2, 0.2)
			10 products	(0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1)
		Different - D	2 products	(0.7, 0.3)
			5 products	(0.35, 0.35, 0.1, 0.1, 0.1)
			10 products	(0.30, 0.30, 0.05, 0.05, 0.05,0.05,0.05, 0.05, 0.05, 0.05)
3	Batch Processor Capacity by Products - C	Equal - E	2 products	(5, 5)
			5 products	(5, 5, 5, 5, 5)
			10 products	(5, 5, 5, 5, 5, 5, 5, 5, 5, 5)
		Different - D	2 products	(7,2)
			5 products	(7, 6, 5, 4, 3)
			10 products	(7, 6, 6, 5, 5, 5, 5, 4, 4, 3)
4	Batch Processing Time By Product - PT	Equal - E	2 products	(25, 25)
			5 products	(25, 25, 25, 25, 25)
			10 products	(25, 25, 25, 25, 25, 25, 25, 25, 25, 25)
		Different - D	2 products	(40, 10)
			5 products	(40, 30, 25, 20, 10)
			10 products	(40, 35, 35, 30, 25, 25, 20, 15, 15, 10)
5	Traffic Intensity - TI	0.5		
		0.8		

Δ = The mean of the weighted normalized waiting time and tardiness differences obtained by 10 replications of NACH-T and the benchmark strategy that the column belongs to. For example, if (*X1*,*X2)*~No-idle and (*Y1*,*Y2)*~NACH-II then,

$$\Delta = \frac{1}{10}\sum_{i=1}^{10}((w_1X1_i + w_2X2_i) - (w_1Y1_i + w_2Y2_i))$$

σ = The standard deviation of the replication differences

conf = The half-width of the 95% confidence interval of the replication differences

sign = + if there is a significant performance difference observed with NACH-II
- if there is not a significant performance difference observed with NACH-II

a) Case: w_1= 0.9, w_2 = 0.1

						Full batch				MBS			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.9014	0.0063	0.0039	+	0.0703	0.0022	0.0014	+
2	2	0.5	E	E	D	1.0147	0.0055	0.0034	+	0.1598	0.0027	0.0017	+
3	2	0.5	E	D	E	1.6565	0.0145	0.0090	+	0.2351	0.0085	0.0053	+
4	2	0.5	E	D	D	0.8132	0.0078	0.0048	+	0.3686	0.0283	0.0176	+
5	2	0.5	D	E	E	0.9145	0.0109	0.0067	+	0.0751	0.0023	0.0014	+
6	2	0.5	D	E	D	1.0941	0.0073	0.0045	+	0.1532	0.0021	0.0013	+
7	2	0.5	D	D	E	1.0611	0.0092	0.0057	+	0.1770	0.0117	0.0073	+
8	2	0.5	D	D	D	0.6484	0.0037	0.0023	+	0.1926	0.0203	0.0126	+
9	2	0.8	E	E	E	0.2566	0.0043	0.0027	+	0.0280	0.0019	0.0012	+
10	2	0.8	E	E	D	0.3637	0.0106	0.0066	+	0.0627	0.0092	0.0057	+
11	2	0.8	E	D	E	0.2334	0.0188	0.0116	+	0.2091	0.0180	0.0112	+
12	2	0.8	E	D	D	0.1556	0.0059	0.0037	+	0.1556	0.0059	0.0037	+
13	2	0.8	D	E	E	0.2618	0.0040	0.0025	+	0.0269	0.0018	0.0011	+
14	2	0.8	D	E	D	0.4060	0.0108	0.0067	+	0.0945	0.0080	0.0049	+
15	2	0.8	D	D	E	0.0590	0.0128	0.0080	+	0.0590	0.0128	0.0080	+
16	2	0.8	D	D	D	0.1169	0.0042	0.0026	+	0.1169	0.0042	0.0026	+
17	5	0.5	E	E	E	2.4532	0.0171	0.0106	+	0.0393	0.0028	0.0017	+
18	5	0.5	E	E	D	2.4816	0.0153	0.0095	+	0.0180	0.0068	0.0042	+
19	5	0.5	E	D	E	2.8330	0.0179	0.0111	+	0.1143	0.0056	0.0035	+
20	5	0.5	E	D	D	2.2987	0.0181	0.0112	+	0.0359	0.0081	0.0050	+
21	5	0.5	D	E	E	2.5313	0.0195	0.0121	+	0.0903	0.0074	0.0046	+
22	5	0.5	D	E	D	2.7403	0.0141	0.0088	+	0.0578	0.0042	0.0026	+
23	5	0.5	D	D	E	2.2039	0.0187	0.0116	+	0.0581	0.0077	0.0048	+
24	5	0.5	D	D	D	2.0155	0.0157	0.0097	+	0.0497	0.0025	0.0015	+
25	5	0.8	E	E	E	0.8660	0.0056	0.0035	+	0.0460	0.0028	0.0017	+
26	5	0.8	E	E	D	0.8937	0.0062	0.0039	+	0.0127	0.0052	0.0032	+
27	5	0.8	E	D	E	0.8455	0.0201	0.0125	+	0.3926	0.0214	0.0133	+
28	5	0.8	E	D	D	0.8934	0.0060	0.0037	+	0.1714	0.0262	0.0163	+
29	5	0.8	D	E	E	0.9418	0.0119	0.0073	+	0.0666	0.0038	0.0024	+
30	5	0.8	D	E	D	1.0048	0.0099	0.0061	+	0.0290	0.0057	0.0035	+
31	5	0.8	D	D	E	0.5604	0.0107	0.0066	+	0.3611	0.0127	0.0079	+
32	5	0.8	D	D	D	0.6816	0.0085	0.0053	+	0.1512	0.0049	0.0030	+
33	10	0.5	E	E	E	5.0429	0.0345	0.0214	+	0.1963	0.0119	0.0074	+
34	10	0.5	E	E	D	5.1145	0.0268	0.0166	+	0.0604	0.0094	0.0058	+
35	10	0.5	E	D	E	5.5061	0.0412	0.0255	+	0.1615	0.0092	0.0057	+

36	10	0.5	E	D	D	4.7621	0.0149	0.0092	+	0.0401	0.0087	0.0054	+
37	10	0.5	D	E	E	5.3390	0.0218	0.0135	+	0.1373	0.0190	0.0118	+
38	10	0.5	D	E	D	5.8493	0.0188	0.0117	+	0.1154	0.0114	0.0071	+
39	10	0.5	D	D	E	4.5389	0.0249	0.0154	+	0.1669	0.0175	0.0109	+
40	10	0.5	D	D	D	4.3828	0.0250	0.0155	+	0.0946	0.0128	0.0080	+
41	10	0.8	E	E	E	1.8037	0.0216	0.0134	+	0.1028	0.0096	0.0059	+
42	10	0.8	E	E	D	1.8632	0.0390	0.0242	+	0.1534	0.0224	0.0139	+
43	10	0.8	E	D	E	1.6889	0.0339	0.0210	+	0.6554	0.0507	0.0314	+
44	10	0.8	E	D	D	1.7763	0.0159	0.0099	+	0.1372	0.0118	0.0073	+
45	10	0.8	D	E	E	1.9990	0.0197	0.0122	+	0.0988	0.0192	0.0119	+
46	10	0.8	D	E	D	2.2149	0.0238	0.0148	+	0.0463	0.0079	0.0049	+
47	10	0.8	D	D	E	1.4202	0.0145	0.0090	+	0.3823	0.0178	0.0110	+
48	10	0.8	D	D	D	1.5036	0.0176	0.0109	+	0.1139	0.0076	0.0047	+

						No-idle			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.0703	0.0022	0.0014	+
2	2	0.5	E	E	D	0.1763	0.0028	0.0017	+
3	2	0.5	E	D	E	0.2351	0.0085	0.0053	+
4	2	0.5	E	D	D	0.2919	0.0036	0.0022	+
5	2	0.5	D	E	E	0.0751	0.0023	0.0014	+
6	2	0.5	D	E	D	0.1956	0.0022	0.0014	+
7	2	0.5	D	D	E	0.2344	0.0121	0.0075	+
8	2	0.5	D	D	D	0.2291	0.0042	0.0026	+
9	2	0.8	E	E	E	0.0280	0.0019	0.0012	+
10	2	0.8	E	E	D	0.1249	0.0044	0.0027	+
11	2	0.8	E	D	E	1.0560	0.0195	0.0121	+
12	2	0.8	E	D	D	0.9936	0.0111	0.0069	+
13	2	0.8	D	E	E	0.0269	0.0018	0.0011	+
14	2	0.8	D	E	D	0.1260	0.0045	0.0028	+
15	2	0.8	D	D	E	0.9345	0.0168	0.0104	+
16	2	0.8	D	D	D	1.0320	0.0053	0.0033	+
17	5	0.5	E	E	E	0.0906	0.0034	0.0021	+
18	5	0.5	E	E	D	0.0180	0.0068	0.0042	+
19	5	0.5	E	D	E	0.1143	0.0056	0.0035	+
20	5	0.5	E	D	D	0.0551	0.0038	0.0024	+
21	5	0.5	D	E	E	0.0903	0.0074	0.0046	+
22	5	0.5	D	E	D	0.0578	0.0042	0.0026	+

23	5	0.5	D	D	E	0.0854	0.0075	0.0047	+
24	5	0.5	D	D	D	0.0497	0.0025	0.0015	+
25	5	0.8	E	E	E	0.0460	0.0028	0.0017	+
26	5	0.8	E	E	D	0.0127	0.0052	0.0032	+
27	5	0.8	E	D	E	1.4515	0.0198	0.0123	+
28	5	0.8	E	D	D	0.2324	0.0040	0.0025	+
29	5	0.8	D	E	E	0.0666	0.0038	0.0024	+
30	5	0.8	D	E	D	0.0307	0.0036	0.0022	+
31	5	0.8	D	D	E	1.1944	0.0107	0.0066	+
32	5	0.8	D	D	D	0.1512	0.0049	0.0030	+
33	10	0.5	E	E	E	0.1963	0.0119	0.0074	+
34	10	0.5	E	E	D	0.0604	0.0094	0.0058	+
35	10	0.5	E	D	E	0.2107	0.0097	0.0060	+
36	10	0.5	E	D	D	0.0401	0.0087	0.0054	+
37	10	0.5	D	E	E	0.1789	0.0192	0.0119	+
38	10	0.5	D	E	D	0.1154	0.0114	0.0071	+
39	10	0.5	D	D	E	0.2123	0.0184	0.0114	+
40	10	0.5	D	D	D	0.0946	0.0128	0.0080	+
41	10	0.8	E	E	E	0.1424	0.0093	0.0058	+
42	10	0.8	E	E	D	0.1534	0.0224	0.0139	+
43	10	0.8	E	D	E	3.4343	0.0043	0.0027	+
44	10	0.8	E	D	D	0.1842	0.0058	0.0036	+
45	10	0.8	D	E	E	0.1208	0.0180	0.0112	+
46	10	0.8	D	E	D	0.0594	0.0055	0.0034	+
47	10	0.8	D	D	E	3.2376	0.0137	0.0085	+
48	10	0.8	D	D	D	0.1305	0.0068	0.0042	+

b) Case: w_1= 0.5, w_2 = 0.5

						Full batch				MBS			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.7493	0.0074	0.0046	+	0.0482	0.0023	0.0014	+
2	2	0.5	E	E	D	0.8810	0.0082	0.0051	+	0.1243	0.0060	0.0037	+
3	2	0.5	E	D	E	1.5203	0.0196	0.0121	+	0.1872	0.0074	0.0046	+
4	2	0.5	E	D	D	0.6146	0.0160	0.0099	+	0.3393	0.0218	0.0135	+
5	2	0.5	D	E	E	0.7756	0.0098	0.0061	+	0.0517	0.0022	0.0013	+
6	2	0.5	D	E	D	0.9658	0.0077	0.0048	+	0.1348	0.0039	0.0024	+
7	2	0.5	D	D	E	0.9165	0.0061	0.0038	+	0.1450	0.0119	0.0073	+
8	2	0.5	D	D	D	0.4735	0.0056	0.0035	+	0.1574	0.0121	0.0075	+
9	2	0.8	E	E	E	0.1947	0.0032	0.0020	+	0.0311	0.0058	0.0036	+
10	2	0.8	E	E	D	0.3210	0.0034	0.0021	+	0.0633	0.0080	0.0050	+
11	2	0.8	E	D	E	0.2325	0.0170	0.0105	+	0.2325	0.0170	0.0105	+
12	2	0.8	E	D	D	0.0750	0.0050	0.0031	+	0.0750	0.0050	0.0031	+
13	2	0.8	D	E	E	0.2081	0.0046	0.0028	+	0.0170	0.0021	0.0013	+
14	2	0.8	D	E	D	0.3630	0.0052	0.0032	+	0.0789	0.0053	0.0033	+
15	2	0.8	D	D	E	0.0074	0.0119	0.0074	-	0.0074	0.0119	0.0074	-
16	2	0.8	D	D	D	0.0527	0.0031	0.0020	+	0.0527	0.0031	0.0020	+
17	5	0.5	E	E	E	2.3022	0.0159	0.0099	+	0.0437	0.0035	0.0021	+
18	5	0.5	E	E	D	2.3246	0.0126	0.0078	+	0.0440	0.0035	0.0022	+
19	5	0.5	E	D	E	2.6996	0.0179	0.0111	+	0.0883	0.0060	0.0037	+
20	5	0.5	E	D	D	2.1222	0.0218	0.0135	+	0.0412	0.0045	0.0028	+
21	5	0.5	D	E	E	2.3885	0.0200	0.0124	+	0.0699	0.0079	0.0049	+
22	5	0.5	D	E	D	2.5636	0.0167	0.0104	+	0.0713	0.0068	0.0042	+
23	5	0.5	D	D	E	2.0461	0.0161	0.0100	+	0.0695	0.0076	0.0047	+
24	5	0.5	D	D	D	1.8098	0.0136	0.0085	+	0.0716	0.0039	0.0024	+
25	5	0.8	E	E	E	0.7669	0.0055	0.0034	+	0.0348	0.0032	0.0020	+
26	5	0.8	E	E	D	0.7663	0.0079	0.0049	+	0.0018	0.0056	0.0035	-
27	5	0.8	E	D	E	0.7723	0.0149	0.0093	+	0.4083	0.0150	0.0093	+
28	5	0.8	E	D	D	0.7538	0.0063	0.0039	+	0.1683	0.0276	0.0171	+
29	5	0.8	D	E	E	0.8265	0.0101	0.0063	+	0.0311	0.0041	0.0026	+
30	5	0.8	D	E	D	0.9289	0.0066	0.0041	+	0.0299	0.0041	0.0026	+
31	5	0.8	D	D	E	0.4876	0.0078	0.0048	+	0.2968	0.0082	0.0051	+
32	5	0.8	D	D	D	0.4815	0.0069	0.0043	+	0.2233	0.0117	0.0072	+
33	10	0.5	E	E	E	4.9409	0.0350	0.0217	+	0.1714	0.0107	0.0066	+
34	10	0.5	E	E	D	4.9390	0.0270	0.0167	+	0.0509	0.0104	0.0064	+
35	10	0.5	E	D	E	5.3982	0.0412	0.0256	+	0.1217	0.0116	0.0072	+

36	10	0.5	E	D	D	4.6143	0.0303	0.0188	+	0.0529	0.0070	0.0043	+
37	10	0.5	D	E	E	5.2205	0.0520	0.0322	+	0.1091	0.0184	0.0114	+
38	10	0.5	D	E	D	5.5738	0.0411	0.0254	+	0.0728	0.0096	0.0059	+
39	10	0.5	D	D	E	4.4056	0.0237	0.0147	+	0.1504	0.0137	0.0085	+
40	10	0.5	D	D	D	4.0979	0.0243	0.0150	+	0.0883	0.0096	0.0059	+
41	10	0.8	E	E	E	1.7267	0.0199	0.0123	+	0.0924	0.0107	0.0066	+
42	10	0.8	E	E	D	1.6448	0.0183	0.0114	+	0.0167	0.0067	0.0042	+
43	10	0.8	E	D	E	1.6707	0.0183	0.0113	+	0.6285	0.0275	0.0171	+
44	10	0.8	E	D	D	1.6002	0.0104	0.0065	+	0.0994	0.0078	0.0048	+
45	10	0.8	D	E	E	1.9185	0.0150	0.0093	+	0.0888	0.0153	0.0095	+
46	10	0.8	D	E	D	2.0908	0.0186	0.0115	+	0.0575	0.0064	0.0040	+
47	10	0.8	D	D	E	1.2300	0.0169	0.0105	+	0.2922	0.0160	0.0099	+
48	10	0.8	D	D	D	1.3735	0.0180	0.0112	+	0.1746	0.0113	0.0070	+

						No-idle			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.0482	0.0023	0.0014	+
2	2	0.5	E	E	D	0.1609	0.0064	0.0040	+
3	2	0.5	E	D	E	0.1872	0.0074	0.0046	+
4	2	0.5	E	D	D	0.5446	0.0235	0.0145	+
5	2	0.5	D	E	E	0.0517	0.0022	0.0013	+
6	2	0.5	D	E	D	0.1704	0.0028	0.0017	+
7	2	0.5	D	D	E	0.2007	0.0117	0.0072	+
8	2	0.5	D	D	D	0.1813	0.0052	0.0032	+
9	2	0.8	E	E	E	0.0311	0.0058	0.0036	+
10	2	0.8	E	E	D	0.0940	0.0072	0.0045	+
11	2	0.8	E	D	E	1.0545	0.0173	0.0107	+
12	2	0.8	E	D	D	0.9986	0.0188	0.0117	+
13	2	0.8	D	E	E	0.0230	0.0022	0.0014	+
14	2	0.8	D	E	D	0.1102	0.0031	0.0019	+
15	2	0.8	D	D	E	0.8270	0.0129	0.0080	+
16	2	0.8	D	D	D	0.8241	0.0073	0.0045	+
17	5	0.5	E	E	E	0.0691	0.0031	0.0019	+
18	5	0.5	E	E	D	0.0440	0.0035	0.0022	+
19	5	0.5	E	D	E	0.0883	0.0060	0.0037	+
20	5	0.5	E	D	D	0.0516	0.0042	0.0026	+
21	5	0.5	D	E	E	0.0699	0.0079	0.0049	+
22	5	0.5	D	E	D	0.0713	0.0068	0.0042	+

23	5	0.5	D	D	E	0.0850	0.0074	0.0046	+
24	5	0.5	D	D	D	0.0716	0.0039	0.0024	+
25	5	0.8	E	E	E	0.0348	0.0032	0.0020	+
26	5	0.8	E	E	D	0.0018	0.0056	0.0035	-
27	5	0.8	E	D	E	1.4118	0.0145	0.0090	+
28	5	0.8	E	D	D	0.2737	0.0190	0.0118	+
29	5	0.8	D	E	E	0.0311	0.0041	0.0026	+
30	5	0.8	D	E	D	0.0402	0.0041	0.0025	+
31	5	0.8	D	D	E	1.1939	0.0078	0.0048	+
32	5	0.8	D	D	D	0.2233	0.0117	0.0072	+
33	10	0.5	E	E	E	0.1714	0.0107	0.0066	+
34	10	0.5	E	E	D	0.0509	0.0104	0.0064	+
35	10	0.5	E	D	E	0.1668	0.0118	0.0073	+
36	10	0.5	E	D	D	0.0529	0.0070	0.0043	+
37	10	0.5	D	E	E	0.1545	0.0184	0.0114	+
38	10	0.5	D	E	D	0.0728	0.0096	0.0059	+
39	10	0.5	D	D	E	0.1896	0.0151	0.0093	+
40	10	0.5	D	D	D	0.0883	0.0096	0.0059	+
41	10	0.8	E	E	E	0.1281	0.0103	0.0064	+
42	10	0.8	E	E	D	0.0167	0.0067	0.0042	+
43	10	0.8	E	D	E	3.4186	0.0215	0.0134	+
44	10	0.8	E	D	D	0.1397	0.0068	0.0042	+
45	10	0.8	D	E	E	0.1134	0.0170	0.0105	+
46	10	0.8	D	E	D	0.0722	0.0064	0.0039	+
47	10	0.8	D	D	E	3.1794	0.0185	0.0114	+
48	10	0.8	D	D	D	0.1410	0.0125	0.0077	+

c) Case: w_1= 0.1, w_2 = 0.9

						Full batch				MBS			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.6194	0.0070	0.0043	+	0.0354	0.0016	0.0010	+
2	2	0.5	E	E	D	0.8121	0.0097	0.0060	+	0.1175	0.0060	0.0037	+
3	2	0.5	E	D	E	1.4585	0.0257	0.0159	+	0.1971	0.0051	0.0032	+
4	2	0.5	E	D	D	0.4941	0.0214	0.0133	+	0.2984	0.0146	0.0091	+
5	2	0.5	D	E	E	0.6695	0.0108	0.0067	+	0.0478	0.0019	0.0012	+
6	2	0.5	D	E	D	0.8955	0.0101	0.0063	+	0.1114	0.0066	0.0041	+
7	2	0.5	D	D	E	0.8143	0.0085	0.0053	+	0.1121	0.0094	0.0058	+
8	2	0.5	D	D	D	0.3452	0.0084	0.0052	+	0.0875	0.0110	0.0068	+
9	2	0.8	E	E	E	0.1450	0.0024	0.0015	+	0.0041	0.0013	0.0008	+
10	2	0.8	E	E	D	0.3236	0.0085	0.0053	+	0.0764	0.0025	0.0015	+
11	2	0.8	E	D	E	0.2634	0.0204	0.0126	+	0.2388	0.0202	0.0125	+
12	2	0.8	E	D	D	0.0726	0.0054	0.0034	+	0.0455	0.0051	0.0031	+
13	2	0.8	D	E	E	0.1589	0.0059	0.0037	+	0.0020	0.0018	0.0011	+
14	2	0.8	D	E	D	0.3625	0.0054	0.0034	+	0.0799	0.0060	0.0037	+
15	2	0.8	D	D	E	0.0059	0.0125	0.0077	-	0.0059	0.0125	0.0077	-
16	2	0.8	D	D	D	0.0440	0.0066	0.0041	+	0.0440	0.0066	0.0041	+
17	5	0.5	E	E	E	2.1593	0.0164	0.0101	+	0.0336	0.0051	0.0032	+
18	5	0.5	E	E	D	2.1832	0.0175	0.0109	+	0.0264	0.0047	0.0029	+
19	5	0.5	E	D	E	2.5676	0.0174	0.0108	+	0.0573	0.0073	0.0045	+
20	5	0.5	E	D	D	1.9546	0.0117	0.0073	+	0.0395	0.0051	0.0032	+
21	5	0.5	D	E	E	2.2491	0.0211	0.0131	+	0.0448	0.0069	0.0043	+
22	5	0.5	D	E	D	2.4478	0.0148	0.0092	+	0.0621	0.0032	0.0020	+
23	5	0.5	D	D	E	1.8913	0.0167	0.0104	+	0.0499	0.0076	0.0047	+
24	5	0.5	D	D	D	1.6568	0.0159	0.0099	+	0.0645	0.0051	0.0031	+
25	5	0.8	E	E	E	0.6705	0.0050	0.0031	+	0.0245	0.0022	0.0014	+
26	5	0.8	E	E	D	0.6951	0.0050	0.0031	+	0.0024	0.0047	0.0029	-
27	5	0.8	E	D	E	0.7450	0.0148	0.0092	+	0.3796	0.0202	0.0125	+
28	5	0.8	E	D	D	0.6994	0.0081	0.0050	+	0.1520	0.0141	0.0087	+
29	5	0.8	D	E	E	0.7395	0.0102	0.0063	+	0.0212	0.0018	0.0011	+
30	5	0.8	D	E	D	0.8739	0.0071	0.0044	+	0.0279	0.0068	0.0042	+
31	5	0.8	D	D	E	0.4243	0.0092	0.0057	+	0.2429	0.0097	0.0060	+
32	5	0.8	D	D	D	0.5486	0.0082	0.0051	+	0.2384	0.0093	0.0058	+
33	10	0.5	E	E	E	4.8273	0.0322	0.0200	+	0.1346	0.0070	0.0043	+
34	10	0.5	E	E	D	4.8204	0.0260	0.0161	+	0.0532	0.0098	0.0061	+
35	10	0.5	E	D	E	5.3248	0.0159	0.0099	+	0.1086	0.0095	0.0059	+

36	10	0.5	E	D	D	4.4763	0.0162	0.0100	+	0.0467	0.0053	0.0033	+
37	10	0.5	D	E	E	5.1101	0.0181	0.0112	+	0.0890	0.0180	0.0112	+
38	10	0.5	D	E	D	5.4664	0.0194	0.0120	+	0.0904	0.0116	0.0072	+
39	10	0.5	D	D	E	4.2731	0.0284	0.0176	+	0.1347	0.0224	0.0139	+
40	10	0.5	D	D	D	3.9688	0.0249	0.0154	+	0.0893	0.0113	0.0070	+
41	10	0.8	E	E	E	1.6564	0.0176	0.0109	+	0.0885	0.0073	0.0046	+
42	10	0.8	E	E	D	1.5985	0.0219	0.0136	+	0.0400	0.0047	0.0029	+
43	10	0.8	E	D	E	1.6476	0.0058	0.0036	+	0.5967	0.0081	0.0050	+
44	10	0.8	E	D	D	1.6056	0.0088	0.0054	+	0.1417	0.0067	0.0042	+
45	10	0.8	D	E	E	1.8340	0.0153	0.0095	+	0.0750	0.0136	0.0084	+
46	10	0.8	D	E	D	2.1292	0.0173	0.0107	+	0.1352	0.0078	0.0048	+
47	10	0.8	D	D	E	1.1592	0.0196	0.0121	+	0.1918	0.0248	0.0154	+
48	10	0.8	D	D	D	1.3365	0.0222	0.0138	+	0.1419	0.0130	0.0080	+

						No-idle			
index	**PN**	**TI**	**PM**	**C**	**PT**	**Δ**	**σ**	**conf**	**sign**
1	2	0.5	E	E	E	0.0354	0.0016	0.0010	+
2	2	0.5	E	E	D	0.1759	0.0053	0.0033	+
3	2	0.5	E	D	E	0.1971	0.0051	0.0032	+
4	2	0.5	E	D	D	0.5588	0.0192	0.0119	+
5	2	0.5	D	E	E	0.0478	0.0019	0.0012	+
6	2	0.5	D	E	D	0.1702	0.0044	0.0028	+
7	2	0.5	D	D	E	0.1868	0.0092	0.0057	+
8	2	0.5	D	D	D	0.1194	0.0101	0.0063	+
9	2	0.8	E	E	E	0.0041	0.0013	0.0008	+
10	2	0.8	E	E	D	0.0972	0.0022	0.0014	+
11	2	0.8	E	D	E	0.9791	0.0207	0.0128	+
12	2	0.8	E	D	D	1.0071	0.0240	0.0149	+
13	2	0.8	D	E	E	0.0037	0.0017	0.0010	+
14	2	0.8	D	E	D	0.1009	0.0023	0.0014	+
15	2	0.8	D	D	E	0.7084	0.0122	0.0075	+
16	2	0.8	D	D	D	0.5790	0.0168	0.0104	+
17	5	0.5	E	E	E	0.0505	0.0042	0.0026	+
18	5	0.5	E	E	D	0.0264	0.0047	0.0029	+
19	5	0.5	E	D	E	0.0573	0.0073	0.0045	+
20	5	0.5	E	D	D	0.0447	0.0026	0.0016	+
21	5	0.5	D	E	E	0.0448	0.0069	0.0043	+
22	5	0.5	D	E	D	0.0621	0.0032	0.0020	+

23	5	0.5	D	D	E	0.0621	0.0070	0.0043	+
24	5	0.5	D	D	D	0.0645	0.0051	0.0031	+
25	5	0.8	E	E	E	0.0245	0.0022	0.0014	+
26	5	0.8	E	E	D	0.0024	0.0047	0.0029	-
27	5	0.8	E	D	E	1.3700	0.0265	0.0164	+
28	5	0.8	E	D	D	0.2665	0.0137	0.0085	+
29	5	0.8	D	E	E	0.0212	0.0018	0.0011	+
30	5	0.8	D	E	D	0.0349	0.0060	0.0037	+
31	5	0.8	D	D	E	1.1872	0.0092	0.0057	+
32	5	0.8	D	D	D	0.2384	0.0093	0.0058	+
33	10	0.5	E	E	E	0.1346	0.0070	0.0043	+
34	10	0.5	E	E	D	0.0532	0.0098	0.0061	+
35	10	0.5	E	D	E	0.1573	0.0105	0.0065	+
36	10	0.5	E	D	D	0.0467	0.0053	0.0033	+
37	10	0.5	D	E	E	0.1390	0.0214	0.0133	+
38	10	0.5	D	E	D	0.0904	0.0116	0.0072	+
39	10	0.5	D	D	E	0.1671	0.0229	0.0142	+
40	10	0.5	D	D	D	0.0893	0.0113	0.0070	+
41	10	0.8	E	E	E	0.1204	0.0070	0.0043	+
42	10	0.8	E	E	D	0.0400	0.0047	0.0029	+
43	10	0.8	E	D	E	3.3981	0.0184	0.0114	+
44	10	0.8	E	D	D	0.1888	0.0117	0.0072	+
45	10	0.8	D	E	E	0.1024	0.0131	0.0081	+
46	10	0.8	D	E	D	0.1473	0.0088	0.0054	+
47	10	0.8	D	D	E	3.0822	0.0183	0.0114	+
48	10	0.8	D	D	D	0.1301	0.0124	0.0077	+

VITA

Abdullah Cerekci received his Bachelor of Science degree in industrial engineering from Bilkent University in 2002. He worked for Ford Motor Company between 2002 and 2003 as a production planning specialist. He entered the Industrial and Systems Engineering Ph.D. program in September 2003. His research interests include simulation modeling, heuristic optimization and strategic control of complex manufacturing systems. His permanent address is 241 Zachry, 3131 TAMU, College Station, TX, 77843.