2016

Due date Quotation and Delivery Schedule In Dual Channel Supply chain

Nooshin Nekoiemehr
University of Windsor

Due Date Quotation and Delivery Schedule in Dual Channel Supply Chain

by

Nooshin Nekoiemehr

A Dissertation

Submitted to the Faculty of Graduate Studies
through Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

2016

Due Date Quotation and Delivery Schedule
in Dual Channel Supply Chain

by

Nooshin Nekoiemehr

APPROVED BY:

____
Dr. L. Fang, External Examiner
Ryerson University

____
Dr. G. Bhandari
Odette School of Business

____
Dr. M. Wang
Department of Mechanical, Automotive & Materials Engineering

____
Dr. A. Alfakih
Department of Mathematics and Statistics

____
Dr. G. Zhang, Advisor
Department of Mechanical, Automotive & Materials Engineering

____
Dr. E. Selvarajah, Advisor
Odette School of Business

14 March 2016

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship Declaration

I hereby declare that this thesis does not incorporate material that is result of joint research. In all cases, the key idea, primary contributions, experimental designs, data analysis and interpretation, were performed by the author and Dr. Guoqing Zhang and Dr Esaignani Selvarajah as the advisors.

I certify that, with the above qualifications, this thesis, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This thesis includes two original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
Chapter 2	Nekoiemehr, N., Selavarajah, E., Zhang, G., Scheduling of jobs with cross families in two stage manufacturing systems, International Journal of Production Economics.	Published
Chapter 3	Nekoiemehr, N., Zhang, G., Selavarajah, E., Due Date Quotation in dual Channel supply chain, Manufacturing & Service Operations Management.	Submitted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

With the growth of e-business, many companies are trying to implement online (e-tail) channel besides their traditional retail stores to provide more convenient access of products for the customers and mainly enhance their customer service. These businesses are the main entities of a network called dual channel supply chain. Improving customer service as one of the main performance measures has got a growing interest in recent years from all entities of supply chain specifically manufacturing/service providers. In this context, we can express customer service as "being able to satisfy customer demands as soon as possible" and from the manufacturer's point of view, it is potentially achievable by coordination of scheduling and reliable due date quotation. In this dissertation, we consider due date quotation problem coordinated with scheduling in a two-echelon dual channel supply chain from the manufacturer point of view. We study three main problems.

We first study a delivery scheduling problem where the manufacturer has to decide the earliest delivery time for the orders received from retail channel. In this problem, a two-echelon supply chain is considered where a retailer places bulk orders of the same product with different families to the manufacturer. Since the manufacture accepts only bulk orders, no online order is assumed for this problem. The analysis with no online customers is relatively easy and therefore, we consider families of products in this problem. For this problem, we consider only retail channel with deterministic demand and cross family setup time which was motivated by an application from the automotive industry.

In the second problem, we have a multi-processor manufacturing system receiving orders from both e-tail and retail channels. Online orders arrive over time, and as they arrive, the manufacturer will decide to accept or reject the orders and quote due dates to the accepted ones. Accepted online orders should be delivered to the customers before the quoted due dates via one of the two available options: directly by the manufacturer or through the retail store. Our goal is to quote due dates to the online orders and schedule them to maximize the total profit while satisfying the maximum acceptable lead time for online orders and distinct production capacity for each channel.

The third problem is an extension of the second case; due date quotation coordinated with scheduling in dual channel supply chain, with primarily similar assumptions. However in this case, we assume that the production capacity of the manufacturer is shared among orders of both channels.

This dissertation provides methodologies, insights, algorithms, competitive analysis and computational results for these three problems.

DEDICATION

To my parents who smoothened the path of my dreams and unconditionally supported me through all my walks of life.

To my husband without whose caring support it would have been much more difficult.

To my brother and his wife who believed in me and were always there for me.

ACKNOWLEDGEMENTS

I express my sincere gratitude to my advisors, Dr. Guoqing Zhang and Dr. Esaignani Selvarajah for their guidance, encouragement, and support during my Ph.D. study at the University of Windsor. I am indebted for their support and advice for my professional and personal development.

I also would like to thank other members of my committee, Dr. Michael Wang, Dr. Abdo Alfakih, and Dr. Gokul Bhandari for their encouragement throughout my doctoral program especially for their professional comments and advice that provide me great help in the preparation of this dissertation. In addition, I would like to thank Dr. Ben Chaouch for allocating valuable time as my comprehensive exam's advisory committee.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Dual channel supply chain

A supply chain encompasses all the activities and facilities integrated with producing and delivering products or services from the supplier to the end customer. Supply chain management objective is to deliver maximum value to the end customer which is possible by integration and collaboration of all the activities and processes along the chain, however it also depends on the analysis perspective. With the growth of e-business, many companies are trying to adopt online (e-tail) channel besides their traditional retail stores to provide more convenient access of products for their customers and in fact enhance the customer service. These businesses have a dual channel supply chain or more generally multi-channel supply chain. In a dual channel supply chain, the manufacturer uses both traditional retail store and e-tail channel to distribute its products. Firms following this dual-channel strategy are referred to as click-and-mortar companies, which is distinct from their traditional brick-and-mortar counterparts (Chand and Chhajed 1992). A review of the literature on multi-channel distribution systems reveals the economic advantages of serving customers through different channels. Although dual channel may help companies increase their customer's awareness and shopping choices, this type of distribution model affects all business functions and operational decisions. Hill et al. (2002) introduced four main strategies for click-and-mortar companies.

In the first strategy, firms separate retail and e-tail channels where each channel has its own warehouse, as well as inventory control and pricing features. Some companies find it difficult to manage the same product in two different channels; therefore as the second strategy, they outsource the e-tail channel to the third party and all the order-fulfilment process is managed by the expert third party firm. Drop-ship is another strategy that some companies apply, in which the third party just picks, packs and delivers the orders to customers in e-tail channel while all the distribution information is available for him. The other strategy used recently is called professional shopper strategy, where customers in e-tail channel order online and then pick up the product from the retail store. In this dissertation, we assume that there exist two delivery options for e-tail customers which implies that the manufacturer's strategy is a combination of drop-ship and professional shopper strategies.

Numerous aspects of dual-channel supply chain such as advantages and disadvantages of having e-tail channel besides traditional retail channel, when to establish e-tail channel, pricing policies or price completion, replenishment policies and return policies have been studied broadly in supply chain literature. Investigating dual channel's impact on company's performance is highly integrated with several related fields, mainly as warehouse design, optimal inventory decisions and pricing, and we have various studies in each field in the literature. However, there is no research till date that has discussed due date quotation problem coordinated with scheduling in a dual channel supply chain, the problem which is considered in this dissertation.

Most of e-business failures are related to operational decisions, and one of the main reasons of early e-business failures is ineffective order fulfillment (Tarn et al. 2003). It is accepted that even a well-designed dual channel supply chain is useless when it is not successful to deliver items as promised. If the due dates are set in advance, the manufacturer has to optimize orders schedule and deliver them as soon as possible, but if the due dates are not preset which is true in most situations, the manufacturer has the option of determining delivery time for the accepted orders. Therefore, effective order fulfilment is tightly related to accurate due date quotation in most real-world situations. Accurate due date quotation is considered as one of the main performance measures as well as cost and quality (Handfield et al. 1999, Stalk and Hout 1990), however it is not an easy task; setting relatively soon due dates specifically for make-to-order environments and scheduling the orders to ensure that they meet the quoted due dates (Kaminsky and Hochbaum 2004) specially when we have unknown demand trend. In fact, capacity constraint makes it impossible to set the ideal due dates, so that the challenging part of these problems are the trade-off between sequencing the jobs to meet the due dates and setting due dates while sequencing is possible. Considering high competitive environment in recent years, dual channel supply chain is vastly increasing and one of the most important challenges for these supply chains will be to quote and manage the most efficient due dates to get the competitive advantage in the market.

1.2 Due date management

Most of the studies in the scheduling literature that involve due date, focus on sequencing the jobs in different stations assuming that the due dates are preset (Keskinocak and Tayur 2003). These problems optimize some objectives while satisfying the given due dates. Minimization of tardiness, completion times, number of tardy jobs or lateness are among the common objective functions in these problems. They also may apply different rules for sequencing the orders such as earliest due date, minimum slack time, and critical ratio when orders' due dates are considered as an input and given parameters. However, in most of the real-case situations, specifically in make-to-order situations or the cases dealing with online customers, the manufacturer needs to set the due dates and schedule the jobs to meet these quoted due dates. This practical problem of combined due date quotation and scheduling is known as due date management problem and is considered in this dissertation.

There exist several studies considering due date management problem, i.e., problems containing both elements of due date (lead time) setting and scheduling. The challenging issue in these problems is setting relatively soon due dates, specifically for make-to-order environments, and scheduling the orders to ensure that they meet the quoted due dates, especially when we have unknown demand trend (Kaminsky and Hochbaum 2004). In fact, capacity constraint makes it impossible to set the ideal due dates, so that the thought-provoking part of these problems is the trade-off between sequencing jobs to meet the due dates and setting due date while sequencing is possible (Figure 1.1).

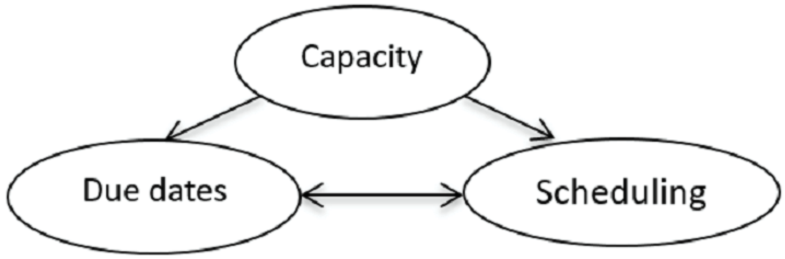

FIGURE 1.1. scheduling combined with due date quotation

In the due date management literature, problems can be generally classified as shown in Figure 1.2. Both problems with dynamic and static demand can be found. In static models all jobs are available at the beginning of the planning horizon, however, in dynamic models jobs have different arrival times. In most of the due date quotation models in the literature there exists no threshold for the quoted due dates. In other words, any due date can be quoted for the customers (Keskinocak and Tayur 2003). Any class of the scheduling problems such as single machine, parallel machine or flow shop models may be considered for both dynamic and static problems in due date management. Studying dynamic models is further divided into online and offline models. In offline problems orders arrival time and other information like processing time are known in advance for the whole planning horizon, and based on them all sequencing and due date quotation decisions are made. It may be the case where the demand trend is highly predictable, or loyal customers are considered who place their orders far ahead. Mathematical programing or different heuristic methods may be used to solve offline models. In these problems, determining due dates and scheduling the jobs to ensure they meet the quoted due dates might be much easier than online problems. In online scheduling the decision about accepting or rejecting the order, quoting the due dates and sequencing is made when the order arrives, using the information of jobs arrived by that time, while there is no information about the future orders. In online problems, mainly unknown demand trend is considered, i.e., orders will arrive over time and once the order is received, the manufacturer becomes aware of the job's information like the processing time. Online models represent many real-world applications when a job's information is not available till it arrives at the system, or there is no information about future arrivals. In this dissertation, online models are considered, where job arrival times are not known in advance.

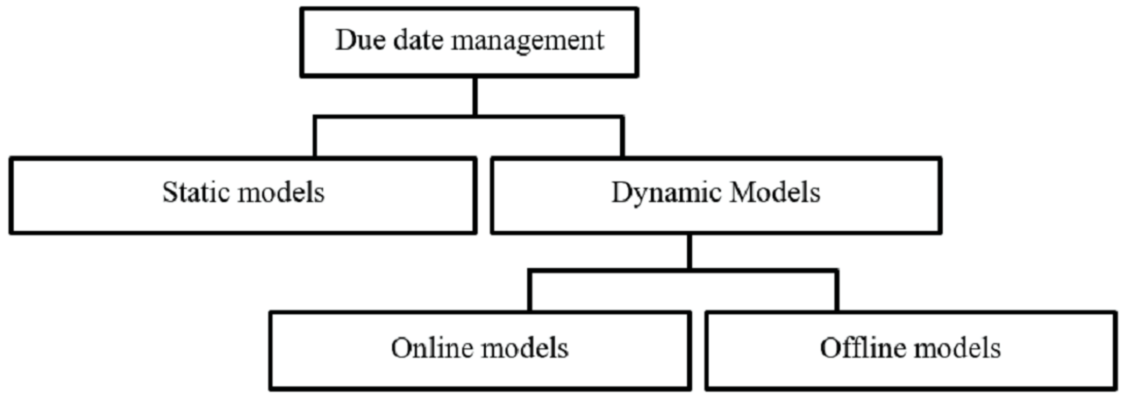

FIGURE 1.2. Due date management problems classification

The most common objectives considered in due date management problems are minimizing average quoted due dates (lead time), minimizing average tardiness or earliness. Mainly, these model's objectives are functions of quoted due dates. There are also several studies maximizing total revenue where the revenue function is also due-date-sensitive and decreases by increasing the quoted lead times (due dates) (Kaminsky and Hochbaum 2004).

Reliability is one of the important features in due date management problems. Some of the models contain 100% reliability constraints, where each order should be completed (delivered) by the quoted due dates. In other words, no tardiness is permitted in these problems. Some others may consider probabilistic reliability constraints, where each job's completion/delivery time may exceed the quoted due dates by specific probabilities. These problems may have restrictions on the number of tardy jobs or total amount of tardiness for all orders.

There exist three versions of online models based on when sequencing or due date quotation decisions have to be made (Keskinocak and Tayur 2003).

1) Traditional online version
2) Quotation online version
3) Delayed quotation version

In the first version, orders arrive over time, there is no information regarding future arrivals and decisions regarding rejecting or accepting the orders as well as sequencing are made based on the information of orders arrived till that time. In these models, decisions can be made any time after an order's arrival time and there is no time limit for decision making. In the second version, making decision about accepting or rejecting the order as well as quoting the due date must be done immediately as the order arrives, while there is no information about the future arrivals. This is the tough version of online models since all decisions must be made immediately and it is even more difficult when quoted due dates should be 100% reliable. In the third version, making decision regarding accepting/rejecting the orders and quoting the due dates can be made within s time units after job's arrival time. In other words, the manufacturer has s units of time to make decision about each order after they arrived at the system.

In all versions of online models, there may be a maximum acceptable lead time (threshold on quoted due dates) which is a realistic assumption. Clearly, the length of acceptable lead time depends on the industry and product type, but in most of the situations, if the customers are offered a due date after their desired lead time, they will not place the order. From the manufacturer's point of view, in fact the manufacturer has the option of rejecting the online order by offering a due date after the desired lead time when there is no benefit in accepting the order.

In this dissertation, we study the problem of 100% reliable and immediate due date quotation for online orders in a dual channel supply chain in order to maximize the total profit, while considering a threshold on quoted due dates.

1.3 Research objectives

The objective of this dissertation is to analyze the problem of due date quotation combined with scheduling (due date management) in a dual channel supply chain. The main goal is to study this problem from the manufacturer's point of view, investigate different production environments and develop proper methodologies for each situation. In the event of this objective, three main sub-problems are considered in this dissertation which are introduced in the following section.

1.4 Problems and solution methodologies

In this section, three main problems studied in this dissertation are introduced following by important approaches/algorithms applied for each problem.

We first study a delivery scheduling problem where the manufacturer has to decide the earliest delivery time for the orders received from retail channel. In this problem, a two-echelon supply chain is considered where a retailer places bulk orders of the same product with different families to the manufacturer. Since the manufacture accepts only bulk orders, no online order is assumed for this problem. The analysis with no online customers is relatively easy and therefore, we consider families of products in this problem. For this problem, we consider only retail channel with deterministic demand and cross family setup time (a novel assumption in literature) which was motivated by an application from the automotive industry.

In a problem with cross family setup, job allocations to families are machine (stage) based. A two-stage manufacturing system is assumed for this problem and therefore, this system can be represented by a two machine permutation flow shop, where each stage is represented by a machine with cross family setups and the objective is to minimize the maximum completion time (makespan) of the jobs in an order. In this problem, job allocations to families are machine based and there are two family classes: family class for machine 1 and family class for machine 2. Each job will have two different family memberships and we call this problem jobs with cross families, i.e., two jobs belonging to two different families in one stage may belong to the same family in another stage. Since the problem is NP-hard for arbitrary number of families, we study the problem when there is a fixed number of families in each stage.

Past studies on scheduling of jobs with family setups have assumed that jobs belonging to the same family in one stage belong to the same family in other stages. However, this assumption may not be applicable for all production cases. For example, consider an automobile manufacturing system where all jobs are processed first in the body shop and then in the paint shop. In the body shop, jobs are categorized into two-door or four-door families and the same jobs are re-categorized in the paint shop based on color requirements. For instance, a two-door job and a four-door job belong to the same family in the paint shop if they both require the same color. To the best of our knowledge in scheduling literature, there is no study on jobs with cross families.

The main approach used to solve this problem is to examine properties of optimal schedules utilized to develop efficient algorithms. For this problem, we first investigated some properties of job sequences and batches in the optimal schedule which led us to develop an efficient branch and bound (B&B) search algorithm for sequencing given ordered batches. In fact an algorithm capable of solving the problem optimally in $O(n^c)$ time is developed. We also developed a hybrid genetic algorithm (HGA) to solve problems with arbitrary number of families.

In the second problem, we have a multi-processor manufacturing system receiving orders from both e-tail and retail channels. Online orders arrive over time, and as they arrive, the manufacturer will quote a due date to the accepted orders. Accepted online orders should be delivered to the customers before the quoted due dates via one of the two available options; directly from the manufacturer or through the retail store (Figure 1.3). Our goal is to quote due dates to online orders and schedule them to maximize the total profit while satisfying the maximum acceptable lead time for online orders and distinct production capacity for orders in each channel.

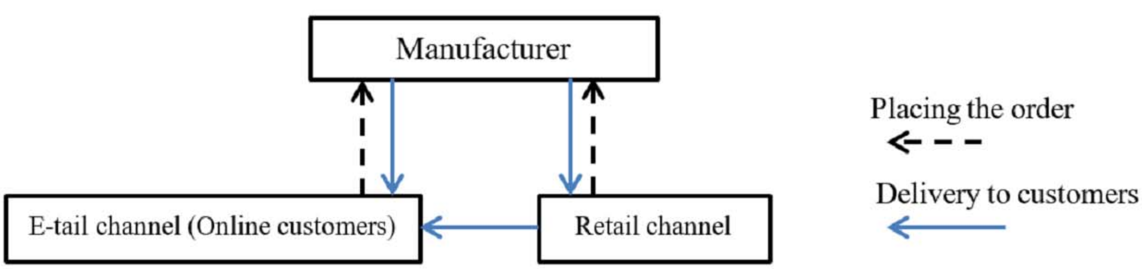

FIGURE 1.3. Due date quotation problem in dual channel

In fact, the second problem we consider is a reliable and immediate due date quotation in a two-echelon dual channel supply chain in order to maximize the total profit which contains a due-date sensitive revenue function and delivery costs of accepted online orders. We have one manufacturer and a retailer as the traditional channel and online customers as the e-tail channel who are served by the manufacturer. There exist two options for delivering products to the online customers i.e., shipping directly from the manufacturer to online customers which is available at any time, and delivering through the retail store which is available at specific periods however imposes less cost to the system. In this model, we have the assumption of unknown demand for e-tail channel (at any time there is no information about future arrivals of online orders) and deterministic demand (with fixed profit) for the traditional retail channel. Distinctive demand segments is considered for e-tail and retail channels. In the problem, we try to maximize a due-date sensitive profit function by quoting immediate and reliable due dates to online customers while considering manufacturer's capacity constraint and also the maximum acceptable lead time of online orders. There exists maximum acceptable lead time (L') for e-tail customers; if they are offered a due date after their desired lead time, they will not place the order. From the manufacturer's point of view, in fact manufacturer has the option of rejecting the online order by offering a due date after the desired lead time when there is no benefit in accepting the orders. The term due date in this problem is referred to the time that the order is shipped to the customer, i.e., the time the order leaves the manufacturer, thus the quoted due date for each order may be different from the time the order production is completed by the manufacturer. We assume that the revenue will decrease linearly if the quoted due dates for e-tail customers increase which was first used by Keskinocak et al. (2001). In order to illustrate the revenue in the objective function, let r be the revenue that is lost for each unit of time that the online order waits before being delivered to the customer, and l the time interval between the order's arrival time and its quoted due date. Then the revenue will be $r(L' - l)$. It is obvious that in this problem, the maximum revenue one can obtain from each online order is rL, where $L = L^{'} - p$ and p is the order production time. In this problem we quote 100% reliable due dates to the online customers, i.e., there is no tardiness cost, and all orders should be delivered by the quoted due dates. We also consider capacity constraint of processing at most N online orders at any time by the manufacturer.

There exist several solution approaches dealing with online optimization problems such as probabilistic analysis of heuristics, worst-case (competitive) analysis of heuristics, and queuing-theory based analysis of problems (Kaminsky and Hochbaum 2004). In this dissertation, as we consider unknown demand for the e-tail channel, competitive analysis is selected to evaluate the performance of online algorithms we will present.

In online optimization problems, online algorithm's performance is mainly evaluated via the competitive analysis, comparing an online algorithm's result with the offline model's optimal solution. For the problem we consider in this study, all the information about the online orders is available in advance for the optimal offline algorithms in order to obtain the maximum possible profit. Then, the results of both online and offline algorithms are compared. Given an instance I, let $Z_{(I)}$ denote the total profit obtained by using an online algorithm, and $Z^*_{(I)}$ denote the maximum profit obtained by an optimal offline algorithm. For maximization problems, the online algorithm is called $\rho-$competitive if $Z^*_{(I)} \leq \rho Z_{(I)} + b$ where $\rho \geq 1$, and b is a constant. We define competitive ratio as $\rho = sup\rho(\frac{Z^*_{(I)}}{Z_{(I)}})$ for $Z_{(I)} > 0$. Determining the bounds of competitive ratio (ρ) is the main issue and also the challenging part of online optimization problems.

The third problem we consider in this dissertation can be defined as an extension of the second problem with the same objective and problem structure. However, capacity constraint in this problem is modified to be a more realistic one. Unlike the second problem, we assume that the production capacity of the manufacturer is shared among the orders of both channels, i.e., there is no distinct production capacity for orders of different channels. This shared capacity assumption affects the number of online orders that the algorithm accepts, their schedule and clearly the schedule of retail orders. In addition, in this problem the fixed profit of retail orders is adjusted and variable revenue lost is considered for orders of this channel.

The solution procedure considered for this online optimization problem is competitive analysis as well.

1.5 Organization of the Dissertation

The dissertation is arranged as follows: Chapter 2 presents the first problem considered in this study: delivery schedule of jobs with cross families in a two stage manufacturing system with review of related literature. In Chapter 3, the problem of due date quotation in dual channel supply chain with channel's distinct-capacity constraint is discussed. An extension of the second problem is proposed in Chapter 4 which considers shared capacity of production for orders of both channels. Finally, Chapter 5 presents conclusions and future works.

Chapter 2

Delivery scheduling in a two-stage manufacturing system with cross families

2.1 Motivation and literature review

In this chapter, we study a delivery scheduling problem where the manufacturer has to decide the earliest delivery time for the orders received from retail channel. In this problem, a two-echelon supply chain is considered where a retailer places bulk orders of the same product with different families to the manufacturer. Since the manufacture accepts only bulk orders, no online order is assumed for this problem. The analysis with no online customers is relatively easy and therefore, we consider families of products in this problem. For this problem, we consider only retail channel with deterministic demand and cross family setup time (a novel assumption in literature) which was motivated by an application from the automotive industry. The problem in this chapter is scheduling of jobs in a two stage production system where jobs have family sequence-independent setup times to minimize the makespan. We assume that job allocations to families are machine based and therefore there are two family classes: family class for stage 1 and family class for stage 2. Thus each job will have two different family memberships and we call this problem jobs with *cross families*, i.e., two jobs belonging to two different families in one stage may belong to the same family in another stage. This system can be represented by a two machine permutation flow shop, where each stage is represented by a machine with cross family setup. Since the problem is NP-hard for arbitrary number of families, we study the problem when there is fixed number of family in each stage.

Past studies on scheduling with family setups have assumed that jobs belonging to the same family in one stage belong to the same family in other stages. However, this assumption may not be applicable for all production cases. For example, consider an automobile manufacturing system where all jobs are processed first in the body shop and then in the paint shop. In the body shop, jobs are categorized into two-door or four-door families and the same jobs are re-categorized in the paint shop based on color requirements. For instance, a two-door job and a four-door job belong to the same family in the paint shop if they both require the same color. To the best of our knowledge in scheduling literature, there is no study on jobs with cross families.

The permutation flow shop scheduling problem with makespan minimization has been well studied due to its important applications in manufacturing systems, assembly lines and information service facilities. Two machine flow shop scheduling problem was first studied by Johnson (1954). Yoshida & Hitomi (1979) studied the problem when setups are required before processing jobs. Special cases of flow shop scheduling problems are studied by Nouweland et al. (1992), Wlodzimierz (1977), Johnny et al. (1992), Chuanli & Hengyong (2012) and Lin-Hui et al. (2012). When there are family setups, jobs of the same family are grouped into batches in order to minimize resource required for setups. There are several studies considering batching in flow shops with different assumptions, such as Logendran et al. (2006), Hendizadeha et al. (2008), Voxa & Wittb (2007) and Bozorgirad & Logendran (2013). Readers may refer to studies of Allahverdi (2008) and Edwin et al. (2000) for a complete survey on batching. The problem addressed in this chapter is closely related to studies on two machine flow shop scheduling with family-sequence-independent setups to minimize makespan. There have been many studies on this problem over the past four decades. Scheduling problems with family setups can be classified into two main classes, (i) scheduling with group technology (GT) assumption, and (ii) scheduling without GT assumption.

In group technology, jobs of the same family are scheduled into single batch and thus it simplifies the problem. Therefore, for problems with fixed number of families, researchers were able to develop polynomially bounded algorithms with GT assumption. Sekiguchi (1983) proved the optimality of Johnson job sequence within each batch and used composite job approach to schedule families under series-parallel precedence constraint. Ham et al. (1985) presented a two-step procedure to find an optimal solution for this problem and Baker (1990) developed a polynomially bounded algorithm based on the results obtained by Ham et al. (1985). Logendran & Sriskandarajah (1993), Marco (2004), Lee & Mirchandani (1988), Cheng & Wang (1999), and Cheng & Wang (1998) studied two machine flow shop problems with setups under special conditions such as zero-buffer, limited-buffer, identical versatile machines, one-setup problem, discrete or batch processor machines, and provided different solution techniques. Yang & Chern (2000) considered job removal time and transportation time between machines and proposed polynomial-time algorithm.

Two machine scheduling problem to minimize makespan without GT assumption is more difficult than the problem with GT assumptions. Therefore, researchers developed heuristic algorithms, approximation algorithms, and for some special cases polynomial time algorithms. Kleinau (1993) showed that this problem is NP-hard for both anticipatory (setup can start on the second machine before the processing of the job on the first machine is finished) and non-anticipatory (setup cannot start on the second machine until the processing of the job is finished on the first machine) setups when there are arbitrary number of families. Zdrzalka (1995) developed heuristic algorithms and investigated their worst-case performances. Danneberg et al. (1998) compared several heuristic algorithms for the problem with limited buffer between machines. Lin & Cheng (2001) studied the problem with no-wait and batch availability assumptions, and proved that the problem is strongly NP-hard. They proposed an optimal batch size formulation when jobs have identical processing time. Chen et al. (1998) proposed two heuristic algorithms with $O(n \log n)$ time to solve the problem with arbitrary number of job families. The first algorithm, with GT assumption and applying Johnson's algorithm and the second one with relaxing GT assumption and applying open shop scheduling technique in order to improve the worst case ratio. Cheng et al. (2000) proved that the problem with batch availability assumption is strongly NP-hard, and presented a heuristic algorithm while investigating some special cases. Agnieszka & Rudek (2013) developed meta heuristic algorithms using tabu search and simulated annealing when processing times follow aging effect function.

In this chapter, we study scheduling of jobs with cross families and sequence independent setups in two machine flow shop to minimize makespan when there is fixed number of families. We first investigate some properties of the optimal schedule and show that Johnson sequence is optimal for jobs belonging to the same family on both machines. We develop an efficient branch and bound algorithm with complexity of $O(n^c)$, where c is the total number of families and is a constant, and a hybrid genetic algorithm for large scale problems using the properties of the optimal schedule. The chapter is organized as follows; Section 2.2 defines the problem with notation and assumptions. Section 2.3 discusses some preliminaries for our problem and analyzes properties of optimal schedule. Section 2.4 presents a branch and bound algorithm to sequence given batches, optimally. Section 2.5 discusses on optimal scheduling, and Section 2.6 presents a hybrid genetic algorithm for large scale problems. Finally computational experiment is provided in Section 2.7 and conclusion is given in Section 2.8.

2.1.1 Solution Procedure

The solution procedure considered in this chapter is first investigating the properties of job sequences as well as job batches in the optimal schedule, and generating ordered batches of the optimal schedule, and then developing a branch and bound algorithm for optimal sequencing of given ordered batches.

2.2 Problem Definition

We are given a set of n jobs $\{J_1, J_2, ..., J_n\}$ that has to be scheduled in a two machine flow shop to minimize the makespan ($C_{\max}$). There are two family types, families on machine 1 (M_1) and families on machine 2 (M_2). Each job has two family memberships, its family on M_1 and its family on M_2. An anticipatory sequence independent setup is required on each machine when switching from one family to another. Job J_j requires a processing time of $p_{j,l}$ and a setup time of $s_{j,l}$ on M_l. When any group of jobs having the same family on both machines are scheduled consecutively, no setup is required on either machine except for the first job in that group. We call such group a batch, and without loss of generality, we interchangeably use $s_{r,l}$ to denote the setup time of J_r on M_l or the setup time of β_r (the r^{th} batch) on M_l.

There is a fixed number of families on each machine; K families on M_1 and L families on M_2. All jobs are available at time zero, and processing of a job on the second machine can be started immediately after the completion of that job on the first machine. Jobs follow the same processing order on both machines and a machine can process at most one job at a time. Processing of a job cannot be interrupted and there is an unlimited buffer capacity between machines. We describe this scheduling problem using the three-field notation of Graham et al. (1979) as $F2/ST, SI, CB/C_{\max}$, where $F2$ stands for a two machine flow shop, ST for setup time, $C_{\max}$ for makespan, SI and CB stands for sequence independent setups and cross families respectively.

We use the following additional notations:

$\tau(f,g)$	The set of all jobs belonging to the f^{th} family on M_1 and g^{th} family on M_2, i.e., jobs having the same family on both machines.
$I_j(\varphi)$	The idle time on M_2 immediately before starting processing (after setup if required) of job J_j in any given sequence φ.
$A_j(\varphi)$	The total idle time on M_2 before processing job J_j, in any given sequence φ.
$T_{j,l}(\varphi)$	The start time of job J_j (or batch β_j) on M_l after any setup, if required, in any given sequence φ.

$$a_{i,j} = a_{j,i} = \begin{cases} 0, \text{ if jobs } J_i \text{ and } J_j \text{ (or batches } \beta_i \text{ and } \beta_j\text{) belong to the same} \\ \text{family on machine } M_1. \\ 1, \text{ otherwise} \end{cases}$$

$$b_{i,j} = b_{j,i} = \begin{cases} 0, \text{ if jobs } J_i \text{ and } J_j \text{ (or batches } \beta_i \text{ and } \beta_j\text{) belong to the same} \\ \text{family on machine } M_2. \\ 1, \text{ otherwise} \end{cases}$$

$U = [a_{i,j}]_{n \times n}$

$V = [b_{i,j}]_{n \times n}$

Note that there will be at most $K \times L$, $\tau(f, g)$ sets in this problem. For illustration consider an example with job set $\{1, 2, ..., 7\}$ and $K = L = 2$. Let job set in families 1 and 2 on M_1 be $\{1, 2, 3, 4\}$ and $\{5, 6, 7\}$ respectively, and job set in families 1 and 2 on M_2 be $\{2, 4, 7\}$ and $\{1, 3, 5, 6\}$. Then, $\tau(1, 1) = \{2, 4\}$, $\tau(1, 2) = \{1, 3\}$, $\tau(2, 1) = \{7\}$, and $\tau(2, 2) = \{5, 6\}$. For jobs 1 and 2, $a_{1,2} = a_{2,1} = 0$, $b_{1,2} = b_{2,1} = 1$ because jobs 1 and 2 belong to the same family on machine M_1 and to different families on machine M_2.

Let σ represent job sequence in the order of job index; $\sigma = \{J_1, J_2, ..., J_{n-1}, J_n\}$. We define $K_r(\sigma)$ as

$K_r(\sigma) =$ (total busy time of M_1 before starting J_r on M_2) $-$ (total busy time of M_2 before starting J_r on M_2). Therefore, we have

$K_1(\sigma) = p_{1,1} + s_{1,1} - s_{1,2}$, and

$$K_r(\sigma) = \Big(\sum_{d=1}^{r} (p_{d,1} + s_{d,1} a_{d-1,d})\Big) - \Big(\sum_{d=1}^{r-1} p_{d,2} + \sum_{d=1}^{r} s_{d,2} b_{d-1,d}\Big), \text{ for } (r > 1)$$

2.3 Preliminaries

In this section, we first study some preliminaries for our problem and then we discuss properties of optimal schedules that lead us to develop efficient algorithms. Consider a two machine flow shop scheduling problem with sequence independent cross family setup and arbitrary number of families on each machine. Kleinau (1993) proved that the problem of batch scheduling with sequence independent anticipatory setup in a two machine flow shop to minimize makespan is NP-hard for arbitrary number of families. The problem with cross families will be equivalent to the problem studied by Kleinau, if all jobs of the same family on one machine belong to the same family on the other machine. Therefore their problem is a special case of the problem with cross families. Thus the problem of batch scheduling with cross families when there are arbitrary number of families is NP-hard. Therefore, the following Theorem holds:

Theorem 2.1. *(Kleinau (1993)) The two machine flow shop problem with cross families to minimize makespan is NP-hard for arbitrary number of families.*

Remark 2.1. *When the number of families is arbitrary, it is still unknown whether the problem of two machine flow shop with cross families to minimize makespan, is strongly or ordinary NP-hard.*

Proposition 2.1. $A_r(\sigma) = \max\limits_{1 \le d \le r} \{K_d(\sigma), 0\}$ *for* $(r = 1, 2, ..., n)$.

Proof: It is clear that for job $J_r \in \sigma$, $A_r(\sigma) = T_{r,2}(\sigma) - \sum_{d=1}^{r-1} p_{d,2} - \sum_{d=1}^{r} s_{d,2} b_{d-1,d}$, where

$T_{r,2}(\sigma) = \max\{\sum_{d=1}^{r} (p_{d,1} + s_{d,1} a_{d-1,d})\ ,\ A_{r-1}(\sigma) + \sum_{d=1}^{r-1} p_{d,2} + \sum_{d=1}^{r} s_{d,2} b_{d-1,d}\}$. Thus,

$A_r(\sigma) = \max\{\sum_{d=1}^{r} (p_{d,1} + s_{d,1} a_{d-1,d}) - \sum_{d=1}^{r-1} p_{d,2} - \sum_{d=1}^{r} s_{d,2} b_{d-1,d}\ ,\ A_{r-1}(\sigma)\}$, and

$A_r(\sigma) = \max\{K_r(\sigma), A_{r-1}(\sigma)\}$ for $r = 1, 2, ..., n$. We have,

$A_r(\sigma) = \max\{K_r(\sigma), K_{r-1}(\sigma), ..., K_2(\sigma), A_1(\sigma)\}$ and $A_1(\sigma) = \max\{K_1(\sigma), 0\}$. Therefore, $A_r(\sigma) = \max_{1 \le d \le r}\{K_d(\sigma), 0\}$. □

In this chapter, we define a *batch* as a partial sequence of jobs in which a setup is required on machine M_1 *and/or* M_2 only for the first job of the partial sequence. When job sequence of a given batch is known, we define it as *ordered batch.* Note that any set $\tau(f, g)$ can be split into more than one batch.

Proposition 2.2. *There exists an optimal schedule with no idle time on machine* M_1 *until the completion of the last job.*

Proof: The proof of this proposition is trivial and is therefore omitted.

Proposition 2.3. *Setup times on machine* M_l *(for* $l = 1, 2$*) satisfy the triangular inequality, i.e.,* $s_{j,1} a_{i,j} + s_{k,1} a_{j,k} \ge s_{k,1} a_{i,k}$, *(similarly* $s_{j,2} b_{i,j} + s_{k,2} b_{j,k} \ge s_{k,2} b_{i,k}$*)*

Proof: First consider the setups on M_1. It is clear that if $(a_{i,k} = 0)$ or $(a_{i,k} = 1$ and $a_{j,k} = 1)$, the inequality is satisfied. If $a_{i,k} = 1$ and $a_{j,k} = 0$, then jobs J_j and J_k belong to the same family on first machine. Therefore, $s_{j,1} = s_{k,1}$, $a_{i,j} = a_{i,k} = 1$, and

$s_{j,1}a_{i,j} + s_{k,1}a_{j,k} = s_{j,1}a_{i,j} = s_{k,1}a_{i,k}$. Thus the inequality is satisfied. Similarly we can prove that the inequality is satisfied for M_2 too, i.e., $s_{j,2}b_{i,j} + s_{k,2}b_{j,k} \geq s_{k,2}b_{i,k}$. □

2.3.1 Some properties of job sequence in the optimal schedule

Sekiguchi (1983) proved that for two machine flow shop scheduling problem with family setup, the Johnson job sequence within batches is optimal. In this subsection, we prove that Johnson job sequence for jobs belonging to the same family on both machines is optimal, i.e., if $J_i, J_j \in \tau(f,g)$ and $\min\{p_{j,1}, p_{i,2}\} \leq \min\{p_{i,1}, p_{j,2}\}$ then there exists an optimal schedule where job j precedes job i ($J_j \prec J_i$). We first prove Lemmas 2.1 and 2.2 in order to prove Johnson property in Theorem 2.2.

Lemma 2.1. *For jobs J_i, $J_j \in \tau(f,g)$, if $p_{j,1} \leq p_{i,2}$, and if ($p_{j,1} \leq p_{i,1} \leq p_{j,2}$ or $p_{j,1} \leq p_{j,2} \leq p_{i,1}$), then there exists an optimal schedule in which $J_j \prec J_i$.*

Proof: Let us assume that the lemma does not hold. Then in the optimal schedule $J_i \prec J_j$. We assume, without loss of generality (w.l.o.g.), that the optimal schedule is σ and $i < j$, (Figure 2.1). Move J_j immediately before J_i in σ and let this modified schedule be σ', (Figure 2.2).

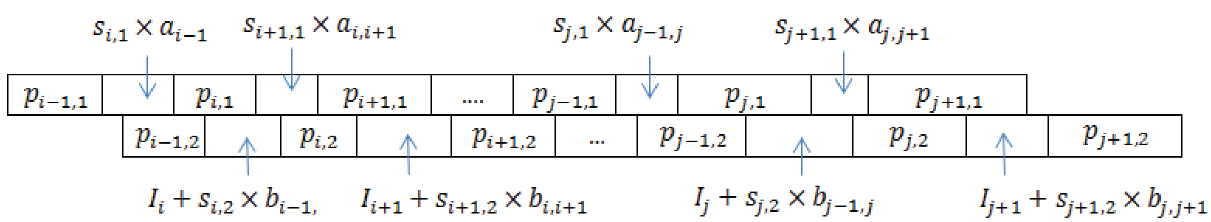

FIGURE 2.1. Optimal Schedule σ

Then we have, $K_1(\sigma) = p_{1,1} + s_{1,1} - s_{1,2}$, and

$$K_r(\sigma) = K_{r-1}(\sigma) + p_{r,1} + s_{r,1}a_{r-1,r} - p_{r-1,2} - s_{r,2}b_{r-1,r}, \text{ for } r = 2, 3, ..., n.$$

$$\begin{aligned}
K_r(\sigma') &= K_r(\sigma), \qquad \text{for } r = 1, 2, ..., i-1 && (2.1)\\
K_j(\sigma') &= K_{i-1}(\sigma') + p_{j,1} + s_{j,1}a_{i-1,j} - p_{i-1,2} - s_{j,2}b_{i-1,j}\\
&= K_{i-1}(\sigma) + p_{j,1} + s_{i,1}a_{i-1,i} - p_{i-1,2} - s_{i,2}b_{i-1,i}\\
&\leq K_{i-1}(\sigma) + p_{i,1} + s_{i,1}a_{i-1,i} - p_{i-1,2} - s_{i,2}b_{i-1,i} = K_i(\sigma) && (2.2)\\
K_i(\sigma') &= K_j(\sigma') + p_{i,1} - p_{j,2}\\
&= K_{i-1}(\sigma) + p_{j,1} + s_{i,1}a_{i-1,i} - p_{i-1,2} - s_{i,2}b_{i-1,i} + p_{i,1} - p_{j,2}\\
&\leq K_{i-1}(\sigma) + s_{i,1}a_{i-1,i} - p_{i-1,2} - s_{i,2}b_{i-1,i} + p_{i,1} = K_i(\sigma) && (2.3)\\
K_r(\sigma') &= K_{r-1}(\sigma') + p_{r,1} + s_{r,1}a_{r-1,r} - p_{r-1,2} - s_{r,2}b_{r-1,r}, \text{ for } r = i+1, i+2, ..., j-1\\
&\leq K_{r-1}(\sigma) + p_{r,1} + s_{r,1}a_{r-1,r} - p_{r-1,2} - s_{r,2}b_{r-1,r}\\
&= K_r(\sigma) && (2.4)
\end{aligned}$$

From Equations (2.1)-(2.4), one can see that $\max\limits_{1\leq r\leq j}\{K_r(\sigma')\} \leq \max\limits_{1\leq r\leq j}\{K_r(\sigma)\}$. Therefore $A_j(\sigma') = \max\limits_{1\leq r\leq j}\{K_r(\sigma'), 0\} \leq \max\limits_{1\leq r\leq j}\{K_r(\sigma), 0\} = A_j(\sigma)$.

From Proposition 2.3, it is clear that $T_{j+1,1}(\sigma') \leq T_{j+1,1}(\sigma)$.

$$\begin{aligned}
T_{j+1,2}(\sigma') &= A_{j-1}(\sigma') + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2}b_{d-1,d} - s_{j,2}b_{j-1,j} - s_{j+1,2}b_{j,j+1} + s_{j+1,2}b_{j-1,j+1}\\
&\leq A_{j-1}(\sigma') + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2}b_{d-1,d}\\
&\leq A_{j-1}(\sigma) + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2}b_{d-1,d} = T_{j+1,2}(\sigma)
\end{aligned}$$

Since $T_{j+1,1}(\sigma') \leq T_{j+1,1}(\sigma)$ and $T_{j+1,2}(\sigma^{'}) \leq T_{j+1,2}(\sigma)$, makespan of σ' cannot be more than the makespan of σ. This contradicts the assumption. □

Lemma 2.2. *For jobs J_i, $J_j \in \tau(f, g)$, if $p_{i2} \leq p_{j1}$, and if ($p_{i2} \leq p_{i1} \leq p_{j2}$ or $p_{i2} \leq p_{j2} \leq p_{i1}$), then there exists an optimal schedule in which $J_j \prec J_i$.*

Proof: Let us assume that the lemma does not hold and $J_i \prec J_j$ in the optimal

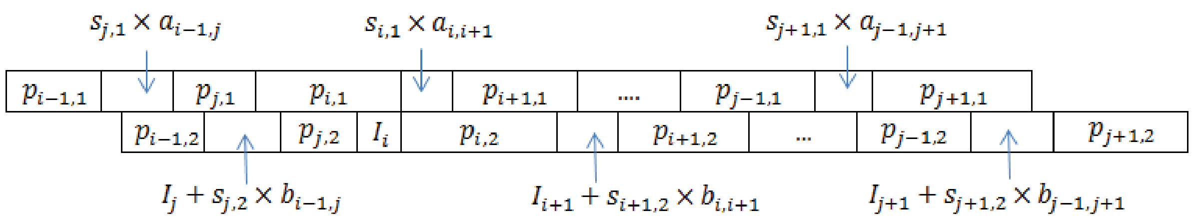

FIGURE 2.2. Modified Schedule σ'

schedule. Also let us assume, w.l.o.g., that the optimal schedule is σ and $i < j$. Move J_i immediately after J_j in σ and let this modified schedule be σ''.

$$
\begin{aligned}
K_r(\sigma'') &= K_r(\sigma), \qquad \text{for } r = 1, 2, ..., i-1 && (2.5)\\
K_{i+1}(\sigma'') &= K_{i-1}(\sigma'') + p_{i+1,1} + s_{i+1,1}a_{i-1,i+1} - p_{i-1,2} - s_{i+1,2}b_{i-1,i+1} &&\\
&= K_{i-1}(\sigma) + p_{i+1,1} + s_{i+1,1}a_{i-1,i+1} - p_{i-1,2} - s_{i+1,2}b_{i-1,i+1} &&\\
&= K_{i+1}(\sigma) + (s_{i+1,1}a_{i-1,i+1} - s_{i,1}a_{i-1,i} - s_{i+1,1}a_{i,i+1}) &&\\
&\quad +(s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1}) + p_{i,2} - p_{i,1} &&\\
&\leq K_{i+1}(\sigma) + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1}) && (2.6)\\
K_r(\sigma'') &= K_{r-1}(\sigma'') + p_{r,1} + s_{r,1}a_{r-1,r} - p_{r-1,2} - s_{r,2}b_{r-1,r}, \qquad \text{for } r = i+2, ..., j &&\\
&\leq K_r(\sigma) + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1}) && (2.7)\\
K_i(\sigma'') &= K_j(\sigma'') + p_{i,1} - p_{i,2} \leq K_j(\sigma'') &&\\
&\leq K_j(\sigma) + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1}) && (2.8)
\end{aligned}
$$

Equations (2.5)-(2.8) show that $\max\limits_{1\leq r\leq j}\{K_r(\sigma'')\} \leq \max\limits_{1\leq r\leq j}\{K_r(\sigma)\} + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1})$. Therefore,

$$
\begin{aligned}
A_i(\sigma'') &= \max_{1\leq r\leq j}\left\{K_r(\sigma''), 0\right\}\\
&\leq \max_{1\leq r\leq j}\{K_r(\sigma), 0\} + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1})\\
&= A_j(\sigma) + (s_{i,2}b_{i-1,i} + s_{i+1,2}b_{i,i+1} - s_{i+1,2}b_{i-1,i+1})
\end{aligned}
$$

It is clear that $T_{j+1,1}(\sigma'') \leq T_{j+1,1}(\sigma)$.

$$
\begin{aligned}
T_{j+1,2}(\sigma) &= A_j(\sigma) + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2} b_{d-1,d} \\
T_{j+1,2}(\sigma'') &= A_i(\sigma'') + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2} b_{d-1,d} - (s_{i,2} b_{i-1,i} + s_{i+1,2} b_{i,i+1} - s_{i+1,2} b_{i-1,i+1}) \\
&\leq A_j(\sigma) + \sum_{d=1}^{j} p_{d,2} + \sum_{d=1}^{j+1} s_{d,2} b_{d-1,d} = T_{j+1,2}(\sigma)
\end{aligned}
$$

Since $T_{j+1,1}(\sigma'') \leq T_{j+1,1}(\sigma)$ and $T_{j+1,2}(\sigma'') \leq T_{j+1,2}(\sigma)$, the makespan of σ''can not be more than the makespan of σ. This contradicts the assumption. □

Theorem 2.2. *In the optimal schedule, jobs belonging to any set $\tau(f,g)$ follow Johnson sequence.*

Proof: Let the theorem is not true. Then in the optimal schedule, there must be at least two jobs $J_i, J_j \in \tau(f,g)$ (for some f,g), where $\min\{p_{j,1}, p_{i,2}\} \leq \min\{p_{i,1}, p_{j,2}\}$ and $J_i \prec J_j$. There are four possible cases:

1. $(p_{j,1} \leq p_{i,2})$ and $(p_{i,1} \leq p_{j,2})$, i.e., $(p_{j,1} \leq p_{i,2})$ and $(p_{j,1} \leq p_{i,1} \leq p_{j,2})$
2. $(p_{j,1} \leq p_{i,2})$ and $(p_{j,2} \leq p_{i,1})$, i.e., $(p_{j,1} \leq p_{i,2})$ and $(p_{j,1} \leq p_{j,2} \leq p_{i,1})$
3. $(p_{i,2} \leq p_{j,1})$ and $(p_{i,1} \leq p_{j,2})$, i.e., $(p_{i,2} \leq p_{j,1})$ and $(p_{i,2} \leq p_{i,1} \leq p_{j,2})$
4. $(p_{i,2} \leq p_{j,1})$ and $(p_{j,2} \leq p_{i,1})$, i.e., $(p_{i,2} \leq p_{j,1})$ and $(p_{i,2} \leq p_{j,2} \leq p_{i,1})$

Lemma 2.1 for cases 1 and 2, and Lemma 2.2 for cases 3 and 4, prove that scheduling J_j before J_i does not increase the makespan. □

2.3.2 Some properties of batches in optimal schedule

In this subsection, we first discuss rules that leads us to form initial batches (Lemmas 2.3 and 2.4). Then, we show that any ordered batch can be converted to an equivalent job that will be used in our algorithm (Lemma 2.6).

Lemma 2.3. *There exists an optimal schedule in which jobs J_i , $J_j \in \tau(f, g)$ are scheduled in the same batch if* $p_{i,1} \leq p_{i,2}$, $p_{j,1} \leq p_{j,2}$, *and* $p_{i,1} \leq p_{j,2}$.

Proof : Let, w.l.o.g., the optimal schedule be σ and $J_i \prec J_j$. Note that from Theorem 2.2, J_i and J_j follow Johnson rule. Let us assume that the lemma does not hold then jobs J_i and J_j are scheduled in different batches in σ. Let σ' be the modified schedule when J_j is scheduled immediately after J_i in σ, i.e., jobs J_i and J_j are in the same batch.

$$
\begin{aligned}
K_r(\sigma) &= K_r(\sigma'), \qquad \text{for} \;\; r = 1, ..., i && (2.9)\\
K_j(\sigma') &= K_i(\sigma') + p_{j,1} - p_{i,2} = K_i(\sigma) + p_{j,1} - p_{i,2} \leq K_i(\sigma) && (2.10)\\
K_{i+1}(\sigma') &= K_j(\sigma') + p_{i+1,1} + s_{i+1,1} \times a_{j,i+1} - p_{j,2} - s_{i+1,2} \times b_{j,i+1} &&\\
&= K_i(\sigma) + p_{j,1} - p_{i,2} + p_{i+1,1} + s_{i+1,1} \times a_{i,i+1} - p_{j,2} - s_{i+1,2} \times b_{i,i+1} &&\\
&= K_{i+1}(\sigma) + p_{j,1} - p_{j,2} \leq K_{i+1}(\sigma) && (2.11)\\
K_r(\sigma') &= K_{r-1}(\sigma') + p_{r,1} + s_{r,1} \times a_{r-1,r} - p_{r-1,2} - s_{r,2} \times b_{r-1,r}, \;\; \text{for } r = i+2, ..., j-1 &&\\
&= K_r(\sigma) + p_{j,1} - p_{j,2} \leq K_r(\sigma) && (2.12)
\end{aligned}
$$

It is clear that $T_{i+1,1}(\sigma') \leq T_{i+1,1}(\sigma)$. From Equations (2.9)-(2.12), it can be easily proved that $T_{i+1,2}(\sigma') \leq T_{i+1,2}(\sigma)$. Therefore makespan of σ'can not be more than the makespan of σ. □

Lemma 2.4. *There exists an optimal schedule in which jobs $J_i, J_j \in \tau(f, g)$ are scheduled in the same batch if* $p_{i,2} \leq p_{i,1}$, $p_{j,2} \leq p_{j,1}$, *and* $p_{i,2} \leq p_{j,1}$.

Proof: The proof of this Lemma is similar to the proof for Lemma 2.3 and therefore it is omitted. □

Algorithm OptBatch given below generates ordered batches using the rules in Lemma 2.3 and Lemma 2.4.

Algorithm OptBatch

Step 1. Sequence jobs of each set $\tau(f, g)$ $(\forall g, h)$ in Johnson order.

Step 2. Group jobs of each family into batches using Lemmas 2.3 and 2.4.

End.

Lemma 2.5. *The complexity of algorithm OptBatch is $O(n^2)$.*

Proof: In algorithm OptBatch in the first step, sequencing jobs in Johnson order takes $O(n\ \log(n))$ time. Lemma 2.3 in the second step requires $O(n)$ time to get the group of jobs that their processing time on M_1 is less than or equal to the processing time on M_2. Batching the jobs in that group (i.e., checking $p_{i,1} \leq p_{j,2}$) requires at most $O(n^2)$ time. Therefore, the complexity of batching jobs using Lemma 2.3 in the second step is $O(n^2)$. Similarly, batching jobs using Lemma 2.4 has the complexity of $O(n^2)$. Thus the complexity of algorithm OptBatch is $O(n^2)$. □

Remark 2.2. *For any optimal ordered batch, i.e., batch with fixed job sequence, all idle times on M_2 can be shifted to the beginning without increasing completion times on M_1 and M_2.(Figure 2.3(a) and 2.3(b))*

Hereafter, we consider the schedule of an ordered batch as the one that follows Remark 2.2. Assume that the r^{th} batch (β_r) has u_r number of jobs. Without loss of generality we assume that the jobs assigned to the batch are $\{J_1, J_2, ..., J_{u_r}\}$.

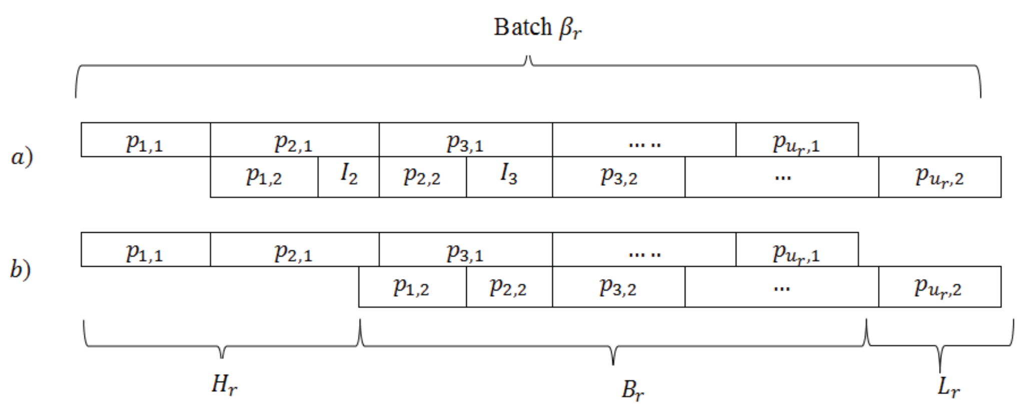

FIGURE 2.3. a) Schedule of batch β_r before shifting all idle times on M_2 to the beginning. b) Schedule of batch β_r after shifting all idle times on M_2 to the beginning

We represent the ordered batch β_r by three components H_r, B_r, and L_r as shown in Figure 2.3(b), where,

(i) H_r is the total idle time on M_2 for the batch β_r,

$H_r = \max\left\{ \max\limits_{2 \le j \le u_r} \{\sum\limits_{d=1}^{j} p_{d,1} - \sum\limits_{d=1}^{j-1} p_{d,2}\} \ ,\ p_{i,1} \right\}.$

(ii) B_r is the time that both machines are busy processing jobs in β_r,

$B_r = \sum\limits_{d=1}^{u_r} p_{d,1} - H_r.$

(iii) L_r is the idle time on M_1 for the batch β_r,

$L_r = H_r + \sum\limits_{d=1}^{u_r} p_{d,2} - \sum\limits_{d=1}^{u_r} p_{d,1}.$

Lemma 2.6. *Consider a schedule ϕ with w ordered batches sequenced in the order of index, i.e., $\phi = \{\beta_1, \beta_2, ..., \beta_{w-1}, \beta_w\}$. Now consider w jobs $J_1, J_2, ..., J_w$ that $p_{i,1} = H_i$, $p_{i,2} = L_i$ and job J_i belongs to the family of batch β_i for $i = 1, ..., w$. Let the schedule ϕ' be the sequence of jobs $\{J_1, J_2, ..., J_w\}$ in the order of index. Then $C_{\max}(\phi) = C_{\max}(\phi') + \sum\limits_{d=1}^{w} B_d$.*

Proof: For job sequence ϕ',

$$K_r(\phi') = \sum_{d=1}^{r} H_d + \sum_{d=1}^{r} s_{d,1} a_{d-1,d} - \sum_{d=1}^{r-1} L_d - \sum_{d=1}^{r} s_{d,2} b_{d-1,d}.$$

For batch sequence ϕ, let u_r be the number of jobs in β_r. Then, for any job $J_i \in \beta_r$,

$$\begin{aligned} K_i(\phi) \quad = \quad & \sum_{d=1}^{r-1}(H_d + B_d) - \sum_{d=1}^{r-1}(L_d + B_d) + \sum_{d=1}^{r} s_{d,1} a_{d-1,d} - \sum_{d=1}^{r} s_{d,2} b_{d-1,d} \ + \\ & \sum_{d \in \{\beta_r | d \le i\}} p_{d,1} - \sum_{d \in \{\beta_r | d \le i\}} p_{d,2}. \end{aligned}$$

Then,

$$\begin{aligned}
\max_{i\in\beta_r}\{K_i(\phi)\} &= \sum_{d=1}^{r-1} H_d - \sum_{d=1}^{r-1} L_d + \sum_{d=1}^{r} s_{d,1}a_{d-1,d} - \sum_{d=1}^{r} s_{d,2}b_{d-1,d} \ + \\
&\max_{i\in\beta_r}\{\sum_{d\in\{\beta_r|d\leq i\}} p_{d,1} - \sum_{d\in\{\beta_r|d\leq i\}} p_{d,2}\} \\
&= \sum_{d=1}^{r-1} H_d - \sum_{d=1}^{r-1} L_d + \sum_{d=1}^{r} s_{d,1}a_{d-1,d} - \sum_{d=1}^{r} s_{d,2}b_{d-1,d} + H_r = K_r(\phi').
\end{aligned}$$

Also, $A_w(\phi) = \max_{i\in\beta_1,\beta_2,\ldots,\beta_w}\{K_i(\phi)\} = \max_{r=1,\ldots,w}\{\max_{i\in\beta_r}\{K_i(\phi)\}\}$.

Therefore, $A_w(\phi) = \max_{r=1,\ldots,w}\{K_r(\phi')\} = A_w(\phi')$. $\square$

Remark 2.3. *Since* $\sum_{d=1}^{w} B_d$ *is a constant for given set of batches, solving the problem* $\{J_i|i = 1, 2, ..., w\}$ *to minimize makespan will be sufficient to get the optimal batch sequence for batches* $\{\beta_i|i = 1, 2, ..., w\}$ *. Therefore, we can replace batch* β_r *by job* J_r *for* $r = 1, 2, ...w$*, and solve the problem for makespan minimization for the jobs.*

Remark 2.4. *Algorithm OptBatch generates initial ordered batches by grouping some jobs. Remark 2.3 claims that generated batches can be replaced by equivalent jobs. Therefore, when equivalent jobs are used in our algorithms, one may expect reduced number of jobs.*

2.4 Branch and bound algorithm

In this section, we develop a branch and bound ($B\&B$) algorithm to sequence given ordered batches. Below we briefly describe the procedure of our branch and bound algorithm. We use depth first search method for branching in our B&B algorithm. In order to fathom branches, we use a lower bound and dominance rules. Subsections 2.4.1 and 2.4.2 describe lower bound and two dominance rules respectively.

Note that any set of ordered batches $\{\beta_i|i = 1, 2, ..., w\}$ can be replaced by corresponding job set $\{J_i|i = 1, 2, ..., w\}$ for makespan minimization as explained in Lemma 2.6 by setting job J_r with $p_{r,1} = H_r$, and $p_{r,2} = L_r$ for batch β_r. Therefore, hereafter, we consider sequence of corresponding job set instead of ordered batch set.

2.4.1 Lower bound

In this subsection, we develop a lower bound for B&B algorithm. Propositions 2.4 and 2.5, and Remarks 2.5 – 2.7 are used to develop the lower bound given in Lemma 2.8. We use the following additional notation in this subsection:

- θ Set of jobs $\{J_1, J_2,...,J_w\}$ corresponding to the set of ordered batches $\{\beta_1, \beta_2,..., \beta_w\}$
- ψ Optimal schedule for partial job set θ' ($\theta' \subset \theta$) in which the last job belongs to set $\tau(g, h)$
- δ_l Set of families on M_l for jobs in $\theta\backslash\theta'$ (for $l = 1, 2$)

Let us assume w.l.o.g., $\theta' = \{J_1, J_2, ..., J_{i-1}\}$, and jobs in ψ are assigned in the order of index. Let C_1 and C_2 be completion times of ψ on M_1 and M_2 respectively. Then $\theta\backslash\theta' = \{J_i, J_{i+1}, ..., J_w\}$. Let ϕ be the schedule obtained when all remaining jobs in $\theta\backslash\theta'$ are appended to ψ in the order of index, i.e., $\phi = \psi, \psi' = J_1, J_2, ..., J_w$.

Proposition 2.4. *Let ϕ' be a fictitious schedule obtained by moving non-zero setup time of a job J_r , $(i \leq r \leq w)$ on M_1 to the end of the schedule ϕ. Then $C_{\max}(\phi') \leq C_{\max}(\phi)$.*

Remark 2.5. *In a schedule ϕ, if setups of all jobs in $\theta\backslash\theta'$ on M_1 are moved to the end, then the makespan of the new schedule will not be larger than $C_{\max}(\phi)$. All the setups which are moved to the end of the schedule ϕ can be considered as a fictitious job.*

Proposition 2.5. *Consider a fictitious schedule ϕ'' obtained when non-zero setup time of job J_r, $(i < r \leq w)$ on M_2 in ϕ is moved to immediately after completion of ψ(immediately before the setup of J_i, if required). Then $C_{\max}(\phi'') \leq C_{\max}(\phi)$.*

Remark 2.6. *In a schedule ϕ, if setups of all jobs in $\theta\backslash\theta'$ on M_2 are moved to immediately after completion of ψ, then the makespan of the new schedule will not be larger than $C_{\max}(\phi)$. All the setups which are moved to immediately after completion of ψ, can be considered as a fictitious job.*

Remark 2.7. *In an optimal schedule, for jobs in $\theta \backslash \theta'$, we need at least one setup for each family of $\delta_1/\{g\}$ on M_1 and at least one setup for each family of $\delta_2/\{h\}$ on M_2. In an optimal schedule, therefore, the total setups for the job set $\theta \backslash \theta'$on M_1 and M_2 satisfies the following inequalities:* $($*the total setup time on* $M_1) \geq p_{w+1,1}$ *and* $($*the total setup time on* $M_2) \geq p_{0,2}$.

Lemma 2.7. *If the sequence of jobs in $\theta \backslash \theta'$ in ϕ is ψ' when appended to ψ (i.e., $\phi = \psi, \psi'$) and if no setup is required for jobs in $\theta \backslash \theta'$, then ψ' must follow Johnson rule for minimum $C_{\max}(\phi)$, where ψ is fixed.*

Proof: Let, w.l.o.g. $\psi' = J_i, J_{i+1}, ..., J_w$. Then $K_r(\phi) = K_r(\psi, \psi')$.

$K_r(\psi, \psi') = K_r(\psi)$ for $r \leq i-1$. When $r \geq i$,

$K_r(\psi, \psi') = K_{i-1}(\psi, \psi') + (\sum_{d=i}^{r} p_{d,1} + s_{d,1}a_{d-1,d} - \sum_{d=i}^{r} s_{d,2}b_{d-1,d} - \sum_{d=i}^{r-1} p_{d,2}) - p_{i-1,2} =$
$K_{i-1}(\psi) + K_r(\psi') - p_{i-1,2}$.

Total idle time on M_2 in ϕ,

$A_w(\psi, \psi') = \max\{\max_{r \geq i}\{K_{i-1}(\psi) + K_r(\psi') - p_{i-1,2}\},$
$\max_{r \leq i-1}\{K_r(\psi)\}\} = \max\{K_{i-1}(\psi) - p_{i-1,2} + \max_{r \geq i}\{K_r(\psi')\}, A_{i-1}(\psi)\} =$
$\max\{K_{i-1}(\psi) - p_{i-1,2} + A_w(\psi'), A_{i-1}(\psi)\}$.

We know that the partial schedule of ψ in ϕ is fixed, and therefore, $K_{i-1}(\psi)$, $p_{i-1,2}$ and $A_{i-1}(\psi)$ are fixed. Thus if schedule ψ' gives minimum makespan for job set $\theta \backslash \theta'$, then the schedule $\phi = \psi, \psi'$ gives the minimum makespan when ψ is fixed. □

Algorithm LowerBound given bellow is used to estimate lower bound at any given node.

Algorithm LowerBound(θ, θ', ψ)

Step 1. Set fictitious job J_0 with

$p_{0,2} = \sum_{r \in \delta_2 / \{h\}}$ (setup time for family r on M_2) and $p_{0,1} = 0$.

Step 2. Set fictitious job J_{w+1} with

$p_{w+1,1} = \sum_{r \in \delta_1 / \{g\}}$ (setup time for family r on M_1) and $p_{w+1,2} = 0$.

Step 3. Append all jobs $J_i \in \{\theta \backslash \theta'\} \cup \{J_0\} \cup \{J_{w+1}\}$ to ψ according to Johnson order

Step 4. Obtain the makespan of the obtained schedule.

End.

Lemma 2.8. *Consider a node in B&B algorithm with ψ as the partial schedule of jobs $\theta' \subset \theta$. Algorithm LowerBound(θ, θ', ψ) finds a lower bound for makespan for jobs in θ.*

Proof: Let ψ' be the Johnson sequence for job set $\theta \backslash \theta' \cup \{J_0\} \cup \{J_{w+1}\}$. Also let us assume no setup is required for jobs in this set. From Remark 2.7, jobs J_0 and J_{w+1} represent minimum setup requirement for job set $\theta \backslash \theta'$. Further, from Remarks 2.5 and 2.6, combining setups on each machine into corresponding single job will not increase the makespan. Finally, Lemma 2.7 shows that Johnson sequence for any job set which is to be appended to ψ will yield the minimum makespan. Thus, $C_{\max}(\psi, \psi')$ is a lower bound for the given node. $\square$

2.4.2 Dominance rules

In this subsection, we present two dominance rules which provide precedence relations for some jobs. Lemma 2.9 presents the rules for precedence relations for jobs having the same family only on M_1, and Lemma 2.10 presents the rules for jobs having the same family only on M_2.

Lemma 2.9. *Consider a schedule σ and let $f(e,h)$, $f(g,u)$, $f(g,v)$, and $f(g,w)$ be families of $(r-1)^{th}$ job, r^{th} job, $(r+1)^{th}$, and $(r+2)^{th}$ job respectively with $h \neq u \neq v \neq w$. If $\min\{p_{r,1} - s_{r,2}, p_{r+1,2}\} \geq \min\{p_{r+1,1} - s_{r+1,2}, p_{r,2}\}$, then there exists an optimal schedule in which job J_{r+1} immediately precedes job J_r.*

Proof: Without loss of generality, we can assume that $\sigma = \{J_1, J_2, ..., J_n\}$. Let σ' be the schedule when J_{r+1} is moved immediately before J_r in σ. Note $K_i(\sigma') = K_i(\sigma)$ for $i = 1, 2, ..., r-1$.

$$\begin{aligned}
K_{r+1}(\sigma') &= K_{r-1}(\sigma') + p_{r+1,1} + s_{r,1} - p_{r-1,2} - s_{r+1,2} \\
&= K_{r-1}(\sigma) + p_{r+1,1} + s_{r,1} - p_{r-1,2} - s_{r+1,2} \\
&= K_r(\sigma) + (p_{r+1,1} - s_{r+1,2}) - (p_{r,1} - s_{r,2}) \qquad (2.13) \\
&= K_{r+1}(\sigma) - (p_{r,1} - s_{r,2}) + p_{r,2} \qquad (2.14)
\end{aligned}$$

$$\begin{aligned}
K_r(\sigma') &= K_{r+1}(\sigma') + p_{r,1} - p_{r+1,2} - s_{r,2} \\
&= K_r(\sigma) + (p_{r+1,1} - s_{r+1,2}) - p_{r+1,2} \qquad (2.15) \\
&= K_{r+1}(\sigma) + p_{r,2} - p_{r+1,2} \qquad (2.16)
\end{aligned}$$

Case 1: $p_{r+1,1} - s_{r+1,2} \le p_{r,2}$. Considering Lemma's assumption, then $p_{r,1} - s_{r,2} \ge p_{r+1,1} - s_{r+1,2}$ and $p_{r+1,2} \ge p_{r+1,1} - s_{r+1,2}$.

From equation (2.13), $K_{r+1}(\sigma') \le K_r(\sigma)$ and from equation (2.15), $K_r(\sigma') \le K_r(\sigma)$.

Case 2: $p_{r,2} < p_{r+1,1} - s_{r+1,2}$. Then $p_{r,1} - s_{r,2} \ge p_{r,2}$ and $p_{r+1,2} \ge p_{r,2}$.

From equation (2.14), $K_{r+1}(\sigma') \le K_{r+1}(\sigma)$ and from equation (2.16), $K_r(\sigma') \le K_{r+1}(\sigma)$.

$$T_{r+2,1}(\sigma') = \sum_{d=1}^{r-1} (p_{d,1} + s_{d,1} a_{d-1,d}) + s_{r,1} + p_{r,1} + p_{r+1,1} = T_{r+2,1}(\sigma)$$

$$\begin{aligned}
T_{r+2,2}(\sigma') &= \max_{1 \le i \le r+1} \{K_r(\sigma')\} + \sum_{d=1}^{r-1} (p_{d,2} + s_{d,2} b_{d-1,d}) + \\
&\quad s_{r+1,2} + p_{r+1,2} + s_{r,2} + p_{r,2} + s_{r+2,2} \\
&\le \max_{1 \le i \le r+1} \{K_r(\sigma)\} + \sum_{d=1}^{r-1} (p_{d,2} + s_{d,2} b_{d-1,d}) + \\
&\quad s_{r+1,2} + p_{r+1,2} + s_{r,2} + p_{r,2} + s_{r+2,2} \\
&= T_{r+2,2}(\sigma).
\end{aligned}$$

Since $T_{r+2,1}(\sigma') = T_{r+2,1}(\sigma)$ and $T_{r+2,2}(\sigma') \leq T_{r+2,2}(\sigma)$,

then $C_{\max}(\sigma') \leq C_{\max}(\sigma)$. □

Lemma 2.10. *Consider a schedule σ and let $f(c,u)$, $f(d,v)$, $f(e,v)$, and $f(f,v)$ be families of $(r-1)^{th}$ job, r^{th} job, $(r+1)^{th}$, and $(r+2)^{th}$ job respectively with $d \neq e \neq f$. If $\min\{p_{r+1,1} + s_{r+1,1}, p_{r,2}\} \leq \min\{p_{r,1} + s_{r,1}, p_{r+1,2}\}$, then there exists an optimal schedule in which job J_{r+1} immediately precedes job J_r.*

Proof: Without loss of generality, we assume that $\sigma = \{J_1, J_2, ..., J_n\}$. Let σ' be the schedule when J_{r+1} is moved immediately before J_r in σ. Note $K_i(\sigma') = K_i(\sigma)$ for $i = 1, 2, ..., r-1$.

$$\begin{aligned}
K_{r+1}(\sigma') &= K_{r-1}(\sigma') + p_{r+1,1} + s_{r+1,1} - p_{r-1,2} - s_{r+1,2} \\
&= K_{r-1}(\sigma) + p_{r+1,1} + s_{r+1,1} - p_{r-1,2} - s_{r+1,2} \\
&= K_r(\sigma) + (p_{r+1,1} + s_{r+1,1}) - (p_{r,1} + s_{r,1}) \qquad (2.17) \\
&= K_{r+1}(\sigma) - (p_{r,1} + s_{r,1}) + p_{r,2} \qquad (2.18)
\end{aligned}$$

$$\begin{aligned}
K_r(\sigma') &= K_{r+1}(\sigma') + p_{r,1} + s_{r,1} - p_{r+1,2} \\
&= K_r(\sigma) + (p_{r+1,1} + s_{r+1,1}) - p_{r+1,2} \qquad (2.19) \\
&= K_{r+1}(\sigma) + p_{r,2} - p_{r+1,2} \qquad (2.20)
\end{aligned}$$

Case 1: $p_{r+1,1} + s_{r+1,1} \leq p_{r,2}$. Considering Lemma's assumption, then $p_{r+1,1} + s_{r+1,1} \leq p_{r,1} + s_{r,1}$ and $p_{r+1,1} + s_{r+1,1} \leq p_{r+1,2}$.

From equation (2.17), $K_{r+1}(\sigma') \leq K_r(\sigma)$ and from equation (2.19), $K_r(\sigma') \leq K_r(\sigma)$.

Case 2: $p_{r,2} < p_{r+1,1} + s_{r+1,1}$. Then $p_{r,2} \leq p_{r,1} + s_{r,1}$ and $p_{r,2} \leq p_{r+1,2}$.

From equation (2.18), $K_{r+1}(\sigma') \leq K_{r+1}(\sigma)$ and from equation (2.20), $K_r(\sigma') \leq K_{r+1}(\sigma)$.

$$T_{r+2,1}(\sigma') = \sum_{d=1}^{r-1} (p_{d,1} + s_{d,1}a_{d-1,d}) + s_{r+1,1} + p_{r+1,1} + s_{r,1} + p_{r,1} + s_{r+2,1} = T_{r+2,1}(\sigma)$$

$$
\begin{aligned}
T_{r+2,2}(\sigma') &= \max_{1\le i\le r+1}\{K_r(\sigma')\} + \sum_{d=1}^{r-1}(p_{d,2}+s_{d,2}b_{d-1,d}) + s_{r+1,2} + p_{r+1,2} + p_{r,2} \\
&\le \max_{1\le i\le r+1}\{K_r(\sigma)\} + \sum_{d=1}^{r-1}(p_{d,2}+s_{d,2}b_{d-1,d}) + s_{r+1,2} + p_{r+1,2} + p_{r,2} \\
&= T_{r+2,2}(\sigma)
\end{aligned}
$$

Since $T_{r+2,1}(\sigma') = T_{r+2,1}(\sigma)$ and $T_{r+2,2}(\sigma') \le T_{r+2,2}(\sigma)$,

then $C_{\max}(\sigma') \le C_{\max}(\sigma)$. □

We set the makespan of a random schedule as an upper bound in $B\&B$ algorithm. This upper bound is updated whenever the makespan at a leaf is less than the upper bound. A branch is fathomed (i) if the lower bound at the corresponding node is not less than the upper bound, or (ii) if the schedule upto the corresponding node violates any of the two proposed preceding relation requirements.

2.5 Optimal Scheduling

Optimal scheduling of jobs with family setups is simultaneously splitting jobs of the same family into batches, sequencing jobs of each batch, and sequencing ordered batches in an optimal way. Now we present Algorithm OptSchedule to generate optimal schedule.

Algorithm OptSchedule

Step 1. Call Algorithm OptBatch and get the ordered batches β_1,...,β_{w-1}, β_w

Step 2. Generate equivalent job for each ordered batch β_r $(r = 1, 2, ..., w)$

Step 3. Use the Branch and Bound algorithm discussed in Section 2.4 to sequence the generated jobs.

End.

Theorem 2.3. *Algorithm OptSchedule solve two machine flow shop with cross family* *setup optimally in $O(n^{KL})$ time.*

Proof: From Theorem 2.2, and Lemmas 2.3 and 2.4, we know there exists an optimal schedule in which non of the ordered batches generated by Algorithm OptBatch is split into more than one batch. Lemma 2.6 proves that the optimal schedule of equivalent jobs yields the optimal schedule for the ordered batches. Finally, B&B algorithm branches along and does a complete search to generate optimal schedule of the jobs. Therefore, Algorithm OptSchedule finds an optimal schedule.

Step 1 (algorithm OptBatch), requires $O(n^2)$ according to Lemma 2.5, Step 2 requires $O(n)$ time to generate equivalent jobs. In Step 3 (in branch and bound algorithm on generated jobs) sequence of jobs in each family is known. Further, at any node there cannot be more than KL branches. Therefore the complexity of the branch and bound algorithm is $O(n^{KL})$.

Therefore, Algorithm OptSchedule finds optimal schedule in $O(n^{KL})$ time. □

Remark 2.8. *KL is a constant for fixed number of families in each stage, therefore OptSchedule is a polynomial time algorithm.*

Remark 2.9. *Note that Theorem 2.2, and Lemmas 2.9 and 2.10 are true for any two machine flow shop scheduling problem without cross family assumption to minimize the makespan. Thus the traditional problem of two machine flow shop with family setup to minimize makespan, i.e., jobs of the same family on one machine belong to the same family on the other machine, can be solved in polynomial time if the number of families are fixed.*

2.6 Hybrid Genetic Algorithm

Computational time for B&B algorithm exponentially increases with the number of families. Therefore, in this section, we develop a hybrid genetic algorithm (HGA) to solve problems with arbitrary number of families. We call Algorithm OptBatch (presented in Section 2.3) to get initial ordered batches (β_1, β_2,...,β_w) and generate equivalent jobs using Lemma 2.6. Note that the generated jobs, belonging to the same family on both machines, have fixed sequence in the optimal schedule. We use random key genetic algorithm (RKGA) to sequence generated jobs. RKGA introduced by Bean (1994) is convenient for scheduling problems without disturbing the feasibility. In RKGA, random numbers are assigned to jobs and jobs are sequenced in the order of their random numbers.

Representation: In a chromosome, all jobs $J_i \in \tau(f,g)$ are represented by $|\tau(f,g)|$ number of genes starting from $(\sum_{i=1}^{f-1}\sum_{j=1}^{L}|\tau(i,j)| + \sum_{j=1}^{g-1}|\tau(f,j)|+1)^{th}$ gene, in Johnson order. For example, jobs in $\tau(1,1)$ are represented by the first $|\tau(1,1)|$ genes in Johnson order, jobs in $\tau(1,2)$ are represented by the next $|\tau(1,2)|$ genes in Johnson order, and so on.

We briefly describe the major steps in our genetic algorithm:

Initial population: We randomly generate 100 chromosomes for initial population. In section 2.3.1, we have shown that jobs having the same family on both machines follow Johnson sequence in the optimal schedule. We assign random numbers in non-decreasing order to jobs belonging to the same family on both machines to maintain Johnson sequence. Therefore, for each family $\tau(f,g)$ (for $f = 1,2,...,K$ and $g = 1,2,...,L$), we generate $|\tau(f,g)|$ random numbers and assign these random numbers to genes starting from $(\sum_{i=1}^{f-1}\sum_{j=1}^{L}|\tau(i,j)| + \sum_{j=1}^{g-1}|\tau(f,j)|+1)^{th}$ gene in non-decreasing order.

Fitness Evaluation: For each candidate, jobs are sequenced in the order of random numbers and the schedule is obtained by introducing setups when required. Fitness value is the makespan of the schedule.

Reproduction: At each iteration, we select the best 20 candidates of the previous iteration and generate 80 new children to get the population of size 100.

Parent Selection: Candidates in the population are ordered randomly. For each candidate in the population, we subtotal the fitness values starting from the first candidate and divide the subtotal of each candidate by the total fitness value of all candidates. Then two parents (candidates) are randomly selected using roulette wheel method to generate new offsprings.

Crossover: A *two point crossover* method is applied to generate two children. Two distinct integer random numbers ($1 \leq r_1, r_2 \leq N$) are generated to divide each parent into 3 segments S_1, S_2 and S_3, where N is the total number of genes in a chromosome. Then random numbers of jobs of selected parents in segments S_1 and S_3 are exchanged. Exchanging genes may violate Johnson sequence for jobs in families. We make sure Johnson sequence of the jobs for children by reassigning random numbers of jobs having the same family on both machines in non-decreasing order. We tested crossover probabilities of $0.6, 0.7, 0.8$ and 0.9 for 30 problems and found the crossover probability of 0.8 best fits to our problem. Therefore, we set a crossover probability of 0.8.

Mutation: We select two distinct random numbers $1 \leq i, k \leq N$ and exchange the random number of ith and kth genes. Exchanging genes may violate Johnson sequence for jobs in families, thus for generated chromosomes, we make sure Johnson sequence of the jobs by reassigning random numbers of jobs having the same family on both machines in non-decreasing order. We tested mutation probabilities of $0.05, 0.1, 0.15, 0.2$ and 0.25 for 30 problems and found the mutation probability of 0.2 performs well to our problem. Therefore, we set a mutation probability of 0.2.

Random Shake: A random shake is used to avoid eventual traps in local optimal. After 50 iterations, if there is no improvement in the best solution, 100 more chromosomes are randomly generated and added to the current population. The best (H) and the worst ($100 - H$) of this pool of 200 candidates are selected for the new generation, where H is a random number between 50 and 100.

Stopping Criterion: The algorithm will stop when the number of iterations exceed 1000 or when there is no improvement in the best fitness value for 50 iterations after the random shake.

2.7 Computational Experiment

In this section, we describe the computational experiment to test performance of the proposed B&B and HGA algorithms. We coded both algorithms in Visual express C++ and ran on Pinetum 4 personal computer with 2.67 GHz and 1 GB RAM. We studied 9 scenarios with $K = 4, 6, 8$ and $L = 4, 6, 8$. For each scenario, the number of jobs was set at $n = 20, 30, 40, 60, 80$ and 100. For each of these 54 problem instances, 10 random problems with job processing times and family setup times generated from discrete uniform distributions, $U(10, 100)$ and $U(10, 20)$, respectively were tested. The results of our experiment are summarized in Table 2.1.

The B&B algorithm takes very long time to solve large scale problems, and therefore, we set a maximum run time of 3600 seconds for the B&B algorithm. It found optimal solution for problems with upto 40 jobs. Therefore, for problems with $n = 60, 80$ and 100 we selected the best solution in the time limit. HGA algorithm took less than 5 seconds to get schedule for any problem. Column "CPU time" in Table 2.1 shows the CPU time (in seconds) for $B\&B$ algorithm.

For each algorithm, we calculated the percentage deviation $\frac{C\ -C_{LB}}{C_{LB}} \times 100$, where C is the makespan of the best schedule obtained by the algorithms and C_{LB} is the lower bound at the root node. The average and maximum of percentage deviations are provided in Table 2.1. When the number of jobs increases, there is high chance of having more than one batch for jobs in the same families, and as a result a high chance for many setup requirements on machines. However, the lower bound at the root node considers single set up for each family on machines. Thus, the gap between optimal solution and the lower bound increases with increasing number of jobs. This explains the increasing values for percentage deviations with increasing number of jobs in Table 2.1. As it is shown in Table 2.1, average optimality gaps never exceeds 5% for the $B\&B$ and 8% for the HGA algorithm.

In order to evaluate efficiency of the fathoming rules (dominance relations and lower bound), the percentage of average unexplored nodes ($\%AUN$) is calculated. By fathoming each node in the search tree, all its children will be fathomed as well, therefore if a fathoming rule occurs at the top of the tree, it will be more efficient by decreasing the search environment more quickly. For a given processing time distribution, increase in the number of jobs will lead to have more precedence constraints and as a result high chance for branches to be fathomed. Last column in Table 2.1 shows that percentage of unexplored nodes increases with increased number of jobs.

TABLE 2.1. Results of $B\&B$ and HGA Algorithms

Type of Instance			*HGA*		*B&B*			
K	*L*	*# of jobs*	*Avg Gap*	*Max Gap*	*Avg Gap*	*Max Gap*	*CPU time*	*%AUN*
4	4	20	0.43	0.78	0.38	0.73	54	74.05
		30	2.18	3.07	1.43	2.32	146	82.36
		40	1.51	2.57	1.50	2.56	197	84.83
		60	1.62	3.08	1.23	2.69	-	87.6
		80	3.04	5.33	2.43	4.72	-	88.5
		100	4.01	6.73	3.25	5.97	-	90.12
	6	20	2.13	2.66	0.35	0.88	573	82.21
		30	3.02	2.81	2.14	2.14	643	83.16
		40	2.51	4.75	1.84	3.87	837	82.36
		60	4.92	7.78	2.14	5	-	86.32
		80	4.85	8.43	2.74	6.32	-	89.4
		100	5.46	8.69	3.58	6.81	-	91.3
	8	20	2.11	3.10	0.03	1.02	1012	77.42
		30	3.30	3.39	2.14	2.23	1325	84.23
		40	3.52	5.82	2.32	4.62	1465	87.27
		60	4.12	7.88	3.02	6.78	-	88.9
		80	5.20	9.18	4.10	8.08	-	90.2
		100	6.40	11.09	6	10.69	-	92.3
6	4	20	1.6	2.15	0.68	1.23	534	78.74
		30	2.57	2.95	1.48	1.86	675	81.64
		40	4.41	7.16	3.14	5.89	902	83.56
		60	3.78	6.07	2.11	4.40	-	85.6
		80	5.94	9.08	3.39	6.08	-	88.25
		100	6.39	9.5	4.56	7.12	-	90.41
	6	20	2.24	2.9	0.28	0.94	1958	84.69
		30	3.12	3.93	1.12	1.93	2401	87.91
		40	3.17	5.72	2.47	4.81	2632	89.34
		60	4.74	6.88	2.67	5.02	-	91.2
		80	4.20	7.44	3.25	6.49	-	91.9
		100	6.20	10.76	5.56	10.12	-	93.15
	8	20	1.87	1.93	1.30	1.36	2489	89.76
		30	3.65	4.54	2.11	3	2745	83.21
		40	4.12	6.54	3.41	5.83	2804	87.47
		60	4.20	6.27	2.35	4.42	-	89.35
		80	6.32	9.46	5.24	8.38	-	91.3
		100	6.73	11.63	5.45	10.35	-	93.45
8	4	20	1.11	2.10	0.36	1.35	1124	78.54
		30	1.23	2.32	1.97	3.06	1256	83.47
		40	4.02	6.32	2.03	4.33	1580	86.27
		60	4.57	8.33	1.54	5.30	-	88.32
		80	4.45	8.43	2.16	6.14	-	91.58
		100	5.22	9.91	4.22	8.91	-	92.03
	6	20	2.33	2.39	0.16	0.22	2563	91.74
		30	2.71	4.01	0.06	0.95	2648	90.34
		40	3.12	5.13	1.38	3.80	2915	91.41
		60	3.38	6.52	1.16	3.23	-	92.23
		80	4.45	6.85	3.65	6.45	-	92.76
		100	4.70	9.6	4.14	10.37	-	93.4
	8	20	3.11	3.78	1.84	2.06	2987	89.68
		30	4.05	5.03	1.18	2.16	3014	91.87
		40	3.51	5.66	2.33	4.48	3542	92.01
		60	4.62	6.81	2.65	4.84	-	92.87
		80	3.14	7.30	3.85	8.01	-	93.4
		100	7.56	12.76	4.05	10.25	-	93.89

2.8 Conclusions

In this chapter, we studied two machine flow shop scheduling problem with cross families and sequence independent setup time to minimize makespan. The specific assumption embedded to this problem is that each stage has its own job families. We proved Johnson sequence is optimal for jobs belonging to the same family on both machines and developed an efficient branch and bound algorithm. This property of Johnson sequence for jobs belonging to the same family on both machines, is also applicable for past studies on two machine flow shop scheduling problems with family setups to minimize makespan. We also developed a hybrid genetic algorithm using properties of the optimal schedule to solve large scale problems. Computational experiment showed the effectiveness of our algorithms.

Chapter 3

Due date quotation with distinct production capacity in dual channel supply chain

3.1 Motivation and literature review

In this chapter, we study the problem of reliable and immediate due date quotation in a two-echelon dual channel supply chain in order to maximize the total profit. We have one manufacturer and one retailer as the traditional channel and online customers as the e-tail channel. Online customers place the order to the manufacturer, however the products may be delivered to them directly by the manufacturer or through the retail store. In this problem, we try to maximize due-date sensitive profit function by quoting immediate and reliable due dates to the online customers while satisfying capacity constraint and maximum acceptable lead time for online orders.

The problem studied in this chapter has been motivated by the following real-world application. Consider a roll-producer company that produces customized steel-rolls for smaller mills (customers) worldwide which produce different steel products. The customers demand customized orders with different production requirements, thus no inventory is kept by the roll-producer company. It has several production lines, however, producing various orders of rolls require similar technology and therefore, the processing times are almost deterministic. The customers can place their customized orders to the roll-producer online (e-tail); however this company also serves its retail channel (distribution center) in specific cycle times for more predictable customers (demand of the retail channel is almost deterministic). Accepted orders from e-tail channel can be shipped directly to the customers or to the distribution center. In fact, the challenging issue in managing this business is not in the manufacturing side but in the coordination of manufacturing and customer service representatives (CSRs).

When the (online) orders arrive, the CSR decides to accept/reject orders and quotes a due date for accepted ones immediately. For longer due dates it is a common practice to give price discounts just not to lose customers who have other options. CSR used to quote due dates without considering the shop floor status in the past, which led to several problems such as missing quoted due dates, losing customers, increasing overtime production. This problems can be resolved by coordination of CSR and manufacturing side. Our model was motivated to support this situation of coordination. There exist similar situations in automotive supply chains, construction industry, and in the paper industry as well.

With the growth of e-business, many companies are trying to adopt online (e-tail) channel besides their traditional retail stores to provide more convenient access of products for their customers. In dual channel supply chain, the manufacturer uses both traditional retail store and e-tail channel to distribute its products. Firms following this dual-channel strategy are referred to as click-and-mortar companies, which is distinct from their traditional brick-and-mortar counterparts (Yao et al., 2009). Although dual channel may help companies increase their customer's awareness and shopping choices, this type of distribution model affects all business functions and operational decisions. Hill et al. (2002) introduced four main strategies for click-and-mortar companies. In the first strategy, firms separate retail and e-tail channels where each channel has its own warehouse, as well as inventory control and pricing features. Some companies find it difficult to manage the same product in two different channels; therefore as the second strategy, they outsource the e-tail channel to the third party and all the order-fulfilment process is managed by the expert third party firm. Drop-ship is another strategy that some companies apply, in which the third party just picks, packs and delivers the orders to customers in e-tail channel while all the distribution information is available for him. The other strategy used recently is called professional shopper strategy, where customers in e-tail channel order online and then pick up the product from the retail store. In this chapter, the assumption of two delivery options for e-tail customers implies the manufacturer's strategy which is a combination of drop-ship and professional shopper strategy.

Investigating dual channel's impact on company's performance is highly integrated with several related fields, mainly as warehouse design, optimal inventory decisions and pricing. There exist several studies in each field which are briefly reviewed in the following.

Main concerns in warehouse design include; selection of a proper storing method, handling equipment and best warehouse layout (de Koster et al., 2007). Among the studies considering warehouse design, we can refer to Johnson and Meller (2002) who developed analytic performance model of automated split-case sorting system. Russel and Meller (2003) addressed an expressive model of the trade-off between picking and packing efficiencies. Xu (2005) presented a model of a stochastic multi item two-stage inventory system with space constraint, where there exit two regions for e-tailing setting, one assigned to order picking and the other one assigned to reserve stock. Noticing that the largest part of warehouse operating cost accounts for order picking cost, there are several studies considering design and controlling order-picking operations in the warehouses in dual/multi channel situations. Readers are referred to de Koster et al. (2007) for more information. Inventory decisions in multi channel supply chains can be categorized into two streams based on the demand structure (Yao et al., 2009). There are several studies in the first stream where demand in each channel is independent; Alptekinoglu and Tang (2005) studied a multi channel distribution system with stochastic demand to minimize total expected distribution cost. They developed decomposition method to get near optimal solution for their model. Abdul-Jafar et al. (2006) also presented multi-echelon inventory supply chain with one warehouse, multiple retailers and constant demand. In the second stream where total demand splits among the channels, Chiang and Monahan (2005) presented inventory decisions in dual channel supply chain with stochastic demand, and used online preference rate for determining customer portion of e-tail channel. Yao et al. (2005) investigated the impact of information sharing between e-tail and retail channels on inventory related decisions. Network design is one of the main strategic decisions in any supply chain and includes selecting best possible facility locations and designing related transportation network. Several studies in literature addressed quantitative models for multi channel distribution network design. Abdul-Jalbar et al. (2006) studied distribution of a single product to multiple sale locations and compared two fulfilment scenarios to minimize total expected distribution costs. Chaing (2005) developed an inventory model for two-echelon dual channel supply chain in which both traditional retail stores and e-tail customers are served from a central warehouse. Singh et al. (2006)provided an analytical inventory model with stochastic demand, investigated the impact of internet channels on retailer capability of product assortment. Readers are referred to Niels et al. (2008), Yao et al. (2009), Qi et al. (2008) and Abdul-Jalbar et al. (2006) for more information.

Most of e-business failures are related to operational decisions, and one of the main reasons of early e-business failures is ineffective order fulfillment (Tarn et al., 2003). It is accepted that even a well-designed dual channel supply chain is useless when it is not successful to deliver items as promised. Effective order fulfilment is tightly related to accurate due date quotation, and according to Niels et al. (2008) and to the best of our knowledge, there is no study specifically addresses due date quotation in dual channel supply chain. Accurate due date quotation is considered as one of the main performance measures as well as cost and quality (Handfield et al., 1999, Stalk and Hout, 1990), however it is not an easy task; setting relatively soon due dates specifically for make-to-order environments, and scheduling the orders to ensure that they meet the quoted due dates (Kaminsky, 2004) specially when we have unknown demand trend. In fact, capacity constraint makes it impossible to set the ideal due dates, thus the trade-off between sequencing jobs to meet the due dates and setting due date so that sequencing is possible, is the challenging part of these problems. Considering high competitive environment in recent years, dual channel supply chain is vastly increasing and one of the most important challenges for these facilities will be to quote and manage the most efficient due dates to get the competitive advantage in the market. There are several studies in literature considering due date management when there exist only online customers, not applicable for dual channel environments, which are briefly reviewed in the following.

Scheduling coordinated with due date quotation for online orders, was first introduced by Keskinocak et al. (2001) on a single machine. They performed competitive analysis for a specific online algorithm to maximize due-date sensitive revenue. In their problem, it is assumed that there is a threshold on the quoted due dates, and the order will be lost if it is not processed within a specific time interval. Kaminsky and Lee (2008) proposed an online heuristic model for due date quotation problem, minimizing total quoted due dates, and investigated the conditions of asymptotical optimality of suggested algorithm. Zheng et al. (2014) studied the same problem as Keskinocak et al. (2001) evaluating competitive ratio of non-linear revenue functions in both discrete and continues time points. Kapuscinski and Tayur (1997), Duenyas (1995) and Chand and Chhajed (1992) used analytical approaches in due date setting problems without any constraint on the time interval in which the due date should be quoted. There exist several studies on due date setting and sequencing problems, investigating the performance of online algorithms with methods rather than competitive analysis such as simulation, (Baker and Bertrand, 1981, Bookbinder and Noor, 1985, Weeks, 1979, Ragatz and Mabert, 1984). Hsu and Sha (2004) studied online scheduling and due date quotation problem applying artificial neural network in order to minimize delay cost objective, and Chang et al. (2005) proposed a fuzzy modeling method embedded by a genetic algorithm for a due date assignment problem. Kaminsky and Kaya (2005) studied the problem of due date quotation and developed three online heuristics in order to minimize the total processing time. They applied probabilistic approach to investigate asymptotical optimality of suggested heuristics. For a comprehensive review on papers related to due date management, readers are referred to Keskinocak and Tayur (2003) and Cheng and Gupta (1989).

In this chapter, we study the problem of reliable due date quotation in a two-echelon dual channel supply chain to maximize the total profit, while there exists a threshold on due dates, i.e., latest acceptable time for the quoted due dates. The online order will be lost if the quoted due date is after the latest acceptable time. There are two options of delivering items to the online customers; directly from the manufacturer or through the retail store. The objective function includes due-date sensitive revenue function and delivery cost. We consider single type of e-tail customers and adopt the competitive analysis (Borodin and El-Yaniv, 1998) to investigate the performance of the online heuristic algorithms. To the best of our knowledge, this is the first study of due date quotation in dual channel supply chain, considering both e-tail and retail channels. The chapter is organized as follows. Section 3.2 explains the problem with assumptions and section 3.2.1 provides the problem notations and mathematical model. Section 3.3 characterizes the profit function of both online and optimal offline algorithms. Section 3.4 presents a parametric upper bound and lower bound for the competitive ratio of any arbitrary online algorithm using concave fractional programing. Section 3.4.1 proposes specific online algorithm and investigates its performance for single-type e-tail customers. A detailed computational experiment is provided in Section 3.5 and finally, Section 3.6 concludes with a summary of the insights from the analysis.

3.1.1 Solution Procedure

the solution procedure considered in this chapter is developing the mathematical model for maximization of the total profit, finding the bounds of objective function for online and optimal offline algorithms of due date quotation, and then applying competitive analysis to investigate bounds of competitive ratio for any arbitrary online algorithm. We also present specific online algorithm for due date quotation and investigate its corresponding competitive ratio for a worst-case scenario.

3.2 Problem Definition

In this chapter, we study the problem of due date quotation for online customers in a two-echelon dual channel supply chain while maximizing the profit function which contains due-date sensitive revenue and delivery costs of accepted online orders. We have one manufacturer and a retailer as the traditional channel and online customers as the e-tail channel who are served by the manufacturer. There exist two options for delivering products to the online customers i.e., shipping directly from the manufacturer to online customers which is available at any time, and delivering through the retail store which is available at specific periods however imposes less cost to the system. In our model we have the assumption of unknown demand for e-tail channel (at any time we have no idea about future arrivals of online orders) and deterministic demand (with fixed profit) for the traditional channel. This assumption implies that there is an optimal cycle time (T) for delivering items to the retail store, and thus the option of delivering items to online customers through the retail store is available every T periods. There exists maximum acceptable lead time ($L^{'}$) for e-tail customers, and if they are offered a due date after their desired lead time, they will not place the order. From the manufacturer's point of view, in fact the manufacturer has the option of rejecting the online order by offering a due date after the desired lead time when there is no benefit in accepting the order. The term due date in this chapter is referred to the time that the order is shipped to the customer, i.e., the time the order leaves the manufacturer, thus the quoted due date for each order may be different from the time the order production is completed by the manufacturer. Since the delivery option through the retail store is available at specific periods with less cost, the online order may be held by the manufacturer after its production is completed in order to use the most cost-effective delivery method. We assume that the revenue will decrease linearly if the quoted due dates for e-tail customers increase which was first used by Keskinocak and Tayur (2001) (review on non-increasing revenue functions can be found in Keskinocak (1997)). In order to illustrate the revenue in the objective function, let r be the revenue that is lost for each unit of time if the online order waits before being delivered to the customer, and l the time interval between the order's arrival time and its quoted due date, then the revenue will be $r(L^{'} - l)$ (Keskinocak and Tayur, 2001). It is obvious that in this problem the maximum revenue one can obtain from each online order is rL, where $L = L^{'} - p$ and p is the order's production time. In this problem we quote 100% reliable due dates to the online customers, i.e., there is no tardiness cost, and all orders should be delivered by the quoted due dates. We also consider capacity constraint of processing at most N online orders at any time by the manufacturer. We consider a basic model, with single type e-tail customers, i.e., all online orders have unit-length processing time, identical L', revenue and delivery cost parameters.

3.2.1 Mathematical Model

In this sub-section, we introduce the batch definition in our problem, then the notations used in the rest of the chapter are provided followed by the developed mathematical model. Assume that ∂ is the schedule of online orders generated by any algorithm, we can divide each schedule to batches where each batch contains consecutively scheduled online orders. Let s_i be the start time of the batch B_i and s_i' the completion time of the last order in the batch. In batch B_i, the order which is processed at time s_i, has also arrived at s_i and all the accepted online orders arrived before s_i are processed before, however they may leave the system after s_i. Batch definition in this chapter is different from phase definition in Keskinocak and Tayur (2001), as the online orders scheduled in each batch may leave the system after the batch completion time, because in our problem the quoted due date for e-tail orders are the time that the orders leave the system and may be different from their completion time.

If we assume that we have a single type of e-tail customers, our problem of due date quotation for online customers will be reduced to determining how many online orders should be accepted, how many accepted orders should be processed and how many processed orders should be shipped in each period. We use the following notations:

i Time index , $i = 1, 2, ..., n$.

T Optimal cycle time of shipments to the retail store.

π Set of time indices that are multiples of T, $\{T, 2T, 3T, ...\}$.

t_i Time interval between period i and the next period of regular shipment to the retail store.

r Penalty (or revenue that is lost) for each unit of time that the online order waits before being delivered to the customer.

L Maximum acceptable lead time excluding the order's processing time.

c_1 Delivery cost per online order shipped through the retail store.

c_2 Delivery cost per online order shipped directly from the manufacturer ($c_1 < c_2$).

N Maximum number of online orders can be processed at any time by the manufacturer.

d_i Number of arrived online orders (e-tail demand) in period i.

$$\sigma(i) = \begin{cases} 1 & \text{if } i \notin \pi \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables:

q_i Number of accepted online orders in period i.

w_i Number of accepted online orders shifted from period i to $i + 1$ before being processed.

u_i Number of online orders processed in period i but not delivered to the customers (being held).

v_i Number of online orders processed in period i and delivered to the customers.

For a schedule ∂ with n periods, we can define the following mathematical model;

$$Max \sum_{i=1}^{n} C_i, \text{ where } C_i = rLq_i - rw_i - u_i(rt_i + c_1) - v_i[\sigma(i)(c_2 - c_1) + c_1]$$

$s.t.$

$$u_i + v_i \leq N \qquad \forall\ i = 1, 2, ..., n,$$

$$q_i + w_{i-1} - w_i = u_i + v_i \qquad \forall\ i = 1, 2, ..., n,$$

$$q_i \leq d_i \qquad \forall\ i = 1, 2, ..., n,$$

where, the first term of the objective function is the maximum possible revenue one can obtain from any accepted online order. The expression $rw_i - u_i rt_i$ represents the revenue lost for the accepted online orders for each unit of time they spend in the manufacturer's system before being delivered. The terms $u_i c_1$ and $v_i[\sigma(i)(c_2 - c_1) + c_1]$ are the delivery costs of online orders shipped directly by the manufacturer and through the retail store, respectively.

The first set of constraints represent the capacity restriction. The second set of constraints represent that at any time, the number of orders produced or delivered should be equal to the number of accepted orders or the ones which are held to be delivered by the manufacturer for the e-tail channel. The last constraints ensure that number of accepted online orders at any time is less than the online arrivals (e-tail channel demand).

3.3 Preliminaries

In any online algorithm dealing with e-tail customers and unknown demand, making decision about accepting or rejecting the order, also quoting the due date must be done immediately when the order is arrived, while there is no information about the future orders. However, in the offline algorithms, all the information about the orders are available in advance. Mainly online algorithm's performance is evaluated by comparing the results of online and offline algorithms for specific instances. In this section, we study the mathematical model provided in section 3.2.1 for both online and offline models. For a given batch with n periods, let $Z_{(n)}$ denotes the total profit obtained from an online algorithm and $Z^*_{(n)}$ the maximum profit one can obtain from the online arrivals during the batch. First, we present some remarks and propositions to illustrate the features of the problem, and then in Lemma 3.1, we prove that for a given batch with n periods, the lower bound of $Z_{(n)}$ is a linear combination of variables q and u, where q and u are the column vectors of n elements; $q = (q_1, ..., q_n)$, $u = (u_1, ..., u_n)$. Then in Lemma 3.2, considering the offline model for a given batch, an upper bound for optimal function of $Z^*_{(n)}$ is provided. These two lemmas are then used for computing the bounds of the competitive ratio for any arbitrary online algorithm.

Consider the following remarks and propositions for any online algorithm.

Remark 3.1. *We know that any online algorithm tries to schedule orders to be processed as soon as possible to guarantee the available capacity for future online arrivals, as they have no idea about the future e-tail's demand, therefore in each period, if $w_i \geq 1$, it means that we are shifting some orders to be processed in next periods, and in this case we should have used all available capacity in that period, i.e., in online strategies if $w_i \geq 1$ then $u_i + v_i = N$.*

Remark 3.2. *In each period if $u_i > 0$, then $i \notin \pi$ and $\sigma(i) = 1$.*

Remark 3.3. *If $i \in \pi$, then $u_i = 0$ and $\sigma(i) = 0$.*

Remark 3.4. $v_n = \sum_{i=1}^{n}(q_i) - (n-1)N - u_n$ *where n is the last period in a batch.*

Proposition 3.1. *If n is the last period in a batch, then*

$$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n}[(n-i)(q_i)] - N(n(n-1))/2$$

Proof: According to the batch definition, $w_i \geq 1$, for $i = 1, 2, ..., n-1$ and $w_n = 0$. Therefore from Remark 3.1, we have $u_i + v_i = N$ for $i = 1, 2, ..., n-1$, and thus $w_i = \sum_{j=1}^{i}(q_j - N)$ for $i = 1, 2, ..., n-1$ and $w_n = 0$. Accordingly,

$$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n}\sum_{j=1}^{i}(q_j - N) = \sum_{i=1}^{n}[(n-i)(q_i)] - N(n(n-1))/2. \quad \square$$

Lemma 3.1. *Considering any arbitrary online algorithm's schedule with single type of e-tail customers and* $q \geq 0$, *the lower bound of profit function* $Z_{(n)}$ *for a batch with* n *periods can be written as* $r''q + c'u + K$ *where* q, u, r' *and* c' *are the column vectors of* n *elements and* $r''_i = rL - r(n-i) - \sigma(n)(c_2 - c_1) - c_1$, $c'_i = -(r+h)t_i - c_1 + c_2$ *and* $K = rN(n(n-1))/2 + (1-\sigma(n))(N(n-1)(c_1 - c_2))$.

Proof: Assume that we have n periods in batch B_l. According to the batch definition, $w_n = 0$, and since the equality $u_i + v_i = N$ is true for $i = 1, 2, ..., n-1$, we first consider $Z_{(n-1)}$ as the profit function of the first $n-1$ periods in B_l. Let Q and P be the sets $Q = \{i \mid u_i = 0, i \neq n\}$ and $P = \{i \mid u_i > 0, i \neq n\}$. Then,

$$\sum_{i \in Q} C_i = \sum_{i \in Q} \{rLq_i - rw_i - N\ [\sigma(i)(c_2 - c_1) + c_1]\}, \tag{3.1}$$

$$\sum_{i \in P} C_i = \sum_{i \in P} \{rLq_i - rw_i - u_i[rt_i + c_1] - (N - u_i)\ [\sigma(i)(c_2 - c_1) + c_1]\}. \tag{3.2}$$

From Remark 3.2, we have $\sum_{i \in P} C_i = \sum_{i \in P} \{rLq_i - rw_i - u_i[rt_i + c_1] - (N - u_i)\ (c_2)\}$. Therefore,

$$\begin{aligned} Z_{(n-1)} \quad = \quad & \sum_{i \in P} C_i + \sum_{i \in Q} C_i = \sum_{i=1}^{n-1} rLq_i - \sum_{i=1}^{n-1} rw_i - \\ & \sum_{i \in Q} N\ [\sigma(i)(c_2 - c_1) + c_1] - \sum_{i \in P} u_i[rt_i + c_1] - \sum_{i \in P} (N - u_i)\ (c_2). \end{aligned} \tag{3.3}$$

For $i = n$, we have two cases;

Case 1. $n \in \pi$: We have $\sigma(n) = 0$, and from Remark 3.3, $u_n = 0$. Therefore, $C_n = rLq_n - v_n c_1$, and from Remark 3.4, we have $C_n = rLq_n - [\sum_{i=1}^{n}(q_i) - (n-1)N][c_1]$. Therefore,

$$
\begin{aligned}
Z_{(n)} \quad = \quad & Z_{(n-1)} + C_n = Z_{(n-1)} + rLq_n - [\sum_{i=1}^{n}(q_i) - (n-1)N][c_1] = \qquad (3.4)\\
& \sum_{i=1}^{n-1} rLq_i - \sum_{i=1}^{n-1} rw_i - \sum_{i\in Q} N\ [\sigma(i)(c_2 - c_1) + c_1] - \sum_{i\in P} u_i[rt_i + c_1] - \\
& \sum_{i\in P}(N - u_i)(c_2) + rLq_n - [\sum_{i=1}^{n}(q_i) - (n-1)N][c_1].
\end{aligned}
$$

It is clear that we can write $\sum_{i\in P} u_i[rt_i + c_1]$ as $\sum_{i=1}^{n} u_i[rt_i + c_1]$, since if $i \in Q$ then $u_i = 0$ and in this case $u_n = 0$. Also note that $\sum_{i=1}^{n-1} rw_i = \sum_{i=1}^{n} rw_i$ because $w_n = 0$. Thus, we can rewrite the equation (3.4) as

$$
\begin{aligned}
Z_{(n)} \quad = \quad & \sum_{i=1}^{n} rLq_i - \sum_{i=1}^{n} rw_i - \sum_{i\in Q} N\ [\sigma(i)(c_2 - c_1) + c_1] - \sum_{i=1}^{n} u_i[rt_i + c_1] \qquad (3.5)\\
& + \sum_{i=1}^{n}(c_2 u_i) - \sum_{i\in P} Nc_2 - [\sum_{i=1}^{n}(q_i) - (n-1)N][c_1] = \sum_{i=1}^{n} rLq_i - \\
& \sum_{i=1}^{n} rq_i(n-i) + rN(n(n-1))/2 - \sum_{i\in Q} N\ [\sigma(i)(c_2 - c_1) + c_1] - \\
& \sum_{i=1}^{n} u_i[rt_i + c_1 - c_2] - \sum_{i\in P} Nc_2 - [\sum_{i=1}^{n}(q_i) - (n-1)N][c_1]
\end{aligned}
$$

Let $r'_i = rL - r(n-i)$, $c'_i = -rt_i - c_1 + c_2$, $Q_1 = \{i\ \in Q\ \&\ i \in \pi\}$ and $Q_2 = \{i\ \in Q\ \&\ i \notin \pi\}$. Then, the total profit function of the batch in case 1 will be

$$
\begin{aligned}
Z_{(n)} \quad = \quad & r'q + c'u - c_1 q - |P|Nc_2 - |Q_1|N\ c_1 - |Q_2|Nc_2 + \qquad (3.6)\\
& rN(n(n-1))/2 + N(n-1)c_1,
\end{aligned}
$$

where q, u, r' and c' are the column vectors of n elements; $q = (q_1, ..., q_n)$, $u = (u_1, ..., u_n)$, $r' = (r'_1, ..., r'_n)$, $c' = (c'_1, ..., c'_n)$.

Case 2. $n \notin \pi$: In this case $\sigma(n) = 1$ and $C_n = rLq_n - u_n(rt_n + c_1) - v_n c_2$. From Remark 3.4, we have $C_n = rLq_n - u_n(rt_n + c_1) - [\sum_{i=1}^{n}(q_i) - (n-1)N - u_n]c_2$. Therefore,

$$\begin{aligned} Z_{(n)} &= Z_{(n-1)} + C_n = \qquad (3.7) \\ & Z_{(n-1)} + rLq_n - u_n(rt_n + c_1) - [\sum_{i=1}^{n}(q_i) - (n-1)N) - u_n]c_2 = \\ & \sum_{i=1}^{n-1} rLq_i - \sum_{i=1}^{n-1} rw_i - \sum_{i \in Q} N\ [\sigma(i)(c_2 - c_1) + c_1] - \sum_{i \in P} u_i[rt_i + c_1] + \\ & \sum_{i \in P}(u_i - N)(c_2) + rLq_n - u_n(rt_n + c_1) - [\sum_{i=1}^{n}(q_i) - (n-1)N) - u_n]c_2 \end{aligned}$$

Setting r_i', c_i', Q_1 and Q_2 as in case 1, then the total profit function of the batch for case 2 will be as

$$\begin{aligned} Z_{(n)} &= r'q + c'u - c_2 q - |P|Nc_2 - |Q_1|N\ c_1 - |Q_2|Nc_2 + \qquad (3.8) \\ & rN(n(n-1))/2 + N(n-1)c_2. \end{aligned}$$

Note that $(n-1) = |P| + |Q_1| + |Q_2|$ and $c_1 < c_2$. In either case 1 or 2, we can replace the expression $(-|P|Nc_2 - |Q_1|N\ c_1 - |Q_2|Nc_2)$ by $(-|P|Nc_2 - |Q_1|N\ c_2 - |Q_2|Nc_2)$ and determine the lower bound of $Z_{(n)}$ as follow;

$$Z_{(n)} \geq \begin{cases} r'q + c'u - c_1 q + rN(n(n-1))/2 + N(n-1)(c_1 - c_2) & n \in \pi \\ r'q + c'u - c_2 q + rN(n(n-1))/2 & n \notin \pi. \end{cases} \qquad (3.9)$$

Therefore, for a given batch in an arbitrary online algorithm's schedule, we have $Z_{(n)} \geq r''q + c'u + K$, where $r_i'' = r_i' - \sigma(n)(c_2 - c_1) - c_1$ and

$K = rN(n(n-1))/2 + (1 - \sigma(n))(N(n-1)(c_1 - c_2))$. $\square$

Lemma 3.2. *The maximum profit one can obtain from the online arrivals during the given batch, i.e.,* $Z^*_{(n)}$ *(profit of optimal offline algorithm) has the following upper bound, where* q'_i *is the number of accepted online orders in period* i *by an optimal offline algorithm and* q' *is the column vectors of* n *elements;* $q' = (q'_1, ..., q'_n)$.

$$Z^*_{(n)} \leq (rL - c_1)q' \tag{3.10}$$

Proof: Assume that we have n periods in batch B_l. For the periods $\{i = 1, ..., n \mid u_i = 0\}$, if $i \in \pi$, delivery cost for each of the online orders in period i is c_1 and if $i \notin \pi$, delivery cost is c_2, thus considering c_1 as the delivery cost for all orders in these periods $\{i = 1, ..., n \mid u_i = 0\}$ does not reduce the profit function since $c_2 > c_1$. For periods $\{i = 1, ..., n \mid u_i > 0\}$, delivery cost is c_1, but we have also the cost rt_i since $i \notin \pi$ and $t_i > 0$. Therefore, in this case considering c_1 as the total delivery and holding cost for each of the online orders in these periods do not decrease the profit function as well. According to the batch definition, we know that $w_i \geq 1$ for $i = 1, 2, ..., n-1$ and $w_n = 0$, thus $rw \geq 0$. Therefore, based on the objective function in section 3.2.1, it is clear that maximum profit one can make from the online arrivals during a given batch has the following upper bound, $Z^*_{(n)} \leq (rL - c_1)q'$. □

3.4 Competitive Analysis

In online optimization problems, online algorithm's performance is mainly evaluated via the competitive analysis, comparing an online algorithm's result with the offline model's optimal solution. For the problem in this chapter, all the information about the online orders are available in advance for the optimal offline algorithm obtaining maximum possible profit. Given an instance I, let $Z_{(I)}$ denote the total profit obtained by using an online algorithm, and $Z^*_{(I)}$ denote the maximum profit obtained by an optimal offline algorithm. For maximization problems, the online algorithm is called $\rho-$competitive if $Z^*_{(I)} \leq \rho Z_{(I)} + b$ where $\rho \geq 1$, and b is a constant. We define competitive ratio as $\rho = sup\rho(\frac{Z^*_{(I)}}{Z_{(I)}})$ for $Z_{(I)} > 0$. Determining the bounds of competitive ratio (ρ) is the main issue and also the challenging part of online optimization problems.

According to the batch definition provided in section 3.2.1, any schedule of online orders generated by an online algorithm can be divided into batches. Therefore, if we investigate the competitive ratio of a given batch, we can generalize the results to determine the competitive performance of the corresponding online algorithm. In this section, we first investigate the competitive ratio of any arbitrary online algorithm (section 3.4.1) and then the parametric bounds of competitive ratio for a specific online strategy are provided (section 3.4.2).

3.4.1 Competitive ratio of any arbitrary online algorithm

In this subsection, we review the concave fractional programming (proposition 3.2) which is used in Lemma 3.3 to prove an upper bound for the competitive ratio of any arbitrary online algorithm ($q \geq 0$). The lower bound for the ratio is also provided in Lemma 3.4.

Proposition 3.2. *Concave Fractional Programming. If $x \in C$, $C \subset R^n$ is a convex set, f is a concave and non-negative function on C and g is a positive and convex function on C, then the optimization problem $\max_{x \in C} \frac{f(x)}{g(x)}$ is equivalent to the following problem*

$$\begin{aligned}
&\min \ \lambda \\
&s.t. \\
&-\nabla f(x) + \lambda \nabla g(x) = 0 \\
&-f(x) + \lambda g(x) \geq 0 \\
&x \in C \\
&\lambda \geq 0
\end{aligned}$$

Proof: The proposition's proof and more general results on concave fractional programming can be found in Avriel et al. (1988). □

In Lemma 3.3, we provide a parametric upper bound for the competitive ratio of any online algorithm using concave fractional programming.

Lemma 3.3. *For an arbitrary online algorithm with single type of e-tail customers, if a finite competitive ratio exists, it satisfies* $\rho \leq \frac{(rL-c_1)}{nr+r'_{\min}-c_2-\frac{r}{2}}$*, where* $r'_{\min} = \min\limits_i\{rL - r(n-i)\}$.

Proof: According to the competitive analysis description, competitive ratio is defined as $\rho = sup\rho(\frac{Z^*_{(I)}}{Z_{(I)}})$ for a given instance I. In order to find ρ, we can solve the optimization problem of max $\frac{Z^*_{(I)}}{Z_{(I)}}$.

According to Lemma 3.1, for each batch with n periods, $Z_{(n)} \geq r''q + c'u + K$, where c' is a column vector of n elements and $c'_i = -rt_i - c_1 + c_2$. Note that $\forall\ i$, if $c'_i \geq 0$ then $u_i \geq 0$, and if $c'_i < 0$, the option of delivering items through the retail store is not cost-effective in any situation, so $u_i = 0$, therefore, $c'u \geq 0$ and we have $Z_{(n)} \geq r''q + K$. Also based on Lemma 3.2, for each batch, we have $Z^*_{(n)} \leq (rL - c_1)q'$. Then it is obvious that the inequality of $\frac{Z^*_{(n)}}{Z_{(n)}} \leq \frac{(rL-c_1)q'}{r''q+K}$ is satisfied and thus $\rho \leq \max(\frac{(rL-c_1)q'}{r''q+K})$.

By the batch definition, we know that $\sum\limits_{i=1}^{n} q_i \geq (n-1)N+1$ and let $r'_{\min} = \min\limits_i\{r'_i\}$. From Lemma 3.1, we have

$$Z_{(n)} \geq \begin{cases} (r'_{\min} - c_1)((n-1)N+1) + rN(n(n-1))/2 + N(n-1)(c_1 - c_2) & n \in \pi \\ (r'_{\min} - c_2)((n-1)N+1) + rN(n(n-1))/2 & n \notin \pi \end{cases} \tag{3.11}$$

As $N(n-1)(c_1 - c_2) - c_1((n-1)N+1) \geq -c_2((n-1)N+1)$, therefore, $Z_{(n)} \geq (r'_{\min} - c_2)((n-1)N+1) + rN(n(n-1))/2$. By rearranging the inequality we have,

$$\begin{aligned} Z_{(n)} \geq\ & (n-1)N(r'_{\min} - c_2 + \frac{nr}{2}) + (r'_{\min} - c_2) \geq \\ & \frac{Nrn^2}{2} + n(Nr'_{\min} - Nc_2 - \frac{Nr}{2}) + Nc_2 - r'_{\min}(N-1) - c_2 \end{aligned} \tag{3.12}$$

Note that $\sum\limits_{i=1}^{n} q'_i \geq \sum\limits_{i=1}^{n} q_i$ and considering the batch definition, the maximum possible number of orders accepted from arrivals during the batch can be at most NL

orders more than the accepted ones by any online algorithm, i.e., $\sum_{i=1}^{n} q'_i \leq \sum_{i=1}^{n} q_i + NL$. Also we know that $\sum_{i=1}^{n} q_i \leq nN$. Therefore, $\sum_{i=1}^{n} q'_i \leq \sum_{i=1}^{n} q_i + NL \leq Nn + NL$, and $Z^*_{(n)} \leq (rL - c_1)q' \leq (rL - c_1)(Nn + NL)$. Based on Proposition 3.2, optimization problem ($\rho \leq \max(\frac{(rL-c_1)q'}{r''q+K})$) can be written as the following dual model if $Z^*_{(n)}$ and $Z_{(n)}$ are concave and convex functions, respectively.

$$\begin{aligned} &\min \ \lambda \\ &s.t. \\ &-\nabla Z^*_{(n)} + \ \lambda \nabla Z_{(n)} = 0 \\ &-Z^*_{(n)} + \ \lambda Z_{(n)} \geq 0 \\ &\lambda \geq 0 \end{aligned}$$

Note that $\frac{d^2}{dn^2}(Z^*_{(n)}) = 0$ and $\frac{d^2}{dn^2}(Z_{(n)}) = Nr$, (the corresponding conditions are satisfied). Therefore, if a finite ratio ρ exists, i.e., there would be a feasible solution for the above dual model, and we have $\rho \leq \lambda = \frac{\nabla Z^*_{(n)}}{\nabla Z_{(n)}} = \frac{N(rL-c_1)}{Nnr+Nr'_{\min}-Nc_2-\frac{Nr}{2}} = \frac{(rL-c_1)}{nr+r'_{\min}-c_2-\frac{r}{2}}$. □

In Lemma 3.4, we provide a parametric lower bound for the competitive ratio of any arbitrary online algorithm.

Lemma 3.4. *For any arbitrary online algorithm with single type of e-tail customers, the lower bound of the competitive ratio is* $\rho \geq \frac{(L+1)/2-L/k_2}{1-1/k_2} \geq 1.5 - \frac{1}{k_2}$ *where* $k_2 = \frac{rL}{c_2}$ *and* $k_2 \geq L \geq 2$.

Proof: In order to find the lower bound of competitive ratio for any online algorithm, we adapt the rule that at any time, the adversary knows all the actions of online algorithm and provides the worst possible arrivals of e-tail customers as an input to maximize the competitive ratio. Based on this rule, at any time, if the algorithm decides to accept even one of the online arrivals to be processed at time t, we will have NL number of new online arrivals in each period afterwards till period t. At any time, if the algorithm decides to reserve the capacity for period t by rejecting the available orders and using future arrivals for period t, there would be no more online arrivals afterwards. Note that if the algorithm decides to accept all the NL

possible orders, it implies that $r \geq c_2$ (which is the worst case and is considered in this Lemma), otherwise, at any time only the number of orders that guarantees the profitability will be accepted.

Assume that in each T-period, the last x orders have the option of delivery through the retail store, and we know that their delivery cost is $c_i^{'} = (r)\left\lfloor \frac{i}{N} \right\rfloor + c_1$ for $i = 1, ..., x$ where $\left\lfloor \frac{i}{N} \right\rfloor$ is the greatest integer which is less than $\frac{i}{N}$. This cost is replaced by c_1 for all x orders in the online profit function, where $c_1 \leq \min_i(c_i^{'})$ and is replaced by $c^{'}$for all x orders in the optimal offline profit function, where $c^{'} = \max_i(c_i^{'}) = (r)\frac{x}{N} + c_1$.

Let at $t = 0$, the online algorithm decides to reserve the capacity for time $t \geq 1$ by rejecting the available e-tail orders and using new arrivals. Based on the adversary rule, there would be no more arrivals after $t = 0$, and thus batch length is $n = 1$. In this case the online profit is $Z_{(n)} \leq NrL - \left\lfloor \frac{1}{T} \right\rfloor x(c_1) - (N - \left\lfloor \frac{1}{T} \right\rfloor x)c_2$. However, the maximum profit one can gain from the arrivals during the batch period (arrivals at $t = 0$) is $Z_{(n)}^{*} = (\frac{NL(L+1)}{2})r - \left\lfloor \frac{L}{T} \right\rfloor xc^{'} - (NL - \left\lfloor \frac{L}{T} \right\rfloor x)c_2$. At $t = 0$, the maximum possible orders that offline algorithm can accept is NL. Therefore, the revenue gained from these accepted orders is $NL + N(L-1) + ... + N(1) = \frac{NL(L+1)}{2}$. In this case, the length of consecutively scheduled orders will be L. The term $\left\lfloor \frac{L}{T} \right\rfloor x$ determines the number of orders that have been delivered through the retail store in each T-period and $(NL - \left\lfloor \frac{L}{T} \right\rfloor x)$ represent the rest of accepted orders shipped directly to the online customers.

If the batch ends at $t \geq 1$, it is clear that in this case, batch length is $n = t+1$ and the maximum revenue one can gain from online algorithm is $NrL+Nt(r)(L-1)$, where NrL is for the first period and $Nt(r)(L-1)$ denotes the maximum possible revenue for the next t periods. Therefore, $Z_{(n)} = NrL - Nc_2 + Nt(r(L-1) - c_2) + \left\lfloor \frac{1+t}{T} \right\rfloor x(c_2 - c_1)$. For this case, the maximum number of orders that the offline algorithm can accept from the arrivals during the batch length is $N(t) + NL$. In the first t periods that we have arrivals the revenue will be Nrt, and in the last period that we have any arrivals, we will accept the maximum number which is NL, where its revenue will be $NL + N(L-1) + ... + N(1) = \frac{NL(L+1)}{2}$. So the partial schedule has $t + L$ periods and the maximum possible profit one can gain is $Z_{(n)}^{*} = (\frac{NL(L+1)}{2})r - NLc_2 + Nt(Lr -$

$c_2) + \lfloor \frac{L+t}{T} \rfloor x(c_2 - c')$.

It is clear that when t and N increase the ratio $Z^*_{(n)}/Z_{(n)}$ increases and when T increases the ratio decreases. Therefore, the minimum amount of the ratio occurs when $t = 0$, $N = 1$ and T equals to infinity, and the lower bound of the competitive ratio for any online algorithm is $\frac{Z^*_{(n)}}{Z_{(n)}} \geq \frac{rL(L+1)/2 - Lc_2}{rL - c_2}$. Let $c_1 \times k_1 = rL$ and $c_2 \times k_2 = rL$ where $k_1 > k_2 > 1$, then $\frac{Z^*_{(n)}}{Z_{(n)}} \geq \frac{(L+1)/2 - L/k_2}{1 - 1/k_2}$. Note that we assumed that if the algorithm decides to accept orders at any time, all the possible NL arrivals may be accepted, therefore we should have $r \geq c_2$. Thus $r \geq \frac{rL}{k_2}$, $k_2 \geq L$. In this situation, $\frac{Z^*_{(n)}}{Z_{(n)}}$ has the minimum amount at $L = 2$, and $\frac{Z^*_{(n)}}{Z_{(n)}} \geq \frac{(3)/2 - 2/k_2}{1 - 1/k_2} \geq 1.5 - \frac{1}{k_2}$ where $k_2 \geq L \geq 2$. $\square$

3.4.2 Algorithm of Due Date Quotation for Online Customers (DQC)

In this subsection, we introduce a specific online algorithm for single type of e-tail customers, called DQC and we investigate its corresponding competitive ratio. Select $0 < \alpha < 1$, among the orders that arrive at time t, the ones that yield at least $\alpha(rL - c_1)$ profit are accepted and others will be rejected. The accepted orders will be scheduled at the earliest possible position. Note that $(rL - c_1)$ represents the maximum possible profit yield from any accepted order, which includes the maximum possible revenue (rL) and the minimum delivery cost (c_1), gained from delivery through the retail store without holding the order. In fact, in this algorithm, an online order is accepted if a certain fraction of maximum profit is guaranteed and the rest arrived orders are rejected to keep the capacity for the later orders that may yield more profit. The main idea of this algorithm gained from the algorithm presented by Keskinocak and Tayur (2001) for the problem of revenue maximization, however there exist influencing differences in details and assumptions.

Lemma 3.5. *The competitive ratio of algorithm (DQC) is at most* $\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}$ *where* $\alpha'' = \alpha(1 - \frac{1}{k_1}) + \frac{1}{k_2}$ *and* α *satisfies the following equation.*

$$\frac{(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2)}{(1 - \alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}} = \frac{(rL - c_1)}{\alpha(rL - c_1) + \frac{1}{T}(c_2 - c_1)}. \tag{3.13}$$

Proof: For the batch B_l with n periods, let $Z_{(n)}$ be the profit obtained from DQC algorithm and $Z^*_{(n)}$ the maximum possible profit one can gain from the arrivals during the batch. Note that all the accepted orders by algorithm DQC yield at least $\alpha(rL - c_1)$ profit. Assume R is the revenue obtained from an accepted online order by DQC, and in worst case it is delivered directly to the customer, then $R - c_2 \geq \alpha(rL - c_1)$. Let $\alpha' = \frac{c_2 - \alpha c_1}{rL}$, then $R \geq \alpha rL + \alpha' rL$, and any accepted order by algorithm DQC yield at least $\alpha'' rL$ revenue where $\alpha'' = \alpha + \alpha'$ and $0 < \alpha'' < 1$. Note that for determining the bounds of competitive ratio, we have to consider the worst-case situation. First assume that $n \geq \lfloor (1-\alpha'')L \rfloor + 1$, the revenue we can get from DQC algorithm is at least

$$\begin{aligned} Z_{(n)} &\geq rN\left(\frac{L(L+1)}{2} - \frac{\lfloor \alpha'' L \rfloor \left(\lfloor \alpha'' L \rfloor + 1\right)}{2}\right) + (n - \lfloor (1-\alpha'')L \rfloor - 1))\alpha'' rLN - \\ &\quad nNc_2 + \left\lfloor \frac{n}{T} \right\rfloor N(c_2 - c_1) \\ &\geq rN\left(\frac{L(L+1)}{2} - \frac{\alpha'' L(\alpha'' L + 1)}{2}\right) + (n - (1-\alpha'')L - 1))\alpha'' rLN - \\ &\quad nNc_2 + \left(\frac{n}{T} - 1\right)N(c_2 - c_1). \end{aligned} \tag{3.14}$$

We assume that all the orders scheduled in the first $\lfloor (1-\alpha'')L \rfloor + 1$ periods have been arrived at $t = 1$ (the worst-case situation), therefore the revenue DQC algorithm can gain from these orders will be $rN(\frac{L(L+1)}{2} - \frac{\lfloor \alpha'' L \rfloor (\lfloor \alpha'' L \rfloor + 1)}{2}) \geq rN(\frac{L(L+1)}{2} - \frac{\alpha'' L(\alpha'' L+1)}{2})$, which is the first term in right-hand side of equation (3.14). $\alpha'' rL$ is the minimum revenue DQC algorithm can get from the remaining periods $(n - \lfloor (1-\alpha'')L \rfloor - 1))$ in batch B_l, thus we have $(n - (1-\alpha'')L - 1))\alpha'' rLN$ as well. The expression $-nNc_2 + \left\lfloor \frac{n}{T} \right\rfloor N(c_2 - c_1)$ denotes the maximum delivery cost for all the orders scheduled in the batch, where we know that $\left\lfloor \frac{n}{T} \right\rfloor \geq \frac{n}{T} - 1$, and therefore $\left\lfloor \frac{n}{T} \right\rfloor N(c_2 - c_1)$ is replaced by $(\frac{n}{T} - 1)N(c_2 - c_1)$ in equation (3.14). Without loss of generality, let $c_1 \times k_1 = rL$ and $c_2 \times k_2 = rL$ where $k_1 > k_2 > 1$, then $-N(c_2 - c_1)$ in equation (3.14) can be replaced by $\frac{NrL^2}{2}(\frac{-2}{Lk_2} + \frac{2}{Lk_1})$. By rearranging the equation (3.14), we have

$$
\begin{aligned}
Z_{(n)} \quad \geq \quad & \frac{NrL^2}{2}((1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}) + \\
& nNr\alpha''L - nNc_2 + \frac{n}{T}N(c_2 - c_1). \qquad (3.15)
\end{aligned}
$$

Note that $\alpha''rL = \alpha rL + c_2 - \alpha c_1$, therefore the term $nNr\alpha''L - nNc_2$ in equation (3.15) is equal to $nN\alpha(rL - c_1)$, and we have

$$
Z_{(n)} \geq \frac{NrL^2}{2}((1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}) + nN\alpha(rL - c_1) + \frac{n}{T}N(c_2 - c_1). \quad (3.16)
$$

The maximum profit we can obtain from the arrivals during the batch B_l (for the considered worst-case situation) is as follows;

$$
\begin{aligned}
Z^*_{(n)} \quad \leq \quad & (n - \lfloor (1-\alpha'')L \rfloor))NrL + Nr(\frac{L(L-1)}{2}) - \\
& nNc_1 - \lfloor \alpha'' L \rfloor Nc_1 - rN\lceil \frac{n}{T} \rceil (1 + 2 + ... + x). \qquad (3.17)
\end{aligned}
$$

Considering the batch definition, capacity constraint N, and the worst-case situation mentioned above, there would be no arrivals during the last $\lfloor (1-\alpha'')L \rfloor$ periods of batch B_l. The first term in the right-hand side of equation (3.17) shows that all the orders arrived in periods $1, 2, ..., n - \lfloor (1-\alpha'')L \rfloor$ can obtain maximum amount of revenue which is rL, and the second term denotes that the maximum possible revenue one can get from the arrivals during the last $\lfloor (1-\alpha'')L \rfloor$ periods of the batch is $Nr(\frac{L(L-1)}{2})$. The last period in the batch in which we could have any online arrival is the period $(n - \lfloor (1-\alpha'')L \rfloor)^{th}$, and in order to obtain maximum possible revenue, we assume the maximum number of orders we can accept (NL) have arrived in that period. Therefore, the maximum possible revenue for the arrivals during the last $\lfloor (1-\alpha'')L \rfloor$ periods is $Nr(L-1) + Nr(L-2) + ... + Nr(1) = Nr(\frac{L(L-1)}{2})$. The terms nNc_1 and $\lfloor \alpha'' L \rfloor Nc_1$ are also the minimum delivery costs for all online arrivals during the batch time. Some of the orders delivered through the retail store may have been held after their completion time, and it is obvious that their holding time would be $rN\lceil \frac{n}{T} \rceil(1+2+...+x)$, where $x = \frac{c_2-c_1}{r} = (\frac{1}{k_2} - \frac{1}{k_1})L$.

Note that $-\lfloor \alpha'' L \rfloor Nc_1 \leq (1-\alpha'' L)Nc_1 = \frac{NrL^2}{2}(\frac{2}{Lk_1} - \frac{2\alpha''}{k_1})$, and $-rN\lceil \frac{n}{T} \rceil \frac{x(x+1)}{2}$ $\leq -rN\frac{x^2}{2} = -\frac{NrL^2}{2}(\frac{1}{k_2} - \frac{1}{k_1})^2$. By rearranging equation (3.17), we have

$$Z^*_{(n)} \leq \frac{NrL^2}{2}(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2) + nN(rL - c_1). \qquad (3.18)$$

Considering the right-hand side of equations (3.16) and (3.18), If we set

$$\frac{(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2)}{(1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}} = \frac{(rL - c_1)}{\alpha(rL - c_1) + \frac{1}{T}(c_2 - c_1)}, \qquad (3.19)$$

we have $Z^*_{(n)} \leq (\frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)})Z_{(n)}$. Note that $\frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)} = \frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}$, then $Z^*_{(n)} \leq (\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})})Z_{(n)}$, where α is obtained from the quadratic equation (3.20). By replacing $\alpha'' = \alpha + \frac{c_2-\alpha c_1}{rL} = \alpha(1-\frac{1}{k_1}) + \frac{1}{k_2}$ in equation 3.19 , we have

$$\alpha^2(a) + \alpha(b) + (c) = 0 \qquad (3.20)$$

$$a = (1-\frac{1}{k_1})^3 - 2(1-\frac{1}{k_1})^2 + \frac{2}{k_1}(1-\frac{1}{k_1})^2$$

$$\begin{aligned} b \;=\; & -2(1-\frac{1}{k_1})(1-\frac{1}{k_2})^2 - \frac{3}{L}(1-\frac{1}{k_1})^2 - \frac{2}{k_1}(1-\frac{1}{k_1}) + \\ & (1-\frac{1}{k_1}) - \frac{1}{L}(1-\frac{1}{k_1}) - \frac{2}{Lk_1}(1-\frac{1}{k_1}) + \frac{2}{k_2k_1}(1-\frac{1}{k_1}) + \\ & (1-\frac{1}{k_1})(\frac{1}{k_2}-\frac{1}{k_1})^2 - \frac{2}{T}(1-\frac{1}{k_1})(\frac{1}{k_2}-\frac{1}{k_1}) + \frac{2}{k_1T}(1-\frac{1}{k_1})(\frac{1}{k_2}-\frac{1}{k_1}) \end{aligned}$$

$$\begin{aligned} c &= (1-\frac{1}{k_1})(1-\frac{1}{k_2})^2+\frac{1}{L}(1-\frac{1}{k_1})-\frac{5}{Lk_2}(1-\frac{1}{k_1})+ \\ &\quad \frac{2}{Lk_1}(1-\frac{1}{k_1})-\frac{2}{Tk_1}(\frac{1}{k_2}-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})-\frac{2}{TL}(\frac{1}{k_2}-\frac{1}{k_1})- \\ &\quad \frac{2}{LTk_1}(\frac{1}{k_2}-\frac{1}{k_1})+\frac{2}{Tk_2k_1}(\frac{1}{k_2}-\frac{1}{k_1})+\frac{2}{T}(\frac{1}{k_2}-\frac{1}{k_1})^3. \end{aligned}$$

It is obvious that $\frac{1}{k_2}-\frac{1}{k_1}\geq 0$, then $\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}\leq\frac{1}{\alpha}$, and $Z^*_{(n)}\leq\frac{1}{\alpha}Z_{(n)}$ where α is obtained from equation (3.20).

Note that if $n<\lfloor(1-\alpha'')L\rfloor+1$, then the online algorithm will accept all possible arrivals during the batch length and schedule them in a non-decreasing order of their arrival times, because any arrival during the $\lfloor(1-\alpha'')L\rfloor+1$ periods yield at least $\alpha''rL$ revenue. It can be simply shown that the online algorithm in this case gives an optimal solution, i.e., $\sum_{i=1}^{n}q'_i=\sum_{i=1}^{n}q_i$ and $Z^*_{(n)}=Z_{(n)}$. □

Corollary 3.1. *In Lemma 3.5, we assumed we have the option of holding the completed orders to be delivered through the retail store, which implies that* $c_2>r+c_1$*. However, we may have two other cases; (*$c_2>r$ *&* $c_2<r+c_1$*) and (* $c_2<r$*, i.e.,* $k_2>L$*), where holding the completed item is not profitable, and completed orders may be delivered through the retail store only if their completion time is at* π *set. For these two cases,* $Z^*_{(n)}$ *in Lemma 3.5 changes to* $Z^*_{(n)}\leq\frac{NrL^2}{2}(2\alpha''-1+\frac{1}{L}+\frac{2}{Lk_1}-\frac{2\alpha''}{k_1})+nN(rL-c_1)$*, and we have* $Z^*_{(n)}\leq\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}Z_{(n)}$ *, where* α *is obtained from equation* $\frac{(2\alpha''-1+\frac{1}{L}+\frac{2}{Lk_1}-\frac{2\alpha''}{k_1})}{(1-\alpha'')^2+\frac{(1-3\alpha'')}{L}-\frac{2}{Lk_2}+\frac{2}{Lk_1}}=\frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)}$.

Corollary 3.2. *Note that based on the results obtained from Lemma 3.3, the upper bound of competitive ratio for any arbitrary online algorithm is* $\rho\leq\frac{(rL-c_1)}{nr+r'_{\min}-c_2-\frac{r}{2}}$*. In Algorithm DQC,* $r'_{\min}=\alpha''rL$*, thus* $\rho\leq\frac{(rL-c_1)}{nr+\alpha''rL-c_2-\frac{r}{2}}=\frac{(1-\frac{1}{k_1})}{\frac{n}{L}+\alpha(1-\frac{1}{k_1})+\frac{1}{k_2}-\frac{1}{k_2}-\frac{1}{2L}}=\frac{(1-\frac{1}{k_1})}{(\frac{n}{L}-\frac{1}{2L})+\alpha(1-\frac{1}{k_1})}$*, and it is clear that* $\rho\leq\frac{1}{\alpha}$*. Therefore, results of both Lemmas 3.5 and 3.3 satisfies the inequality of* $Z^*_{(n)}\leq\frac{1}{\alpha}Z_{(n)}$ *for DQC online algorithm.*

3.5 Experimental Results

In order to evaluate the competitive performance of the proposed algorithm, we define three different cases, 1) $c_2 > r + c_1$ 2) $c_2 > r$ & $c_2 < r + c_1$ 3) $c_2 < r$, i.e., $k_2 > L$. We investigate DQC algorithm's performance by providing computational experiments on its competitive ratio upper-bound (The bounds in Lemma 3.5 and Corollary 1). Note that we may use different scenarios for each case, and in each scenario, the maximum possible ratio is reported since for our performance evaluation the upper bound of the ratio is required. For all the three cases, experimental analysis denoted that by increasing k_1 while other parameters are fixed, the algorithm DQC's ratio increases, thus the maximum amount of ratio occurs with the minimum amount of delivery cost c_1. In addition, by increasing T, the competitive ratio decreases (α increases) and it implies that the maximum amount of ratio occurs when T has the minimum amount. This result also satisfies the argument in Lemma 3.4, which denoted the minimum ratio for any online algorithm occurs when T goes to infinity. For each case, the competitive ratio is calculated for different amounts of L and k_2 while k_1 and T are set to be 10000 and 2, respectively. (Note that $T = 1$ is not considered for this problem, as it eliminates the second option of direct delivery and $k_1 = 10000$ examined and proved to be large enough for the analysis). The results are provided in Table 3.1. For case 1, we have $c_2 > r + c_1$ which means $\frac{1}{k_2} > \frac{1}{k_1} + \frac{1}{L}$, thus for a specific L, the parameter k_2 should be within $1 < k_2 < \frac{Lk_1}{L+k_1}$. In cases 2 and 3, for a specific L, we have $k_2 > \frac{Lk_1}{L+k_1}$ and $k_2 > L$, respectively. In case 1, the minimum amount of k_2 is set to be 1.01 and in cases 2 and 3, the maximum amount of k_2 is set to be 10000. For all the cases, the maximum amount of ratio (ρ_{mx}) is reported obtained from all possible amounts of k_2 for a specific L. As it is shown in Table 3.1, L is changing from 2 to 10000 and the maximum amount of competitive ratio for cases 1, 2 and 3 is at most 2.247761, 2.00128 and 1.990304, respectively. We can claim that the competitive ratio of the DQC algorithm is at most 2.247761, considering all different cases. Note that by increasing L, the competitive ratio in cases 2 and 3 converges to 1.618, which is the number Keskinocak and Tayur (2001) reported as the competitive ratio of their problem where their problem is a special scenario of our problem in these two cases. In their problem, the manufacturer is assumed to be a single machine and they consider only maximizing revenue in an e-tail channel, i.e., ($c_1 = c_2 = 0$). In order to evaluate the performance of bounds provided in lemmas 3.3 and 3.4, for case 2 and 3 of the data sets, the gap between the upper bound and lower bound of the DQC's competitive ratio is presented in Figures 3.1 and 3.2. In these figures, k_1 and T are set as mentioned above, and for each L and its corresponding k_2 amounts, the maximum amount of upper bound and the minimum amount of lower bound is reported.

TABLE 3.1. Competitive ratio of *DQC* Algorithm

	Case 1	*Case* 2	*Case* 3
L	ρ_{mx}	ρ_{mx}	ρ_{mx}
2	1.001	2.00128	1.990304
3	1.05722	1.955236	1.954316
4	1.066855	1.897095	1.896957
5	1.202256	1.857932	1.860584
6	1.307815	1.820385	1.818566
10	1.570692	1.743365	1.750954
50	2.060528	1.637027	1.64149
100	2.150173	1.625775	1.626146
500	2.228154	1.618276	1.613276
1000	2.238121	1.611491	1.6121
10000	2.247761	1.617716	1.617945

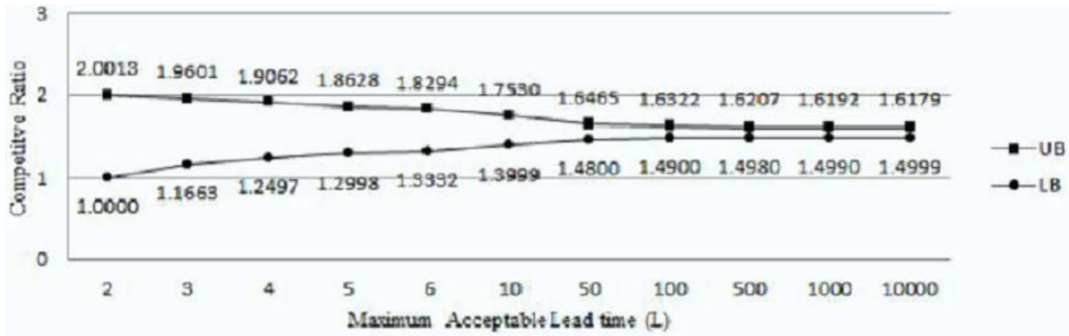

FIGURE 3.1. Upper bound and Lower bound of ρ (Case 2)

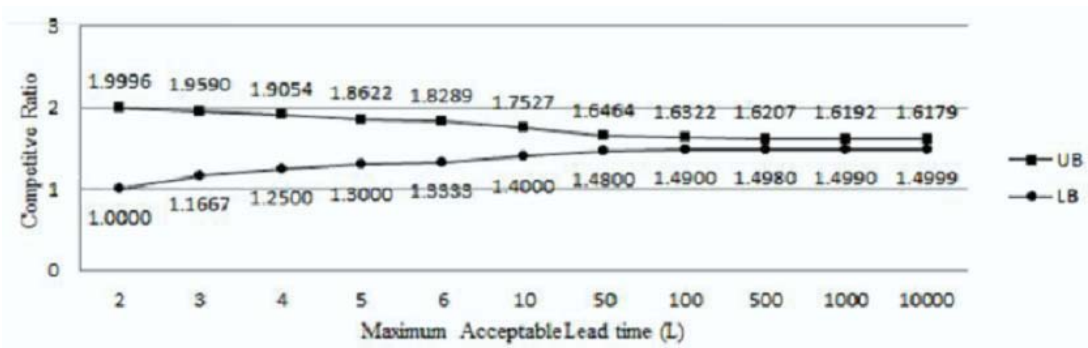

FIGURE 3.2. Upper bound and Lower bound of ρ (Case 3)

3.6 Conclusion

With the growth of e-business, many companies are trying to adopt online (e-tail) channel besides traditional retail stores and provide more convenient access of products by this dual channel strategy. It is accepted that even a well-designed dual channel supply chain is useless when it is not successful to deliver items as promised. One of the most important challenges for these facilities is to quote and manage the most efficient due dates to get the competitive advantage in the market. In this chapter, we studied reliable due date quotation in two-echelon dual channel supply chain while there is an availability interval for online customers. We applied competitive analysis for this problem while maximizing the total profit. The profit function consists of linear due date sensitive revenue and delivery cost. We considered two delivery options for e-tail customers with different costs and availability intervals and also capacity constraint. We provided parametric bounds on the competitive ratio of any arbitrary online strategy, also investigated the competitive ratio of a specific online algorithm for single type of e-tail channel orders. Computational experiments illustrate the effectiveness of the proposed analysis.

Chapter 4

Due date quotation with shared production capacity in dual channel supply chain

4.1 Introduction

In this chapter, we study an extension for the problem of due date quotation coordinated with delivery schedule in a two-echelon dual channel supply chain, which was discussed in chapter 3. Some of the assumptions are modified in the extension problem in order to make it more realistic. In chapter 3, we defined the capacity constraint as the capability of processing at most N online orders at any time by the manufacturer, which is modified in this chapter. Although it is possible that every channel's orders been assigned to different production processes and machines, we may have a number of real-world situations where several channels share the equipment and use the same production process. In other words, it is reasonable to adjust the capacity constraint of previous model to the situation where both e-tail and retailer channel's demand share the production capacity in the manufacturer's system. The problem studied in this chapter assumes that the manufacturer can process at most N orders (both online and retailer's demand) at any time in the system. In the previous problem, the whole capacity was considered only for e-tail channel orders, and it was assumed that the retailer order's schedule do not affect the online order's acceptance or sequence. However, it is clear that considering common capacity of production for both channels affect scheduling and due date settings for e-tail and retail orders.

We first present a modified batch definition for this new problem, and then introduce a specific online algorithm for quoting due dates to e-tail customers (if they are accepted) and scheduling the retailer's orders while maximizing the total profit. Unlike the previous model, in this chapter, the total profit one can obtain from retailer's demands are not fixed. The goal in this problem is to evaluate the performance of the proposed online algorithm. We determine competitive ratio of the proposed algorithm by applying competitive analysis.

This chapter is organized as follows. Section 4.2 explains the problem with assumptions and section 4.2.1 provides the problem notations and mathematical model. Section 4.3 proposes specific online algorithm and investigates its performance for single-type e-tail customers. A detailed computational experiment is provided in Section 4.4 and finally, Section 4.5 concludes with a summary of the insights from the analysis.

4.2 Problem Definition

In this chapter, we study the problem of due date management (due date quotation coordinated with scheduling) for online customers in a two-echelon dual channel supply chain while maximizing total profit obtained from both accepted online orders and retailer's demand. The problem presented in this chapter is as an extension of the problem studied in chapter 3, where there exist some modifications making the problem more realistic. The problem's structure (network) is still the same; we have one manufacturer and a retailer as the traditional channel and online customers as the e-tail channel who are served by the manufacturer. There exist two options for delivering products to the online customers i.e., shipping directly from the manufacturer to online customers which is available at any time, and delivering through the retail store which is available at specific periods and imposes less cost to the system. In this model, we still have the assumption of unknown demand for e-tail channel (at any time there is no information about future arrivals of online orders) and deterministic demand for the traditional channel (which implies that there is an optimal cycle time (T) for delivering items to the retail store). However, unlike the previous model, the total profit one can obtain from retailer's demands is not fixed. The profit one can obtain from any orders of retailer's channel contains; fixed revenue and delivery cost beside the order's earliness cost (penalty per unit of time after order's completion time before being delivered). Lets assume that the manufacturer has to process Q number of retailer's orders in each T-cycles. For each unit of time that the retailer's order is completed but not delivered to the customer (being held by the manufacturer till its delivery time), r' earliness cost is carried out by the system, which represents the retailer's variable cost. It is obvious that $Q \leq NT$, where N is the manufacturer's whole capacity for processing orders of both channels at any time. The problem studied in this chapter assumes that the manufacturer can process at most N orders (both online and retailer's demand) at any time in the system. However, in the previous model, it was presumed that the retailer order's schedule do not affect e-tail order's acceptance or schedule. It is clear that by considering common capacity of production for both channel's demand, scheduling and due date settings would be different.

The other assumptions are similar to the previous model; there exists maximum acceptable lead time (L') for e-tail customers, and if they are offered a due date after their desired lead time, they will not place the order. From the manufacturer's point of view, in fact the manufacturer has the option of rejecting the online order by offering a due date after their desired lead time when there is no benefit in accepting the order or there exists capacity restriction. The term due date is still referred to the time that the order is shipped to the customer, i.e., the time the order leaves the manufacturer, thus the quoted due date for each order may be different from the time the order production is completed. Since the delivery option through the retail store is available at specific periods with less cost, the online order may be held by the manufacturer after its production is completed in order to use the most cost-effective delivery method. Since Q number of retailer's orders must be produced within each T-cycle and being delivered at the end of the cycle, their completion time may be different from delivery time as well, because of the capacity constraints. We assume that the revenue of the e-tail orders will decrease linearly if the quoted due dates increase. In this problem, we quote 100% reliable due dates to the online customers as well and there is no tardiness cost. We consider a basic model, with single-type orders for both channels, i.e., all online and retailers orders have unit-length processing time, and similar cost parameters. Although these basic models look rather restricted, they still enclose many difficulties of online quotation problems.

4.2.1 Mathematical Model

In this sub-section, we first introduce the batch definition for this new problem, then the notations used in the rest of the chapter are provided. The mathematical model is also followed which maximizes the total profit obtained from orders of both channels. Assume that ∂ is the schedule of online and retailer's orders generated by any online algorithm, we can divide each schedule into batches where each batch contains consecutively scheduled orders (it can be both channel's orders). Let s_i be the start time of the batch B_i and s_i' the completion time of the last order in the batch. In batch B_i, the online order which is processed at time s_i, has also arrived at s_i and all the accepted online orders arrived before s_i are processed before, however they may leave the system after s_i. Batch definition in this chapter is different from chapter 3, since it can be a partial schedule of both channel orders, although it starts with the accepted and processed e-tail orders.

As we assume that we have a single type of customers, our problem of due date quotation will be reduced to determining how many online orders should be accepted in each period, how many accepted online orders or retailer's orders should be processed and how many processed orders should be shipped in each period. We use the following notations;

i	Time index , $i = 1, 2, ..., n$.
T	Optimal cycle time of shipments to the retail store.
π	Set of time indices that are multiples of T, $\{T, 2T, 3T, ...\}$.
t_i	Time interval between period i and the next period of regular shipment to the retail store.
r	Penalty (or revenue that is lost) for each unit of time that the online order waits before being delivered to the customer.
r'	Earliness cost per unit of time for each of the retailer's orders.
L	Maximum acceptable lead time for online orders excluding their processing time.
c_1	Delivery cost per order (online or retailer's) through the retail store.
c_2	Delivery cost per online order shipped directly from the manufacturer.
N	Maximum number of orders (from both channels) can be processed by manufacturer at any time
d_i	Number of arrived online orders (e-tail demand) in period i
Q	Number of retailer's orders that should be produced in each T-cycle

$$\sigma(i) = \begin{cases} 1 & \text{if } i \notin \pi \\ 0, & \text{otherwise} \end{cases}$$

Decision Variables:

q_i Number of accepted online orders in period i.

w_i Number of accepted online orders shifted from period i to $i+1$ before being processed.

u_i Number of online orders processed in period i but not delivered to the customers (being held).

v_i Number of online orders processed in period i and delivered to the customers.

q'_i Number of of accepted retailer's orders in period i.

w'_i Number of accepted retail orders shifted from period i to $i+1$ before being processed.

u'_i Number of retail orders processed in period i but not delivered to the customers (being held).

For a schedule ∂ with n periods, we can define the following mathematical model;

$$Max \sum_{i=1}^{n} C_i, \text{ where } C_i = rLq_i - rw_i - u_i(rt_i + c_1) - v_i[\sigma(i)(c_2 - c_1) + c_1] - u'_i(r't_i),$$

$s.t.$

$$\begin{array}{ll}
u_i + v_i + u'_i \leq N & \forall\ i = 1, 2, ..., n, \\
q_i + w_{i-1} - w_i = u_i + v_i & \forall\ i = 1, 2, ..., n, \\
q'_i + w'_{i-1} - w'_i = u'_i & \forall\ i = 1, 2, ..., n, \\
w'_i \leq t_i N \sigma(i) & \forall\ i = 1, 2, ..., n, \\
q'_i = (1 - \sigma(i-1))Q & \forall\ i = 1, 2, ..., n, \\
q_i \leq d_i & \forall\ i = 1, 2, ..., n,
\end{array}$$

where, the first term of the objective function is the maximum possible revenue one can obtain from any accepted online order. The expression $rw_i + u_i rt_i$ represents the revenue lost for the accepted online orders for each unit of time they spend before being delivered. The terms $u_i c_1$ and $v_i[\sigma(i)(c_2 - c_1) + c_1]$ are the delivery costs of online orders shipped through the retail store or directly by the manufacturer, respectively. And $u'_i(r't_i)$ represents the earliness cost of retailer's orders.

The first set of constraints represent the capacity restriction. The second and third set of constraints represent that at any time, the number of orders produced and delivered should be equal to the number of accepted orders and the ones which are held to be delivered by the manufacturer, for e-tail and retailer's channels, respectively. The forth constraint ensures that all retailer's orders in each T-cycle should be processed before their predetermined due date (the end of the cycle). The fifth constraint shows that the manufacturer accepts to process fixed number of retailer's orders (Q) at the beginning each T-cycle, and the last constraint ensures that the number of accepted online orders at any time is less than online channel demand.

We have online orders arrive over time and fixed number of retailer's orders in each T-cycle. In the online version of due date quotation model considered in this chapter, all the information about online orders become available as they arrive at the system, and their release times are not known in advance. In these algorithms decision about accepting or rejecting the online orders and also quoting due dates for the orders (if accepted) must be made as soon as they arrive. It is clear that due date setting is tightly integrated with scheduling and for the model we have in this chapter, we need to consider retailers orders as well. It is assumed that the retailer's demand is deterministic and the manufacturer can not reject or delay retailers orders because of online orders.

We know that any online algorithm tries to schedule accepted online orders as soon as possible to guarantee the available capacity for future arrivals, as there is no information about the future online arrivals. In addition, there may exist retailer's orders that must be processed before their predetermined due dates and should be delivered without any tardiness. Therefore, we accept Remark 4.1.

Remark 4.1. *Considering any online algorithm, in the case of no online orders and having not-processed retailer's demand at any time, the online algorithm will process at least some of the retailer's orders at that time in order to reserve the capacity for future online arrivals. In the case of having both online arrivals and not-processed retailers demand at any time, scheduling online orders have more priority if there is enough capacity to postpone scheduling of retailer's orders upto their predetermined* *due date.*

Competitive analysis is used to evaluate the performance of online algorithms, where the result of online algorithm is compared with optimal offline algorithms' for specific instances. For the offline algorithms all the information about the orders are available in advance. For a given batch with n periods, let $Z_{(n)}$ denotes the total profit obtained from an online algorithm and $Z^*_{(n)}$ the maximum profit one can obtain from retailer's orders and the online arrivals during the batch.

4.3 Competitive Analysis

In online optimization problems, online algorithm's performance is mainly evaluated via the competitive analysis; comparing an online algorithm's result with the offline model's optimal solution. For the problem in this chapter, all the information about the online orders are available in advance for the optimal offline algorithms to obtain maximum possible profit. Given an instance I, let $Z_{(I)}$ denote the total profit obtained by using an online algorithm, and $Z^*_{(I)}$ denote the maximum profit obtained by an optimal offline algorithm. As it was discussed in chapter 3, for maximization problems, the online algorithm is called $\rho-$competitive if $Z^*_{(I)} \leq \rho Z_{(I)} + b$ where $\rho \geq 1$, and b is a constant. We define competitive ratio as $\rho = sup\rho(\frac{Z^*_{(I)}}{Z_{(I)}})$ for $Z_{(I)} > 0$. Determining the bounds for competitiveness of online algorithms, in other words, finding bounds of competitive ratio (ρ) for the online algorithms is the main issue and challenging part of online optimization problems.

According to the batch definition provided in section 4.2.1, any schedule of orders (considering both online and retailer's orders) generated by an online algorithm can be divided into batches. Therefore, if we investigate the competitive ratio of a given batch, we can generalize the results to determine competitiveness of the corresponding online algorithm. In this section, we first introduce a specific online algorithm called DQCC for single type of customers in both channels, then we evaluate the performance of proposed algorithm by determining parametric upper bound for its competitive ratio.

4.3.1 Algorithm of Due Date Quotation for Online Customers with Shared Capacity (DQCC)

In this subsection, we introduce a specific online algorithm of due date quotation for single type of customers, called DQCC and we investigate its corresponding competitive ratio. Select $0 < \alpha < 1$, among the online orders that arrive at time t, the ones that yield at least $\alpha(rL - c_1)$ profit are accepted only if we have enough capacity to postpone existing retailer's demand to be processed upto their due date. At any time, if there is no online order to be processed, the existing retailer's demand will be processed using full possible capacity. The accepted online orders will be scheduled at the earliest possible position. Note that $(rL - c_1)$ like the one discussed in chapter 3, represents the maximum possible profit yield from any accepted online order, which includes the maximum possible revenue (rL) and the minimum delivery cost (c_1), gained from delivery through the retail store without holding the order. In fact, in this algorithm, an online order is accepted if a certain fraction of maximum profit is guaranteed and also if accepting those online orders may not lead to lateness for delivering existing retailer's demand.

Remark 4.2. *Consider the batch B_l with n periods. If the batch contains more than one T-cycle, based on the online algorithm definition and Remark 4.1, in all T-cycles of the batch except the last one, the online algorithm will process all the retailer's orders as late as possible before their due date(at the end of the cycles). In an optimal offline algorithm also retailer's demand will be processed at the end of the cycle to minimize the earliness cost of retailer's demand.*

Lemma 4.1. *The competitive ratio of algorithm (DQCC) is at most $\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}$ where $\alpha'' = \alpha(1 - \frac{1}{k_1}) + \frac{1}{k_2}$ and α satisfies the following equation,*

$$\frac{(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2 + \frac{2T\alpha''}{L})}{(1 - \alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}} = \frac{(rL - c_1)}{\alpha(rL - c_1) + \frac{1}{T}(c_2 - c_1)}. \tag{4.1}$$

Proof: Consider the batch B_l with n periods. If the batch contains more than one T-cycle, according to Remark 4.2, both online and offline algorithms would schedule the retailers demand at the end of the cycle for all T-cycles except the last one. Thus, the difference between the online and offline algorithm's schedules are the number of accepted online orders in the last T-cycle in the batch and therefore the schedule of retailer orders in the last cycle and clearly the profit one can gain from all online order arrivals during the batch time. Therefore, in this proof only the profit got form retailer's orders in the last T-cycle of the batch is considered. Let $Z_{(n)}$ be the profit obtained from DQCC algorithm and $Z^*_{(n)}$ the maximum possible profit one can gain from retailer's orders and the online arrivals during the batch. Note that all the accepted online orders by algorithm DQCC yield at least $\alpha(rL - c_1)$ profit. Assume R is the revenue obtained from an accepted online order by DQCC, and in worst case it is delivered directly to the customer, then $R - c_2 \geq \alpha(rL - c_1)$. Let $\alpha' = \frac{c_2 - \alpha c_1}{rL}$, then $R \geq \alpha rL + \alpha' rL$, and any accepted online order by algorithm DQCC yield at least $\alpha'' rL$ revenue where $\alpha'' = \alpha + \alpha'$ and $0 < \alpha'' < 1$. Let n be the last period of the batch where $n \notin \pi$, and let T' be the next regular shipment to retail store after the batch completion time, i.e., $t_n + n = T'$. Also assume that n_1 is the last period that online orders are scheduled in the last cycle of the batch. Note that for determining the bounds of competitive ratio, we have to consider the worst-case situation. First assume that $n_1 \geq \lfloor (1 - \alpha'')L \rfloor + 1$, the revenue we can get from DQCC algorithm is at least

$$
\begin{aligned}
Z_{(n)} \quad \geq \quad & rN(\frac{L(L+1)}{2} - \frac{\lfloor \alpha'' L \rfloor \left(\lfloor \alpha'' L \rfloor + 1 \right)}{2}) + (n_1 - \lfloor (1 - \alpha'')L \rfloor - 1))\alpha'' rLN \\
& -n_1 N c_2 + \left\lfloor \frac{n_1}{T} \right\rfloor N(c_2 - c_1) - \\
& Nr'(\frac{(T' - n_1)(T' - n_1 - 1)}{2} - \frac{(T' - n)(T' - n - 1)}{2}) \\
\geq \quad & rN(\frac{L(L+1)}{2} - \frac{\alpha'' L(\alpha'' L + 1)}{2}) + (n_1 - (1 - \alpha'')L - 1))\alpha'' rLN \\
& -n_1 N c_2 + (\frac{n_1}{T} - 1)N(c_2 - c_1) - \\
& \frac{Nr'}{2}((T' - n_1)^2 - (T' - n_1) - (T' - n)^2 + (T' - n)). \qquad (4.2)
\end{aligned}
$$

In the worst-case situation, we assume all the online orders scheduled in the first $\lfloor (1 - \alpha'')L \rfloor + 1$ periods have been arrived at $t = 1$, therefore the revenue DQCC algorithm can gain from these orders will be $rN(\frac{L(L+1)}{2}-\frac{\lfloor\alpha''L\rfloor(\lfloor\alpha''L\rfloor+1)}{2}) \geq rN(\frac{L(L+1)}{2}-\frac{\alpha''L(\alpha''L+1)}{2})$, which is the first term in right-hand side of equation (4.2). Term $\alpha''rL$ is the minimum revenue DQCC algorithm can get from the remaining periods $(n_1 - \lfloor(1-\alpha'')L\rfloor - 1))$ in batch B_l, thus we have $(n_1-(1-\alpha'')L-1))\alpha''rLN$ as well. The expression $-n_1Nc_2 + \lfloor\frac{n_1}{T}\rfloor N(c_2-c_1)$ denotes the maximum delivery cost for all the online orders scheduled in the batch, where we know that $\lfloor\frac{n_1}{T}\rfloor \geq \frac{n_1}{T}-1$, and therefore $\lfloor\frac{n_1}{T}\rfloor N(c_2-c_1)$ is replaced by $(\frac{n_1}{T}-1)N(c_2-c_1)$ in equation (4.2). The expression $Nr'(\frac{(T'-n_1)(T'-n_1-1)}{2} - \frac{(T'-n)(T'-n-1)}{2}$ denotes the earliness cost for retailer's orders in the last T-cycle. Without loss of generality, let $c_1 \times k_1 = rL$ and $c_2 \times k_2 = rL$ where $k_1 > k_2 > 1$, then $-N(c_2-c_1)$ in equation (4.2) can be replaced by $\frac{NrL^2}{2}(\frac{-2}{Lk_2}+\frac{2}{Lk_1})$. By rearranging the equation (4.2), we have

$$\begin{aligned} Z_{(n)} \geq\ & \frac{NrL^2}{2}((1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}) + \\ & n_1Nr\alpha''L - n_1Nc_2 + \frac{n_1}{T}N(c_2-c_1) - \\ & \frac{Nr'}{2}((T'-n_1)^2 - (T'-n_1) - (T'-n)^2 + (T'-n)). \end{aligned} \tag{4.3}$$

Note that $\alpha''rL = \alpha rL + c_2 - \alpha c_1$, therefore the term $nNr\alpha''L - nNc_2$ in equation (4.3) is equal to $nN\alpha(rL-c_1)$, and we have

$$\begin{aligned} Z_{(n)} \geq\ & \frac{NrL^2}{2}((1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}) + \\ & n_1N\alpha(rL-c_1) + \frac{n_1}{T}N(c_2-c_1) - \\ & \frac{Nr'}{2}((T'-n_1)^2 - (T'-n_1) - (T'-n)^2 + (T'-n)). \end{aligned} \tag{4.4}$$

For the maximum profit $Z^*_{(n)}$, in the presumed worst-case situation, we need to find out the last possible period we might have online arrivals in the batch. We assumed that $n \notin \pi$, and there exist a gap between n and T'. Therefore, if there were online arrivals after $(n_1 - \lfloor(1-\alpha'')L\rfloor)^{th}$ period, there would be enough capacity to schedule retailer's orders later, and the batch length would be more than n. Thus, the last period in the batch that could have any online arrival is the period $(n_1 - \lfloor(1-\alpha'')L\rfloor)^{th}$. We assume the maximum number of online orders we can accept (NL) have arrived in the that last period (worst-case situation). Thus, we may have two cases; 1) $t_{n_1} \geq \lfloor\alpha''L\rfloor + (n-n_1)$ or 2) $t_{n_1} < \lfloor\alpha''L\rfloor + (n-n_1)$. In the first case, we have enough capacity to schedule all NL online orders accepted in the last period, consecutively before retailers orders in the last cycle. In the second case, some of the accepted online orders may be scheduled after retailers orders (after T'). It is clear that the maximum ratio would be obtained in the first case, which is considered in the following.

The maximum profit one can obtain from retailer's orders and the online arrivals during the batch B_l, (for the considered worst-case situation) is as follows:

$$
\begin{aligned}
Z^*_{(n)} \leq \; & (n_1 - \lfloor(1-\alpha'')L\rfloor))NrL + Nr(\frac{L(L-1)}{2}) - n_1Nc_1 - \\
& \lfloor\alpha''L\rfloor Nc_1 - rN\left\lceil\frac{n_1}{T}\right\rceil(1+2+...+x) - \\
& \frac{Nr'}{2}((T'-n_1-\lfloor\alpha''L\rfloor)(T'-n_1-\lfloor\alpha''L\rfloor-1) - \\
& (T'-n-\lfloor\alpha''L\rfloor)(T'-n-\lfloor\alpha''L\rfloor-1)). \qquad (4.5)
\end{aligned}
$$

Considering the batch definition, capacity constraint N, and the worst-case situation mentioned above, there would be no online arrivals after $(n_1 - \lfloor(1-\alpha'')L\rfloor)^{th}$ period of the batch B_l. The first term in the right-hand side of equation (4.5) shows that all the online orders arrived in periods $1, 2, ..., n_1 - \lfloor(1-\alpha'')L\rfloor$ can obtain maximum amount of revenue which is rL, and the second term denotes that the maximum possible revenue one can get from the online arrivals in periods $(n_1 - \lfloor(1-\alpha'')L\rfloor)^{th}$ till n is $Nr(\frac{L(L-1)}{2})$. As it was mentioned above, the last period in the batch that we could have any online arrival is the period $(n - \lfloor(1-\alpha'')L\rfloor)^{th}$, and in order to obtain maximum possible revenue, we assume the maximum number of orders we can accept (NL) have been arrived in that period. Therefore, the maximum possible revenue for the arrivals during those periods is $Nr(L-1) + Nr(L-2) + ... + Nr(1) = Nr(\frac{L(L-1)}{2})$.

The terms $n_1 N c_1$ and $\lfloor \alpha'' L \rfloor N c_1$ are also the minimum delivery cost for all online arrivals during the batch time. Some of the online orders delivered through the retail store may have been held after their completion time, and it is obvious that their holding time would be $rN\lceil \frac{n_1}{T} \rceil (1+2+...+x)$, where $x = \frac{c_2-c_1}{r} = (\frac{1}{k_2} - \frac{1}{k_1})L$. The expression $\frac{Nr'}{2}((T'-n_1-\lfloor \alpha'' L \rfloor)(T'-n_1-\lfloor \alpha'' L \rfloor - 1) - (T'-n-\lfloor \alpha'' L \rfloor)(T'-n-\lfloor \alpha'' L \rfloor - 1))$, represents the earliness cost for retailers orders in the last cycle of the batch, where the first case ($t_{n_1} \geq \lfloor \alpha'' L \rfloor + (n-n_1)$) is satisfied. In this case, we have enough capacity to schedule all the online orders accepted in the last possible period consecutively and shift the retailer's orders to be scheduled later but before their predetermined due date.

Note that $-\lfloor \alpha'' L \rfloor N c_1 \leq (1-\alpha'' L) N c_1 = \frac{NrL^2}{2}(\frac{2}{Lk_1} - \frac{2\alpha''}{k_1})$, and $-rN\lceil \frac{n_1}{T} \rceil \frac{x(x+1)}{2}$ $\leq -rN\frac{x^2}{2} = -\frac{NrL^2}{2}(\frac{1}{k_2} - \frac{1}{k_1})^2$. By rearranging equation (4.5), we have

$$\begin{aligned} Z^*_{(n)} \leq \; & \frac{NrL^2}{2}(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2) + \\ & n_1 N(rL - c_1) - \frac{Nr'}{2}((T'-n_1-\lfloor \alpha'' L \rfloor)(T'-n_1-\lfloor \alpha'' L \rfloor - 1) - \\ & (T'-n-\lfloor \alpha'' L \rfloor)(T'-n-\lfloor \alpha'' L \rfloor - 1)). \end{aligned} \tag{4.6}$$

After rearranging the last term in the right-hand side of the equation (4.6), we have $-\frac{Nr'}{2}((T'-n_1)^2-(T'-n_1)-(T'-n)^2+(T'-n))+Nr'\lfloor \alpha'' L \rfloor (n-n_1)$. Therefore,

$$\begin{aligned} Z^*_{(n)} \leq \; & \frac{NrL^2}{2}(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2) + \\ & n_1 N(rL - c_1) + Nr'\lfloor \alpha'' L \rfloor (n-n_1) \\ & -\frac{Nr'}{2}((T'-n_1)^2 - (T'-n_1) - (T'-n)^2 + (T'-n)). \end{aligned} \tag{4.7}$$

In case 1, we have $t_{n_1} \geq \lfloor \alpha'' L \rfloor + (n-n_1)$, thus $(n-n_1) \leq t_{n_1} \leq T$, and we can replace the $Nr'\lfloor \alpha'' L \rfloor (n-n_1)$ in equation (4.7) with $Nr'\lfloor \alpha'' L \rfloor T$. We assume that $r' = r$, Therefore, $Nr\lfloor \alpha'' L \rfloor T = \frac{NrL^2}{2}(\frac{2T\alpha''}{L})$, and

$$\begin{aligned} Z^*_{(n)} &\leq \frac{NrL^2}{2}(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2 + \frac{2T\alpha''}{L}) + n_1 N(rL - c_1) \\ &\quad - \frac{Nr'}{2}((T' - n_1)^2 - (T' - n_1) - (T' - n)^2 + (T' - n)). \end{aligned} \quad (4.8)$$

Considering the right-hand side of equations (4.4) and (4.8), If we set

$$\frac{(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2 + \frac{2T\alpha''}{L})}{(1-\alpha'')^2 + \frac{(1-3\alpha'')}{L} - \frac{2}{Lk_2} + \frac{2}{Lk_1}} = \frac{(rL - c_1)}{\alpha(rL - c_1) + \frac{1}{T}(c_2 - c_1)}, \quad (4.9)$$

we have $Z^*_{(n)} \leq (\frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)})Z_{(n)}$. Note that, $\frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)} = \frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})}$, then $Z^*_{(n)} \leq (\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})})Z_{(n)}$, where α is obtained from the quadratic equation (4.10). By replacing $\alpha'' = \alpha + \frac{c_2 - \alpha c_1}{rL} = \alpha(1 - \frac{1}{k_1}) + \frac{1}{k_2}$ in equation (4.9) , we have

$$\alpha^2(a) + \alpha(b) + (c) = 0 \quad (4.10)$$

$$a = -(1 - \frac{1}{k_1})^3 + (1 - \frac{1}{k_1})^2(2 - \frac{2}{k_1} + \frac{2T}{L})$$

$$\begin{aligned} b &= (1 - \frac{1}{k_1})(\frac{1}{k_2})(2 - \frac{2}{k_1} + \frac{2T}{L}) + \frac{1}{T}(1 - \frac{1}{k_1})(\frac{1}{k_2} - \frac{1}{k_1})(2 - \frac{2}{k_1} + \frac{2T}{L}) + \\ &\quad (1 - \frac{1}{k_1})(\frac{1}{L} - 1 + \frac{2}{Lk_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2) - 2(1 - \frac{1}{k_1})^2(\frac{1}{k_2}) - \\ &\quad (1 - \frac{1}{k_1})^2(-2 - \frac{3}{L}) \end{aligned} \quad (4.11)$$

$$\begin{aligned} c &= \frac{1}{k_2 T}(\frac{1}{k_2} - \frac{1}{k_1})(2 - \frac{2}{k_1} + \frac{2T}{L}) + \frac{1}{T}(\frac{1}{k_2} - \frac{1}{k_1})(\frac{1}{L} - 1 + \frac{2}{Lk_1} - (\frac{1}{k_2} - \frac{1}{k_1})^2) - \\ &\quad (1 - \frac{1}{k_1})(\frac{2}{Lk_1} - \frac{2}{Lk_2}) - (1 - \frac{1}{k_1})(\frac{1}{L} + 1 + \frac{1}{k_2^2}) - (1 - \frac{1}{k_1})\frac{1}{k_2}(-2 - \frac{3}{L}) \end{aligned} \quad (4.12)$$

It is obvious that $\frac{1}{k_2} - \frac{1}{k_1} \geq 0$, then $\frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})} \leq \frac{1}{\alpha}$, and $Z^*_{(n)} \leq \frac{1}{\alpha} Z_{(n)}$ where α is obtained from equation (4.10).

Note that if $n_1 < \lfloor (1-\alpha'')L \rfloor + 1$, then all possible online arrivals during the batch length are accepted by the online algorithm and are schedule in a non-decreasing order of their arrival times, since any online arrival during the $\lfloor (1-\alpha'')L \rfloor + 1$ periods yield at least $\alpha'' rL$ revenue. It can be simply shown that the online algorithm in this case gives an optimal solution, i.e., $\sum_{i=1}^{n} q'_i = \sum_{i=1}^{n} q_i$ and $Z^*_{(n)} = Z_{(n)}$. □

Corollary 4.1. *In Lemma 4.1, we assumed we have the option of holding the completed online orders to be delivered through the retail store, which implies that* $c_2 > r + c_1$. *However, we may have two other cases;* ($c_2 > r$ & $c_2 < r + c_1$) *and (* $c_2 < r$, *i.e.,* $k_2 > L$*), where holding the completed online item is not profitable, and completed orders may be delivered through the retail store only if their completion time is at* π *set. For these two cases,* $Z^*_{(n)}$ *in Lemma 4.1 changes to* $Z^*_{(n)} \leq \frac{NrL^2}{2}(2\alpha'' - 1 + \frac{1}{L} + \frac{2}{Lk_1} - \frac{2\alpha''}{k_1} + \frac{2T\alpha''}{L}) + nN(rL - c_1) - \frac{Nr'}{2}((T'-n_1)^2 - (T'-n_1) - (T'-n)^2 + (T'-n))$, *and we have* $Z^*_{(n)} \leq \frac{(1-\frac{1}{k_1})}{\alpha(1-\frac{1}{k_1})+\frac{1}{T}(\frac{1}{k_2}-\frac{1}{k_1})} Z_{(n)}$ *, where* α *is obtained from equation* $\frac{(2\alpha''-1+\frac{1}{L}+\frac{2}{Lk_1}-\frac{2\alpha''}{k_1}+\frac{2T\alpha''}{L})}{(1-\alpha'')^2+\frac{(1-3\alpha'')}{L}-\frac{2}{Lk_2}+\frac{2}{Lk_1}} = \frac{(rL-c_1)}{\alpha(rL-c_1)+\frac{1}{T}(c_2-c_1)}$.

4.4 Experimental Results

In order to evaluate the competitive performance of the proposed algorithm, we define three different cases, 1) $c_2 > r + c_1$ 2) $c_2 > r$ & $c_2 < r + c_1$ 3) $c_2 < r$, i.e., $k_2 > L$. We investigate DQCC algorithm's performance by providing computational experiments on its competitive ratio upper-bound (Lemma 4.1). Note that we may use different scenarios for each case, and in each scenario, the maximum possible ratio is reported since for our performance evaluation the upper bound of the ratio is required. For all the three cases, experimental analysis denoted that by increasing k_1 while other parameters are fixed, the algorithm DQCC's ratio increases, thus the maximum amount of ratio occurs with the minimum amount of delivery cost c_1. In addition, by increasing T, the competitive ratio decreases (α increases) and it implies that the maximum amount of ratio occurs when T has the minimum amount. For each case, the competitive ratio is calculated for different amounts of L and k_2 while k_1 and T are set to be 10000 and 2, respectively. (Note that $T = 1$ is not considered for this problem, as it eliminates the second option of direct delivery and $k_1 = 10000$ examined and proved to be large enough for the analysis). The results are provided in Table 4.1. For case 1, we have $c_2 > r + c_1$ which means $\frac{1}{k_2} > \frac{1}{k_1} + \frac{1}{L}$, thus for a specific L, the parameter k_2 should be within $1 < k_2 < \frac{Lk_1}{L+k_1}$. In cases 2 and 3, for a specific L, we have $k_2 > \frac{Lk_1}{L+k_1}$ and $k_2 > L$, respectively. In case 1, the minimum amount of k_2 is set to be 1.01 and in cases 2 and 3, the maximum amount of k_2 is set to be 10000. For all the cases, the maximum amount of ratio (ρ_{mx}) is reported obtained from all possible amounts of k_2 for a specific L. As it is shown in Table 4.1, L is changing from 2 to 10000 and the maximum amount of competitive ratio for cases 1, 2 and 3 is at most 4.857014, 3.290674 and 3.290669, respectively. We can claim that the competitive ratio of the DQCC algorithm is at most 4.857014, considering all different cases. Note that by increasing L, the competitive ratio in all cases converges to 1.618, which is the number Keskinocak and Tayur (2001) reported as the competitive ratio of their problem where their problem is a special scenario of our problem in these two cases. In their problem, the manufacturer is assumed to be a single machine and they consider only maximizing revenue in an e-tail channel, i.e., ($c_1 = c_2 = 0$, with no retail orders).

Table 4.1. Competitive ratio of $DQCC$ Algorithm

	$Case$ 1	$Case$ 2	$Case$ 3
L	ρ_{mx}	ρ_{mx}	ρ_{mx}
2	4.857014	3.290674	3.290669
3	3.183267	2.622464	2.622462
4	2.57460	2.351725	2.351724
5	2.307077	2.201813	2.201813
6	1.953789	1.94741	1.94741
10	1.680011	1.683079	1.683079
50	1.648414	1.65049	1.65049
100	1.623987	1.624509	1.624509
500	1.621001	1.62127	1.62127
1000	1.618329	1.618356	1.618356
10000	1.618062	1.618065	1.618065

4.5 Conclusion

In this chapter, we studied an extension of the problem presented in chapter 3, due date quotation coordinated with delivery schedule in dual channel supply chain, with the same objective and problem structure. However, capacity constraint in this problem is modified to be a more realistic one. Unlike the problem in chapter 3, it is assumed that the production capacity of the manufacturer is shared among the orders of both channels, i.e., there is no distinct production capacity for orders of different channels. This shared capacity assumption, affects the number of online orders that the algorithm accepts, their schedule and clearly the schedule of retail orders. In addition, in this problem the fixed profit of retail orders is adjusted and variable revenue lost is considered for orders of this channel. The solution procedure considered for this online optimization problem was competitive analysis as well. In this chapter, we investigated the competitive ratio of a specific online algorithm for single type of e-tail channel orders. The computational experiments illustrate the effectiveness of the proposed analysis.

Chapter 5

CONCLUSIONS AND FUTURE RESEARCH

5.1 Summary of contributions

The objective of this dissertation is to analyze the problem of due date quotation and delivery scheduling (due date management) in dual channel supply chain. The main goal is to study this problem from the manufacturer's point of view, investigate different production environments and develop proper methodologies for each situation. To this aim, three main problems (three different production environments) have been investigated and appropriate analysis and solution methodologies for each problem have been developed.

In Chapter 2, we studied a delivery scheduling problem where the manufacturer has to decide the earliest delivery time for the orders received from retail channel. In this problem, a two-echelon supply chain is considered where a retailer places bulk orders of the same product with different families to the manufacturer. Since the manufacture accepts only bulk orders, no online order is assumed for this problem. The analysis with no online customers is relatively easy and therefore, we consider families of products in this problem. For this problem, we consider only retail channel with deterministic demand and cross family setup time (a novel assumption in literature) which was motivated by an application from the automotive industry. In a problem with cross family setup, job allocations to families are machine (stage) based. A two-stage manufacturing system is assumed for this problem and therefore, this system can be represented by a two machine permutation flow shop, where each stage is represented by a machine with cross family setups and the objective is to minimize the maximum completion time (makespan) of the jobs in an order. In this chapter, we first analyzed some properties of the optimal schedule and proved that Johnson sequence is optimal for jobs belonging to the same family on both machines and developed an efficient branch and bound algorithm with complexity of $O(n^c)$, where c is a constant to solve the problem. This property of Johnson sequence for jobs belonging to the same family on both machines, is also applicable for past studies on two machine flow shop scheduling problems with family setups to minimize makespan. We also developed a hybrid genetic algorithm using properties of the optimal schedule to solve large scale problems. Computational experiment showed the effectiveness of our algorithms.

In Chapter 3, we studied a problem of reliable due date quotation coordinated with delivery schedule in a multi-processor manufacturing system receiving orders from both e-tail and retail channels. Online orders arrive over time, and as they arrive, the manufacturer will decide to accept or reject the orders and quote due dates to the accepted orders while deterministic demand is assumed for the retail channel. There exists an availability interval for online customers, i.e., accepted online orders should be delivered to the customers within their acceptable lead time via one of the two available options; directly by the manufacturer or through the retail store. Our goal in this problem was to quote due dates to the online orders and schedule them to maximize the total profit while satisfying the maximum acceptable lead time for online orders and distinct production capacity for each channel. In this chapter, we applied competitive analysis for this maximization problem where the profit function consists of linear due date sensitive revenue function and delivery costs. We first, provided parametric bounds on the competitive ratio of any arbitrary online strategy, and then investigated the competitive ratio of a specific online algorithm for single type of e-tail channel orders. Computational experiments illustrate the effectiveness of the proposed analysis.

In Chapter 4, we studied an extension of the problem presented in Chapter 3, due date quotation coordinated with delivery schedule in dual channel supply chain, with the same objective and problem structure. However, capacity constraint in this problem is modified to be a more realistic one. Unlike the problem in Chapter 3, it is assumed that the production capacity of the manufacturer is shared among the orders of both channels, i.e., there is no distinct production capacity for orders of different channels. This shared capacity assumption, affects the number of online orders that the algorithm accepts, their schedule and clearly the schedule of retail orders. In addition, in this problem the fixed profit of retail orders is adjusted and variable revenue lost is considered for orders of this channel. The solution procedure considered for this online optimization problem was competitive analysis as well. In this chapter, we investigated the competitive ratio of a specific online algorithm for single type orders of both channels. Computational experiments illustrate the effectiveness of the proposed analysis.

The future works for this dissertation are discussed in the next section:

5.2 Future work

In the first part of this dissertation, we studied a two-stage manufacturing system receiving orders with cross family setup time from retail channel and has to decide the earliest delivery time of accepted orders. This problem can be viewed as a two machine permutation flow shop problem with cross families and the objective function of minimizing the makespan. As the cross family setup time is a new assumption introduced to the scheduling literature, for the future work, it can be applied for several other scheduling problems like three machine flow shop problems or flow shop problems with dominant machine(s).

In Chapter 2, we investigated features of optimal schedule and accordingly, developed B&B and HGA algorithms. Developing a tighter lower bound can be a suggestion for improving the performance of such algorithms as the future work. Developing approximation algorithms (worst-case analysis) can also be an appropriate way to extend the presented problem.

For the problems presented in Chapters 3 and 4, where we studied online quotation versions in dual channel supply chain, we concentrated on special cases with single-type of customers. However, in many real-world situations we may have to deal with different types of orders from both e-tail and retail channels. Therefore, one of the extensions suggested as the future work is considering different types of orders, with different processing time and cost parameters. In order to analyze these types of problems, asymptotic probabilistic analysis of the model and heuristics can be helpful. In this type of analysis, a sequence of deterministic instances of the problem are generated randomly, and the objective values of the heuristic algorithms are evaluated when the size of the generated instances grows to infinity.

In addition, for both problems studied in chapters 3 and 4, it is assumed that the retail channel's demand is deterministic, which can be modified to the stochastic demand in an extension models, also the assumption of decreasing linearly the revenue function can be modified to a non-linear (step function) one as well.

Competitive analysis applied to evaluate the online heuristic algorithms in this study provides worst-case estimates and may not be representative of the average-case performance of the algorithms. Thus, computational experiments on online quotation problems in dual channel environment is suggested as a future work to compare average-case performance with worst-case performance of online strategies using randomized algorithms.

References

[1] Abdul-Jalbar, B., Gutierrez, J., Sicilia, J., 2006. Single cycle policies for the one-warehouse N-retailer inventory/distribution system. Omega, 34, 196–208.

[2] Agnieszka, R., Rudek, R., 2013. Makespan minimization flow shop with position dependent job processing times—computational complexity and solution algorithms. Computers & Operations Research, 40, 2071-2082.

[3] Allahverdi, A., Cheng, C.T., Mikhail, T.C.E., Kovalyov, Y., 2008. A survey of scheduling problems with setup times or costs. European Journal of Operational Research, 187, 985-1032.

[4] Alptekinoglu, A., Tang, C.S., 2005. A model for analyzing multichannel distribution systems. European Journal of Operational Research, 163 (3), 802–824.

[5] Avriel, M., Diewert, W. D., Schaible, S., Zang I., 1988. Genaralized Concavity.

[6] Baker, K.R., 1990. Scheduling groups of jobs in the two machine flow shop. Mathematical and Computer Modeling, 13, 29-36.

[7] Baker. K.R., Bertrand, J.W.M., 1981. A comparison of due-date selection rules. AUE Transactions, 13(2) 123-131.

[8] Bean, J.C., 1994. Genetic algorithms and random keys for sequencing and optimization. Operations Research Society of America Journal of Computing, 6, 154-160.

[9] Bookbinder, J.H.,Noor, A.I., 1985. Setting job-shop due-dates with service-level constraints. Operations Research Society. 36(11) 1017-1026.

[10] Borodin, A., El-Yaniv, R., 1998. Online Computation and Competitive Analysis. Cambridge University Press, U.K.

[11] Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press, Oxford, UK.

[12] Bozorgirad, M., Logendran, R., 2013. Bi-criteria group scheduling in hybrid flow shops. International Journal of Production Economics, 145, 599-612.

[13] Chand, S., Chhajed, D., 1992. A single machine model for determination of optimal due dates and sequence. Operations Research, 40(3) 596-602.

[14] Chang, P., Hieh, J., Liao, T.W.,2005. Evolving fuzzy rules for due date assignment problem in semiconductor manufacturing factory. Journal of Intelligent Manufacturing, 16(4–5), 549–557.

[15] Chen, A., Chris, N., Potts, B., Vitaly, A., 1998. Approximation algorithms for two machine flow shop scheduling with batch setup times. Mathematical Programming, 82, 255-271.

[16] Chen, J. Zhang, H., Sun, Y., 2012. Implementing coordination contracts in a manufacturer stackelberg dual channel supply chain. Omega, 40 (5), 571-583.

[17] Cheng, G., Wang, 1998. Batching and scheduling to minimize the makespan in the two machine flow shop. IIE Transactions, 30(5), 447-453.

[18] Cheng, G., Wang, 1999. A Note on two machine flexible manufacturing cell scheduling. IEEE Transactions on Robotics and Automation, 15(1), 187-190.

[19] Cheng, T.C.E., Lin, B.M.T., Toker, A., 2000. Makespan minimization in the two machine flow shop batch scheduling problem. Naval Research Logistics, 47,128-144.

[20] Cheng, T.C.E., Gupta, M.C., 1989. Survey of scheduling research involving due date determination decisions. European Journal of Operational Resaerch, 38,156-166.

[21] Chiang, W-Y.K., Monahan, G.E., 2005. Managing inventories in a two-echelon dual-channel supply chain. European Journal of Operational Research, 162 (2), 325–341.

[22] Chuanli, Z., Hengyong, T., 2012. Two machine flow shop scheduling with deteriorating jobs and chain precedence constraints. International Journal of Production Economics, 136,131-136.

[23] Danneberg, D., Tautenhahn, T., Werner, F., 1998. A comparison of heuristic algorithms for flow shop scheduling problems with setup times and limited batch size. Preprint, Magdeburg, Germany: Otto-von-Guericke University.

[24] De Koster, R., Le-Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. European Journal of Operational Research, 182 (2), 481–501.

[25] Duenyas, I., 1995. Single facility due date setting with multiple customer classes. Management Science, 41(4) 608-619.

[26] Edwin, T. C., Jatinder, N. D., Gupta, Wang, G., 2000. A review of flow shop scheduling research with setup times. Production and Operation Management, 9, 262-282.

[27] Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. Annals of Discrete Mathematics, 5, 287-326.

[28] Ham, I., Hitomi, K., Yoshida, T., 1985. Group technology. Kluwer-Nijhoff Publishing, Boston, MA.

[29] Handfield, R.B., Ernest, L., Nichols, J.r., 1999. Introduction to Supply Chain Management. Prentice Hall, NJ.

[30] Hendizadeha, S.H., Faramarzib, H., Mansouric, S.A., Gupta, N.D.J., ElMekkawy, T.Y., 2008. Meta-heuristics for scheduling a flow line manufactur-

ing cell with sequence dependent family setup times. International Journal of Production Economics, 111, 593-605.

[31] Hill, A., Collier, D., Froehle, C., Goodale, J., Metters, R., Verma, R., 2002. Research opportunities in service process design. Journal of Operations Management, 20,189–202.

[32] Hsu, S.Y., Sha, D.Y., 2004. Due date assignment using artificial neural networks under different shop floor control strategies. International Journal of Production Research, 42(9), 1727–1745.

[33] Johnny, C., Jatinder, N., Gupta, D., 1992. Flow shop scheduling with dominant machines. Computers and Operations Research, 22, 237-246.

[34] Johnson, S.M., 1954. Optimal two and three-stage production schedules with setup times included. Naval Research Logistics Quarterly, 1, 61-68.

[35] Kaminsky, P., Kaya, O., 2005. Centralized versus decentralized scheduling and due date quotation in a make-to-order supply chain. Proceedings of the 2005 MSOM Conference, IL, USA, 149–159.

[36] Kaminsky, P., Lee, Z. H., 2008. Effective on-line algorithms for reliable due date quotation and large-scale scheduling. Journal of Scheduling, 11(3), 187–204.

[37] Kaminsky, P., Hochbaum, D., 2004. Due date quotation models and algorithms. In J. Y. Leung (Ed.), Handbook on scheduling algorithms, methods and models. London: Chapman Hall/CRC.

[38] Kapuscinski, R., Tayur, S., 1997. 100% reliable quoted lead times. GSIA Working Paper, Carnegie Mellon University, Pittsburgh, PA.

[39] Keskinocak, P., 1997. Satisfying customer due dates effectively. Ph.D. thesis, GSIA, Carnegie Mellon University, Pittsburgh, PA.

[40] Keskinocak, P., Ravi, R., Tayur, S., 2001. Scheduling and reliable lead-time quotation for orders with availability intervals and lead-time sensitive revenues. Management Science, 47(2), 264–279.

[41] Keskinocak, P., Tayur, S., 2003. Due Date Management Policies. Handbook of Quantitative Supply Chain Analysis, 485-554.

[42] Kleinau, U., 1993. Two machine shop scheduling problems with batch processing. Mathematical and Computer Modelling, 17, 55-66.

[43] Lee, E., Mirchandani, P. B., 1988. Concurrent routing, sequencing, and setups for a two machine flexible manufacturing cell. IEEE Journal of Robotics and Automation, 4, 256-264.

[44] Lin, B.M.T., Cheng, T.C.E., 2001. Batch scheduling in the no-wait two machine flow shop to minimize the makespan. Computers and Operations Research, 28, 613-624.

[45] Lin-hui Sun, A., Lin-yanyun, B.C.F.G., Ji-bowangb, E.G.N., 2012. Flow shop makespan minimization scheduling with deteriorating jobs under dominating machines. International Journal of Production Economics, 138,195-200.

[46] Logendran, R., Sriskandarajah, C., 1993. Two machine group scheduling problem with blocking and anticipatory setups. European Journal of Operational Research, 69, 467-481.

[47] Logendran, R., Deszoeke, P., Barnard, F., 2006. Sequence dependent group scheduling problems in flexible flow shops. International Journal of Production Economics, 102, 66–86.

[48] Marco, P., 2004. Batch scheduling in a two machine flow shop with limited buffer and sequence independent setup times and removal times. European Journal of Operational Research, 153, 581-592.

[49] Niels, A.H., Agatz, Moritz, F., Jo, A.E.E.,2008. E-fulfillment and multi-channel distribution-A review. European Journal of Operational Research, 187(2) , 339-356.

[50] Nouweland, A., Krabbenborg, M., Potters, J., 1992. Flow shops with a dominant machine. European Journal of Operational Research, 62, 38-46.

[51] Qi, X., Bard, J.F., Yu, G., 2004. Supply chain coordination with demand disruptions.Omega , 32, 300–312.

[52] Ragatz, G.L., Mabert, V.A., 1984. A simulation analysis of due date assignment rules. Operations Management, 5(1), 27-39.

[53] Rub, R., Concepci, M., Javier, A., 2005. Solving the flow shop scheduling problem with sequence dependent setup times using advanced metaheuristics. European Journal of Operational Research, 165, 34-54.

[54] Russell, M.L., Meller, R.D., 2003. Cost and throughput modeling of manual and automated order fulfillment systems. IIE Transactions 35 (7), 589–603.

[55] Sekiguchi, Y., 1983. Optimal schedule in a GT-type flow shop under series-parallel precedence constraints. Journal of Operational Research Society of Japan, 26(3), 226-251.

[56] Stalk, J. G., T.H. Hout. 1990. Competing Against Time. The Free Press, New York.

[57] Tarn J, Razi, M.M., Wen, H.J., Perez, J.R., 2003. E-fulfillment: the strategy and operational requirements. Logistics Information Management, 16, 350–62.

[58] VoXa, S., Wittb, A., 2007. Hybrid flow shop scheduling as a multi-mode multi-project scheduling problem with batching requirements:A real-world application. International Journal of Production Economics, 105, 445-458.

[59] Weeks, J.K., 1979. A simulation study of predictable due-dates. Management Science, 25(4) 363-373.

[60] Wlodzimierz, S., 1977. Special cases of the flow shop problem. Naval Research Logistics Quarterly, 24, 483-492.

[61] Xiao, T., Qi, X., 2008. Price competition, cost and demand disruptions and coordination of a supply chain with one manufacturer and two competing retailers.Omega, 36, 741–53.

[62] Xu, P.J., 2005. Order Fulfillment in Online Retailing: What Goes Where. Ph.D. Thesis. Sloan School of Management. Massachusetts Institute of Technology.

[63] Yang, D.L., Chern, M.S., 2000. Two machine flow shop group scheduling problem. Computers and Operations Research, 27, 975-985.

[64] Yao D. Q., J. J. Liu, 2005. Competitive pricing of mixed retail and e-Tail distribution channels, Omega, 33 (1), 235-247.

[65] Yao, D., Yue, X., Mukhopadhyay, K.S., Wang, Z., 2009. Strategic inventory deplyment for retail and e-tail stores.Omega, 37, 646-658.

[66] Yoshida, T., Hitomi, K., 1979. Optimal two-stage production scheduling with setup times separated. IIE Transactions, 11(3), 261-263.

[67] Zdrzalka, S., 1995. Two machine flow shop scheduling problem with family setup times. Institute of Engineering Cybernetics, Technical University of Wroclaw, Poland, 16-95.

[68] Zheng, F., Xu, Y., Zhang, W.E., 2014. Online scheduling with immediate and reliable lead-time quotation, Journal of scheduling, 17, 225-236.

VITA AUCTORIS

NAME: Nooshin Nekoiemehr

PLACE OF BIRTH: Isfahan, Iran

YEAR OF BIRTH: 1984

EDUCATION:
September 2011 to the present: Ph.D. candidate, Department of Industrial and Manufacturing Systems Engineering (IMSE), University of Windsor, Windsor, Ontario, Canada.

Dissertation subject: Due date quotation and delivery schedule in dual channel supply chain.

2006-2009: M.Sc. Industrial Engineering, Isfahan University of Technology, Isfahan, Iran. Thesis subject: Scheduling Problem Considering Setup Times In Just-In-Time Systems.

2002-2006: B.Sc. Industrial Engineering, Isfahan University of Technology, Isfahan, Iran.

PEER REVIEWED JOURNAL PUBLICATIONS:

1) **Nekoiemehr N**., Zhang G., Selvarajah E., "Due Date Quotation in Dual Channel Supply Chain". Submitted to Manufacturing & service operations management (M&SOM), August 2015.
2) **Nekoiemehr N**., Selvarajah E., Zhang G., "Scheduling of jobs with cross families in two stage manufacturing systems". International Journal of Production Economics, (2015), 167- 88-96.
3) **Nekoiemehr N**., Moslehi G., " Minimize the Sum of Maximum Earliness and Tardiness in the Single Machine Scheduling Problem with Sequence-dependent Setup Time". Journal of Operational Research Society, (2011), 62, 1403–1412.
4) **Nekoiemehr N**., Taslimi E., Kavosh S., "Design and implementation of Comprehensive Inventory Control System for spare parts in Mobarakeh Steel Company, case-study". Automation & Instrumentation Magazine, (2010), 20:32-38.
5) Khayambashi, E., **Nekoiemehr, N**., Arbabshirani, B., Aslani, S., Integrated Model Of Balanced Scorecard with Multiple Criteria Decision Making Method, FANP. Scientific Sharif Magazine. (2010), 49-58.