3-4-2011

The Order Selection and Lot Sizing Problem in the Make-to-Order Environment

Zhongping Zhai

Florida International University, zhaizhongping@hotmail.com

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

THE ORDER SELECTION AND LOT SIZING PROBLEM

IN THE MAKE-TO-ORDER ENVIRONMENT

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

INDUSTRIAL AND SYSTEMS ENGINEERING

by

Zhongping Zhai

2011

To: Dean Amir Mirmiran
      College of Engineering and Computing

This dissertation, written by Zhongping Zhai, and entitled The Order Selection and Lot Sizing Problem in the Make-to-Order Environment, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

_____
Purushothaman Damodaran

_____
Tao Li

_____
Shih-Ming Lee

_____
Chin-Sheng Chen, Major Professor

Date of Defense: March 4, 2011

The dissertation of Zhongping Zhai is approved.

_____
Dean Amir Mirmiran
College of Engineering and Computing

_____
Interim Dean Kevin O'Shea
University Graduate School

Florida International University, 2011

ACKNOWLEDGMENTS

ABSTRACT OF THE DISSERTATION

THE ORDER SELECTION AND LOT SIZING PROBLEM

IN THE MAKE-TO-ORDER ENVIRONMENT

by

Zhongping Zhai

Florida International University, 2011

Miami, Florida

Professor Chin-Sheng Chen, Major Professor

This research is motivated by the need for considering lot sizing while accepting customer orders in a make-to-order (MTO) environment, in which each customer order must be delivered by its due date. Job shop is the typical operation model used in an MTO operation, where the production planner must make three concurrent decisions; they are order selection, lot size, and job schedule. These decisions are usually treated separately in the literature and are mostly led to heuristic solutions.

The first phase of the study is focused on a formal definition of the problem. Mathematical programming techniques are applied to modeling this problem in terms of its objective, decision variables, and constraints. A commercial solver, CPLEX is applied to solve the resulting mixed-integer linear programming model with small instances to validate the mathematical formulation. The computational result shows it is not practical for solving problems of industrial size, using a commercial solver.

The second phase of this study is focused on development of an effective solution approach to this problem of large scale. The proposed solution approach is an iterative process involving three sequential decision steps of order selection, lot sizing, and lot

scheduling. A range of simple sequencing rules are identified for each of the three sub-problems. Using computer simulation as the tool, an experiment is designed to evaluate their performance against a set of system parameters.

For order selection, the proposed weighted most profit rule performs the best. The shifting bottleneck and the earliest operation finish time both are the best scheduling rules. For lot sizing, the proposed minimum cost increase heuristic, based on the Dixon-Silver method performs the best, when the demand-to-capacity ratio at the bottleneck machine is high. The proposed minimum cost heuristic, based on the Wagner-Whitin algorithm is the best lot-sizing heuristic for shops of a low demand-to-capacity ratio. The proposed heuristic is applied to an industrial case to further evaluate its performance. The result shows it can improve an average of total profit by 16.62%. This research contributes to the production planning research community with a complete mathematical definition of the problem and an effective solution approach to solving the problem of industry scale.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1   INTRODUCTION

1.1   Background and Motivation

The operation modes of manufacturing enterprises are classified into two main categories: make-to-stock (MTS) and make-to-order (MTO). The basic distinction between them is timing of production for customer orders. In the MTS mode, production plan is based on demand forecasts. In the MTO mode, production starts only after a customer order is received. In the market place of rapidly increasing global competition, MTO is gaining its popularity because it addresses individual needs by mass-customizing each product (Chen 2006). MTO orders may vary significantly on their routings, material requirements, and engineering tooling, etc. Due to the production nature of wide product/process variety and small quantity, job shop is the typical operation model used in an MTO operation.

To promptly respond to customer demands, detailed production scheduling is important to MTO operations to meet rigid delivery commitment. (Drexl and Kimms 1997) consider it as a concurrent sizing and scheduling problem, assuming accepting all orders. Therefore two questions to be answered are when and how many products to be produced over the planning horizon of multiple time periods. Typical lot sizing problems consider setup cost and holding cost (Jans and Degraeve 2005). Setup cost is associated with preparing the machine for processing. Holding cost is the expense spent on maintaining goods in stock. The total setup cost decreases as the lot size goes up and the number of setups goes down. However, the holding cost goes up along with the inventory level. It incurs no holding cost if the exact amount is produced that satisfies every delivery commitment (lot-for-lot); however, the total setup cost may increase as more

setups are needed. The primary objective for the lot sizing problem is thus to balance between the setup cost and the holding cost. Aside from cost-saving, lot sizing can also be used to improve order feasibility (Low et al. 2004). For example, lot sizing can be applied to splitting a large production order into smaller production lots, nesting them in a schedule such that the overall lead time is reduced.

In the MTO environment, there is an increase demand for on-time delivery. On-time delivery helps the customer reduce inventory and ensure an effective supply chain. Consequently on-time delivery of orders has become important to MTO customers (Charnsirisakskul et al. 2004). In an MTO operation, incoming orders are reviewed periodically (per day or week). When incoming orders exceed shop capacity, rejection of incoming has to come to play; or expected delivery commitment needs to be renegotiated. In the meantime, lot sizing and detailed scheduling need to be exercised to ensure schedule feasibility. As a result, the MTO production planner needs to make three decisions in concurrence: (1) which incoming customer orders to select, (2) how to split each order (if selected) into production lots, and (3) how to schedule each production lot in a job shop.

1.2   Problem Description

This study focuses on the problem of selecting a subset of incoming customer orders to maximize the total profit, while meeting the deadline of each selected order. Each customer order comes with one delivery (product) item only. The product routing is known and fixed. Each order may prescribe more than one delivery date for a fixed quantity. Each commitment is viewed as a delivery deadline; no late delivery is allowed. Job shop is used as the production mode. Each operation in the routing requires a setup in

2

addition to its processing time, which is proportional to the production lot size. Setup cost is usually defined by setup time and unit setup cost. In this problem, both setup cost and time are fixed for each machine type and order type. Therefore, the time on machine required for a production lot consists of its setup time and processing time. A production lot may be scheduled over multiple time intervals in the planning horizon. Each production lot travels down its routing as a whole. It cannot be split. It can be one machine at a time, and one machine can process one lot at a time. This problem also considers inventory, whose cost is defined by inventory level and length of holding. The WIP is not considered as inventory until it becomes finished goods when the last operation is completed.

The objective of this problem is to maximize the total profit, which is defined as the initial profit of selected orders minus lot sizing cost, which consists of setup and holding costs. The initial gross profit for a customer order is defined as the price committed by the customer for the order minus the fixed manufacturing costs as defined in the routing, which does not include setup and holding costs.

This problem assumes (1) the production system is stable and there is no machine breakdown; (2) the inventory cost for work in process (WIP) is negligible; (3) the buffer space between different stages is assumed to be infinite; (4) each production lot is processed as a batch, which moves in a lot; (5) preemption and re-circulation are not allowed; and (6) all processing and setup times are deterministic.

This research is intended to study the integrated problem of order selection, lot sizing and job shop scheduling. Figure 1-1 demonstrates the nature of the decision problem, whose input is a set of incoming orders. Each order comes with a specific

delivery commitment (i.e., due dates and quantity for each due date), customer committed price, and production routing. The decisions are order selection, lot size and a detailed production schedule. The decision starts with order selection. The initially selected orders are fed to lot sizing decision, which are in turn fed for detailed scheduling. The decisions are looped back to improve feasibility and total profit. Adjusting lot size may change lot sizing cost and thus order selection decision.
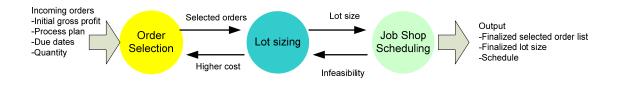
Figure 1-1 The order selection, lot sizing and job shop scheduling process

The following example illustrates the dynamics among order selection, lot sizing and job shop scheduling decisions. This problem considers three orders in a planning horizon of three time periods. Each time period is 10 hours. Other data for this problem are summarized in Table 1-1. The order quantity for order A is 10; the due date is in 20th hour. The routing is machine 1 first and then machine 2. The holding cost is $5 per unit (item) per period of time. The unit processing time for each product on each machine is fixed for one hour. Its setup times and costs for both machines 1 and 2 are 1 hour and $15. For order B, there are two deliveries. The first one is in 20th hour for 2 units and in 30th hour for another 5 units.

4

Table 1-1 Example data

Order	Initial Gross Profit	Delivery 1		Delivery 2		Route	Holding cost	Setup time		Setup cost	
		QTY	Due date	QTY	Due date			m1	m2	m1	m2
A	500	10	20			m1→m2	5	1	1	15	15
B	500	2	20	5	30	m2→m1	5	2	2	25	25
C	200	6	20			m1	10	1	-	5	-

If order A is processed as one lot (lot size=10), the processing time on each machine will be 11 hours including one hour of setup. Therefore, the flow time will be 22 hours, which apparently exceeds its due date. Thus, it is infeasible to complete order A by its due date. Therefore, only the other two orders can be further considered. Figure 1-2 shows a feasible schedule of orders B and C. Both meet their due dates. The notation B3, for example, denotes the production lot of order B to be completed in the third period. There are two production lots of order B and one lot of order C. There incurs no inventory and thus no holding cost in this production schedule. The total lot sizing cost in this example consists of only the setup cost which is $110. The optimal total profit is $590.
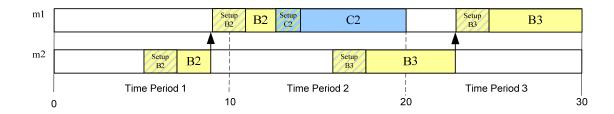
Figure 1-2 An intuitive feasible solution to the example instance

One way to improve the solution in Figure 1-2 is to combine the two demands for order B into one production lot (lot size of 7). When the production lot is to be completed at the second period, the total lot sizing cost will be reduced by $25. By applying the

critical ratio rule, B2 is scheduled before C2, for both are due in the second period and yet B2 needs longer processing time than C2. If so, the earliest finish time for order C is in the 25th hour as shown in Figure 1-3. The schedule is infeasible because order C is late for 5 hours. If order C is scheduled before order B, then a feasible schedule exists as shown in Figure 1-4. The total profit for this case is $615.
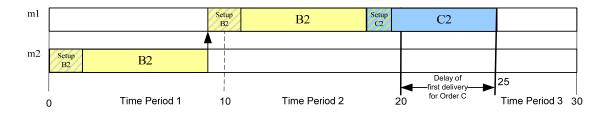
Figure 1-3 An infeasible schedule of scheduling the most critical operation first
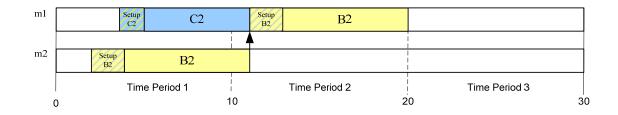
Figure 1-4 An improved solution to the example instance

Both feasible schedules do not include order A due to its long flow time. However, its flow time may be shortened if the order is divided into smaller production lots and scheduled in parallel (Lixiang and Giachetti 2008). Figure 1-5 shows order A is divided into two production lots of A1 and A2 with x_1 and x_2 as their lot size. A feasible schedule is shown in the figure with $x_1 = 3$ and $x_2 = 7$. A2 is scheduled to complete at 20th hour. In principle, x_1 should be as smaller as possible, to minimize its holding cost.

Figure 1-6 considers adding orders B and C to the schedule, after order A has been scheduled as two production lots. Under this situation, only order C can be

scheduled. Though order B is more profitable, it is not feasible, assuming lot-for-lot scheduling is practiced. The figure shows the completion for B3 at 34th hour, a delay of 4 hours.
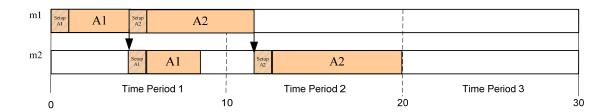
Figure 1-5 A feasible schedule for single order with two production lots
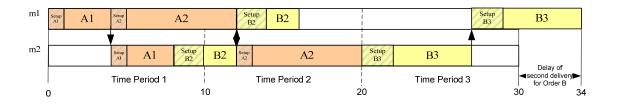
Figure 1-6 An infeasible schedule for both order A and B

The infeasibility problem can be solved by moving 2 units from B3 to B2 and concurrently swapping the schedule for B2 and A1. Figure 1-7 shows a feasible for orders A and B. The total profit is $750, which is the best solution so far.
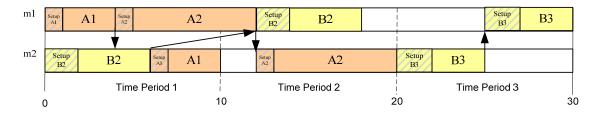
Figure 1-7 A feasible schedule for both order A and B

7

The example demonstrates that each decision among order selection, lot sizing and scheduling can affect other decisions and the objective of optimal profit. Any individual decision on order selection, lot sizing and scheduling may not lead to the optimum. With in mind that the trial-and-error approach is inefficient especially when dealing with large size problems, this research aims at a thorough study of this concurrent order selection, lot sizing and scheduling problem.

1.3   Research Objective and Methodology

The primary objective of this research is to formally define this problem and develop an effective solution technique for solving this problem of large size. The first phase of this study focuses on an analytical definition of this job shop problem in concurrence with lot sizing and order selection consideration. The problem is modeled as a mixed integer linear program (MILP) with its objective, decision variables, and constraints. The proposed model is solved with a commercial solver, CPLEX. The model is solved to optimum for small problems to evaluate its behavior and performance.

The lot sizing part of the problem is considered an NP hard problem by (Chen and Thizy 1990). The job shop scheduling component is identified as a NP hard problem by (Blazewicz et al. 1996). Therefore, the problem that integrates order selection, lot sizing and job shop scheduling is also an NP-hard problem. It is infeasible to solve this problem analytically when the problem size is large. The second phase of this study thus is focused on development of an effective solution technique for this problem of large size.

Heuristics are commonly applied to solve complex problems. They generally lead to good solutions within limited computational time, though they may not be optimal. This study proposes an iterative process for this decision problem of concurrent order

selection, lot sizing and lot scheduling. As part of this study, an experiment is desired to examine the characteristics (including solution quality and run time) of each decision under various simple heuristics and rules.

The results are compared to optimal solutions and/or upper bounds, generated from the commercial solver CPLEX. They are used as benchmarks to evaluate the quality of the proposed solution method. In addition, an industrial case is used to further validate the proposed method and its applicability of solving industrial problems.

1.4 Significance and Contributions

Even though there is voluminous research literature in the production planning area, this particular problem of interest integrating order selection, lot sizing, and job shop scheduling has not been studied. Related studies consider at most two of the three decision problems. This research is the first attempt to address this concurrent decision problem. The proposed mathematical model formally defines order selection, lot sizing, and job shop scheduling decisions in concurrence. The mathematical formulation is innovative in modeling its disjunctive constraints as linear constraints, such that the model can be solved with a commercial solver. In addition, the constraint for ensuring a production lot to be completed in a designated time interval is unique, as it relates lot sizing to job shop scheduling decisions.

The proposed heuristic solution method is efficient for solving large-scale problems. It is built on an experiment designed to evaluate performance of simple heuristic rules that are commonly used for these three decisions. The solution approach makes use of these heuristics and rules to improve the decision process. This research also leads to discovery of new rules for solving this problem. Among them, the proposed

weighted most profit rule is the best for order selection and the earliest operation finish time and shifting bottleneck rules are best for job shop scheduling, while the proposed minimum cost-increase rule performs better for lot sizing in a heavily loaded shop. These heuristics alone could help MTO managers to make better order selection, lot sizing and scheduling decisions. In summary, the two major contributions by this research are: (1) the formal definition of the order selection and lot sizing problem in the job shop environment, and (2) an effective solution technique for solving large scale problems.

1.5   Dissertation Outline

The rest of this dissertation is organized as follows. Chapter 2 is a literature review of related research in the public domain. Chapter 3 presents the MILP model with experiments conducted to validate the proposed model. Chapter 4 presents the framework design for the iterative solution approach. It also summarizes a study of various heuristics (both existing and proposed ones) applicable to each of the three decision problems. Chapter 5 is an experiment design and analysis for performance evaluation of the above heuristics under the proposed solution framework. Chapter 6 presents a real-life case used to assess the applicability of this proposed solution approach. Finally, conclusions and future research are summarized in Chapter 7.

2   LITERATURE REVIEW

This research aims to unify three decision problems — order selection, lot sizing and job shop scheduling. The existing literature is divided into three levels, according to the number of decision problems considered. The first level contains the literature that only studies one decision problem. The second level contains the literature that considers two of the three decision problems. Literature considering all of them belongs to the third level. Although there is a large body of literature in the first and the second levels, there is no literature found for the third level. The Venn diagram in Figure 2-1 illustrates corresponding categories of literature based on the levels. The abbreviations for each category will be used throughout this dissertation. The review on the first level is presented in Sections 2.1- 2.3, then followed by the discussion of the second level in Sections 2.4 - 2.6. Section 2.7 presents a summary of related research.
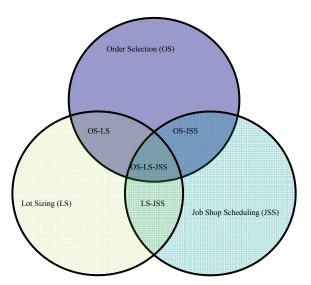
Figure 2-1 Venn diagram of related literature

2.1 Order Selection Problems

Order selection has been a topic of growing interests since Miller (1969), who studies a queuing system with the objective of maximizing the expected value of customer orders. In this review, relevant order selection researches are classified by three criteria: order arrivals, resource setting and selection criteria, as shown in Table 2-1. For dynamic order arrivals, customer demands are described with random distribution; while for static arrivals, customer demands are deterministic. Resource setting refers to the production environment in which selected orders are processed. Selection criteria are a set of objective functions, informed by the tradeoff between rewards obtained from selected orders and the cost of fulfilling them. Herein, meeting latest due date (LDD) refers to that an accepted order must be completed by latest due date; otherwise, it is rejected. Table 2-1 indicates that order selection problems with single resource and tardiness objective attracted most research attentions.

Table 2-1 Order selection literature classification

Criteria	Character	Previous Research
Order arrivals	Dynamic	Miller (1969), Wester (1992), Jalora (2006)
	static	Kern (1990), Slotnick (1996)
Resource setting	Single resource	Kern (1990), Wester (1992), Ten Kate (1995), Akkan (1997), Slotnick (1996), Charnsirisakskul (2004), Jalora (2006), Slotnick (2007), Bilginturk (2007)
	Multiple resources	Hans (2001), Ebben (2005), Roundy (2005)
Selection criteria	Earliness	Akkan (1997), Charnsirisakskul (2004), Jalora (2006)
	Tardiness	Guerrero (1988), Kern (1990), Wester (1992), Slotnick (1996), Akkan (1997), Ghosh (1997), Hans (2001), Lewis (2002), Charnsirisakskul (2004), Bilginturk (2007), Slotnick (2007)
	Meeting LDD	Akkan (1997), Bilginturk (2007), Roundy (2005)

In terms of solution approaches, although queuing theory (Miller 1969), decision theory (Nagraj Balakrishnan 1996), and simulation (Ten Kate 1995) are proposed in literature, formulating mathematical models and applying optimization techniques are more common in order selection literature. In addition, heuristics are developed to solve specific problems in large scales.

Kern and Guerrero (1990) present a conceptual model for demand management in the assemble-to-order environment. They also formulate a MILP model with the objective function of minimizing total cost of lateness, inventory and setup. Slotnick and Morton (1996) explore order selection with weighted lateness penalty. They propose a branch-and-bound method for small-size problems and heuristics for large-size ones. This research is further extended by Lewis (2002) to multi-period scheduling; an optimal dynamic programming algorithm is devised. To achieve overall scheduling feasibility for existing orders and a newly arrived order, Akkan (1997) suggests several practical methods, including backward scheduling, forward scheduling, what-if analysis, minimizing fragment cost and compaction. Charnsirisakskul et al. (2004) develop a mixed integer programming formulation, and use numerical analysis to examine order acceptance, scheduling and due-date setting decisions. In this research, the manufacturer has the flexibility to choose lead-times. Roundy et al. (2005) model a job insertion problem (selected orders are inserted into a set of orders already scheduled) using MILP. They also propose meta-heuristics for this problem, including genetic algorithm (GA), simulated annealing (SA) and Tabu search.

Order acceptance with minimizing weighted tardiness is widely discussed. Slotnick and Morton (2007) examine order acceptance with weighted tardiness penalty.

They present straightforward separation of sequencing and job acceptance, together with a branch-and-bound procedure. Similar problems on order selection with tardiness penalty are solved with SA (Bilginturk et al. 2007) and GA (Rom and Slotnick 2009).

2.2 Lot Sizing

The first lot sizing model is the renowned Economic Order Quantity (EOQ), developed by Harris (1913). Comprehensive survey on modeling lot sizing problem and solution approaches can be found in Maes (1988), Karimi (2003), Quadt (2008) and among others. Lot sizing problems are classified mainly based upon product complexity and existence of resource constraints. If the final product is simply being produced from raw materials, it is referred as a single-level problem. If there exists parent–component relationship among the items, it is regarded as a multi-level problem. When infinite resource capacity is assumed, lot sizing problem is said to be Uncapacitated Lot Sizing Problem (ULSP). On the contrary, if capacity constraints are explicitly stated, the problem is named as Capacitated Lot Sizing Problem (CLSP). Except for single-level ULSP, the other variants of lot sizing problems are strongly NP-hard (Bitran and Yanasse 1982; Chen and Thizy 1990). As the research under study only considers simple products, this review only focuses on the single-level lot sizing problem.

2.2.1 Single-level ULSP

The EOQ model assumes constant demand rate and infinite time horizon. As an extension to EOQ, Wagner-Whitin (WW) algorithm (Wagner and Whitin 1958) applies to time-varying demands and finite discrete planning horizon. It considers all possible alternatives of processing an order in the current or previous periods. Selection of

alternatives is based on a minimum cost policy, under which each lot size is exactly the sum of a set of future demands. To avoid complicated computation of the WW algorithm, some practical heuristics for ULSP are proposed, such as EOQ-MRP, Silver-Meal (S-M), Part-Period Balancing (PPB), Least Unit Cost (LUC), etc. Detailed description of these heuristics can be found in Orlicky (1975), Silver (1985), Eric (1986) and Nahmias (1989). A few researchers also compared the performance among ULSP heuristics. For instance, Blackburn (1980) compares PPB, S-M and WW in a rolling-scheduling environment and revealed that simpler Silver-Meal heuristic can provide a cost performance superior to that of the WW algorithm. Gelders and Wassenhove (1981) state that the priori choice of suitable heuristic depends on variability of demands and particular cost structure at hand. When the demand variability is low, EOQ is suitable; otherwise, S-M heuristics is recommended.

2.2.2 Single-level CLSP

Compared to ULSP, CLSP attract more research interest since the first work of Manne (1958). It is more practical but much more difficult to be solved. Specialized heuristics and mathematical programming based approaches are commonly used to solve CLSP.

(1) Specialized heuristics

Specialized heuristics generally encompass three steps. The first step is lot sizing, which is often based on the ULSP heuristics. For example, Dogramaci (1981) and Gunther (1987) simply employ Lot-for-Lot to generate initial solution. Dixon and Silver (1981) apply S-M method to initialize lot sizes. If the initial lot sizes for all items are constructed from the first period to the last period, it is named as period-by-period

method. This method can be found in the work of Lambrecht (1979), Dixon (1981), and Maes (1986). On the other hand, item-by-item heuristic is proposed by Kirca (1994). The initial plan starts from selecting a single item and planning the item over the entire the planning horizon.

The second step is so-called feasibility routine. It is to ensure that all demands are satisfied without backlogging and capacity constraints are not violated. It is conducted with feedback mechanism or look-ahead mechanism. In the feedback mechanism, excess demands are pushed back to an earlier period with leftover capacity, given that the saving in setup cost can make up the extra holding cost (Lambrecht and Vanderveken 1979). While in the look-ahead method, the minimum required inventory is computed a priori in order to avoid capacity violation in later periods (Dixon and Silver 1981; Maes and Van Wassenhove 1986).

The third step is to improve the existing solution by adjusting lot size. For example, Dixon and Silver (1981) introduce lot elimination, lot merging, lot interchange and use of optimal lot size. Dogramaci et al. (1981) propose left-shift procedure that searches for shifts with the largest reduction in overall cost. Karni and Roll (1982) introduce 10 types of shifts with calculating cost-saving coefficient which is based on the tradeoff between setup and holding costs. Tabu search is also applied to improve CLSP solution; see examples in Hindi (1996) and Karimi et al. (2006). The improvement step involves a large number of shifts and feasibility checking; therefore, it is generally the most time-consuming step in the lot sizing procedure.

(2) Mathematical programming based approaches

The review by Karimi et al. (2003) covers the most important results in exact and approximation algorithms until 2003. They survey commonly used techniques, such as branch and bound, LP-relaxation and network flow algorithm. Recently, Heuvel and Wagelmans (2006) study CLSP with linear costs and present a dynamic programming algorithm that solves the CLSP with special cost function in polynomial time. Absi and Kedad-Sidhoum (2009) consider safety stock and demand shortage in CLSP. They develop a Lagrangian relaxation of the capacity constraints to obtain lower and upper bounds. The resultant uncapacitated problem is modeled as a fixed-charge network and solved with a dynamic programming algorithm.

Verma and Sharma (2010) design two Lagrangian relaxations for CLSP with considering backlogging and setup time. In the first relaxation, CLSP is relaxed to a multi-item ULSP. In the second relaxation, the inventory flow-balance constraint is relaxed; the problem is reduced to a single constraint continuous knapsack problem with an upper bound on the quantity produced.

Compared to specialized heuristics, mathematical programming based methods usually produce solutions with better quality. However, they need more computational efforts, so that they are less applicable to real-life problems.

2.3   Job Shop Scheduling

There are many variants of the JSS problem, according to different scheduling objectives and constraints. Aside from job shop problems with two machines, or with the processing time of operation is either 0 or 1 can be solved in polynomial time, other JSS problems are notoriously NP-hard (Blazewicz et al. 1996; Pinedo 2002). In the problem

under study, lateness is not allowed; therefore, minimizing makespan of a production lot is important for feasibility of scheduling. If a production lot is produced earlier than demanded, holding cost will be incurred. Minimizing earliness cost may contribute to maximizing overall profit. In addition, rejecting an order causes loss of the corresponding revenue. Minimizing this loss is equivalent to minimizing weighted number of tardy jobs (WNTJ) in a job shop scheduling problem. Therefore, the review mainly concentrates on scheduling problems with minimizing makespan, earliness and WNTJ.

For makespan minimization, shifting bottleneck method, developed by Adams (1988), is the most notable approximation algorithm for JSS. With considering job interdependency, Dauzere-Peres and Lasserre (1993) modify Adams' heuristics and obtained better computational performance. More recently, meta-heuristics are applied into JSS problems. For example, Huang and Liao (2008) employ Ant Colony Optimization (ACO) to generate initial solution, which is further improved by applying tabu search iteratively. Zhang et al. (2008) apply a hybrid genetic algorithm for JSS. In their research, genetic algorithm is used for global exploration among the population; local search served as local exploitation around operation-based chromosomes.

Owing to the popularity of Just-in-Time (JIT) concept, scheduling problems involving earliness and tardiness penalties have received considerable research attentions. Early research mainly focuses on single machine systems; see Baker and Scudder (1990) for a survey. Recently, job shop environment is considered. Beck and Refalo (2003) apply a hybrid technique using constraint programming and linear programming to the earliness/tardiness problem in the job shop. Thiagarajan (2005) studies JSS with multi-level jobs, with the objective of minimizing the sum of weighted earliness, weighted

tardiness and weighted flow time of jobs. A set of dispatching rules are presented by incorporating the relative costs of earliness, tardiness and holding of jobs in the form of scalar weights. Philippe (2008) proposes two Lagrangian relaxations of JIT scheduling with relaxation on precedence constraints and machine constraints, respectively.

For WNTJ problem, Karp (1972) establishes the NP-hardness for the single machine system. The special case with common due dates can be viewed as knapsack problem so that the weighted shortest processing time (WSPT) heuristics can be applied (Pinedo 2002). Other than single machine, WNTJ problem are also considered in parallel machine (Ng et al. 2003; M'Hallah and Bulfin 2005), open shop (Brucker et al. 1993; Galambos and Woeginger 1995; Svetlana 2000; Baptiste 2003) and flow shop (Charnsirisakskul et al. 2004). Heuristic solutions are mostly considered in those problems. Limited reserach in job shop can be found in Józefowska et. al (1994), who develop dynamic programming algorithm applying Jackson's indexing method (James 1956). In their work, only job shop with two machines is considered.

These objectives aforementioned are generally considered independently. An exception is Lee (1991), who studies minimizing weighted number of tardy jobs and weighted earliness/tardiness penalties. Their research is under a common due date assumption and "agreeable ratio condition" (if job i is relatively more important than job j, the weight of earliness and tardiness will be greater than that of job j). They proved that the problem is NP-complete in the strong sense, and hence cannot be solved by using any pseudo-polynomial time algorithm.

2.4 Lot Sizing with Order Selection

Although extensive studies have been conducted in the order selection or the lot sizing, extant work integrating them is very limited. Wester et al. (1992) study different order selection heuristics in a single machine system, with order arrivals following Poisson distribution. They consider setup time saving between similar product groups when making a schedule. In the monolithic approach, a new schedule is constructed for orders not yet in production and the new order. The schedule is constructed with a heuristic that minimizes maximum lateness and total setup time sequentially. In the hierarchic approach, re-scheduling of all available orders is based on operation times of scheduled orders, the order to be scheduled, and a work content level chosen from simulation experiments. They also propose priority rules based heuristic named myopic heuristic. It is to select the order that imposes minimum lateness to existing orders. The experiments showed that monolithic approach performs better than the others. Geunes et al. (2002) study order selection with production planning problem, in which different customers order same product with different prices over time periods. They provide a shortest path based solution for uncapacitated order selection problem. With taking account of lot size limit, a Lagrangian relaxation algorithm is proposed for the capacitated order selection problem. In terms of lot sizing, this research only considers single item.

2.5 Order Selection under Job Shop Environments

In the studies on the order selection, job shop is first considered by Ebben et al. (2005), who employ resource loading methods to support the order acceptance decision. They compared four resource loading methods: aggregate resource loading, resource

loading per resource, EDD based order acceptance and Branch-and-Price resource loading. Their experiments indicate that sophisticated approaches significantly outperform the straightforward approaches when there are tight due dates. More recently, Chen et al. (2009) propose a MILP model for capacity planning and order selection problem in the MTO environment. In this research, overtime and outsourcing are considered. The proposed model is further solved by Mestry et al. (2011) with a branch and price approach.

2.6   Integration of Lot Sizing and Job Shop Scheduling

In the classical CLSP, resources used by different products can be simply added up. However, in a job shop environment, workload on each machine depends on detailed schedule of products, which is subject to precedence constraints. Dauzere-Peres and Lasserre (1994) consider an integrated model for lot sizing and scheduling in the job shop. They propose a multi-pass decomposition approach that alternatively solves the integrated problem at two levels. One level is lot sizing for a given sequence of jobs on each machine; the other level is sequencing lots with fixed lot sizes. Their experiments also indicated that a modified shifting bottleneck heuristic can provide a better solution than priority rule-based dispatching methods. Anwar and Nagi (1997) address the integrated scheduling and lot-sizing problem with complex assemblies. The objective is to minimize the cumulative lead time of the production and reduce setup and inventory costs. They propose a two-phase heuristic that addresses both precedence and capacity constraints. Jeong et al. (1999) study a batch splitting method for a job shop scheduling. They employ a modified shifting bottleneck procedure to generate initial schedule and then split a batch to shorten makespan.

2.7   Summary

These three decision problems (order selection, lot sizing and job shop scheduling) are usually treated separately in the literature. The research that integrates two of them is very limited; there is no research that addresses three decision problems simultaneously. Table 2-2 compares similar problems in literature and the problem under study.

Table 2-2 Summary of similar research problems

Research	Objective	Order selection	Lot sizing	Job shop	On-time delivery	Solution
(Mestry et al. 2011)	Maximize profit	Yes	No	Yes	Yes	Branch and price
(Dauzere-Peres and Lasserre 1994)	Minimize cost	No	Multiple items Setup cost Setup time	Yes	No	Heuristics
(Ebben et al. 2005)	Minimize lateness	Yes	No	Yes	Yes	Heuristics/ Branch and price
(Jeong et al. 1999)	Minimize makespan	No	Multiple items Setup time	Yes	No	Heuristics
(Wester et al. 1992)	Minimize lateness	No	Setup time	No	Yes	Heuristics
(Geunes et al. 2002)	Maximize profit	Yes	Single item Setup cost	No	Yes	Network flow/ LP relaxation
Proposed research	Maximize profit	Yes	Multi-items Setup cost Setup time	Yes	Yes	Heuristics

In terms of order selection, the problem under study considers static order arrivals, multiple resources and meeting latest due date of customer demands. There is no lateness cost; but an earliness cost is incurred if a production lot is completed earlier than

demanded. From lot sizing view, the problem under study belongs to the class of single-level CLSP; the capacity is characterized by job shop constraints. Both setup cost and time are explicitly defined; however, they are less addressed jointly in literature. Maximizing total profit relates to on-time completion of production lots and minimizing weighted number of tardy jobs; but there is no research addressing these issues concurrently in job shop scheduling literature.

According to this review, mathematical modeling and developing heuristics are promising in solving complicated order selection and production planning problem. However, most existing models and heuristics strongly depend on corresponding problem definitions. If extant research findings are adopted, modifications are needed to accommodate the characteristics of this problem. In the next chapter, mathematical modeling will be first discussed.

3   A MATHEMATICAL FORMULATION

This chapter proposes a mathematical formulation for the order selection and lot sizing problem in the MTO environment. Section 3.1 presents the mathematical model for the problem under study. The proposed model is verified in Section 3.2. Then the usefulness of this model is illustrated in Section 3.3. Finally, the performance of solving the proposed model with a commercial solver is evaluated with a set of numerical experiments.

3.1   Mathematical Model

The input of the problem under study is a set P of customer orders. Each order $i \in P$ only includes single product. The initial gross profit (r_i), is the price of the order excluded by some fixed production costs (such as labor, utility, and overhead). The production plan is needed only for selected orders over a set of N planning periods. For each period $t \in N$, each demand for customer order i (d_{it}) must be satisfied without delay. This on-time fulfillment is prescribed by fully scheduling production lots on a job shop with a set M of machines. For each machine $k \in M$, the setup time for each order i (τ_{ik}) is given according to process plan. For any production lot, it must be fully processed; therefore the total setup cost incurred from different machines on the route A_i is always fixed. When the last operation of a production lot is completed, inventory is incurred. For each order, the lot sizing cost is the sum of setup cost (setup per lot δ_i) and holding cost (cost rate h_i). As there is no machine breakdown, the capacity of each machine in each planning period always equals to the length of the planning period c.

The objective function of the problem under study is to maximize total profit, which is the initial gross profit excluded by lot sizing cost. More detailed nomenclatures used for the problem formulation are shown as follows.

Indices and sets

i : Order $P = \{1,2..., p\}$ set of orders

t : Time period $N = \{1,2..., n\}$ set of time periods

k : Machine $M = \{1,2..., m\}$ set of machines

Parameters

L_{it} Production lot of producing final product for order i during period t

O_{itk} Operation of lot L_{it} on machine k

h_i Unit holding cost of the product for order i from one period to the next

δ_i Total setup cost for order i over its production route

r_i Initial gross profit obtained from accepting order i

d_{it} Quantity of demand at the end of period t from order i

ρ_{ik} Unit processing time of the product for order i on machine k

τ_{ik} Setup time for order i on machine k

c Length of each time period

M_i Set of machines that can process order i ; $a_i = | M_i |$ denotes the size of M_i

A_i Set of pairs of machines presenting precedence relations for order i

m_i Last machine on the route of order i

P_k Set of orders that machine k can process; $b_k = |P_k|$ denotes the size of P_k

q_{it} Production quantity limit for L_{it}

ρ_i Unit processing of order i on all machines over its route; $\rho_i = \sum\limits_{k \in M_i} \rho_{ik}$

τ_i Setup time of for order i on all machines over its route; $\tau_i = \sum\limits_{k \in M_i} \tau_{ik}$

Decision variables

$$Z_i = \begin{cases} 1, \text{if order } i \text{ is selected} \\ 0, \text{otherwise} \end{cases}$$

X_{it} Lot size of L_{it}

I_{it}^+ Inventory level of the product for order i at the end of time period t

$$Y_{it} = \begin{cases} 1, \text{if setup for product } i \text{ exists at period } t \\ 0, \text{otherwise} \end{cases}$$

$$W_{iti't'k} = \begin{cases} 1, \text{if the sequence is } O_{i't'k} \text{ to } O_{itk} \\ 0, \text{otherwise} \end{cases}$$

S_{itk} Start time of operation O_{itk}

F_{itk} Finish time of operation O_{itk}

The mathematical formulation for the problem under study is presented below.

$$\text{Maximize } \sum_{i \in I} Z_i r_i - \sum_{i \in P} \sum_{t \in T} \delta_i Y_{it} - \sum_{i \in P} \sum_{t \in T} h_i I_{it}^+ \tag{3.1}$$

Subject to

$$X_{it} + I^+_{i,t-1} - Z_i d_{it} = I^+_{it} \qquad \forall i \in P, \forall t \in N \tag{3.2}$$

$$X_{it} \le q_{it} Y_{it} \qquad \forall i \in P, \forall t \in N \tag{3.3}$$

$$F_{itk} = S_{itk} + \rho_{ik} X_{it} + \tau_{ik} Y_{it} \qquad \forall i \in P, \ \forall t \in N, \ \forall k \in M_i \tag{3.4}$$

$$S_{itk} \ge F_{itk'} \qquad \forall i \in P, \ \forall t \in N, \forall k' -> k \in A_i \tag{3.5}$$

$$S_{i't'k} \ge F_{itk} - ct W_{iti't'k} \quad \forall i \in P_k, \forall t \in N, \ \forall i' \in P_k, \forall t' \in N \ \text{with} \ i \ne i' \ \text{or}$$
$$t \ne t' \tag{3.6}$$

$$W_{iti't'k} + W_{i't'itk} = 1 \quad \forall i \in P_k, \ \forall t \in N, \ \forall i' \in P_k, \ \forall t' \in N \ \text{with} \ i = i' \ \text{and}$$
$$t < t' \ \text{or} \ i > i' \tag{3.7}$$

$$F_{itm_i} \ge Y_{it}(t-1)c \qquad \forall i \in P, \forall t \in N \tag{3.8}$$

$$F_{itm_i} \le nc - Y_{it}(n-t)c \qquad \forall i \in P, \forall t \in N \tag{3.9}$$

$$I^+_{i0} = 0 \qquad \forall i \in P \tag{3.10}$$

$$I^+_{in} = 0 \qquad \forall i \in P \tag{3.11}$$

$$X_{it}, \ I^+_{it}, S_{itk}, F_{itk} \ge 0 \tag{3.12}$$

$$X_{it}, \ I^+_{it} \ \text{are integer} \tag{3.13}$$

$$Y_{it} = \begin{cases} = 1 & \text{if product } i \text{ is setup for producing } i \text{ at period } t \\ = 0 & \text{otherwise} \end{cases} \tag{3.14}$$

$$Z_i = \begin{cases} = 1 & \text{if order } i \text{ is selected} \\ = 0 & \text{otherwise} \end{cases} \tag{3.15}$$

$$W_{iti't'k} = \begin{cases} = 1 & \text{if the sequence is } O_{i't'k} \text{ to } O_{itk} \\ = 0 & \text{otherwise} \end{cases} \tag{3.16}$$

Expression (3.1) shows the objective function, in which total initial gross profit, setup cost and holding cost are denoted with $\sum_{i \in P} Z_i r_i$, $\sum_{i \in P} \sum_{t \in N} \delta_i Y_{it}$ and $\sum_{i \in P} \sum_{t \in N} h_i I_{it}^+$, respectively. Constraints (3.2) define inventory balance. They express that if an order i is selected, the entering inventory ($I_{i,t-1}^+$) added the current period production (X_{it}) are used to satisfy the demand (d_{it}); what remains is the inventory at the end of current period (I_{it}^+). Constraints (3.2) and Constraints (3.12) (no inventory shortage) ensure that all accepted orders are satisfied on time. The coupling between setup and production is described in Constraints (3.3). If there is a production lot, a setup is needed. The lot size limit q_{it} is set in two ways. First, X_{it} cannot exceed the total demand of order i, i.e.,

$X_{it} \leq \sum_{j=1}^{n} d_{ij}$. Second, for L_{it} , the largest unit processing time on machines forces that the maximum allowable lot size is $\dfrac{ct}{\max_k \{\rho_{ik}\}}$. Therefore, $q_{it} = \min\{\dfrac{ct}{\max_k \{\rho_{ik}\}}, \sum_{j=1}^{n} d_{ij}\}$.

Constraints (3.4) transfer lot sizing into operational level. A setup time is required when a production lot is processed; the processing time is proportional to lot size. These constraints also indicate no preemption is allowed. Constraints (3.5) state the precedence constraints. For each order i, operations follow a predefined route with k' as the precedent of k. The succeeding operation can only start after its precedent is completed. Constraints (3.6) and (3.7) are disjunctive constraints, which ensure that any two operations cannot be processed simultaneously on the same machine. They are derived from the following constraints:

$$s_{itk} \geq f_{i't'k} \quad or \quad s_{i't'k} \geq f_{itk} \qquad i \neq i' \ or \ t \neq t' \qquad (3.17)$$

28

Constraints (3.17) cannot be handled by most of commercial programs, so they are reformulated by introducing auxiliary variable $W_{iti't'k}$, which indicates the sequence of two operations O_{itk} and $O_{i't'k}$. If the sequencing variable $W_{iti't'k} = 0$, Constraints (3.6) will result in $S_{i't'k} \geq F_{itk}$, i.e., $O_{i't'k}$ is processed after O_{itk}. If $W_{iti't'k} = 1$, Constraints (3.6) turn out to be redundant, because the right hand side is non-positive. However, Constraints (3.7) will enforce $W_{i't'itk} = 0$, which in turn impose $S_{itk} \geq F_{i't'k}$, according to Constraints (3.6). These two situations are illustrated in Figure 3-1.

$$
\begin{array}{cc}
k & \fbox{O_{itk}} \rightarrow \fbox{$O_{i't'k}$} \qquad\qquad k & \fbox{$O_{i't'k}$} \rightarrow \fbox{O_{itk}}
\end{array}
$$

(a) $W_{iti't'k}=0$ (b) $W_{iti't'k}=1$
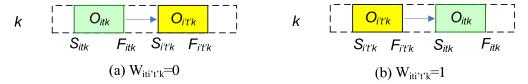
Figure 3-1 Illustration of sequencing variables and disjunctive constraints

Constraints (3.8) and (3.9) reflect the link between lot sizing and scheduling. A production lot is completed only when the last operation is completed. For production lot L_{it}, the completion time should fall inside period t. Constraint (3.10) and (3.11) show that there is no inventory at the beginning or end of the planning horizon. They reflect the basic MTO characteristic that production is only triggered by customer orders and produced products are all used to satisfy customer demands.

Constraints (3.12) impose non-negativity constraints for the lot size, inventory and scheduling variables. Constraints (3.13) further enforce that lot size and inventory variables are integers only. Finally, Constraints (3.14) ~ (3.16) impose the binary restrictions on decision variables *Y, Z* and *W*, respectively.

Table 3-1 Number of decision variables for the proposed model

Variables	Size
X_{it}	np
Y_{it}	np
Z_i	P
I_{it}^+	np
S_{itk}	$n\sum_{i\in I} a_i$
F_{itk}	$n\sum_{i\in I} a_i$
$W_{iti't'k}$	$\sum_{k\in M} nb_k(nb_k-1)$

Table 3-1 lists the number of variables. In total, there are $2np$ integer variables (excluding binary variables), $p+np+\sum_{k\in M} nb_k(nb_k-1)$ binary variables and $2n\sum_{i\in P} a_i$ continuous variables. As the number of binary variables is the most important indicator for problem complexity, the model becomes more complex when the number of orders or planning period increases. Also, if more machines are flexible to process multiple items (i.e., b_k is large), there will be more sequencing variables and constraints; consequently, the model will be more complex as well. Figure 3-2 illustrates this relation with assuming each machine can process equal number of orders. It shows number of binary variables increases significantly when n or b_k increases, with fixed p and m.
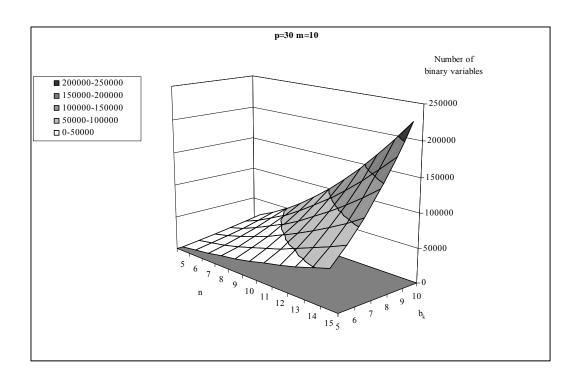
Figure 3-2 Illustration of the increasing of number of binary variables

3.2   Model Verification

The proposed model is coded with AMPL and further solved by CPLEX. These tools are capable of handling MILP optimization. A small-size problem instance is designed to validate the model and verify its optimality. The instance involves three orders being processed on two machines. The planning horizon is divided into three time periods; each period is 10 time units in length. Table 3-2 lists the data for customer orders; Table 3-3 shows job shop related settings. In this instance, the initial gross profit is larger compared expected lot sizing cost, so that an incoming order cannot be obviously rejected. The dynamic demands for each order are specified at each period. The total demand (in time units) is relatively high, especially for the third period. Setup

31

cost and time are significant. The route for each order is unique; different orders share

common machines. Therefore, input data are consistent with the problem definition.

Table 3-2 Customer order related data for the small-size problem instance

i	r_i ($)	h_i ($/unit)	δ_i ($)	d_{it}		
				$t=1$	$t=2$	$t=3$
1	500	1	70	20	10	30
2	800	1	10	0	0	60
3	400	1	50	0	10	30

Table 3-3 Job shop related settings for the small-size problem instance

i	A_i	ρ_{ik}		τ_{ik}	
		$k=1$	$k=2$	$k=1$	$k=2$
1	1→2	0.1	0.1	1	2
2	2→1	0.1	0.1	1	1
3	2	-	0.2	-	2

Figure 3-3 AMPL/CPLEX Solution for the small-size problem instance

By solving the mathematical model, a solution is obtained, as shown in Figure 3-3.

In this solution, all orders are accepted. The feasibility of this solution is verified with the

following aspects: (1) all lot sizes are integer numbers, (2) total production for each order

matches corresponding total demands, (3) each individual demand is ensured with on-

time delivery, (4) schedule of operations follows specified route, (5) no overlapped

operations on any machine, (6) there is no pre-emption, (7) one setup is associated with

each operation, (8) processing time of each operation is proportional to corresponding lot size, and (9) the last operation of any lot is completed in corresponding planning period.

Therefore, the model is verified, as it conforms to problem definition. The optimality of this solution is verified as follows.

(1) Rejecting any order will cause loss of total profit, because all orders are profitable.

(2) Order 1 is planned with the best lot sizing that minimizes total cost. Any change on its lot sizes will increase lot sizing cost.

(3) Order 2 is planned with two lots, which is not optimal in respect of lot sizing. To reduce lot sizing cost, the lot in the second period (lot size=15) should be moved to the third period. However, the lot at the third period is already on a critical path ($O_{111} -> O_{112} -> O_{322} -> O_{232} -> O_{231}$). Even moving one production unit will cause delay of O_{231} for 2 time units, which indicates rejecting order 2 and thus cause loss of profit. Any other changes on production lot of order 2 will decrease total profit as well.

(4) Order 3 is planned with one lot. This is the optimal lot sizing; any change will cause increasing of cost.

(5) Changing the schedule of operations that are not on the critical path does not affect the total profit. For example, if O_{222} is moved to be beginning of the planning horizon, the feasibility can still be sustained.

To sum up, any changes to current solution cannot contribute to increasing of total profit. Therefore, current solution is an optimal solution.

3.3   What-If Analysis Using the Proposed Model

The proposed model not only helps MTO enterprises selectively accept orders and schedule them; it also supports decision-making in response to changes on customer requirements or production environment. To illustrate the usefulness of the proposed model, the problem instance mentioned in previous section is used as a base case in this section.

(1) Increasing of Setup cost

If setup cost is much higher than holding cost, it is preferred to group some lots in order to save the setup cost. For example, setup cost per lot for order 2 is increased as $20, due to the increasing of machine tooling, labor cost, etc. The optimal solution from solving the new model is shown in Figure 3-4. Order 2 is processed with one lot only, so that the minimum lot sizing cost is achieved. However, the resultant production lot uses most of resources in the second period, and then imposes order 3 being process with two lots in the first and the third periods. If the MTO manager still uses the schedule in Figure 3-3, the total profit is $1415, which is lower than the profit of solving the new model. Therefore, this model can help MTO manger make proactive adjustment on production plan when cost structure is changed.

Order#	X_{it}			Schedule			
	t=1	t=2	t=3	m/c #	Period 1	Period 2	Period 3
1	30	0	30	1	O_{111} O_{131}		O_{231}
2	0	0	60	2	O_{312} O_{112}	O_{232} O_{332}	O_{132}
3	10	0	30				
Total profit $1420				setup time manufacturing time			
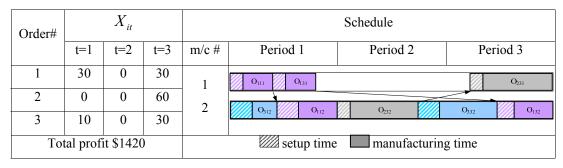
Figure 3-4 Optimal solution after increasing setup cost

(2) Change of engineering

An MTO manager may be faced with change of engineering. For example, if the unit processing time of order 2 on machine 2 is changed to 0.02, for engineering purpose, optimal solution will indicate rejecting order 3 (see Figure 3-5). This is caused by competition of resources between orders. Order 2 appears to be more profitable than order 3, and it requires more resource during the second period on machine 2. As there is no resource to accommodate order 3, it is rejected. Based on this result, the MTO manager may raise the price for order 2 to compensate the loss on order 3.

Order#	X_{it}			Schedule			
	t=1	t=2	t=3	m/c #	Period 1	Period 2	Period 3
1	30	0	30	1	O_{111} O_{131}		O_{231}
2	0	0	60	2	O_{112}	O_{232}	O_{132}
3	Not selected						
Total profit $1140				▨ setup time ▬ processing time			
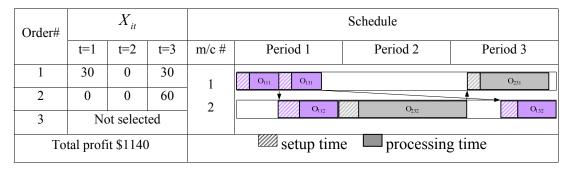
Figure 3-5 Optimal solution after change of engineering

(3) Price negotiation

The MTO manager may wonder what price is profitable for certain order, so that they can accept it. However, due to the complexity of lot sizing and scheduling, it is difficult to estimate production cost accurately when negotiating price with the customer. The proposed model can serve as a reference to such a situation. Significantly decreasing the initial gross profit of an order will result in rejecting this order. For example, the initial gross profit of order 1 is decreased to $150, the optimal solution shown in Figure 3-6 indicates that order 1 is rejected. If the profit for order 1 is adjusted to $200, order 1 is selected; the lot sizes and schedule is same as Figure 3-3. This clearly indicates that the

acceptable price is between \$150 to \$200. Such a what-if analysis can support the MTO manager making decision when quoting an order.

Order#	X_{it}			Schedule			
	t=1	t=2	t=3	m/c #	Period 1	Period 2	Period 3
1	Not selected			1		O_{231}	
2	0	0	60	2	O_{322}	O_{232}	
3	0	40	0				
Total profit \$1110				setup time manufacturing time			
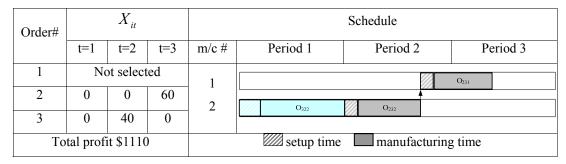
Figure 3-6 Optimal solution after decreasing order initial gross profit

(4) Due date assignment

Customers may inquire if their orders can be fulfilled earlier. The answer can only be made after carefully examining production capacity and potential cost incurred. For example, the customer who placed order 2 expects to get their order delivered at the end of the second period rather than the third period, i.e. $d_{22} = 60, d_{23} = 0$. By solving the proposed model with new parameters on demands, the result is obtained, as shown in Figure 3-7. Although order 2 is still acceptable, but cost is increased by \$35, compared to the base case. If the customer further request expediting order 2 to be delivered at the end of the first period, a new solution is obtained from the proposed model with updated demands (see Figure 3-8). This solution suggests not selecting order 2.

The two situations aforementioned illustrate the proposed model can be used to assist the MTO manager on due date assignment. In the first situation, a higher price may be imposed to the urgent order to make up for loss of profit. In the second situation, it is better to negotiate with customer for a later delivery; otherwise, there is no resource to produce the order and a significant loss of profit will be caused.

36

Order#	X_{it}			Schedule			
	t=1	t=2	t=3	m/c #	Period 1	Period 2	Period 3
1	30	0	30	1	O_{111} O_{131} O_{211} O_{221}		
2	30	30	0	2	O_{212} O_{112} O_{222} O_{322} O_{332} O_{132}		
3	0	10	30				
Total profit $1400				▨ setup time ▮ manufacturing time			
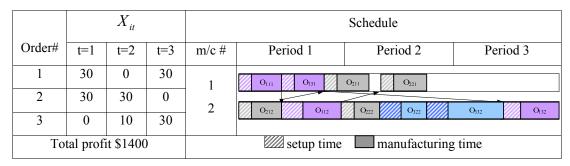
Figure 3-7 Solution for the case with an earlier due date

Order#	X_{it}			Schedule			
	t=1	t=2	t=3	m/c #	Period 1	Period 2	Period 3
1	30	0	30	1	O_{111} O_{131}		
2	Not selected			2	O_{112} O_{322} O_{132}		
3	0	40	0				
Total profit $670				▨ setup time ▮ manufacturing time			
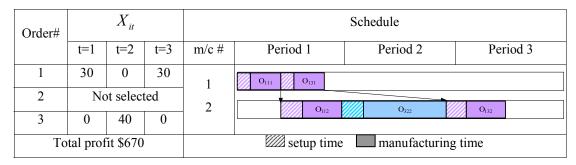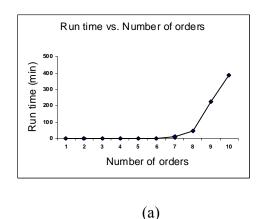
Figure 3-8 Solution for the case with earliest delivery

3.4   Run Time Analysis

As CPLEX uses a branch-and-bound approach to find an optimal solution, it can hardly solve problem instances with large number of integer or binary variables within practical time. To evaluate the performance of using CPLEX to solve the proposed model, a set of large-scale problem instances are tested. The base case is shown in Table 3-4 ($p_{ik} = 0.0005$, $\tau_{ik} = 0.05$, $c = 1$ and $m = 5$). Random problem instances are extended by increasing number of orders (p) and number of periods (n). The number of machines is fixed because resources in a MTO enterprise are relatively stable. Demands are generated with discrete Uniform (0, 100). The route of new order is randomly duplicated from existing orders. For each instance, 20 replicates are generated and solved.

Table 3-4 Base case for large-scale problem instances

i	r_i	h_i	δ_i	A_i	d_{it}				
					t=1	t=2	t=3	t=4	t=5
1	300	0.01	30	1→2→3→4→5	100	100	100	100	100
2	300	0.01	20	2→4→5	0	0	300	0	0
3	500	0.01	30	1→3→4	0	0	0	200	0
4	500	0.01	100	3→2→1	0	0	200	0	200
5	500	0.01	20	5→2→1→3	0	300	0	50	80

(a)

(b)
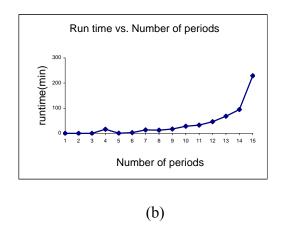
Figure 3-9 CPLEX run time for different problem sizes

Figure 3-9 (a) shows the run time vs. number of orders planned over five periods. Figure 3-9 (b) presents the run time vs. number of periods on which five orders are planned. When the number of orders is more than 8 or the number of periods is more than 10, the run time is over half an hour and increases significantly. The experiments indicate that CPLEX is not suitable for practical use because a practical problem may involve more than 100 orders, 20 periods and 30 machines. A more efficient method is expected to solve the problem under study. Heuristic methods will be proposed to serve for this purpose in the next chapter.

4   HEURISTICS

When the commercial solver CPLEX is used to solve the mathematical model proposed in Chapter 3, it requires a large amount of computational time, even for moderate-scale problem instances. Therefore, the second phase of this research is to explore the applicability of heuristics. Although heuristic methods cannot guarantee an optimal solution, the shorter run time is favorable for practical applications. This chapter includes 6 sections. Section 4.1 provides an overview of heuristic approaches. Sections 4.2 - 4.4 introduce heuristics on order selection, deselecting orders and scheduling, respectively. Finally, two lot sizing procedures are presented in Section 4.5 and Section 4.6.

4.1   Overview of Heuristic Approaches

The input for the problem under study is a set of incoming orders, which trigger production planning; therefore, order selection has to be conducted at the very beginning. Since each customer order includes a unique product and there is no product family, each order has to be treated one by one. Given an order from the set of incoming orders S^c, it has to be processed either into set of selected orders S^+ or set of deselected orders S^-. Therefore, $S^- \cap S^+ = \Phi$ and $S^c = S^- \cup S^+$. If an order in S^c is apparently not profitable or unable to be scheduled, it is directly moved to S^- (see Figure 4-1 arc 1). The other orders can be selected one by one (see Figure 4-1 arc 2). If an order turns out to be not profitable due to a large lot sizing cost, it is moved to S^- (see Figure 4-1 arc 3); otherwise the total profit will be decreased. Aside from possible loss of profit, an order

39

may not be fully scheduled due to resource limit, so it is also deselected to S^- (see Figure 4-1 arc 3), in order to achieve feasibility.
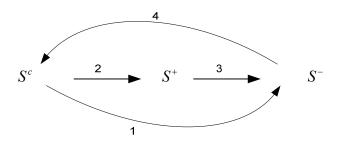
Figure 4-1 Order process direction among sets of orders

When an order is dropped from S^+, the order in S^- can be taken back to S^c (see Figure 4-1 arc 4), if the order S^- is not rejected via arc 1 in Figure 4-1. Consider the following situation. Suppose two orders i and j share a common resource k. Initially, j is fully scheduled, but order i is deselected due to capacity limit of resource k (see Figure 4-2 a) during a planning time length ct (t periods). But in a later iteration, order j is deselected too. This leaves a chance for order i to be scheduled (see Figure 4-2 b). The condition is that the released resource must be more than the production time required by order i. If order i is processed as one lot, it will require least production time on every machine. Therefore, order i may be fully scheduled only when

$$R_{kj} > \tau_{ik} + \sum_{w=1}^{t} d_{iw} .$$
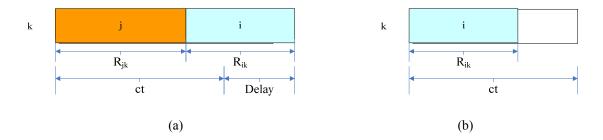
Figure 4-2 Illustration of recourse utilization in deselecting order process

In order to achieve a better profit while ensuring feasibility, lot sizing and scheduling need to be conducted jointly for orders in S^+. Two lot sizing procedures are constructed. One procedure is first conducting lot sizing with least cost and then ensuring scheduling feasibility. Because lot sizing cost is first minimized, this procedure is named as minimum cost heuristic. Another procedure is to conduct lot sizing in a period-by-period manner. Since each period considers minimum cost increasing, it is called as minimum cost increase heuristic. For convenience, they are denoted as LS1 and LS2, respectively.

From the above discussion, the proposed heuristic is an iterative procedure integrating order selection, lot sizing and job shop scheduling. It is described as follows.

Step 1. Select an order from S^c and move it to S^+.

Step 2. Conduct lot sizing and scheduling for S^+.

Step 3. If the schedule is feasible and the profit does not decrease, continue to select the next order. Otherwise, deselect an order from S^+.

Step 4. Take a deselected order back from S^- to S^c. To avoid infinite loop, an order can only go through this step no more than three times.

Step 5. If $S^c = \Phi$, stop; otherwise go to Step 1.

41

In this framework, rules on selecting/ deselecting orders, lot sizing and scheduling are not specified. The following sections in this chapter focus on developing heuristics for each of them. The selection of heuristics will be discussed in the next chapter.

4.2   Order Selection Approaches

4.2.1   Screening of Orders

When an order is associated with significant lot sizing cost or needs long processing time, it is necessary to evaluate if the order can be rejected directly. This screening step at the beginning of the heuristic procedure can avoid unnecessary computations in later steps.

(1)   Screening based on profit

As the total profit is the sum of profit for all selected orders, minimizing cost of every order will contribute to maximizing total profit. Lot sizing for single order without considering scheduling is the uncapacitated lot sizing problem (ULSP), which can be solved to optimality with the WW algorithm. As described in Nahmias (1989), the WW algorithm can be implemented by a one-way network with $n+1$ nodes given there are n periods (see Figure 4-3). For any pair of $i < j$, $arc(i, j)$ represents a setup taking place in period i and the lot size equals to the total demand in period $i, i+1, ..., j-1$. The weight w_{ij} is the lot sizing cost from period i to $j-1$. Therefore, every path from 1 to $n+1$ corresponds to a lot sizing solution; the total lot sizing cost equals to the weight of arcs on the path. The optimal lot sizing can be obtained from solving the shortest path problem.
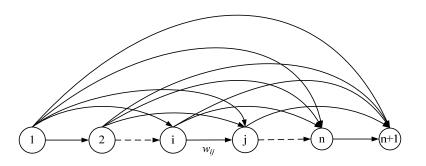
Figure 4-3 A network presentation of lot sizing

Dijkstra Algorithm (Dijkstra 1959) is adopted in this research, as it is an efficient algorithm to solve "one-to-all" network (Rardin 1997). In the network, every arc only starts from the node with smaller number to the node with larger numbers. Therefore, if a node is a permanent node, only the nodes with larger numbers are considered as candidates of next permanent labeled nodes. The pseudo code to find the optimal lot sizing for a customer order is as follows.

Let weight(i,j) be the lot sizing cost;
Let permTable(i,j) temporary path cost, set permTable=Max_value;
Let path(i) be the final shortest path;
permNode=0;
for pass=1 to n+1:
    minCost=Max_value;
    for j=permNode+1 to n+1;
            if pertable(pass-1,j)<minCost:
                minCost= permTable(pass-1,j);
                permNode=j;
            end if;
        end for;
    if permNode=n+1 break;
    for j=permNode+1 to n+1;

43

if minCost+weight(permNode,j)≤permTable(pass-1,j):

           permTable(pass,j)=minCost+weight(permNode,j);

           path(j)=permNode;

    else   permTable(pass,j)=permTable(pass-1,j);

    end if;

end for;

end for   ;

k=n+1;

while k>1:

   t2=k;

   t1=path(k);

   k=t1;

   X(t1)= sum demand from t1 to t2-1;

   k=k-1;

End while;

As this lot sizing method provides the least cost for every order, the maximum profit can be computed. If an order's cost calculated from this method overweighs its initial gross profit, it should be permanently rejected.

(2) Screening based on workload

Consider an extreme case when full capacity is used to serve for one order. The cumulative demand for order i from period 1 to t, is denoted as $D_{it} = \sum_{j=1}^{t} d_{ij}$. The least workload (in time units) on machine k resulted from the cumulative demand is $D_{it}p_{ik} + \tau_{ik}$. As no machine can be overloaded at any period, the order screening condition is obtained, as shown (4.1). If an order violates this condition, it should be permanently rejected.

$$D_{it}\rho_{ik} + \tau_{ik} \leq ct \qquad\qquad \forall i \in P, \forall t \in N, \forall k \in M_i \qquad\qquad (4.1)$$

4.2.2 Heuristics for Selecting Single Order

Without detailed production plan, the cost cannot be exactly estimated. One alternative is to use the lot sizing cost from the WW algorithm. The following indices are used for selecting single order.

(1) Most Profit (MP)

In order to achieve a larger profit for every order, the most profitable order is selected all the time. If an order is finally planed with the WW lot sizing, this order selection rule can ensure maximum profit. Let C_i be the lot sizing cost of order i, then the index I_{MP} (profit) is calculated as:

$$I_{MP} = \max_{i \in S^c}(r_i - C_i) \qquad\qquad (4.2)$$

(2) Least Workload (LWK)

Aggregate workload of an order is an important order selection criterion; examples can be found in the work of Wester (1992) and Ebben (2005). In the problem under study, the aggregate work load is considered with respect to total processing time without lot sizing. To facilitate scheduling, the order with least workload is first considered. The index I_{LWK} (workload) is calculated as:

$$I_{LWK} = \max_{i \in S^c}\left(\rho_i \sum_{t=1}^{n} d_{it}\right) \qquad\qquad (4.3)$$

(3) Weighted Most Profit (WMP)

If the job shop environment is viewed as an aggregate resource, selecting orders with different profits is equivalent to the Knapsack problem. A greedy approximation

algorithm for the knapsack problem is to select the item with weighted profit (Dantzig 1957). Therefore, the index I_{WMP} (ratio) is calculated as:

$$I_{WMP} = \max_{i \in S^c} \left(\frac{r_i - C_i}{\rho_i \sum_{t=1}^{n} d_{it}} \right)$$

(4.4)

(4) Least Resource Competition (LRC)

When conducting order selection, schedule of all former selected orders should be taken into account (Wester et al. 1992). Intuitively, selecting the order whose route differs from that of existing orders may lessen resource competition among orders. The competition level can be measured by the ratio of demand to unused resource. Here, cumulative demand and resource are considered, because they are less sensitive to the lot sizing decision. If the ratio is low, the new order is more likely to seize resources; consequently a feasible schedule can be constructed. This heuristic consists of the following steps.

Step 1. Calculate R_{kt}, the unused resource of machine k at period t.

Step 2. The cumulative resource availability over period t is calculated as

$$C_{kt} = \sum_{j=1}^{t} R_{kj} .$$

Step 3. Determine the cumulative demand on resource k as

$$D_{ikt} = \sum_{j=1}^{t} (X_{ij} \rho_{ik} + \tau_{ik} Y_{ij}) .$$

Step 4. For each order i, calculate the resource competition index as the maximum rate of demand to resource over all machines, i.e.,

$$I_{LRC} = \max\left(\frac{D_{ikt}}{C_{kt}}\right).$$ The order with the least resource competition index is

selected.

4.3 Deselecting Orders

Aforementioned order selection rules are based on estimated resource unitization. Therefore, it is possible that selected orders cannot be fully scheduled. Assume an order i is first selected. If it is deselected due to limited capacity, there will be no effect on the total profit of formerly selected orders. If an order j from the formerly selected orders is deselected, the profit of j should not exceed that of i; otherwise, deselecting i is the best choice to maintain the total profit. If i is more profitable, it should be accepted, but some less profitable orders have to be deselected. Figure 4-4 illustrates this situation. Initially, $S^+ = \{a,b,c\}$ and the schedule is feasible. Suppose including order i results in an infeasible schedule. To achieve feasibility, order a, c are deselected successively, given deselecting them results in less loss of profit than deselecting i. As a result, $S^+ = \{b,i\}$ is the new set of selected orders.

Profit

{a, b, c, i}

{b, c, i}

{b, i}

{a, b, c}

Infeasible

Feasible
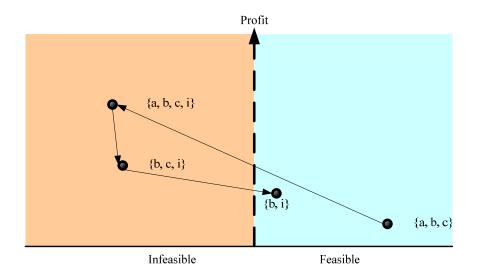
Figure 4-4 Illustration of deselecting multiple orders to achieve feasibility

To identify which order should be deselected, the profit and resource consumption information are taken into account. At this stage, resource utilization and lot sizes are obtained from the existing production plan. The following rules are used to deselect an order from the list of selected orders.

(1) Least Profit (LP): Deselect the order with least profit.

(2) Most Workload (MWK): Deselect the order which imposes the most workload on machines.

(3) Weighted Least Profit (WLP): Deselect the order with least ratio of profit over resource consumption.

(4) Maximum Lateness (LMAX): If an operation results in maximum lateness in the scheduling, corresponding order is deselected.

The first three rules focus on order level; the logic is against the MP, LWK, and WMP, respectively. The last rule focuses at scheduling level. The logic is consistent with

48

Wester et al. (1992), who addressed controlling orders' lateness in the order selection problem.

4.4   Scheduling Production Lots

After lot sizing, each production lot is further planned as a set of operations, according to process specification. In this research, disjunctive graph is employed to represent the job shop scheduling problem and facilitate designing scheduling algorithms.

4.4.1   A Brief Description on Disjunctive Graph Representation

Disjunctive graph is one of the most popular graphic models used for describing instances of the job shop scheduling problem. It is a directed graph $G = (V, C, D)$. Herein, V denotes a set of vertices corresponding to operations, and two additional vertices. The two additional vertices are the source node (S) and terminal node (T). These two nodes represent the start and end of a schedule, respectively. Both nodes have zero processing time. C is a set of conjunctive arcs that reflect the precedence constraints initially connecting every two consecutive operations from the same job. Undirected disjunctive arcs D connect mutually unordered operations, which require the same machine. In a disjunctive graph, each arc is labeled with a positive weight which equals to the processing time of the precedent operation. Generally, release time is labeled on the arc from S to the first operation of the job. Figure 4-5 shows an example of disjunctive graph, which represents the job shop data in Table 4-1. The scheduling task is to determine the sequence of operation 1, 5 on M_1, and operation 2, 4 on M_2.

Table 4-1 Job shop data for disjunctive graph example

Job	Machine sequence	Processing time
J_1	$M_1 -> M_2 -> M_3$	$p_1 = 10, p_2 = 20, p_3 = 25$
J_2	$M_2 -> M_1$	$p_4 = 25, p_5 = 10$
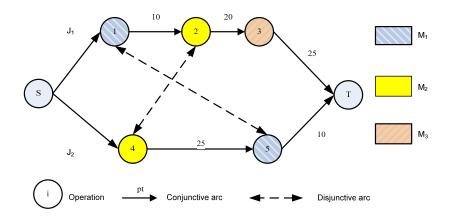
Figure 4-5 An example of the disjunctive graph

4.4.2 Disjunctive Graph for Lot Scheduling

The disjunctive graph for scheduling production lots of the problem under study is constructed with the following steps.

(1) Construct conjunctive arcs

Operations of a production lot are generated according to the predefined process planning. Given an operation O_{itk}, the production time is calculated as $\rho_{ik} X_{it} + \tau_{ik}$. The conjunctive arcs for a production lot correspond to the process planning.

(2) Calculate release time of operations

An operation can only start after the completion of its preceding operation. According to production route, the following formulas on release time of operation O_{itk} are derived:

50

$$r_{itk} = r_{itk'} + X_{it}\rho_{ik'} + \tau_{ik'} \qquad\qquad \forall i \in P, \forall t \in N, \forall k \in M_i \qquad\qquad (4.5)$$

where k' is the immediate precedent machine of k on the route

If an operation is the last operation and another production lot from the same order is planned ahead of it, there are two conditions under consideration. First, given two lots L_{it} and $L_{it'}$ ($t > t'$), $O_{it'm_i}$ (the last operation of $L_{it'}$) should be processed before O_{itm_i} (the last operation of L_{it}). Otherwise, lot $L_{it'}$ cannot be completed on time. Therefore, a disjunctive arc is predefined for $O_{it'm_i} \rightarrow O_{itm_i}$. O_{itm_i} should start no earlier than the completion of $O_{it'm_i}$, which is $r_{it'm_i} + X_{it'}\rho_{im_i} + \tau_{im_i}$. Second, To avoid early completion, the last operation O_{itm_i} cannot start before $c(t-1) - X_{it}\rho_{im_i} - \tau_{im_i}$. Hence, in addition to condition (4.5), the following condition is obtained for the release time of those last operations.

$$r_{itk} = \max(r_{it'm_i} + X_{it'}\rho_{im_i} + \tau_{im_i}, c(t-1) - X_{it}\rho_{im_i} - \tau_{im_i})$$

$$t > t', \ \forall i \in I, \forall k \in M_i \qquad\qquad (4.6)$$

(3) Assign latest finish time to operations

A feasible production plan requires each production lot must be completed in the corresponding planning period. At the operational level, it implies any operation should be completed no later than corresponding latest finish time. The latest finish time can be calculated recursively starting from the last operation of the production lot. For operation O_{itk}, the latest finish time is:

$$d_{itk} = d_{itk'} - X_{it}\rho_{ik'} + \tau_{ik'} \qquad \forall i \in P, \forall t \in N, \forall k \in M_i$$

<div align="right">(4.7)</div>

where k' is the immediate succeeding machine of k on the route.

(4) Add the source node and terminal node

The source node S is connected to all the first operations of production lots; the weights represent corresponding release time of operations. All last operations of production lots are connected with the terminal node T. When those last operations are imposed with release time, they are connected with node S.

Figure 4-6 shows an example of disjunctive graph for scheduling production lots. Two production lots for order i and one production lot for order i' are illustrated in this figure. The routes of both orders are predefined as $k' \rightarrow k$. Conjunctive arcs are added according to this route. Arc $S \rightarrow O_{it'k'}$, $S \rightarrow O_{itk'}$ and $S \rightarrow O_{i't'k'}$ are associated with zero weight, as those first operations can start at the beginning. The weight of arc from S to those last operations (O_{itk}, $O_{it'k}$ and $O_{i't'k}$) are calculated from formula (4.5) and (4.6). At this point, no disjunctive arc is defined, except for $O_{it'k} \rightarrow O_{itk}$, because the are last operations from same order.
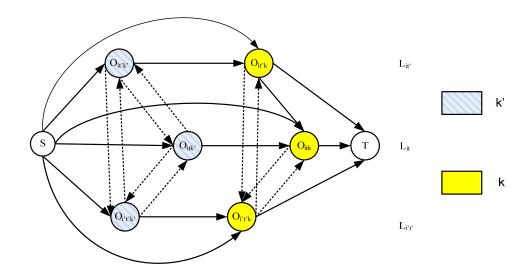
Figure 4-6 Illustration of the disjunctive graph for scheduling production lots

4.4.3 Scheduling Heuristics

The job shop scheduling problem is NP-hard; therefore heuristic methods are most commonly used. In the research under study, two categories of heuristics are under consideration: shifting bottleneck heuristic (SBN) and priority rule-based dispatching (PRD) scheduling.

(1) Shifting Bottleneck (SBN)

In the disjunctive graph of scheduling production lots (see Figure 4-6), if the weight of arc starting from last operation O_{itk} to T is posted as $cn - d_{itk}$, a feasible schedule can be ensured with $C_{max} \leq cn$. Then, a $J_m \| C_{max}$ problem is formed. This problem can be solved with the shifting bottleneck heuristic (Adams et al. 1988). Detailed description of this method can also be found in Pinedo (2002). In this method, the bottleneck machine is identified by finding machine with largest L_{max} in a single machine scheduling problem. Although the branch and bound and preemptive EDD rule

can generate a good solution for $1 \mid r_j, prec \mid L_{\max}$ (Pinedo 2002), it needs a great amount of computational effects, especially when there are more operations defined on every machine. Therefore, earliest due date (i.e. schedule the operation with latest finish time first) rule is employed to solve the maximum lateness problem.

(2)   Priority Rule-based Dispatching (PRD)

Dispatching rules are easy to be implemented and generally requires much shorter computational time. The scheduling priority index is calculated for each operation. Although many PRDs are proposed in literature, there is no single priority rule dominates performance comparisons (Baker 1984). In this research problem, minimizing makespan and number of tardy orders would be advantageous for on-time delivery and total profit maximization. By referring to scheduling rules summarized in Panwalkar (1977) and Haupt (1989), PRDs considered in this research is listed in see Table 4-2. To fit the problem under study, some customizations are made.

Table 4-2 PRDs used for scheduling

Abbreviation	Description
EOFT	Earliest Operation Finish Time. Among all operations that are ready for scheduling, first schedule the operation with the least latest finish time.
MWKR	Most Work Remaining. This scheduling method starts from identifying the order with the most remaining work. The remaining work is calculated as the processing time of all unscheduled lots and unscheduled operations of those partially scheduled lots. Then for the identified orders, select the lot with earliest production period. The prioritized operation is the first unscheduled operation of the selected lot.
CR	Critical Ratio. Critical Ratio is an index computed by dividing the time remaining until the due date by the work time remaining, i.e., $CR_{itk} = \frac{d_{itk} - ES_{itk}}{\theta_{itk}}$. Here, θ_{itk} and ES_{itk} denote the processing time and earliest start time of O_{itk}, respectively. A smaller index indicates the operational time is tighter; therefore, the operation with the least critical ratio is first scheduled.

Abbreviation	Description
SPT	Shortest Processing Time. Schedule the operation with the shortest processing time first. This rule is advantageous for minimizing average flow time.
LPT	Longest Processing Time. Schedule the operation with the longest processing time first. The rule can contribute to minimizing makespan.
PEDD	Profit and EDD. First, select the order with largest profit. Then apply EDD(i.e. least latest finish time) to schedule operations of the selected order.
LSLK	Least Slack. Select the operation with least slack time. This method is to schedule least flexible operation first, so that fewer orders are expected to be rejected due to scheduling infeasibility.
MQ	Most Queues. This method is to first schedule the operation with most subsequent operations. Herein, subsequent operations refer to succeeding operations defined by the route, and all operations in the production lots at later periods. In other words, the operation with greatest effects to other operations is prioritized, so that it is expected to facilitate scheduling more operations within production horizon.
RND	Random Scheduling. The prioritized operation is randomly selected. This method is used as the benchmark for evaluating other scheduling rules.

4.4.4   Feasibility Check

To ensure on-time completion of every operation, the following conditions should be met.

$$r_{itk} + X_{it} p_{ik} + \tau_{ik} \le d_{itk} \qquad \forall t \in T \ \forall i \in I, \forall k \in M_i \qquad (4.8)$$

If this condition is not met, scheduling is terminated. Further actions should be taken, such as deselecting orders or adjusting production lots. The following sections cover these issues.

4.5   Minimum Cost Heuristic (LS1) for Lot Sizing

The minimum cost heuristic follows the concept of improving production schedule from an initial solution (Karimi et al. 2003). The initial lot sizing is generated

with the WW algorithm for every order; therefore, the total profit is maximized. However, this optimal lot sizing may result in an infeasible schedule, especially when customer demands are much heavier than production capacity. As a solution, lot size change methods are used to balance resource consumptions among different planning periods. Since the implementation of the WW algorithm was presented in Section 4.2.1, this section only focuses on lot size change methods.

4.5.1   Identification of the Critical Lot

Two questions are involved in changing lot sizes. The first question is which production lot should be changed; the second question is in what quantity should the lot be changed so that scheduling feasibility is facilitated. The first question can be answered by examining the infeasible schedule, on which at least one operation is delayed. The operation with maximum lateness determines the level of infeasibility; therefore, the production lot corresponding to this operation is the lot to be changed (critical lot). To answer the second question, it is necessary to investigate detailed schedule and evaluate the consequence of lot sizing changing.

(1) Two types of restricts on a schedule

Given a schedule, every operation has two types of constraints on its earliest start time: disjunctive constraint and conjunctive constraint. The conjunctive restricted operation (CRO) is identified as the immediate precedent operation that is scheduled on the same machine. Meanwhile, disjunctive restricted operation (DRO) is the immediate precedent operation according to the order's route. If there is no gap between an operation and its precedent operation, they are regarded as strongly constrained. Otherwise, they are weakly constrained. Figure 4-7 shows two examples. Operation j_1

and j_2 are originated from the same order. In (a), j_1 is the strong CRO for j_2 and operation i is the weak DRO. While in (b), j_1 is the weak CRO for j_2, and operation i is the strong DRO. As forward non-delay scheduling methods are used in this research (see Section 4.4.3), the idle time on a schedule is the starting time of an operation to the finish time of its weakly constrained operation, either CRO or DRO. Changing lot size can affect the processing time of operations defined on different machines; therefore, it is possible to reduce the idle time.

(a)   (b)
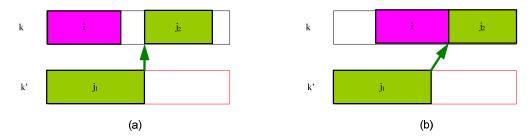
Figure 4-7 Illustration of constraints among operations

(2) Limit of lot size change

To facilitate analysis of lot sizing changing, a topological sorting is applied to label all nodes in the disjunctive graph presented in Section 4.4.2. A precedent node is always labeled with a smaller integer than that of its succeeding nodes.

If the lot size of L_{it} is decreased by one unit, all operations of this lot will be firstly affected, because the processing time of each operation relies on the lot size. Also, if the sequence of operations defined on machines is kept unchanged, changing lot size will cause shift of schedule. Specifically, if the processing time of an operation is changed, all operations with larger label numbers (from the typological sorting) will be affected. Suppose O_{itk} is the source operation (i.e., no operation with smaller label

number is affected by lot size change). The shifting rate of O_{itk} is ρ_{ik}, because one unit lot size decreasing will reduce processing time by ρ_{ik}. Effects of lot size change can be categorized into three situations: (1) if an operation is strongly constrained by its precedent operation, the operation can obtain same shifting rate; (2) if an operation ($O_{itk'}$) is from the same production lot as the source operation, the shifting rate should be increased by the unit processing time of the affected operation itself, i.e., $v = \rho_{ik} + \rho_{ik'}$; (3) an operation cannot obtain shifting rate from its weakly constrained operation, because idle time exists between them.

After shifting of schedule, operations being affected may be on another critical path, which causes continually reducing lot sizing cannot contribute to feasibility. This is illustrated in Figure 4-8. Before lot sizes being changed, the idle time between two operations on a machine is Δ; the shifting rate of two operations are v and v', respectively. Denote the lot size decreasing quantity is x, then the condition for the limit of lot size decreasing is $x \leq \dfrac{\Delta}{v - v'}$. This is the basic limit on lot size change; it should be applied to every pair of two constrained operations.

In addition, on-time completion of the last operation of any production lot may affect the limit of lot size change. Still use schedule in Figure 4-8 as an example and suppose O_{itk} is the last operation of a production lot. If lot size change makes the start time of O_{itk} change from S_0 to S_1, then $S_1 = S_0 - xv$. According to release time condition in equation (4.6), the last operation's starting time $S_1 \geq ct - c - [\rho_{ik}(X_{it} - x) + \tau_{ik}]$. Then, an additional condition for those last operations is obtained:

$$x \le \frac{S_0 - (ct - c - \beta)}{v + p_{ik}}$$ (4.9)

After calculating lot size change limit for all pairs of constrained operations, the smallest limit is denoted as Q^*.
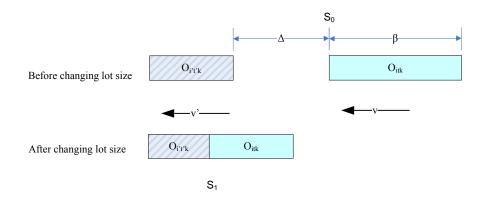
Figure 4-8 Illustration of shifting schedule of an operation when changing lot sizes

4.5.2   *Adjusting Lot Size*

In order to explore all possibilities of achieving a feasible schedule, both backward and forward moving directions are considered. Figure 4-9 demonstrates the two classes of lot size change with moving quantity Δx_i. The source period is t and the target period is t'.

(a) Backward: Move to an earlier period   (b) Forward: Move to a later period
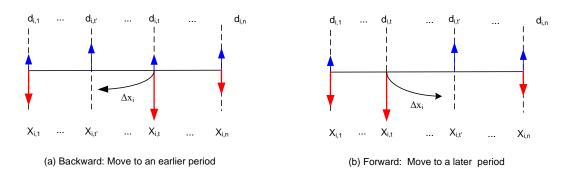
Figure 4-9 Two directions of lot size change

59

Table 4-3 List of lot size change methods

Target period	Moving quantity	Lot moving direction	
		Backward	Forward
Has lot	Partial	(1) Backward lot transferring	(5) Forward lot transferring
	Entire	(2) Backward lot merging	(6) Forward lot merging
No lot	Partial	(3) Backward lot splitting	(7) Forward lot splitting
	Entire	(4) Backward lot shifting	(8) Forward lot shifting

The effect (cost change and resource balance) of lot size change depends on the moving quantity and whether a production lot exists in target period. Table 4-3 lists all situations. If a production lot exists in the target period, moving a partial lot is called lot transferring; while moving the entire lot is equivalent to merging two production lots. If there is no production lot in the target period, moving partial production quantity is equivalent to splitting the source production lot into two lots; while moving the entire production lot to another period is shifting the production lot.
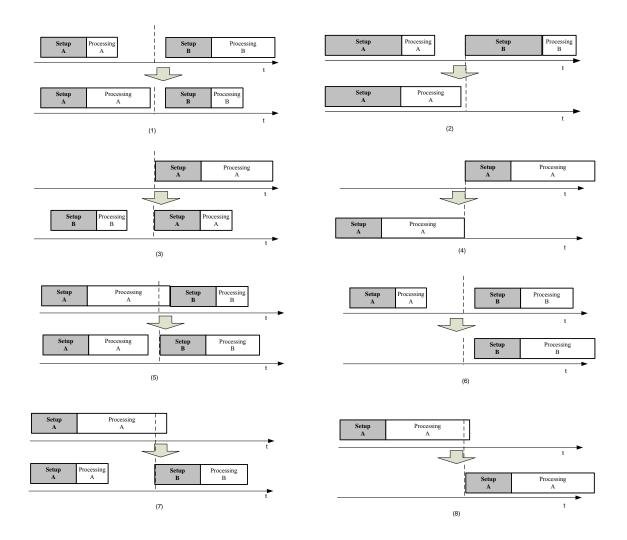
Figure 4-10 Effect on resource consumption by changing lot sizes

The basic lot size change is conducted between two adjacent periods. Lot sizing change over multiple periods can be achieved through multiple iterations of the basic lot size change. The effect on schedule is graphically illustrated in Figure 4-10. For each method, the upper level shows the original schedule and the lower level shows the schedule after lot size change. The cost variation and resource consumption are analyzed as follows.

(1) Backward lot transferring

This method is valid only when no operation from the target lot is on a critical path. It balances current production work load on the current period and the pervious period. In respect of cost change, this method only increases holding cost $h_i \Delta x_i$. For resource consumption, the total resource used (in time unit) keeps unchanged on every machine, because there is no extra setup.

(2) Backward lot merging

Merging production lots reduces a setup, but the holding cost increases because the source lot is produced earlier. Also, a much tighter production is resulted, because the internal due date of the source lot is shorten. Therefore, this method is valid only when setup time is relatively large. Otherwise, it merely contributes to a feasible schedule. The increase of cost is $h_i X_{it} - \delta_i$.

(3) Backward lot splitting

If there is no production in the previous period, move a partial lot to pervious period will create a new production lot. Therefore, the lot sizing cost is increased by $h_i \Delta x_i + \delta_i$. For resource consumption, an extra setup time is needed.

(4) Backward lot shifting

This method is to move current production lot to the previous period, given it cannot be scheduled in the current period. The saved holding cost is $h_i \Delta x_i$, For resource consumption, the total production time needed keeps unchanged.

(5) Forward lot transferring

This method is to move partial current production quantity to a future period. It is valid only if there exists advance production (AP, the portion of current production used

to satisfy demands). The lot sizing cost is reduced by $h_i \Delta x_i$. The total resource consumption keeps unchanged.

(6) Forward lot merging

Merging two production lots not only reduces setup cost, but also save holding cost. In total, the cost decreases by $\delta_i + h_i X_{it}$. However, the precondition is that the entire source lot is advance production.

(7) Forward lot splitting

With this method, a new production lot is created. Therefore, the setup cost is increased; but the holding cost is saved. In total, the cost increasing is $\delta_i - h_i \Delta x_i$. The total resource consumption is increased by a setup time.

(8) Forward lot shifting
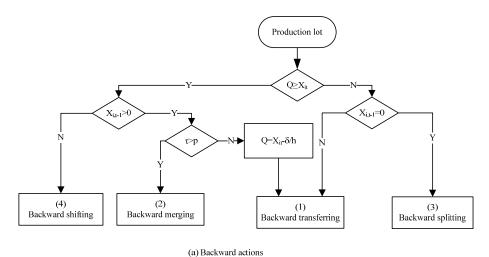
Moving the entire production lot to the future period can reduce holding cost by $h_i X_{it}$. As there is no change on setup, total resource consumption keeps unchanged. Still, this lot changing method is valid only when the entire source lot is advance production.

4.5.3   Selection on Lot Change Method

After the critical lot and limit of lot size change are identified, the moving quantity Q needs to be decided. Intuitively, if the schedule is infeasible with a larger L_{max}, the lot size should be changed with a larger quantity. Otherwise, only a small adjustment is needed. This implies moving quantity should be related to the level of infeasibility, which can be measured by L_{max}. Therefore, the following formula is used to determine the moving quantity.

$$Q = \min(Q^*, \frac{L_{\max}}{\rho_i}) \tag{4.10}$$

This formula integrates lot sizing and scheduling requirement. As a result, any lot size is not required to be exactly the sum of a set of future demands; therefore, it leaves more chances for constructing a feasible schedule.

(a) Backward actions

(b) Forward actions

Figure 4-11 Determination on lot size change method

Figure 4-11 summarizes the selection on lot changing methods for two directions. Selection on moving direction (forward or backward) is based on the cost of lot size

64

change methods. The method resulting in least lot sizing cost increment is selected. For backward merging, it is valid only when the setup time is larger compared to the unit processing time. Otherwise, the moving quantity is set as $X_{it} - \dfrac{\delta_i}{h_i}$, to avoid too much increment on holding cost. For forward lot size change, the moving quantity is set as $Q = \min(Q^*, AP)$, so that all demands are met without delay.

4.5.4   An Iterative Procedure to Relax Infeasibility

To change an infeasible schedule to a feasible one, multiple iterations of lot size change may be needed. The iterative procedure is shown in Figure 4-12. In this procedure, if $L_{\max}$ does not keep decreasing, another critical lot is considered. If changing lot size has been applied to all production lots and still no feasibility can be achieved, a customer order is deselected, base on the rules of deselecting orders introduced in Section 4.3.
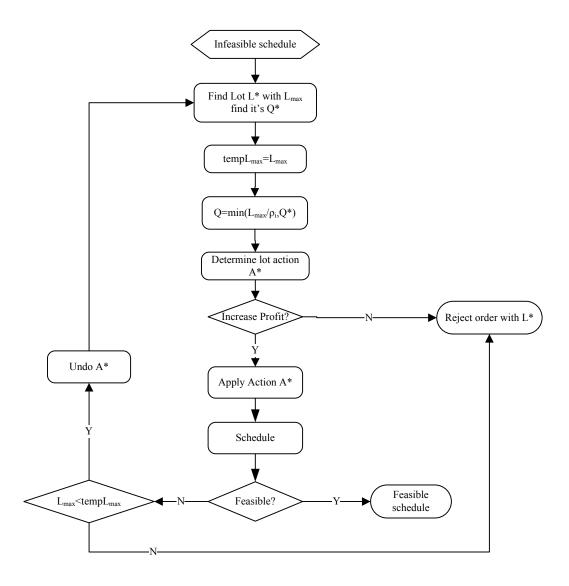
Figure 4-12 Procedure of achieving feasibility by lot size change

4.6   Minimum Cost Increase Heuristic (LS2) for Lot Sizing

Instead of conducting optimal lot sizing for every order at the beginning, lot sizing can be constructed in a period-by-period manner. At each period, feasibility is first ensured and then profitability is desired. One renowned CLSP heuristic applying this concept is the Dixon-Silver method (DS), developed by Dixon and Silver (1981). The proposed LS2 heuristic is an extension to the DS method.

4.6.1   A Brief Description of the DS Method

DS method considers including some future demands into the current period production, given such an inclusion is cost-saving. It is also characterized by a look-ahead mechanism on ensuring feasibility.

(1) Two portions a production lot

The quantity of a production lot can be divided into two portions: advance production (AP) and on-time production. Advance production is the production quantity that accounts for future demands. Including AP saves setup cost, but it increases holding cost. At any period, after advance production is conducted, the net demand turns to be the original demand deducted by advance production. The on-time production should always equal to the net demand; otherwise, production and demands are unbalanced. Given an item i, the integer number of periods of demands that a production lot will exactly satisfy is named as time supply (T_i).

(2) Criterion of including advance production

In the DS method, Including AP is applied to the lot with the largest decrease of average cost per unit time per capacity absorbed. Herein, Average Cost (AC) per unit time is defined in Silver-Meal method (Silver and Meal 1973).

$$AC_i(T_i) = \frac{\delta_i - h_i \sum_{t=1}^{T_i} (t-1)d_{it}}{T_i} \qquad (4.11)$$

Therefore, decrease of average cost per unit per capacity (u_i) is represented as

$$u_i = \frac{AC_i(T_i) - AC_i(T_i + 1)}{k_i d_{T_i+1}}$$ (4.12)

where k_i is the resource utilization rate for item i

Hence, the time supply T_i should be increased by one period for the lot with largest positive u_i.

(3) Consideration of feasibility

Resource capacity in the current period imposes a limit on time supply increasing. In the DS method, resource consumption for different items is addable, maximum advance production that do not validate capacity at current period is derived. In addition, the DS method also provides a look-ahead mechanism: if advance production is not conducted for a certain item at current period, there may not be enough resource to produce it in the future periods. With such a contradiction, advance production should be considered for the sake of future feasibility, even though such a plan increases lot sizing cost.

(4) Period-by-period procedure

On-time production and advance production are conducted from the first period to the last period, given feasibility can be sustained. The basic procedure for the DS is as follows.

Step 1.   Given all input items, initialize the net demands at each period as the original demands.

Step 2.   Set working period $t=1$.

Step 3.   Conduct on-time production for the net demand in period t. If the capacity

constraint is not validated, continue; otherwise, terminate this procedure

with an infeasible solution.

Step 4.   If $t=n$ (the last planning period), stop this procedure with a feasible

solution. Otherwise, conduct advance production and continue.

Step 5.   Move working period to the next period ($t=t+1$), and go to Step 3.

4.6.2 Modification to the DS Method

The setting of the problem under study differs from CLSP in the following

aspects. First, workload from different orders cannot be simply added up, due to job shop

schedule. Second, setup time is explicitly defined, thus there is no constant resource

consumption rate for any order. Third, orders are not mandatory for production;

consequently, feasibility can always be achieved by deselecting orders. To accommodate

characteristics of the problem under study, the following modifications to the DS method

are made.

(1) Condition for conducting advance production

According to equation (4.12), the profit index (u_i) is designed as:

$$u_i = \frac{AC_i(T_i) - AC_i(T_i + 1)}{\tau_i + \rho_i d_{T_i+1}}$$
(4.13)

(2) Feasibility pre-check with a look-ahead mechanism

The on-time delivery requirement enforces that cumulative production should

meet cumulative demand at any period; this condition can be ensured by producing as

much as net demand d_{it}^-. When advance production is conducted, net demand should be

updated accordingly. If the current production does not include any future demands, then the future demands have to be fulfilled by consuming resources in the future periods. However, future resource utilization cannot be exactly calculated, due to possible lot sizing. For simplicity, Lot-for-Lot production is presumed. The expected resource consumption on machine m at each period t calculated as follows.

$$R_{mt} = \sum_{i \in I_m} (\rho_i d_{it}^- + \tau_i) \qquad (4.14)$$

If the expected cumulative resource consumption exceeds cumulative capacity, advance production is needed. Hence, time supply increases until cumulative demand cannot be met by cumulative resource consumption, and the first period found is denoted as t_c. In other word, t_c is the first T_i that violates (4.15). For any order, t_c is calculated for all machines on its route, and the smallest one is selected as a limit on time supply.

$$\sum_{j=t+1}^{T_i} R_{mj} \leq T_i c \qquad \forall m \in M \qquad (4.15)$$

Moreover, schedule of production lots should be taken into account. For simplicity, consider an extreme case that production lots from an order are scheduled in parallel. The cycle time for this order should be no more than the total production time divided by the number of lots (denoted as w). Therefore, when a time supply violates (4.16), another limit on t_c is obtained.

$$\frac{\sum_{j=t+1}^{T_i} (\rho_i d_{ij}^- + \tau_i)}{w} \leq (T_i - t)c \qquad \forall i \in P \qquad (4.16)$$

70

With the two conditions in (4.15) and (4.16) on future production and resource availability, the smallest t_c is chosen as the maximum increment of time supply for an order.

Table 4-4 Decisions on including future demands

Profit index	Future feasibility	
	Feasible	Not feasible
$u_i > 0$	Include only full net demand	Include as much as possible
$u_i \leq 0$	Not include	Include as much as possible

(3) Tradeoff between feasibility and profit

From the cost perspective, include future demands only when $u_i > 0$, given there is enough resource currently. But if future feasibility cannot be ensured, it is necessary to include future demands as much as possible. Table 4-4 lists the determination of including future demands into the current period.

(4) Deal with infeasibility

In the DS method, feasibility check is formulated with a set of closed-form equations. However, in the present research problem, job shop scheduling should be conducted to verify the feasibility. Moreover, a feasible schedule may not be obtained with a single pass of lot sizing, because advance production is based on estimated resource consumption. As shown in Table 4-4, advance production is not conducted when $u_i \leq 0$ and future feasibility is ensured. Considering job shop scheduling, advance production is neither conducted when it worsens infeasibility, which is indicated by $L_{\max}$. More specifically, the following procedure is designed to control a schedule transiting from infeasibility to feasibility.

71

(a) Initially, set the quantity of advance production for item i^* (the item with maximum profit index) as the net demand, i.e., $Q_{AP} = d_{i^*t_c}^-$.

(b) Schedule production lots. A positive L_{max} indicates too much future demands have been included into the current period; therefore, Q_{AP} is reduced by $\dfrac{L_{max}}{\rho_{i^*}}$ and redo scheduling. As a result, it is possible that only a partial demand of a future period is included into the current period production. If a feasible schedule is found, advance production is successfully conducted. Otherwise, keep on reducing Q_{AP} if L_{max} keeps decreasing. If L_{max} does not converge to zero, consider the next item with maximum profit index.

(c) After a feasible schedule is found, update future net demand and increase time supply for i^*.

The flow chart in Figure 4-13 further details the procedure aforementioned at one time period.
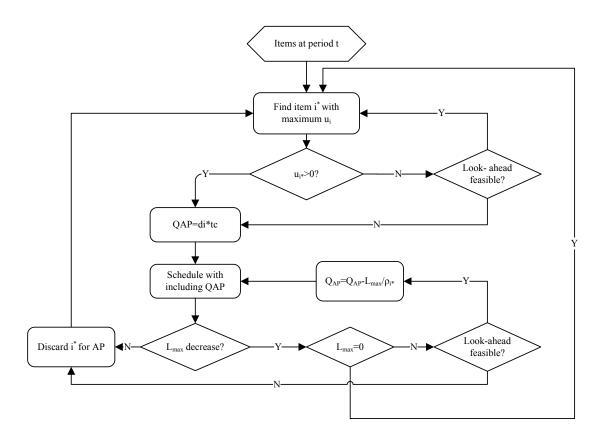
Figure 4-13 Flow chart of LS2 heuristic at one time period

4.6.3   A Summary on Minimum Cost Increase Heuristic

The procedure of minimum cost increase heuristic is summarized in the following steps. The basic framework is the similar to the DS method, but deselecting orders and job shop scheduling requirements are incorporated.

Step 1.   Set working period $t=1$.

Step 2.   Conduct on-time production (lot size equals net demand).

Step 3.   Schedule production lots. If a feasible schedule is obtained, continue; otherwise, deselect order according to an order deselection rule.

Step 4.   If $t=n$, stop. The lot sizes and schedule are the final solution for selected orders.

Step 5.   Estimate expected future resource utilization; find the maximum time supply

t_c for every order.

Step 6.   For all order with its time supply less than t_c (denoted as S^t), calculate u_i.

Find order i^* with $u_{i*} = \max_{i \in S^t}(u_i)$ and consider it for advance production with

the decision matrix presented in Table 4-4.

Step 7.   Schedule the resulted operations. If a feasible schedule is obtained, go to Step

5. Otherwise, work on the next period directly.

Step 8.   If $t<n$, $t=t+1$ and go to Step 2.

5   EXPERIEMENTATION

In Chapter 4, heuristics on order selection, lot sizing and job shop scheduling are presented individually. This chapter focuses on the selecting combined heuristic rules among them, with systematically designed experiments. Moreover, the performance of the proposed heuristic methods is evaluated, by comparing them with the commercial solver CPLEX.

5.1   Experimental Design

The experimentation is to unveil the applicability of heuristics presented in Chapter 4. As there are no benchmark data, problem instances are generated randomly. Then, heuristics are implemented to each random instance.

5.1.1   Random Instances Generation

(1) Basic problem settings

The mathematical model introduced in Chapter 3 reveals that problem complexity not only depends on the number of machines (m), number of periods (n), number of orders (p), but also on the production route and machine flexibility. Hence, a factor called system complexity (sc) is used to create job shop system. It is the ratio of the number of machines on route for an order divided by m. A higher sc indicates more operations are needed for an order; thus, more orders are likely to share common machines. The design for the basic problem settings is shown in Table 5-1. Since these random instances will be further solved by CPLEX, setting of levels is based on the run time analysis presented in Section 3.4. The length of the planning period equals to the capacity of every machine in each period. It can be set arbitrarily, because other related parameters can be adjusted

accordingly; consequently the problem characteristics are not affected. In this experimentation, length of the planning period is set as 100 time units.

Table 5-1 Design for basic problem settings

Factor	Low level	High level
p	Uniform(5, 15)	Uniform(15, 25)
n	Uniform(3, 7)	Uniform(8, 15)
m	Uniform(5, 10)	Uniform(11, 20)
sc	Uniform(0, 0.2)	Uniform(0.3, 0.6)

(2) Parameters

Parameters in the problem under study are categorized into resource related parameters and profit related parameters. Resource related parameters reflect the expected workload of customer orders. They are demands, unit processing time, and setup time. Herein, unit processing time is the base parameter; it is simulated with uniform distribution (0.5, 1.5). Given 10 orders with demand rate of 10 per period, and all orders sharing same machine, all machines will be fully loaded. With such a benchmark, demand at low level is set with a mean of 5; high level is set with a mean of 20. According to Maes and Wassenhove (1988), demand variation over time for a single item (also called lumpiness) affects the lot sizing complexity. Demands of each order can be randomized with setting zero demands in some periods. Two levels on the proportion of zero demands (20% and 80%) are considered in this experimental design. Also, there are two levels of setup time; they are set by referring to the unit processing time.

Profit related parameters include holding cost, setup cost and initial gross profit. The base is the holding cost, which is set as uniform (0.5, 1.5). Setup cost is designed with two levels, with a mean of $10 and $50 respectively. With considering the expected

76

lot sizing cost, the initial gross profit is set with a mean of $500. Such a design is to ensure that customer orders cannot be apparently rejected. To create randomness on discrimination of orders' profit, variation for initial gross profit is designed with two levels. Table 5-2 lists the design of related parameters. With the design in Table 5-1 and Table 5-2, 256 random instances are generated.

Table 5-2 Design of parameters

	Factor	Low level	High level
Resource related parameters	Demand (d_{it})	Uniform(0, 10)	Uniform(10, 30)
	Zero demands	20%	80%
	Setup time (τ_{ik})	Uniform(0, 5)	Uniform(5, 15)
Profit related parameters	Revenue (r_i)	Uniform(400, 600)	Uniform(200, 800)
	Setup cost (δ_i)	Uniform(10, 20)	Uniform(20, 80)

5.1.2 Factor of Heuristics

A Java-based application is developed to implement the proposed heuristic methods introduced in Chapter 4. The following table summarizes all heuristics involved in the experimentation.

Table 5-3 List of heuristic methods

Factors	Levels
Order selection(OS)	(1) Most Profit (MP)
	(2) Least Workload (LWK)
	(3) Weighted Most Profit (WMP)
	(4) Least Resource Competition (LRC)

Factors	Levels
Order Deselection (OD)	(1) Least Profit (LP)
	(2) Most Workload (MWK)
	(3) Weighted Least Profit (WLP)
	(4) Maximum Lateness (LMAX)
Lot sizing procedure (LS)	(1) Minimum cost heuristic (LS1)
	(2) Minimum cost increase heuristic(LS2)
Job shop scheduling (JSS)	(1) Shifting Bottleneck (SBN)
	(2) Earliest Operation Finish Time (EOFT)
	(3) Most Workload Remaining (MWKR)
	(4) Critical Ratio (CR)
	(5) Shortest Processing Time (SPT)
	(6) Largest Processing Time (LPT)
	(7) Profit and EDD (PEDD)
	(8) Least Slack (LSLK)
	(9) Most Queue (MQ)
	(10) Random (RND)

5.2 Result Analysis on Heuristic Methods

5.2.1 Measurements on Solution Performance and System Parameters

The analysis on experiment results is to reveal the relation between solution approaches and solution performance. Therefore, problem setting and applied heuristics are considered as input factors. The target (or response) is the solution performance, which is measured in two aspects: solution quality and run time. The solution quality refers to the total profit of a problem instance solved by heuristics. However, absolute values on profit of different instances are not comparable, because instances may significantly differ from each other on problem setting. Therefore, for each problem instance, the best heuristic solution is first identified. Then, for all heuristics applied to

that instance, the ratio of profit to the best heuristic solution (referred as profit ratio) is used as the indicator of solution quality.

To quantify problem characteristics, the following numerical measurements are used as system parameters for a problem instance.

(1) Basic problem scale measurements: Number of orders (p), Number of planning periods (n), Number of machines (m).

(2) Number of conjunctive constraints (*conjSum*): This factor is calculated as the total number of technical constraints for all orders. Meanwhile, in order to compare different instances, the ratio of conjunctive constraints (*conj_Ratio*) is defined. It is *conjSum* divided by p. These two factors measure the complexity of product process planning.

(3) Number of disjunctive constraints (*disjSum*): This factor is calculated as the total number of disjunctive constraints. Similar to *conj_Ratio*, ratio of disjunctive constraints (*disj_Ratio*) is defined as *disjSum* divided by m. These two factors reflect the complexity of production.

(4) Demand rate (*demand_rate*): It is calculated as the total demand (in time unit) divided by total capacity. Here, total demand is calculated with assuming single production lot for each order; total capacity is calculated as $m \times n \times c$.

(5) Demand-to-capacity ratio: The demand rate does not distinguish process routes of different orders, so it is only a rough estimation. To be more detailed, the demand to capacity ratio on each machine is considered. Given a problem instance, large variation may exist on different machines. Therefore, mean, median, max and

standard deviation for all machines are calculated; they are denoted as *load_mean, load_median, load_max, load_stdev*, respectively.

(6) Variability coefficient of demand (*demand_VC*): As suggested by Silver (1985), demand variability effects lot sizing performance. Herein, *demand_VC* is defined as the variance of demands for every order divided by the square of average demand for all orders. This factor reflects the variability of demand for all orders.

(7) Coefficient of variation for initial gross profit (*revenue_CV*): Coefficient of variation is a normalized measure of dispersion; it is defined as the ratio of the standard deviation to the mean. With this measure on initial gross profit, different instances can be compared. A larger *revenue_CV* indicates the input orders significantly differ from each other in terms of initial gross profit.

(8) Setup time ratio (*st_ratio*): It is calculated as total setup time for all items divided by total unit processing time. It measures the overall significance of setup time for an instance.

(9) Setup cost ratio (*sc_ratio*): Setup cost ratio for every item is defined as setup cost divided by holding cost. For a problem instance, *sc_ratio* is the average of setup cost ratios for all items. This factor measures the significance of the setup cost in contrast with the holding cost.

(10) Model related factors: In Table 3-1, the numbers of different types of decision variables are derived. They are used to quantify problem instance because they reflect problem complexity. Number of continuous variables, integer variables and binary variables are represented with *conVar*, *intVar* and *binVar*, respectively.

5.2.2 Solution Quality of Different Heuristics
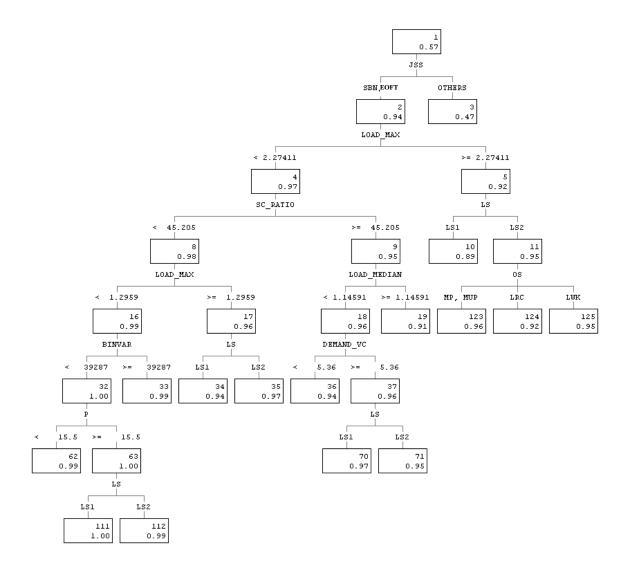
Figure 5-1 Tree diagram on solution quality with including problem setting factors

SAS Enterprise Miner is employed to explore the relations among heuristic rules and solution quality. It can create predictive and descriptive models based on analysis of vast amounts of data, which may involve both continuous and categorical objects (Matignon 2007). To discover the rules of applying heuristic methods, both system parameters and heuristic methods are considered as the input of the tree model. The profit ratio is the target variable. The resultant tree diagram is shown in Figure 5-1. In the

diagram, each branch is associated with the partition of a splitting variable. On each node, the upper number is the node identifier; the lower number is the mean response of that partition. If a node only relates to system parameters, all nodes are pruned. In terms of solution quality, the following basic conclusion can be made.

(1) Job shop scheduling rules significantly affect solution quality. The SBN and EOFT rules perform much better than others.

(2) Lot sizing rules affects solution quality, depending on setting of system parameters. The selection rules are mainly based on the expected maximum demand-to-capacity ratio and setup cost ratio. The LS1 heuristic is suitable for not complex and not heavily loaded job shop. In such a situation, most orders can sustain the profit obtained from the best lot sizing because of less resource conflicts. Also, the LS1 is preferred when setup cost is less significant. This is resulted from to the fact that more small-sized production lots are generated so that scheduling feasibility is more likely to be ensured. The LS2 can better handle complex item's route under higher demands (maximum demand-to-capacity ration on bottleneck machine is higher than 2.27), since more feasibility checks are applied. The setup cost ratio appears to be influential to selection of lot sizing procedure; this is consistent with Gelders (1981), who concluded conducting lot sizing based on cost structure at hand.

(3) For order selection, either WMP or MP can contribute to best solution quality.

(4) The order deselection methods appear to be not significant. This is mainly due to the procedure of taking deselected order back. This procedure tends to produce the same list of deselected orders, no matter what rule is used for deselecting an order.

5.2.3   Evaluation of Scheduling Rules

Since scheduling rules turn to be most significant factors that affect solution quality, Figure 5-2 further depicts the solution quality in terms of profit ratio under different scheduling methods. Compared to the random scheduling, the supervised scheduling methods are shown to be more effective. More importantly, this chart indicates that emphasis on the due date can achieve better a solution. For example, EOFT and SBN consider due date as the most important factor; they achieve much better solution quality than others. CR and LSLK partially incorporate due date; the solution quality is worse than EOFT and SBN. The other methods almost ignoring the due date result in even worse solution quality. The importance of the due date mainly lies in the requirement on non-delay production. Ignoring due date may lead to rejecting profitable orders and thus missing the corresponding revenue.

In terms of run time, different instances are not comparable, because of different problem settings. Therefore, for each instance, run time used by a scheduling rule is compared to the time used by the random scheduling. The time ratio of different scheduling rules is shown in Figure 5-3. The scheduling rules other than EOFT and SBN use less time, mainly because some orders are rejected in an early stage and then solution time is saved. The solution quality of EOFT and SBN significantly outperform the others; meanwhile, computational costs used by them are short and acceptable for implementation purposes (95 percentile of run time for EOFT and SBN are 14.4 and 16.8 seconds, respectively).

	RAND	PEDD	SPT	LPT	MWKR	MQ	CR	LSLK	LOFT	SBN
Profit Ratio	37%	40%	44%	44%	50%	53%	54%	60%	94%	95%
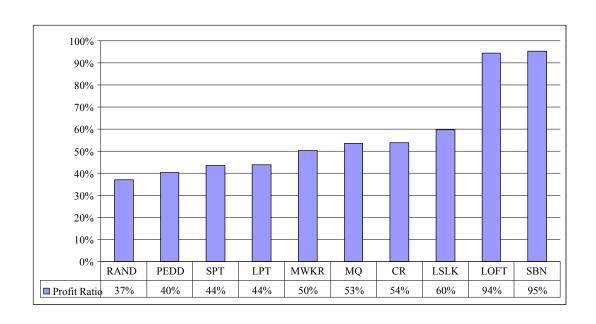
Figure 5-2 Solution quality of different scheduling rules

	PEDD	SPT	LPT	MWKR	MQ	CR	LSLK	EOFT	SBN
Time Ratio	1.30	2.00	2.01	3.10	5.49	2.80	8.51	43.29	45.97
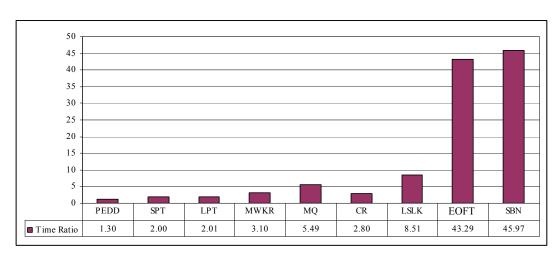
Figure 5-3 Run time comparison for different scheduling rules

5.2.4   Evaluation of Lot Sizing Heuristics
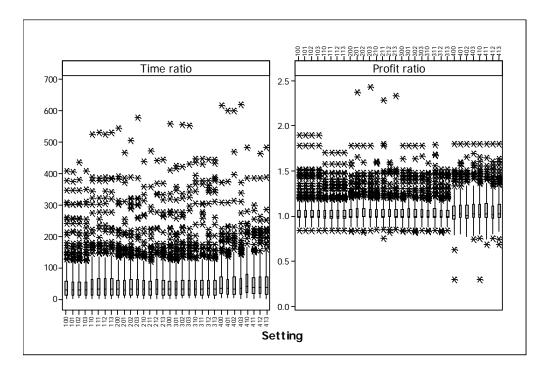
Figure 5-4 Box-plot for profit and time ratio (LS2 to LS1)

Figure 5-4 further plots the detailed comparison on the performance of the two lot sizing procedures. Herein, solutions are classified by the combination of order selection, scheduling and deselection rules (abbreviated numbers are shown in Table 5-3). Figure 5-4 indicates that the LS2 method achieves better profit, but it requires much more run time. Table 5-4 summarizes the numeric statistics between the two heuristics. On the average, applying the LS2 achieves 4.6% better profit. The median of profit ratio is around 1; it indicates the LS2 can produce extremely better solutions for a few instances.

Table 5-4 Summary statistics on solution quality (LS2 to LS1)

	Ratio of Profit	Ratio of Run time
Mean	104.6%	53
Median	100.3%	32

(a) LS1   (b) LS2
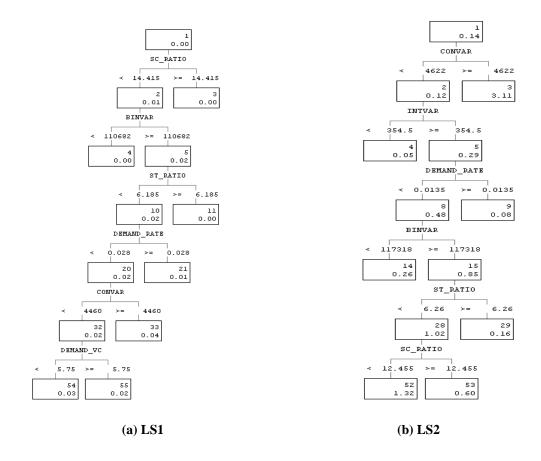
Figure 5-5 Tree diagram of run time (in minutes)

Figure 5-5 further explores the factors that affect run time for the LS1 and LS2. For the LS1, setup cost ratio is the most influential factor. This is because an order with a lower setup ratio is generally planned with more production lots, and thus more computational efforts are needed on scheduling. A lower setup time ratio also results in more run time; this is mainly due to the lot size change procedure, which tends to split an order when the setup time is not significant. If demand rate is high, some orders may be rejected in an earlier stage; therefore, run time is saved, since lot sizing and scheduling are no more needed. In addition, *binVar* also affects run time. This is mainly because more scheduling efforts will be needed when there are more operations on each machine.

For the LS2, *conVar* is the most important factor that affects run time. It is proportional to the number of planning periods and process complexity of orders. *intVar* appear to be significant because the LS2 method runs in a period-by-period manner. Lower setup cost or time ratio results in more production lots; so that run time increases. Still, lower demand causes more orders be accepted, so that more run time is needed.

Since all the instances from the experimental design are solved within a relatively short time (95% instances are solved in 16 seconds), extended experiments with larger problem sizes are designed to test the lot sizing heuristics. Herein, the job shop is fixed with $m = 30$ and $n = 10$. All orders are set with low setup ratio, which requires more run time as indicated by Figure 5-5. The other parameters are randomized. For each p, 10 replicates are tested. Figure 5-6 depicts the run time for both heuristic procedures. For the LS1, the mean and median of run time increase almost linearly; problems with 100 orders can still be solved within 1 minute. For the LS2, there is a large variation on different instances, even for the same p. This is because other parameters appear to be quite influential on number of operations formed, which greatly affects run time. Nevertheless, problems with 100 orders are solved within 6 minutes. Therefore, both methods can be used to solve real-life problem instances.

LS1

LS2
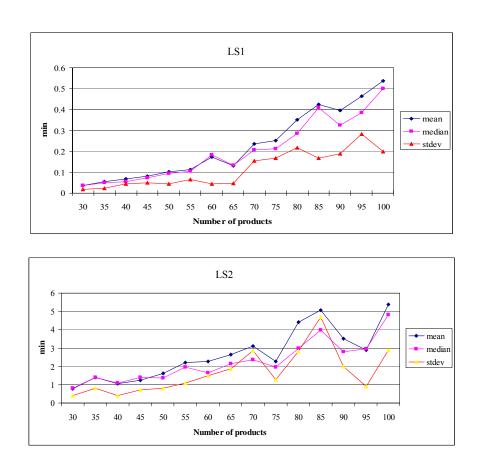
Figure 5-6 Run time vs. Number of products for large-scale problem instances

5.2.5   Comparison on Order Selection Rules

Figure 5-7 further compares different order selection rules, in terms of solution quality and run time. It indicates that the WMP and MP rules achieve the best solution quality. The run times for both rules are acceptable. WMP is slightly better than MP, because it needs relatively shorter run time. LWK and LRC are not effective, which indicates detailed workload estimation cannot contribute to maximizing total profit. To better conduct order selection, the initial gross profit should be the first concern.
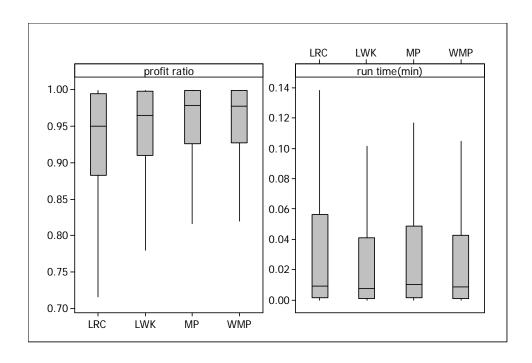
Figure 5-7 Comparison on order selection rules

5.2.6   The Proposed Heuristic Approach

The best heuristic rules discovered in previous sections are incorporated into the heuristic framework introduced in Section 4.1. The proposed procedure is show in Figure 5-8.
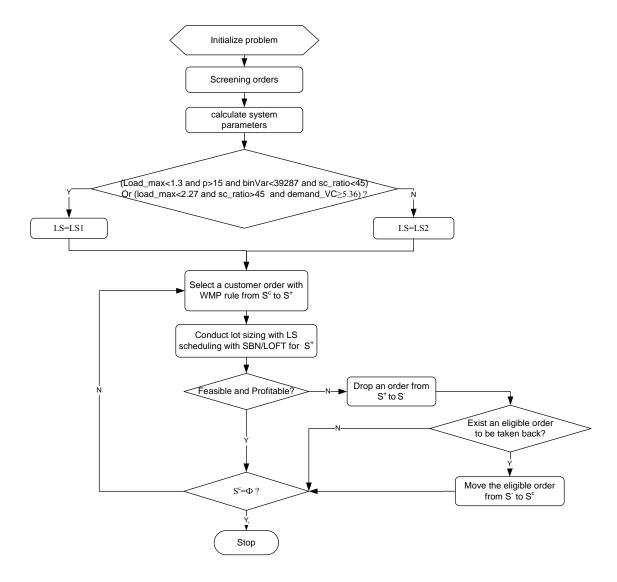
Figure 5-8 The proposed heuristic procedure

5.3   Comparison between Heuristic Solutions and CPLEX Results

The comparison is to validate the performance of the proposed heuristic method, in terms of solution quality and run time.

5.3.1   CPLEX Results

Each randomly generated problem instance is coded with AMPL and solved with CPLEX. Getting the optimal solution for a large-scale problem instance requires a great

amount of computational time, due to the NP-hardness of the problem under study. Therefore, the run time for CPLEX is limited to half an hour. For each instance, CPLEX reports a solution and an *absmipgap* (absolute mixed-integer optimality gap tolerance) value. The *absmipgap* is the difference between the best integer solution and optimal value of LP relaxation. If *absmipgap*=0, the feasible solution found is an optimal solution. As the problem is a maximization problem, the optimal value of LP relaxation can be used as an upper bound.
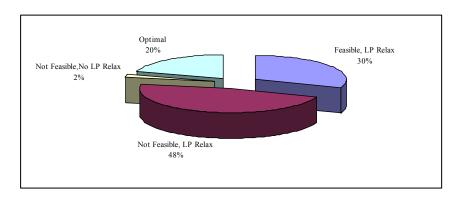
Figure 5-9 Categories of CPLEX solutions

The CPLEX solutions are classified into four categories: (1) optimal solution found within the time limit; (2) feasible but not optimal solution is reported; (3) fail to report feasible solution, but report *absmipgap*; and (4) neither feasible solution nor *absmipgap* is reported due to memory limit. The pie chart in Figure 5-9 shows the proportions among the four categories. It indicates that half of the problem instances cannot be solved to any feasible solution. For around 1/3 instances, feasible but not optimal solutions are found. For the rest instances (20%), optimal solutions are reported. Further analysis on factors affecting each solution category reveals CPLEX cannot obtain an exact solution within the time limit mainly due to the large the number of binary

variables and constant variables (see Figure 5-10). This result is consistent with the analysis in Section 3.1.

For problem instances with LP relaxation, *relmipgap* (relative mixed-integer optimality gap tolerance) is a dimensionless measure for the gap between the feasible solution and the upper bound. It is the ratio of *absmipgap* to the feasible solution. The histogram in Figure 5-11 indicates that most feasible solutions quite approximate to the optimal value of the LP relaxation. Particularly, the *relmipgap* values for 60% LP relaxation solutions are within 10%. The larger gaps are mainly resulted from instances with a large number of binary variables.
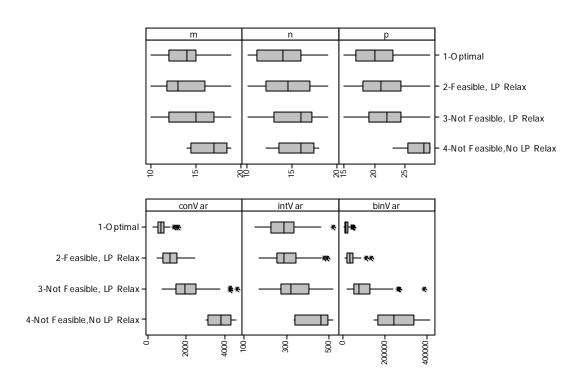
Figure 5-10 Factors affecting run time of CPLEX

92
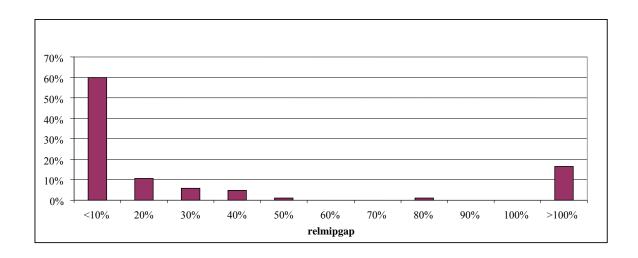
Figure 5-11 Relative frequency of relmipgap in LP relaxation solutions

5.3.2 *Comparison on Solution Quality*

The comparison on solution quality is conducted for different CPLEX solution categories. For an instance optimally solved by CPLEX, the CPLEX solution can be directly used as a benchmark. For an instance that are not optimally solved by CPLEX but a feasible solution is obtained, the heuristic solution is compared to the feasible solution and the upper bound, respectively. If CPLEX fails to report any feasible solution, the heuristic solution is compared to the upper bound only.

Table 5-5 Statistics on the ratio of heuristic solution to CPLEX

Measurement	Optimal	Feasible, not optimal		Not Feasible
		Feasible solution	Bound	Bound
mean	98.3%	2119.4%	90.5%	84.9%
median	100.0%	99.4%	93.0%	84.9%

Table 5-5 lists the comparison result for different CPLEX solution categories. This table indicates that the heuristic method achieves almost same solution quality as the CPLEX optimal solutions. For those instances with feasible but not optimal CPLEX

solutions, on the average, heuristic method significantly outperforms CPLEX. This is because CPLEX can only find feasible production schedule for very few orders within the time limit. Since the ratios are asymmetrically distributed, the median is considered as an indicator as well. A sign test shown in Table 5-6 indicates there is no statistically difference between heuristic method and CPLEX. When CPLEX fails to report any feasible solution, the proposed heuristic method achieves 84.9% solution quality to the upper bound.

Table 5-6 Sign test on the ratio of heuristic solution to CPLEX feasible solution

----

Sign Test for Median: Ratio to CPLEX Feasible

```
Sign test of median =  1.000 versus not = 1.000
                          N   Below  Equal  Above      P   Median
Ratio to CPLEX Feasible  78     45      0     33   0.2129  0.9993
```

5.3.3   Comparison on Run Time

To compare run time of heuristic method with CPLEX, only the instances with optimal CPLEX solution and the LP relaxation solution are considered. The ratio of CPLEX run time to that of heuristic method is used as the indicator. Table 5-7 lists both mean and median on the indicator. It shows that the proposed heuristic method is much faster than CPLEX. When the scale of a problem instance becomes large, CPLEX is unlikely to report an optimal solution; in this situation, the efficiency of heuristic method becomes more significant.

Table 5-7 Ratio of run time (CPLEX to Heuristic)

Measurement	Optimal	LP Relaxation solution
mean	1037	2544
median	13	1316

To sum up, the performance of proposed heuristic procedure is verified. The proposed heuristic method achieves almost same quality as the commercial solver CPLEX for simple-size problem instances. For moderate-size problem instances, the proposed heuristic method reports a feasible solution that is no worse than CPLEX just in a few minutes. For large-scale problem instances, the heuristic approach achieves around 85% solution quality to the upper bound; while CPLEX cannot report any feasible solution. Although the run time of the heuristic method depends on problem setting, the extended experiments indicate that the heuristic methods can solve problem instances up to 100 orders within 6 minutes. Therefore, the proposed heuristic method is effective and efficient for solving practical problem instances of industrial size.

6   A CASE STUDY

To further illustrate modeling and solving order selection and lot sizing problem in the MTO environment, a case study is introduced in this chapter. A comparative study verifies the applicability of the proposed heuristic solution approach.

6.1   Description of a Industrial Case

This research has been applied to a Miami based company. The company makes a large variety of gears, such as bevel gears, sprockets, anti-backlash gears, etc. The manufacturing process is a typical MTO operation and job shop is used for production. Each customer order only contains one single product and no bill of material (BOM) is involved. Customer orders are first received by sales department; engineering department plans manufacturing process according to customer requirements. Review of incoming orders is conducted at the beginning of every week. Now the company faces heavy customer demands; on-time delivery is a primary requirement. However, the production capacity is limited, so order selection decision needs to be made for orders arrived during recent week. Some customers place orders with multiple due dates (mainly for heavy buyers), so the company needs to decide on how to combine demands at different periods. If the requested quantity of a customer order is too large, the company also needs to split the order, so as to expedite order fulfillment. If an order is accepted, it is released to the production department as a work order. Each work order involves several operations according to the process plan. As an example, Figure 6-1 illustrates the process plan for an order on the spur gears. Some work orders involves complicated products may need more than 20 operations; therefore, scheduling operations on the job shop is a difficult task for production planners. Since order selection, lot sizing and

scheduling decisions are involved in this industrial case, it therefore best illustrates the problem discussed in the present study.

Figure 6-1 An exemplary process of work order

6.2   Implementation

The proposed heuristic methods are applied as a Java application. At the beginning of each order review period, a problem instance is formed and solved with the application. If an order is partially completed, only the remaining routes are considered. To avoid an in-process order being rejected, the order is imposed with additional extremely large initial gross profit. After solving the problem, the extra initial gross profit is deducted so that the total profit will not be affected. Since the WMP heuristic is used for order selection, such a conversion can ensure an in-process order always be selected and scheduled first. This case study uses three months operational data. Therefore, 12 problem instances are under consideration. In the operational data, some operations are associated with fixed production time. To facilitate problem modeling, they are converted as operations with setup time only. The basic system parameters for the problem formed are listed in Table 6-1. According to the proposed heuristic procedure (see Section 5.2.6), the minimum cost heuristic (LS1) is used for lot sizing. EOFT is employed as the scheduling rule and the least profit (LP) rule is taken as the order deselection rule.

Table 6-1 System parameters for each instance

week #	p	sc_Ratio	demand_VC	binVar	load_median	load_max
1	17	9.27	374.68	1,259,806	0.03	0.39
2	14	8.64	327.36	1,413,158	0.02	0.46
3	13	6.98	137.54	2,694,432	0.04	0.86
4	29	6.67	43.14	3,477,486	0.04	1.19
5	41	8.57	50.79	4,612,662	0.04	1.29
6	37	10.56	118.9	6,087,406	0.04	1.44
7	38	11.25	70.74	8,986,100	0.04	1.62
8	27	13.06	9385.54	1,438,020	0.01	0.11
9	20	7.39	2864.73	437,398	0.03	0.23
10	15	16.25	612.36	2,707,426	0.03	0.57
11	19	18.9	1695.11	1,736,974	0.03	0.49
12	40	14.8	11458.94	437,998	0.03	0.23

6.3 Result Analysis

The applicability of the proposed heuristics is verified by comparing them with legacy planning tool. With the current tool, orders are firstly classified into priority list and non-priority list. Orders in the priority list are first planned. Orders in the same list are scheduled with the critical ratio rule. The orders that are already in process are set as prioritized orders, since they should not be rejected. The newly incoming orders are put into the non-priority list. They need to be reviewed for scheduling feasibility. An order is rejected if it cannot be delivered on time. In this scheduling tool, lot sizing is done with the Lot-for-Lot policy.

The comparative results are shown in Figure 6-2. Here, P_c represents the total profit of selected orders from current scheduling tool; P_h denotes the total profit obtained from the application which has been applied with the proposed heuristics. It indicates the

proposed solution approach performs better for most planning periods. On the average, it achieves 16.62% better profit than current planning method. This is mainly resulted from selecting more incoming orders. Also, all problem instances can be solved within 15 seconds.
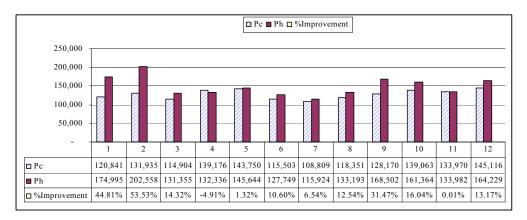

	1	2	3	4	5	6	7	8	9	10	11	12
▫ Pc	120,841	131,935	114,904	139,176	143,750	115,503	108,809	118,351	128,170	139,063	133,970	145,116
▪ Ph	174,995	202,558	131,355	132,336	145,644	127,749	115,924	133,193	168,502	161,364	133,982	164,229
▫ %Improvement	44.81%	53.53%	14.32%	-4.91%	1.32%	10.60%	6.54%	12.54%	31.47%	16.04%	0.01%	13.17%

Figure 6-2 Result of implementing proposed heuristic approach

From this case study, the applicability of the method is verified, as it can be implemented in a real-life case. With the proposed approaches, the company creates a feasible production plan that achieves a better profit and ensures on-time delivery at the same time. Since the proposed heuristic method can runs in a short time, the company can also use it for what-if analysis, such as price negotiation, due date determination, etc. The what-if analysis may require running the heuristics for multiple iterations, but the run time would not be an obstacle.

7 CONCLUSIONS AND FUTURE RESEARCH

This chapter summarizes this research in Section 7.1. Future research is presented in Section 7.2.

7.1 Summary

This research addresses the problem of coordinating order selection, lot sizing, and job shop scheduling problems with its objective of maximal profit. This problem constantly faces managers in an MTO operation. These three decision problems are usually treated separately in the literature and are mostly led to heuristic solutions. Therefore, this research aims to answer the following questions: (1) how to select customer orders to potentially maximize total profit, (2) how to plan orders to make them profitable and schedulable, and (3) How to schedule production lots to ensure on-time completion.

To answer these questions, the research is broken down to two phases. The first phase focuses on analytical definition on the research problem and mathematical modeling. The second phase is to design the heuristic solution approach that can solve problem in practical size.

In the first phase, the research problem is defined as a job shop problem in concurrence with lot sizing and order selection consideration. A subset of incoming customer orders need to be selected in order to maximize the total profit, while meeting the deadline of each selected order. In this problem, only single product is included in each order. Each order may have multiple delivery dates and each due date must be met. To satisfy the time-varying customer demands, lot sizing should be conducted for saving production cost and balancing resource utilization at different planning periods. The lot

sizing cost is composed of setup and holding costs; it offsets the revenue obtained from selected orders. Lot sizing also affects workload in the job shop in each planning period, because lot size determines processing time of corresponding operations. In this problem, profit of an order is defined as the initial gross profit minus lot sizing cost; initial gross profit refers to the price of an order deducted by some fixed production cost. With considering the nature of the concurrent problem, a MILP model is developed by integrates three types of decision variables. In this model, initial gross profit of orders, process plan, manufacturing time and costs are considered. Some what-if scenarios are discussed to illustrate the usefulness of this model, but only for small size problems. Experiments on large-scale problems show that directly solving the mathematical model takes a prohibitively long computation time. Therefore, the second phase is focused on developing an efficient heuristic solution approach.

In this dissertation, a set of rules on order selection, deselection and job shop scheduling are proposed. Also, two lot sizing heuristics are presented. The minimum cost heuristic (LS1) lot sizing method employs the WW algorithm to generate initial lot sizes. Then lot sizes are adjusted to achieve scheduling feasibility. The minimum cost increase heuristic (LS2) employs the DS method as the framework for lot sizing. It works in a chronological manner. At each working period, the decision on conducting advance production is mainly based on a profit index that measures decreasing of average cost per period per capacity. Also, the feasibility check is conducted for both the working period and future periods.

With experiments on simulated problem instances, the rules of selecting heuristics are obtained. Basically, selection of lot sizing heuristic depends on system

parameters, which quantify different aspects of a problem instances. The proposed LS2

heuristic performs the best, when the demand-to-capacity ratio at the bottleneck machine

(i.e., maximum work load ratio) is high. This method can better deal with heavily loaded

job shop, mainly because more feasibility checks are conducted. The proposed LS1

heuristic is suitable for less loaded job shop, because the minimum lot sizing cost are

more likely to be maintained. Order selection with WMP and scheduling with due date

related rules (e.g. SBN and EOFT) are found to be the best choice. Compared to the

commercial solver CPLEX, the heuristic approach is more effective and efficient. Figure

7-1 shows the solution quality under different CPLEX solution categories. Furthermore,

the modeling and heuristic approaches are verified with an industrial case. It shows that

the proposed solution approach achieves 16% better profit compared to the legacy
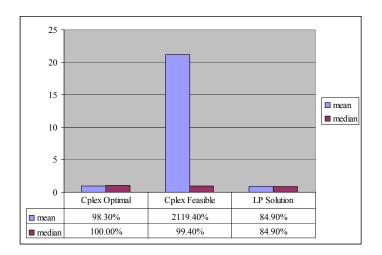
planning tool.

	Cplex Optimal	Cplex Feasible	LP Solution
mean	98.30%	2119.40%	84.90%
median	100.00%	99.40%	84.90%

Figure 7-1 Ratio of heuristic solution to CPLEX solution

The contribution of this research is twofold. First, a mathematical model that

formally defines the order selection and lot sizing problem under the MTO environment

is presented. It is the first model that integrates the three decision problems. In the model

102

formulation, the disjunctive constraints are converted to linear constraints by reformation of a set of logic constraints. Also, a unique constrain that ensures a production lot to be completed in a designated time interval is designed. These constraint servers for coordinating lot sizing and job shop scheduling. Secondly, the proposed solution approach is able to solve large-scale problem within a practical time. The solution performance is shown to be effective and efficient by comparing with the commercial solver CPLEX. The proposed heuristic procedure can be adopted for supporting MTO operations.

7.2   Future Research

This research can be extended by improving current heuristic solution approach. It was found out in developing the heuristic solution that scheduling was most time consuming in search for solutions. One possibility to further improve heuristic's efficiency is to consider rescheduling of partial operations (as defined in the routings). One possible drawback with partial rescheduling is being trapped in a local search loop. Adding random selection to re-scheduling could help break a local trap.

In terms of improving solution quality, meta-heuristic approaches such as GA, TS, and SA could be applied to the order selection process. The solution obtained from the proposed heuristic would be a good initial solution candidate for a meta-heuristics procedure. The neighborhood search could further improve the solution by engaging in pair-wise exchange of order acceptance.

This research can also be extended by changing the problem setting to fit other industrial applications. For example, some customers could accept late delivery with penalties. In such case, the shifting bottleneck concept could be imbedded in the

proposed heuristic framework for job shop scheduling, when the objective is related to weighted lateness. Similarly, the DS method could be extended for lot sizing decision, by incorporating lateness penalty in its look-ahead feasibility routine.

Another possible extension is to consider orders with product assembly and BOM. It would extend the application of this research into companies which engage in build-to-order operations that involve fabrication of components and assembly of finished goods. The extension may start with first solving the problem at a parent product level and then handle components in a hierarchical fashion, similar to a MRP process. Common components typically are more flexible for lot sizing and scheduling, for they are shared by multiple orders. Accordingly one might consider scheduling common components last to maximize the use of available resources.

LIST OF REFERENCES

Absi, N. and S. Kedad-Sidhoum (2009). "The multi-item capacitated lot-sizing problem with safety stocks and demand shortage costs." Computers & Operations Research **36**(11): 2926-2936.

Adams, J., E. Balas, et al. (1988). The shifting bottleneck procedure for job shop scheduling, INFORMS. **34:** 391-401.

Akkan, C. (1997). "Finite-capacity scheduling-based planning for revenue-based capacity management." European Journal of Operational Research **100**(1): 170-179.

Anwar, M. F. and R. Nagi (1997). "Integrated lot-sizing and scheduling for just-in-time production of complex assemblies with finite set-ups." International Journal of Production Research **35**(5): 1447-1470.

Baker, K. R. (1984). "Sequencing rules and due-date assignments in a job shop." Management Science **30**(9): 1093-1104.

Baker, K. R. and G. D. Scudder (1990). "Sequencing with Earliness and Tardiness Penalties: A Review." Operations Research **38**(1): 22-36.

Baptiste, P. (2003). "On minimizing the weighted number of late jobs in unit execution time open-shops." European Journal of Operational Research **149**(2): 344-354.

Beck, J. C. and P. Refalo (2003). "A hybrid approach to scheduling with earliness and tardiness costs." Annals of Operations Research **118**(1).

Bilginturk, Z., C. Oguz, et al. (2007). Order acceptance and scheduling decisions in make-to-order systems. MISTA

Bitran, G. R. and H. H. Yanasse (1982). "Computational complexity of the capacitated lot size problem." Management Science **28**(10): 1174-1186.

Blackburn, J. D. and R. A. Millen (1980). "Heuristic lot-sizing performance in a rolling-schedule environment." Decision Sciences **11**(4): 691-701.

Blazewicz, J., W. Domschke, et al. (1996). "The job shop scheduling problem: Conventional and new solution techniques." European Journal of Operational Research **93**(1): 1-33.

Brucker, P., B. Jurisch, et al. (1993). "Open shop problems with unit time operations." Mathematical Methods of Operations Research **37**(1): 59-73.

Charnsirisakskul, K., P. M. Griffin, et al. (2004). "Order selection and scheduling with leadtime flexibility." IIE Transactions **36**(7): 697-707.

Chen, C.-S., S. Mestry, et al. (2009). "The capacity planning problem in make-to-order enterprises." <u>Mathematical and Computer Modelling</u> **50**(9-10): 1461-1473.

Chen, C. (2006). "Concurrent engineering-to-order operation in the manufacturing engineering contracting industries." <u>International Journal of Industrial and Systems Engineering</u> **1**(1): 37-58.

Chen, W. H. and J. H. Thizy (1990). "Analysis of relaxations for the multi-item capacitated lot-sizing problem." <u>Annals of Operations Research</u> **26**(1-4): 29-72.

Dantzig, G. B. (1957). "Discrete-Variable Extremum Problems." <u>Operations Research</u> **5**(2): 266-277.

Dauzere-Peres, S. and J.-B. Lasserre (1994). "Integration of lotsizing and scheduling decisions in a job-shop." <u>European Journal of Operational Research</u> **75**(2): 413-426.

Dauzere-Peres, S. and J. B. Lasserre (1993). "A modified shifting bottleneck procedure for job-shop scheduling." <u>International Journal of Production Research</u> **31**(4): 923 - 932.

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs." <u>Numerische Mathematik</u> **1**(1): 269-271.

Dixon, P. S. and E. A. Silver (1981). "A heuristic solution procedure for the multi-item, single-level, limited capacity, lot-sizing problem." <u>Journal of Operations Management</u> **2**(1): 23-39.

Dogramaci, A., J. C. Panayiotopoulos, et al. (1981). "The dynamic lot-sizing problem for multiple items under limited capacity." <u>IIE Transactions</u> **13**(4): 294 - 303.

Drexl, A. and A. Kimms (1997). "Lot sizing and scheduling -- Survey and extensions." <u>European Journal of Operational Research</u> **99**(2): 221-235.

Ebben, M. J. R., E. W. Hans, et al. (2005). "Workload based order acceptance in job shop environments." <u>OR Spectrum</u> **27**(1): 107-122.

Eric, R. and K. T. Aaron (1986). "A review of lot-sizing techniques for deterministic time-varying demand." <u>Production and Inventory Management</u> **27**(3): 65-79.

Galambos, G. and G. J. Woeginger (1995). "Minimizing the weighted number of late jobs in UET open shops." <u>Mathematical Methods of Operations Research</u> **41**(1): 109-114.

Gelders, L. F. and L. N. Van Wassenhove (1981). "Production planning: a review." <u>European Journal of Operational Research</u> **7**(2): 101-110.

Geunes, J., H. E. Romeijn, et al. (2002). "Models for integrated production planning and order selection." <u>IIE</u>

Ghosh, J. B. (1997). "Job selection in a heavily loaded shop." Computers & Operations Research **24**(2): 141-145.

Guerrero, H. H. and G. M. Kern (1988). "How to more effectively accept and refuse orders." Production and Inventory Management Journal **29**(4).

Gunther, H. O. (1987). "Planning lot sizes and capacity requirements in a single stage production system." European Journal of Operational Research **31**(2): 223-231.

Hans, E. (2001). Resource loading by branch-and-price techniques, University of Twente. **Ph.D. thesis**.

Harris, F. W. (1913). "How many parts to make at once, Factory " The Magazine of Management **10**: 135-136.

Haupt, R. (1989). "A survey of priority rule-based scheduling." OR Spectrum **11**(1): 3-16.

Hindi, K. S. (1996). "Solving the CLSP by a tabu search heuristic." Journal of the Operational Research Society **47**(1): 150.

Huang, K.-L. and C.-J. Liao (2008). "Ant colony optimization combined with taboo search for the job shop scheduling problem." Computers & Operations Research **35**(4): 1030-1046.

Jalora, A. (2006). Order acceptance and scheduling at a make-to-order system using revenue management, Texas A&M University. **Ph.D. thesis**.

James, R. J. (1956). An extension of Johnson's results on job IDT scheduling. **3:** 201-203.

Jans, R. and Z. Degraeve (2005). Modeling industrial lot sizing problems: A review. ERIM Report Series

Jeong, H.-I., J. Park, et al. (1999). "A batch splitting method for a job shop scheduling problem in an MRP environment." International Journal of Production Research **37**(15): 3583.

Józefowska, J., B. Jurisch, et al. (1994). "Scheduling shops to minimize the weighted number of late jobs." Operations Research Letters **16**(5): 277-283.

Karimi, B., S. M. T. Fatemi Ghomi, et al. (2003). "The capacitated lot sizing problem: a review of models and algorithms." Omega **31**(5): 365-378.

Karimi, B., S. M. T. F. Ghomi, et al. (2006). "A tabu search heuristic for solving the CLSP with backlogging and set-up carry-over." Journal of the Operational Research Society **57**(2): 140-147.

Karni, R. and Y. Roll (1982). "A heuristic algorithm for the multi-item lot-sizing problem with capacity constraints." IIE Transactions **14**(4): 249 - 256.

Karp, R. M. (1972). Reducibility Among Combinatorial Problems. Complexity of Computer Computations. J. W. T. Raymond E. Miller. New York, Plenum Press.

Kern, G. M. and H. H. Guerrero (1990). "A conceptual model for demand management in the assemble-to-order environment." Journal of Operations Management **9**(1): 65-84.

Kirca, Ö. and M. Kökten (1994). "A new heuristic approach for the multi-item dynamic lot sizing problem." European Journal of Operational Research **75**(2): 332-341.

Lambrecht, M. R. and H. Vanderveken (1979). "Heuristic procedures for the single operation, multi-item loading problem." IIE Transactions **11**(4): 319 - 326.

Lee, C.-Y., S. L. Danusaputro, et al. (1991). "Minimizing weighted number of tardy jobs and weighted earliness-tardiness penalties about a common due date." Computers & Operations Research **18**(4): 379-389.

Lewis, H. F. and S. A. Slotnick (2002). "Multi-period job selection: planning work loads to maximize profit." Computers & Operations Research **29**(8): 1081-1098.

Lixiang, J. and R. E. Giachetti (2008). "A queueing network model to analyze the impact of parallelization of care on patient cycle time." Health Care Management Science **11**(3): 248-261.

Low, C., C.-M. Hsu, et al. (2004). "Benefits of lot splitting in job-shop scheduling." The International Journal of Advanced Manufacturing Technology **24**(9): 773-780.

M'Hallah, R. and R. L. Bulfin (2005). "Minimizing the weighted number of tardy jobs on parallel processors." European Journal of Operational Research **160**(2): 471-484.

Maes, J. and L. N. Van Wassenhove (1986). "A simple heuristic for the multi item single level capacitated lotsizing problem." Operations Research Letters **4**(6): 265-273.

Maes, J. and L. V. Wassenhove (1988). "Multi-item single-level capacitated dynamic lot-sizing heuristics: A general review." The Journal of the Operational Research Society **39**(11): 991-1004.

Manne, A. S. (1958). "Programming of economic lot sizes." Management Science **4**(2).

Matignon, R. (2007). Data mining using SAS Enterprise Miner, Wiley Publishing.

Mestry, S., P. Damodaran, et al. (2011). "A branch and price solution approach for order acceptance and capacity planning in make-to-order operations." European Journal of Operational Research.

Miller, B. L. (1969). A queueing reward system with several customer classes. **16:** 234-245.

Nagraj Balakrishnan, V. S. J. W. P. (1996). Rationing capacity between two product classes. **27:** 185-214.

Nahmias, S. (1989). Production and Operations Analysis. Homewood, IL, Irwin Professional Pub

Ng, C. T., T. C. E. Cheng, et al. (2003). "Concurrent open shop scheduling to minimize the weighted number of tardy jobs." Journal of Scheduling **6**(4): 405-412.

Orlicky, J. (1975). Material Requirements planning. New York, McGraw-Hill.

Panwalkar, S. S. and W. Iskander (1977). "A survey of scheduling rules." Operations Research **25**(1): 45-61.

Philippe, B., F. Marta, et al. (2008). Lagrangian bounds for just-in-time job-shop scheduling, Elsevier Science Ltd. **35:** 906-915.

Pinedo, M. (2002). Scheduling: Theory, Algorithms and Systems.

Quadt, D. and H. Kuhn (2008). "Capacitated lot-sizing with extensions: a review." 4OR: A Quarterly Journal of Operations Research **6**(1): 61-83.

Rardin, R. L. (1997). Optimization in Operations Research, Prentice Hall.

Rom, W. O. and S. A. Slotnick (2009). "Order acceptance using genetic algorithms." Computers & Operations Research **36**(6): 1758-1767.

Roundy, R., D. Chen, et al. (2005). "Capacity-driven acceptance of customer orders for a multi-stage batch manufacturing system: models and algorithms." IIE Transactions **37**(12): 1093-1105.

Silver, E. A. and H. C. Meal (1973). "A heuristic for selecting lot size quantities for the case of deterministic time-varying demand rate and discrete opportunities for replenishment." Production and Inventory Management **14**: 64-74.

Silver, E. A. and R. Peterson (1985). Decision Systems for Inventory Management and Production Planning, John Wiley & Sons.

Slotnick, S. A. and T. E. Morton (1996). "Selecting jobs for a heavily loaded shop with lateness penalties." Computers & Operations Research **23**(2): 131-140.

Slotnick, S. A. and T. E. Morton (2007). "Order acceptance with weighted tardiness." Computers & Operations Research **34**(10): 3029-3042.

Svetlana, A. K. (2000). On the complexity of minimizing the number of late jobs in unit time open shop, Elsevier Science Publishers B. V. **100:** 127-132.

Ten Kate, H. (1995). Order acceptance and production control. The Netherlands, University of Groningen. **Ph.D. thesis**.

Thiagarajan, S. and R. Chandrasekharan (2005). Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs, Pergamon Press, Inc. **49:** 463-503.

van den Heuvel, W. and A. P. M. Wagelmans (2006). "An efficient dynamic programming algorithm for a special case of the capacitated lot-sizing problem." Computers & Operations Research **33**(12): 3583-3599.

Verma, M. and R. R. K. Sharma (2010). "A new lagrangian relaxation based approach to solve capacitated lot-sizing problem with backlogging." Global Business & Management Research **2**(2/3): 285-295.

Wagner, H. M. and T. M. Whitin (1958). "Dynamic version of the economic lot size model." Management Science **5**(12 Supplement): 89-96.

Wester, F. A. W., J. Wijngaard, et al. (1992). Order acceptance strategies in a production-to-order environment with setup times and due-dates, Taylor & Francis. **30:** 1313 - 1326.

Zhang, C., Y. Rao, et al. (2008). "An effective hybrid genetic algorithm for the job shop scheduling problem." The International Journal of Advanced Manufacturing Technology **39**(9): 965-974.

VITA

ZHONGPING ZHAI

April 1976	Born, Baiquan, Heilongjiang, China
1994-1998	B.S. Mechanical Engineering Tongji University Shanghai, China
2001-2004	M.S. System Engineering Tongji University Shanghai, China
2003-2006	MIS Developer / Director Tongji University Shanghai, China
2009	Doctoral Candidate in Industrial and Systems Engineering Florida International University Miami, Florida