NEW ALGORITHMS FOR INTEGRATED PRODUCTION AND TRANSPORTATION SCHEDULING PROBLEMS WITH COMMITTED DELIVERY DUE DATES

SHIFU XU

PhD

The Hong Kong Polytechnic University

2022

The Hong Kong Polytechnic University

Department of Logistics and Maritime Studies

New Algorithms for Integrated Production and Transportation Scheduling Problems with Committed Delivery Due Dates

Shifu XU

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

July 2022

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

____________________ (Signed)

Shifu XU (Name of student)

Abstract

Production and transportation, which are two key processes in the supply chains, play critical roles in improving the competitiveness of a company in the global markets. Therefore, integrated production and transportation scheduling becomes more necessary for companies to be responsive to the demands of the customers and reduce the costs to the best of their ability. In this thesis, we focus on two variants of the integrated production and transportation problem faced by manufacturing companies under a make-to-order business strategy and a commit-to-delivery business mode. One variant is to consider the issue of order acceptance. It means that when receiving the orders, the manufacturing company needs to decide which orders are to be accepted and which are to be rejected. The other variant is to incorporate the inventory holding costs incurred during the production and shipping processes of the orders. The original integrated production and transportation problem with committed delivery due dates is known to be strongly NP-hard and the computational hardness can also be applied to these two variants. This thesis contributes to the development of new exact algorithms and approximation algorithms for these two variants.

The first problem we studied in this thesis is the integrated production and transportation scheduling problem with committed delivery due dates and order acceptance (IPTSDA). For this problem, we develop two new exact algorithms that can solve the problem IPTSDA to optimality, and we prove that they can achieve polynomial or pseudo-polynomial running times for two practical cases of problem IPTSDA, respectively. In addition to the two exact algorithms, and by extending the second exact algorithm, we also develop a pseudo-polynomial time approximation scheme for the problem IPTSDA. It not only ensures a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$, but also achieves good computational performance through the computational experiments.

The second problem we studied in this thesis is the integrated production and transportation scheduling problem with committed delivery due dates and inventory holding costs (IPTSDI). The incorporation of inventory holding costs into the objective function makes the problem more complex. To reduce possible inventory holding costs, the manufacturer wants to postpone the production as late as possible. However, this would lead to an increase in the shipping costs due to the decrease in transportation time. Therefore, the manufacturer needs to determine a production plan and a shipping plan that could delicately balance the shipping costs and inventory holding costs. For this problem, we innovatively propose a backward-forward construction algorithm. Based on the backward-forward algorithm, and utilizing our algorithms for problem IPTSDA in the first study, we develop several new exact algorithms with pseudo-polynomial running times for two practical cases of problem IPTSDI. The backward-forward algorithm also helps to develop the new approximation algorithms that can guarantee a worst-case performance ratio of $(1 + \epsilon)$ for any positive constant ϵ.

Keywords: Scheduling; Integrated production and transportation; Commit-to-delivery; Order acceptance; Inventory holding cost; Exact and approximation algorithms

Publications Arising From the Thesis

Chapter 2 and Chapter 3 are based on the following two research papers:

- New Exact and Approximation Algorithms for Integrated Production and Transportation Scheduling with Committed Delivery Due Dates and Order Acceptance, co-authored with Feng Li and Zhou Xu, accepted in *European Journal of Operational Research*.
- Incorporating Inventory Holding Cost in Production and Transportation Integration: New Exact and Approximation Algorithms, co-authored with Feng Li and Zhou Xu, submitted.

Acknowledgments

I would like to take this opportunity to thank all people for helping me in the journey to pursue my PhD degree.

First, I would like to extend my sincere gratefulness to my supervisor, Prof. Zhou Xu for his invaluable and continuous support, not only in academics but also in the other aspects of my life. His instructive guidance, incisive analysis, rigorous academic standards and proficiency in different areas helped me to go through different difficulties in academia and will continue to benefit me in my future career.

Second, I would like to thank Prof. Feng Li for his constructive advice to improve the quality of our research and for his contribution to the chapters of this thesis. Also, I would like to express my gratitude to Prof. Pengfei Guo, Prof. Yulan Wang and Dr. Anran Li for their expertise in their research fields and willingness to share with me their knowledge and unique perspectives on the research problems.

Thirdly, I am also grateful to the committee members of my thesis including the external examiners, Prof. Yanzhi Li and Prof. Feng Chen, as well as the BoE chair, Prof. Kai Pan. Their constructive suggestions and insightful comments greatly improve the quality of my thesis.

Special thanks also go to other faculties in PolyU, Prof. Li Jiang, Prof. Miao Song and Prof. Eric Ngai and so on, who provide chances for me to systematically know and be familiar with the necessary knowledge and skills for a qualified PhD student. In addition, I would like also to dedicate my thanks to the administrative staff in the General Office of LMS, RO, GS and in other organizational units of PolyU for providing assistance and services to my study.

Moreover, I would like to thank my roommates and classmates in MN037 and MN038. The activities with them help to enrich my life in PolyU and Hong Kong.

Finally, I would like to express my appreciation to my parents, my brother and especially my girlfriend Qiaoyu Wu, for their encouragement, support and love. They are indispensable parts of my PhD study and my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The increasingly competitive global market brings more challenges to the management of the supply chains (Mangan and Lalwani (2016)). Production and transportation, which are two key processes in the supply chains, will largely influence the efficiency and cost in the management of the supply chain (see the detailed description in Sarmiento and Nagi (1999); Erengüç et al. (1999); Goetschalckx et al. (2002); Chen (2004)). Therefore, to gain advantages in the fierce competition, many manufacturers adopt *make-to-order* strategy, which means that the production of products begins only after receiving orders (see examples in Li et al. (2005); Chen and Vairaktarakis (2005); Pundoor and Chen (2005); Chen and Pundoor (2006); etc.). Also, these manufacturers would deliver the completed products to the customers before or on the committed delivery due dates and pay the associated fees by themselves, which is referred to as *commit-to-delivery* strategy (Stecke and Zhao (2007)). Normally, these manufacturers would use third-party logistics (3PL) companies to accomplish the delivery tasks after the completion of production (Marasco (2008); Chen (2010)). These 3PLs can provide multiple shipping modes with different shipping times and shipping costs, for example, the one-day and two-day delivery services of UPS and FedEx (Li et al. (2020); Yang et al. (2021)). Moreover, faster shipping modes will take more costs.

It brings several advantages for a manufacturer to operate under the make-to-order and commit-to-delivery strategies (Stecke and Zhao (2007); Li et al. (2020)). The manufacturer can produce customized products and deliver them to the customers in a short time, which will reduce the inventory to a certain extent. Intuitively, the manufacturer can select a shipping mode with shorter shipping times when a production complete day for an order is close to the committed due date for delivery required by a customer; and if the production is finished earlier than the committed delivery due date, the manufacturer can choose a slower shipping mode. Therefore, the combination of these two strategies can decrease the possibility of missing the delivery due dates and increases the flexibility to schedule the production and transportation which can reduce the costs as much as possible.

However, there are still delayed delivery of products due to insufficient production capacity (Korpela et al. (2002); Stecke and Zhao (2007)), inappropriate time schedules (Herrmann (2006); Ghaleb et al. (2020)) and so on. Hence, it becomes more necessary for these companies to jointly schedule the production and transportation in the out so that to improve the service levels and reduce costs (see examples in Ahmadi et al. (2005); Leung et al. (2005); Stecke and Zhao (2007); Armstrong et al. (2008); Liu and Liu (2020); etc.). Therefore, these manufacturing companies are faced with the *Integrated Production and Transportation Scheduling Problems with Committed Delivery Due Dates* (IPTSD) (Stecke and Zhao (2007)).

In this thesis, we consider two variants of problem IPTSD. The first problem is *Integrated Production and Transportation Scheduling Problem with Committed Delivery Due Dates and Order Acceptance* (IPTSDA) where the manufacturer needs to determine whether to accept or reject an order when receiving it. The issues of order acceptance are commonly studied in the manufacturing fields (see examples in Kolisch (1998); Calosso et al. (2003); Roundy et al. (2005); Ivănescu et al. (2006); Rom and Slotnick (2009); Tarhan and Oğuz (2022); etc.). With order acceptance decisions being taken into account, the manufacturer needs to balance the trade-off between the revenue earned by producing an order as well as its associated costs and the penalty cost of rejecting it due to the limitation of the production capacity and lead time. The second problem is *Integrated Production and Transportation Scheduling Problem* *with Committed Delivery Due Dates and Inventory Holding Cost* (IPTSDI). Although the commit-to-delivery business mode is designed for reducing the inventory, the inventory costs may still exist during the processes of production and shipping (see examples in Chan et al. (2002); Hwang (2010); Li et al. (2017); etc.). Therefore, a manufacturer needs to subtly settle the production plan and shipping plan, so as to minimize the total costs including the inventory holding costs.

1.2 Literature Review

In this thesis, we mainly focus on the production and outbound logistics of a manufacturer. While outbound logistics concentrate on the distribution from the manufacturer to customers, inbound logistics focus on the transportation of the material flow from suppliers to the manufacturer (see Cohen and Lee (1988); Vidal and Goetschalckx (1997); Hall and Potts (2003) for an overview). Although inbound logistics problems could have similar objective functions to outbound logistics studied in this thesis, such as minimizing total delivery costs and inventory holding costs, etc., these two logistics processes have several differences. A manufacturer could start production until all raw materials and parts from its suppliers arrive. Therefore coordination of the manufacturer and supplier can be an issue in inbound logistics (see Chen and Hall (2007); Sawik (2009) for an overview). In addition, since the suppliers may be located in different places, distribution strategies (eg., direct, milk-run and cross-dock) are intensively studied in inbound logistics (see Berman and Wang (2006) for introduction and typical examples in the automobile industry in Wang and Chen (2019); Baller et al. (2022)). However, the outbound logistics in this thesis mainly focus on the interaction between a manufacturer and 3PLs that can provide delivery services with different shipping modes. And the manufacturer needs to make a joint schedule of production plan and shipping plan (see Stecke and Zhao (2007) for a industrial example in Dell Technologies).

The studies in this thesis focus on production planning and transportation scheduling, which belong to the planning level problems in supply chains. There are also plenty of studies on the detailed scheduling level problems in supply chains, which examine the optimal schedules of jobs and facilities to complete the production task (see Kreipl and Pinedo (2004) as an overview of these two categories of problems in supply chains).

Due to its significant theoretical and practical importance, integrated production and transportation scheduling has become an increasingly important research topic (see Chen, 2010, for a state-of-the-art survey). Existing studies take into account various production configurations, order restrictions, delivery characteristics, and objective functions for optimization. However, unlike problem IPTSDA and problem IPTSDI examined in this thesis, the problems investigated by most of these studies consider only a single shipping mode for order delivery (e.g., see recent studies by Ullrich (2013), Agnetis et al. (2014), Mensendiek et al. (2015), Azadian et al. (2015), Li et al. (2015), Geismar and Murthy (2015), Sawik (2016), Li et al. (2017), Zhang and Song (2018), Tang et al. (2019), and Bachtenkirch and Bock (2022)). There are several studies on problems considering multiple shipping modes, which differ from problem IPTSD, in production decisions and shipping cost functions (e.g., see Wang and Lee (2005), Chen and Lee (2008) and Agnetis et al. (2016)).

Zhong et al. (2010) study problem IPTSD, where all received orders must be accepted, and the shipping cost function for multiple shipping modes is linearly non-decreasing in shipping quantity and linearly non-increasing in shipping time. By a reduction to a special case of the problem studied in Stecke and Zhao (2007), Zhong et al. (2010) show that problem IPTSD is strongly NP-hard, and they also develop a polynomial time approximation algorithm to solve the problem with a worst-case performance ratio of 2, which is later improved to 5/3 by Zhong (2015).

Stecke and Zhao (2007) study a problem that is similar to problem IPTSD, in that it has a more general form of the shipping cost function that is linearly non-decreasing in shipping quantity and convexly non-increasing in shipping time. However, for this problem, no polynomial time approximation algorithms with constant worst-case performance ratios are known. When partial deliveries are allowed, Stecke and Zhao (2007) show that a simple non-preemptive production plan by an earliest-due-date-first-scheduling strategy is optimal. For the problem studied in Stecke and Zhao (2007), Melo and Wolsey (2010) derive several integer programming models. Different from the two variants of problem IPTSD studied in this thesis, the problems in the above-mentioned literature do not consider the decision of order acceptance and inventory holding costs in their objective function.

As shown in Zhong et al. (2010), problem IPTSD is strongly NP-hard. This implies that unless NP=P, there exists no exact algorithm that can solve problem IPTSD to optimality in polynomial time, and no FPTAS that has a fully polynomial running time and achieves a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$. Since problem IPTSDA and problem IPTSDI are variants of problem IPTSD, these computational hardness results are also applicable to problem IPTSDA and problem IPTSDI.

Problem IPTSDA studied in this thesis considers to incorporate the order acceptance decisions. There are extensive studies on machine scheduling problems that also have taken into account order acceptance decisions (see Slotnick (2011) for a survey). This thesis follows the similar setting for order acceptance in these machine scheduling studies. That is to denote the order acceptance decisions as decision variables that will influence the objective functions and constraints of the problems. There are also several recent studies that integrate machine scheduling with transportation (see more examples in Shams and Salmasi (2014); Liou and Hsieh (2015); Jiang et al. (2017); Sarvestani et al. (2019); etc.). For example, Aminzadegan et al. (2021) use two meta-heuristic solution approaches, namely, an adaptive genetic algorithm and a tabu search, to solve a single machine scheduling problem with order acceptance, delivery scheduling, and resource allocation. Zhong et al. (2022) develop two approximation algorithms for a multiple machine scheduling problem with outbound delivery taken into account, and with rejected orders being outsourced. However, to the best of our knowledge, the literature rarely studies approximation algorithms with a constant worse case performance ratio to solve problem IPTSDA.

Similar to problem IPTSDI studied in this thesis, some studies also consider the inventory holding costs in the integrated production and scheduling problem. For example, Sun et al. (2015) consider job allocation and scheduling problem in multiple factories with minimized total costs including production cost, shipping cost and storage cost (inventory holding cost). Guo et al. (2017) study integrated production and transportation scheduling problem with product batch-based delivery by modeling it as an order assign problem. And the objective is to minimize the total costs of all product batches including the production cost, inventory holding costs and so on. Li et al. (2017) consider production integration problems with inventory and delivery where each order requires services within a delivery time window. Similarly, Han et al. (2019) examine a problem of production and outbound scheduling integration with inventory being taken into account in a three-stage supply chain. There are also some studies considering inventory holding costs in the area of machine scheduling (Li et al. (2008), Wang and Cheng (2009), Ma et al. (2013), Hajiaghaei-Keshteli et al. (2014), Chevroton et al. (2021), Bachtenkirch and Bock (2022), etc.). However, our study on problem IPTSDI considers both the inventory holding costs and committed delivery due dates for the integration of production and scheduling problems and proposes an approximate scheme that can yield close-to-optimal solutions to problem IPTSDI.

To the best of our knowledge, three studies on the variants of IPTSD are closely related to our research on problem IPTSDI. The first study is from Li et al. (2020) and the second is from Yang et al. (2021). Both studies incorporate inventory holding costs into problem IPTSD. Li et al. (2020) consider a more general form of shipping cost that is no-linear in both shipping quantity and shipping time. In addition, for the case when the planning horizon is two days, they also propose a fully polynomial time approximation scheme (FPTAS) and they develop a column generation-based heuristic algorithm for the general case. In Yang et al. (2021), they develop exact and heuristics algorithms for problem IPTSD with order dependent inventory holding costs for the case when the planning horizon is fixed. The major differences between our research on problem IPTSDI and these two closely related studies are: i) we develop a pseudo-polynomial time exact algorithm for the case when the number of possible order quantities are fixed; ii) we also develop a pseudo-polynomial time exact algorithm for the case when the planning horizon is fixed; iii) based on the second exact algorithm, we propose a pseudo-polynomial time approximation scheme which guarantees a worst-case performance ratio of $(1 + \epsilon)$ with $\epsilon > 0$.

The third study that is closely related to our research on problem IPTSDI is from Chapter 2 of this thesis which focuses on problem IPTSDA. Although both studies develop exact and approximation algorithms for the variants of problem IPTSD, the research on problem IPTSDI has some obvious differences from that of problem IPTSDA. The objective function of problem IPTSDI is to minimize the total shipping costs and inventory holding costs while that of problem IPTSDA is to minimize the total shipping costs and rejection costs. The consideration from rejection cost to inventory holding costs changes the structure of the algorithms and increases the complexity of the problem. In the study on problem IPTSDA, since there is no inventory holding cost, the manufacturer can produce to its production capacity each day so as to ship the products as early as possible. This can decrease the shipping cost due to a longer shipping time. This idea provides the basis for the development of the exact algorithms and approximation scheme in the study problem IPTSDA. However, this idea is inapplicable to problem IPTSDI due to the existence of possible inventory holding costs. The manufacturer needs to balance the trade-off between the inventory holding costs and shipping costs. We innovatively propose a backward-forward algorithm that can construct an optimal solution by aggregating the production quantities of all orders in a day into a daily production quantity given a shipping plan. The exact algorithms and approximation scheme for problem IPTSDI are based on the backward-forward algorithm. Moreover, for the case when the number of possible order quantities is fixed, different from the algorithm of problem IPTSDA, we utilize the idea similar to the zero-inventory policy that constructs a solution from several production subsequences, where each production subsequence has no inventory at the end of its production completion day.

1.3 Summary of Contributions

In this thesis, we mainly contribute to develop new polynomial time and pseudo-polynomial time exact algorithms under the case where the number possible of order quantities is bounded by a constant and the case where the planning horizon is bounded by a constant for the two variants of problem IPTSD by considering order acceptance and inventory holding cost, respectively. We also propose pseudo-polynomial time approximation algorithms with constant worse case performance ratios for these two variants of problem IPTSD.

The first problem we considered in this thesis is problem IPTSDA. Except for a production plan and a shipping plan, the manufacturer also needs to determine an order acceptance plan. It means that when receiving the orders, the manufacturing company needs to decide which orders are to be accepted and which are to be rejected with certain rejection costs incurred by orders being rejected. Therefore, the manufacturer should balance the trade-off between the revenue earned by an order with its associated costs (production cost and shipping cost) and the penalty of directly rejecting it. For this problem, we develop two new exact algorithms that are capable to yield optimal solutions to the problem IPTSDA. We also prove that they can achieve polynomial or pseudo-polynomial running times for two practical cases of problem IPTSDA, respectively. In addition to the two exact algorithms, and by extending the second exact algorithm, we also develop a pseudo-polynomial time approximation scheme for the problem IPTSDA. It not only ensures a worst-case performance ratio of $(1+\epsilon)$ for any fixed $\epsilon > 0$, but also achieves good computational performance through the computational experiments.

The second problem we considered is problem IPTSDI. The incorporation of inventory holding costs into the objective function makes the problem more complex. The algorithms proposed for problem IPTSD and its related problem that does not consider inventory holding cost are not applicable to problem IPTSDI. To reduce possible inventory holding costs, the manufacturer wants to postpone the production as late as possible. However, this would lead to an increase in the shipping costs due to the decrease in transportation time. Therefore, the manufacturer needs to determine a production plan and a shipping plan that could delicately balance the shipping costs and inventory holding costs. For this problem, we innovatively propose a backward-forward construction algorithm. Based on the backward-forward algorithm, and utilizing our algorithms for problem IPTSDA in the first study, we develop new exact algorithms with pseudo-polynomial running times for two practical cases of problem IPTSDI. The backward-forward algorithm also helps to develop the new approximation algorithms that can guarantee a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$.

The remainder of this thesis proceeds as follows: Chapter 2 and Chapter 3 will separately examine problem IPTSDA and problem IPTSDI as well as their associated solution algorithms. And Chapter 4 concludes this thesis and discusses possible research directions arising from these problems.

Chapter 2

Integrated Production and Transportation Scheduling with Committed Delivery Due Dates and Order Acceptance

2.1 Introduction

We investigate an integrated production and transportation scheduling (IPTS) problem commonly faced by manufacturing companies under a make-to-order business strategy and a commit-to-delivery business mode. Under the make-to-order business strategy, the manufacturing company starts to produce products only after receiving orders from customers. Under the commit-to-delivery business mode, the manufacturing company is responsible for shipping costs, and needs to guarantee a *committed delivery due date* for each order, meaning that the customer of the order must receive the products on or before this date. To ship the products to customers, the manufacturing company often uses third-party logistics (3PL) providers, which usually offer multiple shipping modes with different shipping time guarantees and different shipping costs. In general, the faster the shipping mode, the higher the shipping cost. When the manufacturing company cannot satisfy all the orders received, it needs to decide which orders are to be accepted and which are to be rejected, with certain rejection costs incurred by orders being rejected. Accordingly, the manufacturing company encounters an IPTS problem with committed delivery due dates and order acceptance (referred to as problem IPTSDA). The problem needs to first accept a subset of orders, and then determine a production plan and a shipping plan for the accepted orders that meet their committed delivery due dates. The objective of the problem is to minimize the total cost, including the operating cost of each accepted order and the rejection cost of each rejected order.

This chapter contributes to the development and analysis of new exact and approximation algorithms for problem IPTSDA. In particular, we develop two new exact algorithms that can solve problem IPTSDA to optimality, and we prove that they achieve polynomial or pseudo-polynomial running times for two practical cases of problem IPTSDA, respectively. These two cases impose different restrictions on two problem parameters, namely, the total number of possible order quantities η and the length of planning horizon m. In practice, m and η are often bounded by certain constants. For example, when a manufacturer offers only a limited number of order quantities for customers to choose from, such as either 10 or 20 units of their products, an upper bound on η is imposed. In Dell Technologies, a global computer manufacturer, most of its individual customer orders require only one or two computers (Stecke and Zhao (2007)), and as a result, η is often bounded by 2. Moreover, when a manufacturer has a short planning horizon for production and transportation, such as a planning horizon of two or three days, an upper bound on m is imposed. In BESTORE, a leading snack manufacturer in China, orders for snack gift boxes placed by individual customers must be delivered within one to two days, and the planning horizon for production and transportation is set to be three days. In this situation, m is bounded by 3.

Accordingly, for the case where η is bounded by a constant, we develop an exact algorithm that runs in polynomial time if the total number of orders is polynomially bounded by the input size of the problem, and which runs in pseudo-polynomial time otherwise. Utilizing this, we show that when orders are allowed to be split, problem IPTSDA can be solved in pseudo-polynomial time. For the case where m is bounded by a constant, we develop an exact algorithm that runs in pseudo-polynomial time. The efficiency of these two exact algorithms in computational experiments has also been examined.

In addition to the two exact algorithms, and by extending the second exact algorithm, we also develop a pseudo-polynomial time approximation scheme for problem IPTSDA, which ensures a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$. As problem IPTSDA is unlikely to have an FPTAS unless NP=P, one may expect to develop a polynomial time approximation scheme (PTAS) at best. For this, our pseudo-polynomial time approximation scheme makes positive progress. Moreover, computational results show that this approximation scheme also performs well in producing close-to-optimal solutions for problem instances that are randomly generated.

Although our new exact and approximation algorithms are developed for problem IPTSDA, they can also be applied to more general problems, such as those with shipping cost functions that are linearly non-decreasing in shipping quantity and convexly non-increasing in shipping time. The analytical results derived for our algorithms can accordingly also be extended. Moreover, our newly developed algorithms and their performance guarantees are also applicable to problem IPTSD, which is a special case of problem IPTSDA.

The remainder of this chapter is organized as follows: We describe the problem in Section 2.2, and analyze its optimality properties in Section 2.3. We then depict the two exact algorithms in Section 2.4 and the pseudo-polynomial time approximation scheme in Section 2.5. Our computational results are presented in Section 2.6, and the chapter is summarized in Section 2.7.

2.2 Problem Description and Formulation

Problem IPTSDA, studied in this chapter, extends the setting of problem IPTSD in Zhong et al. (2010) by incorporating order acceptance decisions. Let us consider a planning horizon of m days, denoted by $T = \{1, 2, \ldots, m\}$. At the beginning of the planning horizon, a manufacturer receives a set of n orders, denoted by $N = \{1, 2, \ldots, n\}$. Each order i requires a quantity of q_i units of certain products and has a committed delivery due date $d_i \in T$ with $d_i \geq 1$. After the orders are received, the manufacturer needs to determine which orders are to be accepted and which are to be rejected. More specifically, if the manufacturer rejects order $i \in N$, the products for order i do not need to be produced or shipped, but a rejection cost r_i is incurred. For those orders that are accepted, they need to be produced on a single production line and then delivered to their respective customers. Accordingly, if order i is accepted, its products must be received by its customer on or before day d_i. Let c denote the production capacity of each day so that the total quantity of the products produced on each day cannot exceed c. We herein follow the setting of problem IPTSD in Zhong et al. (2010) to assume that all the products have the same unit weight and the same production capacity requirement, and that each order quantity does not exceed the production capacity (i.e., $q_i \leq c$ for $i \in N$). These assumptions are consistent with common situations, such as those in the computer industry. Let p denote the unit production cost. Without loss of generality, we assume that $r_i > pq_i$ for $i \in N$, that is, for each order its rejection cost is always larger than its production cost, because otherwise such an order can be rejected without increasing the total cost.

Consider that the manufacturer uses a 3PL provider for transportation who picks up the finished products and ships them out at the end of each day. In this chapter, we also assume that these 3PL providers do not offer partial delivery services. That is, products of each order must be shipped out together on the same day after they complete production. The 3PL provider offers multiple shipping modes, with different shipping times and different shipping costs. Each shipping mode is associated with a shipping time of s days, where $s \in \{0, 1, \ldots, m-1\}$, as well as a shipping cost function $G(s, y)$, where y is the shipping quantity. We refer to the shipping mode with shipping time s as the s-day shipping mode. In line with the setting of problem IPTSD in Zhong et al. (2010), we consider a shipping cost function $G(s, y)$, which is linearly increasing with y and linearly decreasing with s:

$$G(s, y) = y(\alpha - \beta s), \tag{2.1}$$

where parameters α and β are positive. To ensure that the shipping cost is always positive, even for the slowest shipping mode, that is, the $(m-1)$-day shipping mode, α and β satisfy that

$$\alpha - \beta(m - 1) > 0. \tag{2.2}$$

The shipping cost function $G(s, y)$ applies to all orders, regardless of the customer locations, due to common practice in domestic shipping (see Stecke and Zhao (2007) for some examples). For more general shipping cost functions that are linearly non-decreasing in y and convexly non-increasing in s, we will show that the algorithms we present later are also applicable, and that the analytical results derived for them can also be extended.

Accordingly, a *solution* to problem IPTSDA, which needs to be decided by the manufacturer, includes (i) *an order acceptance plan* about which orders to be accepted; (ii) *a production plan* about the quantity of products for each accepted order that must be produced on each day; and (iii) *a shipping plan* about when to ship out the products for delivery for each accepted order. Since the shipping cost function $G(s, y)$ is decreasing in shipping time s, it is always cost-efficient for the manufacturer to choose the slowest shipping mode for each order whereby the customer receives the products exactly on the committed delivery due date. A solution is *feasible* if it satisfies that the customer of each accepted order receives the products ordered on or before the order's committed delivery due date, and that the production capacity of each day cannot be exceeded. A trivial feasible solution is to reject all the orders.

Problem IPTSDA aims to find *an optimal solution* that is feasible and minimizes the total cost. The total cost includes the operating cost for each accepted order and the rejection cost of each rejected order. Although the operating cost includes the production cost, inventory holding cost, and shipping cost, we follow the setting of problem IPTSD in Zhong et al. (2010) to assume zero inventory holding cost, because due to the make-to-order strategy, orders are delivered soon after completion of their production, and inventory holding costs are thus negligible. Moreover, we can transform each problem instance to an instance with zero production cost such that the optimal solution is unchanged. To see this, let decision variables λ_i for $i \in N$ represent the acceptance plan of a solution, with each $\lambda_i \in \{0, 1\}$ indicating whether or not order i is accepted, and being equal to 1 only if order i is accepted. The total cost of production and rejection for a feasible solution equals $\sum_{i \in N}(1 - \lambda_i) \cdot r_i + p \sum_{i \in N} \lambda_i q_i = \sum_{i \in N}(1 - \lambda_i)(r_i - pq_i) + p \sum_{i \in N} q_i$. Since $p \sum_{i \in N} q_i$ is a constant, we can exclude it in the total cost without changing the optimal solution. We now modify the rejection cost for order $i \in N$ to be $r'_i = r_i - pq_i$, which implies that the total cost of production and rejection for a feasible solution is $\sum_{i \in N}(1 - \lambda_i) \cdot r'_i$. Therefore, we obtain a new problem instance with zero production cost and with r'_i for $i \in N$ as the rejection costs, such that the optimal solution is unchanged. Hence, we can assume zero production cost for problem IPTSDA in the remainder of this chapter.

In addition to decision variables λ_i for $i \in N$ defined above for the order acceptance plan, other decision variables are introduced as follows. Let decision variables x_{it} for $i \in N$ and $t \in T$ represent the production plan of a solution, with each $x_{it} \in \mathbb{Z}_+$ indicating the number of units of the products produced for order i on day t, where $\mathbb{Z}_+$ denotes the set of non-negative integers. Let decision variables z_{it} represent the shipping plan of a solution, with each $z_{it} \in \{0, 1\}$ indicating whether or not the products for order i are shipped out on day t. If $z_{it} = 1$, that is, the products for order i are shipped out on day t, then since the slowest shipping mode must be chosen, by (2.1) the shipping cost for order i is $G(d_i - t, q_i) = q_i[\alpha - \beta(d_i - t)]$.

Table 2.1 summarizes the notations introduced above. We can now formulate problem IPTSDA by the following integer linear programming model ILP.

Table 2.1: Basic notation for problem IPTSDA.

m	Number of days in the planning horizon
$T = \{1, 2, \ldots, m\}$	Set of m days in the planning horizon
n	Number of orders
$N = \{1, 2, \ldots, n\}$	Set of n orders
c	Production capacity of each day
q_i	Quantity of the products for order i
r_i	Rejection cost for rejecting order i
$d_i \in \{1, 2, \cdots, m\}$	Committed delivery due date for order i
$G(s, y) = y(\alpha - \beta s)$	Shipping cost function with shipping time of s days and shipping quantity y
$\lambda_i \in \{0, 1\}$	1, if order i is accepted, and 0, if order i is rejected
$x_{it} \in \mathbb{Z}_+$	Production quantity for order i on day t
$z_{it} \in \{0, 1\}$	1, if the products for order i are shipped out on day t, and 0, otherwise

$$
\begin{aligned}
\text{(ILP)} \quad \min \quad & \sum_{i \in N} \sum_{t \in T} G(d_i - t, q_i) \cdot z_{it} + \sum_{i \in N} (1 - \lambda_i) r_i && (2.3)\\
\text{s.t.} \quad & \sum_{i \in N} x_{it} \le c, \ \text{for } t \in T, && (2.4)\\
& \sum_{t \in T} x_{it} = q_i \lambda_i, \ \text{for } i \in N, && (2.5)\\
& \sum_{t=1}^{m} z_{it} = \lambda_i, \ \text{for } i \in N, && (2.6)\\
& \sum_{t=d_i+1}^{m} z_{it} = 0, \ \text{for } i \in N, && (2.7)\\
& \sum_{t'=1}^{t} q_i z_{it'} \le \sum_{t'=1}^{t} x_{it'}, \ \text{for } i \in N, t \in T, && (2.8)\\
& \lambda_i \in \{0, 1\}, \ \text{for } i \in N, && (2.9)\\
& x_{it} \in \mathbb{Z}_+, z_{it} \in \{0, 1\}, \ \text{for } i \in N, \ t \in T. && (2.10)
\end{aligned}
$$

In model ILP, the objective function (2.3) is to minimize the total shipping and rejection cost. Constraint (2.4) ensures that the daily production quantity does not outreach the production capacity c. Constraint (2.5) ensures that all the required units of the products are produced for each accepted order. Constraints (2.6) and (2.7) ensure that the products for each accepted order are shipped out and can be received by the customer on or before the committed delivery date. Constraint (2.8) ensures that the products for each order are shipped out only after production is finished. Constraints (2.9) and (2.10) are integral and binary constraints on decision variables λ_i, x_{it}, and z_{it}.

We use $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to represent a solution to model ILP, where $\boldsymbol{\lambda}$ represents the vector of variables λ_i for $i \in N$, $\mathbf{x}$ represents the vector of variables x_{it} for $i \in N$ and $t \in T$, and $\mathbf{z}$ represents the vector of variables z_{it} for $i \in N$ and $t \in T$.

2.3 Optimality Properties

In this section, we derive several properties such that there always exists an optimal solution to model ILP of problem IPTSDA that satisfies these properties. We will later utilize these properties in our algorithm development. Consider a sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_{|\sigma|})$ of some orders in N, where $|\sigma|$ is the length of the sequence, satisfying $0 \leq |\sigma| \leq n$, and where each $\sigma_j \in N$ indicates the j-th order of σ. For each j, let $\bar{Q}_j = \sum_{j'=1}^{j} q_{\sigma_{j'}}$ indicate the total product quantity of the first j orders of sequence σ. Let $t_j = \lceil \bar{Q}_j / c \rceil$ indicate the minimum number of days required to produce products for these first j orders. Define $\bar{Q}_0 = 0$ and $t_0 = 0$. Since $q_{\sigma_j} \leq c$, we have $\bar{Q}_{j-1} \leq \bar{Q}_j \leq \bar{Q}_{j-1} + c$ and $t_{j-1} \leq t_j \leq t_{j-1} + 1$. From σ, we can construct a solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to model ILP by the following procedure:

Step 1. For each $j = 1, 2, \cdots, |\sigma|$, follow the steps below to determine λ_{σ_j} as well as $x_{\sigma_j,t}$ and $z_{\sigma_j,t}$ for $t \in T$:

Step 1.1. Accept order σ_j, i.e., set $\lambda_{\sigma_j} = 1$;

Step 1.2. No products are produced for order σ_j before day t_{j-1} or after day t_j, i.e., set $x_{\sigma_j,t} = 0$ for $t \in \{1, 2, \cdots, t_{j-1} - 1, t_j + 1, \cdots, m\}$, noting that $t_{j-1} \leq t_j \leq t_{j-1} + 1$ as shown above;

Step 1.3. Produce as many products as possible for order σ_j on day t_{j-1} if there is any production capacity left for the first t_{j-1} days, i.e., set $x_{\sigma_j,t_{j-1}} = \min\{q_{\sigma_j}, (t_{j-1} \cdot c) - \bar{Q}_{j-1}\}$;

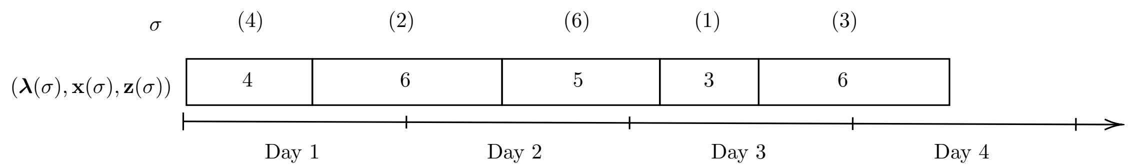
Step 1.4. If there are remaining products that have not been produced for order σ_j, i.e., $q_{\sigma_j} - x_{\sigma_j,t_{j-1}} > 0$, implying that $t_j = t_{j-1} + 1$ (due to $\bar{Q}_j = \bar{Q}_{j-1} + q_{\sigma_j} > \bar{Q}_{j-1} + x_{\sigma_j,t_{j-1}} = \bar{Q}_{j-1} + t_{j-1} \cdot c - \bar{Q}_{j-1} = t_{j-1} \cdot c$), then produce all the remaining products for order σ_j on day t_j, i.e., set $x_{\sigma_j,t_j} = q_{\sigma_j} - x_{\sigma_j,t_{j-1}}$;

Step 1.5. Ship out products for order σ_j on day t_j, i.e., set $z_{\sigma_j,t_j} = 1$ and $z_{\sigma_j,t} = 0$ for $t = \{1, 2, \cdots, t_j - 1, t_j + 1, \cdots, m\}$.

Step 2. For each $j \in N \setminus \{\sigma_1, \ldots, \sigma_{|\sigma|}\}$, set $\lambda_j = 0$ and $x_{jt} = z_{jt} = 0$ for $t \in \{1, \ldots, m\}$.

Let $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ indicate the solution constructed above from the sequence σ. Consider the example shown in Figure 2.1 where the order sequence $\sigma = (4, 2, 6, 1, 3)$ is selected from $N = \{1, 2, \cdots, 6\}$, and where the production capacity of each day $c = 7$. Accordingly, $t_1 = \lceil q_4/c \rceil = \lceil 4/7 \rceil = 1$, $t_2 = \lceil (q_4 + q_2)/c \rceil = \lceil (4 + 6)/7 \rceil = 2$, $t_3 = \lceil (q_4 + q_2 + q_6)/c \rceil = \lceil (4 + 6 + 5)/7 \rceil = 3$, $t_4 = \lceil (q_4 + q_2 + q_6 + q_1)/c \rceil = \lceil (4 + 6 + 5 + 3)/7 \rceil = 3$, and $t_5 = \lceil (q_4 + q_2 + q_6 + q_1 + q_3)/c \rceil = \lceil (4 + 6 + 5 + 3 + 6)/7 \rceil = 4$. From the construction procedure above, we obtain a solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$, where order 5 is rejected and the other five orders are accepted. Among the five accepted orders, order 4 has four units produced and shipped out on day 1, order 2 has three units produced on day 1, three units produced on day 2, and six units shipped out on day 2, order 6 has four units produced on day 2, one unit produced on day 3, and five units produced on day 3, order 1 has three units produced and shipped out on day 3, and order 3 has three units produced on day 3, three units produced on day 4, and six units shipped out on day 4. For each of these five accepted orders, its production completion day is the same as its shipped-out day, which is shown by the right side of each order's rectangle in Figure 2.1.

Figure 2.1: An example for the construction of a solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ from an order sequence σ.



The order sequence $\sigma = (4, 2, 6, 1, 3)$ is indicated by a sequence of order indices in brackets selected from $N = \{1, 2, \cdots, 6\}$, the five orders in σ are indicated by rectangles with order quantities shown inside and represented by the widths of the rectangles, a planning horizon of 4 days is indicated by four consecutive segments on an arrow line, and the production capacity of each day $c = 7$ is represented by the width of each segment.

From the construction procedure above, we can establish Lemma 2.1 below for solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ constructed from any order sequence σ. Let $A(\sigma) = \{\sigma_1, \ldots, \sigma_{|\sigma|}\}$ indicate the set of the accepted orders in solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$.

Lemma 2.1. *For any order sequence σ, solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ satisfies that (i) there is no production after day $\lceil \sum_{i \in A(\sigma)} q_i/c \rceil$, (ii) the total production quantity of each day $t \in \{1, 2, \cdots, \lceil \sum_{i \in A(\sigma)} q_i/c \rceil - 1\}$ equals c, and (iii) the total production quantity of day $\lceil \sum_{i \in A(\sigma)} q_i/c \rceil$ is less than or equal to c.*

Proof. To simplify the notation in this proof, we use $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to denote $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ and use A to denote $A(\sigma)$. As we defined earlier, $t_j = \lceil \bar{Q}_j/c \rceil$ for each $j \in \{1, 2, \cdots, |\sigma|\}$. From Step 1.2 we know that there is no production after day $t_{|\sigma|} = \lceil \sum_{i \in A} q_i/c \rceil$, and thus (i) of Lemma 2.1 is proved. We now prove (ii) by contradiction. Suppose that there exists $t'' \in \{1, 2, \cdots, t_{|\sigma|} - 1\}$ such that the total production quantity of day t'' is not equal to c. Without loss of generality, we assume that the total production quantity of day t' with $t' \leq t'' - 1$ is equal to c. Since $q_{\sigma_1} \leq c$ we have $t_1 = 1$, which, together with $t_{j-1} \leq t_j \leq t_{j-1} + 1$ for each j, implies that $\{t_j | 1 \leq j \leq |\sigma|\} = \{1, 2, \cdots, t_{|\sigma|}\}$. From (i) and due to the production capacity c, we know that the total production quantity of day $t' \in \{1, 2, \ldots, t_{|\sigma|}\}$ is larger than 0. Thus, there exists an index $j \in \{1, 2, \cdots, |\sigma|\}$ such that $t_j = t'' + 1$ and $t_{j-1} = t''$.

Consider the moment right after the construction of $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ determines $x_{\sigma_j,t}$ and $z_{\sigma_j,t}$ for $t \in T$. At this moment, due to Step 1.3 of the construction, the total production quantity of the first $t_{j-1} = t''$ days should not exceed $t_{j-1} \cdot c$. Since the total production quantity of day t' for each $t' \leq t_{j-1} - 1$ is equal to c, the total production quantity of day t_{j-1} cannot exceed c, and thus it must be less than c. This implies that the total production quantity of the first t_{j-1} days must be less than $t_{j-1} \cdot c$, i.e., $\bar{Q}_{j-1} + x_{\sigma_j,t_{j-1}} < t_{j-1} \cdot c$. Thus, from Step 1.3 and Step 1.4 of the construction we know that $x_{\sigma_j,t_{j-1}} = q_{\sigma_j}$, implying that all the products for order σ_j must be produced on day t_{j-1}. Therefore, all the products for the first j orders of σ are produced before or on t_{j-1}, which is earlier than t_j, leading to a contradiction with the definition of t_j. Hence, (ii) of Lemma 2.1 is proved. From (i) and (ii) of Lemma 2.1 we can then obtain that the total production quantity of day $t_{|\sigma|}$ equals $\sum_{i \in A} q_i - (t_{|\sigma|} - 1) \cdot c$, which is less than or equal to $t_{|\sigma|} \cdot c - (t_{|\sigma|} - 1) \cdot c = c$. Hence, (iii) of Lemma 2.1 is also proved. □

Based on Lemma 2.1, we can show in Lemma 2.2 below that solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ ensures satisfying all the constraints of model ILP except constraint (2.7).

Lemma 2.2. *For any order sequence σ, solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ satisfies constraints (2.4), (2.5), (2.6), (2.8), (2.9), and (2.10) of model ILP.*

Proof. To simplify the notation in this proof, we use $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to denote $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ and use A to denote $A(\sigma)$. By Lemma 2.1, the total production quantity of each day does not exceed c, and thus constraint (2.4) of model ILP is satisfied. From Steps 1.2–1.4 of the above construction of $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$, we can see that for each $j \in \{1, 2, \cdots, |\sigma|\}$, $\sum_{t \in T} x_{\sigma_j,t} = x_{\sigma_j,t_{j-1}} + q_{\sigma_j} - x_{\sigma_j,t_{j-1}} = q_{\sigma_j}$, and from Step 2 of the above construction, we can also see that for each $j \in N \setminus A$, $\lambda_j = 0$ and $\sum_{t \in T} x_{jt} = 0$. These imply that constraint (2.5) of model ILP is satisfied.

From Step 1.5 of the construction, we know that for each $j \in A$, $z_{\sigma_j,t} = 1$ if and only if $t = t_j$. Since the daily production capacity is limited by c, which implies that $t_j = \lceil \bar{Q}_j / c \rceil \leq$

$\lceil \sum_{i \in A} q_i/c \rceil \le m$ for $j \in A$, we have that $\sum_{t \in T} z_{\sigma_j,t} = 1$ for each $j \in A$. From Step 2 of the construction, we know that $\sum_{t \in T} z_{jt} = 0$ for $j \in N \setminus A$. Thus, we obtain that constraint (2.6) of model ILP is satisfied. Moreover, from Step 1.3 and Step 1.4 of the construction, we know that in $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$, products for each accepted order are shipped out on the same day as their production is completed. Thus, constraint (2.8) of model ILP is satisfied.

From the construction, we can also see that x_{it} for $i \in N$ and $t \in T$ are all integers. For each $j \in \{1, 2, \cdots, |\sigma|\}$, since $(t_{j-1} \cdot c) - \bar{Q}_{j-1} = \lceil \bar{Q}_{j-1}/c \rceil \cdot c - \bar{Q}_{j-1} \ge 0$, by Step 1.3 of the construction we have that $x_{\sigma_j, t_{j-1}} \ge 0$. By Step 1.4 of the construction, we have that $x_{\sigma_j, t_j} \ge 0$. Since we have shown earlier that $t_{j-1} \le t_j \le t_{j-1} + 1$, by Step 1.2 of the construction we also have that $x_{\sigma_j, t} = 0$ for all $t \in \{1, \cdots, m\} \setminus \{t_{j-1}, t_j\}$. From Step 2, we have that $x_{jt} = 0$ for each $j \in N \setminus A$ and $t \in T$. Thus, we obtain that $x_{it} \in \mathbb{Z}_+$ for $i \in N$ and $t \in T$. Moreover, from Step 1.5 and Step 2 of the construction, we have that $z_{\sigma_i, t} \in \{0, 1\}$ for $i \in \{1, 2, \cdots, |\sigma|\}$ and $t \in T$, and $z_{it} = 0$ for $i \in N \setminus \{\sigma_1, \ldots, \sigma_{|\sigma|}\}$ and $t \in T$, implying that constraint (2.10) of model ILP is satisfied. From Step 1.1 and Step 2, we have that $\lambda_i \in \{0, 1\}$ for $i \in N$, implying that constraint (2.9) of model ILP is also satisfied. This completes the proof of Lemma 2.2. $\square$

Although $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ may not always be a feasible solution to model ILP, we can show that there always exists an order sequence σ such that $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ forms an optimal solution to model ILP. To show this, we observe that for any order acceptance plan $\boldsymbol{\lambda}$ and any shipping plan $\mathbf{z}$, one can obtain a sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_{|\sigma|})$ of orders accepted in $\boldsymbol{\lambda}$ in a non-decreasing order of their shipped-out days under $\mathbf{z}$, breaking ties arbitrarily. We refer to such an order sequence as an accepted order sequence with respect to $\boldsymbol{\lambda}$ and $\mathbf{z}$, which may not be unique, since products for different orders may be shipped out on the same day. We can now establish Theorem 2.1 below.

Theorem 2.1. *Consider any optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ to model ILP. Consider any accepted order sequence σ^* with respect to $\boldsymbol{\lambda}^*$ and $\mathbf{z}^*$. Then, $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ also forms an optimal solution to model ILP.*

Proof. To simplify the notation in this proof, we use $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to denote $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$. By Lemma 2.2, solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ satisfies constraints (2.4), (2.5), (2.6) (2.8), (2.9), and (2.10) of model ILP. Consider the optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ to model ILP, in which we know that products for order σ_j^* are not shipped out before any products for orders $\sigma_{j'}^*$ with $j' \in \{1, 2, \cdots, j-1\}$. Thus, in $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$, products for order σ_j^* are not shipped out before the products for the first j orders of σ^* have all been produced. Thus, products for order σ_j^* must be shipped out on or after day t_j, implying that $t_j \leq d_{\sigma_j^*}$. From Step 1.4 and Step 1.5 of the construction, we know that in $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$, products for order σ_j^*, $j \in \{1, \ldots, \sigma^*_{|\sigma^*|}\}$, are shipped out on the same day as their production is completed. Thus, constraint (2.7) of model ILP is satisfied.

Hence, $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ is a feasible solution to model ILP. As shown above, for each $j \in \{1, 2, \cdots, |\sigma|\}$, products for order σ_j^* are shipped out on or after day t_j in the optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$, whereas they are shipped out on day t_j in the constructed solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$. Thus, since the shipping cost function $G(d_i - t, q_i)$ is non-decreasing in t, the total shipping cost of $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ cannot exceed that of $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$. From Step 1.1 and Step 2, it can be seen that the order acceptance plans of the optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ and the constructed solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ are the same, i.e., $\boldsymbol{\lambda}(\sigma^*) = \boldsymbol{\lambda}^*$, which means the rejection costs of the two solutions are the same. Hence, $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ is also an optimal solution to model ILP. Theorem 2.1 is proved. □

Based on Theorem 2.1, we can further establish Theorem 2.2 below, which indicates that there always exists an order sequence σ such that not only $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ is an optimal solution to model ILP, but also that orders of the same quantity in σ are sorted in a non-decreasing order of their committed delivery due dates, breaking ties by preferring orders with smaller indices.

Theorem 2.2. *There exists an order sequence σ such that (i) $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ is an optimal solution to model ILP, and that (ii) $d_{\sigma_j} < d_{\sigma_h}$ or ($d_{\sigma_j} = d_{\sigma_h}$ and $\sigma_j < \sigma_h$), for each j and h with $1 \leq j < h \leq |\sigma|$ and $q_{\sigma_j} = q_{\sigma_h}$.*

Proof. By Theorem 2.1, there exists an order sequence σ^* such that $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ is an optimal solution to model ILP, which satisfies condition (i) specified in Theorem 2.2. Let $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ indicate $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$. If σ^* does not satisfy condition (ii) specified in Theorem 2.2, then there exist j and h in $\{1, 2, \cdots, |\sigma^*|\}$ with $j < h$ and $q_{\sigma_j^*} = q_{\sigma_h^*} = q$ for some q such that $d_{\sigma_j^*} > d_{\sigma_h^*}$ or ($d_{\sigma_j^*} = d_{\sigma_h^*}$ and $\sigma_j^* > \sigma_h^*$). In this situation, we can swap positions of σ_j^* and σ_h^* in σ^* to obtain a new order sequence σ, so that condition (ii) specified in Theorem 2.2 is satisfied for j and h. Moreover, we can also swap values of $x^*_{\sigma_j^*,t}$ and $x^*_{\sigma_h^*,t}$ for $t \in T$, and swap values of $z^*_{\sigma_j^*,t}$ and $z^*_{\sigma_h^*,t}$ for $t \in T$, to obtain a new solution $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$, which, as shown below, is also an optimal solution to model ILP.

Let $i = \sigma_j^*$ and $i' = \sigma_h^*$. For each $i'' \in N$, let $\tau_{i''}$ and $\tau^*_{i''}$ indicate the shipped-out days of order i'', under $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ and $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$, respectively. Thus, we have that $\tau_{i'} = \tau_i^*$ and $\tau_i = \tau_{i'}^*$. According to the construction of the optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*) = (\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$, since $j < h$, we know that $\tau_i^* \le \tau_{i'}^*$. These, together with $\tau_i^* \le d_i$, $d_{i'} \le d_i$, and $\tau_{i'}^* \le d_{i'}$, imply that $\tau_i = \tau_{i'}^* \le d_{i'} \le d_i$, and that $\tau_{i'} = \tau_i^* \le \tau_{i'}^* \le d_{i'}$. Thus, $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ satisfies constraints (2.6) and (2.7) of model ILP. From $q_i = q_{i'}$ we know that $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ satisfies constraints (2.4), (2.5), (2.8), (2.9), and (2.10) of model ILP. Since λ^* satisfies constraint (2.9), we obtain that $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ is a feasible solution to model ILP. By (2.11) below, we can also see that the shipping costs of order i and order i' are the same under $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ and $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$:

$$\begin{aligned}
&G(d_i - \tau_i, q) + G(d_{i'} - \tau_{i'}, q) = q(\alpha - \beta(d_i - \tau_i)) + q(\alpha - \beta(d_{i'} - \tau_{i'}))\\
&= q(\alpha - \beta(d_i - \tau_{i'})) + q(\alpha - \beta(d_{i'} - \tau_i)) = q(\alpha - \beta(d_i - \tau_i^*)) + q(\alpha - \beta(d_{i'} - \tau_{i'}^*))\\
&= G(d_i - \tau_i^*, q) + G(d_{i'} - \tau_{i'}^*, q). \qquad (2.11)
\end{aligned}$$

Thus, owing to the same acceptance plans of the two solutions, the total shipping and rejection cost of $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ equals that of the optimal solution $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$. Therefore, $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ is also an optimal solution to model ILP. Noting that the order sequence σ can be obtained from σ^* by only swapping the positions of i and i', we can see that $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ is equal to

$(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$.

Hence, by replacing σ^* with σ and repeating the process above iteratively, we can obtain an order sequence σ that satisfies both conditions (i) and (ii) specified in Theorem 2.2. This completes the proof of Theorem 2.2. □

It can be seen that, except for the argument of (2.11), our proofs of Lemma 2.1, Lemma 2.2, Theorem 2.1, and Theorem 2.2 above do not rely on the linearity of the cost function $G(s, y)$ in shipping time s. Thus, Lemma 2.1, Lemma 2.2, and Theorem 2.1 are still valid for a more general problem where $G(s, y)$ is linearly non-decreasing in y and convexly non-increasing in s. To see that Theorem 2.2 is also valid for this more general problem, we only need to prove that (2.11) is still valid. Specifically, consider the two orders i and i' and their shipped-out days τ_i^*, $\tau_{i'}^*$, τ_i, and $\tau_{i'}$ under solutions $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ and $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$, which are defined in the proof of Theorem 2.2, and are shown to satisfy that $d_i \geq d_{i'}$ and $\tau_i^* \leq \tau_{i'}^*$. Since $G(s, y)$ is convex in s, we have that

$$\frac{d_i - d_{i'}}{d_i - d_{i'} + \tau_{i'}^* - \tau_i^*} G(d_{i'} - \tau_{i'}^*, q) + \frac{\tau_{i'}^* - \tau_i^*}{d_i - d_{i'} + \tau_{i'}^* - \tau_i^*} G(d_i - \tau_i^*, q) \geq G(d_{i'} - \tau_i^*, q) \text{ and}$$
$$\frac{\tau_{i'}^* - \tau_i^*}{d_i - d_{i'} + \tau_{i'}^* - \tau_i^*} G(d_{i'} - \tau_{i'}^*, q) + \frac{d_i - d_{i'}}{d_i - d_{i'} + \tau_{i'}^* - \tau_i^*} G(d_i - \tau_i^*, q) \geq G(d_i - \tau_{i'}^*, q)$$

implying that $G(d_{i'} - \tau_{i'}, q) + G(d_i - \tau_i, q) = G(d_{i'} - \tau_i^*, q) + G(d_i - \tau_{i'}^*, q) \leq G(d_{i'} - \tau_{i'}^*, q) + G(d_i - \tau_i^*, q)$. Since $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$ is an optimal solution, the total shipping cost of the two orders i and i' under $(\boldsymbol{\lambda}^*, \mathbf{x}, \mathbf{z})$ cannot be less than that under $(\boldsymbol{\lambda}^*, \mathbf{x}^*, \mathbf{z}^*)$, implying that $G(d_{i'} - \tau_{i'}, q) + G(d_i - \tau_i, q) \geq G(d_{i'} - \tau_{i'}^*, q) + G(d_i - \tau_i^*, q)$. Thus, (2.11) is still valid. Hence, Theorem 2.2 is still valid.

2.4 Two Exact Algorithms

In this section, we present two exact algorithms that solve problem IPTSDA to optimality. They run in polynomial or pseudo-polynomial times for the two practical cases mentioned earlier in Section 2.1, respectively, where the total number of possible order quantities is bounded by a constant, and where the length of the planning horizon is bounded by a constant.

2.4.1 Exact Algorithm 1

Let $E = \{q_i \mid i \in N\}$ denote the set of all possible order quantities. We denote elements in E by $e_1, e_2, \ldots, e_\eta$, where $\eta = |E|$ indicates the total number of possible order quantities. For example, consider a manufacturer who restricts customers to order only ten or twenty units of the products, which implies that $E = \{10, 20\}$ and $\eta = 2$. For each $k \in \{1, 2, \ldots, \eta\}$, let $N_k = \{i \mid q_i = e_k, i \in N\}$ be the set of orders with order quantities equal to e_k, and let $n_k = |N_k|$. We have $N_1 \cup N_2 \cup \ldots \cup N_\eta = N$ and $n_1 + n_2 + \ldots + n_\eta = n$, and thus $N_1, N_2, \ldots, N_\eta$ form a partition of N.

Our development of the first exact algorithm for problem IPTSDA relies on the properties described in Theorem 2.2 for the optimal solutions to model ILP. The algorithm is based on a dynamic program described as follows. For each $k \in \{1, 2, \ldots, \eta\}$, we denote the indices of the orders in N_k by $i(k,1), i(k,2), \ldots, i(k,n_k)$, and without loss of generality, we assume that orders in each N_k are indexed in a non-decreasing order so that $d_{i(k,1)} \leq d_{i(k,2)} \leq \ldots \leq d_{i(k,n_k)}$. For each $(p_1, p_2, \cdots, p_\eta)$ with $p_k \in \{0, 1, \cdots, n_k\}$ for $k \in \{1, 2, \cdots, \eta\}$, we define

$$N(p_1, p_2, \ldots, p_\eta) = \{i(k,r) \mid 1 \leq r \leq p_k, 1 \leq k \leq \eta\}$$

as the set of the first p_k orders $i(k,1), i(k,2), \ldots, i(k,p_k)$ of each N_k for $k \in \{1, 2, \cdots, \eta\}$. We define a value function $F((p_1, \hat{p}_1), \ldots, (p_\eta, \hat{p}_\eta))$ as the minimum total shipping and rejection cost of a subproblem of problem IPTSDA defined for only orders in $N(p_1, \ldots, p_\eta)$, where the number of rejected orders in $\{i(k,1), i(k,2), \ldots, i(k,p_k)\}$ equals $\hat{p}_k$ for $k \in \{1, \ldots, \eta\}$. Let $F((p_1, \hat{p}_1), \ldots, (p_\eta, \hat{p}_\eta))$ to be $+\infty$ if the subproblem has no feasible solution. Accordingly, the minimum total shipping and rejection cost of problem IPTSDA, which is defined for all orders in N, can be represented by $\min\big\{F((n_1,\hat{p}_1),\ldots,(n_\eta,\hat{p}_\eta)) \mid 0 \le \hat{p}_k \le n_k \text{ for } k \in \{1,2,\ldots,\eta\}\big\}$.

The value function $F((p_1,\hat{p}_1),\ldots,(p_\eta,\hat{p}_\eta))$ can be computed recursively as follows. First, since the subproblem of $F((0,0),(0,0),\ldots,(0,0))$ is defined for an empty order set, its minimum total shipping and rejection cost is zero. Thus, we obtain the boundary condition of the dynamic program that $F((0,0),(0,0),\ldots,(0,0)) = 0$.

Next, for each $(p_1,p_2,\cdots,p_\eta)$ with $p_k \in \{0,1,\cdots,n_k\}$ for $k \in \{1,\ldots,\eta\}$ and with $\sum_{r=1}^{\eta} p_r \ge 1$, and for each $(\hat{p}_1,\hat{p}_2,\ldots,\hat{p}_\eta)$ with $\hat{p}_k \in \{0,1,\ldots,p_k\}$ for $k \in \{1,\ldots,\eta\}$, we can apply Theorem 2.2 to the subproblem of $F((p_1,\hat{p}_1),\ldots,(p_\eta,\hat{p}_\eta))$. This implies that there exists an order sequence σ of some orders in $N(p_1,p_2,\ldots,p_\eta)$ such that the solution $(\boldsymbol{\lambda}(\sigma),\mathbf{x}(\sigma),\mathbf{z}(\sigma))$ forms an optimal solution to the subproblem, and that for each $k \in \{1,2,\cdots,\eta\}$, orders in $N_k \cap \{\sigma_1,\sigma_2,\cdots,\sigma_{|\sigma|}\}$, which are of the same order quantity, are sorted in σ in a non-decreasing order of their committed delivery due dates, breaking ties by preferring orders with smaller indices. Consider the following two cases for such an optimal solution $(\boldsymbol{\lambda}(\sigma),\mathbf{x}(\sigma),\mathbf{z}(\sigma))$ to the subproblem of $F((p_1,\hat{p}_1),\ldots,(p_\eta,\hat{p}_\eta))$:

Case 1. (See Figure 2.2 for an illustrative example) All the orders in $\{i(1,p_1),i(2,p_2),\ldots,i(\eta,p_\eta)\}$ are accepted in $(\boldsymbol{\lambda}(\sigma),\mathbf{x}(\sigma),\mathbf{z}(\sigma))$, and thus, they are all contained in σ. Hence, the last order in σ must be order $i(k^*,p_{k^*})$ for some $k^* \in \{1,2,\cdots,\eta\}$. Let

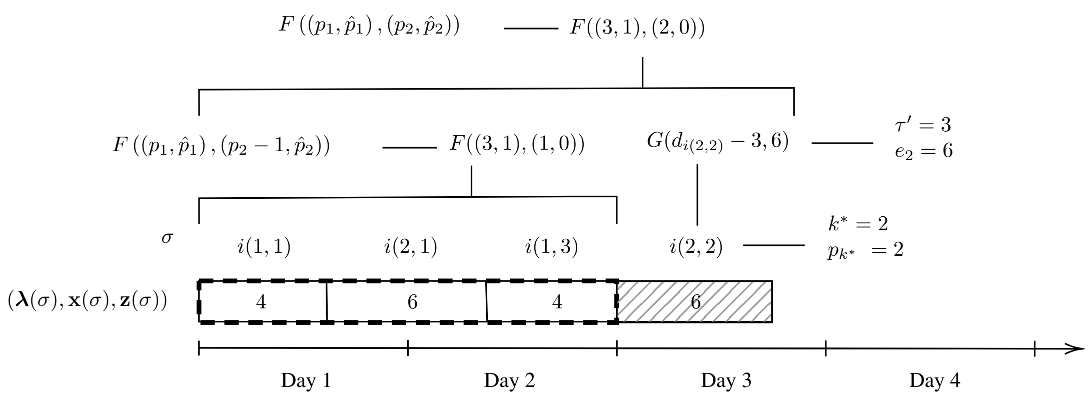
$$\tau' = \left\lceil \sum_{r=1}^{\eta}(p_r-\hat{p}_r)e_r/c \right\rceil,$$

which indicates the minimum number of days required to produce products for accepted orders in $N(p_1,p_2,\ldots,p_\eta)$. According to the construction of $(\boldsymbol{\lambda}(\sigma),\mathbf{x}(\sigma),\mathbf{z}(\sigma))$, the last order $i(k^*,p_{k^*})$ in σ both has its product production completed and has its products shipped out on day τ'. Therefore, k^* satisfies that $p_{k^*} \ge 1$ and $d_{i(k^*,p_{k^*})} \ge \tau'$. If such k^* does not exist, then Case 1 is not possible. Otherwise, the shipping cost for order $i(k^*,p_{k^*})$ equals $G(d_{i(k^*,p_{k^*})}-\tau',e_{k^*})$, and for other orders in $N(p_1,\ldots,p_{k^*-1},p_{k^*}-$

$1, p_{k^*+1}, \ldots, p_\eta)$ (which equals $N(p_1, \ldots, p_{k^*-1}, p_{k^*}, p_{k^*+1}, \ \ldots, p_\eta) \setminus \{i(k^*, p_{k^*})\}$), we know that their acceptance, production and shipping plans in $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ must form an optimal solution to the subproblem of $F((p_1, \hat{p}_1), \ldots, (p_{k^*-1}, \hat{p}_{k^*-1}), \ (p_{k^*} - 1, \hat{p}_{k^*}), (p_{k^*+1}, \hat{p}_{k^*+1}), \ldots, (p_\eta, \hat{p}_\eta))$;

Case 2. There exists an order in $\{i(1, p_1), i(2, p_2), \ldots, i(\eta, p_\eta)\}$ that is rejected in $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$, and we denote such rejected order by $i(k^*, p_{k^*})$ for some $k^* \in \{1, 2, \cdots, \eta\}$, and accordingly, the rejection cost incurred equals $r_{i(k^*, p_{k^*})}$. For other orders in $N(p_1, \ldots, p_{k^*-1}, p_{k^*} - 1, p_{k^*+1}, \ldots, p_\eta)$ (which equals $N(p_1, \ldots, p_{k^*-1}, p_{k^*}, p_{k^*+1}, \ldots, p_\eta) \setminus \{i(k^*, p_{k^*})\}$), we know that their acceptance, production and shipping plans in $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ must form an optimal solution to the subproblem of $F((p_1, \hat{p}_1), \ldots, (p_{k^*-1}, \hat{p}_{k^*-1}), (p_{k^*} - 1, \hat{p}_{k^*} - 1), (p_{k^*+1}, \hat{p}_{k^*+1}), \ldots, (p_\eta, \hat{p}_\eta))$.

Figure 2.2: An example for Case 1 of the illustration of (2.12) with $\eta = 2$ types of order quantities ($e_1 = 4$ and $e_2 = 6$) and with $c = 7$.



For the subproblem of $F((3, 1), (2, 0))$ for orders $\{i(1, 1), i(1, 2), i(1, 3), i(2, 1), i(2, 2)\}$, if the last order of the optimal order sequence σ is $i(2, 2)$, and is accepted in the optimal solution, then $F((3, 1), (2, 0))$ equals $F((3, 1), (1, 0))$ plus $G(d_{i(2,2)} - 3, 6)$ (since $\tau' = 3$ and $e_2 = 6$). Order $i(1, 2)$ is rejected in the optimal solution.

Accordingly, we can enumerate k^* for the two cases above to compute $F((p_1, \hat{p}_1), \ldots, (p_\eta, \hat{p}_\eta))$ by the following recursive equation:

$$F((p_1,\hat{p}_1),\ldots,(p_\eta,\hat{p}_\eta)) = \min \left\{ \begin{array}{l} \min \left\{ \begin{array}{l} F((p_1,\hat{p}_1),\ldots,(p_{k^*-1},\hat{p}_{k^*-1}),(p_{k^*}-1,\hat{p}_{k^*}),(p_{k^*+1},\hat{p}_{k^*+1}),\ldots,(p_\eta,\hat{p}_\eta))+ \\ G(d_{i(k^*,p_{k^*})}-\tau', e_{k^*}) \mid \forall k^* \in \{1,\ldots,\eta\} \text{ with } p_{k^*} \geq 1 \text{ and } d_{i(k^*,p_{k^*})} \geq \tau' \end{array} \right\}, \\ \min \left\{ \begin{array}{l} F((p_1,\hat{p}_1),\ldots,(p_{k^*-1},\hat{p}_{k^*-1}),(p_{k^*}-1,\hat{p}_{k^*}-1),(p_{k^*+1},\hat{p}_{k^*+1}),\ldots,(p_\eta,\hat{p}_\eta))+ \\ r_{i(k^*,p_{k^*})} \mid \forall k^* \in \{1,\ldots,\eta\} \text{ with } \hat{p}_{k^*} \geq 1. \end{array} \right\}. \end{array} \right\}, \quad (2.12)$$

where we assume that taking minimum value over an empty set equals $+\infty$.

Finally, we can enumerate all $(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_\eta)$ for $\hat{p}_k \in \{0, 1, \ldots, n_k\}$ and $k \in \{1, 2, \ldots, \eta\}$ to minimize $F((n_1, \hat{p}_1), \ldots, (n_\eta, \hat{p}_\eta))$, and then return the minimum value, which, as explained earlier, is the minimum total shipping and rejection cost for problem IPTSDA.

We summarize the exact algorithm in Algorithm 2.1 below, and its correctness and time complexity are presented in Theorem 2.3.

Algorithm 2.1 (for problem IPTSDA)

1: $F((0,0),\ldots,(0,0)) \leftarrow 0$ and for $(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_\eta)$ with $\hat{p}_k \in \{0, 1 \ldots, n_k\}$ for $k \in \{1, \ldots, \eta\}$ and with $\sum_{r=1}^{\eta} \hat{p}_r \geq 1$, $F((0,\hat{p}_1),\ldots,(0,\hat{p}_\eta)) \leftarrow \infty$

2: **for all** $(p_1, p_2, \ldots, p_\eta)$ with $p_k \in \{0, 1 \ldots, n_k\}$ for $k \in \{1, \ldots, \eta\}$ and with $\sum_{r=1}^{\eta} p_r \geq 1$ **do**

3:     **for all** $(\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_\eta)$ with $\hat{p}_k \in \{0, 1, \ldots, p_k\}$ for $k \in \{1, \ldots, \eta\}$ **do**

4:         Compute $F((p_1,\hat{p}_1),\ldots,(p_\eta,\hat{p}_\eta))$ by the recursive equation in (2.12)

5:     **end for**

6: **end for**

7: **return** the minimum value of $F((n_1,\hat{p}_1),\ldots,(n_\eta,\hat{p}_\eta))$ over all $\hat{p}_k \in \{0, 1, \ldots, n_k\}$ for $k \in \{1, \ldots, \eta\}$.

Theorem 2.3. *Algorithm 2.1 solves problem IPTSDA to optimality in $O(\eta \cdot (1 + n/\eta)^{2\eta})$ time.*

Proof. As we have shown above, the value function $F((p_1, \hat{p}_1), \ldots, (p_\eta, \hat{p}_\eta))$ can be computed recursively, and the value of $\min\{F((n_1, \hat{p}_1), \ldots, (n_\eta, \hat{p}_\eta)) | \hat{p}_k = 0, 1, \ldots, n_k \text{ for } k \in \{1, 2, \ldots, \eta\}\}$ returned by Algorithm 2.1 equals the minimum total shipping and rejection cost for problem IPTSDA. Thus, Algorithm 2.1 solves problem IPTSDA to optimality. Moreover, the recursive equation (2.12) is computed in Algorithm 2.1 for at most $((1 + n_1)(1 + n_2) \cdots (1 + n_\eta))^2$ times. Noting that $n_1 + n_2 + \ldots + n_\eta = n$, we have $((1 + n_1)(1 + n_2) \cdots (1 + n_\eta))^2 \le [\sum_{k=1}^{\eta}(1 + n_k)/\eta]^{2\eta} = (1 + n/\eta)^{2\eta}$. Since it takes $O(\eta)$ time to compute the recursive equation (2.12), we obtain that the total time complexity of Algorithm 2.1 is $O(\eta \cdot (1 + n/\eta)^{2\eta})$. □

When the number of possible order quantities η is a fixed constant, Theorem 2.3 implies that Algorithm 2.1 solves problem IPTSDA to optimality in $O(n^{2\eta})$, which is polynomial time if n is polynomially bounded by the input size. This is the case in the representation of the problem instance introduced in Section 2.2, in which the input size is linear in n. However, for problem IPTSDA, there is an alternative representation of the problem instance, in which input parameters q_i for $i \in N$ and $d_i \in N$ are replaced by $n_{k,t}$ for $t \in T$ and $k \in \{1, 2, \cdots, \eta\}$, where each $n_{k,t}$ indicates the number of orders with quantities equal to e_k and committed delivery due date equals to t. In such a representation, when η is a fixed constant, the input size is in $O(m)$, and since n can be exponential in m, the running time $O(n^{2\eta})$ is pseudo-polynomial time.

Theorem 2.3 can also be applied to solve a variant of problem IPTSDA where orders are allowed to be split so that parts of an order can be accepted and shipped once they are produced. We refer to this variant as problem IPTSDA-S. It can be seen that problem IPTSDA-S is a relaxation of problem IPTSDA, as any feasible solution to problem IPTSDA is also a feasible solution to problem IPTADA-S. Moreover, for any instance of problem IPTSDA-S, one can transform it equivalently to an instance of problem IPTSDA with equal order quantities, by splitting each order i of quantity q_i to q_i orders of unit quantity. Thus, the resulting instance of problem IPTSDA has $\sum_{i \in N} q_i$ orders and has $\eta = 1$. This, together with Theorem 2.3, implies that problem IPTSDA-S can be solved to optimality in a pseudo-polynomial running time of $O((\sum_{i \in N} q_i)^2)$. Thus, the following corollary is established.

Corollary 2.1. *Problem IPTSDA-S is a relaxation of problem IPTSDA and can be solved to optimality in a pseudo-polynomial running time of* $O((\sum_{i \in N} q_i)^2)$.

Moreover, consider the more general problem where $G(s, y)$ is linearly non-decreasing in y and convexly non-increasing in s. For this problem, since we have shown that Theorem 2.2 is still valid, our description and analysis of Algorithm 2.1 are still valid. Thus, Theorem 2.3 and Corollary 2.1 are still valid.

2.4.2 Exact Algorithm 2

Our second exact algorithm for problem IPTSDA, to be presented below, runs in pseudo-polynomial time when the length of the planning horizon m is a fixed constant. To present this algorithm, we will first prove that given any order acceptance plan $\boldsymbol{\lambda}$ and shipping plan $\mathbf{z}$ that satisfies certain conditions, there always exists a production plan $\mathbf{x}$ such that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ is a feasible solution to model ILP. This implies that to solve model ILP, we need to only optimize the order acceptance plan $\boldsymbol{\lambda}$ and the shipping plan $\mathbf{z}$, for which we can develop a dynamic programming algorithm as follows.

First, consider any feasible solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to model ILP. For each $t \in T$, let Q_t denote the total quantity of products shipped out on day t, i.e., $Q_t = \sum_{i \in N} q_i z_{it}$. From constraint (2.8) of model ILP we know that $\sum_{t'=1}^{t} Q_{t'}$, the total quantity of the products shipped on or before day t, should not exceed the total quantity of the products produced on or before day t, which, by constraint (2.4), should not exceed the total production capacity tc of the first t days. Thus, $(Q_1, Q_2, \cdots, Q_m)$ satisfies condition (2.13) below:

$$\sum_{t'=1}^{t} Q_{t'} \leq tc, \text{ for each } t \in T. \tag{2.13}$$

We can now establish Proposition 2.1 for any $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied.

Proposition 2.1. *Consider any $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ that satisfies condition (2.13).For any order acceptance plan $\boldsymbol{\lambda}$ with $\lambda_i \in \{0,1\}$ for $i \in N$, and any shipping plan $\mathbf{z}$ with $\sum_{i\in N} q_i z_{it} = Q_t$ for $t \in T$, $\sum_{t=1}^{d_i} z_{it} = \lambda_i$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, and $z_{it} \in \{0,1\}$ for $i \in N$ and $t \in T$, there exists a production plan $\mathbf{x}$ such that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ is a feasible solution to model ILP.*

Proof. Consider any order acceptance plan $\boldsymbol{\lambda}$ and any shipping plan $\mathbf{z}$ satisfying the conditions in Proposition 2.1. As illustrated in Section 2.3, we can obtain an accepted order sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_{|\sigma|})$ with respect to $\boldsymbol{\lambda}$ and $\mathbf{z}$ by sorting the orders accepted in $\boldsymbol{\lambda}$ in a non-decreasing order of their shipped-out days under $\mathbf{z}$, breaking ties arbitrarily. From σ, we can follow the procedure described in Section 2.3 to construct a solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ for model ILP. Note that $\sum_{t=1}^{m} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in \{\sigma_1, \ldots, \sigma_{|\sigma|}\}$. We use $\tau_j \in \{1, 2, \cdots, d_i\}$ to indicate the shipped-out day of each order σ_j of σ under the shipping plan $\mathbf{z}$. We know that orders before σ_j in sequence σ must all be shipped out on or before day τ_j under $\mathbf{z}$. This, together with (2.13), implies that $\sum_{j'=1}^{j} q_{\sigma_{j'}} \le \sum_{t'=1}^{\tau_j} Q_{t'} \le \tau_j c$. Thus, by (2.14) below, the earliest possible production completion day t_j for the first j orders of σ cannot exceed τ_j.

$$t_j = \left\lceil \sum_{j'=1}^{j} q_{\sigma_{j'}}/c \right\rceil \le \tau_j. \tag{2.14}$$

This, together with the fact that under solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$, both the production completion day and the shipped-out day of each order σ_j are equal to t_j, implies that $(\boldsymbol{\lambda}, \mathbf{x}(\sigma), \mathbf{z})$ must be a feasible solution to model ILP. Proposition 2.1 is proved. □

Next, for each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied, we define $F(Q_1, Q_2, \ldots, Q_m)$ as the minimum total shipping and rejection cost among all the order acceptance plans $\boldsymbol{\lambda}$ and the shipping plans $\mathbf{z}$ that satisfy $\sum_{i\in N} q_i z_{it} = Q_t$ for $t \in T$ and satisfy $\sum_{t=1}^{m} z_{it} = \lambda_i$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$. Proposition 2.1 implies that to solve model ILP, it is equivalent to minimizing $F(Q_1, Q_2, \ldots, Q_m)$ over all such $(Q_1, Q_2, \cdots, Q_m)$, which can be achieved by the following dynamic program.

For each $i \in \{0, 1, \cdots, n\}$, let $N(i) = \{i' \in N \mid i' \leq i\}$ denote the set of orders $i' \in N$ with $i' \leq i$. For each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied, we define a value function $F(i; Q_1, Q_2, \ldots, Q_m)$ as the minimum total shipping and rejection cost of a subproblem that aims to find an order acceptance plan $\boldsymbol{\lambda}$ and a shipping plan $\mathbf{z}$ only for orders in $N(i)$, such that $\sum_{i' \in N(i)} q_{i'} z_{i',t} = Q_t$ for $t \in T$ and that $\sum_{t=1}^{m} z_{i',t} = \lambda_{i'}$ and $\sum_{t=d_{i'}+1}^{m} z_{i',t} = 0$ for $i' \in N(i)$. If the corresponding subproblem has no such order acceptance plan and shipping plan, the value of $F(i; Q_1, Q_2, \ldots, Q_m)$ is $+\infty$. Accordingly, we have $F(Q_1, Q_2, \ldots, Q_m) = F(n; Q_1, Q_2, \cdots, Q_m)$.

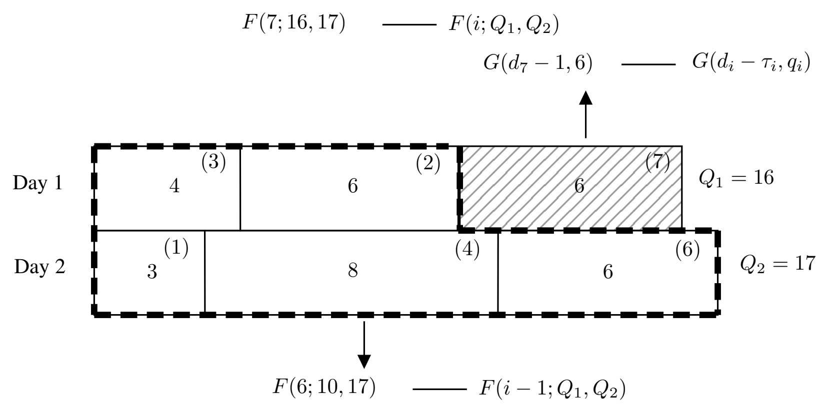
The value function $F(i; Q_1, Q_2, \cdots, Q_m)$ can be computed recursively as follows. Since the subproblem of $F(0; 0, \ldots, 0)$ is defined for an empty order set, we obtain the boundary condition of the dynamic program that $F(0; 0, \ldots, 0) = 0$, and that $F(0; Q_1, Q_2, \ldots, Q_m) = +\infty$ for each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied and with $\sum_{t=1}^{m} Q_t > 0$.

For each $i = 1, 2, \ldots, n$, and for each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied, consider the following two possible cases of an order acceptance plan $\boldsymbol{\lambda}$ and a shipping plan $\mathbf{z}$ that form an optimal solution to the subproblem of $F(i; Q_1, Q_2, \cdots, Q_m)$:

Case 1. (See Figure 2.3 for an illustrative example) Order i is accepted under the order acceptance plan $\boldsymbol{\lambda}$. Let $\tau_i \in \{1, 2, \cdots, m\}$ indicate the shipped-out day of order i under the shipping plan $\mathbf{z}$. We know that τ_i satisfies that $\tau_i \leq d_i$ and $q_i \leq Q_{\tau_i}$, and the shipping cost for order i equals $G(d_i - \tau_i, q_i)$. For other orders, which are in $N(i-1) = N(i) \setminus \{i\}$, we know that their acceptance plan and shipping plan under $\boldsymbol{\lambda}$ and $\mathbf{z}$ must form an optimal order acceptance and shipping plan for the subproblem of $F(i-1; Q_1, \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i+1}, \ldots, Q_m)$. Here, by definition, the subproblem of $F(i-1; Q_1, \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i+1}, \ldots, Q_m)$ is defined for orders in $N(i-1)$, and for the same total shipped-out quantity of each day except day τ_i, for which the quantity Q_{τ_i} is reduced by q_i.

Case 2. Order i is rejected under the order acceptance plan $\boldsymbol{\lambda}$, incurring a rejection cost r_i.

Figure 2.3: An example for Case 1 of the illustration of (2.15) with $m = 2$ days.



Inside rectangles, the numbers in brackets are orders' indices, and the numbers without brackets are order quantities: For the subproblem of $F(7;16,17)$, for orders in $\{1,2,\cdots,7\}$, if order 7 is accepted and shipped out on day 1 in an optimal solution (i.e., $\tau_7 = 1$), then $F(7;16,17)$ equals $F(6;10,17)$ plus $G(d_7-1,6)$ (since $\tau_7 = 1$ and $q_7 = 6$). Order 5 is rejected in the optimal solution.

For other orders, which are in $N(i-1) = N(i) \setminus \{i\}$, we know that their acceptance plan and shipping plan under $\boldsymbol{\lambda}$ and $\mathbf{z}$ must form an optimal order acceptance and shipping plan for the subproblem of $F(i-1;Q_1,\ldots,Q_m)$.

Accordingly, we can obtain the following recursive equation to compute $F(i;Q_1,Q_2,\cdots,Q_m)$:

$$F(i;Q_1,\ldots,Q_m) = \min\left\{\begin{array}{l} \min\Big\{F(i-1;Q_1,\ldots,Q_{\tau_i-1},Q_{\tau_i}-q_i,Q_{\tau_i+1},\ldots,Q_m)+G(d_i-\tau_i,q_i) \mid \\ \qquad\qquad \forall \tau_i \in \{1,2,\ldots m\} \text{ with } \tau_i \le d_i \text{ and } q_i \le Q_{\tau_i}\Big\}, \\ F(i-1;Q_1,\ldots,Q_m)+r_i. \end{array}\right\}, \tag{2.15}$$

where we assume that the minimum over an empty set equals $+\infty$.

Finally, noting that $F(Q_1,Q_2,\ldots,Q_m) = F(n;Q_1,Q_2,\cdots,Q_m)$, we can enumerate all $(Q_1,Q_2,\cdots,Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied to minimize $F(n;Q_1,Q_2,\cdots,Q_m)$, and then return the minimum value, which, as explained earlier, is the minimum total shipping and rejection cost for problem IPTSDA.

We summarize this exact algorithm in Algorithm 2.2, and its correctness and time complexity are presented in Theorem 2.4.

Algorithm 2.2 (for problem IPTSDA)

$F(0; 0, 0, \ldots, 0) \leftarrow 0$, and $F(0; Q_1, Q_2, \ldots, Q_m) \leftarrow +\infty$ for all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied and with $\sum_{t=1}^{m} Q_t > 0$

for all $i = 1, 2, \cdots, n$ **do**

  for all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}^m$ with (2.13) satisfied **do**

    Compute $F(i; Q_1, Q_2, \ldots, Q_m)$ by the recursive equation in (2.15)

  end for

end for

return the minimum value of $F(n; Q_1, Q_2, \cdots, Q_m)$ over all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied

Theorem 2.4. *Algorithm 2.2 solves problem IPTSDA to optimality in $O(nc^m(m!)m)$ time.*

Proof. As we have shown above, the value function $F(i; Q_1, Q_2, \ldots, Q_m)$ can be computed recursively by (2.15), and the minimum value of $F(n; Q_1, Q_2, \ldots, Q_m)$ over all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (2.13) satisfied equals the minimum total shipping and rejection cost for problem IPTSDA. Thus, Algorithm 2.2 solves problem IPTSDA to optimality. Moreover, since (2.13) implies that $Q_t \leq tc$ for $1 \leq t \leq m$, the recursive equation (2.15) is computed in Algorithm 2.2 for at most $n \cdot (1 \cdot c)(2 \cdot c) \cdots (m \cdot c) = nc^m(m!)$ times. Since it takes $O(m)$ time to compute the recursive equation (2.15), we obtain that the total time complexity of Algorithm 2.2 is $O(nc^m(m!)m)$. $\square$

Theorem 2.4 implies that when the number of days m is a fixed constant, Algorithm 2.2 solves problem IPTSDA to optimality in $O(nc^m)$, which is pseudo-polynomial in the input size. This is true no matter whether the input size is $O(n)$ for the instance representation introduced in Section 2.2, or $O(m)$ for the alternative instance representation described in Section 2.4.1.

Furthermore, our description and analysis of Algorithm 2.2 above do not rely on the linearity of the cost function $G(s, y)$ in shipping time s. Therefore, Algorithm 2.2 can also be used to solve more general problems, such as those where the shipping cost function $G(s, y)$ is linearly non-decreasing in y and convexly non-increasing in s, for which the time complexity result in Theorem 2.4 is also valid, and the running time for the case with a fixed m is still pseudo-polynomial time.

2.5 Approximation Scheme

In this section, we develop an approximation algorithm for problem IPTSDA that guarantees a worst-case performance ratio of $(1 + \epsilon)$ and a pseudo-polynomial running time for any fixed constant $\epsilon > 0$. Our main idea is as follows: We first introduce two parameters, $K \in \{1, \cdots, m\}$ and $Q \in \{0, 1, \ldots, \bar{Q}\}$, where $\bar{Q} = \sum_{i \in N} q_i$. For each pair of K and Q satisfying that

$$\lceil Q/c \rceil \leq K, \tag{2.16}$$

we then define a restricted version of problem IPTSDA, denoted by RP(K, Q), which, as we will show later, can be solved to optimality in pseudo-polynomial time by a dynamic programming approach extended from Algorithm 2.2. Based on the optimal solution to the restricted problem RP(K, Q), we can construct a feasible solution to problem IPTSDA. For a selected value of K, our approximation algorithm solves RP(K, Q) for several different values of Q, so as to obtain a set of feasible solutions to problem IPTSDA, among which the one with the lowest total shipping and rejection cost is then returned as an approximation solution to problem IPTSDA.

Given any constant $\epsilon > 0$, we can prove that by choosing $K = \min\{\lceil 1/\epsilon \rceil, m\}$, which is bounded by the constant $\lceil 1/\epsilon \rceil$, the total shipping and rejection cost of our obtained approximation solution is at most $(1+\epsilon)$ times that of the optimal solution, and our approximation algorithm runs in pseudo-polynomial time. Moreover, when $K = m$, implying that m is bounded by the constant $\lceil 1/\epsilon \rceil$, we can prove that the obtained approximation solution is, in fact, optimal.

In the following, we first formulate and solve the restricted problem RP(K, Q) in Section 2.5.1, and then present and analyze the approximation scheme in Section 2.5.2, followed by a discussion of an extension in Section 2.5.3. Without loss of generality, we assume in this section that orders in N are indexed in a non-decreasing order of their committed delivery due dates, so that

$$d_1 \leq d_2 \leq \cdots \leq d_n. \tag{2.17}$$

2.5.1 Restricted Problem RP(K, Q): Formulation and Solution Algorithm

Given the two parameters $K \in \{1, 2, \ldots, m\}$ and $Q \in \{0, 1, \ldots, \bar{Q}\}$ with (2.16) satisfied, the restricted version of problem IPTSDA, denoted by RP(K, Q), is defined so as to determine both a feasible solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to model ILP and a subset I of orders in N such that the order acceptance plan $\boldsymbol{\lambda}$, the order subset I, and the shipping plan $\mathbf{z}$ satisfy the following additional constraints:

(i) Orders in I are all accepted, i.e., $\lambda_i = 1$ for all $i \in I$. In other words, all the rejected orders are in $N \setminus I$, i.e., $i \in N \setminus I$ for all $i \in N$ with $\lambda_i = 0$.

(ii) The total quantity of the accepted orders in $N \setminus I$ equals Q, i.e., $\sum_{i \in N \setminus I : \lambda_i = 1} q_i = Q$.

(iii) For each accepted order $i \in I$, its products are shipped out on day $\lceil (Q + \sum_{i' \in I : i' \leq i} q_{i'})/c \rceil$, i.e., $z_{it} = 1$ for $t = \lceil (Q + \sum_{i' \in I : i' \leq i} q_{i'})/c \rceil$. This is the earliest possible production completion day of all the accepted orders in $N \setminus I$ (whose total quantity equals Q due to constraint (ii) above), and all the accepted orders in I with indices not greater than i (whose total quantity is $\sum_{i' \in I : i' \leq i} q_{i'}$).

(iv) Products for each accepted order in $N \setminus I$ are shipped out only on or before day K', i.e., $\sum_{t'=1}^{K'} z_{it'} = 1$ for $i \in N \setminus I$ with $\lambda_i = 1$. Here, $K' = \lceil Q/c \rceil$ indicates the earliest possible production completion day of all the accepted orders in $N \setminus I$.

A feasible solution to problem RP(K, Q) can be represented by $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$. Similar to problem IPTSDA, problem RP(K, Q) aims to minimize the total shipping and rejection cost. If problem RP(K, Q) has no feasible solution, its minimal total shipping and rejection cost is $+\infty$.

By Theorem 2.1 we know that there exists an order sequence σ^* such that $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ is an optimal solution to problem IPTSDA. Let $\bar{Q}'$ denote the total quantity of accepted orders under $\boldsymbol{\lambda}(\sigma^*)$, which satisfies $\bar{Q}' \leq \bar{Q}$. Consider the situation where $K = m$, $Q = \bar{Q}'$, and $I = \emptyset$. It can be verified that $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ and $I = \emptyset$ satisfy the additional constraints (i), (ii), (iii), and (iv) above. Thus, $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ and $I = \emptyset$ form a feasible solution to the restricted problem RP(K, Q) with $K = m$ and $Q = \bar{Q}'$, implying that they also form an optimal solution to the restricted problem RP(K, Q) with $K = m$ and $Q = \bar{Q}'$.

Therefore, problem IPTSDA is equivalent to problem RP(K, Q) with $K = m$ and $Q = \bar{Q}'$. We know that problem IPTSDA can be solved by Algorithm 2.2 in pseudo-polynomial time when m is bounded by a fixed constant. This motivates us to extend Algorithm 2.2 to develop an exact algorithm for problem RP(K, Q) for any given K and Q, so that it runs in pseudo-polynomial time when K is bounded by a fixed constant. When m is arbitrarily large, by choosing a proper value of K, and by enumerating the values of Q, we can then utilize such an algorithm of problem RP(K, Q) to obtain a close-to-optimal solution to problem IPTSDA in pseudo-polynomial time.

To present our exact algorithm for problem RP(K, Q) for any given $K \in \{1, 2, \ldots, m\}$ and $Q \in \{0, 1, \ldots, \bar{Q}\}$ with (2.16) satisfied, consider any feasible solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$. Due to the additional constraint (iv) above, the accepted orders in $N \setminus I$ must all be shipped out on or before day $K' = \lceil Q/c \rceil$. From (2.16), we know that

$$K' \leq K. \tag{2.18}$$

By extending the notation in Section 2.4.2, we define $Q_t = \sum_{i \in N \setminus I: \lambda_i = 1} q_i z_{it}$ for each $t \in \{1, 2, \ldots, K'\}$ to indicate the total product quantity of the accepted orders in $N \setminus I$ that are shipped out on day t, and define $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ to indicate the total product quantity of the accepted orders in I. From the definition above, we also have $\sum_{t'=1}^{K'} Q_{t'} = Q$ and $\sum_{i \in I} q_i = Q'$. Similar to (2.13), we can obtain that $(Q_1, \cdots, Q_{K'})$ satisfies the following condition:

$$\sum_{t'=1}^{t} Q_{t'} \leq tc, \text{ for each } t \in \{1, 2, \ldots, K'\}. \tag{2.19}$$

Similar to Proposition 2.1, we can establish Proposition 2.2 below for any $(Q_1, Q_2, \ldots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ that satisfies condition (2.19) and for any $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$.

Proposition 2.2. *Consider any $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ that satisfies condition (2.19) and any $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$. For any order acceptance plan $\boldsymbol{\lambda}$, for any subset $I \subseteq N$ that satisfies $\sum_{i \in I} q_i = Q'$, and for any shipping plan $\mathbf{z}$ that satisfies $\sum_{i \in N \setminus I: \lambda_i = 1} q_i z_{it} = Q_t$ for $t \in \{1, 2, \cdots, K'\}$ and that satisfies $\sum_{t=1}^{m} z_{it} = \lambda_i$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, if $\boldsymbol{\lambda}$, $\mathbf{z}$ and I satisfy the additional constraints (i)–(iv) of problem* RP(K, Q)*, there exists a production plan $\mathbf{x}$ such that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ is a feasible solution to problem* RP(K, Q).

Proof. Consider any $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$, any $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$, any order acceptance plan $\boldsymbol{\lambda}$, subset $I \subseteq N$, and any shipping plan $\mathbf{z}$ that satisfy the conditions mentioned in Proposition 2.2. Similar to the proof of Proposition 2.1, we can obtain an accepted order sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_{|\sigma|})$ with respect to $\boldsymbol{\lambda}$ and $\mathbf{z}$ by sorting accepted orders in $\boldsymbol{\lambda}$ in a non-decreasing order of their shipped-out days under $\mathbf{z}$, breaking ties by preferring accepted orders in $N \setminus I$, and then arbitrarily. Let $A(\sigma)$ indicate the set of orders in σ.

Let j^* indicate the largest index of order σ_j of σ such that $\sigma_j \in A(\sigma) \setminus I$. Thus, we have $A(\sigma) \setminus I = \{\sigma_1, \sigma_2, \cdots, \sigma_{j^*}\}$ and $I = \{\sigma_{j^*+1}, \sigma_{j^*+2}, \cdots, \sigma_{|\sigma|}\}$.

Note that $\sum_{t=1}^{m} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in A(\sigma)$. For each $1 \le j \le |\sigma|$, we use $\tau_j \le d_{\sigma_j}$ to indicate the shipped-out day of order σ_j under the shipping plan $\mathbf{z}$, and we use $t_j = \lceil \sum_{j'=1}^{j} q_{\sigma_{j'}}/c \rceil$ to indicate the earliest possible production completion day of the first j orders of σ. We can prove as follows that $t_j \le \tau_j$ for each order σ_j of σ:

- For each order $\sigma_j \in A(\sigma) \setminus I$, we have $1 \le j \le j^*$. By $\sum_{t=1}^{m} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, the first j orders, $\sigma_1, \sigma_2, \cdots, \sigma_j$, in sequence σ must all be shipped out on or before day τ_j under $\mathbf{z}$. Due to the additional condition (iv) of problem RP(K, Q), we know that $\tau_j \le K'$. Thus, by (2.19), we have $\sum_{j'=1}^{j} q_{\sigma_{j'}} \le \sum_{t'=1}^{\tau_j} Q_{t'} \le \tau_j c$, which implies that $t_j = \lceil \sum_{j'=1}^{j} q_{\sigma_{j'}}/c \rceil \le \tau_j$.

- For each order $\sigma_j \in I$, we have $j^* + 1 \le j \le |\sigma|$. By the additional constraint (iii) of problem RP(K, Q), the products for order σ_j must be shipped out on day $t_j = \lceil \sum_{j'=1}^{j} q_{\sigma_{j'}}/c \rceil$ under $\mathbf{z}$, which implies that $t_j = \tau_j$.

From σ, we can follow the procedure in Section 2.3 to construct a solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$ for model ILP. Since, under solution $(\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$, both the production completion day and the shipped-out day of each order σ_j are equal to t_j, we know that under solution $(\boldsymbol{\lambda}, \mathbf{x}(\sigma), \mathbf{z})$, products for each order are completed on or before their shipped-out day. Moreover, according to its construction, and by an argument similar to that in the proof of Lemma 2.1, the production plan $\mathbf{x}(\sigma)$ satisfies the capacity constraint. Thus, we can obtain that $(\boldsymbol{\lambda}, \mathbf{x}(\sigma), \mathbf{z})$ is a feasible solution to model ILP. Therefore, since $\boldsymbol{\lambda}$, $\mathbf{z}$ and I also satisfy the additional constraints (i)–(iv), $(\boldsymbol{\lambda}, \mathbf{x}(\sigma), \mathbf{z}, I)$ is a feasible solution to problem RP(K, Q). Proposition 2.2 is proved. □

Similar to Section 2.4.2, for each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (2.19) satisfied and for any $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$, we define $F(Q'; Q_1, Q_2, \ldots, Q_{K'})$ as the minimum total shipping and rejection cost among all order acceptance plans $\boldsymbol{\lambda}$, all order subsets I

that satisfy $\sum_{i\in I} q_i = Q'$, and all shipping plans $\mathbf{z}$ that satisfy $\sum_{i\in N\setminus I:\lambda_i=1} q_i z_{it} = Q_t$ for $t \in \{1, 2, \cdots, K'\}$, $\sum_{t=1}^{m} z_{it} = \lambda_i$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, and the additional constraints (i)–(iv) of problem RP(K, Q). By Proposition 2.2, we know that to solve problem RP(K, Q), it is equivalent to minimizing $F(Q'; Q_1, Q_2, \ldots, Q_{K'})$, which can be achieved by dynamic programming, as shown below.

For each $i \in \{0, 1, \cdots, n\}$, we still use $N(i) = \{i' \in N \mid i' \leq i\}$ to denote the set of orders $i' \in N$ with $i' \leq i$. For each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (2.19) satisfied, and for each $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$, we define a value function $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ as the minimum shipping and rejection cost of a subproblem of RP(K, Q). The value function aims to find an order acceptance plan $\boldsymbol{\lambda}$ and a shipping plan $\mathbf{z}$ only for orders in $N(i)$ and an order subset I of $N(i)$, such that the following constraints are satisfied:

$$\sum_{i'\in I} q_{i'} = Q', \tag{2.20}$$

$$\sum_{i'\in N(i)\setminus I:\lambda_{i'}=1} q_{i'} z_{i',t} = Q_t, \text{ for } t \in \{1, 2, \cdots, K'\}, \tag{2.21}$$

$$\sum_{t=1}^{m} z_{i',t} = \lambda_{i'}, \text{ for } i' \in N(i), \tag{2.22}$$

$$\sum_{t=d_{i'}+1}^{m} z_{i',t} = 0, \text{ for } i' \in N(i), \tag{2.23}$$

$$\sum_{t'=1}^{K'} z_{i't'} = \lambda_i, \text{ for } i' \in N(i) \setminus I, \tag{2.24}$$

$$z_{it} = 1, \text{ for } i \in I \text{ and for } t = \lceil (Q + \textstyle\sum_{i'\in I:i'\leq i} q_{i'})/c \rceil, \tag{2.25}$$

$$z_{i't} \in \{0, 1\}, \text{ for } i' \in N(i) \text{ and } t \in T, \tag{2.26}$$

$$\lambda_{i'} \in \{0, 1\}, \text{ for } i' \in N(i). \tag{2.27}$$

Among these constraints, (2.20) restricts the total product quantity of accepted orders in I, (2.21) restricts the total shipped-out quantity of accepted orders in $N(i) \setminus I$ on each day t, (2.22) and (2.23) ensure that products for accepted orders in $N(i)$ are shipped out not later than their committed delivery due dates, (2.24) and (2.25) are derived from the additional constraints (iii) and (iv) of problem RP(K, Q) for accepted orders in $N(i) \setminus I$ and accepted orders in I, respectively, (2.26) are binary constraints on $z_{i't}$ for $i' \in N(i), t \in T$, and (2.27) are binary constraints on $\lambda_{i'}$ for $i' \in N(i)$.

If the subproblem defined above has no such order acceptance plan $\boldsymbol{\lambda}$, order subset I and shipping plan $\mathbf{z}$, the value of $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ is $+\infty$. By definition, we have $F(Q'; Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$.

The value function $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$ can be computed recursively as follows. Since the subproblem of $F(0; 0; 0, \ldots, 0)$ is defined for an empty order set, we obtain the boundary condition of the dynamic program that $F(0; 0; 0, \ldots, 0) = 0$, and that $F(0; Q'; Q_1, \ldots, Q_{K'}) = +\infty$ for each $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ and $(Q_1, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$, with (2.19) and $Q' + \sum_{t=1}^{K'} Q_t > 0$ satisfied.

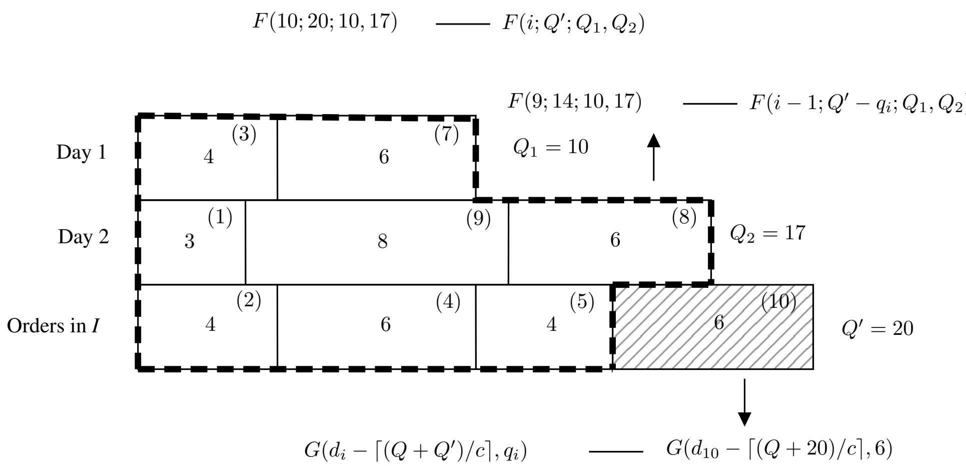
For each $i \in \{1, 2, \ldots, n\}$, for each $Q' \in \{0, 1, \cdots, \bar{Q} - Q\}$, and for each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (2.19) satisfied, consider the following three possible cases of an order acceptance plan $\boldsymbol{\lambda}$, set I, and a shipping plan $\mathbf{z}$ that form an optimal solution to the subproblem of $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$:

Case 1. (see Figure 2.4(a) for an illustration example): Order i is accepted under the order acceptance plan $\boldsymbol{\lambda}$ and $i \in I$. Due to constraints (2.20) and (2.25) of the subproblem, and due to $I \subseteq N(i)$, the products for order i are shipped out on day $\lceil (Q + \sum_{i' \in I : i' \le i} q_i)/c \rceil = \lceil (Q + \sum_{i' \in I} q_i)/c \rceil = \lceil (Q + Q')/c \rceil$ under the shipping plan $\mathbf{z}$, which cannot be later than the committed delivery due date d_i for order i. Thus, order i satisfies the following condition:

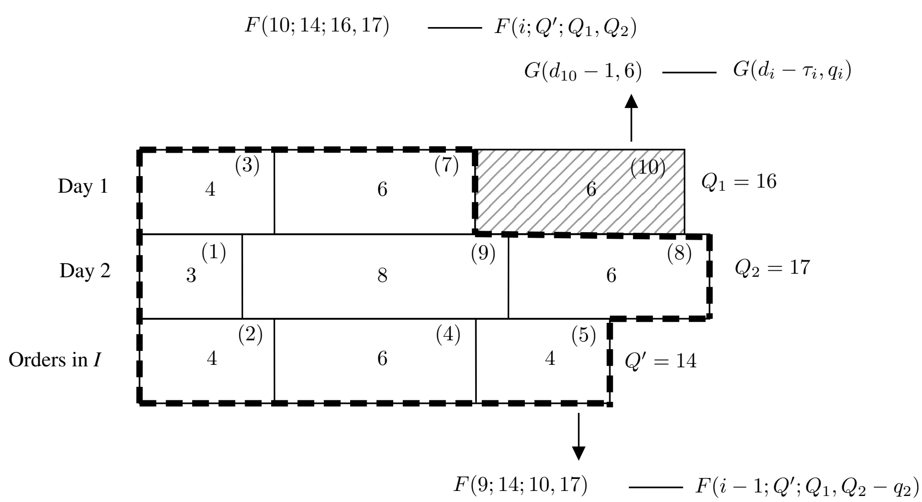
$$\lceil (Q + Q')/c \rceil \le d_i. \tag{2.28}$$

Moreover, due to constraint (2.20) of the subproblem, we know that order i also satisfies that $q_i \le Q'$. Since the products for order i are shipped out on day $\lceil (Q + Q')/c \rceil$, the shipping cost of order i equals $G(d_i - \lceil (Q + Q')/c \rceil, q_i)$. Moreover, the order

Figure 2.4: Examples for Case 1 and Case 2 considered in solving problem RP(K, Q) by dynamic programming, where $K' = 2$ and order 6 is rejected in the optimal solution.



(a) Case 1: For the subproblem of $F(10; 20; 10, 17)$ defined for orders in $\{1, 2, \cdots, 10\}$, if order 10 is accepted and is in I in an optimal solution (i.e., $10 \in I$), then $F(10, 20; 10, 17)$ equals the sum of $F(9; 14; 10, 17)$ and $G(d_{10} - \lceil (Q + 20)/c \rceil, 6)$ (since products for order 10 must be shipped out on day $\lceil (Q + 20)/c \rceil$ and since $q_{10} = 6$).



(b) Case 2: For the subproblem of $F(10; 14; 16, 17)$ defined for orders in $\{1, 2, \cdots, 10\}$, if order 10 is accepted and is not in I, and if order 10 is shipped out on day 1 in an optimal solution (i.e., $\tau_{10} = 1$), then $F(10, 14; 16, 17)$ equals the sum of $F(9; 14; 10, 17)$ and $G(d_{10} - 1, 6)$ (since $\tau_{10} = 1$ and $q_{10} = 6$).

acceptance plan and shipping plan for orders in $N(i-1) = N(i) \setminus \{i\}$ under $\boldsymbol{\lambda}$ and $\mathbf{z}$, together with the order subset $(I \setminus \{i\})$, must form an optimal solution to the subproblem of $F(i-1; Q' - q_i; Q_1, \ldots, Q_{K'})$. Here, by definition, the subproblem of $F(i-1; Q' - q_i; Q_1, \ldots, Q_{K'})$ is defined for orders in $N(i-1)$, for the total product quantity Q' for orders in I reduced by q_i, and for the same total shipped-out quantities for accepted orders in $N(i-1)\setminus I$ on each day. Therefore, $F(i;Q';Q_1,\dots,Q_{K'})$ equals $F(i-1;Q'-q_i;Q_1,\dots,Q_{K'})+G(d_i-\lceil(Q+Q')/c\rceil,q_i)$.

Case 2. (see Figure 2.4(b) for an illustration example): Order i is accepted under the order acceptance plan $\boldsymbol{\lambda}$ and $i\in N(i)\setminus I$. Let $\tau_i\in\{1,2,\cdots,K'\}$ indicate the shipped-out day of order i under the shipping plan $\mathbf{z}$. Due to constraints (2.23) and (2.24) of the subproblem, τ_i satisfies that $\tau_i\le d_i$ and $q_i\le Q_{\tau_i}$. Thus, the shipping cost of order i equals $G(d_i-\tau_i,q_i)$. Moreover, the order acceptance plan and the shipping plan for orders in $N(i-1)=N(i)\setminus\{i\}$ under $\boldsymbol{\lambda}$ and $\mathbf{z}$, together with the order subset I, must form an optimal solution to the subproblem of $F(i-1;Q';Q_1\dots,Q_{\tau_i-1},Q_{\tau_i}-q_i,Q_{\tau_i+1},\dots,Q_{K'})$. Here, by definition, the subproblem of $F(i-1;Q';Q_1\dots,Q_{\tau_i-1},Q_{\tau_i}-q_i,Q_{\tau_i+1},\dots,Q_{K'})$ is defined for orders in $N(i-1)$, for the same total product quantity for accepted orders in I, and for the same total shipped-out quantities for accepted orders in $N(i)\setminus I$ on each day except day τ_i, of which the quantity Q_{τ_i} is reduced by q_i. Therefore, $F(i;Q';Q_1,\dots,Q_{K'})$ is the minimum value of $F(i-1;Q';Q_1\dots,Q_{\tau_i-1},Q_{\tau_i}-q_i,Q_{\tau_i+1},\dots,Q_{K'})+G(d_i-\tau_i,q_i)$ over all $\tau_i\in\{1,2,\dots,K'\}$ with $\tau_i\le d_i$ and $q_i\le Q_{\tau_i}$.

Case 3. Order i is rejected under the order acceptance plan $\boldsymbol{\lambda}$, for which a rejection cost r_i is incurred. The order acceptance plan and shipping plan for orders in $N(i-1)=N(i)\setminus\{i\}$ under $\boldsymbol{\lambda}$ and $\mathbf{z}$, together with the order subset I, must form an optimal solution to the subproblem of $F(i-1;Q';Q_1,\dots,Q_{K'})$.

Accordingly, we can obtain the following recursive equation to compute $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$:

$$F(i; Q'; Q_1, \ldots, Q_{K'})$$
$$= \min \left\{ \begin{array}{l} \left\{ \begin{array}{l} F(i-1; Q'-q_i; Q_1, \ldots, Q_{K'}) + G(d_i - \lceil (Q+Q')/c \rceil, q_i), \\ \qquad\qquad\qquad\qquad \text{if (2.28) is satisfied and } q_i \le Q'; \\ +\infty, \text{ otherwise.} \end{array} \right. \\ \min \Big\{ F(i-1; Q'; Q_1 \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i+1}, \ldots, Q_{K'}) \\ \qquad + G(d_i - \tau_i, q_i) \mid \forall \tau_i \in \{1, 2, \ldots, K'\} \text{ with } \tau_i \le d_i \text{ and } q_i \le Q_{\tau_i} \Big\} \\ F(i-1; Q'; Q_1, \ldots, Q_{K'}) + r_i. \end{array} \right\}, \tag{2.29}$$

where we assume that the minimum over an empty set equals $+\infty$.

Finally, note that $F(Q'; Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$ with $\sum_{t=1}^{K'} Q_t = Q$. If $\min\{F(n; Q'; Q_1, Q_2, \cdots, Q_{K'}) | Q' = 0, 1, \ldots, \bar{Q} - Q\} = +\infty$ with $\sum_{t=1}^{K'} Q_t = Q$, then problem $\mathrm{RP}(K, Q)$ has no feasible solution and we return $+\infty$. Otherwise, we can enumerate all $(Q'; Q_1, Q_2, \cdots, Q_{K'})$ for $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ and $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with $\sum_{t=1}^{K'} Q_t = Q$ and (2.19) satisfied, so as to find $(Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ that minimizes $F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$. By backtracking the computational process of $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ with $\sum_{t=1}^{K'} Q'_t = Q$, we can obtain $\boldsymbol{\lambda}$, $\mathbf{z}$ and I that minimize the total cost for the subproblem of $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ with $\sum_{t=1}^{K'} Q'_t = Q$. By following the proof of Proposition 2.2, we can then construct $\mathbf{x}$ so that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ is a feasible solution to problem $\mathrm{RP}(K, Q)$, having its total cost equal to $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$. We summarize this exact algorithm for problem $\mathrm{RP}(K, Q)$ in Algorithm 2.3.

Algorithm 2.3 (for problem $\mathrm{RP}(K, Q)$)

1: $F(0; 0; 0, 0, \ldots, 0) \leftarrow 0$, and $F(0; Q'; Q_1, Q_2, \ldots, Q_{K'}) \leftarrow +\infty$ for each $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ and for each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (2.19) and $Q' + \sum_{t=1}^{K'} Q_t > 0$ satisfied

2: **for all** $i = 1, 2, \cdots, n$ **do**

3:     **for all** $Q' = 0, 1, \cdots, \bar{Q} - Q$ **do**

for all $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}^{K'}$ with (2.19) satisfied **do**

Compute $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ by the recursive equation in (2.29)

end for

end for

end for

if $F(n; Q'; Q_1, Q_2, \cdots, Q_{K'}) = +\infty$ for each $(Q'; Q_1, Q_2, \cdots, Q_{K'})$ with $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$, $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$, $\sum_{t=1}^{K'} Q_t = Q$, and (2.19) satisfied **then**

return $+\infty$

else

Find $(Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ that minimizes the value of $F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$ among all $(Q'; Q_1, Q_2, \cdots, Q_{K'})$ for $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ and $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with $\sum_{t=1}^{K'} Q_t = Q$ and (2.19) satisfied.

Reconstruct $\boldsymbol{\lambda}$, $\mathbf{z}$ and I that minimize the total cost for the subproblem of $F(n; Q''; Q'_1, \cdots, Q'_K)$ by backtracking the computational process of $F(n; Q''; Q'_1, \cdots, Q'_{K'})$ with $\sum_{t=1}^{K'} Q'_t = Q$

Construct $\mathbf{x}$ according to the proof of Proposition 2.2 so that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ is a feasible solution to problem $\mathrm{RP}(K, Q)$

return $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$

end if

The correctness and time complexity of Algorithm 2.3 are presented in Theorem 2.5, which also indicates that if problem $\mathrm{RP}(K, Q)$ has a feasible solution, then the optimal solution obtained by Algorithm 2.3 for problem $\mathrm{RP}(K, Q)$ leads to a feasible solution to problem IPTSDA with the same total shipping and rejection cost.

Theorem 2.5. *For every $Q \in \{0, 1, \ldots, \bar{Q}\}$ and $K \in \{1, 2, \ldots, m\}$, if problem $\mathrm{RP}(K, Q)$ has no feasible solution, Algorithm 2.3 returns $+\infty$. Otherwise, Algorithm 2.3 returns an optimal solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ to problem $\mathrm{RP}(K, Q)$ in $O(nmc^{K+1}K \cdot K!)$ time, and $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ forms a feasible solution to model ILP of problem IPTSDA with the same total cost as that* *of* $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$.

Proof. As we have shown above, the value function $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ can be computed recursively by (2.29), and $F(Q'; Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$ with $\sum_{t=1}^{K'} Q_t = Q$. Thus, if problem RP(K, Q) has no feasible solution, then $F(n; Q'; Q_1, Q_2, \cdots, Q_{K'}) = +\infty$, and Algorithm 2.3 returns $+\infty$. Otherwise, the minimum total cost for the input instance of problem RP(K, Q) equals the minimum value of $F(n; Q'; Q_1, Q_2, \ldots, Q_{K'})$ among all $Q' \in \{0, 1, \ldots, \bar{Q} - Q\}$ and all $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (2.19) and $\sum_{t=1}^{K'} Q_t = Q$ satisfied, which equals $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ for $(Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ found by Step 12 of Algorithm 2.3. Therefore, $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ obtained in Step 14 and returned by Step 15 of Algorithm 2.3 is an optimal solution to problem RP(K, Q). By the definition of problem RP(K, Q), we know that $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ must also be a feasible solution to model ILP of problem IPTSDA. Since the total shipping cost depends only on $\mathbf{z}$, it must be the same for both $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ and $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$.

Moreover, (2.19) implies that $Q_t \le tc$ for $1 \le t \le K'$. This, together with $Q \le \sum_{i \in N} q_i \le mc$, implies that the recursive equation (2.29) is computed in Algorithm 2.3 for at most $n \cdot mc \cdot (1 \cdot c)(2 \cdot c) \cdots (K' \cdot c) = nmc \cdot c^{K'} \cdot K'!$ time. Since it takes $O(K')$ time to compute the recursive equation (2.29), and since $K' \le K$ as shown in (2.18), we obtain that the total time complexity of Algorithm 2.3 is $O(nmc^{K+1} K \cdot K!)$. This completes the proof of Theorem 2.5. □

2.5.2 Approximation Scheme: Algorithm and Analysis

2.5.2.1 Algorithm

Based on Algorithm 2.3, we can follow the idea presented at the beginning of Section 2.5 to develop an approximation scheme for problem IPTSDA, which is illustrated in Algorithm 2.4. For any given but fixed $\epsilon > 0$, the algorithm first sets $K = \min\{\lceil 1/\epsilon \rceil, m\}$. Then, for each $Q \in \{0, 1, \ldots, \bar{Q}\}$ where $\bar{Q} = \sum_{i \in N} q_i$ such that $K \ge \lceil Q/c \rceil$ in (2.16) is satisfied, it applies Algorithm 2.3 to solve the restricted problem RP(K, Q). By Theorem 2.5, if problem RP(K, Q) has a feasible solution, Algorithm 2.3 returns an optimal solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ to the restricted problem RP(K, Q), which yields a feasible solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to model ILP of problem IPTSDA. Among all such feasible solutions $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ obtained, the algorithm finally selects and returns the one with the lowest total cost as an approximation solution to problem IPTSDA.

Algorithm 2.4 (an approximation scheme for problem IPTSDA)

For given $\epsilon > 0$, set $K \leftarrow \min\left\{\lceil 1/\epsilon \rceil, m\right\}$
for all $Q \in \{0, 1, \ldots, \bar{Q}\}$ with $K \geq \lceil Q/c \rceil$ **do**
  Apply Algorithm 2.3 to solve the restricted problem RP(K, Q), which returns an optimal solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z}, I)$ to problem RP(K, Q), yielding a feasible solution $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ to model ILP, if problem RP(K, Q) has a feasible solution.
end for
return the feasible solution that has the lowest total cost among all $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ obtained for model ILP.

2.5.2.2 Analysis

Theorem 2.6 below can be established for Algorithm 2.4, indicating that Algorithm 2.4 is a pseudo-polynomial time approximation scheme for problem IPTSDA.

Theorem 2.6. *For any given but fixed $\epsilon > 0$, Algorithm 2.4 is a pseudo-polynomial time approximation scheme for problem IPTSDA with a worst-case performance ratio of $(1 + \epsilon)$.*

To prove Theorem 2.6, we first need to prove that Algorithm 2.4 returns a feasible solution in pseudo-polynomial time for any given $\epsilon > 0$, which turns out to be straightforward, as illustrated in the proof of Lemma 2.3 below.

Lemma 2.3. *Algorithm 2.4 runs in $O(nm^2 c^{\lceil 1/\epsilon \rceil + 2} \cdot \lceil 1/\epsilon \rceil! \cdot \lceil 1/\epsilon \rceil)$ time, which is a pseudo-polynomial running time for any given $\epsilon > 0$.*

Proof. Since $\bar{Q} = \sum_{i \in N} q_i \leq mc$, Algorithm 2.4 executes Algorithm 2.3 for at most mc times. Thus, by Theorem 2.5 and $K \leq \lceil 1/\epsilon \rceil$, Algorithm 2.4 runs in $O(nm^2 c^{\lceil 1/\epsilon \rceil + 2} \cdot \lceil 1/\epsilon \rceil! \cdot \lceil 1/\epsilon \rceil)$ time, which is a pseudo-polynomial running time for any given $\epsilon > 0$. $\square$

Due to Lemma 2.3, to prove Theorem 2.6, we now only need to prove that Algorithm 2.4 is an approximation scheme, i.e., Algorithm 2.4 always returns a feasible solution to problem IPTSDA with a total cost not exceeding $(1 + \epsilon)$ times that of an optimal solution, for any given $\epsilon > 0$. The details of the proof is in Section 2.5.2.4.

2.5.2.3 Main idea to prove Theorem 2.6

To prove that Algorithm 2.4 has a worst-case performance ratio of $(1 + \epsilon)$, our main idea is as follows, where $\xi_i(\cdot)$ and $\hat{\xi}_i(\cdot)$ represent certain total shipping and rejection costs of each order $i \in N$ under a certain solution, and their detailed definitions will be explained later.

- First, we can construct a restricted problem RP(K, Q^*) with $K = \min\{\lceil 1/\epsilon \rceil, m\}$ and with Q^* determined from the optimal solution $\pi^* = (\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$, where π^* is constructed from an order sequence σ^* by Theorem 2.1 and the procedure in Section 2.3. We can show that problem RP(K, Q^*) must have been solved in Steps 2–4 of Algorithm 2.4. If problem RP(K, Q^*) has a feasible solution, then an optimal solution $(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ to problem RP(K, Q^*) must have been obtained in Steps 2–4 of Algorithm 2.4, leading to a feasible solution $\tilde{\pi} = (\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ to model ILP. Accordingly, it can be shown that to prove that Algorithm 2.4 has a worst-case performance ratio of $(1 + \epsilon)$, we only need to prove that problem RP(K, Q^*) has a feasible solution, and that the total cost of orders in N under $\tilde{\pi}$ does not exceed $(1 + \epsilon)$ times the total cost under π^*. This, as shown in Section 2.5.2.4, is equivalent to proving that $\sum_{i \in N} \xi_i(\tilde{\pi}) \leq (1 + \epsilon) \sum_{i \in N} \xi_i(\pi^*)$.

- Second, we can construct an order subset I' from π^* and a new order sequence σ' from σ^* and I', which yield a feasible solution π' to model ILP such that π' and

I' also form a feasible solution to the restricted problem RP(K, Q^*), implying that problem RP(K, Q^*) has a feasible solution. We can also prove that the total cost of orders in N under $\tilde{\pi}$ does not exceed that of orders in $N \setminus I'$ under π^* plus that of orders in I' under π'. This, as shown in Section 2.5.2.4, is equivalent to proving that $\sum_{i \in N} \xi_i(\tilde{\pi}) \leq \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \xi_i(\pi')$.

- Third, we can construct a new instance of problem IPTSDA by splitting each accepted order $i \in I'$ into q_i unit orders, each having a unit product quantity. From σ^* and σ', we can obtain order sequences $\hat{\sigma^*}$ and $\hat{\sigma'}$ for this new problem instance, which yield two feasible solutions $\hat{\pi^*}$ and $\hat{\pi'}$ to the new problem instance, respectively. We can prove that the total cost of the unit orders split from accepted orders in I' under $\hat{\pi'}$ does not exceed that under $\hat{\pi^*}$, which does not exceed the total cost of accepted orders in I' under π^*. This, as shown in Section 2.5.2.4, is equivalent to proving that $\sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi^*}) \leq \sum_{i \in I'} \xi_i(\pi^*)$.

- Fourth, we can prove that the difference between the total cost of orders in I' under π' and that of unit orders split from orders in I' under $\hat{\pi'}$ does not exceed ϵ times the total cost of the order in I' under π^*. This, as shown in Section 2.5.2.4, is equivalent to proving that $\sum_{i \in I'} \xi_i(\pi') \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) + \epsilon \sum_{i \in I'} \xi_i(\pi^*)$.

From items 2–4 above we can obtain that $\sum_{i \in N} \xi_i(\tilde{\pi}) \leq (1+\epsilon) \sum_{i \in N} \xi_i(\pi^*)$, as shown below:

$$
\begin{aligned}
\sum_{i \in N} \xi_i(\tilde{\pi}) &\leq \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \xi_i(\pi') \leq \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) + \epsilon \sum_{i \in I'} \xi_i(\pi^*) \\
&\leq \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \xi_i(\pi^*) + \epsilon \sum_{i \in I'} \xi_i(\pi^*) \leq (1+\epsilon) \sum_{i \in N} \xi_i(\pi^*).
\end{aligned}
$$

Thus, from item 1 above we obtain that Algorithm 2.4 has a worst-case performance ratio of $(1+\epsilon)$. Hence, by Lemma 2.3, Theorem 2.6 can be proved.

2.5.2.4 Proof of Theorem 2.6

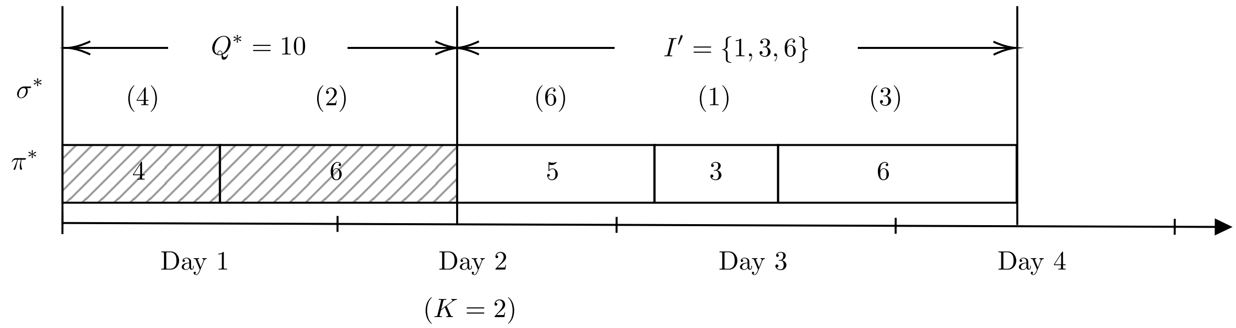
By following the main idea above, we are now going to prove that Algorithm 2.4 has a worst-case performance ratio of $(1+\epsilon)$. First, we construct a restricted problem RP(K, Q^*) as follows. Recall that $K = \min\{\lceil 1/\epsilon \rceil, m\}$. By Theorem 2.1, there must exist an order sequence σ^* such that the solution $\pi^* = (\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$, which is constructed from σ^* by the procedure described in Section 2.3, forms an optimal solution to model ILP. We define Q^* as the total product quantity shipped out on or before day K under the optimal solution π^* (see Figure 2.5(a) for an illustrative example). Thus, $\lceil Q^*/c \rceil$ also indicates the earliest possible day on which the production is completed of all the products for accepted orders that are shipped out on or before day K under π^*. Hence, $K \geq \lceil Q^*/c \rceil$, implying that K and Q^* satisfy (2.16).

Consider the restricted problem RP(K, Q^*). Since $0 \leq Q^* \leq \sum_{i \in N} q_i = \bar{Q}$, during the iteration in Steps 2–4, Algorithm 2.4 must have applied Algorithm 2.3 to solve the restricted problem RP(K, Q^*). Thus, if problem RP(K, Q^*) has a feasible solution, then Algorithm 2.3 must return an optimal solution $(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ to it, which yields a feasible solution $\tilde{\pi} = (\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ to model ILP. For each $i \in N$, let $\xi_i(\pi^*)$ and $\xi_i(\tilde{\pi})$ indicate the costs of order i under the solution π^* and the solution $\tilde{\pi}$, respectively. Since the solution returned by Step 5 of Algorithm 2.4 must have a total cost no greater than that of solution $\tilde{\pi}$, to prove that Algorithm 2.4 has a worst-case performance ratio of $(1+\epsilon)$, we only need to prove that problem RP(K, Q^*) has a feasible solution, and that

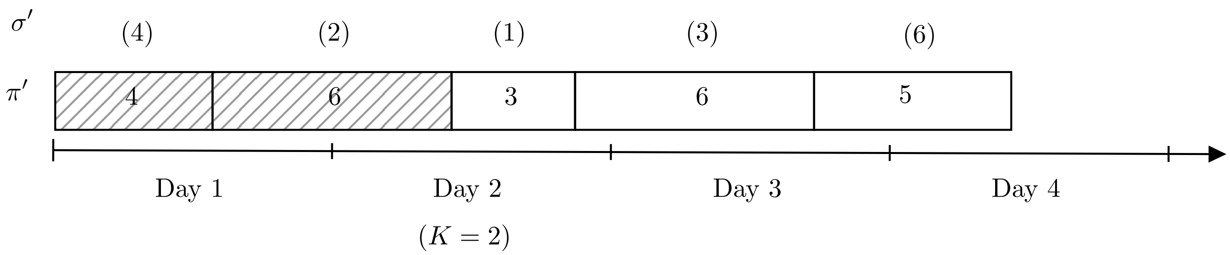
$$\sum_{i \in N} \xi_i(\tilde{\pi}) \leq (1+\epsilon) \sum_{i \in N} \xi_i(\pi^*). \tag{2.30}$$

Second, we construct an order subset I' from π^*, and then construct a new order sequence σ' from σ^* and I' as follows, which yield a feasible solution π' to model ILP such that π' and I' also form a feasible solution to the restricted problem RP(K, Q^*). Let $(\boldsymbol{\lambda}, \mathbf{x}, \mathbf{z})$ denote $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ of solution π^*. We define I' to be the set of accepted orders shipped after

Figure 2.5: Illustrative examples for the proof of Theorem 2.6 where $K = 2$ and $d_1 \leq d_2 \leq \ldots \leq d_6$: Defining σ^*, Q^*, I', σ', and π'. Note that order 5 is rejected in the optimal solutions shown here.



(a) An optimal solution $\pi^* = (\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*))$ is constructed from the order sequence $\sigma^* = (4, 2, 6, 1, 3)$, and from π^*, Q^* is defined to be the total product quantity of orders shipped out on or before day $K = 2$ under π^*, and the orders whose products are shipped out after day $K = 2$ under π^* form set $I' = \{1, 3, 6\}$.



(b) From σ^* and $I' = \{1, 3, 6\}$ shown in Figure 2.5(a), a new order sequence $\sigma' = (4, 2, 1, 3, 6)$ is constructed by rearranging orders of I' in an increasing order of their indices, and from σ' a new solution $\pi' = (\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'))$ is constructed.

Q^* units of products under π^* (see Figure 2.5(a) for an illustrative example). Accordingly, the first $|\{i|N \setminus I', \lambda_i = 1\}|$ orders of sequence σ^* are shipped out on or before day K, forming set $\{i|N \setminus I', \lambda_i = 1\}$, and the last $|I'|$ orders of sequence σ^* are shipped out on or after day K, forming set I'. From π^* and I', we can construct a new order sequence σ' by changing only the subsequence of the orders in I', such that they are in an increasing order of their indices. Following the procedure described in Section 2.3 we can construct from σ' a solution $\pi' = (\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'))$ for model ILP (see Figure 2.5(b) for an illustrative example). Lemma 2.4 can then be established.

Lemma 2.4. $\pi' = (\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'))$ *is a feasible solution to model ILP of problem IPTSDA, and* $(\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'), I')$ *is a feasible solution to the restricted problem* $\mathrm{RP}(K, Q^*)$.

Proof. Let $\bar{n} = |\sigma'|$ denote the length of the order sequence σ'. Since σ' is constructed from σ^* by changing only the subsequence of the last $|I'|$ orders, we have that $|\sigma^*| = |\sigma'| = \bar{n}$. Accordingly, we can represent the order sequences σ^* and σ' by $\sigma^* = (\sigma_1^*, \sigma_2^*, \cdots, \sigma_{\bar{n}}^*)$ and $\sigma' = (\sigma_1', \sigma_2', \cdots, \sigma_{\bar{n}}')$, respectively. Let j indicate the smallest index such that order σ_{j+1}^* appears ahead of order σ_j^* in σ'. If such an index j does not exist, implying that $\sigma^* = \sigma'$, then since $(\boldsymbol{\lambda}(\sigma^*), \mathbf{x}(\sigma^*), \mathbf{z}(\sigma^*), I')$ is a feasible solution to $\text{RP}(K, Q^*)$, which can be seen from the definition of I' and Q^*, we can see that Lemma 2.4 holds true.

Otherwise, from the definition of σ' we know that $\sigma_{j+1}^* < \sigma_j^*$, which, together with (2.17), implies that $d_{\sigma_{j+1}^*} \le d_{\sigma_j^*}$. We can construct a new sequence σ'' from σ^* by swapping the positions of orders σ_j^* and σ_{j+1}^*. Consider the solution $\pi'' = (\boldsymbol{\lambda}(\sigma''), \mathbf{x}(\sigma''), \mathbf{z}(\sigma''))$ constructed from σ'' by the procedure described in Section 2.3. We now show as follows that π'' is a feasible solution to model ILP. First, from the construction procedure we know that under each solution $\pi \in \{\pi^*, \pi''\}$, each order $\sigma_{j''}$ with $j \in \{1, 2, \cdots, \bar{n}\}$ is accepted, and products for each order $\sigma_{j''}$ with $j \in \{1, 2, \cdots, \bar{n}\}$ both complete their productions and are shipped out on day $\lceil \sum_{j'=1}^{j''} q_{\sigma_{j'}} / c \rceil$, so that the production capacity of each day is not exceeded. Second, since $\sigma_{j'}'' = \sigma_{j'}^*$ for $j' \in \{1, 2, \cdots, j-1, j+2, \cdots, \bar{n}\}$, we have $\lceil \sum_{j'=1}^{j''} q_{\sigma_{j'}''} / c \rceil = \lceil \sum_{j'=1}^{j''} q_{\sigma_{j'}^*} / c \rceil$ for $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, \bar{n}\}$. Thus, since the optimal solution π^* is feasible to model ILP, and since $d_{\sigma_{j''}^*} = d_{\sigma_{j''}''}$, we obtain that for each $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, \bar{n}\}$,

$$\lceil \sum_{j'=1}^{j''} q_{\sigma_{j'}''} / c \rceil = \lceil \sum_{j'=1}^{j''} q_{\sigma_{j'}^*} / c \rceil \le d_{\sigma_{j''}^*} = d_{\sigma_{j''}''}. \tag{2.31}$$

Thus, the shipped-out day of each order $\sigma_{j''}''$ with $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, \bar{n}\}$ is not later than its committed delivery due date. For the remaining two orders σ_j'' and σ_{j+1}'', we can also see that their shipped-out days are not later than their committed delivery due dates. To see this, we first know from (2.31) that $\lceil \sum_{j'=1}^{j+1} q_{\sigma_{j'}''} / c \rceil = \lceil \sum_{j'=1}^{j+1} q_{\sigma_{j'}^*} / c \rceil$. For order

σ''_j, its shipped-out day is $\lceil\sum_{j'=1}^{j} q_{\sigma''_{j'}}/c\rceil$, which, due to $\sigma^*_{j+1} = \sigma''_j$ and (2.31), satisfies that

$$\lceil\sum_{j'=1}^{j} q_{\sigma''_{j'}}/c\rceil \le \lceil\sum_{j'=1}^{j+1} q_{\sigma''_{j'}}/c\rceil = \lceil\sum_{j'=1}^{j+1} q_{\sigma^*_{j'}}/c\rceil \le d_{\sigma^*_{j+1}} = d_{\sigma''_j},$$

and thus is not later than its committed delivery due date. For order σ''_{j+1}, its shipped-out day is $\lceil\sum_{j'=1}^{j+1} q_{\sigma''_{j'}}/c\rceil$, which, due to $d_{\sigma^*_{j+1}} \le d_{\sigma^*_j}$ and (2.31), satisfies that

$$\lceil\sum_{j'=1}^{j+1} q_{\sigma''_{j'}}/c\rceil = \lceil\sum_{j'=1}^{j+1} q_{\sigma^*_{j'}}/c\rceil \le d_{\sigma^*_{j+1}} \le d_{\sigma^*_j} = d_{\sigma''_{j+1}},$$

and thus is not later than its committed delivery due date.

Hence, π'' is a feasible solution to model ILP. Replacing σ^* with σ'' and repeating the argument above until $\sigma'' = \sigma'$, we can obtain that the resulting π' is still a feasible solution to model ILP.

Moreover, consider I' and the shipping plans $\mathbf{z}(\sigma')$ and $\mathbf{z}(\sigma^*)$. Let $A(\sigma^*) = \{\sigma^*_1, \sigma^*_2, \cdots, \sigma^*_{\bar{n}}\}$ and $A(\sigma') = \{\sigma'_1, \sigma'_2, \cdots, \sigma'_{\bar{n}}\}$ denote the sets of orders in σ^* and in σ', respectively. By the definition of Q', the definition of I', and the construction of π^*, we know that there exists an index n' such that $I' = \{\sigma^*_{n'+1}, \sigma^*_{n'+2}, \cdots, \sigma^*_{\bar{n}}\}$ contains all the accepted orders shipped out after day K under $\mathbf{z}(\sigma^*)$, and that $A(\sigma^*) \setminus I' = \{\sigma^*_1, \sigma^*_2, \cdots, \sigma^*_{n'}\}$ containing all the accepted orders shipped out on or before day K under $\mathbf{z}(\sigma^*)$. Since σ' and σ^* differ only in the subsequence of orders in I', we know that $A(\sigma^*) = A(\sigma')$. Accordingly, we can show as follows that the order acceptance plan $\boldsymbol{\lambda}(\sigma')$, the order subset I', and the shipping plan $\mathbf{z}(\sigma')$ satisfy the additional conditions (i)–(iv) of problem RP(K, Q^*):

- By the definition of I' and $A(\sigma^*) = A(\sigma')$, we know that orders in I' are all accepted in $\boldsymbol{\lambda}(\sigma')$. The additional condition (i) is satisfied.
- By the definition of I' and $A(\sigma')$, the total quantity of the accepted orders in $N \setminus I'$ under $\boldsymbol{\lambda}(\sigma')$ equals $\sum_{i\in A(\sigma')\setminus I'} q_i$, which, together with the definition of Q^* and $A(\sigma') = A(\sigma^*)$, satisfies that $\sum_{i\in A(\sigma')\setminus I'} q_i = \sum_{i\in A(\sigma^*)\setminus I'} q_i = Q^*$. The additional condition (ii) is

satisfied.

- For each order $\sigma_{j''} \in I'$ where $j'' \geq n'+1$, its shipped-out day under $\mathbf{z}(\sigma')$ is $\lceil \sum_{j'=1}^{j''} q_{\sigma'_{j'}}/c \rceil$. It can be seen that $\sum_{j'=1}^{j''} q_{\sigma'_{j'}} = \sum_{j'=1}^{n'} q_{\sigma'_{j'}} + \sum_{j'=n'+1}^{j''} q_{\sigma'_{j'}} = Q^* + \sum_{j'=n'+1}^{j''} q_{\sigma'_{j'}}$. Since orders in I' are ordered in σ' in an increasing order of their indices, we have $\sum_{j'=n'+1}^{j''} q_{\sigma'_{j'}} = \sum_{i' \in I': i' \leq \sigma'_{j''}} q_{i'}$, implying that $\lceil \sum_{j'=1}^{j''} q_{\sigma'_{j'}}/c \rceil = \lceil (Q^* + \sum_{i' \in I': i' \leq \sigma'_{j''}} q_{i'})/c \rceil$. Thus, for each order $i \in I'$, its shipped-out day under $\mathbf{z}(\sigma')$ is $\lceil (Q^* + \sum_{i' \in I': i' \leq i} q_{i'})/c \rceil$. The additional condition (iii) is satisfied.

- For each order $\sigma'_{j''} \in A(\sigma') \setminus I'$ where $1 \leq j'' \leq n'$, its shipped-out day under $\mathbf{z}(\sigma')$ is $\lceil \sum_{j'=1}^{j''} q_{\sigma'_{j'}}/c \rceil$. It can be seen that $\sum_{j'=1}^{j''} q_{\sigma'_{j'}} \leq \sum_{j'=1}^{n'} q_{\sigma'_{j'}} = \sum_{i \in A(\sigma') \setminus I'} q_i = \sum_{i \in A(\sigma^*) \setminus I'} q_i = Q^*$, implying that $\lceil \sum_{j'=1}^{j''} q_{\sigma'_{j'}}/c \rceil \leq \lceil Q^*/c \rceil = K'$. Thus, for each order $i \in A(\sigma') \setminus I'$, its shipped-out day under $\mathbf{z}(\sigma')$ must be on or before day K'. The additional condition (iv) is satisfied.

Therefore, $(\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'), I')$ is a feasible solution to RP(K, Q^*). Lemma 2.4 is proved.

□

By Lemma 2.4, we obtain that problem RP(K, Q^*) has a feasible solution $(\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'), I')$, and that $\pi' = (\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'))$ is a feasible solution to model ILP. Thus, to show that Algorithm 2.4 has a worst-case performance ratio of $(1+\epsilon)$, we only need to prove (2.30). To prove this, for each $i \in N$, let $\xi_i(\pi')$ indicate the cost of order i under π'. Since $(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ is an optimal solution to problem RP(K, Q^*), by Lemma 2.4, the total cost of $(\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ should not be greater than that of $(\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'), I')$. Thus, the total cost of $\tilde{\pi} = (\tilde{\boldsymbol{\lambda}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ should not be greater than that of $\pi' = (\boldsymbol{\lambda}(\sigma'), \mathbf{x}(\sigma'), \mathbf{z}(\sigma'))$, implying that

$$\sum_{i \in N} \xi_i(\tilde{\pi}) \leq \sum_{i \in N} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi') + \sum_{i \in I'} \xi_i(\pi'). \tag{2.32}$$

Moreover, since the positions of accepted orders of $N \setminus I'$ in σ' are the same as that in σ^*, the shipped-out days for accepted orders of $N \setminus I'$ under π' are the same as that under π^*.

Also, we know that the acceptance plan $\boldsymbol{\lambda}(\sigma')$ is the same as $\boldsymbol{\lambda}(\sigma^*)$. Thus, we have

$$\sum_{i \in N \setminus I'} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi^*). \tag{2.33}$$

From (2.32) and (2.33) we obtain that

$$\sum_{i \in N} \xi_i(\tilde{\pi}) \leq \sum_{i \in N \setminus I'} \xi_i(\pi') + \sum_{i \in I'} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \xi_i(\pi'). \tag{2.34}$$

Third, we construct a new instance of problem IPTSDA by splitting each accepted order $i \in I'$ into q_i orders with each having a unit product quantity and the same committed delivery due date as order i. We denote these unit orders by $(i, 1)$, $(i, 2)$, …, and (i, q_i). Thus, these unit orders split from order i do not need to be shipped out together.

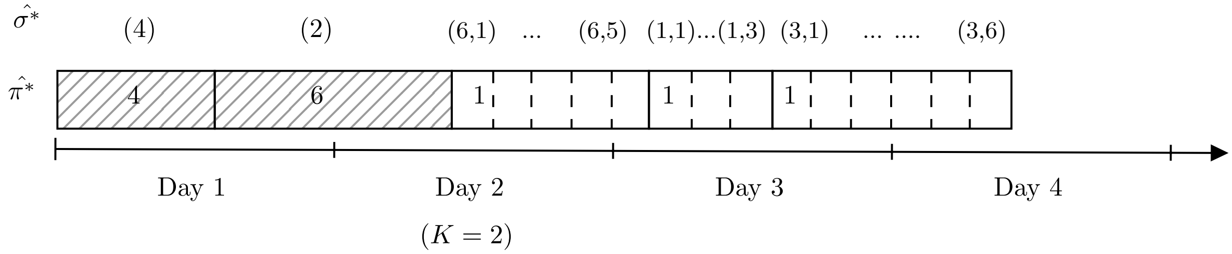
Consider any sequence σ of orders such that $\pi = (\boldsymbol{\lambda}(\sigma), \mathbf{x}(\sigma), \mathbf{z}(\sigma))$, which is constructed from σ by the procedure described in Section 2.3, forms a feasible solution to the original problem instance. From σ, we can construct an order sequence $\hat{\sigma}$ of orders for the new problem instance by replacing each order $i \in I'$ in σ with a subsequence of the unit orders $(i, 1)$, $(i, 2)$, …, and (i, q_i). By the procedure described in Section 2.3 we can also construct from $\hat{\sigma}$ a solution $\hat{\pi} = (\hat{\boldsymbol{\lambda}}(\hat{\sigma}), \hat{\mathbf{x}}(\hat{\sigma}), \hat{\mathbf{z}}(\hat{\sigma}))$ for the new problem instance. For each $i \in I'$, let $\hat{\xi}_i(\hat{\pi})$ indicate the total cost of all the unit orders (i, p) split from order i under $\hat{\pi}$. See Figure 2.6 for two illustrative examples for $\sigma = \sigma^*$ and $\sigma = \sigma'$, respectively. Lemma 2.5 can then be established for $\hat{\pi}$.

Lemma 2.5. *$\hat{\pi}$ is a feasible solution to the new instance of problem IPTSDA satisfying that $\hat{\xi}_i(\hat{\pi}) \leq \xi_i(\pi)$ for each $i \in I'$.*

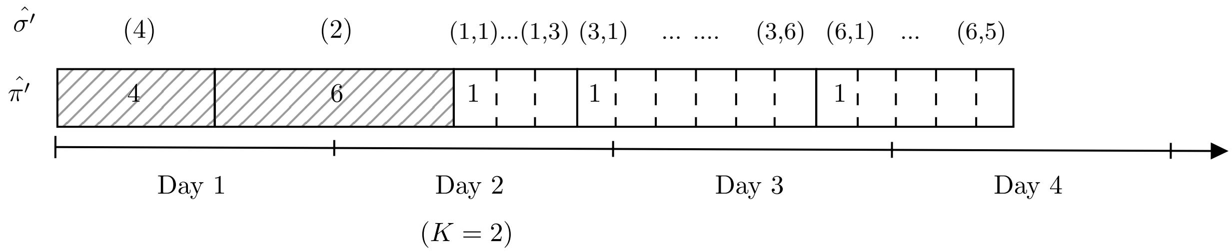
Proof. It can be seen that solution $\hat{\pi}$ of the new instance and the solution π of the original instance satisfies the following properties:

(i) For each accepted order in $N \setminus I'$, the production completion day and the shipped-out day of order i under $\hat{\pi}$ are the same as those under π;

Figure 2.6: Illustrative examples for the proof of Theorem 2.6 where $K = 2$ and $d_1 \le d_2 \le \ldots \le d_6$ where orders in $I' = \{1, 3, 6\}$ are split into unit orders $(1, 1), (1, 2), (1, 3), (3, 1), \cdots, (3, 6), (6, 1), \cdots, (6, 5)$.



(a) From σ^*, an order sequence $\hat{\sigma^*} = \{4, 2, (6, 1), (6, 2), \cdots, (6, 5), (1, 1), (1, 2), (1, 3), (3, 1), (3, 2), \cdots, (3, 6)\}$ is constructed for the new problem instance, and from $\hat{\sigma^*}$ a solution $\hat{\pi^*} = (\hat{\boldsymbol{\lambda}}(\hat{\sigma^*}), \hat{\mathbf{x}}(\hat{\sigma^*}), \hat{\mathbf{z}}(\hat{\sigma^*}))$ is constructed, in which four of the five unit orders split from order 6, and three of the six unit orders split from order 3 are shipped out one day earlier than the production completion days of order 6 and order 3, respectively.



(b) From σ', an order sequence $\hat{\sigma'} = \{4, 2, (1, 1), (1, 2), (1, 3), (3, 1), (3, 2), \cdots, (3, 6), (6, 1), (6, 2), \cdots, (6, 5)\}$ is constructed for the new problem instance, and from $\hat{\sigma'}$ a solution $\hat{\pi'} = (\hat{\boldsymbol{\lambda}}(\hat{\sigma'}), \hat{\mathbf{x}}(\hat{\sigma'}), \hat{\mathbf{z}}(\hat{\sigma'}))$ is constructed, in which one of the six unit orders split from order 3, and two of the five unit orders split from order 6 are shipped out one day earlier than the production completion days of orders 3 and 6, respectively.

(ii) For each $i \in I'$ and each $p \in \{1, 2, \cdots, q_i\}$, the unit order (i, p) is accepted under $\hat{\pi}$, and both the production completion day and the shipped-out day of the unit order (i, p) under $\hat{\pi}$ are the same as the day when the first p product units of order i are produced under π.

Due to (ii) above, for each unit order (i, p) split from order $i \in I'$, its product is shipped out as soon as it is produced, and the shipped-out day under $\hat{\pi}$ is no later than that of order i under π, which cannot be later than the committed delivery due date of order i. Thus, the shipped out day of each unit order (i, p) is no later than its committed delivery due date under $\hat{\pi}$. This, together with (i) above, implies that the solution $\hat{\pi}$ is feasible to the new instance of problem IPTSDA, and that the total cost of all the unit orders (i, p) under $\hat{\pi}$ cannot be greater than the cost of order i under π, i.e., $\hat{\xi}_i(\hat{\pi}) \le \xi_i(\pi)$ for $i \in I'$. Thus, Lemma 2.5 is proved. □

Applying Lemma 2.5 to order sequences σ^* and σ', we can obtain sequences $\hat{\sigma^*}$ and $\hat{\sigma'}$ for the new problem instance, respectively, as well as obtain feasible solutions $\hat{\pi^*} = (\hat{\boldsymbol{\lambda}}(\hat{\sigma^*}), \hat{\mathbf{x}}(\hat{\sigma^*}), \hat{\mathbf{z}}(\hat{\sigma^*}))$ and $\hat{\pi'} = (\hat{\boldsymbol{\lambda}}(\hat{\sigma'}), \hat{\mathbf{x}}(\hat{\sigma'}), \hat{\mathbf{z}}(\hat{\sigma'}))$ to the new problem instance, respectively, satisfying that

$$\hat{\xi}_i(\hat{\pi^*}) \leq \xi_i(\pi^*) \text{ and } \hat{\xi}_i(\hat{\pi'}) \leq \xi_i(\pi'), \text{ for } i \in I'. \tag{2.35}$$

Moreover, sequence $\hat{\sigma'}$ can also be transformed from sequence $\hat{\sigma^*}$ by repetitively interchanging the positions of any two unit orders (i, p) and (i', p') with $i > i'$ and (i, p) produced earlier than (i', p'), or with $i = i'$, $p > p'$, and (i, p) produced earlier than (i', p'), where $i \in I'$, $p \in \{1, 2, \cdots, p_i\}$, $i' \in I'$, and $p' \in \{1, 2, \cdots, p_{i'}\}$. Note that such two unit orders (i, p) and (i', p') have the same order quantity (which is one). By following an argument similar to that in the proof of Theorem 2.2, we can obtain that the total shipping cost of orders in I', under the solution constructed from the order sequence σ^*, is not increased after each interchange of the positions of orders (i, p) and (i', p'). Thus, we have $\sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi^*})$, which, together with (2.35), implies that

$$\sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi^*}) \leq \sum_{i \in I'} \xi_i(\pi^*). \tag{2.36}$$

Fourth, we are now going to investigate the difference between $\sum_{i \in I'} \xi_i(\pi')$ and $\sum_{i \in I'} \hat{\xi}_i(\hat{\pi'})$, that is, the difference between the total shipping cost of orders in I' under π' and that of unit orders split from orders in I' under $\hat{\pi'}$. For this, we establish Lemma 2.6 below.

Lemma 2.6. $\sum_{i \in I'} \xi_i(\pi') \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) + \epsilon \sum_{i \in I'} \xi_i(\pi^*)$.

Proof. If $m \leq \lceil 1/\epsilon \rceil$, i.e., m is bounded by a fixed constant $\lceil 1/\epsilon \rceil$, then $K = \min\{\lceil 1/\epsilon \rceil, m\} = m$. Thus, by definition, I' is empty, implying that $\sum_{i \in I'} \xi_i(\pi') = \sum_{i \in I'} \hat{\xi}_i(\hat{\pi'}) + \epsilon \sum_{i \in I'} \xi_i(\pi^*) = 0$. Lemma 2.6 holds true.

Otherwise, $m > \lceil 1/\epsilon \rceil$, and thus $K = \min\{\lceil 1/\epsilon \rceil, m\} = \lceil 1/\epsilon \rceil$. For each $i \in I'$, let τ_i indicate the shipped-out day of order i under solution π'. Since $q_i \leq c$, by the definitions of solutions π' and $\hat{\pi}'$, we can see that under $\hat{\pi}'$, each unit order (i,p) split from order i for $p \in \{1, 2, \cdots, q_i\}$ is accepted with its shipped-out day being equal to either $(\tau_i - 1)$ or τ_i. Thus, by $G(s,y) = y(\alpha - \beta s)$ in (2.1), we have

$$\xi_i(\pi') \leq \hat{\xi}_i(\hat{\pi}') + q_i\beta\{[d_i - (\tau_i - 1)] - (d_i - \tau_i)]\} = \hat{\xi}_i(\hat{\pi}') + \beta q_i, \text{ for } i \in I'. \tag{2.37}$$

Therefore, by (2.36) and (2.37) we obtain that

$$\sum_{i \in I'} \xi_i(\pi') \leq \sum_{i \in I'} [\hat{\xi}_i(\hat{\pi}') + \beta q_i] = \sum_{i \in I'} \hat{\xi}_i(\hat{\pi}') + \beta \sum_{i \in I'} q_i. \tag{2.38}$$

Since the orders in I' are all accepted and are shipped out after day K under π^*, the total shipping cost $\sum_{i \in I'} \xi_i(\pi^*)$ for these orders cannot be cheaper than $[\alpha - \beta(m-1-K)] \sum_{i \in I'} q_i$. Thus, since $\alpha - \beta(m-1) \geq 0$ stated in (2.2) implies that $\beta \leq [\alpha - \beta(m-1-K)]/K$, we can obtain that

$$\beta \sum_{i \in I'} q_i \leq \{[\alpha - \beta(m-1-K)]/K\} \sum_{i \in I'} q_i \leq [\sum_{i \in I'} \xi_i(\pi^*)]/K. \tag{2.39}$$

Therefore, by (2.38), (2.39), and $K = \lceil 1/\epsilon \rceil \geq 1/\epsilon$, we obtain that

$$\begin{aligned} \sum_{i \in I'} \xi_i(\pi') \leq \sum_{i \in I'} \hat{\xi}_i(\hat{\pi}') + \beta \sum_{i \in I'} q_i \leq & \quad \textstyle\sum_{i \in I'} \hat{\xi}_i(\hat{\pi}') + \sum_{i \in I'} \xi_i(\pi^*)/K \\ \leq & \quad \textstyle\sum_{i \in I'} \hat{\xi}_i(\hat{\pi}') + \epsilon \sum_{i \in I'} \xi_i(\pi^*), \end{aligned} \tag{2.40}$$

implying that Lemma 2.6 also holds true. This completes the proof of Lemma 2.6. □

We can now complete the proof of Theorem 2.6 as follows: From (2.34), (2.36), and

Lemma 2.6, we can prove that (2.30) holds true as follows:

$$\begin{aligned}\sum_{i\in N}\xi_i(\tilde{\pi}) &\leq \sum_{i\in N\setminus I'}\xi_i(\pi^*)+\sum_{i\in I'}\xi_i(\pi')\leq \sum_{i\in N\setminus I'}\xi_i(\pi^*)+\sum_{i\in I'}\hat{\xi}_i(\hat{\pi'})+\epsilon\sum_{i\in I'}\xi_i(\pi^*)\\ &\leq \sum_{i\in N\setminus I'}\xi_i(\pi^*)+\sum_{i\in I'}\xi_i(\pi^*)+\epsilon\sum_{i\in I'}\xi_i(\pi^*)\leq(1+\epsilon)\sum_{i\in N}\xi_i(\pi^*) \qquad (2.41)\end{aligned}$$

With (2.30) proved and Lemma 2.4 established, as we have explained earlier, Algorithm 2.4 must have a worst-case performance ratio of $(1+\epsilon)$ for any given $\epsilon>0$. This, together with Lemma 2.3, implies that Algorithm 2.4 is a pseudo-polynomial time approximation scheme for problem IPTSDA with a worst-case performance ratio of $(1+\epsilon)$ for any given $\epsilon>0$. Hence, Theorem 2.6 is proved.

Moreover, consider the case where $K=\min\{m,\lceil 1/\epsilon\rceil\}=m$. We know that $m\leq\lceil 1/\epsilon\rceil$, i.e., m is bounded by the constant $\lceil 1/\epsilon\rceil$. Thus, by definition, in the restricted problem $R(K,Q)$, there exists a constant $\bar{Q}'$ such that I' is empty when $Q=\bar{Q}'$. This, together with (2.34), (2.36), and Lemma 2.6, implies that $\sum_{i\in N}\xi_i(\tilde{\pi})\leq\sum_{i\in N\setminus I'}\xi_i(\pi^*)=\sum_{i\in N}\xi_i(\pi^*)$. Since π^* is an optimal solution to problem IPTSDA, $\tilde{\pi}$, as well as the solution returned by Algorithm 2.4, must also be so.

2.5.3 Extension

Algorithm 2.4 can be directly applied to solve more general problems, such as those with shipping cost functions $G(s,y)$ that are linearly non-decreasing in y and convexly non-increasing in s. In the proof of the worst-case performance ratio of $(1+\epsilon)$, we utilize the linearity of $G(s,y)$ in y only when proving (2.36), (2.37), (2.39), and (2.40) in Section 2.5.2.4. However, this situation can be extended to a more general case where $G(s,y)$ is linear in y and convexly non-increasing in s, and $G(s,y)$ can be represented by a piecewise linear function in s. Without loss of generality, we assume that the first linear piece of $G(s,y)$ is $y(\alpha-\beta s)$. For this more general case, by an argument similar to that in Section 2.3 for extending the proof of Theorem 2.2, we can obtain that (2.36) in Section 2.5.2.4 still holds. Since $G(s,y)$

is convexly non-increasing in s, which implies that $G(s, y)$ is continuous in s, we can obtain that $G(s-1, y) \leq G(s, y) + \beta y$. Thus, (2.37) of Section 2.5.2.4 still holds. Moreover, if $\alpha - (m-1)\beta \geq -\kappa\beta$ is satisfied for some integer constant $\kappa \geq 0$, then (2.39) of Section 2.5.2.4 can be extended to

$$\beta \sum_{i \in I'} q_i \leq \{[\alpha - \beta(m-1-K)]/(K-\kappa)\} \sum_{i \in I'} q_i \leq [\sum_{i \in I'} z_i(\pi^*)]/(K-\kappa). \tag{2.42}$$

We can modify Algorithm 2.4 by changing the value of K from $\lceil 1/\epsilon \rceil$ to $\lceil 1/\epsilon \rceil + \kappa$, so that (2.40) of Section 2.5.2.4 with K replaced by $(K-\kappa)$ holds. Thus, under the condition that κ is an integer constant, the modified Algorithm 2.4 with $K = \lceil 1/\epsilon \rceil + \kappa$ is a pseudo-polynomial time approximation scheme with a worst-performance ratio of $(1+\epsilon)$ for the more general case where $G(s, y)$ is linearly non-decreasing in y and convexly piecewise non-increasing in s.

2.6 Computational Experiments

We report on the computational experiments carried out to test the performance of the three newly proposed algorithms over randomly generated instances under two settings, with order acceptance decisions not taken into account and with order acceptance decisions taken into account, respectively. The experiments run on a PC with an Intel(R) Core(TM) i7-7700 3.60-GHz CPU and 32 GB of RAM, and the algorithms are in C++.

For each combination of the number of orders n and the length of planning horizon m, given a set E of all possible order quantities, we generate ten instances randomly in the following way.

(i) Order quantity: Each q_i is an integer randomly drawn from E.

(ii) Committed delivery due date: Each d_i is an integer randomly drawn from $\{1, \ldots, m\}$.

(iii) In the instances without order acceptance decisions, production capacity, c is an integer randomly drawn from $\{c_{\min}, c_{\min}+1, \ldots, c_{\max}\}$ where $c_{\min} = \max_{t\in\{1,2,\ldots,m\}}\lceil \sum_{i\in N: d_i \le t} q_i/t\rceil$ and $c_{\max} = \lceil 1.1c_{\min}\rceil$, so as to ensure the existence of feasible solutions for problem IPTSDA.

(iv) In the instances with order acceptance decisions, production capacity, c is an integer randomly drawn from the interval $[0.9c_{\min}, 1.0c_{\min}]$, so as to ensure that order rejections occur in feasible solutions to problem IPTSDA.

(v) Values of α and β: β is an integer randomly drawn from $\{1, 2, \ldots, 5\}$ and α is an integer randomly drawn from $\{(m-1)\beta+1, \ldots, 2(m-1)\beta\}$, so as to satisfy condition (2.2).

(vi) Rejection cost: Each $r_i = \alpha_i q_i$, where α_i is an integer randomly drawn from the interval $[\alpha, 2\alpha]$, so that the order's rejection cost increases in the order quantity, being consistent with the observation in common practice.

For the first algorithm (Algorithm 2.1), by Theorem 2.3 it is an exact algorithm that runs in polynomial or pseudo-polynomial time when η is a constant. Under the setting where order acceptance decisions are not considered and the setting where order acceptance decisions are considered, we conduct experiments for different values η chosen from $\{5, 6, 7\}$ and from $\{1, 2, 3\}$, respectively. or each value of η, we generate E by randomly selecting η elements from $\{1, 2, \cdots, 10\}$. In this way we generate ten random instances for each pair of $m \in \{5, 10, 15\}$ and $n \in \{40, 80, 120, 160, 200\}$ and for each value of η. Under the setting where order acceptance decisions are not taken into account, our computational results show that Algorithm 2.1 with setting $\hat{p}_k = 0$ for each $k \in \{1, 2, \ldots, \eta\}$ in (2.12) can solve all the instances with $\eta = 5$ in 5.0 seconds each. For instances with $\eta = 6$ and $n = 200$ and instances with $\eta = 7$ and $n \ge 160$, our computer does not have sufficient memory to execute Algorithm 2.1, since the space complexity of Algorithm 2.1 grows exponentially with η. Under the setting where order acceptance decisions are taken into account, our computational results show that Algorithm 2.1 can solve all the instances with $\eta = 1$ and

$\eta = 2$ in 6.6 seconds each. For instances with $\eta = 3$ and $n \geq 120$, our computer does not have sufficient memory to execute Algorithm 2.1. These results indicate that Algorithm 2.1 is efficient only when η is small.

For the second algorithm (Algorithm 2.2), by Theorem 2.4 it is an exact algorithm that runs in pseudo-polynomial time when m is a constant. We conduct experiments for different values of $m \in \{2, 3, 4\}$. Given $E = \{1, \ldots, 10\}$, we generate ten random instances for each pair of $m \in \{2, 3, 4\}$ and $n \in \{40, 80, 120, 160, 200\}$. Under the setting where order acceptance decisions are not taken into account, our computational results show that Algorithm 2.2 with $Q_m = \sum_{j=1}^{i} q_i - \sum_{t=1}^{m-1} Q_t$ in (2.15) can solve all the instances with $m \in \{2, 3\}$ in 0.6 second each. For instances with $m = 4$ and $n \geq 80$, our computer does not have sufficient memory to execute Algorithm 2.2, since the space complexity of Algorithm 2.2 grows exponentially with m. Under the setting where order acceptance decisions are taken into account, our computational results show that Algorithm 2.2 can solve all the instances with $m = 2$ in 1.2 second each. For instances with $m \geq 3$, our computer does not have sufficient memory to execute Algorithm 2.2. These results indicate that Algorithm 2.2 is efficient only when m is small.

For the third algorithm (Algorithm 2.4), which is an approximation scheme, by Theorem 2.6 and Lemma 2.3 its worst-case performance ratio and running time depend on ϵ. In our experiment, we let ϵ be 100%. Given $E = \{1, \ldots, 10\}$, we generate ten instances for each pair of $m \in \{5, 10, 15\}$ and $n \in \{40, 80, 120, 160, 200\}$. For each instance, we obtain an approximation solution by Algorithm 2.4, as well as obtain a lower bound on the total cost of the optimal solution by solving its relaxation with orders allowed to be split (according to Corollary 2.1).

The computational results for Algorithm 2.4 are shown in Table 2.2. For each instance, we compute the optimality gap by $(ub - lb)/lb \times 100\%$, where lb denotes the obtained lower bound value, and ub denotes the total cost of the obtained approximation solution. In Table 2.2, for the ten instances of each pair m and n, columns "M_G" and "A_G" present

Table 2.2: Computational results for the approximation scheme.

		Without order acceptance decisions				With order acceptance decisions			
m	n	M_G(%)	A_G(%)	M_T(s)	A_T(s)	M_G(%)	A_G(%)	M_T(s)	A_T(s)
5	40	0.48	0.19	0.0	0.0	0.08	0.01	0.4	0.2
	80	0.76	0.14	0.0	0.0	0.00	0.00	3.8	2.3
	120	0.11	0.01	0.0	0.0	0.00	0.00	11.8	10.0
	160	0.24	0.02	0.0	0.0	0.00	0.00	39.7	31.3
	200	0.06	0.02	0.0	0.0	0.00	0.00	130.2	80.5
10	40	0.72	0.35	0.0	0.0	0.14	0.08	0.1	0.1
	80	0.44	0.13	0.0	0.0	0.04	0.01	2.8	1.0
	120	0.20	0.08	0.0	0.0	0.01	0.00	4.2	3.3
	160	0.19	0.06	0.0	0.0	0.02	0.00	18.7	10.7
	200	0.07	0.03	0.0	0.0	0.01	0.00	39.2	26.3
15	40	0.82	0.52	0.0	0.0	0.41	0.22	0.0	0.0
	80	0.35	0.15	0.0	0.0	0.07	0.03	0.7	0.4
	120	0.30	0.13	0.0	0.0	0.03	0.01	2.4	1.6
	160	0.10	0.06	0.0	0.0	0.01	0.01	9.3	5.2
	200	0.14	0.08	0.0	0.0	0.03	0.01	23.6	12.7
average		0.33	0.13	0.0	0.0	0.06	0.03	19.1	12.4

the maximum and average optimality gaps, and columns “M_T” and “A_T” present the maximum and average running times in seconds. The results in Table 2.2 demonstrate that the approximation scheme in Algorithm 2.4 can produce close-to-optimal solutions for problem IPTSDA in short running times. For the setting where order acceptance decisions are not taken into account, the maximum (average) optimality gap is 0.33% (0.13%) and the maximum (average) running time is 0.0 (0.0) second. For the setting where order acceptance decisions are taken into account, the maximum (average) optimality gap is 0.06% (0.03%), and the maximum (average) running time is 19.1 (12.4) seconds. For all the instances, the maximum optimality gap is less than 1%. This indicates that for randomly generated instances, Algorithm 2.4 can produce solutions of significantly better quality than its worst-case guarantee, and thus has practical value.

2.7 Summary

In this chapter, we have studied problem IPTSDA, which is an integrated production and transportation scheduling problem with committed delivery due dates and with order acceptance decisions taken into account. This problem is commonly faced by make-to-order manufacturing companies under a commit-to-delivery business mode. A special case of problem IPTSDA is known to be strongly NP-hard, and it is thus unlikely that problem IPTSDA can be solved by any polynomial-time or pseudo-polynomial time exact algorithm, or any FPTAS, unless NP=P. We develop three algorithms, two of which are exact algorithms that can solve problem IPTSDA to optimality. We prove that these exact algorithms run in polynomial and pseudo-polynomial times for two practical cases: the case with a fixed number of possible order quantities, and the case with a fixed-length planning horizon. The other algorithm that we develop is a pseudo-polynomial time approximation scheme for solving problem IPTSDA, which guarantees a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$. According to our computational results, this approximation scheme also performs well in producing close-to-optimal solutions for problem instances that are randomly generated.

Chapter 3

Integrated Production and Transportation Scheduling with Committed Delivery Due Dates and Inventory Holding Cost

3.1 Introduction

In this chapter, we focus on a problem that integrates production and transportation scheduling (IPTS) which is always faced by a make-to-order manufacturer under a commit-to-delivery business mode. A make-to-order manufacturer would not produce products until receiving orders from customers. And the manufacturer needs to ship the products to customers before the committed delivery due date of an order and bear the shipping cost under the commit-to-delivery business mode. Typically, a manufacturer will use third-party logistics (3PL) to deliver the products to its customers. These 3PL companies can provide multiple shipping modes with different shipping times and shipping costs to the manufacturer. And it will take more costs to choose faster shipping modes. Also, during the production process of an order, some products that are completed for production but not shipped on the same day will be temporarily stored in a warehouse which incurs inventory holding costs. Hence, the manufacturer faces the problem of integrated production and transportation with committed delivery due dates and inventory holding costs, we refer to it as problem IPTSDI. Accordingly, for problem IPTSDI, the manufacturer needs to determine a production plan and a shipping plan while minimizing the total shipping costs and the total inventory holding costs with the committed delivery due dates of all orders satisfied.

In fact, the problem IPTSDI becomes more complex with incorporating the inventory holding costs into the objective function. Owing to the inventory holding costs incurred by the completed products which are not shipped on the same day, the manufacturer wants to postpone the production as late as possible. However, imposed by the committed delivery due date of each order, this production and shipping policy would lead to an increase in the shipping cost due to the decrease of transportation time. Therefore, the manufacturer needs to subtly balance the shipping costs and inventory holding costs when determining a production plan and a shipping plan.

The main contributions of the research in this chapter can be summarized as follow. First, with a relaxed problem of IPTSDI that focuses on deciding the daily aggregate production quantity and shipping quantity for each order, we develop a backward-forward algorithm that constructs an optimal solution to problem IPTSDI given a shipping plan and we find several properties held by the optimal solution. Second, for the case when the number of possible order quantities is bounded by a constant and the case when the planning horizon is bounded by a constant, we separately propose two pseudo-polynomial time exact algorithms. Third, we analyze the complexity of the problem IPTSDI when the unit inventory holding cost goes to infinity, i.e, no inventory is allowed during the production and shipping procedures. And we prove that there is no finite ratio pseudo-polynomial time approximation algorithm in this case. Fourth, by extending the second exact algorithm that solves the problem IPTSDI for the case when the planning horizon is fixed, we also establish a pseudo-polynomial time approximation algorithm that can solve problem IPTSDI with a worst-case performance ratio of $(1 + \epsilon)$ where $\epsilon > 0$.

The remainder of this chapter proceeds as follows: Section 3.2 describes the problem formally and Section 3.3 discusses the optimality properties of this problem. Section 3.4 examines the two exact algorithms and Section 3.5 shows the complexity of problem IPTSDI when the unit inventory holding cost goes to infinity. Section 3.6 examines the approximation scheme. Finally, we report the results of the computational experiments for the three algorithms in Section 3.7 and summarize this chapter in Section 3.8.

3.2 Problem Description and Formulation

We extend the settings described for problem IPTSD in Zhong et al. (2010) to formulate problem IPTSDI studied in this paper. The planning horizon of the manufacturer is m days and let $T = \{1, 2, \ldots, m\}$ to be the set of days. Before the start of the planning horizon, an order set $N = \{1, 2, \ldots, n\}$ arrives to the manufacturer from n different customers to produce certain products. Each order i is associated with an order quantity q_i representing the number of products needed to be produced and a committed delivery due date d_i and d_i is an integer with $1 \leq d_i \leq m$. This means that the customer should receive the completed products before or on the committed delivery due date. The manufacturer should produce the products on a single production line with a daily production capacity to be c. In other words, the total quantities of products produced in a day should not exceed c. Following the assumptions in Zhong et al. (2010), all these products are identical in unit weight and consumption of production capacity. And every order quantity is also less or equal to the daily production capacity, i.e., $q_i \leq c$ for $i \in N$. In fact, these assumptions are commonly seen in industries, for example, the computer manufacturing industry.

We also assume that 3PL company is used to deliver the completed products to its customers by the manufacturer. The 3PL company would pick up and ship these completed products at the end of each day. We also assume that all products of an order should be shipped out on the same day, that is, no partial deliveries are allowed for each order. The 3PL also adopts multiple shipping modes that are associated with the shipping days (shipping times). To be consistent with Zhong et al. (2010), we also adopt a linear shipping cost function $G(s, y)$:

$$G(s, y) = y(\alpha - \beta s), \tag{3.1}$$

where $\alpha > 0, \beta > 0$ and satisfy (3.2) to keep the $G(s, y)$ positive even with the $(m-1)$-day shipping mode.

$$\alpha - \beta(m-1) > 0. \tag{3.2}$$

Moreover, for each order $i \in N$, products that are processed completely but not shipped on the same day incur inventory holding costs and h represents the inventory holding cost per unit per day. Following the logic in Li et al. (2020), the inventory holding costs are essentially the opportunity costs of the expense associated with inventories of the product, which may often be included in the shipping costs. Hence, if the inventory holding time of one unit of product in order i is at least $1/\rho$ day during its transportation where $\rho \geq 1$, an inventory holding cost h/ρ should be included in the shipping cost. Hence, for all products in order i with order quantity q_i and shipping time t, the total cost for processing it can be written as: $G(t, q_i) + \frac{h}{\rho} \cdot q_i s = q_i[\alpha - (\beta - \frac{h}{\rho})t)]$. Therefore, the term $\beta - \frac{h}{\rho}$ can be treated as an updated parameter for β in the shipping cost function. Therefore, with $\beta - \frac{h}{\rho} \geq 0$, we can assume that $h \leq \rho\beta$.

A *solution* to problem IPTSDI contains two elements: *a production plan* and *a shipping plan*. A production plan is the daily production quantity of products that should be produced for each order and a shipping plan is the time for the 3PL to ship the completed products of each order to the customers. Accordingly, problem IPTSDI aims to decide a *feasible solution* such that customers can receive all the products in their orders no later than their committed delivery due dates. And problem IPTSDI aims to find a feasible solution with minimal total operating cost. Such a feasible solution is referred to as an *optimal solution* to problem IPTSDI.

Although the total operating costs include the costs of production and shipping as well as inventory holding costs, we do not consider the production cost in this study. The reason is that the manufacturing would cost-efficient to produce $\sum_{i\in N} q_i$ units of the products in total. Thus, the total production cost is always a constant that equals $\sum_{i\in N} q_i$ times a unit production cost, and this does not need to be considered.

For each $t \in T$, define $S_t = \{i \in N : d_i \leq t\}$ as the subset of orders for which products must be produced and shipped no later than t so as to meet their committed delivery due dates. Following Zhong et al. (2010), we also assume that the production capacity is sufficient enough for condition (3.3) below to be satisfied to ensure that a feasible solution always exists:

$$\sum_{i\in S_t} q_i \leq c \cdot t, \text{ for } t \in T. \tag{3.3}$$

Let $\mathbb{Z}_+$ denote a set of non-negative integers, we introduce a decision variable $x_{it} \in \mathbb{Z}_+$ to represent the number of the products manufactured for order i on day t. Therefore, for all $i \in N$ and $t \in T$ x_{it} represent a production plan. We also introduce a binary decision variable $z_{it} \in \{0, 1\}$ which is 1 when the 3PL ships order i on day t and is 0 otherwise. And for all $i \in N$ and $t \in T$ z_{it} represent a shipping plan. Since the shipping cost function in (3.1) is decreasing in shipping time, it would be cost-efficient to ship the products of an order such that the orders arrive to the customer on its delivery due date. In other words, if order i is shipped out on day t, which means $z_{it} = 1$, we can then denote incurred cost of shipping for order i as $G(d_i - t, q_i) = q_i[\alpha - \beta(d_i - t)]$.

The notations for problem IPTSDI described above are shown in Table 3.1. Accordingly,

Table 3.1: Notations for problem IPTSDI.

m	Size of the planning horizon
$T = \{1, 2, \ldots, m\}$	Set of days in the planning horizon
n	Size of order set
$N = \{1, 2, \ldots, n\}$	Set of orders
c	Daily production capacity
q_i	Order quantity for order i
$d_i \in \{1, 2, \cdots, m\}$	Committed delivery due date for order i
S_t	Subset of orders for which products must be shipped out on or before day t
$G(s, y) = y(\alpha - \beta s)$	Shipping cost function
h	Unit inventory holding costs per day
$x_{it} \in \mathbb{Z}_+$	Production quantity for order i on day t
$z_{it} \in \{0, 1\}$	1, if the shipping day for order i is day t, and 0, otherwise

we can use an integer linear programming (ILP) model below to formulate problem IPTSDI.

$$\text{(ILP)} \quad \min \quad \sum_{i \in N} \sum_{t \in T} G(d_i - t, q_i) \cdot z_{it} + \sum_{i \in N} \sum_{t \in T} h \left(\sum_{j=1}^{t} x_{ij} - \sum_{j=1}^{t} z_{ij} q_i \right) \tag{3.4}$$

$$\text{s.t.} \quad \sum_{i \in N} x_{it} \leq c, \text{ for } t \in T, \tag{3.5}$$

$$\sum_{t \in T} x_{it} = q_i, \text{ for } i \in N, \tag{3.6}$$

$$\sum_{t=1}^{m} z_{it} = 1, \text{ for } i \in N, \tag{3.7}$$

$$\sum_{t=d_i+1}^{m} z_{it} = 0, \text{ for } i \in N, \tag{3.8}$$

$$\sum_{t'=1}^{t} q_i z_{it'} \leq \sum_{t'=1}^{t} x_{it'}, \text{ for } i \in N, t \in T, \tag{3.9}$$

$$x_{it} \in \mathbb{Z}_+, z_{it} \in \{0, 1\}, \text{ for } i \in N, \, t \in T. \tag{3.10}$$

In model ILP, (3.4) is the objective function that minimizes the total shipping costs and the inventory holding costs. Constraint (3.5) limits the daily production quantity to be less or equal to the production capacity c. Constraint (3.6) means that the manufacturer produces all the products in every order. Constraints (3.7) and (3.8) jointly secure that all the orders are shipped out and can arrive to the customers on or before the committed delivery date. Constraint (3.9) assure that the 3PL can only ship the products after they are completed. Constraints (3.10) are integral and binary constraints on decision variables x_{it} and z_{it}, respectively.

Moreover, for the ease of representation, let $\mathbf{x}$ denote the vector of variables x_{it} for $i \in N$ and $t \in T$ and let $\mathbf{z}$ denote the vector of variables z_{it} for $i \in N$ and $t \in T$. Then, we can use $\pi = (\mathbf{x}, \mathbf{z})$ to denote a solution to model ILP.

3.3 Optimality Properties

In this section, we propose a backward-forward algorithm that constructs a solution to model ILP given a shipping plan and show the properties held by the optimal solutions. This algorithm and the properties aid us to develop the exact and approximation algorithms. We first define a relaxed problem of IPTSDI by aggregating the production quantities of orders in a day into a daily production quantity and then formulate its corresponding model ILP-AG. Every feasible solution can be a lower bound of model ILP. We show that we can obtain an optimal solution to model ILP-AG by a backward process given an optimal shipping plan and convert it to a feasible solution to model ILP by a forward process. Accordingly, we show that these two constructed solutions have the same objective value, which means the solution obtained by the forward process is optimal to model ILP.

From the description above, we can define a relaxed problem of IPTSDI – problem IPTSDI-R. Similar to problem IPTSDI, a solution to problem IPTSDI-R is consist of an aggregate production plan and a shipping plan, where a shipping plan is the same as that of problem IPTSDI. An aggregate production plan is about the quantity of products that must be produced on each day. And an optimal solution to problem IPTSDI-R is the solution with minimal total shipping costs and inventory holding costs where customers can receive their ordered products on or before their committed delivery due dates. Let decision variable $\bar{x}_t \in \mathbb{Z}_+$ for $t \in T$ represent the production quantity on day t. In other words, $\bar{x}_t = \sum_{i\in N} x_{it}$ is an aggregate production quantity on day t. We can then formulate problem IPTSDI-R by the following integer linear programming model ILP-AG.

$$
\begin{aligned}
\text{(ILP-AG)} \quad \min \quad & \sum_{i\in N}\sum_{t\in T} G(d_i - t, q_i)\cdot z_{it} + \sum_{t\in T} h\left(\sum_{t'=1}^{t} \bar{x}_{t'} - \sum_{t'=1}^{t}\sum_{j\in N} z_{jt'}q_j\right) && (3.11)\\
\text{s.t.} \quad & \bar{x}_t \le c, \ \text{for } t \in T, && (3.12)\\
& \sum_{t\in T} \bar{x}_t = \sum_{i\in N} q_i, && (3.13)\\
& \sum_{t=1}^{m} z_{it} = 1, \ \text{for } i \in N, && (3.14)\\
& \sum_{t=d_i+1}^{m} z_{it} = 0, \ \text{for } i \in N, && (3.15)\\
& \sum_{t'=1}^{t}\sum_{i\in N} q_i z_{it} \le \sum_{t'=1}^{t} \bar{x}_{t'}, \ \text{for } t \in T, && (3.16)\\
& z_{it} \in \{0,1\}, \ \text{for } i \in N,\ t \in T, && (3.17)\\
& \bar{x}_t \in \mathbb{Z}_+, \ \text{for } t \in T. && (3.18)
\end{aligned}
$$

In model ILP-AG, (3.11) is the objective function that aims to minimize the total shipping costs and the inventory holding costs which is similar to the objective function (3.4) of model ILP. Constraint (3.12)-(3.15) are also the similar with the constraints (3.5)-(3.8) in model ILP. Constraint (3.16) denotes that the accumulated shipping quantity before or on day t is less or equal to the accumulated production quantity before on day t for each $t \in T$. Constraints (3.17) and (3.18) ensure decision variables $\bar{x}_t$ and z_{it} to be and binary and integral, respectively. Following the notations above, let $\bar{\mathbf{x}}$ be the vector of variables x_t for $t \in T$ as the daily aggregated production plan. Together with the shipping plan represented by the vector $\mathbf{z}$, a solution to model ILP-AG can be denoted as $(\bar{\mathbf{x}}, \mathbf{z})$.

We can obtain an order sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n)$ with each $\sigma_j \in N$ for $j \in$

$\{1, 2, \cdots, n\}$ indicating the j-th order of σ given a shipping plan $\mathbf{z}$ by sorting the shipping out days of these orders in a non-decreasing order, breaking ties arbitrarily. We can also obtain the corresponding sequence of shipping day $\omega = (\omega_{\sigma_1}, \omega_{\sigma_2}, \cdots, \omega_{\sigma_n})$ for orders in σ. From Yang et al. (2021), we know that in the optimal solution, the order is shipped out on its production completion day. Thus, ω_{σ_j} also denote the production completion day of order σ_j. With ω obtained from a shipping plan $\mathbf{z}$, we can construct the daily production plan $\bar{\mathbf{x}}$ for model ILP-AG and the constructed solution is referred to as solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$.

From Li et al. (2020), we know that to reduce the inventory holding cost, an order needs to be processed as late as possible. Therefore, we construct the production plan in a reverse order of ω, i.e., from ω_{σ_n} to ω_{σ_1}. In the following, we describe the construction process. We divide the backward construction process into 3 cases for each order σ_j for $j \in \{1, 2, \cdots, n\}$ and initialize $\bar{x}_{t'} \leftarrow 0$ for all $t' \in \{1, 2, \cdots, m\}$ and $t = m$:

1. In the first case, the daily production quantity on day t is less or equal to production capacity after producing all the products in order σ_j and the production completion time of order σ_j is also on day t. We then increase the production quantity on day ω_{σ_j} (i.e., day t) with q_{σ_j};

2. In the second case, the daily production quantity on day t exceeds the production capacity after producing all the products in order σ_j and the production completion time of order σ_j is on day t. It means that the production quantity of order σ_j needs to be split into two days. And the split two production days are consecutive. The reason is that if they are not consecutive, compared with the case for consecutive production days, extra inventory holding costs in the split days will be incurred while the shipping costs are the same. Then, we set the production quantity on day $t - 1$ to be the exceeded quantity over the daily production capacity and set the production quantity on day t as the production capacity c and update t accordingly;

3. In the third case, the production completion day is on the previous day. This means

that there is an idle time between the production completion day of order σ_{j-1} and the production start day of order σ_j. We then set the production quantity on day ω_{σ_j} to be q_{σ_j} and update t accordingly;

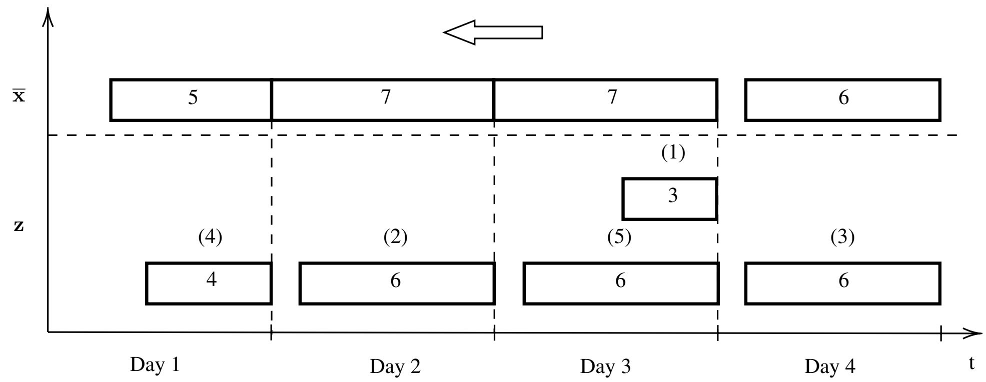
We summarize the process of backward construction described above in Algorithm 3.1.

Algorithm 3.1 (Backward construction)

```
Initialize t ← m and x̄_{t'} ← 0 for all t' ∈ {1, 2, ···, m}
for each j = n, n−1, ···, 1 do
    if x_{ω_{σ_j}} + q_{σ_j} ≤ c and t = ω_{σ_j} then
        x_{ω_{σ_j}} ← x_{ω_{σ_j}} + q_{σ_j}
    else if x_{ω_{σ_j}} + q_{σ_j} > c and t = ω_{σ_j} then
        x_{ω_{σ_j}−1} ← x_{ω_{σ_j}} + q_{σ_j} − c, x_{ω_{σ_j}} ← c
        t ← ω_{σ_j} − 1
    else if t > ω_{σ_j} then
        x_{ω_{σ_j}} ← q_{σ_j}
        t ← ω_{σ_j}
    end if
end for
return x̄_t for all t ∈ {1, 2, ···, m}
```

Consider an example showing in Figure 3.1 for illustration of Algorithm 3.1. The manufacturer has a planning horizon of 4 days with a daily production capacity to be $c = 7$. From the shipping plan $\mathbf{z}$, we can obtain an order sequence $\sigma = (4, 2, 5, 1, 3)$ and its corresponding shipping day is $\omega = (1, 2, 3, 3, 4)$. With these, Algorithm 3.1 can construct a daily production plan $\bar{\mathbf{x}}(\mathbf{z})$ in a reverse order of ω. The construction process can be describe as follow: For order 3, since the production quantity on day 4 is initialized to be 0 and $t = \omega_3 = 4$ which is in accord with case 1, we set $x_4 = q_3 = 6$; next, for order 1, its production completion day is day $3 < t = 4$, which is in case 3, we set $x_3 = 3$ and update $t = 3$; consequently, for order 5, the remaining production capacity on day 3 is not enough and the production completion is also day 3, which is in case 2, we set $x_2 = q_5 + x_3 - 7 = 2$ and update $x_3 = c = 7$ and update $t = 2$. Accordingly, we can update $x_2 = 7, x_1 = 5$ following the same process.

Figure 3.1: Examples of a solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ constructed by Algorithm 3.1 given a shipping plan $\mathbf{z}$.



From the construction process in Algorithm 3.1, we can establish Lemma 3.1 below, showing that the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ satisfies all the constraints in model ILP-AG except for constraints (3.14) and (3.15).

Lemma 3.1. *For any shipping plan* $\mathbf{z}$*, solution* $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ *satisfies constraints (3.12), (3.13), and (3.16), (3.17) (3.18) of model ILP-AG.*

Proof. By the construction process in Algorithm 3.1, the daily production quantity of is less or equal to c, and thus constraint (3.12) of model ILP-AG is satisfied. From the divided 3 cases of the above construction of $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$, we can see that for each $j \in \{1, 2, \cdots, n\}$, q_{σ_j} has been allocated on certain day on certain consecutive two days, implying that constraint (3.13) of model ILP-AG is satisfied.

Moreover, as a consequence that solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ constructed Algorithm 3.1 follows the reverse order of a sequence of completion day ω and each order is delivered on day when their production is completed. Thus, constraint (3.16) of model ILP-AG is satisfied.

From the construction, we can also see that for $t \in T$ x_t are all integers. And according to the definition of a shipping plan, for $i \in N$ and $t \in T$ $z_{it} \in \{0, 1\}$. Therefore, constraint (3.17) and (3.18) of model ILP-AG is satisfied. This completes the proof of Lemma 3.1. $\square$

Based on Lemma 3.1, we can further prove that a shipping plan $\mathbf{z}$ exists such that the constructed solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ is optimal to model ILP.

Lemma 3.2. *Consider any optimal solution to model ILP-AG denoted by* $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$*. Given the shipping plan* $\mathbf{z}^*$*, solution* $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ *obtained by Algorithm 3.1 is also optimal to model ILP-AG.*

Proof. By Lemma 3.1, solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ satisfies constraints (3.12), (3.13), and (3.16), (3.17) (3.18) of model ILP-AG. Consider the optimal solution $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$ to model ILP-AG, in which we know that all orders are shipped, i.e., constraint (3.14) is satisfied. Moreover, from the definition above, σ^* denotes the order sequence obtained from $\mathbf{z}^*$. Thus, we also know that products for order σ_j^* are not shipped out before any products for orders $\sigma_{j'}^*$ for $j' \in \{1, 2, \cdots, j-1\}$. Thus, in $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$, products for order σ_j^* are not shipped out before the products for the first j orders of σ^* are all produced. Thus, products for order σ_j^* must be shipped out on or after day ω_{σ_j}, implying that $\omega_{\sigma_j} \leq d_{\sigma_j^*}$. From the construction in Algorithm 3.1, we know that in $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$, products for each order are shipped out on the same day as their production is completed. Thus, constraint (3.15) of model ILP-AG is satisfied. Hence, $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is a feasible solution to model ILP-AG.

As shown above, for each $j \in \{1, 2, \cdots, n\}$, products for order σ_j^* are shipped out on or after day ω_{σ_j} in the optimal solution $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$, whereas they are shipped out on day ω_{σ_j} in the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$. Thus, since the shipping cost function $G(d_i - t, q_i)$ is non-decreasing in t, the total shipping costs of $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ cannot exceed that of $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$.

Next, we prove that the inventory holding costs of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ also cannot exceed that of $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$ by contradiction. Suppose the inventory holding costs of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is larger than that of $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$. Then, we can find two days τ and τ' such that on day τ the production quantity of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is $\bar{x}_\tau$ and the production quantity of solution $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$ is $\bar{x}_\tau^*$ and $c \geq \bar{x}_\tau > \bar{x}_\tau^*$; and on day τ' the production quantity of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is $\bar{x}_{\tau'}$ and the production quantity of solution $(\bar{\mathbf{x}}^*, \mathbf{z}^*)$ is $\bar{x}_{\tau'}^*$ and $\bar{x}_{\tau'} < \bar{x}_{\tau'}^* \leq c$. We can also find a day t with $\tau \leq t < \tau'$ such that the production quantity on day $t+1$ is less than c

(according to the construction process in Algorithm 3.1), which means that the inventory on day t is 0. In other words, the following constraint should be satisfied, total production quantity on day t equals total shipping quantity on this day, i.e.,

$$\sum_{t'=1}^{t} \bar{x}_{t'} = \sum_{t'=1}^{t} \bar{x}^*_{t'} = \sum_{t'=1}^{t} \sum_{i \in N} q_i z_{i,t-1}. \tag{3.19}$$

However, the left hand side of Equation (3.19) is violated owing to the inequality $\bar{x}_\tau > \bar{x}^*_\tau$. And this contradicts with the fact that solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is a feasible solution.

Hence, $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is also an optimal solution to model ILP-AG. Lemma 3.2 is proved. □

Based on the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ to model ILP-AG by Algorithm 3.1, we can further construct a solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ to model ILP.

The main idea is to construct the production plan for each order by allocating the daily production quantity while following the order sequence σ from the given shipping plan $\mathbf{z}$. Then, we divide the forward construction process into 2 cases for each order σ_j for $j \in \{1, 2, \cdots, n\}$ and initialize $t = 1$ and $x_{it'} = 0$ for all $i \in \{1, 2, \cdots, n\}$ and $t' \in \{1, 2, \cdots, m\}$:

1. In the first case, the daily production quantity on day t is enough to produce all products in order σ_j. We set the production quantity for order σ_j on day t to be q_{σ_j} and update the daily production quantity on day t accordingly. Specifically, if the remaining production quantity on day t drops to 0, we update t to be $t + 1$;

2. In the second case, the daily production quantity on day t is not enough to produce all products in order σ_j. We set the production quantity for order σ_j on day t to be the remaining daily production quantity on day t and set the production quantity for order σ_j on the next day to be the remaining unproduced quantity of products and update the remaining production quantity on day $t + 1$ and update t to be $t + 1$ accordingly.

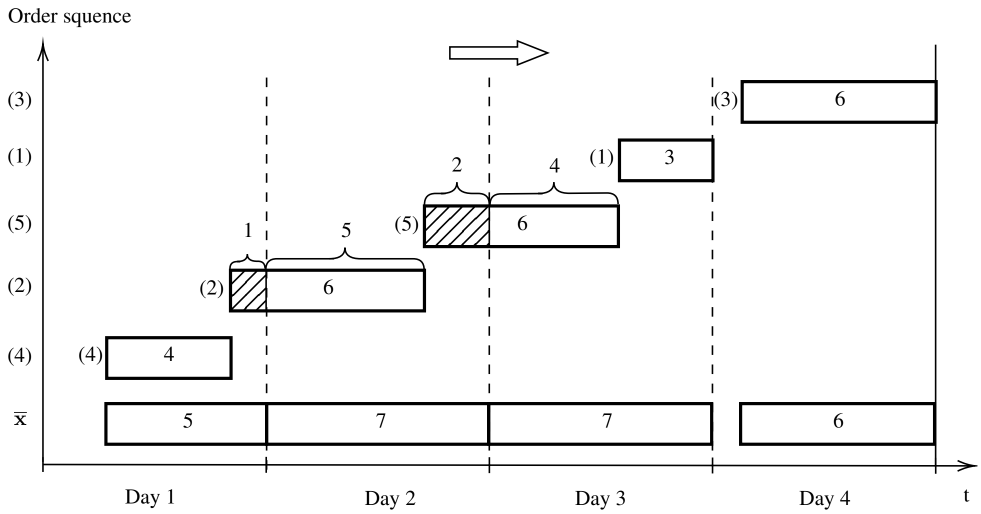
We can summarize the forward construction process in Algorithm 3.2.

Algorithm 3.2 (Forward construction)

Initialize $t \leftarrow 1$ and $x_{it'} \leftarrow 0$ for all $i \in \{1, 2, \cdots, n\}$ and $t' \in \{1, 2, \cdots, m\}$
for each $j = 1, 2, \cdots, n$ **do**
  if $x_t \geq q_{\sigma_j}$ **then**
    $x_{\sigma_j,t} \leftarrow q_{\sigma_j}$
    $x_t \leftarrow x_t - x_{\sigma_j,t}$
    if $x_t = 0$ **then**
      $t \leftarrow t + 1$
    end if
  else
    $x_{\sigma_j,t} \leftarrow x_t, x_{\sigma_j,t+1} \leftarrow q_{\sigma_j} - x_{\sigma_j,t}$
    $x_{t+1} \leftarrow x_{t+1} - x_{\sigma_j,t+1}$
    $t \leftarrow t + 1$
  end if
end for
return x_{it} for all $i \in \{1, 2, \cdots, n\}$ and $t \in \{1, 2, \cdots, m\}$

Following the example in Figure 3.1, we can consider an extended example showing in Figure 3.2 for illustration of Algorithm 3.2, where the order sequence $\sigma = (4, 2, 5, 1, 3)$. Follow Algorithm 3.2, we construct the production plan for each order with the order sequence σ. For order 4, the daily production quantity is enough to produce all products in order 4, which is for case 1. Thus, we set $x_{4,1} = q_4 = 4$ and update the production on day 1 x_1 from 5 to 1; next, for order 2, the remaining production on day 1 is not enough to produce all products, which is for case 2, we then set $x_{2,1} = 1$, which is the remaining production quantity on day 1 and $x_{2,2} = 5$, which is the remaining order quantity and update the production quantity on day 2 to be $x_2 = 7 - 5 = 2$ and update $t = 2$; consequently, for order 5, the remaining production on day 2 is not enough to produce all products, which is for case 2, we then set $x_{5,2} = 2$ and $x_{5,3} = 4$ and update the remaining production quantity on day 3 to be $x_3 = 7 - 4 = 3$ and update $t = 3$; next, for order 1, the remaining production quantity on day 3 is enough to produce all products of it, therefore, we set $x_{1,3} = 3$. Also, as the production quantity on day 3 drops to 0, we update $t = 4$; next, for order 3, the production quantity on day 4 is the same as the order quantity $q_3 = 6$, we set $x_{3,4} = 6$.

Figure 3.2: Examples for the forward construction of a solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ in Algorithm 3.2 given a daily production plan $\bar{\mathbf{x}}(\mathbf{z})$ and $\bar{\mathbf{x}}$ is the daily production plan from the backward construction.



Based on Algorithm 3.1 and Algorithm 3.2, we have Algorithm 3.3 that constructs a solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ to model ILP given a shipping plan $\mathbf{z}$.

Algorithm 3.3 (Construct a solution for problem IPTSDI)

1: Backward construction for $\bar{x}_t$ for all $t \in \{1, 2, \cdots, m\}$ by Algorithm 3.1

2: Forward construction for $x_{i,t}$ for all $i \in \{1, 2, \cdots, n\}$ and $t \in \{1, 2, \cdots, m\}$ with $\bar{x}_t$ for all $t \in \{1, 2, \cdots, m\}$ by Algorithm 3.2

3: **return** $x_{i,t}$ for all $i \in \{1, 2, \cdots, n\}$ and $t \in \{1, 2, \cdots, m\}$

And follow Lemma 3.1 and Lemma 3.2, we can establish Theorem 3.1 as follow, which indicates that given an optimal shipping plan $\mathbf{z}^*$ to model ILP, the constructed solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ by Algorithm 3.3 is also optimal to model ILP.

Theorem 3.1. *Consider any optimal solution to model ILP denoted by* $(\mathbf{x}^*, \mathbf{z}^*)$. *Given the shipping plan* $\mathbf{z}^*$, *solution* $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ *obtained by Algorithm 3.3 also forms an optimal solution to model ILP.*

Proof. By Lemma 3.2, we know that the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is an optimal solution to model ILP-AG. Also, we know that model ILP-AG is a relaxed model of model ILP, which indicates that the objective value of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is a lower bound to model ILP.

From the forward construction process in Algorithm 3.3, we can see that the constructed solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ is a feasible solution to model ILP. The optimality of solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ to model ILP-AG ensures constraints (3.7) (3.8) and (3.10) to be satisfied. Moreover, as solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ also satisfies constraints (3.12), (3.13) and (3.16), thus, the allocation of the daily production quantity to production quantity for each order on each day in step 4–5 and step 10–11 in Algorithm 3.2 ensures constraints (3.5), (3.6) and (3.9) to be satisfied. Thus, solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ is feasible to model ILP. Also, we can see that Algorithm 3.2 maintains the same inventory holding costs for solutions $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ and $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$, which is can be calculated in Algorithm 3.1. Therefore, the objective value from the solution $(\bar{\mathbf{x}}(\mathbf{z}^*), \mathbf{z}^*)$ is not changed in Algorithm 3.3. Therefore, solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ is an optimal solution to model ILP. Theorem 3.1 is proved. $\square$

From Theorem 3.1, we know that Algorithm 3.3 can also find the minimized inventory holding costs given a shipping plan with knowing the shipping quantity on each day. For each $t \in T$, let $Q_t = \sum_{i \in N} q_i z_{it}$ to be the total shipped out quantity of products on day t. And the vector $(Q_1, Q_2, \cdots, Q_m)$ is the combination of Q_t from day 1 to day m. According to constraint (3.9) of model ILP, the total shipped-out quantity of products on or before day t is less or equal to the total produced quantity during this period. Moreover, limited by constraint (3.5), the total produced quantity on or before day t should not be larger than the maximum production quantity tc during the first t days. Therefore, $(Q_1, Q_2, \cdots, Q_m)$ satisfies condition (3.20) below:

$$\sum_{t'=1}^{t} Q_{t'} \leq tc, \text{ for each } t \in T. \tag{3.20}$$

We can then establish Corollary 3.1 in the following,

Corollary 3.1. *For any shipping plan* $\mathbf{z}$*, consider any* $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ *with* $\sum_{i \in N} q_i z_{it} = Q_t$ *for* $t \in T$*,* $\sum_{t=1}^{d_i} z_{it} = 1$ *and* $\sum_{t=d_i+1}^{m} z_{it} = 0$ *for* $i \in N$ *and satisfies (3.20), we can obtain its corresponding minimal inventory holding costs* $H(Q_1, Q_2, \cdots, Q_m)$ *by Algorithm 3.3.*

Proof. From Theorem 3.1, we can infer that for any shipping plan $\mathbf{z}$, solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ constructed by Algorithm 3.3 is the solution with minimal shipping costs and inventory holding costs. Also, in step 2 of Algorithm 3.3, i.e., Algorithm 3.2, we can iteratively calculate the minimal inventory holding cost. From the description of Algorithm 3.2, we know that inventory is only incurred in the second case of the Algorithm. And, for each order x_{σ_j}, for $j \in \{1, 2, \cdots, n\}$, the productions (x_{σ_j}, t) and $(x_{\sigma_j}, t+1)$ of the order in step 10. Thus, the inventory holding costs incurred for order x_{σ_j} can be calculated as $h_{\sigma_j} = h \cdot x_{\sigma_j}$. And we can obtain the total inventory holding costs at the end of Algorithm 3.2 by adding this step after step 10, i.e., $H(Q_1, Q_2, \cdots, Q_m)$ with the given shipping plan $\mathbf{z}$. Therefore, Corollary 3.1 is proved. □

Based on the analysis in Theorem 3.1, we can have Theorem 3.2. It shows that a shipping plan $\mathbf{z}$ always exists such that: i) the constructed solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ is optimal to model ILP; ii) in the optimal solution, for orders with the same order quantity, they are sorted in a non-decreasing order by the committed delivery due dates, breaking ties by preferring smaller indices.

Theorem 3.2. *There exists a shipping plan* $\mathbf{z}$ *with its order sequence* $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n)$ *such that (i)* $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ *is an optimal solution to model ILP, and that (ii)* $d_{\sigma_j} < d_{\sigma_h}$ *or* $(d_{\sigma_j} = d_{\sigma_h}$ *and* $\sigma_j < \sigma_h)$*, for each* j *and* h *with* $1 \leq j < h \leq n$ *and* $q_j = q_h$*.*

Proof. We can show that condition (i) is satisfied by Theorem 3.1, which shows that a shipping plan $\mathbf{z}^*$ always exists such that solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ is optimal to model ILP. However, if there is no σ^* exists to satisfy condition (ii), which means that there exist j and h in $\{1, 2, \cdots, n\}$ with $j < h$ and $q_{\sigma_j^*} = q_{\sigma_h^*} = q$ for some q such that $d_{\sigma_j^*} > d_{\sigma_h^*}$ or $(d_{\sigma_j^*} = d_{\sigma_h^*}$ and

$\sigma^*_j > \sigma^*_h$). In this situation, we can swap positions of σ^*_j and σ^*_h in σ^* to obtain a new order sequence σ, so that condition (ii) specified in Theorem 3.2 is satisfied for j and h. Moreover, we can also swap values of $x^*_{\sigma^*_j,t}$ and $x^*_{\sigma^*_h,t}$ for $t \in T$, and swap values of $z^*_{\sigma^*_j,t}$ and $z^*_{\sigma^*_h,t}$ for $t \in T$, to obtain a new solution $(\mathbf{x}, \mathbf{z})$, which, as shown below, is also an optimal solution to model ILP.

Let $i = \sigma^*_j$ and $i' = \sigma^*_h$. For each $i'' \in N$, let $\tau_{i''}$ and $\tau^*_{i''}$ denote the shipping day of order i'', in the two solutions $(\mathbf{x}, \mathbf{z})$ and $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$, respectively. Thus, we have that $\tau_{i'} = \tau^*_i$ and $\tau_i = \tau^*_{i'}$. From the procedure to construct the optimal solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ in Algorithm 3.3, since $j < h$, we know that $\tau^*_i \leq \tau^*_{i'}$. These, together with $\tau^*_i \leq d_i$, $d_{i'} \leq d_i$, and $\tau^*_{i'} \leq d_{i'}$, imply that $\tau_i = \tau^*_{i'} \leq d_{i'} \leq d_i$, and that $\tau_{i'} = \tau^*_i \leq \tau^*_{i'} \leq d_{i'}$. Thus, $(\mathbf{x}, \mathbf{z})$ satisfies constraints (3.7) and (3.8) of model ILP. From $q_i = q_{i'}$ we know that $(\mathbf{x}, \mathbf{z})$ satisfies constraints (3.5), (3.6), (3.9), and (3.10) of model ILP, implying that $(\mathbf{x}, \mathbf{z})$ is a feasible solution to model ILP. By (3.21) below, we can also see that the shipping costs of order i and order i' are the same under $(\mathbf{x}, \mathbf{z})$ and $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$:

$$\begin{aligned}
&G(d_i - \tau_i, q) + G(d_{i'} - \tau_{i'}, q) = q(\alpha - \beta(d_i - \tau_i)) + q(\alpha - \beta(d_{i'} - \tau_{i'})) \\
&= q(\alpha - \beta(d_i - \tau_{i'})) + q(\alpha - \beta(d_{i'} - \tau_i)) = q(\alpha - \beta(d_i - \tau^*_i)) + q(\alpha - \beta(d_{i'} - \tau^*_{i'})) \\
&= G(d_i - \tau^*_i, q) + G(d_{i'} - \tau^*_{i'}, q). \qquad (3.21)
\end{aligned}$$

Thus, the total shipping costs of $(\mathbf{x}, \mathbf{z})$ equal that of the optimal solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$. Moreover, as a consequence that $q_{\sigma^*_j} = q_{\sigma^*_h}$ and changes in the order sequence only influence the operations in Algorithm 3.2. Therefore, with similar argument in Theorem 3.1, inventory holding costs of $(\mathbf{x}, \mathbf{z})$ is also the same as the optimal solution $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ according to Algorithm 3.3. Therefore, $(\mathbf{x}, \mathbf{z})$ is also an optimal solution to model ILP. Since σ swaps only the positions of i and i' in σ^*, which means that $(\mathbf{x}, \mathbf{z})$ is equal to $(\mathbf{x}(\mathbf{z}), \mathbf{z})$.

Hence, by replacing σ^* with σ and repeating the process above iteratively, an order sequence σ meeting (i) and (ii) stated in Theorem 3.2 can be constructed. This completes the proof of Theorem 3.2. □

3.4 Exact Algorithms

In this section, we examine two exact algorithms for problem IPTSDI. They can obtain optimal solutions and run in pseudo-polynomial times for the two practical cases for this problem.

3.4.1 Exact Algorithm When the Number of Possible Order Quantities is Fixed

We follow the settings in Li et al. (2022) in this case. Let set $E = \{q_i | i \in N\}$ represent all distinct order quantities in the order set N and η denote the number of possible order quantities in the order set N. According to the description, we have $\eta = |E|$. For the ease of representation, we write $E = \{e_1, \cdots, e_\eta\}$. Furthermore, we define subsets of order set N: for every $k \in \{1, 2, \ldots, \eta\}$, denote $N_k = \{i \mid q_i = e_k, i \in N\}$ as the set of orders whose order quantity is exactly e_k, and denote the number of orders in N_k as n_k. For instance, if the order set received by the manufacturer is $N = \{1, 2, 3, 4, 5\}$ where $q_1 = q_2 = 20, q_3 = q_4 = q_5 = 30$. Then $E = \{20, 30\}$ with $\eta = 2$ and $N_1 = \{1, 2\}$ and $N_2 = \{3, 4, 5\}$. From the example, we can also see that set N_k for all $k \in \{1, 2, \ldots, \eta\}$ form a partition of N. For better description of the algorithm, for each $k \in \{1, 2, \ldots, \eta\}$, let $i(k, 1), i(k, 2), \ldots, i(k, n_k)$ denote the indices of n_k orders in the subset N_k. Moreover, we also assume that $d_{i(k,1)} \leq d_{i(k,2)} \leq \ldots \leq d_{i(k,n_k)}$. That is, these n_k orders are indexed in a non-decreasing order of their committed delivery due dates.

Following Li et al. (2022), for each $(p_1, p_2, \cdots, p_\eta)$ with $p_k \in \{0, 1, \cdots, n_k\}$ for $k \in \{1, 2, \cdots, \eta\}$, we define the order set $N(p_1, p_2, \ldots, p_\eta) = \{i(k, r) \mid 0 \leq r \leq p_k, 1 \leq k \leq \eta\}$. It combines first p_k orders in each subset N_k for every $k \in \{1, 2, \ldots, \eta\}$.

The first exact algorithm utilizes Theorem 3.2 to solve problem IPTSDI. We describe this dynamic programming algorithm as follows.

Let $F(p_1, p_2, \ldots, p_\eta; t)$ represent the value function of the minimum total shipping costs and inventory holding costs of a subproblem of problem IPTSDI defined for only orders in $N(p_1, p_2, \ldots, p_\eta)$, which equals $+\infty$ if the subproblem has no feasible solution and the shipping day of the last order in the order set $N(p_1, p_2, \ldots, p_\eta)$ is day t. In other words, it indicates that, for all orders in $N(p_1, p_2, \ldots, p_\eta)$, their shipping days are on or before day t. Accordingly, $F(n_1, n_2, \cdots, n_\eta; m)$ indicates the minimum total shipping costs and inventory holding costs of problem IPTSDI, which is defined for all orders in N.

Particularly, similar to the definition of $N(p_1, p_2, \ldots, p_\eta)$, for each $(p'_1, p'_2, \cdots, p'_\eta)$ with $p'_k \in \{0, 1, \cdots, n_k\}$ and $p'_k + j_k \in \{1, \cdots, n_k\}$ for $k \in \{1, 2, \cdots, \eta\}$, we also define the set

$$N'((p'_1, j_1), \cdots, (p'_\eta, j_\eta)) = \{i(k, r) \mid p'_k < r \le p'_k + j_k, 1 \le j \le \eta\}.$$

It combines j_k orders, i.e., $i(k, p'_k + 1), \ldots, i(k, p'_k + j_k)$, of each subset N_k for every $k \in \{1, 2, \cdots, \eta\}$. Based on the description above, we refer to a order set $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ associated with the production start day of its first order and the production completion day of its last order as a *production subsequence*. From Section 3.3, we know that it is optimal to ship out the orders on the production completion day. Therefore, in each production subsequence, no inventory exists at end of its production completion day. Following the description above, we can see that orders in $N(p_1, p_2, \ldots, p_\eta)$ can be split into several distinct production subsequences according to their production start day and production completion day. Suppose $N(p_1, p_2, \ldots, p_\eta)$ can be split into λ production subsequences, we then have the equation,

$$\begin{aligned} N(p_1, p_2, \ldots, p_\eta) =& N'((0, j_{0,1}), \cdots, (0, j_{0,\eta})) \cup N'((j_{0,1}, j_{1,1}), \cdots, (j_{0,\eta}, j_{1,\eta})) \\ & \cup \ldots \cup N'\left(\left(\sum_{l=0}^{\lambda-1} j_{l,1}, j_{\lambda,1}\right), \cdots, \left(\sum_{l=0}^{\lambda-1} j_{l,\eta}, j_{\lambda,\eta}\right)\right), \\ & \forall k \in \{1, \cdots, \eta\},\ \textstyle\sum_{l=0}^{\lambda} j_{l,k} = p_k. \end{aligned} \tag{3.22}$$

Specifically, when $p_k = n_k$, for each $k \in \{1, \cdots, \eta\}$ and $\lambda = \lambda'$, it indicates that the total order set can be split into λ' distinct production subsequences according to their shipping days, where the inventory is 0 at the end of shipping day of the last order in each production production subsequence.

Based on the description above, we can define another value function $I((p'_1, j_1), \cdots, (p'_\eta, j_\eta); \tau)$ that defined in the calculation of the subproblem of $F(p_1, p_2, \ldots, p_\eta; t)$ in its state transitions. Consider the state transition that the value function transits from $F(p_1, p_2, \ldots, p_\eta; t)$ to $F(p_1+j_1, p_2+j_2, \ldots, p_\eta+j_\eta; t')$. In other words, when the order set of the subproblem transits from $N(p_1, p_2, \ldots, p_\eta)$ to $N(p_1 + j_1, p_2 + j_2, \ldots, p_\eta + j_\eta)$, a production subsequence with order set $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ is added into the original order set $N(p_1, p_2, \ldots, p_\eta)$ in the state transition. And the value function $I((p_1, j_1), \cdots, (p_\eta, j_\eta); t')$ denotes the minimum shipping costs and inventory holding costs in the added production subsequence given order set $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$, whose completion day of its last order is t', which is the same as the last order in $N(p_1, p_2, \ldots, p_\eta)$.

Accordingly, we have the following Lemma 3.3 and Lemma 3.4 that aids the calculation in the state transition of the value function $F(p_1, p_2, \ldots, p_\eta; t)$ and $I((p'_1, j_1), \cdots, (p'_\eta, j_\eta); \tau)$.

Lemma 3.3. *Consider any production subsequence $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ with production start day τ_s and production completion day τ_e, which is in the optimal solution. It satisfies that production quantity on day τ_s is less than c and the production quantity on each day $t \in \{\tau_s + 1, \tau_s + 2, \cdots, \tau_e\}$ equals c.*

Proof. Suppose there exists a day τ during which the production quantity is less than c. We can divide the situation into two cases depending on whether there is inventory at the end of day τ. For the first case, there is no inventory on day τ. According to the definition of the production subsequence, we know that there is no inventory at the end of day τ_e. Then the subsequence $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ with production start day τ_s and production completion time τ_e can be further split on day τ, i.e., it can be split into production subsequence $N'((p_1, j'_1), (p_2, j'_2), \cdots, (p_\eta, j'_\eta))$ with production start day τ_s and production completion time τ and production subsequence $N'((p_1+j_1', j_1''), (p_2+j_2', j_2''), \cdots, (p_\eta+j_\eta', j_\eta''))$ with production start day $\tau+1$ and production completion time τ_e. Therefore, Lemma 3.3 still holds by the split. For the second case, there are inventories on day τ. We can prove it by contradiction. At this moment, we can use the idle time on day τ to produce the products on day $\tau-1$, i.e., producing the products as late as possible, so that the production quantity on day τ reaches to c and the inventory holding costs incurred by the products of partially completed order on day $\tau-1$ can be decreased while its shipping costs remain unchanged. This violates that the production subsequence is in an optimal solution. Therefore, Lemma 3.3 is proved. □

Lemma 3.4. *Consider any two production subsequences* $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ *with production start day* τ_s *and production completion day* τ_e *and* $N'((p_1+j_1, j_1'), (p_2+j_2, j_2'), \cdots, (p_\eta+j_\eta, j_\eta'))$ *with production start day* τ_s' *and production completion day* τ_e'*, which are in the optimal solution. Then, we have* $\tau_e < \tau_s' \leq \tau_e + 1$ *and* $\tau_e' = \tau_e + \left\lceil \frac{\sum_{r=1}^{\eta} j_r' e_r}{c} \right\rceil$.

Proof. Based on Lemma 3.3, for production subsequence $N'((p_1, j_1), (p_2, j_2), \cdots, (p_\eta, j_\eta))$ with production start day τ_s and production completion day τ_e, we know that the production quantity on day τ_e is c. Therefore, since no remaining production capacity available on day τ_e, the production start day for its consecutive production subsequence must be later than τ_e, i.e., $\tau_e < \tau_s'$.

Consequently, we can prove $\tau_s' \leq \tau_e + 1$ by contradiction. Suppose $\tau_s' > \tau_e + 1$, which means the idle time between two production subsequences is larger than 1 day. Then we make the production on day τ_s' and later time 1 day earlier. According to the definition of the production subsequence, the shipping costs can be decreased while the inventory holding costs keep the same. This violates the fact that the production subsequence is in an optimal solution.

Therefore, based on the description above, we know that the production completion day of production subsequence $N'((p_1+j_1, j_1'), (p_2+j_2, j_2'), \cdots, (p_\eta+j_\eta, j_\eta'))$ is its earliest production completion day, which can be calculated as $\tau_e' = \tau_e + \left\lceil \frac{\sum_{r=1}^{\eta} j_r' e_r}{c} \right\rceil$. Therefore, Lemma 3.4 is proved. □

In the following, we study the dynamic programs to recursively calculate the value function $F(p_1, p_2, \ldots, p_\eta; t)$ as well as the value function $I((p'_1, j_1), \cdots, (p'_\eta, j_\eta); \tau)$.

First, since the subproblem of $F(0, 0, \ldots, 0; 0)$ is defined for an empty order set, its minimum total shipping costs and inventory holding costs are zero. Thus, we obtain the boundary condition of the dynamic program that $F(0, 0, \ldots, 0; 0) = 0$. Similarly, we can also obtain the boundary condition for the dynamic program that $I((p'_1, 0), (p'_2, 0), \cdots, (p'_\eta, 0); 0)) = 0$ for each $(p'_1, p'_2, \cdots, p'_\eta)$ with $p'_k \in \{0, 1, \cdots, n_k\}$ for $k \in \{1, 2, \cdots, \eta\}$ since its subproblem is also defined for an empty order set according to the definition described above.

From Equation (3.22) and the definition of value function $I((p'_1, j_1), \cdots, (p'_\eta, j_\eta); \tau)$ and the value function $F(p_1, \cdots, p_\eta; t)$, we can find the optimal production subsequences that can form the order set $N(p_1, \cdots, p_\eta)$. For each $(p_1, p_2, \cdots, p_\eta)$ with $p_k \in \{0, 1, \cdots, n_k\}$ for $k \in \{1, 2, \cdots, \eta\}$ and with $\sum_{r=1}^{\eta} p_r \geq 1$ and for each $t \in T$, we can apply Theorem 3.2 to the subproblem of $F(p_1, p_2, \ldots, p_\eta; t)$. This indicates that there exists a shipping plan $\mathbf{z}$ of orders in $N(p_1, p_2, \ldots, p_\eta)$ such that $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ forms an optimal solution to the subproblem. Accordingly, the last production subsequence in the order set $N(p_1, p_2, \ldots, p_\eta)$ should be $N'((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta))$ with the production completion day of its last order to be on day t for some $l \in \{1, \cdots, \eta\}, \forall j_l \in \{0, 1, \ldots, p_l - 1\}$. And according to Lemma 3.3 and Lemma 3.4, we know that the production completion day of the last order of its previous production subsequence is $t - \left\lceil \frac{\sum_{r=1}^{\eta} j_r e_r}{c} \right\rceil$. Moreover, as no inventory exist after the order completion day, the total shipping costs and inventory holding costs of all orders in $N(p_1, p_2, \ldots, p_\eta)$ equal all that of orders $N(p_1 - j_1, p_2 - p_2, \ldots, p_\eta - j_\eta)$, i.e., $F\left(p_1 - j_1, \cdots, p_\eta - j_\eta, t - \left\lceil \frac{\sum_{r=1}^{\eta} j_r e_r}{c} \right\rceil\right)$ plus that of orders $N'((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta))$, i.e., $I((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t)$. An example of this process is showing in Figure 3.3(a). Therefore, we can enumerate l, j_l to calculate $F(p_1, \cdots, p_\eta; t)$ by the following recursive equation:

$$F(p_1, \cdots, p_\eta; t) = \min \left\{ \begin{array}{l} F\left(p_1 - j_1, \cdots, p_\eta - j_\eta, t - \left\lceil \frac{\sum_{r=1}^{\eta} j_r e_r}{c} \right\rceil\right) + I\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t\right) \\ \mid \forall l \in \{1, \cdots, \eta\}, \forall j_l \in \{0, 1, \ldots, p_l - 1\} \text{ with } \sum_{i=1}^{\eta} j_i > 0,\ p_l - j_l \geq 1 \\ \text{ and } d_{i(l,p_l)} \geq t - \left\lceil \frac{\sum_{r=1}^{\eta} j_r e_r}{c} \right\rceil \end{array} \right\} \quad (3.23)$$

Finally, the value of $F(n_1, \ldots, n_\eta; m)$ is returned which is the minimum total shipping costs and inventory holding costs for problem IPTSDI.

Similarly, given the value of $(p_1 - j_1, p_2 - j_2, \cdots, p_\eta - j_\eta)$ and t, for each $(j'_1, j'_2, \cdots, j'_\eta)$ with $j'_k \in \{0, 1, \cdots, j_k\}$ for $k \in \{1, 2, \cdots, \eta\}$ and with $\sum_{r=1}^{\eta} j'_r \geq 1$, we can also apply Theorem 3.2 to the subproblem of $I\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t\right)$. To minimize the inventory holding cost, we arrange the sequence of orders in $N'\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta)\right)$ of the subproblem from the last to the first. The first order in $N'\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta)\right)$ must be order $i(\omega^*, p_{\omega^*})$ for some $\omega^* \in \{1, 2, \cdots, \eta\}$. Moreover, order $i(\omega^*, p_{\omega^*})$ should be shipped out on or before its committed delivery due date, which means that the ω^* satisfies that $p_{\omega^*} \geq 1$ and $d_{i(\omega^*, p_{\omega^*})} \geq t - \left\lceil \frac{\sum_{r=1}^{\eta} p'_r e_r}{c} \right\rceil$. Thus, if no such ω^* exists, the subproblem of $I\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t\right)$ must have no feasible solution, implying that $F(p_1, p_2, \ldots, p_\eta; t) = +\infty$. Otherwise, the shipping costs and possible inventory holding costs for order $i(\omega^*, p_{\omega^*})$ equals g, and for other orders in $N'((p_1 - j_1, j'_1), \cdots, (p_{w^*-1} - j_{w^*-1}, j'_{w^*-1}), (p_{w^*} - j_{w^*}, j'_{w^*} - 1), (p'_{w^*+1}, j'_{w^*+1}), \cdots, (p_\eta - j_\eta, j'_\eta))$ (which equals $N'((p_1 - j_1, j'_1), \cdots, (p_{w^*-1} - j_{w^*-1}, j'_{w^*-1}), (p_{w^*} - j_{w^*}, j'_{w^*} - 1), (p'_{w^*+1}, j'_{w^*+1}), \cdots, (p_\eta - j_\eta, j'_\eta) \backslash \{i(\omega^*, p_{\omega^*})\})$, their production and shipping plans must form an optimal solution to the subproblem of $I\left((p_1 - j_1, j'_1), \cdots, (p_{w^*-1} - j_{w^*-1}, j'_{w^*-1}), (p_{w^*} - j_{w^*}, j'_{w^*} - 1), (p'_{w^*+1}, j'_{w^*+1}), \cdots, (p_\eta - j_\eta, j'_\eta); t\right)$. (See Figure 3.3(b) for an illustrative example.) Accordingly, we can enumerate ω^* to compute $I\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t\right)$ by the following recursive equation:

$$
\begin{aligned}
&I((p_1 - j_1, j_1'), \cdots, (p_\eta - j_\eta, j_\eta'); t) \\
&= \min \begin{cases}
\min \left\{ \begin{array}{l}
I((p_1 - j_1, j_1'), \cdots, (p_{w^*-1} - j_{w^*-1}, j'_{w^*-1}), (p_{w^*} - j_{w^*}, j'_{w^*} - 1), \\
\quad (p'_{w^*+1}, j'_{w^*+1}), \cdots, (p_\eta - j_\eta, j'_\eta); t) + g | \forall \omega^* \in \{1, \cdots, \eta\}, \\
\quad \text{with } p'_{w^*} \geq 1 \text{ and } d_{i(\omega^*, p'_{w^*})} \geq t - \left\lceil \frac{\sum_{r=1}^{\eta} p'_r e_r}{c} \right\rceil
\end{array} \right\}, \\
\quad \text{if such } w^* \text{ and } p'_{w^*} \text{ exist} \\
+\infty, \text{otherwise;}
\end{cases}
\end{aligned}
\tag{3.24}
$$

where

$$
g = \begin{cases}
G(d_{i(\omega^*, p_{\omega^*})} - (t + \lceil \sum_{r=1}^{\eta} p_r e_r / c \rceil), e_{\omega^*}), \text{if } \lceil (\sum_{r=1}^{\eta} p_r e_r - e_{\omega^*})/c \rceil = \lceil \sum_{r=1}^{\eta} p_r e_r / c \rceil \\
G(d_{i(\omega^*, p_{\omega^*})} - (t + \lceil \sum_{r=1}^{\eta} p_r e_r / c \rceil), e_{\omega^*}) + h(e_{\omega^*} - \lceil (\sum_{r=1}^{\eta} p_r e_r - e_{\omega^*})/c \rceil \cdot c), \\
\qquad \text{if } \lceil (\sum_{r=1}^{\eta} p_r e_r - e_{\omega^*})/c \rceil < \lceil \sum_{r=1}^{\eta} p_r e_r / c \rceil
\end{cases}. \tag{3.25}
$$

Finally, the value of $I\left((p_1 - j_1, j_1), \cdots, (p_\eta - j_\eta, j_\eta); t\right)$ is returned.

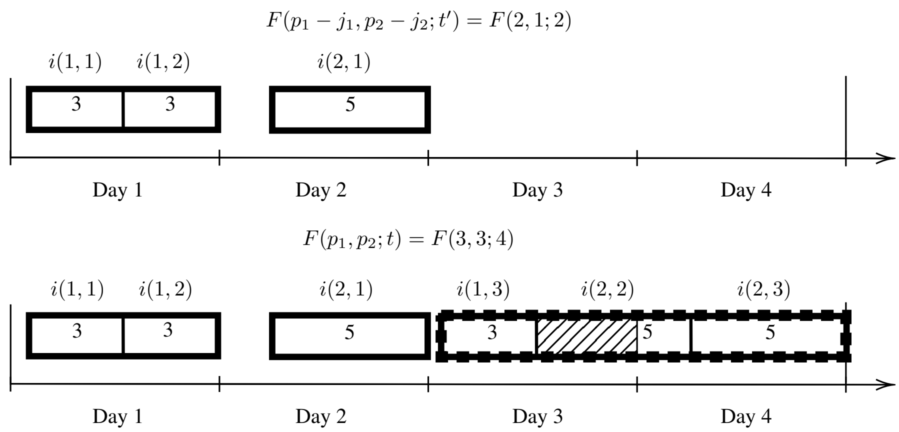
Algorithm 3.4 (for problem IPTSDI)

$F(0, 0, \ldots, 0) \leftarrow 0$
for all $(p_1, p_2, \ldots, p_\eta)$ with $p_k = 0, 1, \ldots, n_k$ for $k \in \{1, \ldots, \eta\}$ and with $\sum_{r=1}^{\eta} p_r \geq 1$ **do**
  for all $t = 0, 1, \cdots, m$ **do**
    Compute $F(p_1, p_2, \ldots, p_\eta, t)$ by the recursive equation in (3.23)
  end for
end for
return $F(n_1, \ldots, n_\eta; m)$

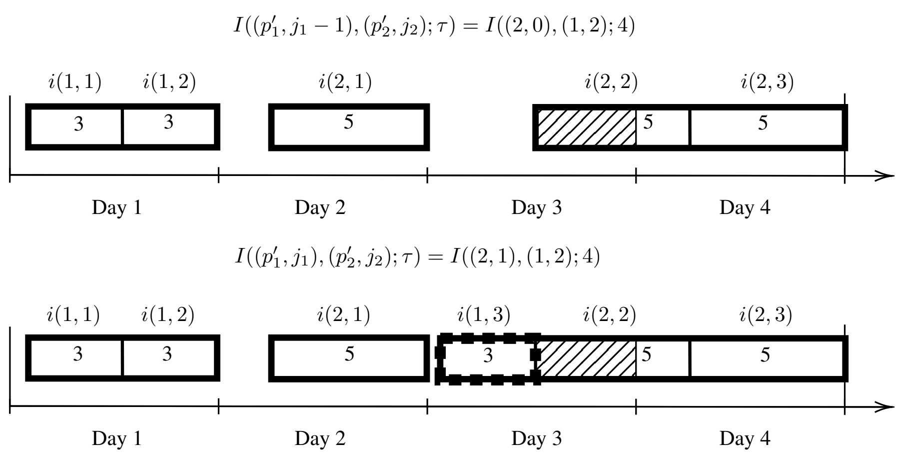
Theorem 3.3. *Algorithm 3.4 solves problem IPTSDI to optimal with $O(nm\eta^2 \cdot (1 + n/\eta)^\eta)$ running times.*

Proof. Since Algorithm 3.4 calculate the value function $F(p_1, p_2, \ldots, p_\eta; t)$ recursively in equation (3.23) and the final value $F(n_1, \ldots, n_\eta; m)$, according to the definition of the value function, is the minimum shipping cost and inventory holding cost for problem IPTSDI, which indicates that Algorithm 3.4 returns an optimal solution to problem problem IPTSDI.

Figure 3.3: An example for Algorithm 3.4 with $\eta = 2$ types of quantities ($e_1 = 3$ and $e_2 = 5$) and the planning horizon is $m = 4$.



(a) For the subproblem of $F(3, 3; 4)$ with six orders in the order set $\{i(1, 1), i(1, 2), i(1, 3), i(2, 1), i(2, 2), i(2, 3)\}$ and order completion day of the last order is 4. Since $(i(1, 3), i(2, 2), i(2, 3))$ is the last production subsequence of in the optimal solution, $F(3, 3; 4)$ equals $F(2, 1; 2) + I((2, 1), (1, 2); 4)$



(b) For the subproblem of $I((2, 1), (1, 2); 4)$ with 3 orders in the order set $\{i(1, 3), i(2, 2), i(2, 3)\}$ and order completion day of the last order is 4. Since $(i(1, 3))$ is the first order of in the optimal solution, $I((2, 1), (1, 2); 4))$ equals $I((2, 0), (1, 2); 4) + G(d_{i(1,3)} - 3, 3)$.

Moreover, we can see that Algorithm 3.4 computes equation (3.23) for at most $(1 + n_1)(1 + n_2) \cdots (1 + n_\eta) \cdot m$ times. Since for all $k \in \{1, 2, \cdots, \eta\}$, N_k form a partition of order N, which means that $n_1 + n_2 + \ldots + n_\eta = n$, then the running time above can reduced to be $(1 + n_1)(1 + n_2) \cdots (1 + n_\eta) \leq [\sum_{k=1}^{\eta}(1 + n_k)/\eta]^\eta = (1 + n/\eta)^\eta$. And equation (3.23) itself will run in $O(n\eta)$ times while equation (3.24) will run in $O(\eta)$ times. Therefore, the running time of Algorithm 3.4 is $O(nm\eta^2 \cdot (1 + n/\eta)^\eta)$. $\square$

3.4.2 Exact Algorithm When the Planning Horizon is Fixed

In this section, we present the second exact algorithm for problem IPTSDI. It is a pseudo-polynomial time algorithm when the planning horizon m is bounded by a constant. We first show that we only need to focus on the optimization of the shipping plan $\mathbf{z}$ to solve this problem to optimality. And then, based on this idea, we develop the exact algorithm with a dynamic program.

First, let us to consider a feasible solution $(\mathbf{x}, \mathbf{z})$ to model ILP. Recall that for each $t \in T$, Q_t denotes the total shipping quantity out on day t, i.e., $Q_t = \sum_{i \in N} q_i z_{it}$. And $(Q_1, Q_2, \cdots, Q_m)$ satisfies condition (3.20). We can now establish Proposition 3.1 for any $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied.

Proposition 3.1. *Consider any $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied. For any shipping plan $\mathbf{z}$ with $\sum_{i \in N} q_i z_{it} = Q_t$ for $t \in T$, $\sum_{t=1}^{d_i} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, and $z_{it} \in \{0, 1\}$ for $i \in N$ and $t \in T$, a production plan $\mathbf{x}$ always exists such that solution $(\mathbf{x}, \mathbf{z})$ is a feasible to model ILP.*

Proof. As illustrated in Section 3.3, Algorithm 3.1 can construct a solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ to model ILP-AG given a shipping plan $\mathbf{z}$. According to Lemma 3.1, we know that for any shipping plan $\mathbf{z}$, the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ satisfies constraints (3.12), (3.13), and (3.16), (3.17) (3.18) of model ILP-AG. Since the shipping plan $\mathbf{z}$ satisfies $\sum_{i \in N} q_i z_{it} = Q_t$ for $t \in T$, $\sum_{t=1}^{d_i} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$, and $z_{it} \in \{0, 1\}$ for $i \in N$ and $t \in T$, which is specified in Proposition 3.1, constraints (3.14) and (3.15) are also satisfied. Therefore, the constructed solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ is feasible to model ILP-AG.

From Algorithm 3.2 and 3.3, we know that the feasibility of solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ to model ILP-AG ensures constraints (3.7) (3.8) and (3.10) to be satisfied. Moreover, with condition (3.20) is satisfied, and since solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ also satisfies constraints (3.12), (3.13) and (3.16), the allocation of the daily production quantity to production quantity for each order on each day in step 4–5 and step 10–11 in Algorithm 3.2 ensures constraints (3.5), (3.6) and

(3.9) to be satisfied. Thus, solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ is feasible to model ILP. Therefore, Proposition 3.1 is proved. $\square$

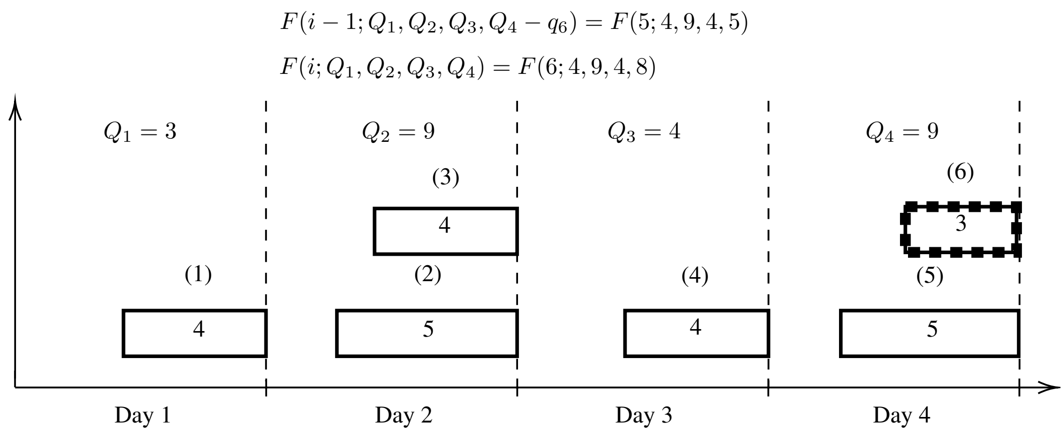
Next, for every $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied, we define $F(Q_1, Q_2, \ldots, Q_m)$ to be the minimum shipping cost among all the shipping plans $\mathbf{z}$ that satisfy $\sum_{i \in N} q_i z_{it} = Q_t$ for $t \in T$ and satisfy $\sum_{t=1}^{m} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$. From the analysis in Proposition 3.1, to solve model ILP is equivalent to minimize $F(Q_1, Q_2, \ldots, Q_m)$ over all such $(Q_1, Q_2, \cdots, Q_m)$. In the following, we present a dynamic programming algorithm to achieve this.

For each $i \in \{0, 1, \cdots, n\}$, denote set $N(i) = \{i' \in N \mid i' \leq i\}$ as the subset of first i orders in the order set N. For each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied and subset $N(i)$, denote value function $F(i; Q_1, Q_2, \ldots, Q_m)$ to be the minimum total shipping cost of all orders in $N(i)$. This subproblem needs to determine a shipping plan $\mathbf{z}$ such that $\sum_{i' \in N(i)} q_{i'} z_{i',t} = Q_t$ for $t \in T$ and that $\sum_{t=1}^{m} z_{i',t} = 1$ and $\sum_{t=d_{i'}+1}^{m} z_{i',t} = 0$ for $i' \in N(i)$. If no shipping plan exist to satisfy these conditions for the subproblem, the value of $F(i; Q_1, Q_2, \ldots, Q_m)$ is $+\infty$. Accordingly, we have $F(Q_1, Q_2, \ldots, Q_m) = F(n; Q_1, Q_2, \cdots, Q_m)$.

The value function $F(i; Q_1, Q_2, \cdots, Q_m)$ can be computed recursively as follows. Since the subproblem of $F(0; 0, \ldots, 0)$ is defined for an empty order set, we obtain the boundary condition of the dynamic program that $F(0; 0, \ldots, 0) = 0$, and that $F(0; Q_1, Q_2, \ldots, Q_m) = +\infty$ for each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied and with $\sum_{t=1}^{m} Q_t > 0$.

For each $i \in \{1, 2, \ldots, n\}$ and for each $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied, for a subproblem of $F(i; Q_1, Q_2, \cdots, Q_m)$ with its optimal shipping plan to be $\mathbf{z}$, denote $\tau_i \in \{1, 2, \cdots, m\}$ to be the shipping day of the last order in $N(i)$, i.e., order i. We know that τ_i must satisfy that $\tau_i \leq d_i$ and $q_i \leq Q_{\tau_i}$, $G(d_i - \tau_i, q_i)$ is the shipping cost for order i. The shipping days of orders in $N(i-1) = N(i) \setminus \{i\}$ which can be obtained from the shipping plan $\mathbf{z}$ are also optimal for the subproblem of $F(i-1; Q_1, \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i + 1}, \ldots, Q_m)$. (See Figure 3.4 for illustration.) Accordingly, we can enumerate τ_i to compute $F(i; Q_1, Q_2, \cdots, Q_m)$ by the following recursive equation:

Figure 3.4: An example for Algorithm 3.5 with a 4-day planning horizon where the numbers in the rectangles are order quantities and numbers above with parenthesizes are order indices: For the subproblem of $F(6;4,9,4,8)$, for orders in $\{1,2,\cdots,5\}$, since order 6 is delivered on day 4 in an optimal shipping plan (i.e., $\tau_6=4$), $F(6;4,9,4,8)$ equals $F(5;4,9,4,5)+G(d_6-4,3)$ (since $\tau_6=4$ and $q_6=3$).



$$
\begin{aligned}
&F(i;Q_1,Q_2,\ldots,Q_m)\\
&=\min\left\{\begin{array}{l} F(i-1;Q_1,\ldots,Q_{\tau_i-1},Q_{\tau_i}-q_i,Q_{\tau_i+1},\ldots,Q_m)+G(d_i-\tau_i,q_i)\mid \\ \forall\tau_i\in\{1,2,\ldots m\} \text{ with } \tau_i\le d_i \text{ and } q_i\le Q_{\tau_i}\end{array}\right\}, \quad (3.26)
\end{aligned}
$$

where we assume that the value function equals $+\infty$ if taking the minimum over an empty set.

Finally, noting that $F(Q_1,Q_2,\ldots,Q_m)=F(n;Q_1,Q_2,\cdots,Q_m)$, we can enumerate all $(Q_1,Q_2,\cdots,Q_m)\in\mathbb{Z}_+^m$ satisfying (3.20) to find the minimal value of $F(n;Q_1,Q_2,\cdots,Q_m)+H(Q_1,Q_2,\cdots,Q_m)$, where $H(Q_1,Q_2,\cdots,Q_m)$ is the minimized inventory holding costs given $(Q_1,Q_2,\cdots,Q_m)$ and can be computed by Algorithm 3.3 according to Corollary 3.1. The found minimum value is the minimum total shipping cost and inventory holding cost for problem IPTSDI.

Algorithm 3.5 (for problem IPTSDI)

1: $F(0;0,0,\ldots,0)\leftarrow 0$, and $F(0;Q_1,Q_2,\ldots,Q_m)\leftarrow+\infty$ for all $(Q_1,Q_2,\cdots,Q_m)\in\mathbb{Z}_+^m$ with (3.20) satisfied and with $\sum_{t=1}^m Q_t>0$

2: **for all** $i = 1, 2, \cdots, n$ **do**

3: **for all** $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}^m$ with (3.20) satisfied **do**

4: Compute $F(i; Q_1, Q_2, \ldots, Q_m)$ by the recursive equation in (3.26)

5: **end for**

6: **end for**

7: **return** the minimum value of $F(n; Q_1, Q_2, \cdots, Q_m) + H(Q_1, Q_2, \cdots, Q_m)$ over all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied

---

Theorem 3.4. *Algorithm 3.5 can solve problem IPTSDI to optimality with $O(nc^m(m!)m)$ running times.*

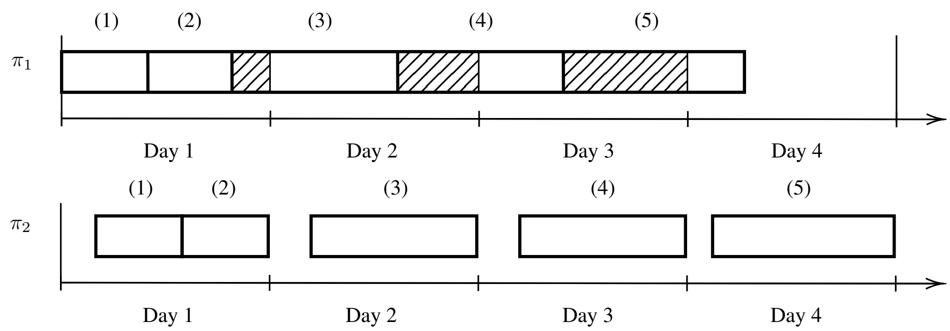
Proof. Algorithm 3.5 computes the value function $F(i; Q_1, Q_2, \ldots, Q_m)$ in (3.26) over all $(Q_1, Q_2, \cdots, Q_m) \in \mathbb{Z}_+^m$ with (3.20) satisfied. Denote the combination of shipping quantities in the optimal solution to be $(Q_1^*, Q_2^*, \cdots Q_m^*)$. From step 7 in Algorithm 3.5, we can know that for the obtain solution $(Q_1, Q_2, \cdots, Q_m)$, we have $F(n; Q_1, Q_2, \ldots, Q_m) \leq F(n; Q_1^*, Q_2^*, \cdots, Q_m^*)$. And from Corollary 3.1, we also have $H(Q_1, Q_2, \cdots, Q_m) = H(Q_1^*, Q_2^*, \cdots, Q_m^*)$. Therefore, the total shipping costs and inventory holding costs of the obtained solution in Algorithm 3.5 are less or equal to the total shipping costs and inventory holding costs of the optimal solution to problem IPTSDI. Therefore, Algorithm 3.5 can return an optimal solution to problem IPTSDI.

Moreover, Algorithm 3.5 computes (3.26) for at most $n \cdot (1 \cdot c)(2 \cdot c) \cdots (m \cdot c) = nc^m(m!)$ times since $Q_t \leq tc$ for $1 \leq t \leq m$. And the recursive equation (3.26) itself will run in $O(m)$ time. These lead $O(nc^m(m!)m)$ to be running times of Algorithm 3.4. $\square$

3.5 Infinite Unit Inventory Holding Cost

In this section, we examine the problem IPTSDI when the unit inventory holding cost h is $+\infty$, i.e., no inventory is allowed during the production process. We can see two solution examples in Figure 3.5. Under this setting, no inventory holding cost is incurred in a feasible

Figure 3.5: Solution examples for problem IPTSDI when h is $+\infty$. A rectangle represents an order and the number with bracket above the rectangle is the order index. There are inventories (shaded rectangles) in solution π_1, which means the total cost is $+\infty$ and no inventory incurs in solution π_2.



solution, otherwise, the total cost would be $+\infty$. Next, we study the complexity of problem IPTSDI when h is $+\infty$.

Theorem 3.5. *There is no finite ratio pseudo-polynomial time approximation algorithm for problem IPTSDI when the unit inventory holding cost $h = +\infty$ unless NP=P.*

Proof. From Stecke and Zhao (2007) and Zhong et al. (2010), we know that the problem IPTSD is strongly NP-hard and it can be reduced to 3-Partition Problem (3-PP) which is also strongly NP-hard (Garey et al. (1978)). Therefore, we can prove this theorem using the NP-hardness of 3-PP.

3-Partition Problem (3-PP). Given integers N and B and given a set of integers $a_1, \cdots, a_{3N}$ such that $B/4 < a_i < B/2$, for $i = 1, \cdots, 3N$ and $\sum_{i=1}^{3N} a_i = NB$, the 3-PP is to determine whether or not there exist n pairwise disjoint three-element subsets $A_j \subset \{1, \cdots, 3N\}$ such that $\sum_{i \in A_j} a_i = B$ for $j = 1, \cdots, N$.

Follow Stecke and Zhao (2007), we can construct an instance U_I of problem IPTSDI given an instance U_P of 3-PP in the following way. The total number of orders $n = 3N$, the planning horizon $m = N$ and the daily production capacity $c = 3X + B$ where X is a positive constant. For each order $i = 1, \cdots, N$, $q_i = X + a_i$ with $\sum_{i=1}^{n} q_i = 3NX + NB$ and $d_i = N$. And $h = +\infty$. Suppose the daily shipping cost is $c_i = c_{i-1} + 1$ for $i = 2, \cdots, N$ and $c_1 = 1$. Then the total cost is $Z = \sum_{i=1}^{N} c_i(3X + B)$.

From U_I constructed above, we can see that objective value of an optimal solution to U_I is Z whenever a feasible solution to the instance U_P exists. However, the objective value of an optimal solution to instance U_I would be at least $Z + h$ when no feasible solution to U_P exists. In this case, it means that inventory costs are incurred due to the deliveries of some products are on the next day after they are completed in the optimal solution.

Assuming that an approximation algorithm A can solve U_I in pseudo-polynomial running time with a worst-case ratio a ($a < \infty$), i.e, $Z^A/Z^* \leq a$, where Z^A is the value of the objective function from algorithm H and Z^* is that from the optimal solution π. If $Z^A < Z + h$, which means that U_P has a feasible solution. If $Z^A \geq Z + h$, then U_P is infeasible. In other words, algorithm A can also solve U_P in pseudo-polynomial time, which contradicts the strongly NP-hardness of problem 3-pp. Therefore, Theorem 3.5 is proved. $\square$

3.6 Approximate Scheme

In this section, we describe the approximation scheme with pseudo-polynomial running time for problem IPTSDI. And we show that the worst-case performance ratio of the approximation scheme is $(1 + \epsilon)$ for any fixed and positive constant ϵ.

This section is arranged as follows: In Section 3.6.1, we formally define the restricted problem of IPTSDI and propose the algorithms extended from Algorithm 3.5 to solve it. In Section 3.6.2, we present the analytical results for the approximation scheme. In this section, we also assume that all orders are sorted according to their committed due dates in a non-decreasing order

3.6.1 Restricted problem

Denote $\bar{Q} = mc$ as full production capacity for m days and $\bar{Q}' = \sum_{i \in N} q_i$ as the total production quantities of all orders in N. Given $K \in \{1, 2, \ldots, m\}$ and $Q \in \{0, 1, \ldots, \bar{Q}\}$

with

$$K \geq \lceil Q/c \rceil, \tag{3.27}$$

satisfied, denote RP(K, Q) as the restricted problem of IPTSDI. To solve the restricted problem RP(K, Q), we need to find a feasible solution $(\mathbf{x}, \mathbf{z})$ to model ILP and a subset $I \subseteq N$ such that $\mathbf{z}$ in the feasible solution and I satisfy the constraints listed as follow:

(i) the total order quantities in the subset $N \setminus I$, i.e. $\sum_{i \in N \setminus I} q_i$, together with the unused production capacity during idle time equal Q;

(ii) For every $i \in I$, the shipping day for an order i is $\lceil (Q + \sum_{i' \in I : i' \leq i} q_{i'})/c \rceil$, i.e., $z_{it} = 1$ for $t = \lceil (Q + \sum_{i' \in I : i' \leq i} q_{i'})/c \rceil$;

(iii) For every $i \in N \setminus I$, the shipping day for an order i is no later than day $K' = \lceil Q/c \rceil$, i.e., $\sum_{t'=1}^{K'} z_{it'} = 1$ for $i \in N \setminus I$.

From the description above, we can see that orders in N are split into two subsets $N \setminus I$ and I. And the shipping day of an order in the subset $N \setminus I$ is no later than the earliest possible completion day of all orders K', which satisfies

$$K' \leq K. \tag{3.28}$$

Also, the shipping days of an order in the subset I are arranged according to its earliest possible completion day.

We can examine a case when $K = m$, $\bar{Q}' \leq Q \leq \bar{Q}$ and $I = \emptyset$. According to Theorem 3.1, there is a shipping plan $\mathbf{z}$ such that solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ is optimal to problem IPTSDI. Then, for solution under this case, (i), (ii) and (iii) are satisfied. In other words, the restricted problem RP(K, Q) is the same as problem IPTSDI when $K = m$ and $\bar{Q}' \leq Q \leq \bar{Q}$ and $I = \emptyset$. Therefore, we can utilize Algorithm 3.5 to design a pseudo-polynomial running time algorithm for the restricted problem RP(K, Q) when K is bounded. Moreover, when

m is arbitrarily large, we can further develop an approximation scheme based on the exact algorithm for restricted problem RP(K, Q) to get a high-quality solution to problem IPTSDI with a properly chosen value of K and several values of Q.

Following Section 3.3, for each $t \in \{1, 2, \ldots, K'\}$, let $Q_t = \sum_{i \in N \setminus I} q_i z_{it}$ denote the total shipping quantity of orders in subset $N \setminus I$ on day t. Similar to (3.20), we can obtain that $(Q_1, \cdots, Q_{K'})$ satisfies the following condition:

$$\sum_{t'=1}^{t} Q_{t'} \leq tc, \text{ for each } t \in \{1, 2, \ldots, K'\}. \tag{3.29}$$

Similarly, let $Q_t = \sum_{i \in I} q_i z_{it}$ for each $t \in \{K'+1, K'+2, \cdots, m\}$ denote the total shipping quantity of orders in subset I on day t.

Denote $(\mathbf{x}, \mathbf{z}, I)$ as a feasible solution to problem RP(K, Q) and Q' as the total order quantity of orders in I, i.e., $Q' = \sum_{i \in I} q_i$. We have Proposition 3.2 below.

Proposition 3.2. *Consider any* $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ *that satisfies condition (3.29) and any* $Q' \in \{1, 2, \cdots, \bar{Q}'\}$ *with* $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$*. For any subset* $I \subseteq N$ *that satisfies* $\sum_{i \in I} q_i z_{it} = Q'$ *and for any shipping plan* $\mathbf{z}$ *that satisfies* $\sum_{i \in N \setminus I} q_i z_{it} = Q_t$ *for* $t \in \{1, 2, \cdots, K'\}$ *and* $\sum_{t=1}^{m} z_{it} = 1$ *and* $\sum_{t=d_i+1}^{m} z_{it} = 0$ *for* $i \in N$*, a production plan* $\mathbf{x}$ *will always exist such that solution* $(\mathbf{x}, \mathbf{z}, I)$ *is a feasible to problem* RP(K, Q)*, if condition (i)–(iii) of problem* RP(K, Q) *are satisfied by the subset* I *and the shipping plan* $\mathbf{z}$*.*

Proof. Consider a $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$, a subset I, and a shipping plan $\mathbf{z}$ satisfying all the conditions specified in Proposition 3.2. We can first construct a solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ to model ILP-AG using Algorithm 3.1. And we can prove the constructed solution is feasible to model ILP-AG. According to Lemma 3.1, solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ satisfies constraints (3.12), (3.13), and (3.16), (3.17) (3.18).

Then, from the shipping plan $\mathbf{z}$, an order sequence $\sigma = (\sigma_1, \sigma_2, \cdots, \sigma_n)$ can be achieved by sorting the shipping day of the orders in a non-decreasing order. More specifically, denote j^* as the largest index of order $\sigma_j \in N \setminus I$. Thus, then the order sequence can be split accordingly as $N \setminus I = \{\sigma_1, \sigma_2, \cdots, \sigma_{j^*}\}$ and $I = \{\sigma_{j^*+1}, \sigma_{j^*+2}, \cdots, \sigma_n\}$.

Recall that, in Section 3.3, for $j \in \{1, 2, \cdots, n\}$, ω_{σ_j} is the shipping day of order σ_j under the shipping plan $\mathbf{z}$. And denote $t_j = \lceil \sum_{\tau=1}^{j} q_{\sigma_\tau}/c \rceil$ as the earliest possible production completion day of the first j orders in σ. In the following, we show that $t_j \leq \omega_{\sigma_j}$ for each order σ_j of σ:

- For each order $\sigma_j \in N \setminus I$, i.e., $j \in \{1, 2, \cdots, j^*\}$. We know that in the shipping plan $\mathbf{z}$, the first j orders in sequence σ, would be shipped no later than the shipping day of the last order ω_{σ_j} where $\omega_{\sigma_j} \leq K'$ according to the definition of problem RP(K, Q). Together with (3.29), we obtain $\sum_{j'=1}^{j} q_{\sigma_{j'}} \leq \sum_{t'=1}^{\tau_j} Q_{t'} \leq \omega_{\sigma_j} c$. Therefore, combing with the backward construction process in Algorithm 3.1, we can conclude that that $t_j \leq \omega_{\sigma_j}$.

- For each order $\sigma_j \in I$, i.e., $j \in \{j^* + 1, \cdots, n\}$. From the definition of problem RP(K, Q) for orders in subset I, $t_j = \omega_{\sigma_j}$.

Thus, solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ also satisfies constraints (3.14) and (3.15), which means it is feasible to model ILP-AG. Follow the arguments in Proposition 3.1, a feasible solution $(\mathbf{x}(\mathbf{z}), \mathbf{z})$ to model ILP can be also constructed from solution $(\bar{\mathbf{x}}(\mathbf{z}), \mathbf{z})$ by Algorithm 3.2. Therefore, Proposition 3.2 is proved. $\square$

For each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (3.29) satisfied and for any $Q' \in \{1, 2, \cdots, \bar{Q}'\}$ with $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$, let $F(Q'; Q_1, Q_2, \ldots, Q_{K'})$ denote the minimum total shipping costs and inventory holding costs among all the subsets $I \subseteq N$ and shipping plans $\mathbf{z}$ that satisfy $\sum_{i \in I} q_i z_{it} = Q'$ and $\sum_{i \in N} q_i z_{it} = Q_t$ for $t \in \{1, 2, \cdots, K'\}$, $\sum_{t=1}^{m} z_{it} = 1$ and $\sum_{t=d_i+1}^{m} z_{it} = 0$ for $i \in N$ and the condition (i)–(iii) specified in problem RP(K, Q). From this definition and analysis in Proposition 3.2, solving problem RP(K, Q) is equivalent to minimize $F(Q'; Q_1, Q_2, \ldots, Q_{K'})$.

Follow Section 3.4.2, we still use $N(i) = \{i' \in N \mid i' \leq i\}$ to denote the first i orders in the order set N, for each $i \in \{0, 1, \cdots, n\}$. For each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$

with (3.29) satisfied, and for each $Q' \in \{0, 1, \cdots, \bar{Q}'\}$ with $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$, denote $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ as a value function to minimize total shipping cost of a subproblem of RP(K, Q) with an order subset $I \subseteq N(i)$ and a shipping plan $\mathbf{z}$ in $N(i)$ and with constraints listed in the follow are all satisfied:

$$\sum_{i' \in I} q_{i'} = Q', \tag{3.30}$$

$$\sum_{i' \in N(i) \setminus I} q_{i'} z_{i',t} = Q_t, \text{ for } t \in \{1, 2, \cdots, K'\}, \tag{3.31}$$

$$\sum_{t=1}^{m} z_{i',t} = 1, \text{ for } i' \in N(i), \tag{3.32}$$

$$\sum_{t=d_{i'}+1}^{m} z_{i',t} = 0, \text{ for } i' \in N(i), \tag{3.33}$$

$$\sum_{t'=1}^{K'} z_{i't'} = 1, \text{ for } i' \in N(i) \setminus I, \tag{3.34}$$

$$z_{it} = 1, \text{ for } i \in I \text{ and for } t = \lceil (Q + \textstyle\sum_{i' \in I : i' \leq i} q_{i'})/c \rceil, \tag{3.35}$$

$$z_{it} \in \{0, 1\}, \text{ for } i \in I \text{ and } t \in T. \tag{3.36}$$

Constraint (3.30) ensures that the total quantity in the subset I is Q' and (3.31) ensures that Q_t is the total shipping quantity on day t. (3.32) and (3.33) limits the shipping day of orders in $N(i)$ to be no later than their committed delivery due dates. Moreover, condition (ii) for orders in I and (iii) for orders in $N(i) \setminus I$ in restricted problem RP(K, Q) are reflected in constraints (3.34) and (3.35). And (3.36) is binary constraints on variables z_{it}.

The value of $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ would be set to $+\infty$ when no combination of I and $\mathbf{z}$ can be found for this subproblem. And $F(Q'; Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q'; Q_1, \ldots, Q_{K'})$ with $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$ according to the definition.

The value function $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$ can be computed recursively as follows. Since the subproblem of $F(0; 0; 0, \ldots, 0)$ is defined for an empty order set, we obtain the boundary condition of the dynamic program that $F(0; 0; 0, \ldots, 0) = 0$, and $F(0; Q'; Q_1, \ldots, Q_{K'}) = +\infty$ for each $(Q_1, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (3.29) satisfied, for each $Q' \in \{0, 1, \cdots, \bar{Q}'\}$ and with

$0 < Q' + \sum_{t=1}^{K'} Q_t \leq \bar{Q}'$.

For each $i \in \{1, 2, \ldots, n\}$, for each $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (3.29) satisfied and for each $Q' \in \{0, 1, \cdots, \bar{Q}'\}$ with $\sum_{t=1}^{K'} Q_t + Q' \leq \bar{Q}'$, there are two possible cases when the solution of subproblem $F(i; Q', Q_1, Q_2, \cdots, Q_{K'})$ is optimal for different order subset I and shipping plan $\mathbf{z}$ respectively:

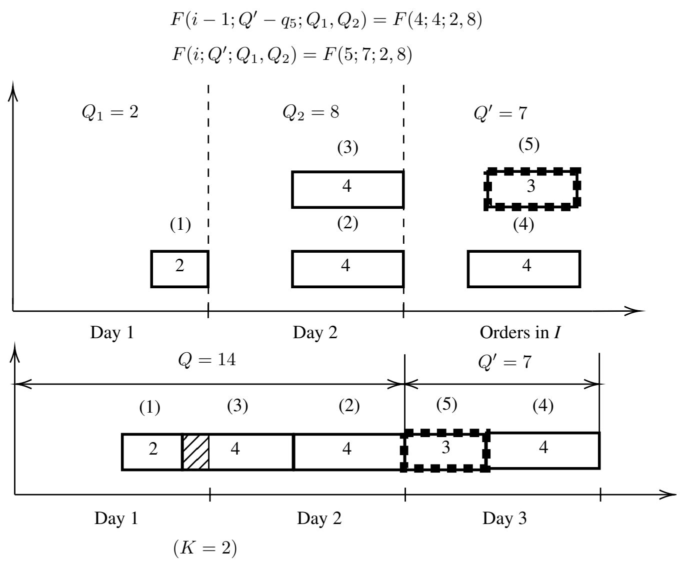
Case 1 (see Figure 3.6(a) for illustration): In this case, $i \in I$ in the optimal solution for the subproblem of $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$ with I and $\mathbf{z}$. According the constraints (3.30) and (3.35) shown above, we can obtain that $\lceil (Q + \sum_{i' \in I: i' \leq i} q_i)/c \rceil = \lceil (Q + \sum_{i' \in I} q_i)/c \rceil = \lceil (Q + Q')/c \rceil$ is the shipping day of order i and it satisfies

$$\lceil (Q + Q')/c \rceil \leq d_i. \tag{3.37}$$

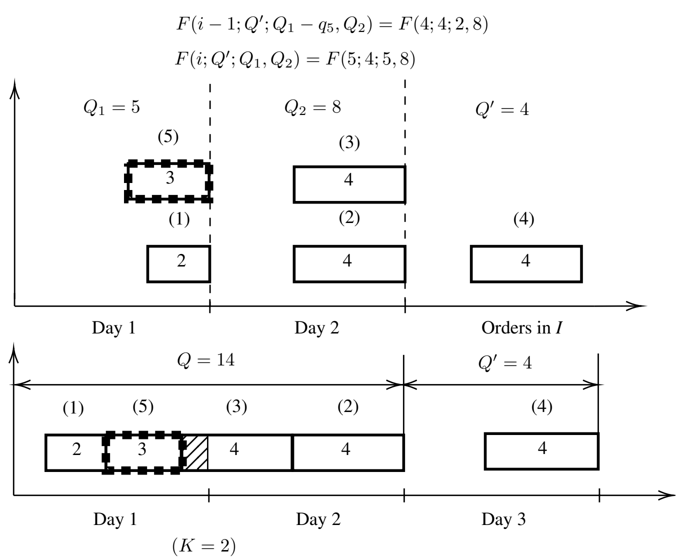
Thus, the shipping cost for order i can be written as $G(d_i - \lceil (Q + Q')/c \rceil, q_i)$. In addition, in the subproblem $F(i-1; Q' - q_i; Q_1, \ldots, Q_{K'})$ for orders in the subset $N(i-1)$, we know that the order subset $I \setminus \{i\}$ and the shipping plan for orders in $N(i-1) = N(i) \setminus \{i\}$ under $\mathbf{z}$ are also optimal. That is, $F(i; Q'; Q_1, \ldots, Q_{K'})$ equals $F(i-1; Q' - q_i; Q_1, \ldots, Q_{K'}) + G(d_i - \lceil (Q + Q')/c \rceil, q_i)$.

Case 2 (see Figure 3.6(b) for illustration): In this case $i \in N(i) \setminus I$ in the optimal solution for the subproblem of $F(i; Q'; Q_1, Q_2, \cdots, Q_{K'})$ with I and $\mathbf{z}$. Similar to Section 3.4.2, denote $\tau_i \in \{1, 2, \cdots, K'\}$ to be the shipping day of the last order in $N(i)$, i.e., order i. We know that τ_i must satisfy that $\tau_i \leq d_i$ and $q_i \leq Q_{\tau_i}$ with constraints (3.33) and (3.34) satisfied. Thus, we can write $G(d_i - \tau_i, q_i)$ to be the shipping cost for order i. Also, the order subset I and the shipping days of orders in $N(i-1) = N(i) \setminus \{i\}$ which can be obtained from the shipping plan $\mathbf{z}$ are also optimal for the subproblem of $F(i-1; Q'; Q_1 \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i + 1}, \ldots, Q_{K'})$. Therefore, $F(i; Q'; Q_1, \ldots, Q_{K'})$ can be obtained by minimizing the value of $F(i-1; Q'; Q_1 \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i + 1}, \ldots, Q_{K'}) + G(d_i - \tau_i, q_i)$ over all $\tau_i \in \{1, 2, \ldots, K'\}$ with $\tau_i \leq d_i$ and $q_i \leq Q_{\tau_i}$.

Figure 3.6: Examples for two possible cases for restricted problem RP(K, Q) where $K' = 2$ and $N(i) = \{1, 2, \cdots, 5\}$



(a) Case 1: $F(5; 7; 2, 8)$ equals the sum of $F(4; 4; 2, 8)$ and $G(d_5 - \lceil (Q+7)/c \rceil, 3)$ when order 5 is in I in an optimal solution (i.e., $5 \in I$) (since the shipping day of order 5 is $\lceil (Q+7)/c \rceil$ and $q_5 = 3$).



(b) Case 2: $F(5; 4; 5, 8)$ equals the sum of $F(4; 4; 2, 8)$ and $G(d_5 - 1, 3)$ when the shipping day of order 5 is day 1 in an optimal solution (i.e., $\tau_5 = 1$) (since $\tau_5 = 1$ and $q_5 = 3$).

According to the two cases above, we can compute $F(i; Q_1, Q_2, \cdots, Q_m)$ by the following recursive equation:

$$
\begin{aligned}
&F(i; Q'; Q_1, \ldots, Q_{K'}) \\
&= \min \left\{ \begin{array}{l}
\left\{ \begin{array}{l}
F(i-1; Q'-q_i; Q_1, \ldots, Q_{K'}) + G(d_i - \lceil (Q+Q')/c \rceil, q_i), \\
\hfill \text{if } \lceil (Q+Q')/c \rceil \leq d_i \text{ is satisfied and } q_i \leq Q', \\
+\infty, \text{ otherwise.}
\end{array} \right. \\
\min \left\{ \begin{array}{l}
F(i-1; Q_1, \ldots, Q_{\tau_i - 1}, Q_{\tau_i} - q_i, Q_{\tau_i+1}, \ldots, Q_{K'}) + G(d_i - \tau_i, q_i) \mid \\
\forall \tau_i \in \{1, 2, \ldots K'\} \text{ with } \tau_i \leq d_i \text{ and } q_i \leq Q_{\tau_i}
\end{array} \right\}.
\end{array} \right.
\end{aligned}
\tag{3.38}
$$

where we assume that the value function equals $+\infty$ if it is taking the minimum over an empty set.

Finally, note that $F(Q'; Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q'; Q_1, Q_2, \cdots, Q_{K'})$ with $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$. If $F(n; Q; Q_1, Q_2, \cdots, Q_{K'}) = +\infty$, which means no feasible solution can be found for problem RP(K, Q), we return $+\infty$. Otherwise, we can first find the daily shipping quantity after day K' $(Q_{K'+1}, Q_{K'+2}, \cdots, Q_m)$ by enumerating all $(Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (3.29) and $\sum_{t=1}^{K'} Q_t \leq \bar{Q}'$ satisfied. And find $(Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ that has the minimal values for $F(n; Q'; Q_1, Q_2, \cdots, Q_{K'}) + H(Q_1, Q_2, \cdots, Q_m)$ for all $(Q'; Q_1, Q_2, \cdots, Q_{K'}) \in \mathbb{Z}_+^{K'}$ with (3.29) and $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$ satisfied, where vector $(Q_1, Q_2, \cdots, Q_m)$ can found in the previous step and $H(Q_1, Q_2, \cdots, Q_m)$ is the minimized inventory holding costs given $(Q_1, Q_2, \cdots, Q_m)$ and can be computed by Algorithm 3.3 according to Corollary 3.1. Furthermore, during this process, the solution containing subset I and shipping plan $\mathbf{z}$ that has the minimum value of total shipping costs and inventory holding costs for the subproblem of $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ can also be obtained. Accordingly, from the analysis in Proposition 3.2, I and $\mathbf{z}$ can help to construct a feasible solution $(\mathbf{x}, \mathbf{z}, I)$ to problem RP(K, Q) with the total cost $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'}) + H(Q'_1, Q'_2, \cdots, Q'_m)$. The described procedure is shown in Algorithm 3.6.

Algorithm 3.6 (for problem RP(K, Q))

$F(0;0;0,0,\ldots,0) \leftarrow 0$, and $F(0;Q';Q_1,Q_2,\ldots,Q_{K'}) \leftarrow +\infty$ for each $(Q_1,Q_2,\cdots,Q_{K'}) \in \mathbb{Z}_+^{K'}$ with $0 < Q' + \sum_{t=1}^{K'} Q_t \leq \bar{Q}'$ and (3.29) satisfied and

for all $i = 1,2,\cdots,n$ **do**

  for all $Q' = 0,1,\cdots,\bar{Q}'$ **do**

    for all $(Q_1,Q_2,\cdots,Q_{K'}) \in \mathbb{Z}^{K'}$ with $\sum_{t=1}^{K'} Q_t + Q' \leq \bar{Q}'$ and (3.29) satisfied **do**

      Compute $F(i;Q';Q_1,\ldots,Q_{K'})$ by the recursive equation in (3.38)

    end for

  end for

end for

if $F(n;Q';Q_1,\cdots,Q_{K'}) = +\infty$ for each $(Q';Q_1,\cdots,Q_{K'})$ with $Q' \in \{0,1,\ldots,\bar{Q}'\}$, $(Q_1,\cdots,Q_{K'}) \in \mathbb{Z}_+^{K'}$ and $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$, and (3.29) satisfied **then**

  return $+\infty$

else

  for all $(Q_1,Q_2,\cdots,Q_{K'}) \in \mathbb{Z}^{K'}$ with $\sum_{t=1}^{K'} Q_t \leq \bar{Q}'$ and (3.29) satisfied **do**

    Compute the shipping quantity vector after day K' $(Q_{K'+1},\cdots,Q_m)$ and obtain the daily shipping quantity vector $(Q_1,Q_2,\cdots,Q_m)$

  end for

  Find $(Q'';Q_1',\cdots,Q_{K'}')$ that minimizes the value of $F(n;Q';Q_1,\cdots,Q_{K'})+H(Q_1,\cdots,Q_m)$ among all $(Q';Q_1,\cdots,Q_{K'})$ for each $(Q';Q_1,\cdots,Q_{K'})$ with $Q' \in \{0,1,\ldots,\bar{Q}'\}$, $(Q_1,\cdots,Q_{K'}) \in \mathbb{Z}_+^{K'}$ and $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$, and (3.29) satisfied

  Backtrack the computational process of $F(n;Q'';Q_1',\cdots,Q_K')$ with $\sum_{t=1}^{K'} Q_t' + Q'' = \bar{Q}'$ and construct I and $\mathbf{z}$ with the minimal objective value for the subproblem of $F(n;Q'';Q_1',\cdots,Q_K')$

  Construct a feasible solution $(\mathbf{x},\mathbf{z},I)$ to problem RP(K,Q) according to Proposition 3.2

  return $(\mathbf{x},\mathbf{z},I)$

end if

Theorem 3.6. *For every* $Q \in \{0, 1, \ldots, mc\}$ *and* $K \in \{1, 2, \ldots, m\}$*, (i) Algorithm 3.6 returns* $+\infty$ *when no feasible solution can be found for problem* $\mathrm{RP}(K, Q)$*; (ii) Algorithm 3.6 returns an optimal solution* $(\mathbf{x}, \mathbf{z}, I)$ *to problem* $\mathrm{RP}(K, Q)$ *with running time to be* $O(nmc^{K+1}K \cdot K!)$*, and* $(\mathbf{x}, \mathbf{z})$ *is also a feasible solution to model ILP of problem IPTSDI with the same total cost with problem* $\mathrm{RP}(K, Q)$*.*

Proof. We can use similar logic in the proof of Theorem 3.4 to prove the optimality of Algorithm 3.6 stated in this Theorem. In Algorithm 3.6, it calculates value function $F(i; Q'; Q_1, Q_2, \ldots, Q_{K'})$ recursively by (3.38), and $F(Q_1, Q_2, \ldots, Q_{K'}) = F(n; Q; Q_1, Q_2, \cdots, Q_{K'})$ with $\sum_{t=1}^{K'} Q_t + Q' = \bar{Q}'$. Therefore, $F(n; Q; Q_1, Q_2, \cdots, Q_{K'}) = +\infty$ when there is no feasible solution to problem $\mathrm{RP}(K, Q)$ and Algorithm 3.6 also returns $+\infty$. Otherwise, the minimum total shipping costs and inventory holding costs for problem $\mathrm{RP}(K, Q)$ equals $F(n; Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ for $(Q''; Q'_1, Q'_2, \cdots, Q'_{K'})$ found by Step 15 of Algorithm 3.6. The reason is shown as follows. Suppose the shipping quantities of the optimal solution is $(Q'^{*}; Q^*_1, Q^*_2, \cdots Q^*_m)$. From step 15 of Algorithm 3.6, we can know that for the obtain solution $(Q_1, Q_2, \cdots, Q_m)$, we have $F(n; Q''; Q'_1, Q'_2, \ldots, Q'_{K'}) \leq F(n; Q'^{*}; Q^*_1, Q^*_2, \cdots, Q^*_{K'})$. And from Corollary 3.1, we also have $H(Q_1, Q_2, \cdots, Q_m) = H(Q^*_1, Q^*_2, \cdots, Q^*_m)$. Therefore, the total shipping costs and inventory holding costs of the obtained solution in Algorithm 3.6 is less or equal to the total shipping costs and inventory holding costs of the optimal solution to problem IPTSDI. Therefore, the total shipping costs and inventory holding costs of the obtained solution in Algorithm 3.6 is less or equal to the total shipping costs and inventory holding costs of the optimal solution to problem IPTSDI. Therefore, the optimality of solution $(\mathbf{x}, \mathbf{z}, I)$ holds for problem $\mathrm{RP}(K, Q)$.

The feasibility of $(\mathbf{x}, \mathbf{z})$ also holds for problem IPTSDI due to the definition of problem $\mathrm{RP}(K, Q)$. In addition, the total costs of solutions $(\mathbf{x}, \mathbf{z})$ and $(\mathbf{x}, \mathbf{z}, I)$ are the same since $\mathbf{z}$ can determine both the shipping cost and inventory holding cost.

Moreover, Algorithm 3.6 computes (3.38) for at most $n \cdot mc \cdot (1 \cdot c)(2 \cdot c) \cdots (K' \cdot c) = nmc \cdot c^{K'} \cdot K'!$ times since $Q_t \leq tc$ for $1 \leq t \leq K'$ and $Q' \leq Q \leq \sum_{i \in N} q_i \leq mc$. And the recursive equation (3.26) itself will run in $O(K')$ time. That is, $O(nmc^{K+1}K \cdot K!)$ is the running times of Algorithm 3.6. And thus, Theorem 3.6 is proved. $\square$

3.6.2 Approximation Scheme: Algorithm and Analysis

Illustrated in Algorithm 3.7, we show the approximation scheme for problem IPTSDI based on Algorithm 3.6. At first, For any fixed $\epsilon > 0$, the algorithm selects $K = \min\left\{\lceil(1+\rho)/\epsilon\rceil, m\right\}$. Next, it solves the restricted problem RP(K, Q) by utilizing Algorithm 3.6, which iterates all values of $Q \in \{0, 1, \ldots, \bar{Q}\}$ where $\bar{Q} = mc$ with $K \geq \lceil Q/c \rceil$ and (3.27) satisfied. According to Theorem 3.6, Algorithm 3.6 can construct a feasible solution to problem IPTSDI if a feasible solution exists for problem RP(K, Q). Finally, the algorithm returns the solution with the lowest total shipping costs and inventory holding costs among all feasible solutions in the iteration process.

Algorithm 3.7 (an approximation scheme for problem IPTSDI)

1: For a fixed and positive ϵ, set $K \leftarrow \min\left\{\lceil(1+\rho)/\epsilon\rceil, m\right\}$

2: **for all** $Q \in \{0, 1, \ldots, \bar{Q}\}$ with $K \geq \lceil Q/c \rceil$ **do**

3: Use Algorithm 3.6 to solve the restricted problem RP(K, Q) and return an optimal solution $(\mathbf{x}, \mathbf{z}, I)$ to problem RP(K, Q) if it is feasible. Construct a feasible solution $(\mathbf{x}, \mathbf{z})$ to model ILP with solution $(\mathbf{x}, \mathbf{z}, I)$

4: **end for**

5: **return** the feasible solution that has the lowest total shipping costs and inventory holding costs among all $(\mathbf{x}, \mathbf{z})$ obtained to model ILP

3.6.2.1 Analysis

Lemma 3.5. *Algorithm 3.7 returns in $O(nm^2c^{\lceil 1/\epsilon\rceil+2} \cdot \lceil 1/\epsilon\rceil! \cdot \lceil 1/\epsilon\rceil)$ time, which is a pseudo-polynomial running time for any given $\epsilon > 0$.*

Proof. By Theorem 3.6, every solution $(\mathbf{x}, \mathbf{z})$ obtained in Step 3 of Algorithm 3.7 is feasible to model ILP. Thus, the one returned by Step 5 of Algorithm 3.7 is also a feasible solution to model ILP.

Moreover, Algorithm 3.6 will be invoked for at most mc times since (3.3) and $\bar{Q} = mc$. Thus, from the complexity analysis in Theorem 3.6 and $K \leq \lceil (1+\rho)/\epsilon \rceil$, we can see that Algorithm 3.7 runs pseudo-polynomial running time in $O(nm^2 c^{\lceil 1/\epsilon \rceil + 2} \cdot \lceil 1/\epsilon \rceil! \cdot \lceil 1/\epsilon \rceil)$ time for any given $\epsilon > 0$. $\square$

Based on Lemma 3.5, we can then have Theorem 3.7.

Theorem 3.7. *For any given but fixed $\epsilon > 0$, Algorithm 3.7 is a pseudo-polynomial time approximation scheme for problem IPTSDI with a worst-case performance ratio of $(1+\epsilon)$.*

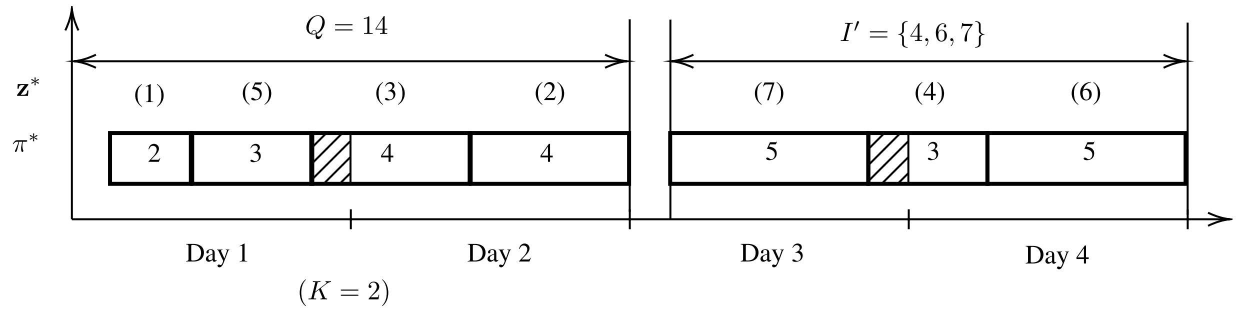
In the proof of Theorem 3.7, we need to show that for any given $\epsilon > 0$, Algorithm 3.7 can yield a feasible solution to problem IPTSDI whose total shipping and inventory holding costs are less or equal to $(1+\epsilon)$ times that of an optimal solution.

3.6.2.2 Proof of Theorem 3.7

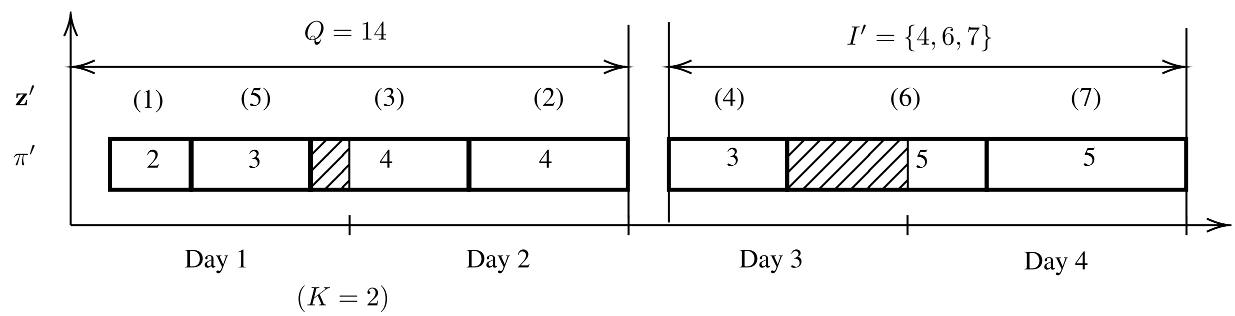
First, we construct a restricted problem RP(K, Q^*) as follows. Recall that $K = \min\{\lceil (1+\rho)/\epsilon \rceil, m\}$. By Theorem 3.1, there must exist an shipping plan $\mathbf{z}^*$ such that the solution $\pi^* = (\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$, which is constructed from $\mathbf{z}^*$ by the Algorithm 3.3 described in Section 3.3, forms an optimal solution to model ILP. We define Q^* as the total product quantity shipped out after day K under the optimal solution π^* (see Figure 3.7(a) for an illustrative example). Thus, $\lceil Q^*/c \rceil$ also indicates the earliest possible day on which the productions are completed for all the products for orders shipped out on or before day K under π^*. Hence, $K \geq \lceil Q^*/c \rceil$, implying that K and Q^* satisfy (3.27).

Consider the restricted problem RP(K, Q^*). Since $0 \leq Q^* \leq mc = \bar{Q}$, during the iteration in Steps 2–4, Algorithm 3.7 must have applied Algorithm 3.6 to solve the restricted problem RP(K, Q^*). Thus, if problem RP(K, Q^*) has a feasible solution, then Algorithm 3.6 must return an optimal solution $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ to it, which yields a feasible solution $\tilde{\pi} = (\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ to model ILP. For each $i \in N$, let $\xi_i(\pi^*)$ and $\xi_i(\tilde{\pi})$ indicate the shipping cost of order i under

Figure 3.7: Illustrative examples for the proof of Theorem 3.7 where $K = 2$ and $d_1 \leq d_2 \leq \ldots \leq d_7$: Defining $\mathbf{z}^*$, σ^*, Q^*, I', $\mathbf{z}'$, σ', and π'.



(a) An optimal solution $\pi^* = (\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*)$ is constructed from the order sequence $\sigma^* = (1, 5, 3, 2, 7, 4, 6)$, and from π^*, Q^* is defined to be the total product quantity of orders shipped out after day $K = 2$ under π^*, and these orders form set $I' = \{4, 6, 7\}$.



(b) From $\mathbf{z}^*$ and $I' = \{4, 6, 7\}$ shown in Figure 3.7(a), a new order sequence $\sigma' = (1, 5, 3, 2, 4, 6, 7)$ and new shipping plan $\mathbf{z}'$ is constructed by rearranging orders of I' in an increasing order of their indices, and from $\mathbf{z}'$ a new solution $\pi' = (\mathbf{x}(\mathbf{z}'), \mathbf{z}')$ is constructed.

the solution π^* and the solution $\tilde{\pi}$, respectively. Similarly, let $\mu_i(\pi^*)$ and $\mu_i(\tilde{\pi})$ denote the inventory holding costs of order i under the solution π^* and the solution $\tilde{\pi}$. Since the solution returned by Step 5 of Algorithm 3.7 must have a total cost no greater than that of solution $\tilde{\pi}$, to prove that Algorithm 3.7 has a worst-case performance ratio of $(1 + \epsilon)$, we only need to prove that problem RP(K, Q^*) has a feasible solution, and that

$$\sum_{i \in N} (\xi_i(\tilde{\pi}) + \mu_i(\tilde{\pi})) \leq (1 + \epsilon) \sum_{i \in N} (\xi_i(\pi^*) + \mu_i(\pi^*)). \tag{3.39}$$

Second, we construct an order subset I' from π^* and a shipping plan $\mathbf{z}'$ from $\mathbf{z}^*$ and I' as follows, which yields a feasible solution π' to model ILP such that π' and I' also form a feasible solution to the restricted problem RP(K, Q^*). We define I' to be the subset of orders shipped after day K under solution π^* (see Figure 3.7(a) for an illustrative example). Accordingly, in sequence σ^* of the shipping plan $\mathbf{z}^*$, the first $(n - |I'|)$ orders are shipped out on or before day K, forming set $N \setminus I'$, and the last $|I'|$ orders are shipped out after day K, forming set I'. From π^* and I', we can construct a new order sequence σ' by changing only the subsequence of the last $|I'|$ orders, such that they are in an increasing order of their indices and form a new shipping plan $\mathbf{z}'$. Following the Algorithm 3.3 described in Section 3.3 we can construct from $\mathbf{z}'$ a solution $\pi' = (\mathbf{x}(\mathbf{z}'), \mathbf{z}')$ for model ILP (see Figure 3.7(b) for an illustrative example). Lemma 3.6 can then be established.

Lemma 3.6. $\pi' = (\mathbf{x}(\mathbf{z}'), \mathbf{z}')$ *is a feasible solution to model ILP of problem IPTSDI, and* $(\mathbf{x}(\mathbf{z}'), \mathbf{z}', I')$ *is a feasible solution to the restricted problem* $\mathrm{RP}(K, Q^*)$.

Proof. Consider the order sequence of the shipping plan $\mathbf{z}^*$ in the optimal solution π^* is $\sigma^* = (\sigma_1^*, \sigma_2^*, \cdots, \sigma_n^*)$ and the order sequence of the shipping plan $\mathbf{z}'$ in the constructed solution π' is $\sigma' = (\sigma_1', \sigma_2', \cdots, \sigma_n')$. Let j indicate the smallest index such that order σ_{j+1}^* appears ahead of order σ_j^* in σ'. If such an index j does not exist, implying that $\sigma^* = \sigma'$, which means $\mathbf{z}^* = \mathbf{z}'$ then since by the definition of I' and Q^*, $(\mathbf{x}(\mathbf{z}^*), \mathbf{z}^*, I')$ is a feasible solution to $\mathrm{RP}(K, Q^*)$, we can see that Lemma 3.6 holds true.

Otherwise, from the definition of σ' we know that $\sigma_{j+1}^* < \sigma_j^*$, which, together with $d_1 \le d_2 \le \cdots \le d_n$, implies that $d_{\sigma_{j+1}^*} \le d_{\sigma_j^*}$. We can construct a new sequence σ'' from σ^* by swapping the positions of orders σ_j^* and σ_{j+1}^*, which leads to a new shipping plan $\mathbf{z}''$. Consider the solution $\pi'' = (\mathbf{x}(\mathbf{z}''), \mathbf{z}'')$ constructed from $\mathbf{z}''$ by the Algorithm 3.3 described in Section 3.3. We now show as follows that π'' is a feasible solution to model ILP.

First, from the backward construction of Algorithm 3.3, we know that under each solution $\pi \in \{\pi^*, \pi''\}$, the production capacity of each day is not exceeded. Second, suppose Ω_i denotes the total idle time before shipping out order i for each $i \in N$. Since $\sigma_{j'}'' = \sigma_{j'}^*$ for $j' \in \{1, 2, \cdots, j-1, j+2, \cdots, n\}$, from the backward and forward construction in Algorithm 3.3, we have $\lceil(\sum_{j'=1}^{j''} q_{\sigma_{j'}''} + \Omega_{\sigma_{j''}''})/c\rceil = \lceil(\sum_{j'=1}^{j''} q_{\sigma_{j'}^*} + \Omega_{\sigma_{j''}^*})/c\rceil$ for $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, n\}$. Thus, since the optimal solution π^* is feasible to model ILP, and since $d_{\sigma_{j''}^*} = d_{\sigma_{j''}''}$, we obtain that for each $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, n\}$,

$$\lceil (\sum_{j'=1}^{j''} q_{\sigma''_{j'}} + \Omega_{\sigma''_{j''}})/c \rceil = \lceil (\sum_{j'=1}^{j''} q_{\sigma^*_{j'}} + \Omega_{\sigma^*_{j''}})/c \rceil \leq d_{\sigma^*_{j''}} = d_{\sigma''_{j''}}. \tag{3.40}$$

Thus, the shipped-out day of each order $\sigma''_{j''}$ with $j'' \in \{1, 2, \cdots, j-1, j+1, \cdots, n\}$ is not later than its committed delivery due date. For the remaining two orders σ''_j and σ''_{j+1}, we can also see that their shipped-out days are not later than their committed delivery due dates. To see this, we first know from (3.40) that $\lceil (\sum_{j'=1}^{j+1} q_{\sigma''_{j'}} + \Omega_{\sigma''_{j+1}})/c \rceil = \lceil (\sum_{j'=1}^{j+1} q_{\sigma^*_{j'}} + \Omega_{\sigma^*_{j+1}})/c \rceil$. For order σ''_j, its shipped-out day is $\lceil (\sum_{j'=1}^{j} q_{\sigma''_{j'}} + \Omega_{\sigma''_j})/c \rceil$, which, due to $\sigma^*_{j+1} = \sigma''_j$ and (3.40), satisfies that

$$\lceil (\sum_{j'=1}^{j} q_{\sigma''_{j'}} + \Omega_{\sigma''_j})/c \rceil \leq \lceil (\sum_{j'=1}^{j+1} q_{\sigma''_{j'}} + \Omega_{\sigma''_{j+1}})/c \rceil = \lceil (\sum_{j'=1}^{j+1} q_{\sigma^*_{j'}} + \Omega_{\sigma^*_{j+1}})/c \rceil \leq d_{\sigma^*_{j+1}} = d_{\sigma''_j},$$

and thus is not later than its committed delivery due date. For order σ''_{j+1}, its shipped-out day is $\lceil (\sum_{j'=1}^{j+1} q_{\sigma''_{j'}} + \Omega_{\sigma''_{j+1}})/c \rceil$, which, due to $d_{\sigma^*_{j+1}} \leq d_{\sigma^*_j}$ and (3.40), satisfies that

$$\lceil (\sum_{j'=1}^{j+1} q_{\sigma''_{j'}} + \Omega_{\sigma''_{j+1}})/c \rceil = \lceil (\sum_{j'=1}^{j+1} q_{\sigma^*_{j'}} + \Omega_{\sigma^*_{j+1}})/c \rceil \leq d_{\sigma^*_{j+1}} \leq d_{\sigma^*_j} = d_{\sigma''_{j+1}},$$

and thus is not later than its committed delivery due date.

Hence, π'' is a feasible solution to model ILP. Replacing σ^* with σ'' and repeating the argument above until $\sigma'' = \sigma'$, we can obtain that the resulting π' is still a feasible solution to model ILP.

Moreover, consider I' and the shipping plans $\mathbf{z}'$ and $\mathbf{z}^*$. By the definition of I' and the construction of π^*, we know that there exists an index n' such that $I' = \{\sigma^*_{n'+1}, \sigma^*_{n'+2}, \cdots, \sigma^*_n\}$, containing all the orders shipped out after day K under $\mathbf{z}^*$, and that $N \backslash I' = \{\sigma^*_1, \sigma^*_2, \cdots, \sigma^*_{n'}\}$, containing all the orders shipped out on or before day K under $\mathbf{z}^*$. Since σ' and σ^* differ only in the subsequence of orders in I', by the construction of π', we know that I' still contains all the orders shipped out after day K under $\mathbf{z}'$, and $N \setminus I'$ still contains all the orders shipped out on or before day K under $\mathbf{z}'$. Accordingly, we can show as follows that the shipped-out days of orders in $\sigma'_{j''} \in N \setminus I'$ under $\mathbf{z}'$, together with the order subset I', satisfy the additional conditions (i)–(iii) of problem $\mathrm{RP}(K, Q^*)$:

- By the definition of Q^*, we know that I' satisfies that the total shipping orders in the set $N \setminus I$ plus the total idle time equal Q^*, which means the additional condition (i) is satisfied.

- For each order $\sigma_{j''} \in I'$ where $j'' \geq n'+1$, its shipped-out day is $\lceil (Q^* + \sum_{j'=n'+1}^{j''} q_{\sigma'_{j'}})/c \rceil$. Since orders in I' are ordered in σ' in an increasing order of their indices, we have $\sum_{j'=n'+1}^{j''} q_{\sigma'_{j'}} = \sum_{i' \in I' : i' \leq \sigma'_{j''}} q_{i'}$. Thus, for each order $i \in I'$, its shipped-out day is $\lceil (Q^* + \sum_{i' \in I' : i' \leq i} q_{i'})/c \rceil$. The additional condition (ii) is satisfied.

- For each order $\sigma'_{j''} \in N \setminus I'$ where $1 \leq j'' \leq n'$, its shipped-out day is $\lceil (\sum_{j'=1}^{j''} q_{\sigma'_{j'}} + \Omega_{\sigma'_{j''}})/c \rceil$, where Ω_i denotes the total idle time before the shipped out day of order i. It can be seen that $\sum_{j'=1}^{j''} q_{\sigma'_{j'}} + \Omega_{\sigma'_{j''}} \leq \sum_{j'=1}^{n'} q_{\sigma'_{j'}} + \Omega_{n'} = Q^*$, implying that $\lceil (\sum_{j'=1}^{j''} q_{\sigma'_{j'}} + \Omega_{\sigma'_{j''}})/c \rceil \leq \lceil Q^*/c \rceil = K'$. Thus, for each order $i \in N \setminus I'$, its shipped-out day must be on or before day K'. The additional condition (iii) is satisfied.

Therefore, $(\mathbf{x}(\mathbf{z}'), \mathbf{z}', I')$ is a feasible solution to $\mathrm{RP}(K, Q^*)$. Lemma 3.6 is proved. □

By Lemma 3.6, we obtain that problem $\mathrm{RP}(K, Q^*)$ has a feasible solution $(\mathbf{x}(\mathbf{z}'), \mathbf{z}', I')$, and that $\pi' = (\mathbf{x}(\mathbf{z}'), \mathbf{z}')$ is a feasible solution to model ILP. Thus, to show that Algorithm 3.7 has a worst-case performance ratio of $(1 + \epsilon)$, we only need to prove (3.39). To prove this, for each $i \in N$, let $\xi_i(\pi')$ indicate the shipping cost of order i under π'. Since $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ is an optimal solution to problem $\mathrm{RP}(K, Q^*)$, by Lemma 3.6, the total shipping costs and inventory holding costs solution of $(\tilde{\mathbf{x}}, \tilde{\mathbf{z}}, \tilde{I})$ should not be greater than that of solution $(\mathbf{x}(\mathbf{z}'), \mathbf{z}', I')$. Thus, the total cost of $\tilde{\pi} = (\tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ should not be greater than that of $\pi' = (\mathbf{x}(\mathbf{z}'), \mathbf{z}')$, implying that

$$\sum_{i \in N} \xi_i(\tilde{\pi}) \leq \sum_{i \in N} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi') + \sum_{i \in I'} \xi_i(\pi'), \tag{3.41}$$

$$\sum_{i \in N} \mu_i(\tilde{\pi}) \leq \sum_{i \in N} \mu_i(\pi') = \sum_{i \in N \setminus I'} \mu_i(\pi') + \sum_{i \in I'} \mu_i(\pi'). \tag{3.42}$$

Moreover, since the positions of orders of $N \setminus I'$ in σ' are the same as that in σ^*, the shipped-out days for orders of $N \setminus I'$ under π' are the same as that under π^*. Thus, we have

$$\sum_{i \in N \setminus I'} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi^*), \tag{3.43}$$

$$\sum_{i \in N \setminus I'} \mu_i(\pi') = \sum_{i \in N \setminus I'} \mu_i(\pi^*). \tag{3.44}$$

From (3.42) and (3.44) we obtain that

$$\sum_{i \in N} \xi_i(\tilde{\pi}) \leq \sum_{i \in N \setminus I'} \xi_i(\pi') + \sum_{i \in I'} \xi_i(\pi') = \sum_{i \in N \setminus I'} \xi_i(\pi^*) + \sum_{i \in I'} \xi_i(\pi'), \tag{3.45}$$

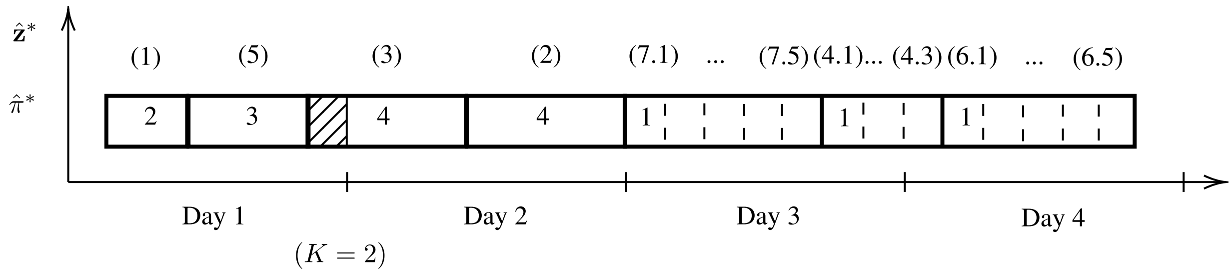
$$\sum_{i \in N} \mu_i(\tilde{\pi}) \leq \sum_{i \in N \setminus I'} \mu_i(\pi') + \sum_{i \in I'} \mu_i(\pi') = \sum_{i \in N \setminus I'} \mu_i(\pi^*) + \sum_{i \in I'} \mu_i(\pi'). \tag{3.46}$$

Third, we construct a new instance of problem IPTSDI by splitting each order $i \in I'$ into q_i orders with each having a unit product quantity and the same committed delivery due date as order i. We denote these unit orders by $(i, 1)$, $(i, 2)$, ..., and (i, q_i). Thus, these unit orders split from order i do not need to be shipped out together.

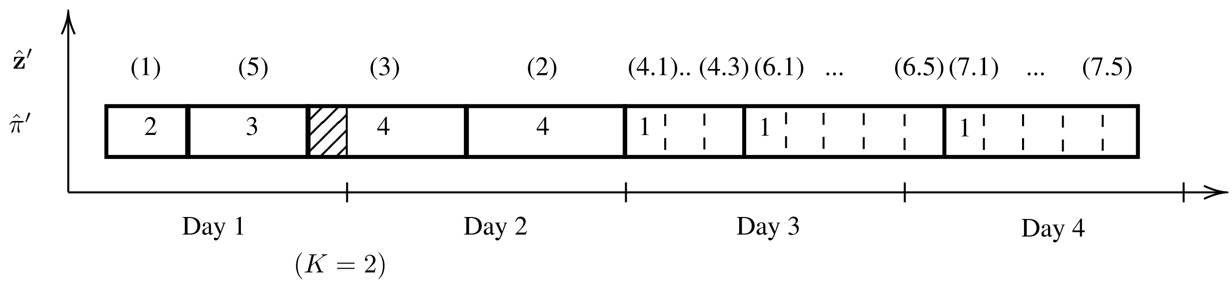
Consider any shipping plan $\mathbf{z}$ with order sequence σ of orders in N such that $\pi = (\mathbf{x}(\mathbf{z}), \mathbf{z})$, which is constructed from $\mathbf{z}$ by Algorithm 3.3 described in Section 3.3, forms a feasible solution to the original problem instance. From σ, we can construct an order sequence $\hat{\sigma}$ of orders for the new problem instance by replacing each order $i \in I'$ in σ with a subsequence of the unit orders $(i, 1)$, $(i, 2)$, ..., and (i, q_i). By the Algorithm 3.3 described in Section 3.3 we can also construct from $\hat{\mathbf{z}}$ a solution $\hat{\pi} = (\hat{\mathbf{x}}(\hat{\mathbf{z}}), \hat{\mathbf{z}})$ for the new problem instance. For each

$i \in I'$, let $\hat{\xi}_i(\hat{\pi})$ indicate the total shipping cost of all the unit orders (i,p) split from order i under $\hat{\pi}$. See Figure 3.8 for two illustrative examples for $\sigma = \sigma^*$ and $\sigma = \sigma'$, respectively. Lemma 3.7 can then be established for $\hat{\pi}$.

Figure 3.8: Illustrative examples for the proof of Theorem 3.7 where $K = 2$ and $d_1 \leq d_2 \leq \ldots \leq d_7$ where orders in $I' = \{4, 6, 7\}$ are split into unit orders $(4,1), \cdots, (4,4), (6,1), \cdots, (6,3), (7,1), \cdots, (7,5)$.



(a) From σ^*, an order sequence $\hat{\sigma^*} = \{1, 5, 3, 2, (7,1), \cdots, (7,5), (4,1), \cdots, (4,4), (6,1), \cdots, (6,3)\}$ is constructed for the new problem instance, and from $\hat{\sigma^*}$ a shipping plan $\hat{\mathbf{z}}^*$ and a solution $\hat{\pi^*} = (\hat{\mathbf{x}}(\hat{\mathbf{z}}^*), \hat{\mathbf{z}}^*)$ is constructed, in which two of the three unit orders split from order 4 are shipped out one day earlier than the production completion days of order 6



(b) From σ', an order sequence $\hat{\sigma'} = \{1, 5, 3, 2, (4,1), \cdots, (4,4), (6,1), \cdots, (6,3), (7,1), \cdots, (7,5)\}$ is constructed for the new problem instance, and from $\hat{\sigma'}$ a shipping plan $\hat{\mathbf{z}}'$ and a solution $\hat{\pi'} = (\hat{\mathbf{x}}(\hat{\mathbf{z}}'), \hat{\mathbf{z}}')$ is constructed, in which four of the five unit orders split from order 6 are shipped out one day earlier than the production completion days of order 4.

Lemma 3.7. *$\hat{\pi}$ is a feasible solution to the new instance of problem IPTSDI, satisfying that $\hat{\xi}_i(\hat{\pi}) \leq \xi_i(\pi)$, $0 = \hat{\mu}_i(\hat{\pi}) \leq \mu_i(\pi)$, for each $i \in I'$.*

Proof. It can be seen that solution $\hat{\pi}$ of the new instance and the solution π of the original instance satisfies the following properties:

(i) For each $i \in N \setminus I'$, the production completion day and the shipped-out day of order i under $\hat{\pi}$ are the same as those under π;

(ii) For each $i \in I'$ and each $p \in \{1, 2, \cdots, q_i\}$, both the production completion day and the shipped-out day of the unit order (i, p) under $\hat{\pi}$ are the same as the day when the first p product units of order i are produced under π.

Due to (ii) above, for each unit order (i, p) split from order $i \in I'$, its product is shipped out as soon as it is produced, and the shipped-out day under $\hat{\pi}$ is no later than that of order i under π, which cannot be later than the committed delivery due date of order i and order (i, p). Also, no inventory holding cost would be incurred when the products are shipped out as soon as they produced, i.e., $0 = \hat{\mu}_i(\hat{\pi}) \leq \mu_i(\pi)$, for each $i \in I'$. This, together with (i) above, implies that the solution $\hat{\pi}$ is feasible to the new instance of problem IPTSDI, and that the total shipping costs and inventory holding costs of all the unit orders (i, p) under $\hat{\pi}$ cannot be greater than the shipping cost of order i under π, i.e., $\hat{\xi}_i(\hat{\pi}) \leq \xi_i(\pi)$ for $i \in I'$. Thus, Lemma 3.7 is proved. □

Applying Lemma 3.7 to sequences σ^* and σ' of orders in N, we can obtain sequences $\hat{\sigma^*}$ and $\hat{\sigma'}$ and shipping plans $\hat{\mathbf{z}^*}$ and $\hat{\mathbf{z}'}$ for the new problem instance, respectively, as well as obtain feasible solutions $\hat{\pi^*} = (\hat{\mathbf{x}}(\hat{\mathbf{z}}^*), \hat{\mathbf{z}}^*)$ and $\hat{\pi'} = (\hat{\mathbf{x}}(\hat{\mathbf{z}}'), \hat{\mathbf{z}}')$ to the new problem instance, respectively, satisfying that

$$\hat{\xi}_i(\hat{\pi^*}) \leq \xi_i(\pi^*) \text{ and } \hat{\xi}_i(\hat{\pi'}) \leq \xi_i(\pi'), \text{ for } i \in I'. \tag{3.47}$$

Moreover, sequence $\hat{\sigma'}$ can be transformed from sequence $\hat{\sigma^*}$ by repetitively interchanging the positions of any two unit orders (i, p) and (i', p') with $i > i'$ or with $i = i'$ and $p > p'$, where $i \in I'$, $p \in \{1, 2, \cdots, p_i\}$, $i' \in I'$, and $p' \in \{1, 2, \cdots, p_{i'}\}$. Note that such two unit orders (i, p) and (i', p') have the same order quantity (which is one). By following an argument similar to that in the proof of Theorem 3.2, we can obtain that the total shipping cost of orders in I', under the solution constructed from the order sequence, is not increased after each interchange of the positions of orders (i, p) and (i', p'). Thus, we have

$\sum_{i\in I'}\hat{\xi}_i(\hat{\pi'}) \le \sum_{i\in I'}\hat{\xi}_i(\hat{\pi^*})$, which, together with (3.47), implies that

$$\sum_{i\in I'}\hat{\xi}_i(\hat{\pi'}) \le \sum_{i\in I'}\hat{\xi}_i(\hat{\pi^*}) \le \sum_{i\in I'}\xi_i(\pi^*). \tag{3.48}$$

Fourth, we are now going to investigate the difference between $\sum_{i\in I'}\xi_i(\pi')$ and $\sum_{i\in I'}\hat{\xi}_i(\hat{\pi'})$, that is, the difference between the total shipping costs of orders in I' under π' and that of unit orders split from orders in I' under $\hat{\pi'}$. For this, we establish Lemma 3.8 below.

Lemma 3.8. $\sum_{i\in I'}(\xi_i(\pi') + \mu_i(\pi')) \le \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi'}) + \hat{\mu}_i(\hat{\pi'})) + \epsilon\sum_{i\in I'}\xi_i(\pi^*)$.

Proof. If $m \le \lceil(1+\rho)/\epsilon\rceil$, i.e., m is bounded by a fixed constant $\lceil(1+\rho)/\epsilon\rceil$, then $K = \min\{\lceil(1+\rho)/\epsilon\rceil, m\} = m$. Thus, by definition, I' is empty, implying that $\sum_{i\in I'}\xi_i(\pi') + \sum_{i\in I'}\mu_i(\pi') = \sum_{i\in I'}\hat{\xi}_i(\hat{\pi'}) + \sum_{i\in I'}\hat{\mu}_i(\hat{\pi'}) + \epsilon\sum_{i\in I'}\xi_i(\pi^*) = 0$. Lemma 3.8 holds true.

Otherwise, $m > \lceil(1+\rho)/\epsilon\rceil$, and thus $K = \min\{\lceil(1+\rho)/\epsilon\rceil, m\} = \lceil(1+\rho)/\epsilon\rceil$. For each $i \in I'$, let τ_i indicate the shipped-out day of order i under solution π'. Since $q_i \le c$, by the definitions of solutions π' and $\hat{\pi'}$, we can see that under $\hat{\pi'}$ the shipped-out day of each unit order (i,p) split from order i for $p \in \{1, 2, \cdots, q_i\}$ must be either $(\tau_i - 1)$ or τ_i. Thus, by $G(s,y) = y(\alpha - \beta s)$ in (3.1), we have

$$\xi_i(\pi') \le \hat{\xi}_i(\hat{\pi'}) + q_i\beta\{[d_i - (\tau_i - 1)] - (d_i - \tau_i)]\} = \hat{\xi}_i(\hat{\pi'}) + \beta q_i, \text{ for } i \in I'. \tag{3.49}$$

Therefore, we can obtain that

$$\sum_{i\in I'}\xi_i(\pi') \le \sum_{i\in I'}[\hat{\xi}_i(\hat{\pi'}) + \beta q_i] = \sum_{i\in I'}\hat{\xi}_i(\hat{\pi'}) + \beta\sum_{i\in I'}q_i. \tag{3.50}$$

Also, from Lemma 3.7, we know that there is no inventory holding cost for solution $\hat{\pi'}$. Thus, we also have the following,

$$\sum_{i\in I'}\mu_i(\pi') \le \sum_{i\in I'}\hat{\mu}_i(\hat{\pi'}) + h\sum_{i\in I'}q_i. \tag{3.51}$$

Since the orders in I' are shipped out after day K under π^*, the total shipping cost $\sum_{i\in I'} \xi_i(\pi^*)$ for these orders cannot be cheaper than $[\alpha-\beta(m-1-K)]\sum_{i\in I'} q_i$. Thus, since $\alpha-\beta(m-1) \geq 0$ stated in (3.2) implies that $\beta \leq \alpha-\beta(m-1-K)]/K$, we can obtain that

$$\beta \sum_{i\in I'} q_i \leq \{[\alpha-\beta(m-1-K)]/K\} \sum_{i\in I'} q_i \;\leq [\sum_{i\in I'} \xi_i(\pi^*)]/K. \tag{3.52}$$

Therefore, by the inequality above, and $K = \lceil (1+\rho)/\epsilon \rceil \geq 1/\epsilon$, we obtain that

$$\begin{aligned}
\sum_{i\in I'}(\xi_i(\pi') + \mu_i(\pi')) &\leq \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + \beta \sum_{i\in I'} q_i + h \sum_{i\in I'} q_i \\
&= \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + (\beta + h) \sum_{i\in I'} q_i \\
&\leq \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + (1+\rho)\beta \sum_{i\in I'} q_i \\
&\leq \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + (1+\rho) \sum_{i\in I'} \xi_i(\pi^*)/K \\
&\leq \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + \epsilon \sum_{i\in I'} \xi_i(\pi^*),
\end{aligned} \tag{3.53}$$

This implies that Lemma 3.8 also holds true when $m > \lceil (1+\rho)/\epsilon \rceil$. And This completes the proof of Lemma 3.8. $\square$

We can now complete the proof of Theorem 3.7 as follows: From (3.46), (3.48), and Lemma 3.8, we can prove that (3.39) holds true as follows:

$$\begin{aligned}
\sum_{i\in N}(\xi_i(\tilde{\pi}) + \mu_i(\tilde{\pi})) &\leq \sum_{i\in N\setminus I'} (\xi_i(\pi^*) + \mu_i(\pi^*)) + \sum_{i\in I'}(\xi_i(\pi') + \mu_i(\pi')) \\
&\leq \sum_{i\in N\setminus I'} (\xi_i(\pi^*) + \mu_i(\pi^*)) + \sum_{i\in I'}(\hat{\xi}_i(\hat{\pi}') + \hat{\mu}_i(\hat{\pi}')) + \epsilon \sum_{i\in I'} \xi_i(\pi^*) \\
&\leq \sum_{i\in N\setminus I'} (\xi_i(\pi^*) + \mu_i(\pi^*)) + \sum_{i\in I'}(\xi_i(\pi^*) + \mu_i(\pi^*)) + \epsilon \sum_{i\in I'} \xi_i(\pi^*) \\
&= \sum_{i\in N}(\xi_i(\pi^*) + \mu_i(\pi^*)) + \epsilon \sum_{i\in I'} \xi_i(\pi^*) \\
&\leq (1+\epsilon) \sum_{i\in N} (\xi_i(\pi^*) + \mu_i(\pi^*)) .
\end{aligned} \tag{3.54}$$

With (3.39) proved and Lemma 3.6 established, as we have explained earlier, Algorithm 3.7 must have a worst-case performance ratio of $(1+\epsilon)$ for any given $\epsilon > 0$. This, together with Lemma 3.5, implies that Algorithm 3.7 is a pseudo-polynomial time approximation scheme for problem IPTSDI with a worst-case performance ratio of $(1+\epsilon)$ for any given $\epsilon > 0$. Hence, Theorem 3.7 is proved.

In addition, let us consider the case where $K = \min\{m, \lceil(1+\rho)/\epsilon\rceil\} = m$. In other words, m is bounded by the constant $\lceil(1+\rho)/\epsilon\rceil$. Thus, by definition, in the restricted problem $R(K, Q)$, there exists a constant $\bar{Q}''$ such that I' is an empty set when $Q = \bar{Q}''$. With this we can show that $\tilde{\pi}$, as well as the solution returned by Algorithm 3.7, must an optimal solution to problem IPTSDI.

3.7 Computational Experiments

In this section, we show the computational experiments for the three algorithms proposed in Section 3.4 and Section 3.6, namely Algorithm 3.4 to deal with the case when the number of possible order quantities η is bounded by a constant and Algorithm 3.5 for the case when the planning horizon m is bounded by a constant and Algorithm 3.7 in the approximation scheme. These algorithms are coded in C++ and all the experiments are carried out on a PC in Windows 10 system with an Intel(R) Core(TM) i7-3770 CPU 3.40GHz CPU and 32 GB of RAM.

Each test instance contains (i) an order set N with the number of orders $n = |N|$ and each order is associated with an order quantity q_i and a committed delivery due date d_i; (ii) a set of possible order quantities E with $\eta = |E|$; (iii) parameters includes the planning horizon m, production capacity c, values of α, β in the shipping cost functions and unit inventory holding cost h and value of ρ. For each combination of n and m given the set E, we follow Li et al. (2022) to randomly generate the test instances in the following way:

(i) Possible order quantity set $E = \{1, 2, \cdots, 10\}$;

(ii) For every $i \in \{1, 2, \cdots, n\}$, q_i is randomly picked from the set E;

(iii) For every $i \in \{1, 2, \cdots, n\}$, d_i is randomly picked from the set $\{1, \ldots, m\}$;

(iv) c is randomly picked from the set $\{c_{min}, c_{min}+1, \ldots, c_{max}\}$ where $c_{min} = \max_{t\in\{1,2,\ldots,m\}} \lceil \sum_{i\in N: d_i \leq t} q_i/t \rceil$ and $c_{max} = \lceil 1.1 c_{min} \rceil$. Feasible solutions for problem IPTSDI will always exist by generating c in this way.

(v) β is randomly picked from the set $\{1, 2, \ldots, 5\}$. To meet condition (3.2), α is randomly picked from the set $\{(m-1)\beta + 1, \ldots, 2(m-1)\beta\}$.

(vi) ρ is an continuous number randomly picked from the interval $[1, 2]$ and h is randomly picked from the set $\{\lceil 0.5 * \rho\beta \rceil, \lceil 0.8 * \rho\beta \rceil + 1, \cdots, \lceil \rho\beta \rceil\}$.

At first, we describe the computational results for Algorithm 3.4. It is a pseudo-polynomial time algorithm for the case when η, the number of possible quantity, is a fixed constant from the analysis of Theorem 3.3. Therefore, the test instances are associated with m, n and η. Moreover, in step (ii) to generate order quantity, we use a subset of E' by randomly picking η numbers from the original set E. And m is from the set $\{5, 10, 15\}$, n is from the set $\{40, 80, 120, 160, 200\}$ and η is from the set $\{3, 4, 5, 6\}$. For each combination of m, n and η, we generate 10 instances. From the results of these test instances, we find that for $\eta \in \{3, 4\}$, Algorithm 3.4 can find the optimal solution averagely in 6.7 seconds over all the cases. However, for $\eta \in \{5, 6\}$ with large value of n ($n \geq 160$), Algorithm 3.4 cannot find the optimal solution due to long running times and insufficient memory of the computer. From these results, we can see that Algorithm 3.4 is efficient in solving problem IPTSDI only for the cases with a small value of η.

Then, we present the results for Algorithm 3.5. It is a pseudo-polynomial time algorithm for the case when m, the planning horizon is bounded by a constant from the analysis of Theorem 3.4. Therefore, the test instances are associated with m and n. And m is from the set $\{2, 3, 4\}$, n is from the set $\{40, 80, 120, 160, 200\}$. For each combination of m and

n, we also generate 10 instances. From the results of these test instances, we find that for $m \in \{2, 3\}$, Algorithm 3.5 can find the optimal solution averagely in 2.3 seconds over all the cases. However, for $m = 4$ with a large value of n ($n \geq 120$), Algorithm 3.5 cannot find the optimal solution due to long running times and insufficient memory of the computer. From these results, we can see that Algorithm 3.5 is efficient in solving problem IPTSDI only for the cases with a small value of m.

Lastly, we present the results for Algorithm 3.7. It is a pseudo-polynomial time approximation scheme with a worst-case performance ratio to be $(1 + \epsilon)$ according to the results of Theorem 3.7 and Lemma 3.5. Therefore, the test instances are associated with m, n and ϵ. And m is from the set $\{5, 10, 15\}$, n is from the set $\{40, 80, 120, 160, 200\}$ and $\epsilon = 100\%$. For each instance, we also obtain a lower bound from an optimal solution to a relaxed problem of IPTSDI. In this problem, deliveries of orders can be split such that products that are completed on the current day can be shipped out. Therefore, there are no inventory holding costs in these optimal solutions. We consider two settings in the experiments for Algorithm 3.7. The unit inventory holding cost is close to 0 in the first setting and that is not close to 0 in the second setting.

Table 3.2 summarizes the results for the approximation scheme. For every test instance, it shows the percentage of the gap of the solution obtained by Algorithm 3.7 and the lower bound described above. These gaps are shown in columns 'Max_G" and "Ave_G" which means maximum and average gaps respectively. Particularly, the gap is calculated by $(ub - lb)/lb \times 100\%$, where lb is the value of the lower bound, and ub is objective value obtained by Algorithm 3.7. In addition, columns "Max_T" and "Ave_T" show the maximum and average running time in seconds. From the results in Table 3.2, we can find that for all the instances with $\epsilon = 100\%$ and $\rho = 1.0$, Algorithm 3.7 can achieve a maximum gap to be 0.59% with a maximum running time to be 57.89s, while the corresponding gap in average decreases to 0.29% with running time decreasing to 20.05s. And for all the instances with $\epsilon = 200\%$ and $\rho = 1.0$, Algorithm 3.7 can achieve a maximum gap to be 0.81% with a maximum running time to be 6.76 seconds while the corresponding gap in average decreases to 0.42% with running time decreasing to 3.87s. By observing the data in Table 3.2, especially the last row that reflects the average of the test instances for a setting, we can find that the gaps of the solution change only in a small range where the running time decrease dramatically with ϵ change from 100% to 200%. Moreover, among all these instances, the maximum value of the maximum gap is 2.29% which is significantly smaller than the worst-case performance ratio $(1 + \epsilon)$. Therefore, through the results of the experiment, we can see that Algorithm 3.7 is capable to generate solutions with high quality and has the potential to perform well in practice.

3.8 Summary

In this chapter, we study problem IPTSDI which is a variant of problem IPTSD by incorporating inventory holding costs. The manufacturer needs to determine the daily production quantity and shipping date for each order before or on its committed delivery due date. The problem is known to be strongly NP-hard in past literature. In addition to the shipping cost that is investigated in previous literature, we also consider the inventory holding costs incurred in the production procedure. The objective of this problem is to minimize these two costs. With the inventory holding costs being taken into account, the problem becomes more complex. The manufacturer needs to balance the shipping costs and inventory holding costs for these orders. Particularly, we prove that there is no finite ratio pseudo-polynomial time approximation algorithm for the problem when the unit inventory holding cost is extremely high. Furthermore, we propose three algorithms to solve the problem. Among them, the first exact algorithm runs in pseudo-polynomial for the case where the number of possible order quantities is fixed and the second exact algorithm runs in pseudo-polynomial for the case where the planning horizon is fixed. The third algorithm is a pseudo-polynomial time approximate scheme algorithm that can solve the problem with a worst-case performance

Table 3.2: Computational results for the approximation scheme.

		$\epsilon = 100\%$ and $\rho = 1.0$				$\epsilon = 200\%$ and $\rho = 1.0$			
m	n	M_G(%)	A_G(%)	M_T(s)	A_T(s)	M_G(%)	A_G(%)	M_T(s)	A_T(s)
5	40	0.60	0.18	1.86	0.43	2.29	0.84	0.43	0.29
	80	0.77	0.28	7.00	4.15	0.90	0.52	2.80	1.89
	120	0.39	0.17	69.91	23.03	0.56	0.32	9.96	5.57
	160	0.23	0.08	306.59	77.92	0.40	0.23	16.14	12.31
	200	0.30	0.15	206.25	108.22	0.43	0.26	34.99	17.34
10	40	1.25	0.53	0.29	0.09	1.40	0.65	0.19	0.10
	80	0.73	0.36	3.66	1.36	0.95	0.50	0.86	0.55
	120	0.36	0.24	23.87	5.75	0.49	0.30	4.14	2.29
	160	0.39	0.25	32.73	14.66	0.44	0.32	6.34	3.56
	200	0.29	0.17	150.77	45.60	0.39	0.21	13.75	7.64
15	40	1.23	0.60	0.22	0.07	1.47	0.71	0.22	0.09
	80	1.04	0.56	2.11	0.46	1.06	0.58	0.55	0.30
	120	0.52	0.35	3.47	1.85	0.59	0.39	1.34	0.81
	160	0.30	0.20	22.76	8.53	0.33	0.23	3.17	1.99
	200	0.38	0.20	36.92	8.70	0.40	0.23	6.52	3.30
Average		0.59	0.29	57.89	20.05	0.81	0.42	6.76	3.87

ratio of $(1+\epsilon)$ for a fixed and positive constant ϵ. The results of computational experiments show that the approximation scheme also has good performance to produce close-to-optimal solutions.

Chapter 4

Conclusions and Future Research

4.1 Conclusions

In this thesis, we consider two variants of the integrated production and transportation scheduling problems by incorporating order acceptance decisions and inventory holding costs, respectively. For these two variants, we propose exact and approximation algorithms for them separately.

In this thesis, the first problem we studied is problem IPTSDA, where the manufacturer needs to determine a production plan, a shipping plan and an order acceptance plan. The original IPTSD is known to be strongly NP-hard and the hardness of complexity can also be applied to problem IPTSDA. For this problem, we develop two exact algorithms that can yield optimal solutions to problem IPTSDA. We further prove that these exact algorithms run in polynomial and pseudo-polynomial times for two practical cases: the case with a fixed number of possible order quantities and the case with a fixed-length planning horizon. By extending the second exact algorithm, we also develop a pseudo-polynomial time approximation scheme for solving problem IPTSDA, which guarantees a worst-case performance ratio of $(1 + \epsilon)$ for any fixed $\epsilon > 0$. According to our computational results, this approximation scheme also performs well in producing close-to-optimal solutions.

The second problem we considered in this thesis is problem IPTSDI. The manufacturer needs to determine the daily production quantity and the shipping date for each order. In addition to the shipping cost, we also incorporate the inventory holding costs incurred in the production and shipping procedures. The objective of this problem is to jointly minimize the shipping costs and inventory holding costs. With the inventory holding costs being taken into account, the problem becomes more complex. Particularly, we prove that there is no finite ratio pseudo-polynomial time approximation algorithm for problem IPTSDI when the unit inventory holding cost is extremely high. To solve problem IPTSDI, we firstly propose a backward-forward construction algorithm that can yield an optimal solution to problem IPTSDI given an optimal shipping plan. Furthermore, based on the backward-forward construction algorithm, we also develop two exact algorithms that run in pseudo-polynomial times for two practical cases. Moreover, to make our algorithms more adaptive, we develop a pseudo-polynomial time approximate scheme algorithm that can solve the problem with a worst-case performance ratio of $(1+\epsilon)$ for any constant $\epsilon > 0$. The results of computational experiments on randomly generated instances show that the approximation scheme also has good performance to produce solutions with high qualities.

The analytical results and exact and approximation algorithms in this thesis also provide insights for practitioners in industries. The approximation schemes with worst-case performance guarantees are applicable for real world problems. Furthermore, more efficient algorithms (e.g., beam search) can be embedded into the approximation scheme to obtain high quality solutions within less time.

4.2 Future Research Directions

One of the interesting topics for future research is to investigate whether there exists a polynomial time approximation scheme for problem IPTSDA and problem IPTSDI. It is also of interest to develop new polynomial time exact algorithms for some special cases of the problem other than those studied in this thesis, such as the case where the total number of possible combinations of order quantities and committed delivery due dates is bounded by a fixed constant. It is also of interest to study more general variations of problem IPTSD, such as those with the committed delivery due dates taken into account as decisions.

Moreover, problem IPTSDA studied in this thesis, aims to minimize the total shipping and rejection cost. With each order's rejection cost replaced by its revenue, the minimization of the total shipping and rejection cost is equivalent to the maximization of the total profit, which equals the total revenue of the accepted orders subtracted by their total shipping cost. Accordingly, the two exact algorithms developed in this paper are still valid. However, for the pseudo-polynomial time approximation scheme developed in this paper, the proof of its worst-case performance ratio for problem IPTSDA under cost minimization is not valid under profit maximization. Therefore, it is worthy to investigate the development and analysis of approximation algorithms for problem IPTSDA under profit maximization in future studies.

In addition, no partial delivery is allowed in the assumptions of problem IPTSDI. Hence, it is also worthwhile to relax this assumption, i.e., the delivery of an order can be split into different days, and study whether a polynomial approximation scheme with a constant worst case performance ratio exists when the unit inventory holding cost is sufficiently small. Furthermore, the shipping cost function considered in this thesis is linearly increasing or non-decreasing in the shipping quantity. Future studies may also consider shipment consolidation, which means shipments with larger quantities would have discounts, i.e., the shipping cost function is no longer linear with the shipping quantity. Under this assumption, shipping costs could be reduced by postponing the delivery of orders since they can be consolidated. However, this would lead to extra inventory holding costs. Therefore, it is of great interest to investigate possible exact algorithms and approximation algorithms with a constant worst case performance ratio for the problem with shipment consolidation.

Furthermore, since some of inbound logistics problems are also involved multiple transportation modes, delivery costs and inventory holding costs that are similar in outbound logistics problems, it is also interesting to examine exact algorithms and approximation schemes for these inbound logistics problems based on the analytical results and algorithms of this thesis.

Finally, future studies can also consider that orders can arrive to the manufacturer during the execution of the production plan and shipping plan determined by previously arrived orders. Although problems in this thesis are deterministic, one can also develop rolling-horizon algorithms in dynamic settings by leveraging the exact and approximation algorithms proposed in this thesis.

Bibliography

Agnetis, A., Aloulou, M.A., Fu, L.L., 2014. Coordination of production and interstage batch delivery with outsourced distribution. European Journal of Operational Research 238, 130–142.

Agnetis, A., Aloulou, M.A., Fu, L.L., 2016. Production and interplant batch delivery scheduling: Dominance and cooperation. International Journal of Production Economics 182, 38–49.

Ahmadi, R., Bagchi, U., Roemer, T.A., 2005. Coordinated scheduling of customer orders for quick response. Naval Research Logistics 52, 493–512.

Aminzadegan, S., Tamannaei, M., Fazeli, M., 2021. An integrated production and transportation scheduling problem with order acceptance and resource allocation decisions. Applied Soft Computing 112, 107770.

Armstrong, R., Gao, S., Lei, L., 2008. A zero-inventory production and distribution problem with a fixed customer sequence. Annals of Operations Research 159, 395–414.

Azadian, F., Murat, A., Chinnam, R.B., 2015. Integrated production and logistics planning: Contract manufacturing and choice of air/surface transportation. European Journal of Operational Research 247, 113–123.

Bachtenkirch, D., Bock, S., 2022. Finding efficient make-to-order production and batch delivery schedules. European Journal of Operational Research 297, 133–152.

Baller, R., Fontaine, P., Minner, S., Lai, Z., 2022. Optimizing automotive inbound logistics: A mixed-integer linear programming approach. Transportation Research Part E: Logistics and Transportation Review 163, 102734.

Berman, O., Wang, Q., 2006. Inbound logistic planning: minimizing transportation and inventory cost. Transportation science 40, 287–299.

Calosso, T., Cantamessa, M., Vu, D., Villa, A., 2003. Production planning and order acceptance in business to business electronic commerce. International Journal of Production Economics 85, 233–249.

Chan, L.M.A., Muriel, A., Shen, Z.J., Simchi-Levi, D., 2002. On the effectiveness of zero-inventory-ordering policies for the economic lot-sizing model with a class of piecewise linear cost structures. Operations Research 50, 1058–1067.

Chen, B., Lee, C.Y., 2008. Logistics scheduling with batching and transportation. European Journal of Operational Research 189, 871–876.

Chen, Z.L., 2004. Integrated Production and Distribution Operations. Springer US, Boston, MA. pp. 711–745.

Chen, Z.L., 2010. Integrated production and outbound distribution scheduling: review and extensions. Operations Research 58, 130–148.

Chen, Z.L., Hall, N.G., 2007. Supply chain scheduling: Conflict and cooperation in assembly systems. Operations Research 55, 1072–1089.

Chen, Z.L., Pundoor, G., 2006. Order assignment and scheduling in a supply chain. Operations Research 54, 555–572.

Chen, Z.L., Vairaktarakis, G.L., 2005. Integrated scheduling of production and distribution operations. Management Science 51, 614–628.

Chevroton, H., Kergosien, Y., Berghman, L., Billaut, J.C., 2021. Solving an integrated scheduling and routing problem with inventory, routing and penalty costs. European Journal of Operational Research 294, 571–589.

Cohen, M.A., Lee, H.L., 1988. Strategic analysis of integrated production-distribution systems: models and methods. Operations research 36, 216–228.

Erengüç, Ş.S., Simpson, N.C., Vakharia, A.J., 1999. Integrated production/distribution planning in supply chains: An invited review. European Journal of Operational Research 115, 219–236.

Garey, M.R., Graham, R.L., Johnson, D.S., 1978. Performance guarantees for scheduling algorithms. Operations Research 26, 3–21.

Geismar, H.N., Murthy, N.M., 2015. Balancing production and distribution in paper manufacturing. Production and Operations Management 24, 1164–1178.

Ghaleb, M., Zolfagharinia, H., Taghipour, S., 2020. Real-time production scheduling in the industry-4.0 context: Addressing uncertainties in job arrivals and machine breakdowns. Computers & Operations Research 123, 105031.

Goetschalckx, M., Vidal, C.J., Dogan, K., 2002. Modeling and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms. European Journal of Operational Research 143, 1–18.

Guo, Z., Shi, L., Chen, L., Liang, Y., 2017. A harmony search-based memetic optimization model for integrated production and transportation scheduling in mto manufacturing. Omega 66, 327–343.

Hajiaghaei-Keshteli, M., Aminnayeri, M., Ghomi, S.F., 2014. Integrated scheduling of production and rail transportation. Computers & Industrial Engineering 74, 240–256.

Hall, N.G., Potts, C.N., 2003. Supply chain scheduling: Batching and delivery. Operations Research 51, 566–584.

Han, D., Yang, Y., Wang, D., Cheng, T., Yin, Y., 2019. Integrated production, inventory, and outbound distribution operations with fixed departure times in a three-stage supply chain. Transportation Research Part E: Logistics and Transportation Review 125, 334–347.

Herrmann, J.W., 2006. Handbook of production scheduling. volume 89. Springer Science & Business Media.

Hwang, H.C., 2010. Economic lot-sizing for integrated production and transportation. Operations Research 58, 428–444.

Ivănescu, V.C., Fransoo, J.C., Bertrand, J.W.M., 2006. A hybrid policy for order acceptance in batch process industries. OR Spectrum 28, 199–222.

Jiang, D., Tan, J., Li, B., 2017. Order acceptance and scheduling with batch delivery. Computers & Industrial Engineering 107, 100–104.

Kolisch, R., 1998. Integrated production planning, order acceptance, and due date setting for make-to-order manufacturing, in: Operations Research Proceedings 1997, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 492–497.

Korpela, J., Kyläheiko, K., Lehmusvaara, A., Tuominen, M., 2002. An analytic approach to production capacity allocation and supply chain design. International Journal of Production Economics 78, 187–195.

Kreipl, S., Pinedo, M., 2004. Planning and scheduling in supply chains: an overview of issues in practice. Production and Operations management 13, 77–92.

Leung, J.Y.T., Li, H., Pinedo, M., 2005. Order scheduling models: An overview, in: Kendall, G., Burke, E.K., Petrovic, S., Gendreau, M. (Eds.), Multidisciplinary Scheduling: Theory and Applications, Springer US, Boston, MA. pp. 37–53.

Li, F., Chen, Z.L., Tang, L., 2017. Integrated production, inventory and delivery problems: Complexity and algorithms. INFORMS Journal on Computing 29, 232–250.

Li, F., Xu, S., Xu, Z., 2022. New exact and approximation algorithms for integrated production and transportation scheduling with committed delivery due dates and order acceptance. Working paper .

Li, F., Xu, Z., Chen, Z.L., 2020. Production and transportation integration for commit-to-delivery mode with general shipping costs. INFORMS Journal on Computing 32, 855–1186.

Li, K., Ganesan, V., Sivakumar, A., 2005. Synchronized scheduling of assembly and multi-destination air-transportation in a consumer electronics supply chain. International Journal of Production Research 43, 2671–2685.

Li, K., Jia, Z., Leung, J.Y.T., 2015. Integrated production and delivery on parallel batching machines. European Journal of Operational Research 247, 755 – 763.

Li, K., Sivakumar, A.I., Ganesan, V.K., 2008. Complexities and algorithms for synchronized scheduling of parallel machine assembly and air transportation in consumer electronics supply chain. European Journal of Operational Research 187, 442–455.

Liou, C.D., Hsieh, Y.C., 2015. A hybrid algorithm for the multi-stage flow shop group scheduling with sequence-dependent setup and transportation times. International Journal of Production Economics 170, 258–267.

Liu, L., Liu, S., 2020. Integrated production and distribution problem of perishable products with a minimum total order weighted delivery time. Mathematics 8, 146.

Ma, H., Chan, F.T., Chung, S.H., 2013. Minimising earliness and tardiness by integrating production scheduling with shipping information. International Journal of Production Research 51, 2253–2267.

Mangan, J., Lalwani, C., 2016. Global logistics and supply chain management. John Wiley & Sons.

Marasco, A., 2008. Third-party logistics: A literature review. International Journal of Production Economics 113, 127–147.

Melo, R.A., Wolsey, L.A., 2010. Optimizing production and transportation in a commit-to-delivery business mode. European Journal of Operational Research 203, 614–618.

Mensendiek, A., Gupta, J.N., Herrmann, J., 2015. Scheduling identical parallel machines with fixed delivery dates to minimize total tardiness. European Journal of Operational Research 243, 514–522.

Pundoor, G., Chen, Z.L., 2005. Scheduling a production–distribution system to optimize the tradeoff between delivery tardiness and distribution cost. Naval Research Logistics 52, 571–589.

Rom, W.O., Slotnick, S.A., 2009. Order acceptance using genetic algorithms. Computers & Operations Research 36, 1758–1767.

Roundy, R., Chen, D., Chen, P., Çakanyildirim, M., Freimer, M.B., Melkonian, V., 2005. Capacity-driven acceptance of customer orders for a multi-stage batch manufacturing system: models and algorithms. IIE Transactions 37, 1093–1105.

Sarmiento, A.M., Nagi, R., 1999. A review of integrated analysis of production-distribution systems. IIE Transactions 31, 1061–1074.

Sarvestani, H.K., Zadeh, A., Seyfi, M., Rasti-Barzoki, M., 2019. Integrated order acceptance and supply chain scheduling problem with supplier selection and due date assignment. Applied Soft Computing 75, 72–83.

Sawik, T., 2009. Coordinated supply chain scheduling. International Journal of Production Economics 120, 437–451.

Sawik, T., 2016. Integrated supply, production and distribution scheduling under disruption risks. Omega 62, 131–144.

Shams, H., Salmasi, N., 2014. Parallel machine scheduling problem with preemptive jobs and transportation delay. Computers & Operations Research 50, 14–23.

Slotnick, S.A., 2011. Order acceptance and scheduling: A taxonomy and review. European Journal of Operational Research 212, 1–11.

Stecke, K.E., Zhao, X., 2007. Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. Manufacturing & Service Operations Management 9, 206–224.

Sun, X., Chung, S.H., Chan, F.T., 2015. Integrated scheduling of a multi-product multi-factory manufacturing system with maritime transport limits. Transportation Research Part E: Logistics and Transportation Review 79, 110–127.

Tang, L., Li, F., Chen, Z.L., 2019. Integrated scheduling of production and two-stage delivery of make-to-order products: Offline and online algorithms. INFORMS Journal on Computing 31, 493–514.

Tarhan, İ., Oğuz, C., 2022. A matheuristic for the generalized order acceptance and scheduling problem. European Journal of Operational Research 299, 87–103.

Ullrich, C.A., 2013. Integrated machine scheduling and vehicle routing with time windows. European Journal of Operational Research 227, 152–165.

Vidal, C.J., Goetschalckx, M., 1997. Strategic production-distribution models: A critical review with emphasis on global supply chain models. European journal of operational research 98, 1–18.

Wang, H., Lee, C.Y., 2005. Production and transport logistics scheduling with two transport mode choices. Naval Research Logistics 52, 796–809.

Wang, X., Cheng, T., 2009. Logistics scheduling to minimize inventory and transport costs. International Journal of Production Economics 121, 266–273.

Wang, Y., Chen, F., 2019. Packed parts delivery problem of automotive inbound logistics with a supplier park. Computers & Operations Research 101, 116–129.

Yang, X., Liu, Z., Li, F., Xu, Z., 2021. Integrated production and transportation scheduling with order-dependent inventory holding costs. Computers & Operations Research 136, 105477.

Zhang, S., Song, H., 2018. Production and distribution planning in Danone Waters China Division. Interfaces 48, 578–590.

Zhong, W., 2015. An improved algorithm for integrated production and distribution scheduling problem with committed delivery dates. Optimization Letters 9, 537–567.

Zhong, W., Chen, Z.L., Chen, M., 2010. Integrated production and distribution scheduling with committed delivery dates. Operations Research Letters 38, 133–138.

Zhong, X., Fan, J., Ou, J., 2022. Coordinated scheduling of the outsourcing, in-house production and distribution operations. European Journal of Operational Research 302, 427–437.