Three Game-Theoretic Models in Operations Management

ZHANG, Feng

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Systems Engineering and Engineering Management

The Chinese University of Hong Kong
September 2010

Thesis/Assessment Committee

Professor Chen Youhua (Chair)

Professor Cai Xiaoqiang (Thesis Supervisor)

Professor Zhou Xiang (Committee Member)

Professor Zhou Xian (External Examiner)

論文評審委員會

陳友華 教授 （主席）

蔡小強 教授 （論文導師）

周翔 教授 （委員）

周憲 教授 （校外委員）

ABSTRACT

This thesis investigates three problems in operations management, by using different concepts and techniques in *Game Theory*. The first problem is a two-echelon supply chain problem involving wholesaling, transporting and retailing of certain kind of perishable product. A key characteristic of the problem is that the upstream supplier adopts a Group Buying Scheme (GBS) as his pricing mechanism and the downstream retailers, taking into consideration of the supplier's pricing mechanism, their respective market demands and other retailers' likely reactions, compete with each other to maximize their profit respectively. We model this problem as a Stackberg game where supplier is the leader and retailers are the followers. Furthermore, the retailers' optimal ordering problem is solved by applying the solution concepts in *Competition Game Theory* and we prove that the Nash equilibrium always exists. Moreover, the equilibrium is the only Pareto optimal Nash equilibrium and a strong equilibrium as well. Finally we show that the GBS pricing mechanism, as compared with the traditional Flat Price scheme, can bring the supplier and retailers to a win-win situation.

The second is a project management problem with task subcontracting. The project owner (PO) outsources the tasks in his project to different subcontractors (SCs), with contracts to govern the completions of the tasks and the associated costs and bonus. We model the subcontractors' task processing problem as a *Cooperative Game* so that subcontractors can benefit by resource sharing and execution time rescheduling. We prove that our cooperative game is balanced and propose a core allocation vector constructed from the optimal dual solution.

Meanwhile, the project owner's optimal strategy to design the contracts is also obtained by implicit optimization skills.

The third problem we consider concerns about manufacturing outsourcing, where multiple manufacturers outsource their jobs to a third-party firm. The manufacturers book time windows from the third-party to process their jobs whose processing times are stochastic. Due to the capacity limitation of the third-party and the uncertainty in their processing times, it may be beneficial for the manufacturers to cooperate, provided that a proper cooperative mechanism can be devised. We model this problem as a *Cooperative Game*. However, it is more than a *Sequencing Game* commonly studied in the literature, because we consider the optimal booking decisions and the random processing times, which make it possible for the manufacturers to achieve a risk pooling effect by collaborating and booking together. We prove that the outsourcing game is balanced in the situation where the unit booking cost for each time window is unique. We also construct a core allocation based on the core vector derived form a *Permutation Game*. A main breakthrough is that the connective admissible rearrangement assumption is removed for the stochastic sequencing/booking game, following Slikker's technique.

摘要

本论文运用博弈论中的相关概念和技术，研究了运筹管理学领域中的三个实际问题。 第一个问题是一个包含批发，运输和零售的二级供应链问题。在这个问题中，一个上游供应商会利用“逢低买入”（GBS）这种价格机制向多个下游零售商出售某种易腐烂的食品。 而每个下游零售商通过综合考虑供应商提供的价格机制，各自的市场需求以及其他零售商的可能策略来制定自己的决策，从而最大化自己的利润。 我们用Stackberg博弈模型来研究这个问题，其中批发商是主导者，而零售商是跟随者。进一步地，我们将零售商的最优化问题建模为竞争博弈问题, 并且证明该问题不仅存在纳什均衡（Nash equilibrium），而且该均衡是唯一的帕累托最优(Pareto Optimal)均衡和强均衡（strong equilibrium）。最后我们证明，对比传统的固定价格的机制，“逢低买入”这种价格机制可以为供应商和零售商带来共赢的结果。

我们研究的第二个问题是一个涉及到项目外包的工程管理问题。在这个问题中，工程拥有者会将该工程中不同的 项目外包给不同的承包商，并制定合约来规定项目的完成时间、经费、奖金等各项指标。我们将所有承包商的项目执行问题建模为一个合作博弈的问题，从而承包商可以通过资源共享以及项目重新调度等方式来获取更多的利润。我们证明了这个合作博弈问题是平衡（balanced）的并且利用对偶问题的最优解构造了一组核心（core）分配向量。最后我们用隐优化的技巧 找到了工程拥有者在合同制定问题上的最优决策。

最后我们研究了一个制造业外包问题。在该问题中，多个制造商会将各自的业务（可多于一个）外包给同一个第三方公司。各个制造商 会向第三方公司预定时间窗口来制造他们的产品，但产品制造的时间是随机的。因为第三方

公司的产能是有限的，并且考虑到产品制造时间的随机性，制造商们或许能通过合作来降低各自的制造费用，但前提是能设计出合理有效的合作机制。我们将这个问题建模为一个合作博弈问题。但我们这里的博弈问题并不是一些文献中所研究的“排序博弈”（sequencing game）问题。因为在我们的问题中，由于制造时间是随机的，我们需要额外考虑每个制造商预定时间窗口数量的问题。而这正式合作能为制造商带来的“聚集效应”（risk pooling effect）目前我们的研究结果还局限在一种特殊的情况下，即我们假设每个时间窗口的预定费用都是相同的。在这种情况下，我们可以证明制造商的合作博弈问题是平衡的。并且利用“排列博弈”（permutation game）问题核存在性的结果，我们可以建立一组核心分配向量。在以往的有关随机排序/时间预定博弈的文献中，“只有能连通的参与者才能合作”是一个基本的假设。我们的一个主要的突破是，通过运用Slikker的技巧，我们能将这个假设去除。

ACKNOWLEDGEMENTS

My utmost gratitude goes to my supervisor, Prof. Cai Xiaoqiang for four-year guidance on my research work, for his expertise, kindness, and most of all, for his trust. Through the four-year program in Hong Kong, Professor Cai not only gave me a wealth of advice on both studying and growing but also provided me various of opportunities to contact with the industry and society, from which I realized the great aim in my life, and the determination to attain it. My thanks and appreciation also goes to Professor Zhou Xian and Professor Nicholas G. Hall, both of whom kindly and patiently offered me guidance on my researches in different areas. Furthermore, I would like to express my deepest appreciation to my thesis committee members Professor Chen Youhua and Professor Zhou Xiang. I am greatly indebted to them for helpful suggestions and consistently support to my research and thesis writing. Last but not least, I own my thanks to my father and my mother who always stood beside me and encouraged me constantly during my whole life.

Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

Operations Management(OM) is an area of business concerned with managing the production of goods and services to ensure that the process is efficient in terms of using as little resource as needed, and effective in terms of meeting customer requirements. Research in Operations Management has been started in early nineteenth century and the aim is to advance both the theory and practice of operations management in varies industry areas, such as *Supply Chain Management*and *Project Management*.

Supply Chain Management(SCM) encompasses the planning and management of a network of interconnected businesses which is involved in providing products and services to the end customers. It always includes the crucial components of coordination and collaboration with channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. The traditional studies in this area are mainly focused on the single channel management problem in which single participant in each echelon of the supply chain is a pre-assumption. In these studies, rational profit sharing schemes are designed to motivate cooperation among partners and to form centralized system, so that chain profit can be maximized and chain members can benefit. However in practice, a more popular situation is that multiple participants will be involved in each echelon of the supply chain. These participants could be downstream companies who fight for the contract from the upstream manufacturer, or they could be the intermediate retailers who collaborate with each other to grasp more market sales.

Project Management(PM) involves the means, techniques and the concepts that are used to run a specific project and achieve its objectives subject to various constraints/limits on resources, schedules, and personnel. The major components in managing a typical project includes project selection & organization, task analysis, project scheduling & budgeting, research management and project execution & control. Research in project management usually will focus on a single component as listed above and the aim is to complete the project by using as little "resource" as possible, while all the objectives can still be achieved. One major assumption in the traditional research in this area is that the management work of the project is implemented by a single project owner/manager. Hence the PM problem for the researchers usually become an optimization problem with numbers of constraints on resources, schedules and so on. And the objective of the optimization problem usually is to maximize the project owner/manager's profit. But with the industry development and specialization, a project, nowadays in most situation, must be implemented by different teams of workers from different departments, firms or even industries. Every team involved in this project has his own objectives by completing his part in this project and faces his own resource, schedule and personnel constraints. The management of the project under such multi-participant situations will be more complicate than traditional ones. And how to balance the return of different project participants so that everyone can be satisfied becomes the main issue in these studies.

As discussed in the two areas of operations management, a common and interesting situation in this field is that multiple parties are now be involved in the same business while the target of each party is self-benefit maximization. Note that in such situations, an individual's success in making choices depends on the choices of others. Therefore to study such problems we need to apply a new mathematical tool: *Game Theory*. As we know, there are two major concepts in Game theory, which are *Cooperative Game* and *Competition Game*.

Through this thesis we plan to show some applications of both the two games in the study of SCM problem.

A *Competition Game* is a game where a player, taking into consideration of other players' likely performance, works independently to achieve the optimal payback. Hence the players never cooperate with each other. "Nash equilibrium", which is named after J. Nash who proposed it, is an important solution concept in competition game involving multiple players, where each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his or her own strategy unilaterally. A set of strategy choices and the corresponding payoffs is said to constitute a Nash equilibrium when no player can benefit by changing his own strategy while other players keep theirs unchanged.

A *Cooperative Game* is a game where groups of players (coalitions) may enforce cooperative behavior, hence the game is a competition between coalitions of players, rather than between individual players. A cooperative game (N, v) is given by specifying a value $v(S)$ for any coalition $S \subseteq N$, where N is called the grand coalition and v is the characteristic function defined as $v : 2^N \to \mathbb{R}$ and $v(\emptyset) = 0$. Among numbers of solution concepts in cooperative game, the core is a most significant one which is a set of payoff (allocation) vectors defined as follows

$$Core(v) = \{x \in \mathbb{R}^N : \sum_{i \in S} x_i \geq (\leq) v(N), \forall S \subseteq N; \sum_{i \in N} x_i = v(N)\}$$

From the definition above we can observe that if the core exists in a cooperative game, then the players will form the grand coalition since no player can benefit by deviating from the grand coalition and forming sub-coalitions. But the core of a cooperative game may be empty: see the Bondareva-Shapley theorem in Bondareva et al. [6] and Shapley et al. [43] where they showed that the cooperative game has a non-empty core if and only if the game is balanced. A cooperative game is called "Convex" if the characteristic function v is supermodular

(submodular)

$$v(S \cup T) + v(S \cap T) \geq (\leq) v(S) + v(T), \forall S, T \subseteq N$$

Shapley et al [44] proved that the convex games always have non-empty cores However, convexity is not a necessary condition for a cooperative game to have a non-empty core The "Linear Production Game introduced by Owen et al [37] and "Permutation Games" presented in Tijs et al [46], which are not necessarily convex, are also proved to be cooperative games with non-empty cores

Based on the definitions above and works have been done in the literature, in this thesis we propose three problems which will be modeled as different game problems Since detail introductions to this problems can be unfolded in the following chapters, here we just give a brief description to the major problems and the approaches planed to exercise to solve these problems Generally, they are listed as follows

- The application of Group-Buying Scheme(GBS) in a two-echelon supply chain with perishable product In this problem, upstream supplier sells a kind of perishable product to multiple downstream retailers using the GBS pricing mechanism Retailers order the products from the supplier and then sell the products in their markets respectively Regarding to the GBS pricing mechanism, retailers will compete with each other to maximize their own profit and the supplier, taking into consideration of the retailers' order actions, maximize his profit through adjusting the price curve in GBS We try to study the supplier and retailers' performance assuming that retailers' market demand are independent from each other so that extra effort should only be made during the ordering procedure Further more we want to verify whether the GBS can coordinate the two parties by bringing them a win-win situation compared with the traditional flat price mechanism Solution concepts in *Competition Game* will be utilized to find the retailers' optimal ordering strategies
- The application of *Linear Programming Game* In this problem, a Project

Owner (PO) outsource the tasks in his project to multiple Subcontractors (SCs) and sign contracts with them specifying the clauses on completing the tasks, e.g., the required starting and finishing time of each task, the normal payment a SC can receive by completing the task and so on. A tricky issue is that the PO gives the subcontractors incentives to finish the tasks earlier than normal schedule so that he can receive a cost saving on tasks' completion times. Therefore the SCs have another option to perform task crashings to cut down the tasks' processing times to generate more profits. Since task crashing consumes resources and there are resource capacity restrictions, the SCs will cooperate with each other to achieve higer profit through sharing resources and rescheduling the project. To study the interactions among the SCs and the optimal strategy of PO on designing the contracts, we model the problem as a *Linear Programming Game* (Owen et al. [37]'s *Linear Production Game* is a typical linear programming game). We are interested in whether this game is balanced and how to find the core or how to construct an allocation vector that lies in the core.

- The application of "Sequencing Game" in a manufacturing outsourcing problem. In fact, the idea comes from the above project subcontracting problem when we tried to investigate the multi-project case and realized that the problem in which multiple players outsource jobs to the same third-party firm is quite common in practice. Generally in this problem, we have a single third-party firm who can process some types of jobs but the production capacity is limited. The third-party will announce his available production time windows and associated booking cost for each time window. The manufacturers then will reserve these time windows for processing their jobs, with a first come first book (FCFB) policy. Besides the booking cost, we assume each manufacturer also bears a weighted flow time cost on his jobs. Therefore in order to reduce the total cost, the manufacturers will cooperate with each other to achieve a cost saving generated from job sequence optimization. In order to study the cooperation behaviors of the manufacturers, we will apply the cooperative game theory. The solution approach is largely inspired by the "Sequencing Games". But our game model is more complicated since we consider a stochastic job processing time, which leads to new issues on optimal booking quantity of time windows. It can be seen that with stochastic job processing time, manufacturers are more willing to cooperate because they can achieve a risk pooling effect by booking time windows together. Again we are interested in finding the core of these cooperative game.

Through the studies on the above problems, we can observe the power of game theory in solving operations management problems with multiple participants. In fact in the recent literature, we can also find a number of studies focusing on this area. Therefore we can conclude that Game Theory application in OM is an interesting and promising research area and it will attract more scholars to join in. The rest of this thesis is organized as follows: in Chapter 2, we introduce the SCM problem with perishable product and use solution concepts in competition game to study the optimal strategies of the chain members. Then we introduce our "Linear Project Game" (LPG) in Chapter 3 to solve the project subcontracting problem. And the last problem regarding to manufacturing outsourcing game is investigated in Chapter 4. Finally we will make a general conclusion in Chapter 5 on the results we obtained so far on the applications of games theory in OM problems. We will also propose some future directions of this research area and discuss the difficulties.

CHAPTER 2

Group-Buying Scheme in Supply Chain with Perishable Product

In this chapter, we focus on a supply chain problem in which some retailers procures from the supplier a type of perishable products whose quality is decaying on time. And the retailers create their revenue from selling these products to the downstream markets. We try to study the supplier and retailers' optimal performances under the Group-Buying Scheme(GBS). We apply *Competition Game Theory* to solve this two-echelon supply chain problem and obtain some valuable results, including the different parties' optimal performances under GBS and the comparison of profits they obtained under traditional flat price and GBS. The results indicate that both the supplier and retailers will benefit from the GBS pricing mechanism.

2.1. Introduction

Perishability is a common characteristic of many products such as fruit, vegetable, meat, flowers or even more abstract objects like air tickets. In most cases, this characteristic is unfavorable because it means the product will decay in quality or quantity and its market will decline. Therefore, how to arrange the production, distribution and selling of such products to achieve their best value is always an important but challenging problem in both industry and academic such as supply chain management. As known, pricing is one of the key issues in supply chain management even when the products are not perishable. By adding the deteriorating factor, this issue becomes more complex. Meanwhile, if the supply chain involves multiple members, cooperation and competition are also two important factors to be studied. Our model, basically a two-echelon supply chain model with perishable products, consists all these issues to be studied. The supplier who owns certain amount of products needs to set his wholesale price scheme and the retailers' objectives are to decide their order times, order quantities and prices respectively according to the wholesale price scheme released, taking account of the uncertain market demands as well as the likely actions of the other retailers. In particular, we want to verify whether the Group-Buying Scheme(GBS) can benefit both the two parties of supplier and retailers, so that the supplier will prefer it over the traditional Flat Pricing Scheme(FPS).

While we are studying a new model characterized by a supply chain that involves perishable products and GBS, our model and approaches have been largely inspired by many early works in the research areas of perishable product supply chain management and Group-Buying. We next give a brief review to the related literature below.

Studies on the supply chain management of perishable products was started with the concerns about inventory management of such products, which is to analyze and determine the replenishment policies for inventory. Early works can be found in Nahmias et al. [35] who provides a comprehensive survey of research published before the 1980s. More recent studies on the deteriorating inventory models can be referred to Raafat et al. [40] and Goyal and Giri et al. [26], which review the relevant literature published in 1980s and 1990s respectively. The main difference between our model and all the models appeared in the literature is that our model studies a multi-retailer problem instead of single-retailer problem. Therefore our solution approaches require knowledge and techniques in Game Theory, which is fundamentally different from the above works in the literature. Among others, our model is greatly inspired by Cai, Chen and Xu et al. [16], which considers a fresh product supply chain management problem involving long distance transportation. We use a similar demand function in our model and utilize their solution approach and techniques when we study the decentralized system. The main difference is that their model allows only one retailer who just considers the profit maximization problem regarding to the relevant information on wholesale price scheme and market demand, where as in our model deal with multiple retailers whose profit can be affected by other retailers' ordering performance. As a result, we have to use competitive game theory to solve out problem.

GBS is a type of pricing mechanism under which a buyer is guaranteed to get the product at the present price, and can further get a fund at the end of the selling period if the total accumulated order volume received by the supplier exceeds a pre-specified level. Usually supply chain problems with such pricing mechanism are modeled as a Stackelberg game (see Gibbons et al. [25]) and the supplier acts like the leader to determine the parameters of GBS. Essentially this novel scheme is designed to facilitate the coordination between suppliers and retailers. There are some famous pricing mechanisms in the literature which help the chain members to coordinate with each other and obtain more profits, such as the revenue sharing in Cachon et al. [14], backup agreements in Eppen et al. [24], buy back or return policy in Pasternack et al. [38] and quantity flexibility in Tsay et al. [47]. The difference between GBS and the mechanisms mentioned above is that under GBS, a downstream buyer can get benefit from the mechanism not only by adjusting his own market decisions, but also by promoting other buyers to make new decisions. Studies on GBS started a few years ago, but in the retail industry there already had been a lot of applications, such as eWinWin, MobShop.com and so on. Some important works in this field are listed as follows: Chen et al. [20] builds a dynamic game model for the GBS and prove that GBS is incentively compatible for bidders under IPV(independent private values) assumption. Kauffman and Wang et al. [31] conduct an experimental study and find the three efforts: "positive participation externality effort", "price drop effort" and "ending effort". Our model in a certain sense is an extension of the model in Cai, Chen and Song et al. [15]. In their paper, GBS is proved to be better than the traditional flat wholesale price scheme and can bring the supplier and retailers a win-win situation. The main difference between our model and theirs is that in their model, "time" is not an important factor to be studied since the product is not perishable and retailers' ordering sequence just follows a first-come-first-order policy. In our model the other hand, the products are deteriorating over time and extra management on the "time" factor is required. For example, the ordering time now becomes a decision variable which should be carefully investigated. Hence, how to construct the optimal strategy on "time" for all the chain members also becomes a critical issue.

The rest of this chapter is organized as follows: in Section 2.2, we will detail the problem and introduce our models, and we will show some important assumptions in our models. Then we derive the optimal strategies of supplier and retailers under two pricing mechanisms: Flat Price (FP) in Section 2.3 and Group-Buying Scheme (GBS) in Section 2.4. Furthermore in Section 2.5, we compare the optimal payoffs of all the supply chain members under both FP and GBS. Finally, we summarize all the results obtained in this chapter and propose some possible extensions in Section 2.6.

2.2. Model and Assumptions

We study the following supply chain problem. It is a two-echelon supply chain with one supplier selling perishable products to N retailers. The retailers receive the product immediately after they place orders and they need to delivery these products to their markets. The products are assumed to be totally fresh in the supplier's facility but start to deteriorate continuously in quality from the moment they are transported. The supplier endures a remarkable holding cost to keep the products fresh which is stated as $h_0(t)$ per unit product for a time period t. We assume that the retailers' transportation times are deterministic and denoted by a_i, $i \in 1, ..., N$. We also assume that each market has an open time which represents the deadline for the products to reach the market. Specifically for retailer i, products should arrive at market i no later than the open time b_i, $i \in 1, ..., N$. If the products arrive at market i before b_i, the retailer i has to undertake extra holding cost, with $h_i(t)$ per item for time length t. But late delivery after b_i is forbidden. The retailers generate their revenue from selling the products in their markets respectively where the demand in each market is affected by the retail price and the quality of the products. Here we use a similar demand function as in Cai et al. [15] and assume that all the retailers' market demands are independent from each other. We model this problem as Stackelberg game(Gibbons et al. [25]). The supplier and retailers are independent parties, and each party has the objective to maximize its own profit. The supplier is the leader of the game and sets the wholesale price scheme W in the first stage. The retailers are the followers, each of them places order in the second stage according to W, the market demand and other retailers' likely order actions.

The following assumptions are necessary for our model:

- All the members in this supply chain are "greedy" win the sense that each member only makes decision to maximize his own profit.

- All the information is publicly available, which implies that all the members in the supply chain knows everything about the selling prices, transportation times and demand distributions in all markets.

- All the retailers are rational. Therefore when the Nash equilibrium is not unique, they will select the Pareto-optimal equilibrium.

- For each retailer, if there exist two kinds of ordering strategies which bring him the same profit, then he will choose the strategy with larger order quantity since his service level can be raised if he possesses more products.

- $b_i \geq a_i$ $\forall i = 1, ..., N$, which is a necessary condition for retailers to be able to deliver the products to the market in time.

- The products of the supplier are sufficient to satisfy the orders from the retailers. Unsold products have no salvage value.

With the assumptions above, we can start our analysis on the supplier and retailers optimal strategies under pricing mechanisms FP and GBS.

2.3. Modeling with FP

In this section we will discuss the chain members' strategies under FP. Suppose that the supplier sets the wholesale price as w at time 0 and we start the analysis by formulating the expected profit of each retailer under FP. Suppose the market demand of retailer i is

$$D_i(t_i, p_i) = y_i p_i^{-k_i} \theta(t_i)\varepsilon_i, \quad k_i > 1, \tag{2.1}$$

where p_i is the retail price and t_i denotes the deterioration time of the products before they arrive at the market. Similarly as in [15], y_i is a constant represents the potential size of the market, k_i is the price-elasticity index, $\theta(t_i)$ is the level of freshness of the product after deteriorating by time t_i and ε_i is a random variable with distribution function $F_i(x)$ and probability function $f_i(x)$. Then π_i, the expected profit for retailer i, becomes

$$\pi_i(t_i, q_i, p_i|\ w) = p_i E\{\min(D_i(p_i, b_i - t_i), q_i)\} - q_i(w + h_i(b_i - a_i - t_i)), \quad t_i \leq b_i - a_i \tag{2.2}$$

where q_i is the order quantity. Without loss of generality we arrange the retailers on an increasing order of $b_i - a_i$, which can be regarded as the latest order time of retailer i. Hence we have $b_1 - a_1 \leq b_2 - a_2 \leq, \cdots, \leq b_N - a_N$. In order to maximize the expected profit, retailer i should find the optimal strategy (t_i^*, q_i^*, p_i^*). Since $D_i(p, t)$ is a decreasing function of t and $h_i(t)$ is an increasing function of t, we can conclude that $\pi_i(t_i, q_i, p_i|\ w)$ is an increasing function of t_i. Therefore in order to maximize retailer i's profit, we should set $t_i^* = b_i - a_i$ and then find the optimal order quantity q_i^* and retailer price p_i^*. We can apply the backwards approach to obtain them. Specifically, let

$$p_i = (\frac{y_i\theta(b_i - t_i)z_i}{q_i})^{\frac{1}{k_i}} \quad \forall i = 1, ..., N \tag{2.3}$$

and note that p_i and $z_i > 0$ are one-to-one correspondence. Hence the problem left is to find the optimal z_i^* and q_i^*. Given the order quantity q_i, we can obtain $z_i^*(q_i)$ by setting the first derivative of $\pi_i(z_i | q_i, b_i - a_i, w)$ equals to 0, that is

$$\frac{d\pi_i(z_i| q_i, b_i - a_i, w)}{dz_i} = (\frac{y_i\theta(a_i)}{q_i})^{\frac{1}{k_i}} \frac{q_i}{k_i}[(1 - k_i) \in T^{z_i} \frac{x f_i(x)}{z_i} dx + \int_{z_i}^{\infty} f_i(x)dx] = 0$$

Therefore $z_i^* > 0$ should satisfy the following equation

$$(k_i - 1) \int_0^{z_i} x f_i(x)dx = z_i \bar{F}_i(z_i) \quad \forall i = 1, ..., N \tag{2.4}$$

Note that equation (2.4) is similar to the formula obtained in Lemma 1 of Cai et al. [15]. And using the same method in their paper, we can easily prove that z_i^* has unique positive value if ε_i has a generalized increasing failure rate (GIFR) and $\lim_{x\to\infty} x\bar{F}_i(x) = 0$.

Based on the optimal z_i^* above, we can rewrite the expected profit function of retailer i as

$$\pi_i(q_i| b_i-a_i, w) = \pi(z_i^*| q_i, b_i-a_i, w) = \frac{k_i}{k_i - 1}(y_i\theta(a_i))^{\frac{1}{k_i}}(q_i)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - (w+h_i(0))q_i \tag{2.5}$$

Set the first derivative equal to 0 and we can obtain the optimal q_i^* as

$$q_i^* = \frac{y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w^{k_i}} \tag{2.6}$$

The next theorem summarizes the optimal strategy of retailer i under FP:

Theorem 2.1. *Under FP with wholesale price w, in order to maximize the expected profit, retailer i will set his optimal strategy on ordering time, ordering quantity and retail price as*

$$(t_i^*, q_i^*, p_i^*) = (b_i - a_i, \frac{y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w^{k_i}}, \frac{w}{\bar{F}_i(z_i^*)}) \tag{2.7}$$

The maximum expected profit therefore is

$$\pi_i^*(w) = \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}(z_i^*)}{(k_i - 1)w^{k_i-1}} \tag{2.8}$$

Next we consider the supplier's strategy on the wholesale price w. Let c denote the unit cost of the product and assume there are always sufficient products to satisfy the retailers' orders. Then the supplier's profit is

$$\pi_S(w) = \sum_{i=1}^{N}(w - c - h_0(b_i - a_i))q_i^* \tag{2.9}$$

The first derivative of $\pi_S(w)$ is

$$\frac{d\pi_S(w)}{dw} = \sum_{i=1}^{N} \frac{[k_i(c + h_0(b_i - a_i) - (k_i - 1)w]y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w^{k_i+1}}$$

Therefore the optimal wholesale price that maximizes supplier's profit should be

$$w^* = \arg_w\{\frac{d\pi_S(w)}{dw} = 0\} \tag{2.10}$$

Furthermore from the first derivative of $\pi_S(w)$ we can conclude that

$$w^* \in (\min_i \frac{k_i(c + h_0(b_i - a_i))}{k_i - 1}, \max_i \frac{k_i(c + h_0(b_i - a_i))}{k_i - 1})$$

Theorem 2.2. *Under FP, the supplier, in order to maximize his profit, should set the optimal wholesale price as w^* which satisfies*

$$\frac{d\pi_S(w)}{dw} = \sum_{i=1}^{N} \frac{[k_i(c + h_0(b_i - a_i) - (k_i - 1)w^*]y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{(w^*)^{k_i+1}} \tag{2.11}$$

and his maximum profit is

$$\pi_S^* = \sum_{i=1}^{N}(w^* - c - h_0(b_i - a_i))\frac{y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w^{*k_i}}$$

Remarks:

- Suppose that the price-elasticity index k_i, $i = 1, \cdots, N$ are quite similar with each other, then the difference between $\frac{k_i(c+h_0(b_i-a_i))}{k_i-1}, i = 1, ..., N$ will mainly depend on $b_i - a_i$. If we arrange the retailers in an ascending order on $b_i - a_i$, we roughly have

$$w^* \in (\frac{k_1(c + h_0(b_1 - a_1))}{k_1 - 1}, \frac{k_N(c + h_0(b_N - a_N))}{k_N - 1}) \tag{2.12}$$

since $b_i - a_i$ is increasing on i. Note that condition (2.12) is necessary in our analysis in Section 1.4.

- Note that $w - h_0(b_i - a_i) \geq c$ is a necessary condition for the supplier to sell the product to retailer i. Therefore we assume

$$\max_i c + h_0(b_i - a_i) \leq \min_i \frac{k_i(c + h_0(b_i - a_i))}{k_i - 1} \tag{2.13}$$

which ensures that the optimal wholesale price will bring positive profit to the supplier in any trade with the retailers.

Theorems 2.1 and 2.2 provide the optimal strategies of supplier and retailers under FP pricing mechanism. Obviously, since the wholesale price is fixed and the market demands are independent, it is not necessary for a retailer to consider other retailers possible ordering strategies when he makes his optimal ordering decision. The supplier can maximize his profit by adjusting the wholesale price w according to the total ordering quantities $\sum_{i=1}^{N}$ from the retailers. An interesting question is that if Q, the amount of products on supplier's hand, is greater than the total order quantity from the retailers, then whether the supplier can use any pricing mechanism to motivate the retailers to order more so that his profit can further increase. In the next section, we will answer this question.

2.4. Modeling with GBS

In this section we focus on the supplier and retailers' performances under GBS which means the wholesale price scheme provided by the supplier becomes $w_G = (w_1, w_2, l, T)$ $w_2 < w_1$. This scheme represents that a retailer is guaranteed to get the product with wholesale price w_1 and if the accumulated order quantity during period $[0, T]$ exceeds l, the supplier repays the retailers who ordered on or before time T with $(w_1 - w_2)$ for each unit of product they ordered. In other words, the wholesale price for these retailers becomes w_2. But the wholesale price for retailers who order after the expiry time T is always w_1. The rationale of this scheme is that supplier can use a lower wholesale price to motivate the retailers to put earlier orders (before T) and meanwhile to order more so that the threshold l can be reached. In this way, the supplier can hopefully save the holding cost and generate more profit through selling more products to the retailers. On the other side, GBS is also attractive to the retailers. Because it offers them another option to place orders while most importantly, they can at least receive the same profit as they did under FP if their original strategies are unchanged.

2.4.1. The retailers' strategy

Given the GBS w_G, each retailer should consider whether to retain his optimal order strategy under FP or to modify it to possibly achieve higher profit. It is not difficult to verify that retailer i's expected profit under GBS is

$$\pi_i(t_i, q_i, p_i \mid t_{-i}, q_{-i}, W_G) = p_i E\{\min(D_i(p_i, b_i - t_i), q_i)\} - q_i(w(t_i, q_i \mid t_{-i}, q_{-i}, T) + h_i(b_i - a_i - t_i)) \tag{2.14}$$

where

$$w(t_i, q_i \mid t_{-i}, q_{-i}, T) = \begin{cases} w_2 & \text{If } t_i \leq T, \sum_{j \in \{k \mid t_k \leq T, k=1, \dots, N\}} q_j \geq l \\ w_1 & \text{Otherwise.} \end{cases} \tag{2.15}$$

Here the decision variables are order time t_i, order quantity q_i and retail price p_i. Other retailers' decision variables are

$$t_{-i} = \{t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_N\} \text{ and } q_{-i} = \{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_N\}$$

Using backwards approach, we can simplify this three decision variables problem into one decision variable problem. We first study the optimal retail price p_i^* condition on q_i and t_i. Since $w(t_i, q_i)$ is determined when t_i and q_i are given, we can observe that retailer i's expected profit can be rewritten as

$$\pi_i(p_i \mid t_i, q_i, t_{-i}, q_{-i}, W_G) = p_i E\{\min(y_i p_i^{k_i} \theta(b_i - t_i)\varepsilon, q_i)\} - q_i(w(t_i, q_i) + h_i(b_i - a_i - t_i)) \tag{2.16}$$

Similarly as in Section 3, we can define $p_i = (\frac{y_i \theta(b_i - t_i) z_i}{q_i})^{\frac{1}{k_i}}$ $\forall i = 1, \dots, N$ and obtain the optimal z_i^* condition on t_i and q_i. Note that z_i^* again is given by equation (2.4). Therefore we can easily obtain p_i^* accordingly. Furthermore, given the order time t_i, the expected profit of retailer i now becomes

$$\begin{aligned} & \pi_i(q_i|\ t_i, t_{-i}, q_{-i}, W_G) \\ = \ & \pi_i(p_i^*(t_i, q_i)|\ t_i, q_i, t_{-i}, q_{-i}, W_G) \\ = \ & \tfrac{k_i}{k_i-1}(y_i\theta(b_i - t_i))^{\frac{1}{k_i}}(q_i)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - q_i(w(t_i, q_i) + h_i(b_i - a_i - t_i)) \end{aligned} \tag{2.17}$$

Since the wholesale price $w(t_i, q_i)$ depends on t_i and q_i, in order to find the optimal order quantity $q_i^*(t_i)$, we should consider the following situations:

1. $t_i > T$. In this situation, the wholesale price for the retailer is w_1 and the optimal order quantity is easily obtained by taking first derivative of the expected profit function. We can verify that

$$q_i^*(t_i, w_1) = \frac{y_i\theta(b_i - t_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{[w_1 + h_i(b_i - a_i - t_i)]^{k_i}} \tag{2.18}$$

2. $t_i \leq T$. In this situation, the wholesale price can be w_2 if $\sum_{j\in\{k|\ t_k\leq T\}} q_j \geq l$. Otherwise, the wholesale price remains w_1 and the optimal order quantity $q_i^*(t_i)$ is also given by equation (2.18). Denote $\pi_i(q_i|\ t_i, w_1)$ and $\pi_i(q_i|\ t_i, w_2)$ as retailer i's expected profit condition on order time t_i together with wholesale price w_1 and w_2 respectively. Note that the optimal order quantity for $\pi_i(q_i|\ t_i, w_1)$ is given by equation (2.18) and the optimal order quantity $q_i^*(t_i, w_2)$ for $\pi_i(q_i|\ t_i, w_2)$ will be

$$q_i^*(t_i, w_2) = \frac{y_i\theta(b_i - t_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{[w_2 + h_i(b_i - a_i - t_i)]^{k_i}} \tag{2.19}$$

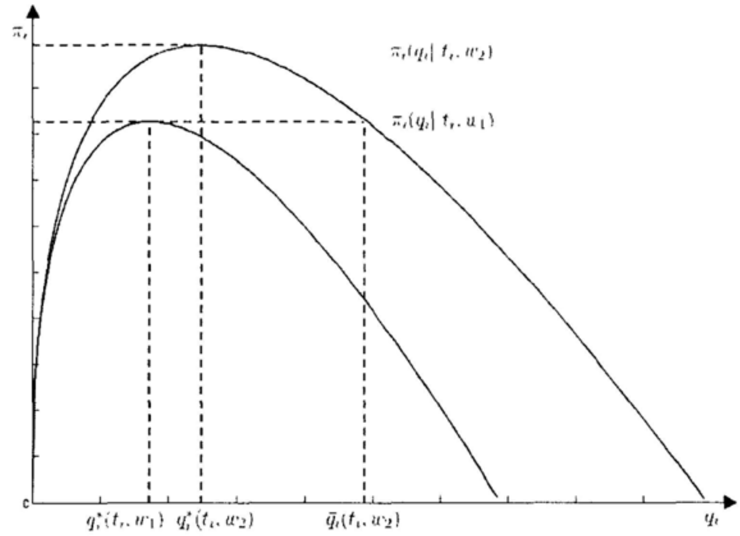
Obviously $q_i^*(t_i, w_2) > q_i^*(t_i, w_1)$ and we can verify that

$$\begin{aligned} \pi_i(q_i^*(t_i, w_2)|\ t_i, w_2) &= \tfrac{y_i\theta(b_i-t_i)z_i^*\bar{F}^{k_i}}{(k_i-1)[w_2+h_i(b_i-a_i-t_i)]^{k_i-1}} \\ &> \tfrac{y_i\theta(b_i-t_i)z_i^*\bar{F}^{k_i}}{(k_i-1)[w_1+h_i(b_i-a_i-t_i)]^{k_i-1}} = \pi_i(q_i^*(t_i, w_2)|\ t_i, w_1) \end{aligned}$$

Hence, there must exist $q_i > q_i^*(t_i, w_2)$ such that

$$\pi_i(q_i|\ t_i, w_2) = \pi_i(q_i^*(t_i, w_1)|\ t_i, w_1)$$

Figure 2.1: The illustration of different order quantity



Denote this q_i as $\bar{q}_i(t_i, w_2, \pi_i(t_i, w_1))$ and Figure 2.1 shows a simple example to illustrate the meaning of $\bar{q}_i(t_i, w_2)$.

In Figure 2.1, the upper curve represents retailer i's profit if he orders at time t_i with wholesale price w_2 and the lower curve states his profit if the wholesale price is w_1. It is obvious that retailer i will never order more than $\bar{q}_i(t_i, w_2)$, otherwise his expected profit will be even less than the profit obtained with wholesale price w_1. Hence $\bar{q}_i(t_i, w_2, \pi_i(t_i, w_1))$ is a upper bound of the ordering quantity at time t_i given that retailer i's profit is at least $\pi_i(q_i^*(t_i, w_1)|\ t_i, w_1)$. So the optimal strategy on order quantity in this situation is

$$q_i^*(t_i) = \begin{cases} q_i^*(t_i, w_1) & \text{If } \bar{q}_i(t_i, w_2) < l - \sum\limits_{j \in \{k|\ t_k < T, k \neq i\}} q_i \\ q_i^*(t_i, w_2) & \text{If } q_i^*(t_i, w_2) \geq l - \sum\limits_{j \in \{k|\ t_k < T, k \neq i\}} q_i \\ l - \sum\limits_{j \in \{k|\ t_k < T, k \neq i\}} q_i & \text{Otherwise.} \end{cases} \tag{2.20}$$

Basing on the above analysis, we can derive the optimal order time t_i^*. Let

$q_i(l,T) = l - \sum_{j\in\{k|\ t_k\leq T, k\neq i\}} q_j$ denote the necessary order quantity to reach the threshold set by the supplier. The result is summarized in the following lemma:

Lemma 2.3. *In order to maximize the profit, retailer i should order at time $t_i^* = b_i - a_i$ if $T \geq b_i - a_i$. Otherwise if $T \leq b_i - a_i$, he should consider the following two cases:*

1. *If $\bar{q}_i(T, w_2, \pi(b_i - a_i, w_1)) < q_i(l,T)$, the optimal order time for retailer i is $t_i^* = b_i - a_i$.*

2. *If $\bar{q}_i(T, w_2, \pi(b_i - a_i, w_1)) \geq q_i(l,T)$, denote $\hat{q}_i = \max(q_i^*(T, w_2), q_i(l,T))$. If the following inequality (2.21) holds, the optimal order time is $t_i^* = T$. Otherwise, retailer i should set $t_i^* = b_i - a_i$.*

$$\frac{k_i}{k_i - 1}(y_i\theta(b_i - T))^{\frac{1}{k_i}}(\hat{q}_i)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - \hat{q}_i(w_2 + h_i(b_i - a_i - T)) \geq \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}}{(k_i - 1)w_1^{k_i - 1}} \tag{2.21}$$

Proof When $T \geq b_i - a_i$, since retailer i cannot order after time $b_i - a_i$, the optimal order time $t_i^* \leq T$. Hence the wholesale price will be w_2 if the cumulated order quantity reaches the threshold l. Otherwise, the wholesale price for retailer i remains w_1. According to Theorem 2.1, retailer i should order at time $b_i - a_i$ if the wholesale price is w_1. Furthermore, if the wholesale price is w_2, the optimal order time should be $b_i - a_i$ again. Otherwise if $t_i^* < b_i - a_i$, according to equation (2.19), the optimal order quantity of retailer i will be less than the optimal one at ordering time $b_i - a_i$. This implies that if retailer i, by placing optimal order quantity at time $b_i - a_i$, can not enjoy the lower wholesale price w_2, then he can neither enjoy w_2 with optimal ordering at time $t_i^* < b_i - a_i$. Combining with Theorem 2.1, which implies that the profit retailer i can obtain by ordering at time t_i^* will be less than ordering at time $b_i - a_i$, we know the optimal ordering time should be $t_i^* = b_i - a_i$.

However, when $T < b_i - a_i$, retailer i should consider whether to order no later than T and hopefully enjoy a lower wholesale price w_2 or just to remain the original optimal order strategy with wholesale price w_1. Note that $\bar{q}_i(t_i, w_2, \pi(b_i -$

$a_i, w_1))$ is increasing in t_i when $t_i \leq T$ (See Appendix [1]). Hence if $\bar{q}_i(T, w_2, \pi(b_i - a_i, w_1)) \geq q_i(l, T)$, which implies that the lower wholesale price has chance to be obtained, the retailer will consider to order during period $[0, T]$ and the expected profit for him is $(t_i \leq T)$

$$\pi_i(t_i|\, t_{-i}, q_{-i}, w_2) = \begin{cases} \pi_i(t_i, q_i^*(t_i, w_2), p_i^*(t_i, q_i^*(t_i, w_2))|\, t_{-i}, q_{-i}, w_2) & \text{If } q_i^*(t_i, w_2) \geq q_i(l) \\ \pi_i(t_i, q_i(l, T), p_i^*(t_i, q_i(l, T))|\, t_{-i}, q_{-i}, w_2) & \text{Otherwise.} \end{cases}$$

It is not difficult to verify that $\pi_i(t_i, q_i^*(t_i, w_2), p_i^*(t_i, q_i^*(t_i, w_2))|\, t_{-i}, q_{-i}, w_2)$, $\pi_i(t_i, q_i(l, T), p_i^*(t_i, q_i(l, T))|\, t_{-i}, q_{-i}, w_2)$ and $q_i^*(t_i, w_2)$ are increasing on t_i when $t_i \leq T$, thus we know $\pi_i(t_i|\, t_{-i}, q_{-i}, w_2)$ is also increasing in t_i when $t_i \leq T$. (See details in Appendix [1].) In other words, retailer i will order at time T if he decides to put earlier order between time interval $[0, T]$. Hence we have

$$\pi_i^*(T|\, t_{-i}, q_{-i}, w_2) = \frac{k_i}{k_i - 1}(y_i\theta(b_i - T))^{\frac{1}{k_i}}(\hat{q}_i)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - \hat{q}_i(w_2 + h_i(b_i - a_i - T))$$

Meanwhile the maximum profit retailer i can obtain if he orders at time $b_i - a_i$ with wholesale price w_1 should be $\pi_i^*(w_1)$ as in equation (2.8). Thus if inequality (2.21) holds, retailer i will order at time T. Otherwise, he will still put order at time $b_i - a_i$.

When $T < b_i - a_i$ and $\bar{q}_i(T, w_2, \pi(b_i - a_i, w_1)) < q_i(l, T)$, retailer i never has chance to enjoy a lower wholesale price w_2. So he will order at time $b_i - a_i$ with wholesale price w_1. □

Remarks:

- The left hand side of inequality (2.21) has two possible values when $\hat{q}_i$ equals to $q_i^*(T, w_2)$ and $q_i(l, T)$ separately. According to the definition of $q_i^*(T, w_2)$, we know the left hand side of inequality (2.21) will have a larger value when $\hat{q}_i = q_i^*(T, w_2)$. Therefore if there is a chance that inequality (2.21) holds, then

$$\begin{aligned} & \tfrac{k_i}{k_i-1}(y_i\theta(b_i - T))^{\frac{1}{k_i}}(q_i^*(T, w_2))^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - q_i^*(T, w_2)(w_2 + h_i(b_i - a_i - T)) \\ \geq\ & \tfrac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}}{(k_i-1)w_1^{k_i-1}} \end{aligned}$$

or equivalently

$$\frac{y_i\theta(b_i - T)z_i^*\bar{F}^{k_i}}{(k_i-1)(w_2+h_i(b_i-a_i-T))^{k_i-1}} \geq \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}}{(k_i-1)w_1^{k_i-1}}$$

Since $b_i - T \geq a_i$ and $\theta(t)$ is decreasing on t, a necessary condition for the above inequality to hold is

$$w_2 + h_i(b_i - a_i - T) \leq w_1 \tag{2.22}$$

This condition is easy to understand and it actually implies that retailer i will only put earlier order when his saving on the wholesale price can cover the extra holding cost.

- From the above observation we can conclude that retailer i will order at time $b_i - a_i$ if $T < b_i - a_i - h^{-1}(w_1 - w_2)$

With the above analysis, we introduce the first main result of this chapter as follows:

Theorem 2.4. *Given the supplier's wholesale price scheme $w_G = (w_1, w_2, l, T)$, and other retailers' order strategies (t_{-i}, q_{-i}), retailer i's optimal strategy (t_i^*, q_i^*, p_i^*) should be*

$$\begin{cases} (b_i - a_i, \frac{y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w_2^{k_i}}, \frac{w_2}{\bar{F}_i(z_i^*)}) & \textit{If } b_i - a_i \leq T, q_i^*(b_i - a_i, w_2) \geq q_i(l,T) \\ (b_i - a_i, q_i(l,T), (\frac{y_i\theta(a_i)z_i^*}{q_i(l,T)})^{\frac{1}{k_i}}) & \textit{If } b_i - a_i \leq T, q_i^*(b_i - a_i, w_2) < q_i(l,T) \leq \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) \\ (T, \frac{y_i\theta(b_i - T)z_i^*F_i^{k_i}(z_i^*)}{[w_2+h_i(b_i-a_i-T)]^{k_i}}, \frac{w_2}{\bar{F}_i(z_i^*)}) & \textit{If } b_i - a_i > T, q_i^*(T, w_2) \geq q_i(l,T), w_2 + h_i(b_i - a_i - T) \leq w_1 \\ (T, q_i(l,T), (\frac{y_i\theta(b_i - T)z_i^*}{q_i(l,T)})^{\frac{1}{k_i}}) & \textit{If } b_i - a_i > T, q_i^*(T, w_2) < q_i(l,T) \leq \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1)) \\ & , \pi_i(T, q_i(l,T), p_i^*(T, q_i(l,T))) \geq \pi_i^*(w_1) \\ (b_i - a_i, \frac{y_i\theta(a_i)z_i^*\bar{F}_i^{k_i}(z_i^*)}{w_1^{k_i}}, \frac{w_1}{\bar{F}_i(z_i^*)}) & \textit{Otherwise} \end{cases} \tag{2.23}$$

And the relevant maximum profit of retailer i is

$$\begin{cases} \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}}{(k_i-1)w_2^{k_i-1}} \\ \frac{k_i}{k_i-1}(y_i\theta(a_i))^{\frac{1}{k_i}}q_i(l,T)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - q_i(l,T)w_2 \\ \frac{y_i\theta(b_i-T)z_i^*\bar{F}^{k_i}}{(k_i-1)(w_2+h_i(b_i-a_i-T))^{k_i-1}} \\ \frac{k_i}{k_i-1}(y_i\theta(b_i-T))^{\frac{1}{k_i}}q_i(l,T)^{1-\frac{1}{k_i}}(z_i^*)^{\frac{1}{k_i}}\bar{F}_i(z_i^*) - q_i(l,T)(w_2+h_i(b_i-a_i-T)) \\ \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}}{(k_i-1)w_1^{k_i-1}} \end{cases}$$

Theorem 2.4 can be verified straightforwardly from Lemma 2.3 and it gives the optimal strategy of retailer i under GBS given all the other retailers' order information. When all the retailers place orders, a static procurement game is formed and their order times and order quantities are their strategies in this game. We are trying to find the equilibrium ordering vector $(t_i^e, t_{-i}^e, q_i^e, q_{-i}^e)$ for all retailers which satisfy

$$\pi_i(t_i^e, t_{-i}^e, q_i^e, q_{-i}^e, W_G) \geq \pi_i(t_i, t_{-i}^e, q_i, q_{-i}^e, W_G) \text{ for all feasible } t_i, q_i,\ i = 1, 2, .., N \tag{2.24}$$

Before we construct the optimal ordering strategies for all the retailers, we first introduce some notations which will be used in the rest of this chapter. Define the set $S(T) = \{i |\ b_i - a_i > T, i = 1, ..., N\}$ which represents the set of retailers who can order later than the expiry time T. Furthermore let

$$S_1(T) = \{i |\ \pi_i(T, q_i^*(T, w_2), p_i^*(T, q_i^*(T, w_2)))|w_2) \geq \pi_i^*(w_1), i \in S(T)\}$$

We can verify that $S_1(T) \subseteq S(T)$. We can also observe from the definition of $S_1(T)$ that if $i \in S(T) \setminus S_1(T)$, retailer i will always order at time $b_i - a_i$ with wholesale price w_1. The following Theorem 2.5 describes the retailers' optimal order strategies.

Theorem 2.5. *Under GBS pricing mechanism $w_G = (w_1, w_2, l, T)$, the Nash equilibrium of retailer i's order strategy on order time and order quantity is*

1) If $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) \geq l$,

$$(t_i^e, q_i^e) = \begin{cases} (b_i - a_i, q_i^*(b_i - a_i, w_2)) & \textit{If } i \notin S(T) \\ (T, q_i^*(T, w_2)) & \textit{If } i \in S_1(T) \\ (b_i - a_i, q_i^*(b_i - a_i, w_1)) & \textit{Otherwise.} \end{cases} \tag{2.25}$$

2) If $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) < l \leq \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i -$

$a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

$$(t_i^e, q_i^e) = \begin{cases} (b_i - a_i, q_i(l,T)) & \text{If } i \notin S(T) \\ (T, q_i(l,T)) & \text{If } i \in S_1(T) \\ (b_i - a_i, q_i^*(b_i - a_i, w_1)) & \text{Otherwise.} \end{cases} \quad (2.26)$$

where

$$\begin{cases} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) \geq q_i(l,T) > q_i^*(b_i - a_i, w_2) & \text{If } i \notin S(T) \\ \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1)) \geq q_i(l,T) > q_i^*(T, w_2) & \text{If } i \in S_1(T) \end{cases}$$

3) Otherwise, if $l > \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$.

$$(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_1)) \quad (2.27)$$

Proof. **Case I:** $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) \geq l$

Sub-Case I-a: $i \notin S(T)$

In this case $b_i - a_i \leq T$ and according to Theorem 2.4, the optimal order time $t_i^* = b_i - a_i$. Suppose that the order strategy of other retailers (t_{-i}^e, q_{-i}^e) is given by equation (2.25). Then we need to prove that retailer i can not benefit by changing his strategy $(b_i - a_i, q_i^*(b_i - a_i, w_2))$ while (t_{-i}^e, q_{-i}^e) remains unchanged. Recall the definition of $q_i^*(b_i - a_i, w_2)$, we know $\pi_i(b_i - a_i, q_i, p_i^*(b_i - a_i, q_i))$ achieves the maximum value at $q_i^*(b_i - a_i, w_2)$. Hence changing q_i^e to other values will never benefit the retailer. Therefore $(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_2))$ is retailer i's Nash equilibrium point.

Sub-Case I-b: $i \in S_1(T)$

In this case $b_i - a_i > T$ and according to Theorem 2.4, the optimal order time t_i^* can be either $b_i - a_i$ or T. By the definition of $S_1(T)$, we can conclude that if retailer i face a wholesale price w_2, he will benefit by ordering at T. Suppose that the order strategy of other retailers (t_{-i}^e, q_{-i}^e) is given by equation (2.25). If $(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_2))$, then the cumulated order quantity exceeds the threshold l, which brings the wholesale price down to w_2. Thus $t_i^e = T$. Then using the similar approach in Sub-Case I-a, we can prove that $q_i^*(b_i - a_i, w_2)$ brings the retailer maximum profit. Therefore $(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_2))$ is retailer i's Nash equilibrium point.

Sub-Case I-c: $i \in S(T) \setminus S_1(T)$

In this case $b_i - a_i > T$ and according to the definition of $S_1(T)$, the maximum profit retailer i can obtain by ordering before T with wholesale price w_2 is less than the maximum profit he can achieve by ordering at time $b_i - a_i$. Hence $(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_1))$ is retailer i's Nash equilibrium point.

Case II: $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) < l \leq \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

Sub-Case II-a: $i \notin S(T)$

In this case, $b_i - a_i \leq T$ and according to Theorem 2.4, the optimal order time $t_i^* = b_i - a_i$. Suppose that the order strategy of other retailers (t_{-i}^e, q_{-i}^e) is given by equation (2.26). Then we need to prove that retailer i will not benefit by setting $(t_i^e, q_i^e) \neq (b_i - a_i, q_i(l, T))$. If $q_i^e > q_i(l, T)$, then the wholesale price is still w_2. Since $\pi_i(q_i|t_i, w_2)$ is decreasing on q_i when $q_i > q_i^*(t_i, w_2)$, we know for $q_i^e > q_i(l, T) > q_i^*(t_i^e, w_2)$

$$\pi_i(t_i^e, q_i^e, p_i^*(t_i^e, q_i^e)|w_2) < \pi_i(t_i^e, q_i(l, T), p_i^*(t_i^e, q_i(l, T))|w_2)$$

Therefore retailer i should not set $q_i^e > q_i(l, T)$.

Furthermore, if $q_i^e < q_i(l, T)$, then the threshold will not be reached and the wholesale price for retailer i becomes w_1. And we can verify that

$$\begin{aligned} \pi_i(t_i^e, q_i^e, p_i^*(t_i^e, q_i^e)|w_1) &\leq \pi_i^*(w_1) \\ &= \pi_i(t_i^e, \bar{q}_i(t_i^e, w_2, \pi_i(b_i - a_i, w_1)), p_i^*(t_i^e, \bar{q}_i)|w_2) \\ &\leq \pi_i(t_i^e, q_i(l, T), p_i^*(t_i^e, q_i(l, T))|w_2) \end{aligned}$$

where the last inequality is due to the decreasing of $\pi_i(q_i|t_i, w_2)$ on $q_i > q_i^*(t_i, w_2)$. Hence, the retailer i will not benefit by setting $q_i^e < q_i(l, T)$. Meanwhile due to the fourth assumption in Section 1.2, retailer's Nash equilibrium point is $(t_i^e, q_i^e) = (b_i - a_i, q_i(l, T))$.

Sub-Case II-b: $i \in S_1(T)$

In this case $b_i - a_i > T$ and according to Theorem 2.4, the optimal order time t_i^* can be either $b_i - a_i$ or T. Suppose the order strategy of other retailers (t_{-i}^e, q_{-i}^e) is given by equation (2.26) and retailer i sets $q_i^e > q_i(l, T)$. Then the wholesale price is w_2 and according to the definition of $S_1(T)$ and Theorem 2.4, the optimal ordering time is T. Hence $t_i^e = T$. Using the same approach in Sub-Case II-a, we can prove that retailer i can not benefit by setting $q_i^e > q_i(l, T)$.

If retailer i sets $q_i^e < q_i(l, T)$, then the wholesale price becomes w_1. Again with the method applied in Sub-Case II-a, we can prove that the profit of retailer i will be less. Therefore we can conclude in this case that the Nash equilibrium point of retailer i is $(t_i^e, q_i^e) = (T, q_i(l, T))$.

Sub-Case II-c: $i \in S(T) \setminus S_1(T)$

Similarly as proved in Sub-Case I-c, retailer i always place order at time $b_i - a_i$ with wholesale price w_1. Hence the Nash equilibrium point is given by (2.26).

Case III: $l > \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

In this case, retailers will give up pursuing the lower wholesale price w_2 since the threshold is too large. All of them will face the normal wholesale price w_1 and they just maximize their profit independently. Therefore the Nash equilibrium point for retailer i is $(t_i^e, q_i^e) = (b_i - a_i, q_i^*(b_i - a_i, w_1))$. □

Theorem 2.5 gives us a Nash equilibrium strategy, and a spontaneous question follows is that whether there exists another equilibrium which can improve all retailers' profits. Moreover, have the retailers any incentive to build a sub-coalition to move away from the equilibrium? Theorem 2.6 gives the answers.

Theorem 2.6. **(i)** *The equilibrium in Theorem 2.5 is the only Pareto optimal Nash equilibrium.* **(ii)** *The equilibrium in Theorem 2.5 is a strong equilibrium.*

Proof. **(i)** In order to prove the Nash equilibrium (t_i^e, q_i^e) obtain in Theorem 2.5 is the only Pareto optimal Nash equilibrium, we should verify that if there exists another Nash equilibrium with the ith component as (t_i', q_i'), which satisfies $\pi_i(t_i', q_i', p_i^*(t_i', q_i')) > \pi_i(t_i^e, q_i^e, p_i^*(t_i^e, q_i^e))$, then there must exist $j \neq i$, so that

$\pi_j(t'_j, q'_j, p^*_j(t'_j, q'_j)) < \pi_j(t^e_j, q^e_j, p^*_j(t^e_j, q^e_j))$. In other words, non of the retailers can improve his own profit while not harming others' profits by deviating from the equilibrium in Theorem 2.5. Again we consider the three situations discussed in Theorem 2.5 one by one

Case I: $\sum_{i \notin S(T)} q^*_i(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q^*_i(T, w_2) \geq l$

In this case we can observe that all the retailers profits have already been maximized following the Nash equilibrium point in Theorem 2.5. Hence, none of the retailers can benefit by deviating from the original equilibrium and (t^e_i, q^e_i) certainly is the unique Pareto optimal Nash equilibrium in this case.

Case II: $\sum_{i \notin S(T)} q^*_i(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q^*_i(T, w_2) < l \leq \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

In this case for retailer i, if $i \in S(T) \setminus S_1(T)$, then his profit is already maximized with t^e_i, q^e_i. Hence he has no incentive to deviate from this equilibrium point. If $i \in S_1(t)$ or $i \in S(t)$, we know $(t^e_i, q^e_i) = (T, q(l.T)$. In order to achieve more profit, retailer i must set $q'_i < q^e_i$. Otherwise if $q'_i \geq q^e_i$,

$$\pi_i(t'_i, q'_i, p^*_i(t'_i, q'_i)|w_2) \leq \pi_i(T, q'_i, p^*_i(T, q'_i)|w_2) \leq \pi_i(T, q^e_i, p^*_i(T, q^e_i)|w_2)$$

And the equality holds when and only when $(t'_i, q'_i) = (t^e_i, q^e_i)$. Meanwhile, since $\sum_{i \in S_1(T)} q'_i + \sum_{i \notin S(T)} q'_i = l$ we know there must exist q'_j, $j \neq i$ so that $q'_j < q^e_j$. Again we can prove

$$\pi_j(t'_j, q'_j, p^*_i(t'_j, q'_j)|w_2) \leq \pi_j(T, q_i(l, T), p^*_i(T, q_i(l, T))|w_2) = \pi_j(t^e_j, q^e_j, p^*_i(t^e_j, q^e_j)|w_2)$$

Therefore, retailer j will be less profitable if retailer i obtains more. In other words, (t^e_i, q^e_i) is the unique Pareto optimal Nash equilibrium for retailer i in this case.

Case III: $l > \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

In this case since retailers can not benefit from lower wholesale price w_2, their profits are maximized at the original Nash equilibrium point given in Theorem 2.5. Hence, (t^e_i, q^e_i) obviously is the unique Pareto optimal Nash equilibrium in this case.

(ii) In order to prove the Nash equilibrium is a strong equilibrium, we need to verify that for any coalition C, if trans-payments among retailers are not allowed, there always exists a retailer $i \in C$ who cannot improve his profit by placing order at $t_i' \neq t_i^e$ or with quantity $q_i' \neq q_i^e$.

Case I: $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) \geq l$ or $l > \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

In this case, since every retailer's profit is maximized (with wholesale price equals to either w_1 or w_2), none of them can obtain more profit by forming a group. Hence, (t_i^e, q_i^e) is a strong equilibrium point.

Case II: $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) < l \leq \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2, \pi_i(b_i - a_i, w_1)) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2, \pi_i(b_i - a_i, w_1))$

For any coalition C, if $C \subseteq S(T) \setminus S_1(T)$, then all the retailers profits in C are already maximized with (t_i^e, q_i^e), non of them can benefit by forming this coalition. Thus we next assume $C \nsubseteq S(T) \setminus S_1(T)$, which implies that there exists subset $C' \subseteq C$ and $C' \cap (S(T) \setminus S_1(T)) = \emptyset$. Regarding to the total order quantity of the coalition C we have two sub-cases.

Sub-Case IIa: $\sum_{i \in C} q_i' \geq \sum_{i \in C} q_i^e$

In this case, there must exists one retailer i whose order quantity $q_i' \geq q_i^e$ and obviously $i \in C'$. Similarly as examined in Case II of (i), retailer i's profit will decrease. Hence, retailer i is worth off in this coalition.

Sub-Case IIb: $\sum_{i \in C} q_i' < \sum_{i \in C} q_i^e$

Since $\sum_{i \in S_1(T)} q_i^e + \sum_{i \notin S(T)} q_i^e = l$ and $C \nsubseteq S(T) \setminus S_1(T)$, we have

$$\sum_{i \in C} q_i' + \sum_{i \in S_1(T) \setminus C} q_i^e + \sum_{i \notin S(T) \setminus C} q_i^e < l$$

Thus the wholesale price is w_1 and the profit of retailer $i \in C'$ is

$$\begin{aligned} \pi_i(t_i', q_i', p_i^*(t_i', q_i')|w_1) &\leq \pi_i^*(w_1) \\ &= \pi_i(t_i^e, \bar{q}_i(t_i^e, w_2, \pi_i(b_i - a_i, w_1)), p_i^*(t_i^e, \bar{q}_i)) \\ &\leq \pi_i(t_i^e, q_i^e, p_i^*(t_i^e, q_i^e)) \end{aligned}$$

Meanwhile the profit of retailer i in $C \setminus C'$ can not be larger. Thus, none of the retailers in this coalition C can be better off.

Combining with the above analysis, it shows that $(t_i^e, t_{-i}^e; q_i^e, q_{-i}^e)$ is strong equilibrium. □

Theorem 2.6 implies that there does not exist another Nash equilibrium which can improve all retailers' profits. Meanwhile, if trans-payments is not allowed, none of the retailers will move away from the equilibrium and form a coalition.

2.4.2. The Supplier's Strategy

Above we discussed the retailer's optimal reactions to $w_G = (w_1, w_2, l, T)$ designed by the supplier. We can observe that with different parameter setting of w_G, retailers' strategies may have huge differences. Hence a natural question is that how should the supplier set w_G so that his profit can be maximized. Furthermore, how much more profit will GBS bring to the supplier compared with FP will also be an interesting issue for us to investigate.

Assume the supplier has sufficient product to sell and his profit function can be written as

1. If $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) \geq l$

$$\begin{aligned}
&\pi_S(w_1, w_2, l, T) \\
= \ &\sum_{i \notin S(T)} (w_2 - c - h_0(b_i - a_i))q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} (w_2 - c - h_0(T))q_i^*(T, w_2) \\
&+ \sum_{i \in S(T) \setminus S_1(T)} (w_1 - c - h_0(b_i - a_i))q_i^*(b_i - a - i, w_1)
\end{aligned}$$

2. If $\sum_{i \notin S(T)} q_i^*(b_i - a_i, w_2) + \sum_{i \in S_1(T)} q_i^*(T, w_2) < l \leq \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i, w_2) + \sum_{i \in S_1(T)} \bar{q}_i(T, w_2)$

$$\begin{aligned}
&\pi_S(w_1, w_2, l, T) \\
= &\sum_{i \notin S(T)} (w_2 - c - h_0(b_i - a_i))q_i(l,T) + \sum_{i \in S_1(T)} (w_2 - c - h_0(T))q_i(l,T) \\
+ &\sum_{i \in S(T) \setminus S_1(T)} (w_1 - c - h_0(b_i - a_i))q_i^*(b_i - a - i, w_1)
\end{aligned}$$

3. Otherwise

$$\pi_S(w_1, w_2, l, T) = \sum_{i=1}^{N} (w_1 - c - h_0(b_i - a_i))q_i^*(b_i - a_i, w_1)$$

As we can see, in order to determine the optimal parameters w_1, w_2, l and T, we have to solve this 4-parameter optimization problem, which can be very difficult. But according to the definition of the model and together with the structure of the above profit function, we can actually reduce the problem into two parameters only. First, without loss of generality we can set $w_1 = w^*$ given by equation (2.10). In fact, if w_2, l and T are appropriately set, such a mechanism ensures that the supplier can at least obtain the same profit as he does by providing FP. Second, we can determine l by studying the relations between the three cases above. We can observe that given w_1, w_2 and T, $\pi_S(l|\ w_1, w_2, T)$ is increasing on l in the second case and remains unchanged in first and third case. Moreover, supplier's profit in the first case is always less than or equal to the profit he can obtain in the second case since $q_i(l,T) \geq q_i^*(T, w_2)$, $\forall i \in S_1(T)$ or $i \notin S(T)$. Therefore we just need to compare the profit the supplier can obtain by setting $l^* = \sum_{i \in S_1(T)} \bar{q}_i(T) + \sum_{i \notin S(T)} \bar{q}_i(b_i - a_i)$ in the second case with the profit the supplier can obtain in the third case. If

$$\begin{aligned}
&\sum_{i \notin S(T)} (w_2 - c - h_0(b_i - a_i))q_i(l^*,T) + \sum_{i \in S_1(T)} (w_2 - c - h_0(T))q_i(l^*,T) \\
\geq &\sum_{i \in \{j|\ j \notin S(T),\ \text{or}\ j \in S_1(T)\}} (w_1 - c - h_0(b_i - a_i))q_i^*(b_i - a_i, w_1)
\end{aligned} \tag{2.28}$$

the supplier should set $l = l^*$. Otherwise, l can be set as any constant which is larger then l^*. And it also implies that in this situation, retailers will place orders the same as under FP.

With the above analysis, we have the following Theorem for the supplier's optimal strategy with GBS.

Theorem 2.7. *When the supplier chooses the GBS as the pricing mechanism, $(w^*, w_2^*, T^*, l(w_2^*, T^*))$ is his optimal price curve decision, where w^* is given by equation (2.10),*

$$l^*(w_2^*, T^*) = \sum_{i \notin S(T^*)} \bar{q}_i(b_i - a_i, w_2^*(T)) + \sum_{i \in S_1(T^*)} \bar{q}_i(T^*, w_2^*(T^*)) \tag{2.29}$$

while w_2^* and T^* can be obtained by solving the following optimization problem

$$\begin{aligned} &\pi_S(w_2, T) \\ = &\max_{0 \le w_2 \le w^*, 0 \le T \le b_N - a_N} \{ \sum_{i \notin S(T)} (w_2 - c - h_0(b_i - a_i))\bar{q}_i(b_i - a_i, w_2) \\ &+ \sum_{i \in S_1(T)} (w_2 - c - h_0(t_i))\bar{q}_i(T, w_2) + \sum_{i \in S(T) \backslash S_1(T)} (w^* - c - h_0(b_i - a_i))q_i^*(b_i - a - i, w^*)\} \end{aligned} \tag{2.30}$$

Proof. The supplier will set GBS price curve parameters w_G to provide the discount based on the optimal FP price w^*. Hence we have $w_1 = w^*$ and $w_2 \le w^*$. According to Theorem 2.1, if the wholesale price is w^*, the retailer i's optimal order strategy is $(b_i - a_i, q_i^*(b_i - a_i, w^*))$. And the total order quantity from the retailers are $\sum_{i=1}^{N} q_i^*(b_i - a_i, w^*)$. Otherwise, if the wholesale price is w_2^* and the expiry time is T_i^*, according to Theorem 2.5, the Nash equilibrium of the retailers' order strategies can be derived. And it is suffices to show that

$$\begin{aligned} \pi_S \le &\sum_{i \notin S(T^*)} (w_2^* - c - h_0(b_i - a_i))\bar{q}_i(b_i - a_i, w_2) + \sum_{i \in S_1(T^*)} (w_2^* - c - h_0(t_i))\bar{q}_i(T^*, w_2) \\ &+ \sum_{i \in S(T^*) \backslash S_1(T^*)} (w^* - c - h_0(b_i - a_i))q_i^*(b_i - a - i, w^*) \end{aligned} \tag{2.31}$$

and furthermore,

$$\pi_S \le \sum_{i=1}^{N} (w^* - c - h_0(b_i - a_i))q_i^*(b_i - a - i, w^*) \tag{2.32}$$

Note that when $w_2^* = w^*$ and $T^* = b_N - a_N$, $\bar{q}_i(b_i - a_i, w_2) = q_i^*(b_i - a_i, w_2)$ and $S_1(T^*) = \emptyset$, hence the right hand side of the above two inequalities becomes the same. Therefore equation (2.30) holds. In other words, by setting $w_2^* = w^*$ and $T^* = b_N - a_N$ under GBS, the supplier at least can receive the same profit as he does under FP. And by solving optimization problem (2.30), we always find optimal w_2^* and T^* which bring the supplier more profit than FP. Finally, in order to motivate the retailers to order up to $\bar{q}_i(b_i - a_i, w_2^*)$ or $\bar{q}_i(T^*, w_2^*)$, according to Theorem 2.5, the supplier should set

$$l^*(w_2^*, T^*) = \sum_{i \notin S(T^*)} \bar{q}_i(b_i - a_i, w_2^*(T)) + \sum_{i \in S_1(T^*)} \bar{q}_i(T^*, w_2^*(T^*))$$

Thus, $(w^*, w_2^*, T^*, l^*(w_2^*, T^*))$ is the optimal price curve when the supplier choose the GBS as the pricing mechanism. □

Theorem 2.7 provides the supplier's optimal strategy on GBS setting and a natural question is how the retailers will respond to the supplier's strategy. Theorem 2.8 gives the answer to this question.

Theorem 2.8. *Under GBS pricing mechanism, the supplier's optimal strategy on setting the price curve is $w_G = (w^*, w_2^*, T^*, l^*(w_2^*, T^*))$. And there exists a unique Nash equilibrium which provides the optimal ordering strategies of the retailers, which is indicated as follows*

$$(t_i^e, q_i^e) = \begin{cases} (b_i - a_i, \bar{q}_i(b_i - a_i, w_2^*, \pi_i(b_i - a_i, w^*))) & \text{If } i \notin S(T^*) \\ (T^*, \bar{q}_i(T^*, w_2^*, \pi_i(b_i - a_i, w^*))) & \text{If } i \in S_1(T^*) \\ (b_i - a_i, q_i^*(b_i - a_i, w^*)) & \text{Otherwise.} \end{cases} \tag{2.33}$$

Proof. This theorem follows straightforwardly from Theorem 2.5 and Theorem 2.7. □

Theorem 2.8 ensures that a sub-game perfect equilibrium of the Stackberg Game always exists and further more is unique. Supplier, as the leader of this game, can set w_G appropriately to generate a most profitable ordering equilibrium for him while none of retailers will deviate from this equilibrium. In the next section, we will compare the returns of supplier and retailers under FP with their returns under GBS. Through the comparison we can observe which of the two pricing mechanisms will be more favorable.

2.5. Comparison of Supplier Chain Member's profits under GBS and FP

2.5.1. Profit comparison under GBS and flat price scheme

In this section, we compare the profits the supplier and retailers obtained under two different pricing scheme: GBS and FP. From the definition of $\bar{q}_i(t, w_2)$ and Theorem 2.8, it is obvious that the the retailers' profits are always the same under both pricing mechanisms. But retailers' order quantities increase under GBS and they can use the extra inventories to raise their service levels. This implies the GBS will benefit the retailers. On the other side, we can also conclude from Theorem 2.7 that the supplier will receive more profit under GBS pricing mechanism than he does under FP. Generally we have the following theorem:

Theorem 2.9. *The GBS pricing mechanism can bring win-win situation to the supplier and retailers. The improvements are higher profit for the supplier and advanced service levels for the retailers.*

2.5.2. Numerical study

In this section, we will use some numerical example to show the improvements of GBS can bring to the supplier and retailers. As mentioned in Theorem 2.9, the improvement for the supplier is higher profit while for the retailers is more products (advanced service levels). And we will also examine the impacts of different parameters in our model. These impacts can not be obviously derived from our theoretical results but using the numerical experiments, we can observe some insights. The following parameters are included:

- The different price-elasticity index in the market, $k_i, i = 1, ..., N$;
- The fluctuation of the unit holding cost, h_0 and the deteriorating rate of the product;
- The different transportation time a_i and due time $b_i, i = 1, ..., N$;

In our numerical study we consider a model with five retailers and one supplier. Each retailer faces a newsvendor demand with a random variable ε_i, which follows a uniform distribution $U(0,1)$. We evaluated in different situations the supplier's expected profit and the retailers' service level, which can be indicated from the retailers' order quantities. First we initialize the base parameters as follows:

- For the product, the deterioration follows exponential distribution and the deteriorating rate $\alpha = 0.05$. Hence the degree of freshness of the product at time t should be $e^{-0.1(t-t^o)}$ where t^o is the order (transported) time.
- For the supplier, the production cost $c = 5$ and unit holding cost $h_0 = 1.2$;
- For the retailers, unit holding cost $h_i = 0.2$ and transportation time $a_i = 2$ $i = 1, ..., 5$. The market open time is $b = [2.5, 3.5, 4.5, 5.5, 6.5]$;
- For the market demands, the price-elasticity index k_i follows uniform distribution $U(1.5, 2.5)$.

We first check the impact of price-elasticity index. Initially we will generate a group of price-elasticity index $k = (k_1, ...k_5)$, each of whose components follow uniform distribution $U(1.5, 2.5)$. Hence we obtain $k =$ [1.71, 1.90, 1.96, 2.29, 2.32] randomly. Then we vary the value of k form $k - 0.3$ to $k + 0.2$ by adding 0.05 each time. This implies that the market demands now are more and more sensitive to the retailer prices. The optimal results of different parties under FP and GBS are listed in Table 4.1. The first column $\bar{k}$ is the average price-elasticity index. The columns w^* and w_2^* are the optimal two-level wholesale prices. The fourth column states the higher profit supplier can achieve under GBS. And for each retailer, the new optimal order time and increase in order quantity are listed. (Order time under FP for retailer i is $b_i - a_i$. Hence, the original order times are $[0.5, 1.5, 2.5, 3.5, 4.5]$.) From Table 2.1 we have the following observations:

Table 2.1: Comparison under varied price-elasticity indexes

$\bar{k}$	w^*	w_2^*	Supplier			Retailer1		Retailer2		Retailer3		Retailer4		Retailer5	
			FP	GBS		GBS		GBS		GBS		GBS		GBS	
			π_S^*	T^*	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%
1.89	16.9	10.6	7990	3.5	43	0.5	414	1.5	126	2.5	133	3.5	171	3.5	60
1.94	16.4	10.1	6976	3.4	41.9	0.5	451	1.5	139	2.5	146	3.4	174	3.4	61
1.99	15.4	9.5	6087	2.4	35.9	0.5	465	1.5	144	2.4	138	2.4	62	4.5	0
2.04	15.4	9.1	5310	2.3	34.5	0.5	536	1.5	172	2.3	152	2.3	72	4.5	0
2.09	14.9	8.7	4633	2.2	31.5	0.5	579	1.5	186	2.2	150	2.2	72	4.5	0
2.14	14.4	8.3	4042	1.4	26.3	0.5	624	1.4	185	1.4	70	3.5	0	4.5	0
2.19	13.9	7.9	3526	1.4	18.7	0.5	672	1.4	201	1.4	79	3.5	0	4.5	0
2.24	13.9	7.6	3077	1.2	15.5	0.5	772	1.2	201	1.2	79	3.5	0	4.5	0

- As k_i increases, the suppliers' profit will decrease under both FP and GBS. Meanwhile the improvement under GBS compared with FP is decreasing from 43% to 15.5%. Furthermore, the time of expiry of the GBS pricing mechanism becomes earlier. Thus GBS is more efficient for the supplier when the market becomes less sensitive to the retail price.

- Although in all cases the service levels of retailers will be advanced under GBS, the improvement rates are different. Basically, for retailer i, if his original order time $b_i - a_i$ is earlier than T, then the improvement of his service level under GBS will keep increasing on k. Otherwise, the improvement will becomes less. The reason is that the retailer in this case has to place earlier orders which brings him extra loss from holding cost and more deteriorated products to be sold at the market. Hence he should order less to avoid such loss.

- In each replication of $k = (k_1, ..., k_5)$, we can observe that if k_i is larger, the improvement of service levels for retailer i is greater. Hence, when the market demand is more sensitive to the retailer price compared with other markets demands, then the GBS pricing mechanism can improve the retailer's service level more.

Secondly, we would like to check the impact of the supplier's unit holding cost h_0 and the deteriorating rate of the product α. We first vary h_0 from 1 to 1.7 with fixed $\alpha = 0.5$. With each value of h_0, we find the optimal strategies of supplier and retailers under both FP and GBS. Then we compare their optimal performances under two pricing mechanism and the result is summarized in Table 2.2. And then we fix h_0 and change α from 0.35 to 0.7 by adding 0.05 each time. Again we calculate the optimal values for each members both under FP and GBS. The results are summarized in Table 2.3.

Table 2.2: Comparison under different unit holding cost of supplier, h_0

h_0	w^*	w_2^*	Supplier			Retailer1		Retailer2		Retailer3		Retailer4		Retailer5	
			FP	GBS		GBS		GBS		GBS		GBS		GBS	
			π_S^*	T	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%
1	14.5	8.9	5542	3.3	32.3	0.5	483	1.5	153	2.5	160	3.3	174	3.3	60
1.1	15	9	5423	2.3	34.9	0.5	507	1.5	160	2.3	140	2.3	63	4.5	0
1.2	15.4	9.1	5310	2.3	34.5	0.5	536	1.5	172	2.3	152	2.3	72	4.5	0
1.3	15.4	9.2	5204	2.3	34	0.5	531	1.5	167	2.3	147	2.3	69	4.5	0
1.4	15.8	9.2	5103	2.2	34.2	0.5	560	1.5	179	2.2	145	2.2	69	4.5	0
1.5	16.3	9.4	5006	2.2	37.4	0.5	585	1.5	186	2.2	151	2.2	74	4.5	0
1.6	16.7	9.4	4913	2.1	37.9	0.5	616	1.5	198	2.1	148	2.1	73	4.5	0
1.7	17.2	9.5	4825	1.4	38.7	0.5	646	1.4	194	1.4	77	3.5	0	4.5	0

Table 2.3: Comparison under different deteriorating rate, α

α	w^*	w_2^*	Supplier			Retailer1		Retailer2		Retailer3		Retailer4		Retailer5	
			FP	GBS		GBS		GBS		GBS		GBS		GBS	
			π_S^*	T	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%
0.35	15.4	10.2	7168	3.1	47.3	0.5	392	1.5	119	2.5	124	3.1	119	3.1	50
0.4	15.4	9.8	6486	3.2	43.1	0.5	436	1.5	136	2.5	143	3.2	146	3.2	60
0.45	15.4	9.5	5869	2.2	40.8	0.5	482	1.5	150	2.2	122	2.2	58	4.5	0
0.5	15.4	9.1	5310	2.3	34.5	0.5	536	1.5	172	2.3	152	2.3	72	4.5	0
0.55	15.4	8.8	4805	2.3	30.4	0.5	592	1.5	190	2.3	166	2.3	75	4.5	0
0.6	15.4	8.5	4348	2.3	25.1	0.5	653	1.5	209	2.3	182	2.3	78	4.5	0
0.65	15.4	8.1	3934	1.6	14.7	0.5	727	1.5	239	1.6	88	3.5	0	4.5	0
0.7	15.4	7.8	3560	1.6	6.8	0.5	803	1.5	264	1.6	93	3.5	0	4.5	0

From Table 2.2 and Table 2.3, we can conclude that:

- For the supplier, as h_0 increases, his profits obtained under both FP and GBS are decreasing. But the improvement of GBS generally becomes greater. But when α increases, the improvement of GBS becomes less. Therefore, GBS will bring more improvement when the supplier's holding cost to keep the product fresh is high. Because in this situation, the supplier will set the expiry time T earlier so that his holding cost can be reduced. And when product is easy to go bad, the improvement turns out to be less. An important reason for such a situation is that when the products deteriorate very fast, the retailers are not willing to place earlier orders, which may bring them losses from selling less fresh products. So under GBS the supplier's saving on holding cost becomes less.

- For the retailers, when h_0 and α increase, the improvements of GBS are quite different. For retailers who originally order later than others, the improvements even become 0. This is because the expiry time set by the supplier is too early and it will be unwise for them to place order before T. But for the retailers who originally order earlier than others, their improvements increase a lot. The reason is the expiry time is still comfortable for them. They can still order enough products to enjoy the price reduction.

Finally we focus on the impact of the different transportation times a_i and market open times b_i $i = 1, ..., 5$. We will first fix $a = 2$ and vary b from $b - 0.5$ to $b + 0.25$ by adding 0.25 each time. In this way, we actually delay the retailers' original order times under FP. But since the transportation times are fixed, the freshness of the products when they arrive at the markets remains unchanged. The result is summarized in Table 4.4 where the column 1 represents the average of b_i, $i = 1, ..., 5$. Then in Table 2.5 we fix $b - a = [0.5, 1.5, 2.5, 3.5, 4.5]$ and change a from 1.7 to 2.4. Hence the transportation times become longer while the order times under FP keep the same. The purpose is to examine the impact of transportation time to the efficiency of GBS compared with FP, and the result is listed in Table 2.5. From these two tables, we have the following observations:

Table 2.4: Comparison under different market open time b

$\bar{b}$	w^*	w_2^*	Supplier			Retailer1		Retailer2		Retailer3		Retailer4		Retailer5	
			FP	GBS		GBS		GBS		GBS		GBS		GBS	
			π_S^*	T	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%
4	13.8	8.8	5735	2	41.3	0	440	1	135	2	142	2	61	4	0
4.25	14.6	8.9	5515	2.1	36.8	0.3	490	1.3	156	2.1	137	2.1	60	4.3	0
4.5	15.4	9.1	5310	2.3	34.5	0.5	536	1.5	172	2.3	152	2.3	72	4.5	0
4.75	15.7	9.2	5123	2.4	28.9	0.8	553	1.8	176	2.4	142	2.4	67	4.8	0
5	16.5	9.4	4948	2.7	26.2	1	601	2	191	2.7	156	2.7	78	5	0
5.25	16.8	9.4	4785	2.8	16.1	1.3	623	2.3	201	2.8	151	2.8	76	5.3	0
5.5	17.6	9.6	4634	3.1	12.9	1.5	673	2.5	216	3.1	164	3.1	86	5.5	0
5.75	18.4	9.7	4492	2.7	6	1.8	728	2.7	220	2.7	94	4.8	0	5.8	0

Table 2.5: Comparison under different transportation time a

$\bar{a}$	w^*	w_2^*	Supplier			Retailer1		Retailer2		Retailer3		Retailer4		Retailer5	
			FP	GBS		GBS		GBS		GBS		GBS		GBS	
			π_S^*	T	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%	t	Inc%
1.8	15.4	9.5	5869	3.3	40.3	0.5	482	1.5	150	2.5	158	3.3	171	3.3	59
1.9	15.4	9.3	5582	2.3	37.7	0.5	508	1.5	161	2.3	141	2.3	64	4.5	0
2	15.4	9.1	5310	2.3	34.5	0.5	536	1.5	172	2.3	152	2.3	72	4.5	0
2.1	15.4	9	5051	2.2	34.9	0.5	561	1.5	177	2.2	143	2.2	67	4.5	0
2.2	15.4	8.8	4805	2.2	31.1	0.5	592	1.5	190	2.2	154	2.2	76	4.5	0
2.3	15.4	8.6	4571	2.1	27.2	0.5	625	1.5	203	2.1	152	2.1	75	4.5	0
2.4	15.4	8.5	4348	2.1	26.6	0.5	653	1.5	209	2.1	158	2.1	80	4.5	0
2.5	15.4	8.3	4136	2	22	0.5	689	1.5	224	2	155	2	79	4.5	0

- When $\bar{b}$, the average of b_i, increases from 4 to 5.75, the supplier's wholesale prices also rise due to the increasing holding cost. But his profits under both FP and GBS drop. Meanwhile the improvement of GBS to the supplier becomes less. This implies that GBS will be more efficient if the original order times of the retailers are earlier. But for the retailers, the improvement of GBS is greater when b_i is larger.

- When $\bar{a}$, the average transportation time of retailers, increases from 1.8 to 2.5, the original wholesale price w^* remains the same because $b - a$ is fixed. But the reduced wholesale price w_2^* drops and so do the supplier's profit under both FP and GBS. The reason is that due to the long transportation, products are less fresh which leads to smaller demand. Furthermore, the GBS also becomes less efficient.

2.6. Conclusions and Future work

We study a two-echelon supply chain, where one supplier sells a kind of perishable product to N retailers. We derive the optimal strategies of all the chain members under both FP and the GBS pricing mechanisms. We proved that GBS will always bring a win-win situation to all the chain members in the sense that suppliers can receive more profit and retailers can advance their service levels. So far we have assumed that supplier's products are always sufficient to satisfy the market demands. A natural question is that if there is a capacity restriction of the product, Q, then what will be the optimal strategies of the chain members? Note that in such case, supplier may set $l = Q$ if Q is not sufficient to satisfy the maximum demands under GBS pricing mechanism. Then retailers' problem on determine the order time and order quantity under GBS requires further consideration. Another extension can be the cooperation-retailers scenario, in which inside trans-payment among the retailers is allowed. In this situation, retailers may form coalitions to pursue more profit and the problem needs to be solved by applying the *Cooperative Game Theory*. So far in our model, we assume the information of each retailer is public to all the chain members. This is a necessary assumption and without it the Nash equilibrium will be difficult to achieve. But in practice, sometimes it is difficult to persuade all the retailers to share their information. But if we consider the situation that retailers can cooperate with each other, then information will automatically be shared among retailers who belong to the same coalition. Therefore the model may become more convinced.

As mentioned above, due to the information sharing restriction in practical problems, models constructed with *Competition Game Theory* may not be appropriate for analyzing some multi-player problems where information is not public. In the next chapter, we move our attention to another important topic in Game Theory: *Cooperative Game Theory.* With solution concepts and theocratical results in *Cooperative Game Theory*, we can solve a group of problems in which information is not shared unless coalitions are formed. In particularly we want to check whether the *Cooperative Game Theory* can also be applied to model and solve problems in the SCM area.

2.7. Appendix

[1] Supplement to the proof of Lemma 2.3

- According to equation (2.19), $q_i^*(t_i, w_2) = \frac{y_i\theta(b_i - t_i)z_i^*\bar{F}^{k_i}(z_i^*)}{(w_2 + h_i(b_i - a_i - t_i))^{k_i}}$. Since $\theta(b_i - t_i)$ is increasing on t_i and $h_i(b_i - a_i - t_i)$ is decreasing on t_i, we know $q_i^*(t_i, w_2)$ is increasing on t_i.

- According to equation (2.19) and (2.17), we have

$$\pi_i(t_i, q_i^*(t_i, w_2), p_i^*(t_i, q_i^*(t_i, w_2)) |\, t_{-i}, q_{-i}, w_2) = \frac{y_i\theta(b_i - t_i)z_i^*\bar{F}^{k_i}(z_i^*)}{(k_i - 1)(w_2 + h_i(b_i - a_i - t_i))^{k_i - 1}}$$

  Since $\theta(b_i - t_i)$ is increasing on t_i and $h_i(b_i - a_i - t_i)$ is decreasing on t_i, we know $\pi_i(t_i, q_i^*(t_i, w_2), p_i^*(t_i, q_i^*(t_i, w_2)) |\, t_{-i}, q_{-i}, w_2)$ is increasing on t_i.

- According to equation (2.17), we know $\pi_i(t_i | q_i, p_i^*(t_i, q_i), w_2)$ is increasing on t_i. Therefore $\pi_i(t_i, q_i(l, T), p_i^*(t_i, q_i(l, T)) |\, t_{-i}, q_{-i}, w_2)$ is increasing on t_i.

- With the definition of $\bar{q}_i(t_i, w_2, \pi(b_i - a_i, w_1))$, we have

$$\pi_i(t_i, \bar{q}_i(t_i, w_2, \pi(b_i - a_i, w_1)), p_i^*(t_i, \bar{q}_i) | w_2) = \frac{y_i\theta(a_i)z_i^*\bar{F}^{k_i}(z_i^*)}{(k_i - 1)w_1^{k_i - 1}}$$

According to equation (2.17), we know $\pi_i(t_i|q_i, p_i^*(t_i, q_i), w_2)$ is increasing on t_i. Therefore, given $\forall t_i^1 > t_i^2$, we have

$$\begin{aligned} & \pi_i(t_i^2, \bar{q}_i(t_i^2, w_2, \pi_i(b_i - a_i, w_1)), p_i^*(t_i^2, \bar{q}_i)|w_2) \\ = \; & \pi_i(t_i^1, \bar{q}_i(t_i^1, w_2, \pi_i(b_i - a_i, w_1)), p_i^*(t_i^1, \bar{q}_i)|w_2) \\ > \; & \pi_i(t_i^2, \bar{q}_i(t_i^1, w_2, \pi_i(b_i - a_i, w_1)), p_i^*(t_i^2, \bar{q}_i)|w_2) \end{aligned}$$

which implies that $\bar{q}_i(t_i^2, w_2, \pi(b_i - a_i, w_1)) < \bar{q}_i(t_i^1, w_2, \pi(b_i - a_i, w_1))$. In other words, $\bar{q}_i(t_i, w_2, \pi_i(b_i - a_i, w_1))$ is increasing on t_i.

CHAPTER 3

LINEAR PROJECT GAME: SUBCONTRACTING AND COOPERATION

In the previous chapter, we study an application of *Competitive Game Theory* in a two-echelon supply chain problem. The result shows that the new pricing mechanism (GBS) will benefit all the chain members. In this chapter, we continue our study on a two-echelon project management problem which is different from the traditional project scheduling problems. Specifically, in this problem we have a Project Owner (PO) and numbers of Subcontractors (SCs). The PO who has secured a project will subcontract all the tasks in the project network to different SCs. Getting rid of the processing costs, SCs gain profits from PO's payments on completing the tasks. And PO's profit comes from the entire return of the project completion deducted by the total payments to the SCs. In fact this problem is quite similar as the supply chain problem we discussed in the last chapter, where we also have an "Upstream Supplier" (PO) and numbers of "Downstream Retailers" (SCs). The "Supplier" provides certain kind of product (task) to each "Retailer" with "wholesale price" (cost to finish the task). And the payment from PO, which is the "cost" of PO, can be regarded as the "retail price" for the retailer. Therefore, each member of this "supply chain" should find out his optimal strategy to maximize his own profit.

However, in this chapter we model the downstream SCs' problem as a *Cooperative Game*. Applying the solution concepts and advanced techniques in *Cooperative Game Theory*, we can prove that SCs will always be better off by cooperating with each other and forming a grand coalition.

3.1. Introduction

The use of project management as a methodology for planning and implementing business activities has increased greatly in recent years. As evidence, membership in the Project Management Institute has increased from 50,000 in 1996 to over 500,000 today. There appear to be several reasons for this trend. The first reason is motivation provided by newly developing applications, for example information technology, that have substantial further growth potential. A second reason is the use of shorter life cycles for products and services (Value Based Management.net 2009), which motivates the use of project management methodology to bring new products and services to market more quickly. A third reason is the usefulness of project management in effectively implementing organizational change, as a response to new technologies, more intense competition, and more demanding and less predictable customers (1000ventures.com 2009).

Our work considers several issues in connection with the scheduling of resource constrained projects. Recent overviews of the related research literature are provided by Herroelen et al. (1998) and Brucker et al. (1999). These works document the intractability of the resource constrained project scheduling problem, and discuss many attempts to solve it. Most of this literature considers the project owner as a single decision maker who owns all the necessary resources and controls all the tasks. For example, scheduling decisions that involve expediting or "crashing" the individual tasks are made by the project owner with a view to optimizing some overall time and cost objective for the project. However, this perspective ignores the role of *outsourcing*, which is the procurement of products or services from external source providers. In all but the smallest of projects the project owner has to use the services of other suppliers (Turner 2003). Outsourcing is major business activity with a value of about $530 billion in business processes and information technology alone, and a predicted annual growth rate of 8–10% (Garib 2009).

In order to focus on the role of outsourcing in projects, we consider a project in which all the tasks are outsourced. However, our results are also applicable to projects where only some of the tasks are outsourced, and others are managed directly by the project owner. The subcontractors can process their tasks independently. However, due to precedence relationships between the tasks and their limited resources, the subcontractors can also cooperate with each other. If they cooperate, then they share their resources and also the resulting profit. An important issue is how to allocate the profit in such a way that all the subcontractors cooperate. We model this problem as a cooperative game (Peleg and Sudhölter 2003). Related works on cooperative games include Owen (1975), Granot (1986), and Samet and Zemel (1984).

Several previous studies apply cooperative game concepts to project management. Bergantiños and s'anchez (2002a) consider a project that experiences a delay in completion time. As a result of lost revenue, additional cost incurred, or contract penalties, the client may be entitled to compensation. A natural question is which tasks, and which task operators, are responsible for the delay. Moreover, the answer to this question may suggest how the cost of compensation may be divided among the task owners. They propose a cost sharing rule based on cooperative game theory, under the assumption that the project is delayed. Brânzei et al. (2002) provide two alternative game theory approaches to this problem. The first approach is based on an optimistic and a pessimistic game, and the second approach is based on a game between paths and a serial cost sharing rule. The latter approach is extended to a weighted serial cost sharing rule by Castro et al. (2008). In all three of the above studies, it is assumed that the project is delayed and no incentives are available for expediting the tasks.

Our work focuses on a more general situation where the project is not necessarily delayed, due to the availability of incentives for expediting the tasks and the imposition of penalties for late completion of the tasks. This problem is studied by Estévez-Fernández et al. (2005) using an expedited project game where tasks cannot be delayed. This game is shown to be convex. The authors also consider a *project game* where some tasks are delayed and other are expedited. This environment is also the focus of our work. However, Estévez-Fernández et al (2005) make a simplifying assumption. In defining the value of a coalition, it is assumed that the delayable tasks are indeed delayed and the expeditable tasks outside the coalition are completed ontime. Under this condition, the game is convex. Castro et al. (2007) also consider a project game, where the characteristic function for a given set of activities represents the amount of expediting that these activities induce in the project. Using various concepts of slack from classical project management, the authors show that the game is balanced.

However, the problem we consider is more general, in that we measure cost rather than time. That is, different tasks may have different per time unit incentives for expeditng, and different per time unit penalties for late completion. This generalization is natural in a setting where there is an overall project owner besides the various subcontractors. Moreover, we allow the subcontractors to have different crashing costs, which is representative of actual practice when their tasks are not necessarily similar. We show that this more general game is balanced, and provide a closed form expression for a core solution.

Since the subcontractors process their tasks according to the requirements specified by the project owner, we consider the issue of contract design from the perspective of the project owner. This problem is modeled as a Stackelberg game, where the project owner is the leader of the game who specifies the contract, and the subcontractors are followers who cooperate with each other to achieve maximum profit. Two alternative situations are considered. In the first situation, both parties have full information. In the second situation, the project owner does not have information about the crash cost or resource usage of the subcontractors. We study how the project owner should design the contract to maximize its profit. Also, we study to what extent and how the subcontractors should cooperate.

The rest of this chapter is arranged as follows: We first elaborate the project subcontracting problem and model setting in Section 3.2; Then we introduce our "Linear Project Game" in Section 3.3 and show that the game is balanced. In section 3.4 and 3.5, we will further discuss the PO's optimal strategy on designing the contracts according to the SCs' reactions. Basically in Section 3.4 we investigate the situation where crashing information is freely shared and in Section 3.5, we focus on the cases when information is asymmetric between PO and SCs. Afterwards in Section 3.6, we introduce another problem associated with project subcontracting. In stead of single project subcontracting, we study a multiple project subcontracting problem in this section. Again we prove the new cooperative game is balanced. Finally in Section 3.7, we generally conclude all the results obtained in this chapter and discuss the future work.

3.2. Preliminaries

We start with a project which a project owner has contracted with a client. The project contains several tasks with precedence relations between them, which can be characterized as an acyclic network. However, the project owner has no resources for performing the tasks; hence, all the tasks will be outsourced to subcontractors with the necessary facilities and expertise to execute them. For ease of exposition, we assume that each task is undertaken by one subcontractor, selected for that task by a tendering process. The subcontractor offers to complete the task at a *primary* completion time in exchange for for a primary agreed payment from the project owner. Together, these constitute the primary part of the contract between the project owner and the subcontractors.

Based on its contract with the client, the project owner has a cost associated with the completion time of each task in the project. This completion time cost is a nondecreasing function of the completion time. A task may be crashed so as to reduce its completion cost; however, this incurs an extra crashing cost and requires additional resources at the relevant subcontractor. The project owner's objective is to maximize its total profit of completing the project. Given a fixed price contract with the client, this is equivalent to minimizing the total project cost. To achieve this, the project owner signs a two part contract with each subcontractor. The contract specifies:

(a) Primary completion time and payment.

(b) A secondary option of crashing, including:

i) a linear bonus rate payable by the project owner to the subcontractor for task completion before the primary completion time,

ii) a linear cost rate for crashing the task below its normal duration, payable by the project owner to the subcontractor, and

iii) the ranges of allowable amount of crashing and task completion time.

In specifying the contract, the project owner needs to know the normal task duration. We assume that this information is provided by the subcontractor as part of the original tender. by the subcontractor. While the project owner does not necessarily know some other parameters, such as the subcontractor's crashing cost and resource availability, these are not necessary for contract specification s described above. Also, we may later allow the crashing to be negative, to allow the subcontractor to lengthen the duration of the task, thereby releasing resources that can be shared by other cooperating subcontractors.

Let $\mathcal{N} = \{1, \ldots, n\}$ denote the set of subcontractors, or equivalently tasks. Without loss of generality, we assume that task 1 is a dummy start task without predecessors, and n is a dummy end task without successors. We define the following notation:

d_i :	Normal duration of task i
x_i :	Number of time units to crash task i
θ_i :	Cost for subcontractor i if task i is crashed one unit of time
ϑ_i :	Cost for the project owner if task i is crashed one unit of time
Θ_i :	Unit crashing cost of task i; $\Theta_i = \theta_i + \vartheta_i$
$\mathcal{R}$:	Set of all types of resources
m :	Number of resource types($\lvert\mathcal{R}\rvert$)
r_{ij} :	Unit resource j consumed by crashing task i
$\overline{x}_i$:	Maximum number of time units that task i can be crashed
$\overline{r}_{ij}$:	Upper bound of the resource j available to subcontractor i
t_i^0 :	Nomal completion time of task i
t_i :	Expedited completion time of task i, with $t_i \leq t_i^0$
$\overline{t}_i$:	Upper bound of t_i
γ_i :	Normal payment to subcontractor i if task i is completed no later than t_i^0
β_i :	Bonus to subcontractor i if task i is completed one unit of time earlier than t_i^0
$\varphi_i t_i$:	Cost incurred to the project owner if task i is completed at time t_i
$\mathcal{P}^i$:	Set of tasks that are immediate predecessors of task i
$\mathcal{A}^i$:	Set of tasks that are immediate successors of task i
D :	Deadline of the project
$\pi(\mathcal{S})$:	Maximum total profit if all subcontractors in coalition S act optimally
$c(\mathcal{S})$:	Minimium total cost if all subcontractors in coalition S act optimally

3.2.1. Subcontractor's problem

After signing the contract with the project owner, a subcontractor can either work independently, or join a coalition and cooperate with other subcontractors. If he works alone, his problem is to find the optimal crashing amounts $x_1^*, \ldots, x_n^*$ and completion times $t_1^*, \ldots, t_n^*$ that maximize his total profit:

$$
\begin{aligned}
\pi_i &= \max_{\{x_i,t_i\}} \{\beta_i(t_i^0 - t_i) - \theta_i x_i + \gamma_i\} \\
s.t. \quad t_i - d_i + x_i &\geq t_j^0, \qquad \forall j \in \mathcal{P}^i \\
t_i &\leq t_i^0 \\
t_1 - d_1 + x_1 &= 0 \qquad \text{if } i = 1 \\
t_n &\leq D \qquad \text{if } i = n \\
\sum_{i \in \mathcal{N}} r_{ij} \cdot x_i &\leq \overline{r}_{ij}, \qquad \forall j \in \mathcal{R} \\
0 \leq x_i &\leq \overline{x}_i \\
0 \leq t_i &\leq \overline{t}_i
\end{aligned}
$$

Denote this problem as P1(i). Let the feasible set of P1(i) be denoted as $\mathcal{F}^i$.

A coalition S of subcontractors solves the following optimization problem:

$$\pi(\mathcal{S}) = \max_{\{x_i,t_i\}} \sum_{i \in \mathcal{S}} \{\beta_i(t_i^0 - t_i) - \theta_i x_i + \gamma_i\} \tag{3.1}$$

s.t.

$$\begin{cases} t_i - d_i + x_i \geq t_j, & \forall i \in \mathcal{S}, j \in \mathcal{P}^i \bigcap \mathcal{S} \\ t_i - d_i + x_i \geq t_j^0, & \forall i \in \mathcal{S}, j \in \mathcal{P}^i \setminus \mathcal{S} \\ t_i \leq t_i^0, & \forall i \in \mathcal{S}, j \in \mathcal{A}^i \setminus \mathcal{S} \end{cases} \tag{3.1.1}$$

$$\begin{cases} t_1 - d_1 + x_1 \leq 0, & \text{if } 1 \in \mathcal{S} \\ t_1 - d_1 + x_1 \geq 0, & \text{if } 1 \in \mathcal{S} \\ t_n \leq D, & \text{if } n \in \mathcal{S} \end{cases} \tag{3.1.2}$$

$$\sum_{i \in \mathcal{S}} r_{ij} \cdot x_i \leq \sum_{i \in \mathcal{S}} \overline{r}_{ij} \quad \forall j \in \mathcal{R} \tag{3.1.3}$$

$$0 \leq x_i \leq \overline{x}_i, \quad \forall i \in \mathcal{S} \tag{3.1.4}$$

$$0 \leq t_i \leq \overline{t}_i, \quad \forall i \in \mathcal{S} \tag{3.1.5}$$

Similarly, denote this problem as P1($\mathcal{S}$). Let the feasible set of P1($\mathcal{S}$) be $\mathcal{F}(\mathcal{S})$. Note that constraints (3.1.1) represent the precedence relations among tasks whereas constraints (3.1.2) state the project start time and completion time restrictions.

Let $\mathbf{x}_\mathcal{N} = (x_1, x_2, \cdots, x_n)^\top$, $\mathbf{t}_\mathcal{N} = (t_1, t_2, \cdots, t_n)^\top$, $\mathbf{t}^0_\mathcal{N} = (t^0_1, t^0_2, \cdots, t^0_n)^\top$, $\mathbf{r}_{\mathcal{N}j} = (r_{1j}, r_{2j}, \cdots, r_{nj})^\top$, $\forall j \in \mathcal{R}$, $\mathbf{R}^\top_\mathcal{N} = (\mathbf{r}_{\mathcal{N}1}, \mathbf{r}_{\mathcal{N}2}, \cdots, \mathbf{r}_{\mathcal{N}m})$, $\bar{\mathbf{r}}_\mathcal{N} = (\sum_{i\in\mathcal{N}} \bar{r}_{i1}, \sum_{i\in\mathcal{N}} \bar{r}_{i2}, \cdots, \sum_{i\in\mathcal{N}} \bar{r}_{im})^\top$, $\bar{\mathbf{x}}_\mathcal{N} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n)^\top$, $\bar{\mathbf{t}}_\mathcal{N} = (\bar{t}_1, \bar{t}_2, \cdots, \bar{t}_n)^\top$, $\beta_\mathcal{N} = (\beta_1, \beta_2, \cdots, \beta_n)^\top$, $\theta_\mathcal{N} = (\theta_1, \theta_2, \cdots, \theta_n)^\top$, and $\gamma_\mathcal{N} = (\gamma_1, \gamma_2, \cdots, \gamma_n)$. Define $\mathbf{0} = (0, 0, \cdots, 0)^\top$, $\mathbf{1} = (1, 1, \cdots, 1)^\top$, and $\mathbf{I}$ as an identity matrix of appropriate dimension. Then, the problem for the grand coalition, P1($\mathcal{N}$), can be rewritten in the matrix form as follows:

$$\pi(\mathcal{N}) = \max_{\{\mathbf{x}_\mathcal{N}, \mathbf{t}_\mathcal{N}\}} \quad \beta^\top_\mathcal{N}(\mathbf{t}^0_\mathcal{N} - \mathbf{t}_\mathcal{N}) - \theta^\top_\mathcal{N}\mathbf{x}_\mathcal{N} + \mathbf{1}^\top\gamma_\mathcal{N} \tag{3.2}$$

s.t.

$$\mathbf{A}^1_\mathcal{N}\mathbf{x}_\mathcal{N} + \mathbf{A}^2_\mathcal{N}\mathbf{t}_\mathcal{N} \leq \mathbf{b}^1_\mathcal{N} \tag{3.2.1}$$

$$\mathbf{B}^1_\mathcal{N}\mathbf{x}_\mathcal{N} + \mathbf{B}^2_\mathcal{N}\mathbf{t}_\mathcal{N} \leq \mathbf{b}^2_\mathcal{N} \tag{3.2.2}$$

$$\mathbf{R}_\mathcal{N}\mathbf{x}_\mathcal{N} \leq \bar{\mathbf{r}}_\mathcal{N} \tag{3.2.3}$$

$$\mathbf{I}\ \mathbf{x}_\mathcal{N} \leq \bar{\mathbf{x}}_\mathcal{N} \tag{3.2.4}$$

$$\mathbf{I}\ \mathbf{t}_\mathcal{N} \leq \bar{\mathbf{t}}_\mathcal{N} \tag{3.2.5}$$

$$\mathbf{x}_\mathcal{N} \geq \mathbf{0}; \quad \mathbf{t}_\mathcal{N} \geq \mathbf{0} \tag{3.2.6}$$

where $\mathbf{A}^i_\mathcal{N}$, $\mathbf{B}^i_\mathcal{N}$, and $\mathbf{b}^i_\mathcal{N}(i)$, $i = 1, 2$, are appropriately defined matrices and vectors to represent the constraints (3.1.1) and (3.1.2), which are shown in the Appendix A[1]. See Example 3.2 in Appendix B for an illustration of these matrices.

For any coalition $\mathcal{S}$, we can also appropriately define the corresponding matrices and vectors similar to those for the grand coalition, so that P1($\mathcal{S}$) can be rewritten in the matrix form as follows:

$$\pi(\mathcal{S}) = \max_{\{\mathbf{x}_\mathcal{S},\mathbf{t}_\mathcal{S}\}} \quad \beta_\mathcal{S}^\top(\mathbf{t}_\mathcal{S}^0 - \mathbf{t}_\mathcal{S}) - \theta_\mathcal{S}^\top \mathbf{x}_\mathcal{S} + \mathbf{1}^\top \gamma_\mathcal{S} \tag{3.3}$$

s.t.

$$\mathbf{A}_\mathcal{S}^1 \mathbf{x}_\mathcal{S} + \mathbf{A}_\mathcal{S}^2 \mathbf{t}_\mathcal{S} \leq \mathbf{b}_\mathcal{S}^1 \tag{3.3.1}$$

$$\mathbf{B}_\mathcal{S}^1 \mathbf{x}_\mathcal{S} + \mathbf{B}_\mathcal{S}^2 \mathbf{t}_\mathcal{S} \leq \mathbf{b}_\mathcal{S}^2 \tag{3.3.2}$$

$$\mathbf{R}_\mathcal{S} \mathbf{x}_\mathcal{S} \leq \bar{\mathbf{r}}_\mathcal{S} \tag{3.3.3}$$

$$\mathbf{I}\ \mathbf{x}_\mathcal{S} \leq \overline{\mathbf{x}}_\mathcal{S} \tag{3.3.4}$$

$$\mathbf{I}\ \mathbf{t}_\mathcal{S} \leq \bar{\mathbf{t}}_\mathcal{S} \tag{3.3.5}$$

$$\mathbf{x}_\mathcal{S} \geq \mathbf{0}; \quad \mathbf{t}_\mathcal{S} \geq \mathbf{0} \tag{3.3.6}$$

Remarks:

- If $\beta_i \geq 0\ \forall i \in \mathcal{N}$, then $\bar{t}_i$ can be simply set as t_i^0. The reason is that in this case, subcontractors will never complete their jobs later than the normal completion times in order to avoid penalties. If we allow the task i to be lengthened so that extra resource can be released to share with other tasks, then we may set $\bar{t}_i \geq t_i^0$.

- Two types of benefits can be achieved by members in a coalition to cooperate: (i) they can share resources, to achieve a better crashing solution; and (ii) they can coordinate their completion times, to achieve a better timing solution.

- We should consider the possibility that an subcontractor may want to lengthen the duration of his task so as to release more resource to achieve better resource sharing. But the model above has not considered this option.

Appendix B contains an example that is used repeatedly throughout the paper.

3.2.2. The problem for the project owner

The problem for the project owner is that how to design the contracts with subcontractors so that his total cost can be minimized. The project owner faces the following costs: (i) the completion cost $\varphi_i t_i$, (ii) the normal payment γ_i to an subcontractor if the task is successfully completed no later than t_i^0, (iii) the unit bonus β_i if task i is expedited (i.e., $t_i < t_i^0$), and (iv) the share of the unit crashing cost, ϑ_i, that the project owner promises to undertake for crashing task i. The first two types of costs are unavoidable and we assume that φ_i is a fixed constant, and γ_i should also be a fixed constant (which is the usual price to perform such a task in the marketplace). The last two types of costs are "optional" because β_i and ϑ_i are decision variables for the project owner when he designs his contracts with the subcontractors. If project owner sets $\beta_i = \vartheta_i = 0\ \forall i \in \mathcal{N}$, then all the tasks will be completed normally without any crashing. In such situation, project owner just undertakes the first two types of costs. However, in order to minimize the total cost, project owner should determine the values of β_i, ϑ_i, as well as $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{t}}_i$ more carefully. Generally project owner needs the following steps to minimize his total cost:

1. Collect parameter information of crashing and resource from the subcontractors, e.g., r_{ij}, $\overline{r}_{ij}$ and Θ_i. project owner can directly ask the subcontractors for these information if subcontractors are willing to provide. However, he can also estimate these parameters according to his prior experience, if subcontractors don't sharing information or if the information the subcontractors provide can not convince the project owner.

2. With available information, project owner solve the following optimization problem which show the minimum cost project owner can possibly obtained.

$$\underline{C} = \min_{\{\mathbf{t}_\mathcal{N}, \mathbf{x}_\mathcal{N}\}} \varphi_\mathcal{N}^\top \mathbf{t}_\mathcal{N} + \mathbf{\Theta}_\mathcal{N}^\top \mathbf{x}_\mathcal{N} \tag{3.4}$$

s.t.

$$(3.2.1) - (3.2.3), (3.2.6)$$

where $\varphi_{\mathcal{N}} = (\varphi_1, \varphi_2, \cdots, \varphi_n)^\top$ and $\mathbf{\Theta}_{\mathcal{N}} = (\Theta_1, \Theta_2, \cdots, \Theta_n)^\top$. Suppose the optimal solution for the above optimization problem is $(\mathbf{t}^*_{\mathcal{N}}, \mathbf{x}^*_{\mathcal{N}})$, then we can set $\overline{\mathbf{t}}_{\mathcal{N}} = \max(\mathbf{t}^0_{\mathcal{N}}, \mathbf{t}^*_{\mathcal{N}})$ and $\overline{\mathbf{x}}_{\mathcal{N}} = \mathbf{x}^*_{\mathcal{N}}$.

3. Determine β_i, ϑ_i carefully (optimally) so as to achieve the effect that the subcontractors are motivated appropriately (they affect x_i and t_i). Define $\vartheta_{\mathcal{N}} = (\vartheta_1, \vartheta_2, \cdots, \vartheta_n)^\top$. Note that for a given group of $(\beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}, \overline{\mathbf{t}}_{\mathcal{N}}, \overline{\mathbf{x}}_{\mathcal{N}})$, subcontractors will solve problem P1($\mathcal{N}$) accordingly. Assume the optimal solution set is $\mathcal{F}^*(\mathcal{N}, \beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}, \overline{\mathbf{t}}_{\mathcal{N}}, \overline{\mathbf{x}}_{\mathcal{N}})$. The problem for the project owner is therefore to determine the optimal values of $\{\beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}\}$ so that his total cost is minimized:

$$PC = \min_{\{\beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}\}} \{\varphi_{\mathcal{N}}^\top \mathbf{t}_{\mathcal{N}} + \beta_{\mathcal{N}}^\top (\mathbf{t}^0_{\mathcal{N}} - \mathbf{t}_{\mathcal{N}}) + \vartheta_{\mathcal{N}}^\top \mathbf{x}_{\mathcal{N}} + \mathbf{1}^\top \gamma_{\mathcal{N}}\} \tag{3.5}$$

s.t.

$$\{\mathbf{x}_{\mathcal{N}}, \mathbf{t}_{\mathcal{N}}\} \in \mathcal{F}^*(\mathcal{N}, \beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}, \overline{\mathbf{t}}_{\mathcal{N}}, \overline{\mathbf{x}}_{\mathcal{N}})$$

Denote this problem as P2.

Since $\mathbf{x}_{\mathcal{N}}$ and $\mathbf{t}_{\mathcal{N}}$ are implicit functions of $\{\beta_{\mathcal{N}}, \vartheta_{\mathcal{N}}\}$, problem P2 is not a linear optimization problem. Clearly, $PC \geq \underline{C} + \mathbf{1}^\top \gamma_{\mathcal{N}}$. The project owner faces the problem of finding $\beta^*_{\mathcal{N}}$, $\vartheta^*_{\mathcal{N}}$, such that $PC = \underline{C} + \mathbf{1}^\top \gamma_{\mathcal{N}}$. Whether the subcontractors are cooperative or not affects the project owner's decisions. Also, the quality of the project owner's decisions depends in part on the accuracy of information known about the subcontractors. We study these issues in Section 4.

3.3. Subcontractors' Cooperative Game

In this section, we study the issue of cooperation among the subcontractors. Since the project owner needs to design an optimal contract with the subcontractors, an understanding of this issue is also essential to the project owner.

For any coalition $\mathcal{S} \subseteq \mathcal{N}$, its characteristic function $v(\mathcal{S})$ is defined as the maximum total gain of the problem P1($\mathcal{S}$). That is, $v(\mathcal{S}) = \pi(\mathcal{S})$, where $\pi(\mathcal{S})$ is the optimal solution of (2). A cooperative game $(\mathcal{N}, v)$ is thus defined.

3.3.1. Nonconvexity

We first study the convexity of the game. Convexity is equivalent to supermodularity of the characteristic function. For any coalition $\mathcal{S}, \mathcal{T} \subseteq \mathcal{N}$, the supermodularity of the characteristic function is defined as

$$v(\mathcal{S} \cup \mathcal{T}) + v(\mathcal{S} \cap \mathcal{T}) \geq v(\mathcal{S}) + v(\mathcal{T}) \tag{3.6}$$

. The following example shows that equation (3.6) does not hold, and hence the cooperative game is not convex.

Example 3.1. Consider the project shown in Figure 3.1, where $\mathcal{N} = \{1, 2, 3, 4, 5\}$. Tasks 2,3 and 4 are the immediate successors of task 1, and immediate predecessors of task 5. The processing times of the tasks are 2. Therefore, the normal completion times of the tasks are $\mathbf{t}^0_{\mathcal{N}} = (2, 4, 4, 4, 6)$, respectively. Suppose that there is no crashing cost for any task. The maximum crashing time allowed for each task is $\overline{x}_i = 1$, and no task lengthening is allowed. Assume there is only one type of resource and let the resource consumption rate of each subcontractor be 1, i.e., $r_{i1} = 1, i = 1, \ldots, 5$. Also, assume that the maximum resources which the subcontractors can use are $\overline{r}_{11} = \overline{r}_{51} = 2, \overline{r}_{21} = \overline{r}_{31} = 0.5, \overline{r}_{41} = 1.5$, respectively. Finally, let $\beta_i = 1$ and $\gamma_i = \gamma\, i = 1, \ldots, 5$. Consider coalitions $\mathcal{S} = \{2, 4\}$ and $\mathcal{T} = \{3, 4\}$. Since both of them have enough resources for optimal crashing, we have

$$v(\mathcal{S}) = v(\mathcal{T}) = 2[\gamma + 1 \cdot (4 - 3)] = 2\gamma + 2$$

And task 4 also has enough resource to do crashing by himself, thus

$$v(\mathcal{S} \cap \mathcal{T}) = v(\{4\}) = \gamma + 1 \cdot (4 - 3) = \gamma + 1$$

But for the coalition $\mathcal{S} \cup \mathcal{T}$, since they only share 2.5 units of resource, the value

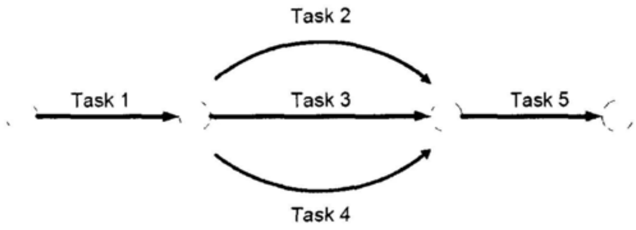


Figure 3.1: Simple Example

function is

$$v(\mathcal{S} \cup \mathcal{T}) = v(\{2,3,4\}) = 2[\gamma + 1 \cdot (4-3)] + \gamma + 1 \cdot (4-3.5) = 3\gamma + 2.5.$$

Hence,

$$v(\mathcal{S} \cup \mathcal{T}) + v(\mathcal{S} \cap \mathcal{T}) = 4\gamma + 3.5 < 4\gamma + 4 = v(\mathcal{S}) + v(\mathcal{T})$$

which contradicts supermodularity (3.6). This shows that the game is not convex.

3.3.2. Balancedness and The Core

Since the problem we formulate is an LP model, it is natural to see if the results of Owen (1975) can be generalized to study balancedness. Our subcontractors' cooperative game is more general than Owen's problem in two ways:

- In Owen's problem, every agent can invest in all products. Therefore, for any coalition $S \subseteq N$, the resource consumption rate matrix **A** is the same. Only the right hand side vectors b_S, which represent the total resource owned by coalition S, differs when S changes. But in our problem, both the left side coefficient matrix and the right hand side resource vector change with the coalition $\mathcal{S}$;

- The available "resources" and "resource consumption rates" are sometimes negative in our problem. Therefore when adding in a new player, the "resources" may decrease.

Let the dual variables corresponding to constraints (3.2.1) be denoted $\rho_{\mathcal{N}} = (\cdots, \rho_{ij}, \cdots)^\top, i \in \mathcal{N}, j \in \mathcal{P}^i$. Also, let the dual variables corresponding to constraints (3.2.2) be denoted $\tau_{\mathcal{N}} = (\tau_1, \tau_2, \tau_3)$ and (3.2.3) as $\omega = (\omega_1, \cdots, \omega_m)$. Finally, let the dual variables corresponding to constraints (3.2.4)-(3.2.5)be denoted $\mu_{\mathcal{N}}^\top$ and $\zeta_{\mathcal{N}}^\top$ respectively. (For illustration of these dual variables, refer to constraints (3.36.1)-(3.36.5) of Example 3.2.) Denote $\mathbf{z}_{\mathcal{N}}^\top = (\rho_{\mathcal{N}}^\top, \tau_{\mathcal{N}}^\top, \omega^\top, \mu_{\mathcal{N}}^\top, \zeta_{\mathcal{N}}^\top)$. With these definitions, it is not difficult to verify that the dual of problem P1($\mathcal{N}$) can be written as:

$$\min_{\mathbf{z}_{\mathcal{N}}} \quad (\mathbf{b}_{\mathcal{N}}^1)^\top \rho_{\mathcal{N}} + (\mathbf{b}_{\mathcal{N}}^2)^\top \tau_{\mathcal{N}} + \bar{\mathbf{r}}_{\mathcal{N}}^\top \omega + \bar{\mathbf{x}}_{\mathcal{N}}^\top \mu_{\mathcal{N}} + \bar{\mathbf{t}}_{\mathcal{N}}^\top \zeta_{\mathcal{N}} + \beta_{\mathcal{N}}^\top \mathbf{t}_{\mathcal{N}}^0 + \mathbf{1}^\top \gamma_{\mathcal{N}} \tag{3.7}$$

s.t.

$$\begin{pmatrix} (\mathbf{A}_{\mathcal{N}}^1)^\top & (\mathbf{B}_{\mathcal{N}}^1)^\top & \mathbf{R}_{\mathcal{N}}^\top & \mathbf{I} & \mathbf{0} \\ (\mathbf{A}_{\mathcal{N}}^2)^\top & (\mathbf{B}_{\mathcal{N}}^2)^\top & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \rho_{\mathcal{N}} \\ \tau_{\mathcal{N}} \\ \omega \\ \mu_{\mathcal{N}} \\ \zeta_{\mathcal{N}} \end{pmatrix} \geq \begin{pmatrix} -\theta_{\mathcal{N}} \\ -\beta_{\mathcal{N}} \end{pmatrix} \tag{3.8}$$

$$\mathbf{z}_{\mathcal{N}} \geq \mathbf{0} \tag{3.9}$$

We denote this dual problem by D1($\mathcal{N}$) and let $\mathbf{z}_{\mathcal{N}}^*$ denote its optimal solution.

Similarly we formulate the dual problem, D1($\mathcal{S}$), of P1($\mathcal{S}$) as

$$\min_{\mathbf{z}_{\mathcal{S}}} \quad (\mathbf{b}_{\mathcal{S}}^1)^\top \rho_{\mathcal{S}} + (\mathbf{b}_{\mathcal{S}}^2)^\top \tau_{\mathcal{S}} + \bar{\mathbf{r}}_{\mathcal{S}}^\top \omega + \bar{\mathbf{x}}_{\mathcal{S}}^\top \mu_{\mathcal{S}} + \bar{\mathbf{t}}_{\mathcal{S}}^\top \zeta_{\mathcal{S}} + \beta_{\mathcal{S}}^\top \mathbf{t}_{\mathcal{S}}^0 + \mathbf{1}^\top \gamma_{\mathcal{S}} \tag{3.10}$$

s.t.

$$\begin{pmatrix} (\mathbf{A}_{\mathcal{S}}^1)^\top & (\mathbf{B}_{\mathcal{S}}^1)^\top & \mathbf{R}_{\mathcal{S}}^\top & \mathbf{I} & \mathbf{0} \\ (\mathbf{A}_{\mathcal{S}}^2)^\top & (\mathbf{B}_{\mathcal{S}}^2)^\top & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \rho_{\mathcal{S}} \\ \tau_{\mathcal{S}} \\ \omega \\ \mu_{\mathcal{S}} \\ \zeta_{\mathcal{S}} \end{pmatrix} \geq \begin{pmatrix} -\theta_{\mathcal{S}} \\ -\beta_{\mathcal{S}} \end{pmatrix} \tag{3.11}$$

$$\mathbf{z}_{\mathcal{S}} \geq \mathbf{0} \tag{3.12}$$

Observe that $\mathbf{z}_{\mathcal{N}}$ and $\mathbf{z}_{\mathcal{S}}$ have different dimensions. For notational convenience, we expand $\mathbf{z}_{\mathcal{S}}$ to the same dimension as $\mathbf{z}_{\mathcal{N}}$ by adding in dummy variables.

We denote the expanded vector by $\bar{\mathbf{z}}_{\mathcal{S}}$. Furthermore, in order to make sure the above notation of problem D1($\mathcal{S}$) is still feasible, we also add some 0 vectors into equation (3.10) and (3.11) accordingly. If we denote the coefficient vector in objective function of D1($\mathcal{S}$) as $\mathbf{b}_{\mathcal{S}}$, then we have

$$\mathbf{b}_{\mathcal{S}}^{\top} = ((\mathbf{b}_{\mathcal{S}}^{1})^{\top}, (\mathbf{b}_{\mathcal{S}}^{2})^{\top}, \bar{\mathbf{r}}_{\mathcal{S}}^{\top}, \bar{\mathbf{x}}_{\mathcal{S}}^{\top}, \bar{\mathbf{t}}_{\mathcal{S}}^{\top}). \tag{3.13}$$

We rewrite problem D1($\mathcal{S}$) using the following steps:

1. Expand $\mathbf{b}_{\mathcal{S}}$ to $\bar{\mathbf{b}}_{\mathcal{S}}$ by adding in 0, according to the dummy variables which are added into $\mathbf{z}_{\mathcal{S}}$. Make sure $\bar{\mathbf{b}}_{\mathcal{S}}^{\top}\bar{\mathbf{z}}_{\mathcal{S}} = \mathbf{b}_{\mathcal{S}}^{\top}\mathbf{z}_{\mathcal{S}}$.

2. Adding 0 vectors into the left hand side coefficient matrix of equation (3.11) according to the dummy variables which are added into $\mathbf{z}_{\mathcal{S}}$. Make sure that the product of the revised coefficient matrix and $\bar{\mathbf{z}}_{\mathcal{S}}$ still equals to the right hand side vector.

Henceforth, we use the notation of D1($\mathcal{S}$), as in (3.10)-(3.12). But note that the dimension of the variables have already been expanded appropriately. And the $\mathbf{b}_{\mathcal{S}}$ defined in equation (3.13) is also augmented to the same dimension as $\mathbf{b}_{\mathcal{N}}$. The example in Appendix C illustrates these changes.

In order to study the balancedness of the subcontractors' game, we first establish the following lemma about the relationship between D1($\mathcal{N}$) and D1($\mathcal{S}$).

Lemma 3.1. *The optimal solution $\mathbf{z}_{\mathcal{N}}^{*}$ of D1($\mathcal{N}$) is a feasible solution of D1($\mathcal{S}$).*

Proof. First, consider the matrix $\left((\mathbf{A}_{\mathcal{N}}^{1})^{\top}|(\mathbf{B}_{\mathcal{N}}^{1})^{\top}\right)$ in the dual problem D1($\mathcal{N}$). Row i corresponds to a primal variable x_i in the primal problem P1($\mathcal{N}$). Thus, if task i does not belong to the coalition $\mathcal{S}$, then $\left((\mathbf{A}_{\mathcal{S}}^{1})^{\top}|(\mathbf{B}_{\mathcal{S}}^{1})^{\top}\right)$ does not contain row i of $\left((\mathbf{A}_{\mathcal{N}}^{1})^{\top}|(\mathbf{B}_{\mathcal{N}}^{1})^{\top}\right)$. Similarly, row i of the matrix $\left((\mathbf{A}_{\mathcal{N}}^{2})^{\top}|(\mathbf{B}_{\mathcal{N}}^{2})^{\top}\right)$ corresponds to the primal variable t_i. If task i does not belong to the coalition $\mathcal{S}$, then $\left((\mathbf{A}_{\mathcal{S}}^{2})^{\top}|(\mathbf{B}_{\mathcal{S}}^{2})^{\top}\right)$ does not contain row i of $\left((\mathbf{A}_{\mathcal{N}}^{2})^{\top}|(\mathbf{B}_{\mathcal{N}}^{2})^{\top}\right)$. A similar relationship exists between $\mathbf{r}_{\mathcal{N}}$ and $\mathbf{r}_{\mathcal{S}}$.

Second, any nonzero column of the matrices $\left((\mathbf{A}_{\mathcal{S}}^{k})^{\top}|(\mathbf{B}_{\mathcal{S}}^{k})^{\top}\right)$, $k = 1, 2$ must correspond to a constraint in the primal problem P1($\mathcal{N}$) that includes at least one of the primal variables x_i and t_i of task $i \in \mathcal{S}$. Moreover, those columns in $\left((\mathbf{A}^k_{\mathcal{N}})^\top | (\mathbf{B}^k_{\mathcal{N}})^\top\right)$ that correspond to constraints involving no primal variables in $\mathcal{S}$ will appear in $\left((\mathbf{A}^k_{\mathcal{S}})^\top | (\mathbf{B}^k_{\mathcal{S}})^\top\right)$ as zero columns. A similar observation can be found for the identity matrices $\mathbf{I}_{\mathcal{N}}$ and $\mathbf{I}_{\mathcal{S}}$

Let $\mathbf{E}_{\mathcal{N}}$ and $\mathbf{E}_{\mathcal{S}}$ denote the LHS matrices of constraints (3.8) and (3.11) of the dual problems D1($\mathcal{N}$) and D1($\mathcal{S}$), respectively. It follows from the observations above that, by

(i) deleting those rows i of $\mathbf{E}_{\mathcal{N}}$ that correspond to the tasks $i \notin \mathcal{S}$, and

(ii) changing those columns j of $\mathbf{E}_{\mathcal{N}}$ that correspond to the constraints which do not involve primal variables in the coalition $\mathcal{S}$ into zero columns,

then we obtain $\mathbf{E}_{\mathcal{S}}$. Furthermore, after deleting all the ith variables of $\theta^\top_{\mathcal{N}}$ and $\beta^\top_{\mathcal{N}}$, $i \notin \mathcal{S}$, we can get $\theta^\top_{\mathcal{S}}$ and $\beta^\top_{\mathcal{S}}$.

The analysis above implies that the constraints of D1($\mathcal{S}$) are a subset of the constraints of D1($\mathcal{N}$). Therefore, any solution $\mathbf{z}_{\mathcal{N}}$ satisfying the constraints of D1($\mathcal{N}$) also satisfies the constraints of D1($\mathcal{S}$). Hence the optimal solution $\mathbf{z}^*_{\mathcal{N}}$ of D1($\mathcal{N}$) also satisfies all the constraints of D1($\mathcal{S}$). □

An illustration is given by Example 3.2, where the matrix of D1($\mathcal{S}$) is made up of rows 2,3,4,7,8 and 9 of that of D1($\mathcal{N}$), and thus any solution $z_{\mathcal{N}}$ that satisfies the constraints of D1($\mathcal{N}$) also satisfies the constraints of D1($\mathcal{S}$).

From Lemma 3.1, we conclude that

$$\mathbf{b}^\top_{\mathcal{S}}\mathbf{z}^*_{\mathcal{S}} \leq \mathbf{b}^\top_{\mathcal{S}}\mathbf{z}^*_{\mathcal{N}}, \tag{3.14}$$

where $\mathbf{z}^*_{\mathcal{S}}$ is the optimal solution of D1($\mathcal{S}$).

Next, we construct an allocation vector λ based on $\mathbf{z}^*_{\mathcal{N}}$. By using the inequality above, we show that λ lies in the core of the game $(\mathcal{N}, v)$.

Note that $\mathbf{b}_{\mathcal{S}}$ can be regarded as the resource bundle of the coalition $\mathcal{S}$. Let $\mathbf{b}_{\{i\}}$ be the resources owned by subcontractor i as prescribed by the problem P1(i). Note that again we assume $\mathbf{b}_{\{i\}}$ has the same dimension as $\mathbf{b}_{\mathcal{S}}$ and $\mathbf{b}_{\mathcal{S}}$, after augmenting the vector $\mathbf{b}_{\{i\}}$ with elements zero. Then we can prove the following Lemma which states the relation between $\mathbf{b}_{\mathcal{S}}$ and $\mathbf{b}_{\{i\}}$, $i \in \mathcal{S}$.

Lemma 3.2. *For any coalition $\mathcal{S}$,*

$$\mathbf{b}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \mathbf{b}_{\{i\}}. \tag{3.15}$$

Proof. From (3.13) and the definition of $\mathbf{b}_{\{i\}}$, it is not difficult to see that, to prove (3.15), we need only to show that

$$\mathbf{b}^1_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \mathbf{b}^1_{\{i\}}; \quad \mathbf{b}^2_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \mathbf{b}^2_{\{i\}}, \tag{3.16}$$

because the equality in other elements between $\mathbf{b}_{\mathcal{S}}$ and $\sum_{i \in \mathcal{S}} \mathbf{b}_{\{i\}}$ is straightforward.

Let us first examine the kth component of $\mathbf{b}^1_{\mathcal{S}}$. Assume that it corresponds to the precedence relation between tasks i and j, $i \in \mathcal{S}$. Then we know

(1) If $j \in \mathcal{P}^i \cap \mathcal{S}$, then this component of $\mathbf{b}^1_{\mathcal{S}}$ should equal to $-d_i$. On the other hand,

$$\sum_{i \in \mathcal{S}} \mathbf{b}^1_{\{i\}} = \mathbf{b}^1_{\{i\}} + \mathbf{b}^1_{\{j\}} = (-d_i - t^0_j) + t^0_j = -d_i.$$

(2) If $j \in \mathcal{P}^i \setminus \mathcal{S}$, then this component of $\mathbf{b}^1_{\mathcal{S}}$ should equal to $-d_i - t^0_j$. And

$$\sum_{i \in \mathcal{S}} \mathbf{b}^1_{\{i\}} = \mathbf{b}^1_{\{i\}} = -d_i - t^0_j.$$

(3) If $j \in \mathcal{A}^i \cap \mathcal{S}$, then this component of $\mathbf{b}^1_{\mathcal{S}}$ should equal to $-d_j$. And

$$\sum_{i \in \mathcal{S}} \mathbf{b}^1_{\{i\}} = \mathbf{b}^1_{\{i\}} + \hat{\mathbf{b}}^1_{\{j\}} = t^0_i + (-d_j - t^0_i) = -d_j.$$

(4) If $j \in \mathcal{A}^i \setminus \mathcal{S}$, then this component of $\mathbf{b}^1_{\mathcal{S}}$ should equal to t^0_i. And

$$\sum_{i \in \mathcal{S}} \mathbf{b}^1_{\{i\}} = \mathbf{b}^1_{\{i\}} = t^0_i.$$

If follows from cases (1)-(4) that the first equality of (3.16) holds.

Since $\mathbf{b}^2_{\mathcal{N}} = (d_1, -d_1, D)^\top$, we can easily verify the second equality of (3.16) according to whether task 1 or task n belongs to the coalition $\mathcal{S}$.

□

We can now establish the following theorem which indicates that the game $(\mathcal{N}, v)$ is balanced and provides a core allocation vector.

Theorem 3.3. *Let $\lambda = (\lambda_1, \dots, \lambda_n)$, where*

$$\lambda_i = \mathbf{b}_{\{i\}}^\top \mathbf{z}_{\mathcal{N}}^* + \beta_i t_i^0 + \gamma_i. \tag{3.17}$$

Then λ constitutes a core allocation of the game $(\mathcal{N}, v)$.

Proof. Let $\mathbf{t}_{\mathcal{N}}^*$ and $\mathbf{x}_{\mathcal{N}}^*$ be the optimal solution of P1($\mathcal{N}$) and $\pi^*(\mathcal{N})$ be the corresponding optimal objective value. For the grand coalition, we have

$$\begin{aligned}
\sum_{i=1}^{n} \lambda_i &= \sum_{i \in \mathcal{N}} [\mathbf{b}_{\{i\}}^\top \mathbf{z}_{\mathcal{N}}^* + \beta_i t_i^0 + \gamma_i] \\
&= \mathbf{b}_{\mathcal{N}}^\top \mathbf{z}_{\mathcal{N}}^* + \beta_{\mathcal{N}}^\top \mathbf{t}_{\mathcal{N}}^0 + \mathbf{1}^\top \gamma_{\mathcal{N}} \\
&= \beta_{\mathcal{N}}^\top (\mathbf{t}_{\mathcal{N}}^0 - \mathbf{t}_{\mathcal{N}}^*) - \theta_{\mathcal{N}}^T \mathbf{x}_{\mathcal{N}}^* + \mathbf{1}^\top \gamma_{\mathcal{N}} \\
&= \pi^*(\mathcal{N}),
\end{aligned}$$

where the second equality follows from (3.15) and the third equality follows from strong duality. Similarly, let $\mathbf{t}_{\mathcal{S}}^*$ and $\mathbf{x}_{\mathcal{S}}^*$ be the optimal solution of P1($\mathcal{S}$) and $\pi^*(\mathcal{S})$ be the corresponding optimal objective value. For any coalition $\mathcal{S} \subseteq \mathcal{N}$, we have

$$\begin{aligned}
\sum_{i \in \mathcal{S}} \lambda_i &= \sum_{i \in \mathcal{S}} [\mathbf{b}_{\{i\}}^\top \mathbf{z}_{\mathcal{N}}^* + \beta_i t_i^0 + \gamma_i] \\
&= \mathbf{b}_{\mathcal{S}}^\top \mathbf{z}_{\mathcal{N}}^* + \beta_{\mathcal{S}}^\top \mathbf{t}_{\mathcal{S}}^0 + \mathbf{1}^\top \gamma_{\mathcal{S}} \\
&\geq \mathbf{b}_{\mathcal{S}}^\top \mathbf{z}_{\mathcal{S}}^* + \beta_{\mathcal{S}}^\top \mathbf{t}_{\mathcal{S}}^0 + \mathbf{1}^\top \gamma_{\mathcal{S}} \\
&= \beta_{\mathcal{S}}^\top (\mathbf{t}_{\mathcal{S}}^0 - \mathbf{t}_{\mathcal{S}}^*) - \theta_{\mathcal{S}}^T \mathbf{x}_{\mathcal{S}}^* + \mathbf{1}^\top \gamma_{\mathcal{S}} \\
&= \pi^*(\mathcal{S}),
\end{aligned}$$

where the inequality follows from (3.14) and the third equality follows from strong duality. To summarize, we know that $\lambda = (\lambda_1, \dots, \lambda_n)$ lies in the core of the game $(\mathcal{N}, v)$. □

Theorem 3.3 implies that the game is balanced, and the subcontractors should always cooperate.

Remarks:

- The specific form of equation (3.17) is stated as follow

$$\lambda_i = \gamma_i + \beta_i t_i^0 + \sum_{j=1}^{m} \overline{r}_{ij}\omega_j^* + \overline{t}_i\zeta_i^* + \overline{x}_i\mu_i^* + \begin{cases} t_1^0 \sum_{k\in\mathcal{A}^1} \rho_{k1}^* - d_1(\tau_1^* - \tau_2^*) & i = 1 \\ t_i^0 \sum_{k\in\mathcal{A}^i} \rho_{ki}^* - (d_i + t_j^0) \sum_{j\in\mathcal{P}^i} \rho_{ij}^* & i = 2, ...n-1 \\ D\tau_3^* - (d_n + t_j^0) \sum_{j\in\mathcal{P}^n} \rho_{nj}^* & i = n \end{cases}$$

Appendix D provides an example of the payoff distribution.

3.3.3. Task Lengthening

We consider an extension of the model above, where tasks can be lengthened to release extra resources for use in crashing.

Define $\mathbf{y}_\mathcal{N} = (y_1, y_2, \cdots, y_n)^\top$ and $\overline{\mathbf{y}}_\mathcal{N} = (\overline{y}_1, \overline{y}_2, \cdots, \overline{y}_n)^\top$, where y_i denotes the number of time units that task i is lengthened and $\overline{y}_i$ denotes the maximum amount that task i can be lengthened. Problem P1($\mathcal{N}$) can be rewritten as

$$\pi(\mathcal{N}) = \max_{\{\mathbf{x}_\mathcal{N}, \mathbf{y}_\mathcal{N}, \mathbf{t}_\mathcal{N}\}} \beta_\mathcal{N}^\top(\mathbf{t}_\mathcal{N}^0 - \mathbf{t}_\mathcal{N}) - \theta_\mathcal{N}^\top\mathbf{x}_\mathcal{N} + \mathbf{1}^\top\gamma_\mathcal{N} \tag{3.18}$$

s.t.

$$\mathbf{A}_\mathcal{N}^1(\mathbf{x}_\mathcal{N} - \mathbf{y}_\mathcal{N}) + \mathbf{A}_\mathcal{N}^2\mathbf{t}_\mathcal{N} \leq \mathbf{b}_\mathcal{N}^1 \tag{3.18.1}$$

$$\mathbf{B}_\mathcal{N}^1(\mathbf{x}_\mathcal{N} - \mathbf{y}_\mathcal{N}) + \mathbf{B}_\mathcal{N}^2\mathbf{t}_\mathcal{N} \leq \mathbf{b}_\mathcal{N}^2 \tag{3.18.2}$$

$$\mathbf{R}_\mathcal{N}(\mathbf{x}_\mathcal{N} - \mathbf{y}_\mathcal{N}) \leq \overline{\mathbf{r}}_\mathcal{N} \tag{3.18.3}$$

$$\mathbf{I}\ \mathbf{x}_\mathcal{N} \leq \overline{\mathbf{x}}_\mathcal{N}\ ;\ \mathbf{I}\ \mathbf{y}_\mathcal{N} \leq \overline{\mathbf{y}}_\mathcal{N} \tag{3.18.4}$$

$$\mathbf{I}\ \mathbf{t}_\mathcal{N} \leq \overline{\mathbf{t}}_\mathcal{N} \tag{3.18.5}$$

$$\mathbf{x}_\mathcal{N} \geq \mathbf{0};\quad \mathbf{y}_\mathcal{N} \geq \mathbf{0};\quad \mathbf{t}_\mathcal{N} \geq \mathbf{0} \tag{3.18.6}$$

The dual problem D1($\mathcal{N}$) is

$$\min_{\hat{\mathbf{z}}_\mathcal{N}} \ (\mathbf{b}_\mathcal{N}^1)^\top\rho_\mathcal{N} + (\mathbf{b}_\mathcal{N}^2)^\top\tau_\mathcal{N} + \overline{\mathbf{r}}_\mathcal{N}^\top\omega + \overline{\mathbf{x}}_\mathcal{N}^\top\mu_\mathcal{N} + \overline{\mathbf{y}}_\mathcal{N}^\top\nu_\mathcal{N} + \overline{\mathbf{t}}_\mathcal{N}^\top\zeta_\mathcal{N} + \beta_\mathcal{N}^\top\mathbf{t}_\mathcal{N}^0 + \mathbf{1}^\top\gamma_\mathcal{N} \tag{3.19}$$

s.t.

$$\begin{pmatrix} (\mathbf{A}^1_{\mathcal{N}})^\top & (\mathbf{B}^1_{\mathcal{N}})^\top & \mathbf{R}^\top_{\mathcal{N}} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -(\mathbf{A}^1_{\mathcal{N}})^\top & -(\mathbf{B}^1_{\mathcal{N}})^\top & -\mathbf{R}^\top_{\mathcal{N}} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ (\mathbf{A}^2_{\mathcal{N}})^\top & (\mathbf{B}^2_{\mathcal{N}})^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \rho_{\mathcal{N}} \\ \tau_{\mathcal{N}} \\ \omega \\ \mu_{\mathcal{N}} \\ \nu_{\mathcal{N}} \\ \zeta_{\mathcal{N}} \end{pmatrix} \geq \begin{pmatrix} -\theta_{\mathcal{N}} \\ \mathbf{0} \\ -\beta_{\mathcal{N}} \end{pmatrix} \tag{3.20}$$

$$\hat{\mathbf{z}}_{\mathcal{N}} \geq \mathbf{0}, \tag{3.21}$$

where $\hat{\mathbf{z}}_{\mathcal{N}}$ is the vector $\mathbf{z}_{\mathcal{N}}$ augmented with the dual variables $\nu_{\mathcal{N}}$ corresponding to the constraints $\mathbf{y}_{\mathcal{N}} \leq \overline{\mathbf{y}}_{\mathcal{N}}$ in primal problem (3.19). We can verify that all the results obtained above on $\mathbf{z}_{\mathcal{N}}$ can be extended to $\hat{\mathbf{z}}_{\mathcal{N}}$ by appropriate modification with respect to $\nu_{\mathcal{N}}$. The allocation scheme in Theorem 3.3 is now adjusted to $\lambda = (\lambda_1, \ldots, \lambda_n)$, where

$$\lambda_i = \hat{\mathbf{b}}^\top_{\{i\}} \hat{\mathbf{z}}^*_{\mathcal{N}} + \beta_i t^0_i + \gamma_i, \tag{3.22}$$

where $\hat{\mathbf{b}}_{\{i\}}$ is the vector $\mathbf{b}_{\{i\}}$ augmented with $(0, \ldots, \overline{\mathbf{y}}^\top, \ldots, 0)^\top$.

Appendix E provides an example of task lengthening.

3.4. Contract Design

In this section, we investigate the optimal design of the project owner's contracts with the subcontractors' problem P2. Recall that t^0_i is the normal completion time of task i and $\varphi_i t^0_i$ is the corresponding completion time cost. If task i is completed earlier, at time $t_i < t^0_i$, then the saving in the completion cost is: $\varphi_i(t^0_i - t_i)$. The project owner may share this saving with the subcontractor by offering him a bonus $\alpha_i \varphi_i (t^0_i - t_i)$, then $\beta_i = \alpha_i \varphi_i$. Consequently, the saving the project owner retains is $(1 - \alpha_i)\varphi_i(t^0_i - t_i)$. The project owner may also share a portion of the crashing cost with the subcontractor. If the per unit crashing cost for task i is Θ_i, then the project owner shares $\varepsilon_i \Theta_i$. In this case, $\vartheta_i = \varepsilon_i \Theta_i$ and $\theta_i = (1 - \varepsilon)\Theta_i$. However, the project owner may not know the crashing cost information Θ_i. To resolve this problem, we assume project owner can follow two different schemes separately:

1. By request, the subcontractors provide their crashing cost information Θ_i $\forall i \in \mathcal{N}$. However, subcontractors may provide an exaggerated number $\bar{\Theta}_i$. Then, $\Theta_i \leq \bar{\Theta}_i$, $\forall i \in \mathcal{N}$, and the crashing cost of task i that project owner actually shares is $(1-\varepsilon_i)\bar{\Theta}_i$.

2. After subcontractors provide the information, the project owner further estimates the unit crashing cost for each task, based on experience. Suppose the estimated unit crashing cost of task i is $\tilde{\Theta}_i$, where $\tilde{\Theta}_i \leq \bar{\Theta}_i$. Then the project owner establishes a cost sharing mechanism basing on the estimated value $\tilde{\Theta}_i$. With this scheme, the project owner can avoid overcompensating subcontractors who request prohibitive crashing cost but may incur a loss due to poor estimation.

We study the project owner's outcomes under the second scheme in the next section, using sensitivity analysis. Here, we focus on the project owner's payoff under the first scheme. The project owner shares $\varepsilon_i\bar{\Theta}_i$ of the crashing cost of task i, and rewards subcontractor i with $\alpha_i\varphi_i$ for completing his task one time unit earlier. We also assume in this scheme that other information provided by the subcontractors, besides $\bar{\mathbf{\Theta}}_{\mathcal{N}}$, is true. We will investigate the impact of misleading information to the project owner in the next section. In this section we will show that if project owner sets the decision variables appropriately, then he will always obtain the maximum profit as he has expected.

According to the analysis in Section 2.2, we know after the project owner receive all the information from the subcontractors, he needs to solve optimization problem (3.4) first. Denote the optimal solution by $\mathbf{t}^*_{\mathcal{N}}(\bar{\mathbf{\Theta}}_{\mathcal{N}})$ and $\mathbf{x}^*_{\mathcal{N}}(\bar{\mathbf{\Theta}}_{\mathcal{N}})$. Then, $\bar{\mathbf{t}}_{\mathcal{N}}(\bar{\mathbf{\Theta}}_{\mathcal{N}})$ and $\bar{\mathbf{x}}_{\mathcal{N}}(\bar{\mathbf{\Theta}}_{\mathcal{N}})$ can be determined according to the optimal solution.

3.4.1. Uniform contracts

We first study the project owner's payoff for the special case of uniform contracts. That is, we let $\alpha_i = \alpha$ and $\varepsilon_i = \varepsilon$, hence the project owner offers uniform contracts to all the subcontractors. Furthermore, we let $\alpha = 1 - \varepsilon$, which implies that the proportion of earlier completion benefit and the proportion of crashing cost shared by the project owner are always the same. This specification is administratively simple, and avoids concerns about discriminatory pricing. The problem for the grand coalition of the subcontractors: P1($\mathcal{N}$), and the problem for the project owner: P2, can be stated as follows:

$$\pi(\mathcal{N}) = \max_{\{\mathbf{x}_\mathcal{N}, \mathbf{t}_\mathcal{N}\}} \quad \alpha\varphi_\mathcal{N}^\top \mathbf{t}_\mathcal{N}^0 + \mathbf{1}^\top \gamma_\mathcal{N} + [(1-\alpha)\bar{\mathbf{\Theta}}_\mathcal{N}^T - \mathbf{\Theta}_\mathcal{N}^T]\mathbf{x}_\mathcal{N} - \alpha\varphi_\mathcal{N}^\top \mathbf{t}_\mathcal{N} \quad (3.23)$$

s.t.

$$\mathbf{A}_\mathcal{N}^1 \mathbf{x}_\mathcal{N} + \mathbf{A}_\mathcal{N}^2 \mathbf{t}_\mathcal{N} \leq \mathbf{b}_\mathcal{N}^1 \quad (3.2.1)$$

$$\mathbf{B}_\mathcal{N}^1 \mathbf{x}_\mathcal{N} + \mathbf{B}_\mathcal{N}^2 \mathbf{t}_\mathcal{N} \leq \mathbf{b}_\mathcal{N}^2 \quad (3.2.2)$$

$$\mathbf{R}_\mathcal{N} \mathbf{x}_\mathcal{N} \leq \bar{\mathbf{r}}_\mathcal{N} \quad (3.2.3)$$

$$\mathbf{I}\ \mathbf{x}_\mathcal{N} \leq \bar{\mathbf{x}}_\mathcal{N}(\bar{\mathbf{\Theta}}_\mathcal{N}) \quad (3.2.4')$$

$$\mathbf{I}\ \mathbf{t}_\mathcal{N} \leq \bar{\mathbf{t}}_\mathcal{N}(\bar{\mathbf{\Theta}}_\mathcal{N}) \quad (3.2.5')$$

$$\mathbf{x}_\mathcal{N} \geq \mathbf{0}; \quad \mathbf{t}_\mathcal{N} \geq \mathbf{0} \quad (3.2.6)$$

and for P2 we have

$$PC = \min_\alpha \{\alpha\varphi_\mathcal{N}^\top \mathbf{t}_\mathcal{N}^0 + \mathbf{1}^\top \gamma_\mathcal{N} + \min_{\{\mathbf{x}_\mathcal{N}, \mathbf{t}_\mathcal{N}\}} \quad [(1-\alpha)\bar{\mathbf{\Theta}}_\mathcal{N}^\top \mathbf{x}_\mathcal{N} + (1-\alpha)\varphi_\mathcal{N}^\top \mathbf{t}_\mathcal{N}]\} \quad (3.24)$$

s.t.

$$(3.2.1) - (3.2.3), (3.2.4'), (3.2.5'), (3.2.6)$$

First, we show that the optimal solution of problem (3.23) is just $\mathbf{t}_\mathcal{N}^*(\bar{\mathbf{\Theta}}_\mathcal{N})$ and $\mathbf{x}_\mathcal{N}^*(\bar{\mathbf{\Theta}}_\mathcal{N})$.

Lemma 3.4. *The optimal solution of problem (3 4),* $\{\mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}),\ \mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$*, is also the optimal solution of problem (3 23)*

Proof Define $\pi'(\mathcal{N})$ as follow

$$\pi'(\mathcal{N}) = \max_{\{\mathbf{x}_{\mathcal{N}}\ \mathbf{t}_{\mathcal{N}}\}} \quad \alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}} + \mathbf{1}^{\top}\gamma_{\mathcal{N}} - \alpha\bar{\Theta}^T_{\mathcal{N}}\mathbf{x}_{\mathcal{N}} - \alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}_{\mathcal{N}} \tag{3 25}$$

s t

$$(3\ 2\ 1) - (3\ 2\ 3), (3\ 2\ 4'), (3\ 2\ 5'), (3\ 2\ 6)$$

From the definition of $\pi'(\mathcal{N})$ and $\underline{C}$, we can easily verify that with the same $\bar{\Theta}$, problems (3 25) and (3 4) have the same optimal solution $\{\mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}),\ \mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$ And

$$\pi'(\mathcal{N}) = \alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}} + \mathbf{1}^{\top}\gamma_{\mathcal{N}} - \alpha\underline{C}$$

Suppose that the optimal solution of problem (3 23) is $\{\mathbf{t}'_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}),\ \mathbf{x}'_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$, which is different from $\{\mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}),\ \mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$ Then we have

$$[(1-\alpha)\bar{\Theta}^T_{\mathcal{N}} - \Theta^T_{\mathcal{N}}](\mathbf{x}^*_{\mathcal{N}} - \mathbf{x}'_{\mathcal{N}}) - \alpha\varphi^{\top}_{\mathcal{N}}(\mathbf{t}^*_{\mathcal{N}} - \mathbf{t}'_{\mathcal{N}}) \le 0$$

Since $\mathbf{x}^*_{\mathcal{N}} = \bar{\mathbf{x}}_{\mathcal{N}}$, we know $\mathbf{x}'_{\mathcal{N}} \le \mathbf{x}^*_{\mathcal{N}}$ from constraints (3 2 4') Hence, we have

$$-\alpha\bar{\Theta}^T_{\mathcal{N}}(\mathbf{x}^*_{\mathcal{N}} - \mathbf{x}'_{\mathcal{N}}) \le [(1-\alpha)\bar{\Theta}^T_{\mathcal{N}} - \Theta^T_{\mathcal{N}}](\mathbf{x}^*_{\mathcal{N}} - \mathbf{x}'_{\mathcal{N}})$$

Therefore,

$$\begin{aligned} & \pi'(\mathcal{N}) - \{\alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}} + \mathbf{1}^{\top}\gamma_{\mathcal{N}} - \alpha\bar{\Theta}^T_{\mathcal{N}}\mathbf{x}'_{\mathcal{N}} - \alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}'_{\mathcal{N}}\} \\ = & -\alpha\bar{\Theta}^T_{\mathcal{N}}(\mathbf{x}^*_{\mathcal{N}} - \mathbf{x}'_{\mathcal{N}}) - \alpha\varphi^{\top}_{\mathcal{N}}(\mathbf{t}^*_{\mathcal{N}} - \mathbf{t}'_{\mathcal{N}}) \\ \le & [(1-\alpha)\bar{\Theta}^T_{\mathcal{N}} - \Theta^T_{\mathcal{N}}](\mathbf{x}^*_{\mathcal{N}} - \mathbf{x}'_{\mathcal{N}}) - \alpha\varphi^{\top}_{\mathcal{N}}(\mathbf{t}^*_{\mathcal{N}} - \mathbf{t}'_{\mathcal{N}}) \\ \le & \ 0 \end{aligned}$$

which contradicts the definition of $\pi'(\mathcal{N})$ Hence, the optimal solution of problem (3 23) is also $\{\mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}),\ \mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$ □

From Lemma 3 4 and the definition of problem (3 24), we can construct the project owner's optimal strategy under uniform contracts Note that in practise, there usually exists a lower bound for α, which prevents the upstream project owner from sharing too less profit with downstream subcontractors. More specifically, for each $\alpha_i, i \in \mathcal{N}$ there should be a lower bound $\underline{\alpha}_i \in (0,1)$ which depends on the market power of project owner against subcontractor i. Hence, with the uniform contract, we must set $\underline{\alpha} = \max_i\{\underline{\alpha}_i\} \in (0,1)$.

Theorem 3.5. *With uniform contracts, the project owner should set* $\overline{\mathbf{x}}_{\mathcal{N}} = \mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})$, $\bar{\mathbf{t}}_{\mathcal{N}} = \max\{\mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}), \mathbf{t}^0_{\mathcal{N}}\}$ *and* $\alpha = \underline{\alpha}$, *where* $\{\mathbf{x}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}}), \mathbf{t}^*_{\mathcal{N}}(\bar{\Theta}_{\mathcal{N}})\}$ *is the optimal solution of problem (3.4) by replacing* $\Theta_{\mathcal{N}}$ *with* $\bar{\Theta}_{\mathcal{N}}$.

Proof. With Lemma 3.4, we can rewrite problem (3.24) as

$$\begin{aligned} PC &= \min_{\alpha}\{\alpha\varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}} + \mathbf{1}^{\top}\gamma_{\mathcal{N}} + (1-\alpha)\underline{C}\} \\ &= \min_{\alpha}\{\alpha(\varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}} - \underline{C})\} + \mathbf{1}^{\top}\gamma_{\mathcal{N}} + \underline{C}) \end{aligned}$$

Note that $\underline{C} \leq \varphi^{\top}_{\mathcal{N}}\mathbf{t}^0_{\mathcal{N}}$. That is, the total task completion time cost should not be greater than that in the normal schedule. Hence PC is an increasing function of α, which implies that the project owner should set $\alpha = \underline{\alpha}$ to achieve a minimum cost. □

Theorem 3.5 implies that the project owner, deducted by normal payment, can always receive $1 - \underline{\alpha}$ proportion of the total system benefit. If project owner is a monopoly enterprise or has huge power in market, then he can enjoy a small $\underline{\alpha}$ which bring him most part of the benefit. This uniform contract appears quite fair since we set all the sharing rate the same to all the subcontractors. However, as we mentioned, subcontractors can complete their tasks earlier either by self-crashing or motivating his predecessors to crash. Hence, it seems to be more fair for the project owner to set different α_i and ε_i so that subcontractors will feel more conformable to accept the contracts, especially for those subcontractors who do plenty of crashings.

3.4.2. Nonuniform contracts

We now consider a situation where the project owner has the administrative and legal flexibility to use different values of α_i and ε_i for each subcontractor i. The project owner sets these incentives such that the subcontractors are motivated to crash their task optimally, and moreover, the project owner maximizes its profit.

As discussed above, task i can complete one time unit earlier in two ways. The first way is that task i is crashed. The second way is that the predecessors of task i crash their tasks, so that task i can start one time unit earlier. In both cases, subcontractor i receives an early completion incentive $\alpha_i\varphi_i$ per unit time. However, the subcontractor incurs a unit crashing cost $(1-\varepsilon_i)\Theta_i$ in the first case only. Therefore, the problem for the project owner is to set $(\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}})$ appropriately so that each unit of crashing is compensated, and that its profit is maximized. We propose Algorithm AE below to solve this problem.

Algorithm AE

Input: $\bar{\Theta}_{\mathcal{N}}$, $\bar{\mathbf{r}}_{\mathcal{N}}$, $\mathbf{d}_{\mathcal{N}}$ and the precedence network of the project.

Step 1: Solve the following linear program:

$$\underline{C}' = \min_{\{x_i,t_i\}_{i\in\mathcal{N}}} \sum_{i\in\mathcal{N}} [\varphi_i t_i + \bar{\Theta}_i x_i + \underline{\alpha}_i \varphi_i (t_i^0 - t_i)] \tag{3.26}$$

s.t.

$$(3.2.1) - (3.2.6)$$

Let the optimal solution to problem (3.26) be denoted $\mathbf{x}^* = (x_1^*, \cdots, x_n^*)$, $\mathbf{t}^* = (t_1^*, \cdots, t_n^*)$.

Set $x_i^L = x_i^U = 0, t_i^L = t_i^U = t_i^0, i = 1, \ldots, n$, $X_L = 0$, $X_U = \sum_{i\in N} x_i^*$, $q = 0$ and $C^L = \emptyset$.

Step 2: Set $X = \lceil (X_L + X_U)/2 \rceil$. Solve the following linear program:

$$\underline{C}' = \min_{\{x_i',t_i\}_{i\in\mathcal{N}}} \sum_{i\in\mathcal{N}} [\varphi_i t_i + \bar{\Theta}_i x_i' + \underline{\alpha}_i \varphi_i (t_i^0 - t_i)] \tag{3.27}$$

s.t.

$$\begin{cases} (3.2.1)-(3.2.3),(3.2.5),(3.2.6) \\ x'_i \le x^*_i,\ \forall i \in \mathcal{N} \\ \sum_{i\in\mathcal{N}} x'_i \le X \\ x'_i \ge x^L_i,\ \forall i \in C^L, \end{cases}$$

and let $C' = \{i | x'_i > x^L_i\}$.

If $X < X_U$, then go to Step 4. Otherwise, set $x^U_i = x'_i, t^U_i = t_i, i = 1, \dots, n$.

Step 3: Set $q = q+1$, $C(q) = \{i | x^U_i > x^L_i\}$ and $C^L = \{i | x^U_i > 0\}$. Let $A' = \{i | t^U_i = t^L_i\}$ and $A(q) = (\cup_{i\in C(q)} \mathcal{A}^i) \setminus A'$. Set $x^L_i = x^U_i, t^L_i = t^U_i, i = 1, \dots, n$ If $X_L = \sum_{i\in N} x^*_i$, then go to Step 5; otherwise, set $X_L = X_U$, $X_U = \sum_{i\in N} x^*_i$, and go to Step 2.

Step 4: If $C' = \emptyset$, then set $X_L = X$, and go to Step 2; otherwise, set $x^U_i = x'_i, t^U_i = t_i, i = 1, \dots, n$ and $X_U = X$, and go to Step 2.

Step 5: Solve the following linear program:

$$\max_{\{\alpha_\mathcal{N}, \varepsilon_\mathcal{N}\}} \sum_{i\in\mathcal{N}} (1-\alpha_i)\varphi_i(t^0_i - t^*_i) - \varepsilon_i \bar{\Theta}_i x^*_i \tag{3.28}$$

s.t.

$$\begin{cases} \sum_{i\in C(j)} (\alpha_i - \underline{\alpha}_i)\varphi_i + \sum_{i\in\mathcal{A}(j)} (\alpha_i - \underline{\alpha}_i)\varphi_i = \sum_{i\in C(j)} (1-\varepsilon_i)\bar{\Theta}_i & \forall j = 1, \dots, q \\ (\alpha_i - \underline{\alpha}_i)\varphi_i \le (1-\varepsilon_i)\bar{\Theta}_i & \forall i \in \mathcal{N} \\ \underline{\alpha}_i \le \alpha_i < 1,\ 0 < \varepsilon_i < 1 & \forall i \in \mathcal{N}, \end{cases}$$

where $\underline{\alpha}_i \in (0,1)$ is defined in the uniform contract case above.

Output: $\alpha_i, \varepsilon_i, \forall i \in \mathcal{N}$ and $C(1), \dots, C(q)$.

We prove below that the α_N, ϵ_N found by Algorithm AE represent the optimal contract design for the project owner. The proof requires the following preliminary result.

Lemma 3.6. *LP problem (3.28) has a feasible solution.*

Proof. By construction, there are at most n of the first constraints, and they are linearly independent equations of the variables $\alpha_i, \epsilon_i, i = 1, \dots, n$. Similarly, the upper and lower bounds on α_i and ϵ_i introduce $4n$ linearly independent equations in $4n$ slack variables. □

We now provide the main result of this section.

Theorem 3.7. *Given the crashing cost information $\bar{\Theta}_{\mathcal{N}}$ provided by the subcontractors and all other necessary information, the project owner specifies the contract incentives $(\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}})$ from Algorithm AE. Then doing so minimizes the project owner's total cost $\underline{C}' + \mathbf{1}^\top \gamma_{\mathcal{N}}$. Moreover, the time complexity of Algorithm AE is $O(n \log(\sum_i x_i^*) LP(2n))$.*

Proof. From Lemma 3.6, we know Algorithm AE provides a feasible solution $\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}}$. It remains to show that (a) the subcontractors perform an optimal crashing schedule $(\mathbf{x}^*_{\mathcal{N}}, \mathbf{t}^*_{\mathcal{N}})$, and (b) the project owner's total cost is $\underline{C}' + \mathbf{1}^\top \gamma_{\mathcal{N}}$.

First, due to the structure of LP we observe that if there exists a solution to (3.27) with $\sum_{i \in} x'_i \leq X$ where $x'_i = a_i$, then there exists a solution to (3.27) with $\sum_{i \in} x'_i \leq X'$ where $x'_i \geq a_i$, if $X' > X$. This justifies the last constraint in (3 27).

Second, we check that each subcontractor receives sufficient compensation for crashing. If we let $C(j)$ denote the crashing set in step j, then this requires

$$\sum_{i \in C(j)} \alpha_i \varphi_i + \sum_{i \in A(j)} \alpha_i \varphi_i \geq \sum_{i \in C(j)} (1 - \varepsilon_i) \bar{\Theta}_i, \qquad j = 1, \ldots, q$$

where $A(j)$ is are the successors of tasks in set S that have been brought forward by one time unit in step j. The optimal solution, $\alpha_{\mathcal{N}}$ and $\varepsilon_{\mathcal{N}}$, of problem (3.28) satisfies this inequality since

$$\begin{aligned} & \sum_{i \in C(j)} \alpha_i \varphi_i + \sum_{i \in A(j)} \alpha_i \varphi_i \\ = & \sum_{i \in C(j)} (1 - \varepsilon_i) \bar{\Theta}_i + \sum_{i \in C(j)} \underline{\alpha}_i \varphi_i + \sum_{i \in A(j)} \underline{\alpha}_i \varphi_i \\ \geq & \sum_{i \in C(j)} (1 - \varepsilon_i) \bar{\Theta}_i, \end{aligned}$$

where the last inequality holds because $\underline{\alpha}_i \geq 0$. Therefore, $\alpha_{\mathcal{N}}$ and $\varepsilon_{\mathcal{N}}$ from Algorithm AE provide sufficient compensation to ensure at least optimal crashing by the subcontractors. Moreover, by construction, no subcontractor can crash more than the optimal amount.

Next, we consider the optimality of α_N, ε_N. The project owner's benefit from the crashing of tasks in set $C(j)$ is

$$
\begin{aligned}
& \sum_{i \in C(j)} (1-\alpha_i)\varphi_i + \sum_{i \in A(j)} (1-\alpha_i)\varphi_i - \varepsilon_i \bar{\Theta}_i \\
= & \sum_{i \in C(j)} (1-\alpha_i)\varphi_i + \sum_{i \in A(j)} (1-\alpha_i)\varphi_i + (1-\varepsilon_i)\bar{\Theta}_i - \bar{\Theta}_i \\
= & \sum_{i \in C(j)} (1-\underline{\alpha}_i)\varphi_i + \sum_{i \in A(j)} (1-\underline{\alpha}_i)\varphi_i - \bar{\Theta}_i
\end{aligned}
$$

where the second equality follows from the first group of constraints in problem (3.28). The total benefit obtained by the project owner is the sum of the benefit from crashing each task, or $\sum_{i \in \mathcal{N}} [(1-\underline{\alpha}_i)\varphi_i(t_i^0 - t_i^*) - \bar{\Theta}_i x_i^*]$. Therefore, the project owner's total cost is

$$
\begin{aligned}
& \sum_{i \in \mathcal{N}} [\varphi_i t_i^0 + \gamma_i] - \sum_{i \in \mathcal{N}} [(1-\underline{\alpha}_i)\varphi_i(t_i^0 - t_i^*) - \bar{\Theta}_i x_i^*] \\
= & \sum_{i \in \mathcal{N}} [\varphi_i t_i^* + \gamma_i + \bar{\Theta}_i x_i^* + \underline{\alpha}_i \varphi_i (t_i^0 - t_i^*)] \\
= & \underline{C}' + \mathbf{1}^\top \gamma_{\mathcal{N}}
\end{aligned}
$$

The second equality holds since $(x_i^*, t_i^*), \forall i \in \mathcal{N}$ is the optimal solution of LP problem (3.26). Therefore, the project owner's total cost is minimized.

Finally, we consider the time complexity of Algorithm AE. In Step 1, we solve a linear program with time complexity $O(LP(2n))$. The binary search procedure in Steps 2 through 4 identifies $q \leq n$ nested subsets $C(q)$ of tasks that are being crashed. The identification of each subset requires searching $O(\sum_{i \in N} x_i^*)$ values of X which requires $O(\log(\sum_{i \in N} x_i^*))$ binary search iterations; each iteration requires the solution of a linear program with time complexity $O(LP(2n))$. Step 5 also requires the solution of a linear program with time complexity $O(LP(2n))$. Therefore, the overall time complexity of Algorithm AE is $O(n \log(\sum_i x_i^*) LP(2n))$. □

We observe that, due to the lower bound constraints $\alpha_i, \forall i \in \mathcal{N}$, only a portion of the bonus paid by the project owner is used to cover the necessary crashing cost. The extra profit obtained by the subcontractors is $\sum_{i \in \mathcal{N}} \underline{\alpha}_i \varphi_i (t_i^0 - t_i^*)$. Appendix F provides an example of Algorithm AE. This analysis of this section assumes that the project owner has full and precise information from the subcontractors. The following section relaxes this assumption.

3.5. Imprecise Information

In this section, we study the problem where the project owner does not have the precise information that is needed for optimal contract design. This may lead the project owner to a contract design that fails to minimize his cost. We now focus on the outcome to the project owner when various types of information are unavailable. In such situation, the project owner needs to estimate various parameters related to crashing. If the project owner makes any inaccurate estimation, then the project may not be implemented by the subcontractors in the way that the project owner expects. We show that the project owner can either benefit or lose as a result of different kind of estimations. We will also show that the subcontractors may lose by providing imprecise information, for example by exaggerating their crashing costs.

3.5.1. Unit crashing cost

If the project owner asks a subcontractor for unit crashing cost information, the subcontractor may either refuse to tell the project owner or exaggerate the cost. If the subcontractors do not share unit crashing cost information, project owner has to estimate parameter $\Theta_{\mathcal{N}}$. If the subcontractors ask for a high price $\bar{\Theta}_{\mathcal{N}}$ that exceeds the project owner's valuation based on his prior experience, the project owner may estimate the parameters by himself instead of using $\bar{\Theta}_{\mathcal{N}}$ directly to design his contracts. We assume that subcontractors are always willing to offer a unit crashing cost parameter $\bar{\Theta}_{\mathcal{N}}$, else we just regard the ith component of $\bar{\Theta}_{\mathcal{N}}$ as ∞. Suppose the project owner's estimation is $\tilde{\Theta}_{\mathcal{N}} \leq \bar{\Theta}_{\mathcal{N}}$. The theorem below shows that if the project owner overestimates the unit crash cost, then the closer

$\tilde{\Theta}_{\mathcal{N}}$ and $\Theta_{\mathcal{N}}$ are, the lower the cost that the project owner can achieve.

Theorem 3.8. *Given all information except the unit crashing cost $\Theta_{\mathcal{N}}$, suppose the project owner has two different estimations $\Theta^1_{\mathcal{N}}$ and $\Theta^2_{\mathcal{N}}$ on the crashing cost parameters which satisfy $\Theta^1_{\mathcal{N}} \geq \Theta^2_{\mathcal{N}} \geq \Theta_{\mathcal{N}}$. If we denote $\underline{C}'(\Theta_{\mathcal{N}})$ as the optimal value of LP problem (3.26) by substituting $\Theta_{\mathcal{N}}$ for $\bar{\Theta}_{\mathcal{N}}$, then $\underline{C}'(\Theta^1_{\mathcal{N}}) \geq \underline{C}'(\Theta^2_{\mathcal{N}})$*

Proof. Suppose now the estimation of the project owner is $\Theta^1_{\mathcal{N}}$. The project owner will solve problem (3.26) first to find the optimal crashing schedule. Assume the optimal solution obtained by solving LP (3.26) is $\{\mathbf{x}^*_{\mathcal{N}}(\Theta^1_{\mathcal{N}}), \mathbf{t}^*_{\mathcal{N}}(\Theta^1_{\mathcal{N}})\}$ and the minimum cost is $\underline{C}'(\Theta^1_{\mathcal{N}}))$. The project owner then sets $\overline{\mathbf{x}}_{\mathcal{N}} = \mathbf{x}^*_{\mathcal{N}}(\Theta^1_{\mathcal{N}})$ and $\overline{\mathbf{t}}_{\mathcal{N}} = \max(\mathbf{t}^*_{\mathcal{N}}(\Theta^2_{\mathcal{N}}), \mathbf{t}^0_{\mathcal{N}})$. Moreover, following the procedures in Algorithm AE, the project owner can determine the parameter $\alpha^1_{\mathcal{N}}$ and $\varepsilon^1_{\mathcal{N}}$, which satisfy all the constraints of LP problem (3.28). That is

$$\begin{cases} \sum\limits_{i\in C(j)} (\alpha^1_i - \underline{\alpha}_i)\varphi_i + \sum\limits_{i\in \mathcal{A}(j)} (\alpha^1_i - \underline{\alpha}_i)\varphi_i = \sum\limits_{i\in C(j)} (1-\varepsilon^1_i)\bar{\Theta}^1_i & \forall j = 1,\dots,q \\ (\alpha^1_i - \underline{\alpha}_i)\varphi_i \leq (1-\varepsilon^1_i)\bar{\Theta}^1_i & \forall i \in \mathcal{N} \\ \underline{\alpha}_i \leq \alpha^1_i < 1,\ 0 < \varepsilon^1_i < 1 & \forall i \in \mathcal{N}, \end{cases} \tag{3.29}$$

The purpose is to motivate the subcontractors to crash as project owner expects. Since the cost estimation $\Theta^1_{\mathcal{N}}$ is great than or equal to the real crashing cost $\Theta_{\mathcal{N}}$, with the above $\alpha^1_{\mathcal{N}}$ and $\varepsilon^1_{\mathcal{N}}$, subcontractors are willing to do crashings since the bonus received will be great than the crashing cost. Therefore, the optimal crashing schedule expected by the project owner is achieved. According to Theorem 3.7, the cost for the project owner will be $\underline{C}'(\Theta^1_{\mathcal{N}})) + \mathbf{1}^\top \gamma_{\mathcal{N}}$.

Suppose now the estimation becomes $\Theta^2_{\mathcal{N}}$, then again project owner will solve the LP (3.26) to obtain the optimal crashing schedule for himself. Suppose now the optimal solution is $\{\mathbf{x}^*_{\mathcal{N}}(\Theta^2_{\mathcal{N}}), \mathbf{t}^*_{\mathcal{N}}(\Theta^2_{\mathcal{N}})\}$ and the minimum cost is $\underline{C}'(\Theta^2_{\mathcal{N}}))$. Compare the optimal solution with the previous one we have two possible situations to discuss: 1) the two optimal solutions are the same; and 2) the two optimal solutions are different.

If $\{\mathbf{x}^*_{\mathcal{N}}(\Theta^1_{\mathcal{N}}), \mathbf{t}^*_{\mathcal{N}}(\Theta^1_{\mathcal{N}})\} = \{\mathbf{x}^*_{\mathcal{N}}(\Theta^2_{\mathcal{N}}), \mathbf{t}^*_{\mathcal{N}}(\Theta^2_{\mathcal{N}})\}$, then we can similarly verify that project owner's minimum cost with estimation $\mathbf{\Theta}_{\mathcal{N}}^2$ will be $\underline{C}'(\mathbf{\Theta}_{\mathcal{N}}^2))+\mathbf{1}^\top\gamma_{\mathcal{N}}$. Furthermore, we can easily verify that $\underline{C}'(\mathbf{\Theta}_{\mathcal{N}}^2)) \leq \underline{C}'(\mathbf{\Theta}_{\mathcal{N}}^1))$ since the optimal solutions are the same and $\mathbf{\Theta}_{\mathcal{N}}^2 \leq \mathbf{\Theta}_{\mathcal{N}}^1$.

If $\{\mathbf{x}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^1), \mathbf{t}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^1)\} \neq \{\mathbf{x}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^2), \mathbf{t}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^2)\}$, then we know $\{\mathbf{x}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^1), \mathbf{t}_{\mathcal{N}}^*(\mathbf{\Theta}_{\mathcal{N}}^1)\}$ is a feasible solution in LP (3.26) with $\mathbf{\Theta}_{\mathcal{N}}^2$. Then basing on this feasible schedule, project owner can examine whether there exists $\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}}$ which satisfy all the constraints in (3.29) by substituting $\mathbf{\Theta}_{\mathcal{N}}^2$ for $\mathbf{\Theta}_{\mathcal{N}}^1$. Since there always exists a solution $\{\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}}\} = \{\underline{\alpha}_{\mathcal{N}}, \mathbf{1}\}$, we know this feasible schedule can always be achieved when project owner's estimation is $\mathbf{\Theta}_{\mathcal{N}}^2$. And with this feasible schedule, the objective value of LP (3.26) is just $\underline{C}'(\mathbf{\Theta}_{\mathcal{N}}^1)$. Hence the optimal value should satisfy $\underline{C}'(\theta_{\mathcal{N}}^2) \leq \underline{C}'(\theta_{\mathcal{N}}^1)$ □

Theorem 3.8 implies that if project owner does not believe the unit crashing cost information ($\bar{\mathbf{\Theta}}_{\mathcal{N}}$) provided by the subcontractor and prefers to establish contracts based on his own estimations, then the closer his estimations and the real value are, the lower total cost he can achieve. In this sense, the project owner should set $\tilde{\mathbf{\Theta}}_{\mathcal{N}}$ as small as possible. However, if the project owner's estimation is too small, which leads to underestimation of $\mathbf{\Theta}_{\mathcal{N}}$, the project owner may incur a huge loss if the compensation offered by the project owner is insufficient to induce crashing.

We conduct a computational experiment to evaluate the effects of the estimation of $\mathbf{\Theta}_{\mathcal{N}}$. The experiment is based on Example 3.2 and is designed as follows. The resource that subcontractor 5 owns is changed to 3 units and other subcontractors' resource remain the same, i.e., $\bar{\mathbf{r}}_{\mathcal{N}} = (1,1,1,2,3)$. The unit crashing cost is $\mathbf{\Theta}_{\mathcal{N}} = (1,1,3,1,1)$, but the information provided by the subcontractors is 100% more than the real cost, i.e., $\bar{\mathbf{\Theta}}_{\mathcal{N}} = (2,2,6,2,2)$. The project owner, based on experience, believes that the information is true except that $\bar{\Theta}_3 = 6$ is too high. Hence, he establishes his contracts based on information provided by subcontractor 1, subcontractor 2, subcontractor 4 and subcontractor 5, together with his own estimate of subcontractor 3's unit crashing cost. Thus,

$\tilde{\Theta}_{\mathcal{N}} = (2, 2, \tilde{\Theta}_3, 2, 2)$. We also assume the lower bound condition for setting $\alpha_{\mathcal{N}}$ in this example does not exist. Hence $\underline{\alpha}_i = 0, \forall i \in \mathcal{N}$.

In Table 3.1, we vary the project owner's estimate $\tilde{\Theta}_3$ from 4 to 1 in 0.5 decrements. For each estimation, we derive the maximum profit of the subcontractors and the minimum total variable cost of the project owner. We also show the optimal schedule used by the subcontractors.

Table 3.1: project owner and subcontractors' paybacks with different estimation on Θ_3

$\tilde{\Theta}_3$	project owner's expected cost saving	SC's schedule						project owner's real cost saving	SCs' extra Profit
		x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	t_5^*		
6.0	12	1	2	0	0	1	6	12	4
4.0	12	1	2	0	0	1	6	12	4
3.5	12	1	2	0	0	1	6	12	4
3.0	12	1	2	2	2	1	4	12	6
2.5	13	1	2	2	2	1	4	13	5
2.0	14	1	2	2	2	1	4	14	4
1.5	15	1	2	0	0	1	6	8	6
1.0	16	1	2	0	0	1	6	8	6

Observation I:

- If $\tilde{\Theta}_3 > 3$, it is not worthwhile for the project owner to crash tasks 3 and 4 together by 1 time unit since $\tilde{\Theta}_3 + \tilde{\Theta}_4 > \varphi_5$. Hence, he designs the same contracts and achieves the expected cost saving. If $\tilde{\Theta}_3 \leq 3$, it is worthwhile for the project owner to crash tasks 3 and 4. Therefore, he designs contracts which compensate subcontractors 3 and 4 for crashing. The result is that his cost saving keeps increasing until $\tilde{\Theta}_3$ reduces to 2. When $\tilde{\Theta}_3$ is below 2, the project owner underestimates Θ_3, leading to insufficient sharing of crashing cost for tasks 3 and 4. As a result, the subcontractors' schedule becomes different from what the project owner expects, which reduces the project owner's cost saving. And we can observe that the total cost of the project owner will be lower if his estimation is closer to the real value as long as project owner still overestimates the crashing cost.

- Providing overstated crashing cost information is not always a wise decision for the subcontractors. From the Table 3.1 we can see that if subcontractor 3 announces his cost greater than 3 and the project owner believes this information, then the extra profit that subcontractors can obtain is only 4. But if he just provides the true information, e.g., $\bar{\Theta}_3 = 3$, then the subcontractors can possibly receive 6 from the project owner. The reason is that overstated crashing cost may mislead the project owner into thinking it is not worthwhile to crash the task. However, both the subcontractor and project owner can benefit from task crashing, if the cost information is well stated. Another observation is that subcontractors may benefit when the project owner makes wrong estimations. As we can see, subcontractors' extra profit increases to 6 when project owner underestimates Θ_3.

- When $2 \leq \tilde{\Theta}_3 \leq 3$, the project owner underestimates the unit crashing cost of task 3. But it is interesting to see that the project owner's cost saving still increases as $\tilde{\Theta}_3$ declines. The reason is that although the project owner underestimates task 3's unit crashing cost, he overestimates the unit crashing cost of task 4. Through cooperation between the subcontractors, the effects of these imprecise estimations offset each other, and the schedule is as expected by the project owner. This result occurs when there are parallel crashing tasks in the optimal schedule, which generates a "risk pooling" effect.

3.5.2. Resource availability

The subcontractors' resources play an important role in their coordination. Resource sharing enables the subcontractors to do more task crashing than is possible using their own resources, and thereby increases their profit. In order to design contracts with subcontractors to encourage crashing, the project owner needs to know the resource amount owned by all the subcontractors. Since resources will have no value if unused, subcontractors are typically willing to share their resource amount information with the project owner. However, some subcontractors may have confidentiality concerns. In that case, the project owner needs to estimate the total amount of resource that *all* the subcontractors own. It is not necessary for the project owner to estimate the resource amount owned by each subcontractor, since the subcontractors game is balanced. This reduces the project owner's risk of imprecise estimation on the resource amount.

Let $\tilde{\mathbf{r}}_{\mathcal{N}}$ denote the project owner's estimate of the total resource owned by all the subcontractors which actually is $\mathbf{r}_{\mathcal{N}}$. Before we introduce the computational experiment, we first state a proposition which implies that project owner should not estimate too much amount of resources.

Proposition 3.9. *If project owner overestimate the amount of resources owned by the subcontractors, then the project may not implemented as he expects and his total cost may increase.*

The insight of Proposition 3.9 is that when project owner believes that the subcontractors own much resources, he will expect more crashings which leads to more cost savings on task completion time. Moreover, in his contract he sets the upper bounds of crashing quite high and these large upper bounds give SCs more flexibility to do crashing. And the optimal crashing schedule to subcontractors may be different from the schedule that project owner expects, which leads a huge loss to project owner. Next we will introduce a computational experiment which examines the impact of imprecise estimation $\tilde{\mathbf{r}}_{\mathcal{N}}$ to the maximum cost saving of project owner and profit of subcontractors. The observation in Proposition 3.9

can also be verified in this numerical example.

The experiment again is based on Example 3.2, with the following modifications. The real unit crashing cost is $\Theta_{\mathcal{N}} = (1, 1, 2, 1, 1)$, but the information provided by the subcontractors is $\bar{\Theta}_{\mathcal{N}} = (2, 1.5, 2.5, 2, 1.5)$. The project owner designs the contracts according to $\bar{\Theta}_{\mathcal{N}}$ and we assume $\underline{\alpha}_i = 0, \forall i \in \mathcal{N}$. The resources that subcontractors own are $\bar{\mathbf{r}}_{\mathcal{N}} = (1, 1, 1, 1, 2)$. But this time subcontractors refuse to tell the project owner their resource on hand. Thus, the project owner has to estimate the total resource amount. Suppose the estimated value is $\tilde{r}$ and we change it from 3 to 9. For each $\tilde{r}$, we derive the real schedule performed by the subcontractors, the maximum variable profit which subcontractors can receive and the cost saving of the project owner. The results are summarized in Table 3.2.

Table 3.2: project owner and subcontractors' paybacks with different estimation on $\bar{\mathbf{r}}_{\mathcal{N}}$

	PO's expected	SC's schedule						PO's real	SCs' extra
$\tilde{r}$	cost saving	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	t_5^*	cost saving	Profit
3	10	1	1	0	0	1	7	10	2
4	13.5	1	2	0	0	1	6	13.5	2.5
5	13.5	1	2	0	0	1	6	13.5	2.5
6	14	1	2	1	1	1	5	14	4
7	14	1	2	1	1	1	5	14	4
8	14.5	1	1	2	1	1	5	$12\frac{1}{3}$	$4\frac{2}{3}$
9	14.5	1	1	2	1	1	5	$12\frac{1}{3}$	$4\frac{2}{3}$

Observation II:

- When the project owner underestimate the subcontractors' resources, his cost saving is quite low, although the project is implemented as he expects since he sets the crashing upper bounds. As his estimation becomes closer to the real value, his cost saving increases. However, when the project owner overestimate the resource amount, the project may not be implemented as he plans and his cost saving may reduce. This phenomenon also verifies our Proposition 3.9. The reason is that overestimation brings larger crashing upper bound which gives subcontractors more flexibility to do crashings. Due to insufficient resource constraint, they just choose the most profitable crashing to process, which may lead to unexpected project schedule.

- The subcontractors' profit in this experiment keeps increasing when project owner's estimation becomes greater. The increase when $\tilde{r}$ changes from 3 to 6 is easy to understand, because as $\tilde{r}$ increases, the project owner also sets $\overline{\mathbf{x}}_{\mathcal{N}}$ larger, so that the subcontractors can perform more task crashing to receive more compensation from the project owner. But the increase when $\tilde{r}$ changes from 6 to 9 is unusual. Since the project owner has overestimated the crashing cost of tasks 3 and 4, the subcontractors crash task 3 instead of task 2, which increases their compensation. Generally, if the project owner overestimates the subcontractors' resource amount, the subcontractor receives at least the same profit as when the project owner's estimation is exactly correct.

3.6. Multiple Projects

In this section we consider another type of problems in project scheduling. Generally in such problems we have numbers of POs, each of whom owns an entire project to be carried out. Each project consists of several tasks and the duration times of these tasks are deterministic. POs can use additional resources to crash their tasks and complete their project at earlier times. But meanwhile they also encounter costs on crashing tasks. Therefore Cost/Time Tradeoff exists in each project. Furthermore, task crashing consumes resources which are assumed to be limited for each PO. Thus when the POs make project scheduling decisions, they should also take such resource constraints into consideration. In our model, we assume that POs can either subcontract their tasks to subcontractors or complete these tasks by themselves. Of course subcontractors are more professional on specific tasks and the cost to complete these tasks will be less if the tasks are processed by these professional SCs. Hence POs are willing to do the subcontracting and sometimes one SC who with special expertise will be subcontracted similar tasks from different POs. But the situation a PO may face is that the SC has already been booked by other POs during the time window the PO would like to process his task. In this case the PO has to process the task by himself or subcontract it to other less professional SCs, in the event that he wants the task to be completed as scheduled. The result is that the cost to process this task will be higher. Another choice for the PO is to postpone the start time of this task so that the most professional SC (with least cost) will be available then. Again the PO faces a Cost/Time tradeoff here. Therefore if POs collaborate with each other, they can benefit not only because resource sharing can provide them with the opportunity to do extra crashing that can not be performed if working independently, but also because the cost to process some similar tasks can be reduced from better scheduling by the professional SC in charge of these tasks. Again cooperation game theory will be applied to establish fair cost allocation scheme in a coalition, so that POs will perform cooperations automatically.

Below are some variable notations in our linear multi-project game problem.

We just list the variables which are not defined in the single project game case.

$\mathcal{K}$: Set of Projects(POs)

$\mathcal{N}_k$: Set of tasks belong to Project k

n_k : Number of tasks in project k. Assume 1 is the start task and n_k is the finish one

$\mathbf{x}^k$: $(x_1^k, \cdots, x_{n_k}^k)$, Number of time units to crash tasks in project k

$\mathbf{t}^k$: $(t_1^k, \cdots, t_{n_k}^k)$, Completion times of tasks in project k

$(\mathbf{t}^k)^0$: $((t_1^k)^0, \cdots, (t_{n_k}^k)^0)$, Earliest intial completion times of tasks in project k

$\bar{\mathbf{t}}^k$: $(t_1^k, \cdots, t_{n_k}^k)$, Upper bound of $\mathbf{t}^k$

$\mathbf{d}^k$: $(d_1^k, \cdots, d_{n_k}^k)$, Duration times of tasks in project k

$\bar{\mathbf{x}}^k$: $(\bar{x}_1^k, \cdots, \bar{x}_{n_k}^k)$, The maximum time units that tasks in project k can be crashed

T_k : Desired makespan in time units(Deadline) for project k

$\bar{\mathbf{r}}^k$: $(\bar{r}_1^k, \cdots, \bar{r}_m^k)$Total amount of different resource on PO m's hand

$\mathbf{r}^{\imath k}$: $(r_1^{\imath k}, \cdots, r_m^{\imath k})$, Amount of different types of resource consumed if we crash task $\imath$ in project k by one time unit

$\mathcal{P}_\imath^k$: Set of tasks that are immediate predecessors of task $\imath$ in project k

$\mathcal{A}_\imath^k$: Set of tasks that are immediate successors of task i in project k

$\mathbf{\Theta}^k$: $(\Theta_1^k, \cdots, \Theta_{n_k}^k)$Unit crashing costs of tasks in project k

$(\varphi^k)^\top \mathbf{t}^k$: Total cost incurred to POk if tasks are completed at time $\mathbf{t}^k$

$\mathcal{O}$: Set of professional SCs

$\mathcal{O}^\jmath$: Set of tasks that can be processed by professional SC $\jmath$

$\phi_\jmath^{\imath k}$: Unit cost saving on task $\imath$ in project k if the task is processed by professional SC $\jmath$

$\mathcal{S}$: Arbitary coaltion in $\mathcal{K}, \mathcal{S} \subseteq \mathcal{K}$

$C(\mathcal{S})$: Minimum total cost of projects(POs) in coalition $\mathcal{S}$

Initially, each PO will perform optimal scheduling independently according to the tasks' fore-and-aft relations, the tasks' normal duration times, the resource they own for crashing and the available time window at the professional SCs' side to subcontract tasks, so that they can obtain the optimal start and completion times of all his tasks. For example for PO k, suppose the optimal start and completion times of task $\imath$ to be s^* and t^*. If SC $\jmath \in \mathcal{O}$ can process task $\imath \in \mathcal{N}_k$

(in other words, $i \in \mathcal{O}^j$), PO k can subcontract this task to SC j, requiring that the task should be started at time s^* and finished at time t^*. If SC j is available during time window $[s^*,t^*]$, this task will be processed and PO k can achieve a cost saving as $\phi_j^{ik}(t^* - s^*)$. But if the time window is already booked or partially booked by other POs, then PO k can only subcontract a proportion of the task to the SC. Generally we have the following assumptions to specify the disciplines of task subcontracting in our model.

- SCs will process the tasks that POs subcontract to them in a ETFS (Earliest start Time First Serve) policy while preemption is not allowed. That is they will process the task with the earliest start time first until the task is finished.

- PO can out source any partial tasks to the SCs in case that SCs are not available to process the whole tasks. If PO k subcontracts α proportion of task i to SC j, then the cost saving he can achieve is $\alpha\phi_j^{ik}d_i^k$.

- Once the SC $j\mathcal{O}$ receives all the subcontracting requests from POs, the sequence to process the tasks is fixed and queue jumping is not allowed. In other words, each PO k's task i $i \in \mathcal{O}^j$ original has a position $\sigma^j(i)$ to be processed. It will not be started until his predecessor, which positions at $\sigma^j(i) - 1$, is finished. Note that such queue-jumping forbiddenness is quite common in practise. The purpose is to protect the priority of POs who place earlier orders and should be promised to be served during the booked time windows. Therefore we assume that for each SC j, there is an initial sequence to perform the tasks from different PO. This sequence depends on the initial project scheduling of POs.

Define $\mathbf{y}^k = (y_1^k, \cdots, y_{n_k}^k)$ the time length of tasks that PO wants to subcontract to SCs. If PO k works independently, he has to solve the following optimization problem to obtain the minimum cost

$$C(\{k\}) = \min_{\{\mathbf{x}^k,\mathbf{t}^k,\mathbf{y}^k\}} (\varphi^k)^\top \mathbf{t}^k + (\Theta^k)^\top \mathbf{x}^k - \sum_{j\in\mathcal{O}} \sum_{i\in\mathcal{O}^j} \phi_j^{ik} y_i^k \qquad (3.30)$$

s.t.

$$\begin{cases} t_i^k - x_i^k - t_j^k \geq d_i^k, & \forall i \in \mathcal{N}_k, j \in \mathcal{P}_i^k \\ t_1^k = 0, & \\ t_{n_k}^k \leq T_k, & \\ \sum\limits_{i \in \mathcal{N}_k} r_l^{ik} x_i^k \leq \bar{r}_l^k, & \forall l \in \mathcal{R} \\ z_i^k \leq d_i^k - x_i^k & \forall i \in \mathcal{N}_k \\ z_i^k \leq t_i^k - (t_{i'}^{k'})^0, & \forall j \in \mathcal{O}, \forall i \in \mathcal{N}_k \cap \mathcal{O}^j, i' \in \mathcal{N}_{k'} \cap \mathcal{O}^j, \sigma^j(i') + 1 = \sigma^j(i) \\ y_i^k \geq z_i^k, & \forall i \in \mathcal{N}_k \\ 0 \leq x_i^k \leq \bar{x}_i^k, & \forall i \in \mathcal{N}_k \\ t_i^k \geq 0 & \forall i \in \mathcal{N}_k \\ y_i^k \geq 0 & \forall i \in \mathcal{N}_k \end{cases}$$

Similar to in the single project case, the first three constraints of optimization problem (3.30) denote the task fore-and-aft relations while the fourth constraint is the resource limitation faced by the PO. The fifth to seventh constraints together with the last constraint demonstrate the rules that task subcontracting should follow. Basically, y_i^k should satisfy

$$y_i^k = \max\{0, \min\{d_i^k - x_i^k, t_i^k - (t_{i'}^{k'})^0\}\}, \forall i \in \mathcal{N}_k \cap \mathcal{O}^j, i' \in \mathcal{N}_{k'} \cap \mathcal{O}^j, \sigma^j(i') + 1 = \sigma^j(i) \tag{3.31}$$

which implies that the time length that task i can be subcontracted to SC j should not be greater than the entire processing time and the available time window exists at SC's side. Meanwhile, if task i is scheduled to be completed earlier than its initial predecessor in subcontracting, then the task should be finished by PO himself and the cost saving PO can achieve is always 0.

Since a task can at most be subcontracted to one SC, for notation convenience, we next consider ϕ_j^{ik} as ϕ^{ik} and hence $\phi^k = (\phi^{1k}, \cdots, \phi^{n_k k})$. Furthermore, denote the subcontracting predecessor of task i as i^- in project k^-. Also denote the subcontracting successor of task i as i^+ in project k^+. If a set of PO $\mathcal{S}$ cooperate with each other, then $\mathcal{S}$ should solve the optimizaton problem as follows:

$$C(\mathcal{S}) = \min_{\{\mathbf{x}^k,\mathbf{t}^k,\mathbf{y}^k\}} \sum_{k\in\mathcal{S}}\{(\varphi^k)^\top \mathbf{t}^k + (\Theta^k)^\top \mathbf{x}^k - (\phi^k)^\top \mathbf{y}^k\} \tag{3.32}$$

s.t.

$$\begin{cases} t_i^k - x_i^k - t_j^k \geq d_i^k, & \forall k\in\mathcal{S}, i\in\mathcal{N}_k, j\in\mathcal{P}_i^k \\ t_1^k = 0, & \forall k\in\mathcal{S} \\ t_{n_k}^k \leq T_k, & \forall k\in\mathcal{S} \end{cases} \tag{32.1}$$

$$\left\{ \sum_{k\in\mathcal{S}}\sum_{i\in\mathcal{N}_k} r_l^{ik} x_i^k \leq \sum_{k\in\mathcal{S}} \bar{r}_l^k, \quad \forall l\in\mathcal{R} \right. \tag{32.2}$$

$$\begin{cases} z_i^k \leq t_i^k - t_{i^-}^{k^-}, & \forall k\in\mathcal{S}, k^-\in\mathcal{S} \\ z_i^k \leq t_i^k - (t_{i^-}^{k^-})^0, & \forall k\in\mathcal{S}, k^-\notin\mathcal{S} \\ z_i^k \leq d_i^k - x_i^k & \forall k\in\mathcal{S}, k^+\in\mathcal{S} \\ z_i^k \leq (t_i^k)^0 - (t_i^k + x_i^k - d_i^k), & \forall k\in\mathcal{S}, k^+\notin\mathcal{S} \end{cases} \tag{32.3}$$

$$y_i^k \geq z_i^k, \ \forall i\in\mathcal{N}_k \tag{32.4}$$

$$x_i^k \leq \bar{x}_i^k, \ \forall i\in\mathcal{N}_k \tag{32.5}$$

$$x_i^k, t_i^k, y_i^k \geq 0 \ \forall i\in\mathcal{N}_k \tag{32.6}$$

Similar to the single project problem, we can derive the matrix form of the multi-project grand coalition problem as follows

$$C(\mathcal{K}) = \min_{\{\mathbf{x}^k,\mathbf{t}^k,\mathbf{y}^k\}} \sum_{k\in\mathcal{K}}\{(\varphi^k)^\top \mathbf{t}^k + (\Theta^k)^\top \mathbf{x}^k - (\phi^k)^\top \mathbf{y}^k\} \tag{3.33}$$

s.t.

$$\mathbf{A}^1_{\mathcal{N}_k}(k)\mathbf{x}^k + \mathbf{A}^2_{\mathcal{N}_k}(k)\mathbf{t}^k \geq \mathbf{b}^1_{\mathcal{N}_k}, \quad \forall k\in\mathcal{K} \tag{33.1}$$

$$\mathbf{B}^1_{\mathcal{N}_k}(k)\mathbf{x}^k + \mathbf{B}^2_{\mathcal{N}_k}(k)\mathbf{t}^k \geq \mathbf{b}^2_{\mathcal{N}_k}, \quad \forall k\in\mathcal{K} \tag{33.2}$$

$$\sum_{k\in\mathcal{K}} \mathbf{R}_{\mathcal{N}_k}\mathbf{x}^k \geq -\sum_{k\in\mathcal{K}} \bar{\mathbf{r}}^k \tag{33.3}$$

$$\mathbf{C}_{\mathcal{K}} \begin{pmatrix} \mathbf{t}^1 \\ \vdots \\ \mathbf{t}^K \end{pmatrix} - \begin{pmatrix} \mathbf{I}_{\mathcal{N}_1} & & \\ & \ddots & \\ & & \mathbf{I}_{\mathcal{N}_K} \end{pmatrix} \begin{pmatrix} \mathbf{z}^1 \\ \vdots \\ \mathbf{z}^K \end{pmatrix} \geq \mathbf{0} \tag{33.4}$$

$$-(\mathbf{x}^k + \mathbf{z}^k) \geq -\mathbf{d}^k, \quad \forall k\in\mathcal{K} \tag{33.5}$$

$$\mathbf{y}^k - \mathbf{z}^k \geq \mathbf{0}, \quad \forall k \in \mathcal{K} \tag{33.6}$$

$$-\mathbf{x}^k \geq -\bar{\mathbf{x}}^k, \quad \forall k \in \mathcal{K} \tag{33.7}$$

$$\mathbf{x}^k \geq \mathbf{0}, \mathbf{t}^k \geq \mathbf{0}, \mathbf{y}^k \geq \mathbf{0}, \quad \forall k \in \mathcal{K} \tag{33.8}$$

where $\mathbf{A}^{\imath}_{\mathcal{N}_k}(k)$, $\mathbf{B}^{\imath}_{\mathcal{N}_k}(k)$, $\mathbf{b}^{\imath}_{\mathcal{N}_k}(k)$, $\imath = 1, 2$ and $\mathbf{C}_{\mathcal{K}}$ are appropriately defined matrices and vectors to represent constraints (32.1) and (32.3), which are shown in Appendix B[2]

Similarly with appropriately defined corresponding matrices and vectors, we can rewrite the optimization problem for any coalition $\mathcal{S}$ into matrix form:

$$C(\mathcal{S}) = \min_{\{\mathbf{x}^k, \mathbf{t}^k, \mathbf{y}^k\}} \sum_{k \in \mathcal{S}} \{(\varphi^k)^\top \mathbf{t}^k + (\mathbf{\Theta}^k)^\top \mathbf{x}^k - (\phi^k)^\top \mathbf{y}^k\} \tag{3.34}$$

s.t.

$$\mathbf{A}^1_{\mathcal{N}_k}(k)\mathbf{x}^k + \mathbf{A}^2_{\mathcal{N}_k}(k)\mathbf{t}^k \geq \mathbf{b}^1_{\mathcal{N}_k}, \quad \forall k \in \mathcal{S} \tag{34.1}$$

$$\mathbf{B}^1_{\mathcal{N}_k}(k)\mathbf{x}^k + \mathbf{B}^2_{\mathcal{N}_k}(k)\mathbf{t}^k \geq \mathbf{b}^2_{\mathcal{N}_k}, \quad \forall k \in \mathcal{S} \tag{34.2}$$

$$\sum_{k \in \mathcal{S}} \mathbf{R}_{\mathcal{N}_k} \mathbf{x}^k \geq -\sum_{k \in \mathcal{S}} \bar{\mathbf{r}}^k \tag{34.3}$$

$$\mathbf{C}_{\mathcal{S}} \begin{pmatrix} \mathbf{t}^1 \\ \vdots \\ \mathbf{t}^K \end{pmatrix} - \begin{pmatrix} \mathbf{I}_{\mathcal{N}_1} & & \\ & \ddots & \\ & & \mathbf{I}_{\mathcal{N}_K} \end{pmatrix} \begin{pmatrix} \mathbf{z}^1 \\ \vdots \\ \mathbf{z}^K \end{pmatrix} \geq \mathbf{D}_{\mathcal{S}} \begin{pmatrix} (\mathbf{t}^1)^0 \\ \vdots \\ (\mathbf{t}^K)^0 \end{pmatrix} \tag{34.4}$$

$$\begin{cases} -(\mathbf{x}^k + \mathbf{z}^k) \geq -\mathbf{d}^k, & \forall k \in \mathcal{K}, k^+ \in \mathcal{K} \\ -(\mathbf{t}^k + \mathbf{x}^k + \mathbf{z}^k) \geq -((\mathbf{t}^k)^0 + \mathbf{d}^k), & \forall k \in \mathcal{K}, k^+ \notin \mathcal{K} \end{cases} \tag{34.5}$$

$$\mathbf{y}^k - \mathbf{z}^k \geq \mathbf{0}, \quad \forall k \in \mathcal{S} \tag{34.6}$$

$$-\mathbf{x}^k \geq -\bar{\mathbf{x}}^k, \quad \forall k \in \mathcal{S} \tag{34.7}$$

$$\mathbf{x}^k \geq \mathbf{0}, \mathbf{t}^k \geq \mathbf{0}, \mathbf{y}^k \geq \mathbf{0}, \quad \forall k \in \mathcal{S} \tag{34.8}$$

Remarks:

- The differences of constraints between problem (3.33) and (3.34) are: i) constraints (34.1),(34.2) and (34.6)-(34.8) are included in problem (3.33).

ii) both the left hand side (LHS) matrices and right hand side (RHS) vectors in constraints (34.3)-(34.5) are different. In fact the second type of difference is the reason why grand coalition can achieve the POs less cost.

- The detail form of $\mathbf{C}_{\mathcal{S}}$ and $\mathbf{D}_{\mathcal{S}}$ are provided in Appendix[2].

3.6.1. Linear Multi-Project Game and the Core

We next define the cooperative game, which we call Linear Multi-Project Game $(\mathcal{K}, v)$. In this game, the grand coalition is the set of the PO $\mathcal{K}$. For each subset $\mathcal{S} \subseteq \mathcal{K}$, the characteristic function is $v(\mathcal{S}) = C(\mathcal{S})$, which is defined by the optimization problem (3.34). Use the duality property and similar method as applied in linear single project game problem, we can prove the following theorem, which states that the linear multi-project game is also balanced and there is a core distribution which is constructed on the optimal solution to the dual problem of grand coalition.

Theorem 3.10. *Let* $\mathbf{b}^k$ *denote the RHS vector in the constraints of problem (3.30) and* $\mathbf{Z}^*_{\mathcal{K}}$ *the optimal solution of the dual problem of (3.33). The* $\lambda = (\lambda_1, \cdots, \lambda_K)$ *constitutes a core allocation of* $(\mathcal{K}, v)$*, where*

$$\lambda_k = (\mathbf{b}^k)^\top \mathbf{Z}^*_{\mathcal{K}} \tag{3.35}$$

The proof will follow the similar approach as we applied in Section 3. We will first verify that according to the special structure of our problems, the optimal dual solution of problem (3.33) will be a feasible solution in the dual problem of (3.34). Then due to the optimality of dual solution, we can further conclude that

$$\begin{cases} C(\mathcal{S}) = \sum\limits_{k\in\mathcal{S}} (\mathbf{b}^k)^\top \mathbf{Z}^*_{\mathcal{S}} \geq \sum\limits_{k\in\mathcal{S}} (\mathbf{b}^k)^\top \mathbf{Z}^*_{\mathcal{K}} \\ C(\mathcal{K}) = \sum\limits_{k\in\mathcal{K}} (\mathbf{b}^k)^\top \mathbf{Z}^*_{\mathcal{K}} \end{cases}$$

Therefore λ lies in the core of linar multi-project game $(\mathcal{K}, v)$.

3.7. Conclusions

In this chapter we study different project scheduling and subcontracting problems modeled as "Linear Project Games". In the single project case, we solve a Stackberg game problem between the PO and SCs first and find the optimal strategy of PO on designing the contracts with SCs. Then we use linear programming duality to analyze SCs' cooperative game problem and show that there exists an allocation in the core, which can be formulated with the optimal dual solution of the grand coalition. In the multi-project case, we model the problem as a piece-wise linear program and then transfer it to the linear case. Therefore by applying the similar approach in the single project case, we also proved the balancedness of the cooperative game and find the core allocation which benefit all the POs.

Since we expect this is the first work on the application of *Cooperative Game Theory* in the project scheduling problem with task subcontracting, the model we established is still quite basic and the approach is still generally based on duality property. An interesting question is whether more structural properties of the core allocation obtained from linear programming game can be discovered to uncover insights pertaining to the project management problem. More specifically, what are the distributions to be given to the critical tasks and non-critical tasks ? How much the SC can be allocated by giving one unit of resource to other SCs ? By solving these questions, we can have a better understanding on the core. Moreover, whether it is necessary to constraint our model setting as linear form also needs further consideration.

There are also other directions for future research. First, although we can construct an allocation in the core with the optimal dual solution, the problem of checking whether a given allocation is in the core or not is still open. Moreover, besides the core there are several other important concepts in the cooperative games such as the Shapley value [42] and the nucleolus (Maschler et al. [34]). Whether we can design efficient algorithm to compute them needs further discussion.

Finally in the multi-project case, due to technical issue, we have to assume the task processing sequence at the professional SC's facility can not be changed after being initially set. This assumption certainly should be relaxed if considering the practical situation. Hence, whether we can remodel the problem without this assumption and the original approach is still applicable is an interesting question to be discussed later. In fact in the next chapter we will discuss a problem related to manufacturing outsourcing. And in that problem we allow the job processor to change the processing sequence of the jobs to achieve the minimum cost. The techniques applied in the next chapter may inspire the solution approach of the question we leave here.

Appendix A:

[1]. **Detail form of $A^i_{\mathcal{N}}$, $B^i_{\mathcal{N}}$ and $\mathbf{b}^i_{\mathcal{N}}$, $i = 1, 2$**

Let $n_i = |\mathcal{P}^i|$ which denotes the number of immediate predecessors of task i, $i = 1, ..., n$. Assume the jth task in $\mathcal{P}^i$ is task $k^i_j \in \mathcal{P}^i$, $j = 1, ...n_i$.

- $\mathbf{A}^1_{\mathcal{N}} = \begin{pmatrix} \mathbf{A}^1_2 \\ \vdots \\ \mathbf{A}^1_n \end{pmatrix}$ where $\mathbf{A}^1_i = \begin{pmatrix} -e^i \\ \vdots \\ -e^i \end{pmatrix}$, $i = 2, ..., n$ is a $n_i \times n$ matrix and e^i is a n-dimension unit row vector with the ith component equals to 1.
- $\mathbf{A}^2_{\mathcal{N}} = \begin{pmatrix} \mathbf{A}^2_2 \\ \vdots \\ \mathbf{A}^2_n \end{pmatrix}$ where $A^i_2 = \begin{pmatrix} -e^i + e^{k^i_1} \\ \vdots \\ -e^i + e^{k^i_{n_i}} \end{pmatrix}$.
- $\mathbf{B}^1_{\mathcal{N}} = \begin{pmatrix} e^1 \\ -e^1 \\ \mathbf{0} \end{pmatrix}$ where $\mathbf{0}$ is a n-dimension zero row vector. $\mathbf{B}^2_{\mathcal{N}} =$

$$\begin{pmatrix} e^1 \\ -e^1 \\ e^n \end{pmatrix}.$$

- $\mathbf{b}^1_{\mathcal{N}} = \begin{pmatrix} \mathbf{b}^1_2 \\ \vdots \\ \mathbf{b}^1_n \end{pmatrix}$ where $\mathbf{b}^1_i = (-d_i, ..., -d_i)^\top$ is a n_i-dimension vector with all components equal to $-d_i$ (d_i is the normal duration of task i). $\mathbf{b}^2_{\mathcal{N}} = (d_1, -d_1, D)^\top$.

[2]. **Detail form of $A^i_{\mathcal{N}_k}$, $B^i_{\mathcal{N}_k}$, $\mathbf{b}^i_{\mathcal{N}_k}$, $i = 1, 2$ and $\mathbf{C}_{\mathcal{K}}$, $\mathbf{C}_{\mathcal{S}}$, $\mathbf{D}_{\mathcal{S}}$**

With the definition of $A^i_{\mathcal{N}}$, $B^i_{\mathcal{N}}$ and $\mathbf{b}^i_{\mathcal{N}}$, $i = 1, 2$, we know $A^i_{\mathcal{N}_k}$ has the same form as $A^i_{\mathcal{N}}$ if replace n in $A^i_{\mathcal{N}}$ with n_k. And the same relation can be derived between $B^i_{\mathcal{N}_k}$ and $B^i_{\mathcal{N}}$, as well as $\mathbf{b}^i_{\mathcal{N}_k}$ and $\mathbf{b}^i_{\mathcal{N}}$.

The detail form of $\mathbf{C}_{\mathcal{K}}$, $\mathbf{C}_{\mathcal{S}}$ and $\mathbf{D}_{\mathcal{S}}$ are listed as follows:

- $\mathbf{C}_{\mathcal{K}} = \begin{pmatrix} \mathbf{I}_{\mathcal{N}_1} & & \\ & \ddots & \\ & & \mathbf{I}_{\mathcal{N}_K} \end{pmatrix} - \begin{pmatrix} \mathbf{E}_{\mathcal{N}_1} & & \\ & \ddots & \\ & & \mathbf{E}_{\mathcal{N}_K} \end{pmatrix}$ where $\mathbf{E}_{\mathcal{N}_k}$ is a $n_k \times \sum_{k\in\mathcal{K}} n_k$ matrices with the ith row vector equals to $\mathbf{e}^{i^-}$, $i \in \mathcal{N}_k$. Here $\mathbf{e}^{i^-}$ is a $\sum_{k\in\mathcal{K}} n_k$-dimension unit vector with the component corresponding to task i^- equals to 1.

- $\mathbf{C}_{\mathcal{S}}$ is the same as $\mathbf{C}_{\mathcal{K}}$ after 1)replacing $\mathbf{e}^{i^-}$ with $\mathbf{0}$ if task i^- is not in coalition $\mathcal{S}$; 2) replacing $\mathbf{I}_{\mathcal{N}_k}$ with $\mathbf{0}_{\mathcal{N}_k}$ if $k \notin \mathcal{S}$

- $\mathbf{D}_{\mathcal{S}} = \begin{pmatrix} \mathbf{I}'_{\mathcal{N}_1} & & \\ & \ddots & \\ & & \mathbf{I}'_{\mathcal{N}_K} \end{pmatrix} - C(\mathcal{S})$, where $\mathbf{I}'_{\mathcal{N}_k} = \begin{cases} \mathbf{I}_{\mathcal{N}_k} & \text{If } k \in \mathcal{S} \\ \mathbf{0}_{\mathcal{N}_k} & \text{Otherwise} \end{cases}$

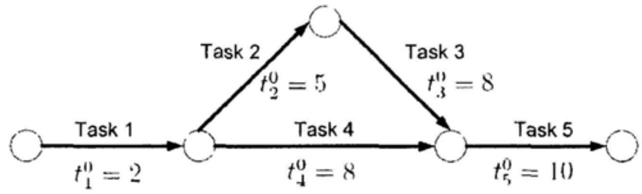


Figure 3.2: Normal Schedule

Appendix B: Example Problem

Example 3.2. Suppose that we have a project of 5 tasks. Task 1 is the first which has immediate successors 2 and 4, while task 5 is the last with immediate predecessors 3 and 4. Task 2 is the immediate predecessor of 3. See Figure 2. We assume there is only one type of resource, i.e., $\mathcal{R} = \{1\}$ and the related data of the tasks are listed in Table 1.

parameters	Task 1	Task 2	Task 3	Task 4	Task 5
d_i	2	3	3	1	2
$\overline{r}_{i1}$	1	1	1	1	0
r_{i1}	1	1	1	1	1
θ_i	1	1	2	1	1

Table 3.3: Data of the tasks/subcontractors

Under the normal schedule, the project will be implemented as shown in Figure 3.2 and the completion time for the project will be 10.

Suppose that the project owner can save $5x$ dollars if the project (last task) can be finished x (≥ 0) time units earlier, which implies that $\varphi_{\mathcal{N}} = (0, 0, 0, 0, 5)^{\top}$. Assume that the project owner signs contracts with the subcontractors with parameters as shown in Table 3.2.

variables	Task 1	Task 2	Task 3	Task 4	Task 5
β_i	0	0	0	0	5
ϑ_i	0	0	0	0	0
t_i^0	2	5	8	8	10
$\bar{t}_i$	2	5	8	8	10
$\bar{x}_i$	0	2	2	0	1

Table 3.4: Data of contracts

It is not difficult to verify that P1($\mathcal{N}$) now becomes

$$\max \quad \pi(\mathcal{N}) = -x_1 - x_2 - 2x_3 - x_4 - x_5 - 5t_5 + \sum_{i=1}^{5} \gamma_i + 50 \tag{3.36}$$

s.t.

$$\begin{pmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix} \mathbf{x}_{\mathcal{N}} + \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \mathbf{t}_{\mathcal{N}} \leq \begin{pmatrix} -3 \\ -3 \\ -1 \\ -2 \\ -2 \end{pmatrix} \tag{3.36.1}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \mathbf{x}_{\mathcal{N}} + \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{t}_{\mathcal{N}} \leq \begin{pmatrix} 2 \\ -2 \\ 10 \end{pmatrix} \tag{3 36.2}$$

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \mathbf{x}_{\mathcal{N}} \leq 4 \tag{3.36.3}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{x}_{\mathcal{N}} \leq \begin{pmatrix} 0 \\ 2 \\ 2 \\ 0 \\ 1 \end{pmatrix} \tag{3.36.4}$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{t}_{\mathcal{N}} \leq \begin{pmatrix} 2 \\ 5 \\ 8 \\ 8 \\ 10 \end{pmatrix} \tag{3.36.5}$$

$$\mathbf{x}_{\mathcal{N}}, \mathbf{t}_{\mathcal{N}} \geq 0 \tag{3.36.6}$$

The matrices and vectors $\mathbf{A}^i_{\mathcal{N}}$, $\mathbf{B}^i_{\mathcal{N}}$, and $\mathbf{b}^i_{\mathcal{N}}$, $i = 1, 2$, are clearly shown in the expressions (3.36.1) and (3.36.2) above.

Now consider the coalition $\mathcal{S} = \{2, 3, 4\}$. We can verify that problem P1($\mathcal{S}$) can be written as follows:

$$\max \quad \pi(\mathcal{S}) = -x_2 - 2x_3 - x_4 + \gamma_2 + \gamma_3 + \gamma_4 \tag{3.37}$$

s.t.

$$\begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{x}_{\mathcal{S}} + \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{t}_{\mathcal{S}} \leq \begin{pmatrix} -5 \\ -3 \\ -3 \\ 8 \\ 8 \end{pmatrix} \tag{3.37.1}$$

$$\begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \mathbf{x}_{\mathcal{S}} \leq 3 \tag{3.37.3}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{x}_{\mathcal{S}} \leq \begin{pmatrix} 2 \\ 2 \\ 0 \end{pmatrix} \tag{3.37.4}$$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{t}_{\mathcal{S}} \leq \begin{pmatrix} 5 \\ 8 \\ 8 \end{pmatrix} \tag{3.37.5}$$

$$\mathbf{x}_{\mathcal{S}}, \mathbf{t}_{\mathcal{S}} \geq 0 \tag{3.37.6}$$

The matrices $\mathbf{A}_{\mathcal{S}}^{i}$, $i = 1, 2$, and the vector $\mathbf{b}_{\mathcal{S}}^{1}$ are also clearly shown in the expression (3.37.1). Note that $\mathbf{B}_{\mathcal{S}}^{i}$, $i = 1, 2$, and $\mathbf{b}_{\mathcal{S}}^{2}$ are all zero because constraint (3.1.2) does not exist for the coalition $\mathcal{S}$ (so there is no constraint (3.37.2)).

Appendix C: Dual of the Example Problem

Consider the problem introduced in Appendix B. We obtain the dual problem D1($\mathcal{N}$) as follows.

$$\begin{aligned} \min \quad & (-3,-1,-3,-2,-2)(\rho_{21},\rho_{32},\rho_{41},\rho_{53},\rho_{54})^{\top} + (2,-2,10)\tau_{\mathcal{N}} + 4\omega_1 + (0,2,2,0,1)\mu_{\mathcal{N}} \\ & +(2,5,8,8,10)\zeta_{\mathcal{N}} + \sum_{i=1}^{5}\gamma_i + 50 \end{aligned}$$

s.t.

$$\left(\begin{array}{ccccc:ccc:c:ccccc:ccccc} 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hdashline 1 & 0 & 1 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} \rho_{\mathcal{N}} \\ \tau_{\mathcal{N}} \\ \omega_1 \\ \mu_{\mathcal{N}} \\ \zeta_{\mathcal{N}} \end{pmatrix} \geq \begin{pmatrix} -1 \\ -1 \\ -2 \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -5 \end{pmatrix}$$

$$\mathbf{z}_{\mathcal{N}} \geq \mathbf{0}$$

Also, originally without expansion, D1($\mathcal{S}$) can be written as follows:

$$\min \quad (-5,-3,-3,8,8)(\rho_{21},\rho_{32},\rho_{41},\rho_{53},\rho_{54})^{\top} + 3\omega_1 + (2,2,0)\mu_{\mathcal{S}} + (5,8,8)\zeta_{\mathcal{S}} + \sum_{i=2}^{4}\gamma_i$$

s.t.

$$\left(\begin{array}{ccccc:c:ccc:ccc} -1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hdashline -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right) \begin{pmatrix} \rho_{\mathcal{S}} \\ \omega_1 \\ \mu_{\mathcal{S}} \\ \zeta_{\mathcal{S}} \end{pmatrix} \geq \begin{pmatrix} -1 \\ -2 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{z}_{\mathcal{S}} \geq \mathbf{0}$$

In order for $\mathbf{z}_{\mathcal{S}}$ and $\mathbf{z}_{\mathcal{N}}$ to have the same dimension, we rewrite the above D1($\mathcal{S}$) as follow

$$\min \quad (-5,-3,-3,8,8)(\rho_{21},\rho_{32},\rho_{41},\rho_{53},\rho_{54})^{\top} + (\mathbf{0},\mathbf{0},\mathbf{0})\tau_{\mathcal{S}} + 3\omega_1 + (\mathbf{0},2,2,0,\mathbf{0})\mu_{\mathcal{S}}$$
$$+(\mathbf{0},5,8,8,\mathbf{0})\zeta_{\mathcal{S}} + \sum_{i=2}^{4}\gamma_i$$

s.t.

$$\left(\begin{array}{ccccc:ccc:c:ccccc:ccccc}
-1 & 0 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & 1 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} \\
0 & -1 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & 0 & 1 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} \\
0 & 0 & -1 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & 0 & 0 & 1 & \mathbf{0} & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} \\
\hdashline
-1 & 1 & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 1 & 0 & 0 & \mathbf{0} \\
0 & -1 & 0 & 1 & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 1 & 0 & \mathbf{0} \\
0 & 0 & -1 & 0 & 1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} & 0 & 0 & 1 & \mathbf{0}
\end{array}\right)
\begin{pmatrix} \rho_{\mathcal{S}} \\ \tau_{\mathcal{S}} \\ \omega_1 \\ \mu_{\mathcal{S}} \\ \zeta_{\mathcal{S}} \end{pmatrix}
\geq
\begin{pmatrix} -1 \\ -2 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\mathbf{z}_{\mathcal{S}} \geq \mathbf{0}$$

Note that the matrix in problem D1($\mathcal{S}$) consists of rows 2,3,4,7,8 and 9 of that in problem D1($\mathcal{N}$).

Appendix D: Payoff Distribution for the Example

Consider the example in Appendix B. Given the contracts with the project owner, the grand coalition's maximum profit can be obtained by solving problem P1($\mathcal{N}$). An optimal solution is

$$z^*_{\mathcal{N}} = (5,5,0,5,0,5,0,0,3,1,1,0,0,1,0,0,0,0,0)$$

That is, $\tau^*_1 = 5, \rho^*_{21} = 5, \rho^*_{32} = 5, \rho^*_{53} = 5, \omega^*_1 = 3, \mu^*_1 = \mu^*_2 = \mu^*_5 = 1$ and other variables equal to zero. Hence, the optimal value is $\pi(\mathcal{N}) = \sum_{i=1}^{5}\gamma_i + 15$.

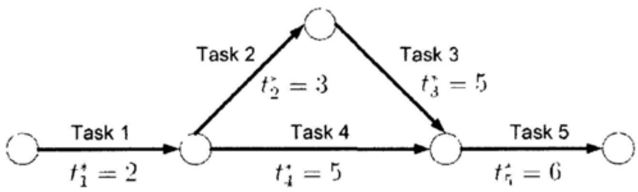


Figure 3.3: Crashing Schedule

Meanwhile, from Theorem 3.3, we have

$$\begin{cases} \lambda_1 = \gamma_1 + \omega_1^* + 2\rho_{21}^* - 2\tau_1 + 0 * \mu_1^* = \gamma_1 + 3 \\ \lambda_2 = \gamma_2 + \omega_1^* + 5\rho_{32}^* - (3+2)\rho_{21}^* + 2 * \mu_2^* = \gamma_2 + 5 \\ \lambda_3 = \gamma_3 + \omega_1^* + 8\rho_{53}^* - (3+5)\rho_{32}^* = \gamma_3 + 3 \\ \lambda_4 = \gamma_4 + \omega_1^* = \gamma_4 + 3 \\ \lambda_5 = \gamma_5 + 5t_5^0 - (2+8)\rho_{53}^* + \mu_5^* = \gamma_5 + 1 \end{cases}$$

and the new schedule is shown in Figure 3.3.

Without cooperation, each subcontractor's maximum profit is: $(\gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5)$. With cooperation and after distribution from the grand coalition, their profits become $(\gamma_1 + 3, \gamma_2 + 5, \gamma_3 + 3, \gamma_4 + 3, \gamma_5 + 1)$. Hence, no coalition can make more profit outside the grand coalition, and λ is a core solution. cooperative game.

Appendix E: Task Lengthening in the Example

Consider Example 3.2 again. Now let $\overline{r}_i = 0$, $i = 1, \ldots, 5$, which implies that no subcontractors own any resource for crashing. Thus, in order to do crashing, some tasks should be lengthened to release resource.

Solving the problem P1($\mathcal{N}$), we can get the optimal primal solution: $t_1^* = 2, t_2^* = 3, t_3^* = 6, t_4^* = 6, t_5^* = 7$, which indicates that task 4 is lengthened by 3 time units (to release 3 units of resource), task 2 is crashed 2 time units and task 5 is crashed 1 time unit.

By solving the problem D1($\mathcal{N}$), we can get the optimal dual solution:

$$\rho_{21} = 3,\ \rho_{32} = 3,\ \rho_{41} = 2,\ \rho_{53} = 3,\ \rho_{54} = 2,\ \tau_1 = 5,\ \omega_1 = 2,\ \mu_1 = 2,\ \mu_5 = 2$$

while other variables are 0. According to (3.22) we can compute the core distribution:

$$\begin{cases} \lambda_1 = \gamma_1 - \overline{x}_1\mu_1 - d_1\tau_1 + t_1^0\rho_{21} + t_1^0\rho_{41} = \gamma_1 - 2*5 + 2*3 + 2*2 = \gamma_1 \\ \lambda_2 = \gamma_2 - (t_1^0 + d_2)\rho_{21} + t_2^0\rho_{32} = \gamma_2 - (2+3)*3 - 5*3 = \gamma_2 \\ \lambda_3 = \gamma_3 - (t_2^0 + d_3)\rho_{32} + t_3^0\rho_{53} = \gamma_3 - (5+3)*3 - 8*3 = \gamma_3 \\ \lambda_4 = \gamma_4 - (t_1^0 + d_4)\rho_{41} + t_4^0\rho_{32} = \gamma_4 - (2+1)*2 - 8*2 = \gamma_4 + 10 \\ \lambda_5 = \gamma_5 + 5t_5^0 + \overline{x}_5\mu_5 - (t_3^0 + d_5)\rho_{53} - (t_4^0 + d_5)\rho_{54} = \gamma_5 + 50 + 2 - 10*3 - 10*2 = \gamma_5 + 2 \end{cases}$$

The distribution above indicates that task 4 is awarded \$10 for the resource it released, task 5 is awarded an extra of \$2. (But task 2 gets nothing although it was crashed; This is a bit surprising; We are still considering why.)

Appendix F: Nonuniform Contracts for the Example

Example 3.3. Recall example 3.2 in Appendix B, now we remain the structure of the project network the same but make some changes on the data of the tasks as follows

parameters	Task 1	Task 2	Task 3	Task 4	Task 5
d_i	2	3	3	1	2
$\overline{r}_{i1}$	0	1	1	0	0
r_{i1}	1	1	1	1	1
θ_i	∞	5	2	∞	∞

Table 3.5: *Data of the tasks/subcontractors

Suppose $\varphi_{\mathcal{N}} = (0, 6, 4, 0, 0)$ which implies that project owner can benefit only by the earlier completion of task 2 or 3. And we assume both task 2 and 3 can at most be crashed by 1 time unit. The lower bounds of task 2 and task 3 are set as $\underline{\alpha}_2 = 0.2$ and $\underline{\alpha}_3 = 0.1$. Next we check how Algorithm AE works on this example.

First with Step 1, we can solve LP problem (3.26) and obtain the optimal crashing schedule as $\mathbf{x}^*_{\mathcal{N}} = (0,1,1,0,0)$ and $\mathbf{t}^*_{\mathcal{N}} = (2,4,6,8,10)$. Moreover, the optimal objective value is $\underline{C}' = 57$. So we can set $X_L = 0$, $X_U = 2$, $x^L_i = x^U_i = 0, t^L_i = t^U_i = t^0_i, \forall i \in \mathcal{N}$, $C^L = \emptyset$ and q=0. Then we go to Step 2.

Then in Step 2 we set $X = \lceil (X_L + X_U)/2 \rceil = 1$. The we solve LP problem (3.28) and the optimal solution is $\mathbf{x}'_{\mathcal{N}} = (0,1,0,0,0)$ and $\mathbf{t}'_{\mathcal{N}} = (2,4,7,8,10)$. Hence, $C' = \{2\}$. Since $X < X_U$, we move to Step 4.

In Step 4, Since $C' \neq \emptyset$, we set $\mathbf{x}^U_{\mathcal{N}} = (0,1,0,0,0)$, $\mathbf{t}^U_{\mathcal{N}} = (2,4,7,8,10)$ and $X_U = 1$. Go to Step 2.

In Step 2, we set $X = \lceil (X_L + X_U)/2 \rceil = 1$. The we again need to solve LP problem (3.28) with $X = 1$. However, this time since $X = X_U = 1$, we move to Step 3.

In Step 3, we set $q = 1$, $C(1) = \{2\}$ and $C^L = \{2\}$. Since $A' = \{1,4,5\}$ and $\mathcal{A}^2 = \{3,5\}$, we have $A(1) = \{4\}$. Then we set $\mathbf{x}^L_{\mathcal{N}} = \mathbf{x}^U_{\mathcal{N}} = (0,1,0,0,0)$, $X^L = 1$ and $X_U = 2$, and go to Step 2.

In Step 2, we set $X = \lceil (X_L + X_U)/2 \rceil = 2$. The we solve LP problem (3.28) and the optimal solution is $\mathbf{x}'_{\mathcal{N}} = (0,1,1,0,0)$ and $\mathbf{t}'_{\mathcal{N}} = (2,4,6,8,10)$. We can verify that $C' = \{3\}$ and since $X = X_U$, we set $\mathbf{x}^U_{\mathcal{N}} = (0,1,1,0,0)$, $\mathbf{t}^U_{\mathcal{N}} = (2,4,6,8,10)$ and move to Step 3.

In Step 3, we let $q = 2$, $C(2) = \{3\}$ and $C^L = \{2,3\}$. Since $A' = \{1,2,4,5\}$ and $\mathcal{A}^3 = \{5\}$, we have $A(2) = \emptyset$. Set $\mathbf{x}^L_{\mathcal{N}} = \mathbf{x}^U_{\mathcal{N}}$, $\mathbf{t}^L_{\mathcal{N}} = \mathbf{t}^U_{\mathcal{N}}$. Since $X = \sum_{i \in \mathcal{N}} x^*_i$, we move to Step 5

In Step 5, we should solve the LP problem (3.28) which now becomes

$$\max_{\{\alpha_{\mathcal{N}}, \varepsilon_{\mathcal{N}}\}} 6(1-\alpha_2) + 8(1-\alpha_3) - 5\varepsilon_2 - 2\varepsilon_3$$

s.t.

$$\begin{cases} 6(\alpha_2 - 0.2) + 4(\alpha_3 - 0.1) = 5(1 - \varepsilon_2) \\ 4(\alpha_3 - 0.1) = 2(1 - \varepsilon_3) \\ 6(\alpha_2 - 0.2) \leq 5(1 - \varepsilon_2) \\ 4(\alpha_3 - 0.1) \leq 2(1 - \varepsilon_3) \\ 0.1 \leq \alpha_2 < 1, 0.2 \leq \alpha_3 < 1, 0 < \varepsilon_2 < 1, 0 < \varepsilon_3 < 1 \end{cases}$$

We can verify that $(\alpha_2, \alpha_3, \varepsilon_2, \varepsilon_3) = (0.4, 0.5, 0.44, 0.2)$ is one optimal solution. We next check if the project owner can achieve a minimum cost as $57 + \mathbf{1}^\top \gamma_\mathcal{N}$ by designing the contract with $(\alpha_2, \alpha_3, \varepsilon_2, \varepsilon_3) = (0.4, 0.5, 0.44, 0.2)$. For subcontractor 3, if he crashes one time unit, then his benefit will be $\alpha_3\varphi_3 - (1 - \varepsilon_3)\bar{\Theta}_3 = 0.4$. Hence, subcontractor 3 will do the crashing. For subcontractor 2, if he crashes one time unit, his cost will be $\bar{\Theta}_2 = 5$. But he can receive a bonus $\alpha_2\varphi_2 + \varepsilon_2\bar{\Theta}_2 = 4.6$ from the project owner. Furthermore, he can also receive a transfer payment from subcontractor 3 because subcontractor 3 can enjoy an earlier completion bonus $\alpha_3\varphi_3 = 2$ due to the crashing of task 2. And the amount of transfer payment depends on the bargaining power between subcontractor 2 and 3. Suppose the amount is TP and we know $TP \in (0.4, 2)$. So the total benefit received will be $4.6 + TP > 4.6 + 0.4 = 5$. Hence, subcontractor 2 will also do the crashing. With the above analysis, we know the project will be implemented as project owner has expected. The total cost for him then becomes

$$\varphi_2 t_2^* + \varphi_3 t_3^* + \varepsilon_2\bar{\Theta}_2 + \varepsilon_3\bar{\Theta}_3 + \alpha_2\varphi_2(t_2^0 - t_2^*) + \alpha_3\varphi_3(t_3^0 - t_3^*) + \mathbf{1}^\top\gamma_\mathcal{N} = 57 + \mathbf{1}^\top\gamma_\mathcal{N}$$

Therefore, we can conclude that with the $\alpha_\mathcal{N}$ and $\varepsilon_\mathcal{N}$ obtained from Algorithm AE, project owner can achieve the minimum cost he can expect.

CHAPTER 4

Cooperative Game in an Outsourcing Problem with Stochastic Processing Times

In the previous chapter, we study a project scheduling problem with subcontracting behaviors. We model the SCs' problem as a cooperative game which we called "Linear Project Game". And then solve the optimization problem for PO based on the SCs' reactions to his subcontracting contracts. In the last part of the previous chapter, we introduced a Linear Multi-Project Game where tasks belong to different project owners can be subcontracted to the same SC. Note that such subcontracting behavior is also quite common in practice, especially in the manufacturing industry. Taking into consideration of efficiency and economy, many manufacturing companies will outsource (subcontract) a certain part of their manufacturing work (other than the internal core business) to some third-party processing enterprises. And some professional third-party companies will be in charge of the jobs from different manufacturing companies. In this chapter, we will investigate whether *Cooperative Game theory* can again be applied in the these job-outsourcing problems. Note that in these problems, we should allow outsoured jobs to be re-sequenced by the processor if they belong to manufacturers in the same coalition. So especially we want to examine whether the cooperative games basing on these problems is still balanced.

4.1. Introduction and Literature Review

In this chapter we study a business model where a number of manufacturers are provided with outsourcing service by a single outside firm. Nowadays this problems are becoming more and more popular due to the industry division. Large manufacturers, in order to focus on the internal core competencies, prefer to outsource the non-core operations to third-party companies with specialized facilities. According to the survey constructed in Cai et al. [17], we can find a lot of practical cases related to these outsourcing behaviors. And in this chapter we will focus on a similar model as studied in [17]. Specifically, we assume the external third-party firm will first announce his available time slots for production and associated price for each time slot. Each manufacturer then places order, in a First Come First Book (FCFB) discipline, on time slots needed for his jobs' productions(operations). Note that in the model of Cai et al. [17], the job's production time is assumed to be deterministic. Hence when a manufacturer books the time slots for processing one of his job, the booking quantity is fixed and it is exactly the same as the production time of this job. But in our problem, we assume the production times are stochastic and independent from each other. Therefore, when booking the time slots, manufacturers should also taking the booking quantity into consideration. Besides the cost spent on booking time slots, the manufacturers also need to undertake weighted costs related to their jobs' completion times. Such cost can roughly be understood as the flow time cost occurred at the third-party side. The cause of such cost is that manufacturers have their own measures on the completion(delivery) time of their products. Because the products might sequentially be used in directly selling to customers, assembling and packaging or re-machining. Thus a corresponding cost will be charged job's completion time, which can be interpreted as manufacturer's lost on time delay of his whole project. Recently Aydinliyim and Variaktarakis et al. [2] studied a production planning setting similar to ours, where the cost criterion for each manufacturer is job flow time cost plus the booking costs. But again the job production time in their study was assumed to be fixed.

Since the production time is random for each job, there will be chances that the job can not be finished within the time slots originally booked by the manufacturer. To handle such fail-to-complete problem, we propose two mechanisms as follows:

1. Semi-finished job is acceptable. That is the job that can not be finished within the reserved time slots will be immediately removed off the production line at the third-party facility and be delivered to the associated manufacturer. The flow time of this job is calculated as the time that the manufacturer receive this job. And this job is still valuable in the sense that manufacturer can sell the finished part of the job (products) to the customers. However the manufacturer has to undertake a penalty fee charged on the unfinished part of this job. Hence in this mechanism the manufacturer should balance the cost saving, which is generated from booking fewer time slots, with the risk to suffer huge penalty fees.

2. Semi-finished job is forbidden. Based on this assumption, manufacturers need to make spot purchasing on extra time slots to process his jobs. The price for spot purchasing is greater than regular booking price and overtime work by the third-party will be arranged until the production is completed. Note that the over time production can only be implemented when the production line is free and without affecting the regular production. Hence the flow time of these jobs may be delayed. In this case, manufacturers' costs contain booking costs, job flow time costs, and the spot purchasing costs.

The main objective of this chapter is to study how the manufacturers could be coordinated so that all the manufacturers will benefit. Hence, we will model the manufacturers' interactions as a cooperative sequencing game. Sequencing games are at the interface of sequencing and scheduling of operations and cooperative game theory. Therefore they always involve two types of problems. The first type includes a series of optimization problems, where the optimal schedule that can minimize a certain cost objective will be derived for any given subset of players. And the second type is a game theoretic problem that models the interactions of the players as a cooperative game. To ensure the coordination of all the players, an incentive cost allocation scheme needs to be constructed so that any subset of players can not be better off by deviating from the grand coalition.

Sequencing games were introduced by Curiel et al. [21]. They considered a simple machine problem with cost criterion as the weighted flow time. They show that the associated sequencing game is convex so that the core is guaranteed to be non-empty; see Shapley et al. [44] for the core allocations of such convex games. For more general cost criterion and a special class of games (referred to as σ_0-component additive games), Curiel et al. [22] provide a core allocation which is named as β-rule. Hamers et al. [28] considered single machine games for jobs with release times and weighted flow time cost criterion and showed that the core of these games are not convex except when all the jobs have unit processing time or unit weights. Borm et al. [7] studied the case with due dates and they showed that for three different due date related cost criteria the core is nonempty and the convexity was proven for only a special subclass. In 2002, Curiel et al. [23] presented a survey of sequencing games from 1989. The first work where each player may have more than one job to be processed is presented in Calleja et al. [18]. They showed that the core is non-empty if the cost function is additive with respect to the initial order of jobs. Hamers et al. [29] consider the game with multiple machines for the first time and they also proved the balacedness of the associated m parallel machine sequencing game. Another model using the multiple machines is investigated in Calleja et al. [19] where a two-machine and two operations of each player problem is studied. They proved that although the game is not convex, the core exists.

The rest of this chapter will be organized as follows: We start the next section with formal notations of our model and some supporting assumptions.

In Section 4.3, we discusse the coordination results under the situation that the booking costs for all the time slots are the same. We will show that the cooperative game is balanced and the core can be derived based on the core allocation vector of a permutation game. Finally in Section 4.4, we will conclude our results obtained so far and propose some interesting extensions of our model.

4.2. Notations and Models

The problem can be stated as follows. Let M denote the set of manufacturers and P the single third party. Each manufacturer $m \in M$ has a set of N_m jobs to be operated by P. The outsourcing procedure follows the protocol like this: third-party P announce W_t, the available time window for production each day and the booking cost h_t, $t = 1, 2, ..., K$. Each manufacturer m then will book a set of time windows $\mathcal{W}_m \subseteq 1, 2, ..., K$ to implement his jobs' production. The objective for each manufacturer is to minimize his total expected cost, which includes booking cost $\sum_{t \in \mathcal{W}_m} h_t$ paid to P to reserve $|\mathcal{W}_m|$ days of production on P's facility, the jobs' completion time cost and the unfinished penalty(spot purchasing cost). Here we also allow partial-time-window order, hence booking cost can be stated as $\sum_{t \in \mathcal{W}_m} \theta_t h_t$, where $\theta_t \in [0, 1]$.

Without loss of generality, we assume that during a small time period T, P announces the time window information at time 0 and afterwards, the manufacturers place orders in a first-come-first-book (FCFB) policy. This indicates

$$\mathcal{W}_m \subseteq \{1, 2, ..., K\} - \mathcal{W}_1 - ...\mathcal{W}_{m-1}$$

for $m \in M$. And without loss of generality we assume all these activities happen simultaneously at time 0 and the production at the facility of P also starts at time 0.

Each job $j \in N$ takes a stochastic processing time p_j with c.d.f $F_j(\cdot)$ and p.d.f $f_j(\cdot)$. Let $|N_m| = n_m$ and denote the jth job of manufacturer m as the k_m^jth job in N. Assume the processing time of this job to be p_m^j and manufacturer m

will reserve time slot $\mathcal{W}_m^j \in \mathcal{W}_m$ to process it. Completed jobs will be delivered back to the manufacturer and the shipping time here is ignored. If jobs cannot be finished within the booked time slots, they will be immediately unloaded from the production line so that latter jobs can get started without any delay. We also use off-line setups to ensure that there are no setup delays between consecutive jobs. If spot purchasing is allowed, the overtime work for the unfinished jobs will be carried out in the earliest available time slots on the third-party's hand (originally booked time slots by other manufacturers can not be occupied).

Since the transportation time is ignored in this work, the finish time C_m^j for manufacturer m's job j is marked as the job flow time at the third-party. We assume that manufacturer m has an evaluation index h_m^j on the finish time of the jth job. Therefore the expected cost related to the jobs' finish times for manufacturer m is $\sum_{j \in N_m} h_m^j E[C_m^j]$.

As mentioned above, jobs' production times are stochastic and there will be chances that some jobs cannot be completed within booked time slots. We propose two models to handle such semi-finished products. The idea for the first model is "compensation", which means the manufacturer will not ask the third-party P to go on finishing the entire jobs. Instead the manufacturer m will pay a penalty fee to the unfinished part of job j at a unit price ρ_m^j. And the completion time C_m^j equals to the time when the job is moved off the production line. However in the second model, manufacturer m can make "spot purchasing" and use the purchased time slots to finish his jobs. Note that such spot purchasing behaviors can only order those time slots that are not previously reserved by the other jobs. In this case the manufacturer does not have to undertake the compensation cost, but an overtime production cost will be charged with a higher unit price h_O and the job's completion time will also be put off. The first model is suitable when manufacturer's jobs are contracted in the quantities of products, e.g., the whole job is to produce 1,000 automobile tires, and within the booked time slots, only 800 tires are produced. Then the manufacturer can pay penalty fees to the 200 tires to represent his loss. The second model are more appropriate when the manufacturer' job is indivisible, e.g., the spray-paint of a car. The manufacturer cannot sell the partially spray-painted cars to the customers with a discount for the unfinished paint part. So the manufacturer have to make spot purchasing and wait until the job is entirely completed to make profit.

In this chapter we just focus on the first model and we leave the study of the second model as future work. Following are some important model assumptions we required in this chapter:

1. All the time windows $W_t = [a_t, b_t]$ has the same length L, i.e., $b_t = a_t + L$ for t=1,2,...,K. Therefore, each time window can divided into L time slots. Since partial time window booking is allowed, in the following part of this chapter we prefer to use one time slot as a unit, instead of one time window;

2. The booking cost for every time windows takes two values: h_R for regular demand days and h_P for peak demand days. In other words, to order one unit time slot, the cost for the manufacturer should be either h_R/L or h_P/L. Obviously we should set $h_R/L < h_P/L < h_O$. And furthermore the unit booking cost should also satisfy $h_R/L(h_p/L) + h^j_m < \rho^j_m \; \forall m \in M, j \in N$, otherwise there is no incentive for manufacturer to book any time slot to process job j.

3. Preemption is allowed in our model. Furthermore, we assume that if job j, after processing for a time period t, is preempted by another job and thereby be moved off the production line, then it can be continued again later. The remaining time to finish this job will be $p_j - t$, which is still random.

4.3. Optimal Strategies in a Special Case: $h_R = h_P = h$

In this section we begins our study on the first model we proposed in the last section, that is manufacturers pay compensations when jobs are not finished within time windows which are booked originally. We also constrain our study on a special case in this model, that is we assume $h_R = h_P = h$, which implies that all the time windows have the same booking cost. The reason we focus on this special case is that the approaches applied and results obtained in this special case will be very instructive. The study of the general case should be greatly inspired from the approaches applied and results obtained in this special case. Since each manufacturer can own more than one job, we first study the optimal booking strategy of each manufacturer who work independently with each other. Then we allow the manufacturers to collaborate and use cooperative game (especially sequencing game) theory to study the interactions among the manufacturers.

4.3.1. Manufacturers' Booking Strategy: Without Cooperation

When the booking costs for all the time windows are equal, it is not difficult to verify that manufacturers will book the earliest available time slots exists on the third-party P's hand. The problems left for the manufacturers are: 1)determine the optimal quantities of time slots to reserve for processing their jobs and 2)find the optimal sequence to process his jobs. Suppose the first time window the manufacturer m books is $\mathcal{W}_m^e$ and the last one is $\mathcal{W}_m^l$. Then the booking cost for the manufacturer should be

$$BC_m = [\theta_m^e + \theta_m^l + (|\mathcal{W}_m| - 2)]h \tag{4.1}$$

where $\theta_m^e, \theta_m^l \in (0,1]$ and their values depend on how much proportion the manufacturer books in his first and last time windows. And the total time he can used for production is

$$T_m = [\theta_m^e + \theta_m^l + (|\mathcal{W}_m| - 2)]L \tag{4.2}$$

Define TEC_m to be the total expected cost of manufacturer m and let TEC_m^j be the expected total cost related to his jth job. Also define T_m^j as the maximum time reserved for the jth job. In other words, if the jth job starts at time s_j and finishes production at time c_j before $s_j + T_m^j$, then its following job will be started immediately at time c_j. But if the jth job fails to be completed within time period T_m^j, it has to be removed off from the production line at time $s_j + T_m^j$ and the succeeded job will be started without any delay. And naturally from the definition, we have

$$\sum_{j \in N_m} T_m^j = T_m$$

The expected total cost of manufacturer m can be stated as follows:

$$TEC_m = \sum_{j \in N_m} TEC_m^j = \sum_{j \in N_m} \{\frac{h}{L} T_m^j + \rho_m^j \int_{T_m^j}^{\infty} (x - T_m^j) dF_{m_j}(x) + h_m^j E[C_m^j]\} \tag{4.3}$$

where $E[C_m^j]$ is the expected completion time of the jth job, which is actually condition on the completion time of his preceding job. Specifically we have

$$E[C_m^j] = \begin{cases} \int_0^{T_m^j} x dF_{m_j}(x) + \int_{T_m^j}^{\infty} T_m^j dF_{m_j}(x) + E[C_m^{j-1}] & j \in N_m \setminus \{1\} \\ S_m + \int_0^{T_m^1} x dF_{m_1}(x) + \int_{T_m^1}^{\infty} T_m^1 dF_{m_1}(x) & j = 1 \end{cases} \tag{4.4}$$

where S_m is the earliest start time of manufacturer m's jobs. Combining equations 4.4 and 4.3, we have

$$TEC_m = \frac{h}{L} T_m + \sum_{j \in N_m} \{\rho_m^j \int_{T_m^j}^{\infty} (x - T_m^j) dF_{m_j}(x) + h_m^j S_m + \sum_{k=j}^{n_m} h_m^k [\int_0^{T_m^j} x dF_{m_j}(x) + \int_{T_m^j}^{\infty} T_m^j dF_{m_j}(x)]\} \tag{4.5}$$

With the expected cost given in equation (4.5), manufacturers m has to determine the optimal booking quantity $\mathbf{T}_m^* = \{T_m^{1*}, T_m^{2*}, ..., T_m^{n_m *}\}$ and optimal sequence σ_m^* so that

$$J(\mathbf{T}_m^*, \sigma_m^*) = \min_{\mathbf{T}_m, \sigma_m} TEC_m \tag{4.6}$$

Notice that (4.6) is a two-variable minimization problem where $\mathbf{T}_m^*$ and σ_m^* are dependent. So next we will apply the Backwards Approach to find some structural properties of the optimal solution(s). Basically we want to investigate the optimal solution of one parameter when the other one is fixed(given). For example we assume now an arbitrary booking time quantity vector $\mathbf{T}_m$ is given. Let

$$P_m^j(T_m^j) = \frac{\int_0^{T_m^j} x dF_j(x) + \int_{T_m^j}^{\infty} T_m^j dF_j(x)}{h_m^j}$$

Then we have the following Lemma which states the optimal sequence to processing the jobs condition on $\mathbf{T}_m = (T_m^1, \cdots, T_m^{n_m})$.

Lemma 4.1. *If the time slot booked by manufacturer m is $\mathbf{T}_m = (T_m^1, T_m^2, ..., T_m^{n_m})$, the optimal sequence $\sigma_{\mathbf{T}_m}$ to process his jobs should follow an ascending order of $P_m^j(T_m^j)$, that is*

$$P_m^{k_1}(T_m^{k_1}) \leq P_m^{k_2}(T_m^{k_2}) \leq ... \leq P_m^{k_{n_m}}(T_m^{k_{n_m}}) \tag{4.7}$$

where k_i is the ith component of $\sigma_{\mathbf{T}_m}$, $i \in \{1, ..., n_m\}$.

Proof. We will use the interchange argument approach to verify (4.7). Let σ_m an arbitrary sequence other than $\sigma_{\mathbf{T}_m}$. Then there must exist $i \in N_m$ so that $P_m^{\sigma_m(i)}(T_m^{\sigma_m(i)}) > P_m^{\sigma_m(i+1)}(T_m^{\sigma_m(i+1)})$. Exchange the positions of these two jobs and keep the other jobs' positions unchanged. Suppose the new sequence we obtained is σ'_m, and for notational convenience we just denote the ith component of σ_m as i. Then we have

$$\begin{aligned}
& TEC_m(\sigma_m, \mathbf{T}_m) - TEC_m(\sigma'_m, \mathbf{T}_m) \\
= \; & h_m^{i+1}[\textstyle\int_0^{T_m^i} x dF_i(x) + \int_{T_m^i}^{\infty} T_m^i dx] - h_m^i[\int_0^{T_m^{i+1}} x dF_{i+1}(x) + \int_{T_m^{i+1}}^{\infty} T_m^{i+1} dx] \\
= \; & h_m^{i+1} h_m^i [P_m^i(T_m^i) - P_m^{i+1}(T_m^{i+1})] \\
> \; & 0
\end{aligned}$$

Therefore σ'_m can bring the manufacturer less cost and it is better than σ_m. We can repeat the above exchanging procedure until we can not find any job i which satisfies $P_m^i(T_m^i) > P_m^{i+1}(T_m^{i+1})$. Obviously the final optimal sequence we obtained will follow the ascending order of $P_m^i(T_m^i)$. □

On the other hand, if the sequence is given as σ_m, then by taking derivatives to TEC_m on T_m^j $j \in N_m$ respectively, we can obtain the optimal booking quantity for each job. The result is concluded in Lemma (4.2)

Lemma 4.2. *Given arbitrary sequence σ_m, without loss of generality we denote the jth job as job $j \in N_m$. The manufacturer m's optimal strategy on booking time slot for the jth job is*

$$T_m^{j*}(\sigma_m) = \begin{cases} F_{m_j}^{-1}\left(\frac{\rho_m^j - \sum_{k=j}^{n_m} h_m^k - h/L}{\rho_m^j - \sum_{k=j}^{n_m} h_m^k}\right) & \text{if } \rho_m^j - \sum_{k=j}^{n_m} h_m^k - h \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad j = 1, 2, \ldots n_m \tag{4.8}$$

Proof. Taking first and second derivatives of equation (4.5) on T_m^j, we have

$$\frac{dTEC_m^j}{dT_m^j} = \frac{h}{L} - [\rho_m^j - \sum_{k=j}^{n_m} h_m^k]\bar{F}_{m_j}(T_m^j)$$

$$\frac{d^2TEC_m^j}{d(T_m^j)^2} = [\rho_m^j - \sum_{k=j}^{n_m} h_m^k]f_{m_j}(T_m^j)$$

Since $\bar{F}_{m_j}(T_m^j) \in [0, 1]$, if $\rho_m^j - \sum_{k=j}^{n_m} h_m^k - \frac{h}{L} < 0$, the value of $\frac{dTEC_m^j}{dT_m^j}$ will always be nonnegative. Hence manufacturer m should set $T_m^{j*} = 0$. Otherwise if $\rho_m^j - \sum_{k=j}^{n_m} h_m^k - h \geq 0$, $\frac{d^2TEC_m^j}{d(T_m^j)^2}$ will always be positive and there is a unique minimum point T_m^{j*} which should satisfy

$$\frac{h}{L} - [\rho_m^j - \sum_{k=j}^{n_m} h_m^k]\bar{F}_{m_j}(T_m^{j*}) = 0$$

or equivalently $T_m^{j*} = F_{m_j}^{-1}\left(\frac{\rho_m^j - \sum_{k=j}^{n_m} h_m^k - h/L}{\rho_m^j - \sum_{k=j}^{n_m} h_m^k}\right)$ □

Remark:

- In Lemma 4.1, P_m^j can be considered as the weighted expected processing time of the jth job conditioned on an upper bound T_m^j. If preemption is not allowed or if we consider $T_m^j = \infty$, then P_m^j just becomes $E[p_j]/h_m^j$, which implies that the manufacturer should sequence his jobs by weighted shortest expected processing time (WSEPT) rule.

- $\rho_m^j - \sum_{k=j}^{n_m} h_m^k - h/L < 0$ in equation (4.8) actually implies that the penalty cost for each uncompleted unit of the jth job is smaller than the total cost generated to the manufacturer if he reserves one more time unit to process it. In such situation, the jth job will never be processed. To make the model more reasonable, we assume such situation should be avoided in our model. That is we assume for any job $j \in N_m$, ρ_j should satisfy

$$\rho_j - \sum_{k=1}^{n_m} h_m^k - h/L \geq 0 \tag{4.9}$$

With this assumption, we can remove the case $T_m^{j\,*}(\sigma_m) = 0$ in equation (4 8).

- From equation (4.8) in Lemma 4.2 we can observe that the optimal processing time of the jth job of manufacturer m in fact just depends on what its successors are. The optimal processing time of the jth job will not be affected by the sequence of his successors. We conclude this observation in the corollary below.

Corollary 4.3. *For job j of manufacturer m, let this job be sequenced at the k_jth position and the successors of this job be fixed. Denote $Sr^j = \{i|k_i \geq k_j\}$ which represents the job set that includes job j and all its successors. Then manufacturer m's optimal booking time strategy for job j should be*

$$T_m^{j\,*}(Sr^j) = \begin{cases} F_{m_j}^{-1}\left(\frac{\rho_m^j - \sum_{k\in Sr^j} h_m^k - h/L}{\rho_m^j - \sum_{k\in Sr^j} h_m^k}\right) & \text{if } \rho_m^j - \sum_{k\in Sr^j} h_m^k - h \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad j = 1, 2, \ldots n_m \tag{4.10}$$

Lemma 4.1 and 4.2 show two characteristics of the optimal solution(s). The question is that how to find the optimal solution so that these two characteristics can be satisfied simultaneously. We will use"back to front" dynamic programming algorithm to find the optimal strategy. That is we begin with the selection of the last job that manufacturer m should process and derive the optimal sequence from back to front. Meanwhile the optimal booking quantities will be calculated once their position in the sequence is determined. Before we show the detailed algorithm to obtain the optimal solution for problem (4.6), we first introduce a simple example to illustrate the idea of our approach.

Example 4.1. Suppose that the manufacturer m has three jobs to process labeled as jobs 1, 2 and 3. According to Lemma 4.2, if job i, $i = 1,2,3$, is the last job in the optimal sequence, then the booking time for this job should be $T_m^{i\,*} = F_i^{-1}(\frac{\rho_m^i - h_m^i - h/L}{\rho_m^i - h_m^i})$. And if job j is the second one to be processed, then the optimal booking time for j is $T_m^{j\,*} = F_j^{-1}(\frac{\rho_m^j - h_m^j - h_m^i - h/L}{\rho_j - h_m^j - h_m^i})$, $j \neq i$. Furthermore, according to Lemma 4.1, job i and j should satisfy

$$\frac{\int_0^{T_m^{j\,*}} x f_j(x)dx + \int_{T_m^{j\,*}}^{\infty} T_m^{j\,*} f_j(x)dx}{h_m^j} \leq \frac{\int_0^{T_m^{i\,*}} x f_i(x)dx + \int_{T_m^{i\,*}}^{\infty} T_m^{i\,*} f_i(x)dx}{h_m^i} \tag{4.11}$$

Therefore we should sequence job i as the last job if the above inequality can be satisfied with all $j \in \{1,2,3\}\setminus\{i\}$. And we can prove by contradiction that there exists at least one such job. The details of the contradiction method is provided as follows:

Denote job i's optimal booking time as $T_m^{i\,*}(k)$, $k = 1,2,3$ if i is sequenced at the kth position. Then we have

$$T_m^{i\,*}(1) = F_i^{-1}\left(\frac{\rho_m^i - h/L - \sum_{k=1}^{3} h_m^k}{\rho_m^i - \sum_{k=1}^{3} h_m^k}\right) < F_i^{-1}\left(\frac{\rho_m^i - h/L - h_m^i - h_m^j}{\rho_m^i - h_m^i - h_m^j}\right) = T_m^{i\,*}(2),\ i,j \in \{1,2,3\}, i \neq j$$

and similarly $T_m^{i\,*}(2) < T_m^{i\,*}(3)$. Furthermore form the definition of $P_m^i(T_m^i)$ in Lemma 4.1 and the relation among $T_m^{i\,*}(1)$, $T_m^{i\,*}(2)$ and $T_m^{i\,*}(3)$, we have

$$P_m^i(T_m^{i\,*}(1)) \leq P_m^i(T_m^{i\,*}(2)) \leq P_m^i(T_m^{i\,*}(3)) \tag{4.12}$$

Suppose there is no job i that can satisfy inequality (4.11) for all $j \in \{1,2,3\}\backslash\{i\}$. Then equivalently we have

$$P_m^1(T_m^{1*}(2)) > P_m^2(T_m^{2*}(3)) \tag{4.13}$$

$$P_m^2(T_m^{2*}(2)) > P_m^3(T_m^{3*}(3)) \tag{4.14}$$

$$P_m^3(T_m^{3*}(2)) > P_m^1(T_m^{1*}(3)) \tag{4.15}$$

Note that equation (4.13)-(4.15) can not be satisfied simultaneously. Otherwise

$$P_m^1(T_m^{1*}(2)) > P_m^2(T_m^{2*}(3)) > P_m^2(T_m^{2*}(2)) > P_m^3(T_m^{3*}(3)) > P_m^3(T_m^{3*}(2)) > P_m^1(T_m^{1*}(3))$$

which is a contradiction to inequality (4.12). Hence there exists at least one job that can satisfy inequality (4.11) for all $j \in \{1,2,3\} \setminus \{i\}$. And we can sequence this job as the last job.

Without loss of generality, we assume the last job to be 3. Then next we can select the second job from $\{1,2\}$. The procedure is quite similar. We first calculate $T_m^{1*}(1)$, $T_m^{1*}(2)$, $T_m^{2*}(1)$ and $T_m^{2*}(2)$ respectively. Then we compare $P_m^1(T_m^{1*}(1))$ with $P_m^2(T_m^{2*}(2))$. If $P_m^1(T_m^{1*}(1)) \le P_m^2(T_m^{2*}(2))$, we choose job 2 as the second job to be processed. Otherwise if $P_m^1(T_m^{1*}(1)) > P_m^2(T_m^{2*}(2))$, then we must have $P_m^2(T_m^{2*}(1)) \le P_m^1(T_m^{1*}(2))$ (similar proof by contradiction as above). So we choose job 1 the second job to be processed.

With the above backward approach, we can find the optimal strategy of manufacturer m on both the booking quantities and the sequence.

From Example 4.1, we can have a more clear view about the backward approach we are going to apply to search the optimal solution for problem (4.6). The procedure is summarized in DP OPTIMAL where $Z_m(j, S_j)$ is defined as the minimum expected total cost of manufacturer m if he sequences his jobs in $S_j \subseteq N_m$ in the first j positions.

DP OPTIMAL: Optimal Booking Strategy for manufacture m

Recursive relation:

$$Z_m(j, S_j) = \min_{i \in S_j} \left\{ (\sum_{k \in Sr^i} h_m^k) h_m^i P_m^i((T_m^i)^*(Sr^i)) + Z_m(j-1, S_j \setminus \{i\}) \right\}$$

for $j = 1, ..., n_m$, $S_j \subseteq N_m$. And Sr^i follows the definition in Corollary 4.3.

Boundary conditions: $Z_m(0, \emptyset) = 0$.

Optimal value: $TEC_m^* = Z_m(n_m, N_m)$.

4.3.2. Manufacturers' Booking Strategy: With Cooperation

In the above section we discussed the optimal strategy of each manufacturer when he work independently. As we may observe, the total expected cost of each manufacturer is significantly affected by his arrival time at the third party P, since all the early time windows will be booked by manufacturers who arrive at P before him. If the manufacturers can cooperate with each other, then the jobs required to be processed at P can be rescheduled and the total expected cost of the manufacturers can be reduced due to two possible reasons

1. An overall optimal sequence will be derived so that the total cost can be further minimized;

2. Fewer time slots will be reserved to process certain jobs, because reducing the booking time of a job can increase the saving on its successors' completion time cost, which may be greater than the possible penalty cost on incomplete job. Therefore the booking cost can be further reduced.

Therefore if the manufacturers form a coalition $S \subseteq M$, the total expected cost of the coalition TEC_S will be less than $\sum_{i \in S} TEC_i$. And the problem left is whether we can find a cost allocation scheme so that all members in S will not deviate from coalition S. To study the cost allocation problem, we model it as a cooperative game denoted by (M, v). Here M is the grand coalition of the manufacturers and any subset $S \subseteq M$ is a small coalition. The characteristic function $v(S) = \min TEC_S$ where TEC_S is total expected cost achieved by coalition S. Recall that the booking procedure obeys the FCFB protocol. For this reason we assume that if manufacturer m is not in coalition S, then the time slots he booked can not be occupied by manufacturers in coalition S. We denote the start time and finish time of manufacturer m's booking time slots as a_m and b_m respectively. If the sequence to process the jobs is σ_S, then TEC_S can be stated as follows

$$TEC_S = \sum_{m\in S}\sum_{j\in N_m} TEC_m^j = \sum_{m\in S}\sum_{j\in N_m}\{\frac{h}{L}T_m^j + \rho_m^j\int_{T_m^j}^{\infty}(x - T_m^j)dF_{m_j}(x) + h_m^j E[C_m^j]\} \tag{4.16}$$

Here we suppose the jth job of manufacture m to be sequenced at the k_m^jth position in σ_S, $k_m^j \in \{1, ..., \sum_{m\in S} n_m\}$. And we assume its predecessor is the j'th job of manufacturer m'. Hence

$$E[C_m^j] = \begin{cases} E[C_{m'}^{j'}] + \int_0^{T_m^j} x dF_{m_j}(x) + \int_{T_m^j}^{\infty} T_m^j dF_{m_j}(x) + \sum\limits_{i\notin S} H_i(E[C_{m'}^{j'}], h_m^j P_m^j(T_m^j))T_i & \text{if } k_m^j \neq 1 \\ \min\limits_{i\in S} a_i + \int_0^{T_m^j} x dF_{m_j}(x) + \int_{T_m^j}^{\infty} T_m^j dF_{m_j}(x) + \sum\limits_{i\notin S} H_i(E[C_{m'}^{j'}], h_m^j P_m^j(T_m^j))T_i & \text{otherwise} \end{cases}$$

where $H_i, i \in M$ is an indicator function and

$$H_i(x, y) = \begin{cases} 1 & \text{if } x < a_i < x + y \\ 0 & \text{otherwise} \end{cases}$$

Similarly as discussed in Cai et al. [17], we can transfer this manufacturer game problem into the job game (N_J, v_J), where $N_j = \cup_{m\in M} N_m$ is the grand coalition of all the jobs and $v_J(N_J) = \min \sum\limits_{m\in M}\sum\limits_{j\in N_m} TEC_m^j$. If the cost allocated to job j in N is x_j, then the cost allocated to manufacturer m is just $\sum\limits_{j\in N_m} x_j$. Hence in the rest of this chapter, we just consider the allocation scheme in the job game (N_J, v_J). We will show that the game (N_J, v_J) is balanced and a core allocation vector can be found. Note that in Aydinliyim et al. [2] and Cai et al. [17], similar manufacturer(job) game problems were discussed. But in their models, a major assumption is that all the coalitions must be "contiguous". More specifically, in their models the original booking order of the manufacturers is σ and without loss of generality we assume $\sigma(i) = i, i \in M$. For any coalition $S \subseteq M$, if $i, j \in S$ and $i < k < j$, then $k \in S$. In other words, "queue jumping" in their models is not allowed. But in our model, this restriction is relaxed. We allow different manufacturers to cooperate with each other even if they are not "contiguous". The only constraint for coalition S is that the original booking time of manufacturers who are outside S can not be occupied or affected. Our approach is inspired from the techniques applied in Slikker et al. [45], where he discussed a sequencing game with fixed job processing time. We extend his approach into a stochastic processing time case.

For the grand coalition N_J, suppose that the original booking strategy is σ_0 and $\mathbf{T}^0 = (T_1^0, \cdots, T_{|N_J|}^0)$. Here without loss of generality we assume the processing time of the jth job in N_J is p_j, the completion time cost index is h_j and the penalty cost index is ρ_j. Hence the expected processing time of job j condition on the upper bound T_j^0 is

$$P_j^0 = \int_0^{T_j^0} x dF_j(x) + \int_{T_j^0}^{\infty} T_j^0 dF_j(x), j \in N_J$$

Furthermore, define $\mathbf{P}^0 = (P_1^0, \cdots, P_{|N_J|}^0)$, $\mathbf{h} = (h_1, \cdots, h_{|N_J|})$ and $\rho = (\rho_1, \cdots, \rho_{|N_J|})$. After solving the minimization problem $v_J(N_J)$ using the algorithm provided in the above section, we can obtain the optimal sequence σ^* and booking quantity $\mathbf{T}^* = (T_1^*, \cdots, T_{|N_J|}^*)$. Similarly we can define the expected processing time of job j condition on the upper bound T_j^* as $\mathbf{P}^* = (P_1^*, \cdots, P_{|N_J|}^*)$ where

$$P_j^* = \int_0^{T_j^*} x dF_j(x) + \int_{T_j^*}^{\infty} T_j^* dF_j(x), j \in N_J$$

Let $\mathcal{S}$ denote the set of sequencing situations and $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h}) \in \mathcal{S}$ be a sequencing situation. We define the associated sequencing situation $(\overline{N}_J, \overline{\sigma_0}, \overline{\mathbf{P}}^*, \overline{\mathbf{h}})$ by

- $\overline{N}_J = N^1 \cup N^2 \cup \cdots \cup N^{|N_J|}$ with $N^i = \{k_i^1, \cdots, k_i^{P_i^*}\}, \forall i \in N_J$

- $\overline{\sigma_0}(k_i^j) = j + \sum_{r=1}^{i-1} P^*_{(\sigma_0)^{-1}(r)} \forall i \in N_J, j \in \{1, ..., P_i^*\}$
- $\overline{P}_i^j = 1, \forall i \in N_J, j \in \{1, ..., P_i^*\}$
- $\overline{h}_i^j = \frac{h_i}{P_i^*}, \forall i \in N_J, j \in \{1, ..., P_i^*\}$

In fact in such associated sequencing situation, we split the original jobs into a number of unit-jobs which have unitary processing time. And the weight (completion time cost index) $\mathbf{h}$ is split over the unit-jobs as well. The new unit-job sequence $\overline{\sigma_0}$ respects the original job sequence σ_0 in the sense that all the unit-jobs that belong to one job will be processed during the same time span as the original job. But since all the unit-jobs of a job are the same, the intra-order of these unit-jobs is not important.

Remark:

- Note that P_i^* may not be integer and hence our statement on $j \in \{1, ..., P_i^*\}$ is not well defined. In fact, the idea here is to split the original job into numbers of sub-jobs which have equal unit processing times. If the smallest time unit in our model is considered as t_u, then $\frac{P_i^*}{t_u}$ must be integer. Hence we can split the original job into $\frac{P_i^*}{t_u}$ pieces, each of which has a processing time t_u. Then the rest of the analysis will be the same. In our model for convenience, we just consider 1 as the smallest time unit. Hence P_i^* is always an integer.

Since $(\overline{N}_J, \overline{\sigma_0}, \overline{\mathbf{P}}^*, \overline{\mathbf{h}})$ is a sequencing situation, we can further consider its associated sequencing game, which we denote as $(\overline{N}_J, w_J)$. The characteristic function $w_J(\overline{S})$ for any coalition $\overline{S} \subset \overline{N}_J$ is stated as follows

$$w_J(\overline{S}) = \min_{\sigma_{\overline{S}}} \sum_{k_i^j \in \overline{S}} \overline{h}_i^j C_{k_i^j}$$

where $C_{k_i^j}$ is the completion time of the unit-job k_i^j.

Since all processing times in $(\overline{N}_J, \overline{\sigma_0}, \overline{\mathbf{P}}^*, \overline{\mathbf{h}})$ are equal (to 1), the associated sequencing game coincides with a permutation game, which is balanced as already argued by Tijs et al. [46]. Therefore we state the following lemma without a proof.

Lemma 4.4. *Let $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h}) \in \mathcal{S}$ be a sequencing situation. Then its associated unit sequencing game $(\overline{N}_J, w_J)$ is balanced.*

Lemma 4.4 shows the balancedness of the associated sequencing game $(\overline{N}_J, w)$. Next we will derive the relation between the original sequencing game and the associated one. We conclude the result in Lemma 4.5. Note that based on the relation we derive in Lemma 4.5 and the balancedeness of the associated sequencing game, we can further prove the balancedness of the original sequencing game $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h}) \in \mathcal{S}$.

Lemma 4.5. *Let $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h}) \in \mathcal{S}$ be a sequencing situation and $S \in N_J$ an arbitrary sub-coalition. Denote $\overline{S}$ as the associated unit-job set of S. Then $v_J(S) - \sum_{i\in S} \Delta_i \geq w(\overline{S})$ where*

$$\Delta_i = \rho_i \int_{T_i^*}^{\infty} (x - T_i^*) dF_i(x) + \frac{hT_i^*}{L} + h_i P_i^* - \sum_{j=1}^{P_i^*} j\overline{h}_i^j$$

And the equality holds if and only if $S = N_J$.

Proof. Let $\sigma \in \Pi^{N_J}$ be an sequence of the jobs, then for any job $i \in N_J$ we have

$$\begin{aligned} TEC_i(\sigma) &= \rho_i \textstyle\int_{T_i^*}^{\infty} (x - T_i^*) dF_i(x) + \frac{hT_i^*}{L} + h_i \sum_{k\ \sigma(k)\leq \sigma i} P_k^* \\ &= \rho_i \textstyle\int_{T_i^*}^{\infty} (x - T_i^*) dF_i(x) + \frac{hT_i^*}{L} + h_i P_i^* + h_i \sum_{k\ \sigma(k)<\sigma i} P_k^* \\ &= \Delta_i + \sum_{j=1}^{P_i^*} j\overline{h}_i^j + \sum_{k=1}^{P_i^*} \left(\frac{h_i}{P_i^*} \sum_{k\ \sigma(k)<\sigma i} P_k^*\right) \\ &= \Delta_i + \sum_{j=1}^{P_i^*} C_{k_i^j}(\overline{\sigma}) \end{aligned}$$

Therefore we know for all $S \subseteq N_J$

$$TEC_S(\sigma) = \sum_{k_i^j \in \overline{S}} \overline{h}_i^j C_{k_i^j}(\overline{\sigma}) + \sum_{i\in S} \Delta_i = TEC_{\overline{S}}(\overline{\sigma}) + \sum_{i\in S} \Delta_i$$

If the optimal sequence for coalition S is σ_S and its associated unit-job sequence is $\overline{\sigma_S}$, then we know $\overline{\sigma_S}$ is admissible for $\overline{S}$. With the optimal sequence $\sigma_{\overline{S}}$ for $\overline{S}$, we have

$$\begin{aligned} w_J(\overline{S}) &= TEC_{\overline{S}}(\sigma_{\overline{S}}) \\ &\leq TEC_{\overline{S}}(\overline{\sigma_S}) \\ &= TEC_S(\sigma_S) - \sum_{i\in S} \Delta_i \\ &= v_J(S) - \sum_{i\in S} \Delta_i \end{aligned}$$

Finally consider the grand coalition N_J. Since the optimal sequence is stated as σ^*, we can define the associated unit-job sequence as $\overline{\sigma^*}$. For any job i_1 and $i_2 \in N_J$, we have the associated unit-jobs as $k_{i_1}^{j_1}, k_{i_2}^{j_2} \in \overline{N}_J$, $j_1 = 1, \cdots, P_{i_1}^*, j_2 = 1, \cdots, P_{i_2}^*$. If $(\sigma^*)^{-1}(i_1) < (\sigma^*)^{-1}(i_2)$, or equivalently $(\overline{\sigma^*})^{-1}(k_{i_1}^{j_1}) < (\overline{\sigma^*})^{-1}(k_{i_2}^{j_2})$ then

$$\frac{\overline{h}_{i_1}^{j_1}}{\overline{P}_{i_1}^{j_1}} = \overline{h}_{i_1}^{j_1} = \frac{h_{i_1}}{P_{i_1}^*} \leq \frac{h_{i_2}}{P_{i_2}^*} = \overline{h}_{i_2}^{j_2} = \frac{\overline{h}_{i_2}^{j_2}}{\overline{P}_{i_2}^{j_2}} \tag{4.17}$$

where the inequality follows by the optimality of σ^* stated in Lemma 4.1. Meanwhile in the optimal sequence of $\overline{N}_J$, we can verify that inequality (4.17) should also be satisfied if unit-job $k_{i_1}^{j_1}$ is processed before unit-job $k_{i_2}^{j_2}$. Hence, $\overline{\sigma^*}$ is the optimal sequence for $\overline{N}_J$, which leads to

$$\begin{aligned} w_J(\overline{N}_J) &= TEC_{\overline{N}_J}(\overline{\sigma^*}) \\ &= TEC_{N_J}(\sigma^*) - \sum_{i\in N_J} \Delta_i \\ &= v_J(N_J) - \sum_{i\in N_J} \Delta_i \end{aligned}$$

This completes the proof. □

Combining Lemma 4.4 and Lemma 4.5, we can prove the first main result in this chapter.

Theorem 4.6. *Let $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h}) \in \mathcal{S}$ be a sequencing situation, then its associated sequencing game (N_J, v_J) is balanced.*

Proof. According to Lemma 4.4, we know the unit-job sequencing game $(\overline{N}_J, w_J)$ is balanced. Let $(y_i^j)_{i\in N_J, 1\leq j\leq P_i^*}$ be a core allocation vector of $(\overline{N}_J, w_J)$. Then let

$$x_i = \sum_{j=1}^{P_i^*} y_i^j + \Delta_i$$

where Δ_i was introduced in Lemma 4.5. According to 4.5 we have

$$\sum_{i\in S} x_i = \sum_{i\in S}(\sum_{j=1}^{P_i^*} y_i^j + \Delta_i) \leq \sum_{i\in S} \Delta_i + w_J(\overline{S}) \leq v_J(S), \forall S \subseteq N_J$$

and

$$\sum_{i\in N_J} x_i = \sum_{i\in N_J}(\sum_{j=1}^{P_i^*} y_i^j + \Delta_i) = \sum_{i\in S} \Delta_i + w_J(\overline{N}_J) = v_J(N_J)$$

Therefore $(x_i)_{i\in N_J}$ is a core allocation vector and the sequencing game (N_J, v_J) is balanced. □

Theorem 4.6 actually shows that when the booking time reserved for processing each job is fixed, then associated sequencing game will be balanced. But from Lemma 4.2 and Corollary 4.3 we know the optimal time slots reserved for processing a job is affected by its successors' schedule. Hence the optimal booking time for this job will be different when it belongs to different coalition with different successors. Specifically, we should model our job game as a cooperative game (N_J, u) where N_J is the grand coalition consists of all jobs and $u(N_J)$ is the characteristic function. For any coalition $S \in N_J$, we have

$$u(S) = \min_{\mathbf{T}_S, \sigma_S} \sum_{i\in S} TEC_i(\mathbf{T}_S, \sigma_S), \forall S \subseteq N_J$$

where $\mathbf{T}_S$ is the time slots reserved for processing jobs in S and σ_S is the sequence to follow. The differences between $u(S)$ and $v_J(S)$ include 1) the latter one does not take booking time optimization into consideration; 2) the processing times of jobs outside the coalition S are different in the two cases, which also affect the values of $u(S)$ and $v_J(S)$. But we can also easily verify that $u(N_J) = v_J(N_J)$.

Since we are interested in the balancedness of the job game (N_J, w) while we already proved the balancedness of job sequencing game (N_J, v_J), a natural idea is to utilize the relation between $u(S)$ and $v_J(S)$. We next introduce a simple example, which can help us to better understand the relation between these two characteristic functions.

Example 4.2. Suppose now we have 3 manufacturers each of whom has one job to be processed by third party P. Hence we have $M = N_J = \{1,2,3\}$. Originally the manufacturers' booking strategies are $\mathbf{T} = (T_1, T_2, T_3)$ and the expected processing times are (p_1, p_2, p_3). The initial sequence $\sigma_0 = \{1,2,3\}$. Suppose now with cooperation, the optimal strategy of the grand coalition is $\sigma^* = (2,1,3)$ and $\mathbf{T}^* = (T_1^*, T_2^*, T_3^*)$. Therefore the expected processing times of all the jobs can be calculated and we denote them as $\mathbf{P} = (P_1^*, P_2^*, P_3^*)$. According to Theorem 4.6, we know for a given sequencing situation $(N_J, \sigma_0, \mathbf{P}^*, \mathbf{h})$, the associated sequencing game (N_J, v_J) is balanced. Let (x_1, x_2, x_3) a core allocation vector of this game and we have

$$\begin{cases} x_1 \leq h_1 P_1^* + \rho_1(T_1^*);\ x_2 \leq h_2(P_1^* + P_2^*) + \rho_2(T_2^*);\ x_3 \leq h_3(P_1^* + P_2^* + P_3^*) + \rho_3(T_3^*); \\ x_1 + x_2 \leq h_1(P_1^* + P_2^*) + h_2 P_2^* + \rho_1(T_1^*) + \rho_2(T_2^*); \\ x_1 + x_3 \leq h_1 P_1^* + h_3(P_1^* + P_2^* + P_3^*) + \rho_1(T_1^*) + \rho_3(T_3^*); \\ x_2 + x_3 \leq h_2(P_1^* + P_2^*) + h_3(P_1^* + P_2^* + P_3^*) + \rho_2(T_2^*) + \rho_3(T_3^*); \\ x_1 + x_2 + x_3 = h_1(P_1^* + P_2^*) + h_2 P_2^* + h_3(P_1^* + P_2^* + P_3^*) + \rho_1(T_1^*) + \rho_2(T_2^*) + \rho_3(T_3^*). \end{cases}$$

where $\rho_i(T_i^*)$ is defined as $\rho_i \int_{T_i^*}^{\infty}(x - T_i^*)dF_i^*(x) + \frac{h}{L}T_i^*$. Meanwhile if we want to show that our job game (N_J, u) is balanced, then we have to find a core allocation vector $(\lambda_1, \lambda_2, \lambda_3)$ which satisfies

$$\begin{cases} \lambda_1 \leq h_1 P_1 + \rho_1(T_1);\ \lambda_2 \leq h_2(P_1 + P_2) + \rho_2(T_2);\ \lambda_3 \leq h_3(P_1 + P_2 + P_3) + \rho_3(T_3) \\ \lambda_1 + \lambda_2 \leq u(\{1,2\});\ \lambda_2 + \lambda_3 \leq u(\{2,3\});\ \lambda_1 + \lambda_3 \leq u(\{1,3\}) \\ \lambda_1 + \lambda_2 + \lambda_3 = x_1 + x_2 + x_3 \end{cases}$$

According to the last equation, a nature idea to construct $(\lambda_1, \lambda_2, \lambda_3)$ is to redistribute the cost basing on (x_1, x_2, x_3). Specifically, denote $\delta_i j$ as the cost that job i throws to job j, then we have, for example to job 1, $\lambda_1 = x_1 + \delta_{21} + \delta_{31}$. In this example we let

$$\begin{cases} \delta_{21} = -x_1 + h_1 P_1^* + \rho_1(T_1^*) - \varepsilon(\{1,3\}) \\ \delta_{31} = h_3(P_1^* - P_1) \\ \delta_{32} = h_3(P_2^* - P_2) - \Delta u(\{1,3\}) \end{cases}$$

where $\varepsilon(\{1,3\}) = (h_1 + h_3)(P_1^* - P_1) + \rho_1(T_1^*) - \rho_1(T_1) \leq 0$ and $\Delta u(\{1,3\}) = u(\{1,3\}) - u(\{1\}) - u(\{3\}) \leq 0$. Note that δ_{21} can be rewritten as

$$\delta_{21} = -x_1 - \delta_{31} + u(\{1\})$$

Then we know

$$\begin{aligned} \lambda_1 = {} & x_1 + \delta_{21} + \delta_{31} = u(\{1\}) \leq h_1 P_1 + \rho_1(T_1) \\ \lambda_3 = {} & x_3 - \delta_{31} - \delta_{32} \leq h_3(P_1 + P_2 + P_3^*) + \rho_3(T_3^*) + \Delta u(\{1,3\}) \leq h_3(P_1 + P_2 + P_3) + \rho_3(T_3) \\ \lambda_2 = {} & x_2 + \delta_{32} - \delta_{21} = x_2 + x_1 + h_3(P_1^* + P_2^*) - h_1 P_1 - \rho_1(T_1) - h_3 P_1 - h_3 P_2 - \Delta u\{1,3\}) \\ \leq {} & h_1(P_1^* + P_2^*) + h_2 P_2^* + \rho_1(T_1^*) + \rho_2(T_2^*) + h_3(P_1^* + P_2^* + P_3^*) + \rho_3(T_3^*) \\ & -h_1 P_1 - \rho_1(T_1) - h_3(P_1 + P_2 + P_3) - \rho_3(T_3) - \Delta u(\{1,3\}) \\ = {} & \Delta u(\{1,2,3\}) - \Delta u(\{1,3\}) + h_2(P_1 + P_2) + \rho_2(T_2) \\ \leq {} & h_2(P_1 + P_2) + \rho_2(T_2) \end{aligned}$$

where the last equality of λ_3 holds because $P_3^* = P_3$ (recall Lemma 4.2)and the last inequality of λ_2 holds since the cost saving generated by 3 jobs will always be greater than the saving generated by 2 jobs. Furthermore, we can check

$$
\begin{aligned}
\lambda_1 + \lambda_2 = \; & x_1 + x_2 + h_3(P_1^* + P_2^*) - h_3(P_1 + P_2) \\
& = x_1 + x_2 + h_3(P_1^* + P_2^* + P_3^*) - h_3(P_1 + P_2 + P_3) + \rho_3(T_3^*) - \rho_3(T_3) \\
& \leq u(\{1,2,3\}) - u(\{3\}) \\
& \leq u(\{1,2\}) \\
\lambda_1 + \lambda_3 = \; & x_1 + x_3 + \delta_{21} - \delta_{32} \\
& \leq \Delta u(\{1,3\}) + u(\{1\}) + h_3(P_1 + P_2 + P_3^*) + \rho_3(T_3^*) \\
& = u(\{1,3\}) - u(\{3\}) + u(\{3\}) \\
& = u(\{1,3\}) \\
\lambda_2 + \lambda_3 = \; & x_2 + x_3 - \delta_{21} - \delta_{31} \\
& = x_1 + x_2 + x_3 - u(\{1\}) \\
& = u(\{1,2,3\}) - u(\{1\}) \\
& \leq u(\{2,3\})
\end{aligned}
$$

And obviously we can verify that

$$\lambda_1 + \lambda_2 + \lambda_3 = x_1 + x_2 + x_3 = v_J(\{1,2,3\}) = u(\{1,2,3\})$$

Hence, we constructed a core allocation vector $(\lambda_1, \lambda_2, \lambda_3)$ based on (x_1, x_2, x_3).

According to the example above, we can derive the relation between $u(S)$ and $v_J(S)$. The result is concluded in Lemma 4.7. For notational convenience, we again use $Sr^i(\sigma)$ to denote the set of succeeding jobs of job i in a given sequence σ. Similarly, denote $Pr^i(\sigma)$ a set of preceding jobs of job i in sequence σ. That is $Pr^i(\sigma) = \{k : (\sigma)^{-1}(k) < (\sigma)^{-1}(i)\}$. Without loss of generality, in the rest of this chapter we assume the initial sequence σ_0 to satisfy the condition $\sigma_0(k) = k, \forall k \in N_J$.

Lemma 4.7. *Define* $(\delta_{ij})_{i,j\in N_J, j<i}$ *as follows: for all* $i \in N_J$

$$\delta_{ij} = \begin{cases} h_i(P_j^* - P_j^0) & \text{If } j \in Pr^i(\sigma^*) \setminus \tilde{S}; \\ h_i(P_j^* - P_j^0) - \Delta u(\{j,i\}) & \text{If } j \in Pr^i(\sigma^*) \cap \tilde{S}; \\ -x_j + h_j P_j + \rho_j(T_j) - \sum\limits_{k\in Sr^j(\sigma^*), k>j} \delta_{kj} & \text{If } (\sigma^*)^{-1}(j) = (\sigma^*)^{-1}(i) + 1; \\ 0 & \text{Otherwise.} \end{cases} \tag{4.18}$$

where σ^* *is the optimal sequence in the grand coalition and* $\tilde{S} = \{j | \exists i > j, s.t. i \in Pr^j(\sigma^*)\}$. *Then* $\sum\limits_{i\in S} x_i + \sum\limits_{i\in S}\{\sum\limits_{j>i}\delta_{ji} - \sum\limits_{j<i}\delta_{ij}\} \le u(S)$ *for all* $S \subset N_J$. *Especially when* $S = N_J$, $\sum\limits_{i\in N_J} x_i + \sum\limits_{i\in N_J}\{\sum\limits_{j>i}\delta_{ji} - \sum\limits_{j<i}\delta_{ij}\} = u(N_J)$.

Proof. Let σ_S be the optimal sequence of coalition S in job game problem (N_J, u). Hence it is an admissible sequence of coalition S in sequencing game problem (N_J, v_J). Note that in problem (N_J, v_J), the booking time for job j is fixed as T_j^* while in problem (N_J, u), the booking time is T_j^S if $j \in S$, or T_j^0 if $j \notin S$. We can first prove the following inequality

$$\begin{aligned} & \sum_{i\in S} x_i + \sum_{i\in S}\{\sum_{j>i}\delta_{ji} - \sum_{j<i}\delta_{ij}\} \\ = \; & \sum_{i\in S} x_i + \sum_{i\in S}\{\sum_{j>i, j\notin S}\delta_{ji} - \sum_{j<i, j\notin S}\delta_{ij}\} \\ = \; & \sum_{i\in S\setminus S^-}[x_i + \sum_{j>i, j\in Sr^i(\sigma^*)\setminus S} h_j(P_i^* - P_i^0)] \\ & + \sum_{i\in S^-} x_i - x_j - h_j P_j^* - \rho_j^*(T_j^*) + \varepsilon(\{j\} \cup Sr^j(\sigma^*)) + \Delta u(\{j\} \cup Sr^i(\sigma^*)) \end{aligned}$$

where $S^- = \{j | j \notin S, j < i \; \exists i \in S, (\sigma^*)^{-1}(i) + 1 = (\sigma^*)^{-1}(j)\}$. Through simplification, we can further rewrite the inequality as

$$\begin{aligned} & \sum_{i\in S} x_i + \sum_{i\in S}\{\sum_{j>i}\delta_{ji} - \sum_{j<i}\delta_{ij}\} \\ \le \; & \min\{u(N_J) - \cup_k u(S_k), u(\{S\})\} \\ \le \; & \min\{u(N_J) - u(N_J \setminus S), u(\{S\})\} \\ \le \; & u(S) \end{aligned}$$

where $\cup_k S_k = N_J \setminus S$. When $S = N_J$, we have

$$\begin{aligned} & \sum_{i \in N_J} x_i + \sum_{i \in N_J} \{\sum_{j>i} \delta_{ji} - \sum_{j<i} \delta_{ij}\} \\ = & \sum_{i \in N_J} x_i \\ = & v_J(N_J) \\ = & u(N_J) \end{aligned}$$

The first equality holds due to the symmetric relation. Hence we proved the lemma. □

Based on the Lemma 4.7, we can derive the second main result in this chapter which states that our job game (N_J, u) is balanced.

Theorem 4.8. *The job game (N_J, u) is balanced. Let $z_i = x_i + \sum_{j>i} \delta_{ji} - \sum_{j<i} \delta_{ij}$ for all $i \in N_J$ where $(x_i)_{i \in N_J}$ is the core allocation vector defined in Theorem 4.6 and δ_i is given by equation 4.18. Then allocation vector $(z_i)_{N_J}$ lies in the core of (N_J, u).*

Proof. For any coalition $S \subseteq N_J$, we have

$$\sum_{i \in S} z_i = \sum_{i \in S} x_i + \sum_{i \in S} \delta_i \leq u(S)$$

where the inequality follows by Lemma 4.7. And when $S = N_J$, we have

$$\sum_{i \in N_J} z_i = \sum_{i \in N_J} x_i + \sum_{i \in N_J} \delta_i = v_J(N_J) = u(N_J)$$

Hence $(z_i)_{i \in N_J}$ is a core allocation vector of job game (N_J, u). And the job game (N_J, u) is balanced. □

Theorem 4.8 shows that the job game is balanced and provides a core allocation vector $(z_i)_{i \in N_J}$. Recall the relation between the manufacturer game (M, v) and the job game N_J, u, we can easily prove the next theorem which states that the the manufacturer game (M, v) is also balanced. Hence the grand coalition will always be formed in which all the manufacturers can benefit.

Theorem 4.9. *The manufacturer game (M, v) is balanced.*

Proof. Let $\lambda_m = \sum_{i \in N_m} z_i$, then for any coalition $T \subseteq M$ we can prove

$$\sum_{m \in T} \lambda_m = \sum_{m \in T} \sum_{i \in N_m} z_i \leq u(\cup_{m \in T} N_m) = v(T)$$

and

$$\sum_{m \in M} \lambda_m = \sum_{m \in M} \sum_{i \in N_m} z_i = u(N_J) = v(M)$$

Hence λ_i consists of a core allocation and the manufacturer game (M, v) is balanced. □

Theorem 4.9 is straightforward from the discussion in the early part of this chapter. Therefore we just give it without proof.

4.4. Numerical study

In this section, we will construct some numerical experiments to examine the benefit of cooperation. Basically we will compare the manufacturers' total expected cost without collaborating with each other and their total expected cost if he grand coalition is formed. That is to compare $\sum_{m \in M} ETC_m$ and ETC_M. Meanwhile, we will compare different parameter settings so that we can also observe the impacts of these factors. These impacts can not be obviously derived from our theoretical results but using the numerical experiments, we can gain some insights. Generally the following parameters will be examined:

- The different completion time evaluation index h_m^j, $\forall (m, j) : m \in M, j \in N_m$;
- The different unit penalty cost ρ_m^j,$\forall (m, j) : m \in M, j \in N_m$;
- The fluctuation of the numbers of manufacturers $|M|$ and the numbers of jobs they own $|N_m|, \forall m \in M$;

Before we show our numerical experiments, we first introduce some basic parameter settings. We will consider a basic model which includes 10 manufactures, each of whom owns 20 jobs to be processed by the third party. We assume all the jobs follow a same uniform distribution $[(1-r)P, (1+r)P]$. Hence the mean processing time is P and the range is $2rP$. The basic value of P is set to be 4. Meanwhile, we let the window booking cost $h = 10$ and the unit window length $L = 10$ which implies that the unit booking cost for a time slot is 1. Furthermore we let all the h_m^j follow a uniform distribution $U[0.5, 1.5]$. Note that the mean of the evaluation index h_m^j is also 1 and recall equation (4.9) in the second remark of section 4.3.1, we let the unit penalty cost satisfy $\rho_m^j \geq \sum_{n \in N_m, m \in M} h_m^j + h/L$. More specially in this section, we make an important assumption that all the unit penalty costs are equal to each other. The reason is that under such a situation, we can prove that the optimal sequence of the jobs will follow a descending order of h_m^j (see Appendix I). In this way, we can speed up our algorithm to find the optimal booking and sequencing solution. Hence we let the basic value of ρ_m^j be

$$\rho_m^j = \sum_{n \in N_m, m \in M} E[h_m^j] + h/L + 1 = 2 * |M| * |N| + 2$$

Note that in such a setting, jobs have no chance to be removed from the processing line without any production because we can verify that the inequality (4.9) will be satisfied with such settings. And later we will vary the value of $E[h_m^j]$ and ρ_m^j. But we will still make sure that the inequality (4.9) will be satisfied.

We first check the impact of different $E[h_m^j]$ to the relation between total expected cost with cooperation and without cooperation. We will vary $E[h_m^j]$ from 0.6 to 1.5 but the range of h_m^j will be fixed as 1. That is the distribution of h_m^j will change from $[0.1, 1.1]$ to $[1, 2]$. The following Table 4.1 summarizes the result of our comparison. The first column denotes the changing indexes $E[h_m^j]$. Column 2 to 4 include the cost information of the manufacturers in both cooperation situation and non-cooperated situation. The benefits of cooperation are listed in column 4. And the last 3 columns summarize the total booking cost information of all the manufacturers. We also compare the costs in the two situations and show the improvements in the last column.

Table 4.1: Comparison under varied completion time evaluation indexes

$E[h_m^j]$	Total Expected Cost			Total Booking Cost		
	$\sum_{m\in M} TEC_m$	TEC_M	Improvement (%)	$\sum_{m\in M} BC_m$	BC_M	Improvement (%)
0.6	47126	35177	25.36	299	299	0.11
0.7	55343	43416	21.55	299	299	0.16
0.8	63450	51434	18.94	299	298	0.23
0.9	71354	59377	16.78	299	298	0.36
1	79565	67515	15.14	299	295	1.22
1.1	87212	75223	13.74	299	283	5.19
1.2	94333	82555	12.48	299	272	9.03
1.3	101010	89586	11.3	299	261	12.58
1.4	107297	96085	10.45	299	252	15.73
1.5	113272	102524	9.49	299	243	18.7

From Table 4.1, we have the following observation:

- As $E[h_m^j]$ increases, the total expected cost under both situations become larger since the completion time cost increases. However, the booking cost reduces, since some jobs are not worthy to be reserved with long processing time which will delay the successors' processing.

- The expected total cost is smaller in the cooperation situation. Basically the benefit is around 20% when the unit completion time cost is smaller and it moves to around 10% when the unit completion cost is larger. The reason is that when $E[h_m^j]$ becomes larger, manufacturers will consider reducing the booking quantities. They would rather pay the penalty fees in case the jobs are not completed within the reserved time window to avoid huge completion time cost. Hence the benefit is reduced by the increase of penalty fees.

- The quantity of time window reserved in the cooperation situation is smaller than that in the non-cooperated situation. The reason is similar as what is discussed above, that is the manufacturers will book fewer time window to process their jobs so that the completion times of the jobs can be reduced, although sometimes penalty fees are required.

Next we look at the fluctuation of the unit penalty cost, ρ^j_m. As spoken, we have fixed the value of ρ^j_m is our numerical experiments and the basic value is set as $(\rho^j_m)_0 = \sum_{n\in N_m, m\in M} E[h^j_m] + h/L + 1 = 1.6 * |M| * |N| + 2$. Hence we next change it from $(\rho^j_m)_0$ to $(\rho^j_m)_0 + 90$ by adding 10 each time. And in order to make the value of ρ^j_m smaller, we let $E[h^j_m] = 0.6$. The result is concluded in Table 4.2:

Table 4.2: Comparison under varied unit penalty cost on unfinished jobs

ρ^j_m	Total Expected Cost			Total Booking Cost		
	$\sum_{m\in M} TEC_m$	TEC_M	Improvement (%)	$\sum_{m\in M} BC_m$	BC_M	Improvement (%)
211	46923	34961	25.5	300	300	0.16
221	50177	38123	24.02	291	288	0.89
231	53276	41273	22.53	283	279	1.45
241	56876	44814	21.21	277	272	1.83
251	60501	48312	20.14	271	265	2.07
261	64119	51976	18.93	266	260	2.24
271	67743	55597	17.92	262	256	2.34
281	71526	59366	16.99	258	252	2.4
291	75261	63027	16.25	255	249	2.43
301	78941	66746	15.44	252	246	2.44

From Table 4.2, we can conclude that

- As ρ^j_m becomes larger, the expected total cost of manufacturers increase under both the situations which is easy to understand. Furthermore, the benefit of cooperation becomes less since time windows reserved for processing jobs can not be reduced too much(Otherwise, it is too risky to suffer a heavy penalty fee). Therefore, the manufacturers can only earn benefit from optimal re-sequencing of their jobs when ρ^j_m increases. This is also the reason why we can observe that the booking cost improvement almost remains the same level. Because the penalty cost already stays at a high level compared with the h^j_m and h. The booking times are close to the upper bounds of the processing times even in the non-cooperated situation, Hence, under the cooperation situation, windows reserved for production should not be reduced too much. Otherwise the penalty fees of unfinished jobs will cost the manufacturers a lot.

Finally we want to check the impact of different numbers of manufacturers as well as the impact of different numbers of jobs. In Table 4.3 we first change the number of manufacturers from 5 to 50 while the number of jobs that each manufacturer owns is fixed as 20. And then we use Table 4.4 to represent the impact of number of jobs. Generally we fix the number of manufacturers as 10 and then very the number of jobs that each manufacturer owns from 10 to 100. From these two tables we can have the following observations:

Table 4.3: Comparison under varied number of manufacturers

$\|M\|$	Total Expected Cost			Total Booking Cost		
	$\sum_{m \in M} TEC_m$	TEC_M	Improvement (%)	$\sum_{m \in M} BC_m$	BC_M	Improvement (%)
5	30036	27175	9.52	102	96	5.5
10	81042	68640	15.3	237	225	4.9
15	155389	126851	18.36	380	364	4.12
20	252928	201635	20.28	526	508	3.49
25	374669	293933	21.55	673	653	3.03
30	520589	403929	22.41	821	799	2.67
40	882477	673703	23.66	1119	1095	2.14
50	1341491	1013675	24.44	1417	1392	1.79

- From the above two tables we can clearly see that the benefit of cooperation on the total expected cost will become larger when there are more manufacturers or more jobs. However, the booking cost reduction becomes less when the grand coalition becomes greater. In other words, when there are plenty of manufacturers (jobs), the benefit of the cooperation is generally resulted from optimal re-sequencing rather than booking time optimization.

- Compare the corresponding rows of two tables, we can find that in each corresponding row, the total numbers of jobs owned by the manufacturers are the same. Hence we can see that when the number of jobs are the same, if these jobs belongs to more manufacturers, then the cooperation can bring more benefit. This observation is consistent with common sense.

Table 4.4: Comparison under varied number of jobs owned by each manufacturer

$\|N\|$	Total Expected Cost			Total Booking Cost		
	$\sum_{m\in M} TEC_m$	TEC_M	Improvement (%)	$\sum_{m\in M} BC_m$	BC_M	Improvement (%)
10	30378	27152	10.61	102	96	6.01
20	81149	68756	15.27	237	225	4.92
30	154230	126660	17.87	380	364	4.01
40	250187	201524	19.45	525	508	3.36
50	369404	293621	20.51	672	653	2.89
60	511232	402300	21.31	820	800	2.53
80	866413	673584	22.26	1118	1095	2.03
100	1312946	1012494	22.88	1416	1392	1.69

4.5. Conclusions and Future work

In this chapter we model a manufacturing outsourcing problem as a cooperative game where the players are manufacturers who own numbers of jobs to be outsourced. Unlike the traditional sequencing game problems, we bring stochastic job processing time into our model. Therefore, in addition to job sequencing issue we also need to consider the booking quantity problem, which makes the scheduling problem much more complicated. We have proved that when the unit booking cost for each time slot is unique, the manufacturer game (M, v) is balanced and a core allocation vector can be constructed based on the core vector obtained in the permutation game. When the unit booking costs for time windows are not unique, the booking problem for the manufacturers becomes more complicated. If they book the earliest time windows to process their jobs, then they can reduce the completion time costs but meanwhile their booking costs increase since some of these windows may be peak ones. But if they skip these peak windows and book later regular time windows to process their jobs, higher completion time costs will probably be generated. Therefore, the key point for each manufacture is to find a critical time slot, after which only regular time windows will be booked for processing jobs. An interesting question is that after the manufacturers' booking strategies are determined, whether we can apply the similar approaches in the above section to prove the balancedness of the manufacturer game? And whether the third-party P can act as the coordinator of this game so that peak time slots that have not been reserved can also be utilized, with appropriate charge, to process the jobs.

As mentioned, there is another mechanism to handle the unfinished jobs within reserved time slots, that is "spot purchasing". With such scheme, the model will be different from the one we studied in this chapter. New issues on overtime production and rescheduling of the regular production should be considered. A natural idea is to transfer this "spot purchasing" model to the "penalty" model we studied in this chapter. And it is interesting to investigate whether a stationary relation between the cost functions in these two models can be derived.

4.6. Appendix

[1]. **Details of the optimal sequencing rule**

In the numerical study section above, we assume the jobs' processing times follows a uniform distribution $[(1-r)P,(1+r)P]$ and they all have the same unit penalty cost $\rho^j_m = \rho, \forall j \in N_m, m \in M$. We claimed that with such settings, the optimal sequence of the jobs will just follow a descending order of the evaluation indexes $h^j_m, \forall j \in N_m, m \in M$. We next show the detail proof of this proposition.

For state convenience, in the rest of this part, we just refer h^j_m as h_j. Suppose the optimal sequence is σ^* and without generality we label these jobs from 1 to $|N|$ according their position in σ^*. Suppose job i is the last job in the sequence that does satisfy $h_i < h_j, j = i+1$. Just change the position of this two jobs and let σ be the new sequence obtained. Then we want to compare the expected total cost with sequence σ^* and sequence σ.

According to Lemma 4.2, we know once the sequence is fixed, the optimal booking times for each job will be determined too. Compare σ and σ^* we can easily verify that

$$T_k^*(\sigma) = T_k^*(\sigma^*), \forall k \neq i, j$$

Recall the definition of $P_m^j(T_m^j)$ in Section 4.3.1, we denote

$$P_j(\sigma^*) = \frac{\int_0^{T_j^*(\sigma^*)} x dF_j(x) + \int_{T_j^*(\sigma^*)}^{\infty} T_j^*(\sigma^*) dF_j(x)}{h_j}$$

Since we assume the jobs follows uniform distribution on $[(1-r)P, (1+r)P]$, we can calculate the optimal booking time $T_i^*(\sigma^*)$ for job i according to Lemma 4.2. Suppose that the set of successors of job j in σ^* is S and $h_s = \sum_{k \in S} h_k$. Then we can prove

$$T_i^*(\sigma^*) = 1-r+2*r\frac{\rho - h_i - h_j - h_s - h/L}{\rho - h_i - h_j - h_s}, T_j^*(\sigma^*) = 1-r+2*r\frac{\rho - h_j - h_s - h/L}{\rho - h_j - h_s}, j = i+1$$

Furthermore we can obtain the relevant $P_i(\sigma^*)$ and $P_j(\sigma^*)$ as follows

$$P_i(\sigma^*) = \frac{P(1 - r(\frac{h}{\rho - h_i - h_j - h_s})^2)}{h_i}, P_j(\sigma^*) = \frac{P(1 - r(\frac{h}{\rho - h_j - h_s})^2)}{h_j}$$

Therefore

$$\begin{aligned}
&\sum_{m \in M} TEC_m(\sigma^*) - \sum_{m \in M} TEC_m(\sigma) \\
= \; & (h_i + h_j + h_s + h/L)P[1 - r(\tfrac{h/L}{\rho - h_i - h_j - h_s - h})^2] + (h_j + h_s + h/L)P[1 - r(\tfrac{h/L}{\rho - h_j - h_s - h})^2] \\
& + \rho r P[(\tfrac{h}{\rho - h_i - h_j - h_s - h})^2 + (\tfrac{h/L}{\rho - h_j - h_s - h})^2] \\
& - \{(h_i + h_j + h_s + h/L)P[1 - r(\tfrac{h/L}{\rho - h_i - h_j - h_s - h})^2] + (h_i + h_s + h/L)P[1 - r(\tfrac{h/L}{\rho - h_i - h_s - h})^2] \\
& + \rho r P[(\tfrac{h}{\rho - h_i - h_j - h_s - h})^2 + (\tfrac{h/L}{\rho - h_i - h_s - h})^2]\} \\
& = (h_j - h_i)P + \rho r P h^2 \tfrac{h_j - h_i}{(\rho - h_i - h_s - h/L)(\rho - h_j - h_s - h/L)} \\
& = (h_j - h_i)P(1 + \tfrac{r h^2}{(\rho - h_i - h_s - h/L)(\rho - h_j - h_s - h/L)}) \\
& > 0
\end{aligned}$$

The last inequality holds since we assume $\rho_m^j > \sum_{j \in N_m, m \in M}$ in (4.9). But this is a contradiction since $\sum_{m \in M} TEC_m(\sigma^*)$ should be the minimum cost. Hence we can claim that in the optimal sequence σ^*, the jobs follow a descending order of h_i.

CHAPTER 5

CONCLUSION

In this thesis we discussed the applications of both *Competition Game Theory* and *Cooperative Game Theory* in the *Operations Management* problems involve multiple participants. In Chapter 2 we model the retailers' ordering problem as a competition game and find the Nash equilibrium of optimal ordering times and optimal ordering quantities for all the retailers and further more show that the equilibrium is not only a strong equilibrium but also the only Pareto optimal Nash equilibrium. And we also find the optimal strategy for the supplier by taking into consideration of retailers likely action with respected to the GBS pricing mechanism. An important conclusion is that all the chain members can benefit from the GBS pricing mechanism compared with the the traditional flat price scheme. The supplier can achieve a higher profit by selling more products and the retailers, although the profit remains the same, the quantities of product received from the supplier increase, which improve their service to the customers. The difficulty in implementing such GBS pricing mechanism in the practice is that the market demand information of each retailer in our model is assumed to be public known while in practice such information should be private. Thus both the supplier and retailers should figure out ways to acquire the information. For example they can pay for the information. The retailers may also consider collaboration, so that the information can be shared automatically by retailers in the same coalition. Hence it is interesting to investigate we can model the retailers' ordering problem as a cooperative game.

In Chapter 3, we studied the project subcontracting problem by modeling it as a linear programming game. Using the duality property we proved the balancedness of our "Linear Project Game" and we also construct a core allocation vector by utilizing the optimal dual solution. And the PO's optimal strategy in designing the contracts is also discussed. The results imply that the project owner can receive all the benefit if SCs' crashing information is known or precisely estimated. And the benefit may become less when the SCs provide incorrect information on crashing. And in the end of Chapter 3, we further introduce a multi-project problem where tasks from different project owner may be subcontracted to the same SC. We model this multi-project problem as linear programming game again with an important assumption, that is the sequence to process the tasks by the same subcontractor cannot be changed. But with the study in Chapter 4 on a manufacturing outsourcing problem, we may relax this constraint so that the problem setting becomes more reasonable. But how to combine the solution concepts in these two different cooperative games still needs further consideration.

Finally in Chapter 4, we construct a cooperative game to model the manufacturers' job outsourcing problem. So far we just proved the game balancedness in a special case where all the time slot booking costs are the same. And obviously we still have a lot of work to be carried out in the future. An urgent topic is to investigate the balancedness of the manufacturer game (M, v) when the booking costs for time slots are differentiated by peak time and regular time. As discussed in Chapter 4, in this situation the optimal job sequencing problem may become NP-hard. And another question is that since the booking cost is not unique, we should investigate whether we can continue to apply the job splitting method to construct a permutation game and find the core allocation vector basing on the core of this permutation game.

As we can observe, *Game Theory* can be a powerful tool to study the operations management problems when multiple players are involved in. So far in this thesis we just studied three problems using a small section of the solution concepts in *Game Theory*, e.g., Nash equilibrium, linear programming game as well as its duality, and the sequencing game. There are certainly a lot more problems in the operations management area that can be solved applying different solution concepts in game theory. And we believe that more scholar's attention will be attracted to this research area in the near future.

Bibliography

[1] Anand, K. and R. Aron, "GBS on the web: A comparison of price discovery mechanism", *Management Science*, 48: 1546-1562, 2003.

[2] T. Aydinliyim, G. Vairaktarakis. (2010). "Coordination of Outsourced Operations to Minimize Weighted Flow Time and Capacity Bokking Costs". *Manufacturing & Service Operations Management*, 12(2): p. 236-255

[3] Bergantiños, G., E. Sánchez. 2002a. "NTU PERT games". *Operations Research Letters*, 30: p. 130-140.

[4] Bergantiños, G., E. Sánchez. 2002b. "The proportional rule for problems with constraints and claims". *Mathematical Social Sciences* 43: p. 225-249.

[5] Bergantiños, G., E. Sánchez. 2002c. "How to distribute costs associated with a delayed project". *Annals of Operations Research* 109: p. 159-174.

[6] N. BondarevaN, (1963). "Some applications of linear programming methods to the theory of cooperative games". *Problemy Kybernetiki* 10: p. 119 - 139.

[7] P. Borm, G. Fiestras-Janerio, H. Hamers, E. Sánchez, M. Voorneveld, (2002) "On the convexity of games corresponding to sequencing situations with due dates". *European Journal of Operational Research*, 136: p. 616-634

[8] Brânzei, R., G. Ferrari, V. Fragnelli, S. Tijs. 2002. "Two approaches to the problem of sharing delay costs in joint projects". *Annals of Operations Research* 109: p. 359-374.

[9] Castro, J., D. Gómez, J. Tejada. 2007. "A project game for PERT networks". *Operations Research Letters* 35: p. 791-798.

[10] Castro, J., D. Gómez, J. Tejada. 2008a. "A rule for slack allocation proportional to the duration in a PERT network". *European Journal of Operational Research.* 187: p. 556-570.

[11] Castro, J., D. Gómez, J. Tejada. 2008b. "A polynomial rule for the problem of sharing delay costs in PERT networks". *Computers & Operations Research* 35: p. 2376-2387.

[12] Estévez-Fernáandez, A., P. Born, H. Hamers. 2005. "Project games". Working paper, CentER and Department of Econometrics and Operations Research, Tilburg University, Tilburg, Netherlands.

[13] P. Brucker, A. Drexel, R. H. Möhring, K. Neumann, and E. Pesch, 1999. "Resource-constrained project scheduling: Notation, classification, models, and methods". *European Journal of Operational Research,* 112(1): p. 3-41.

[14] Cachon, G.P., M.a. Larivere. 2001. Supply chain coordination with revenue-sharing contracts: strengths and limitations. *Management Science.* 51 30-44.

[15] X. Cai, J. Chen, Y. Xiao, X. Xu, G. Yu, 2006. "Optimal Decision of the Producer and Distributor in a Fresh Product Supply Chain Involving Long Distance Transportation".

[16] X. Cai, J. Chen, X.P. Song, 2006. "The supplier and Retailer's Optimal Strategies in a Supply Chain under the Group-Buying Pricing Mechanism".

[17] X. Cai, G. Vairaktarakis, 2007. "Cooperative strategies for manufacturering planning with negotiable third-party capacity". *Working Paper TM-820, Whetherhead School of Management, Case Western Reserve University.*

[18] P. Calleja, A. Estévez-Fernández, P. Borm, H. Hamers, (2006) "Job sceduling, cooperation, and control". *Operations Research Letters,* 134(1): p. 22-28

[19] P. Calleja, H. Hamers, F. Klijin, M. Slikker. (2002) "On a new class of parallel sequencing situations and related games". *Annals of Operations Research*, 109: p. 263-276

[20] J. Chen., X.L. Chen and X.P. Song, "Bidding's strategy under GBS auction on the Internet". *IEEE Transaction on System, Man and Cybernetics* Part A, 32: 680-690, 2002.

[21] I. Curiel, G. Pederzoli, S. Tijs, B. Veltman (1989). "Sequencing games". *Europen Journal of Operations Research*, 40(3): p. 344-351

[22] I. Curiel, J. Potters, V. Prasad, S. Tijs, B. Veltman (1994). "Sequencing and cooperation". *Operations Research*, 42: p. 566-568

[23] I. Curiel, H. Hamers, F. Klijin, (2002) "Sequencing Games: a Survey" in *Chapters in Game Theory: In Honor of Stef Tijs*, Kluwer Academic Publishers (eds P. Borm and H. Peters), Boston p. 27-50

[24] G. Eppen and A. Iyer. 1997. "Backup agreements in fashion buying - the value of upstream flexibility". *Management Science.* 43(11): p. 1469-1484.

[25] R. Gibbons, 1992. "A Primer in Game Theory", *Harvester Wheatsheaf*

[26] S. Goyal and B. Giri. 2001. "Recent Trends in Modeling of Deteriorating Inventory". *European Journal of Operational Research.* 134: p. 1-16.

[27] Granot, D. (1986). "A generalized linear production model: A unifying model". *Mathematical Programming*, 34(2): p. 212-222

[28] H. Hamers, P. Borm, S. Tijs, (1995) "On games cooresponding to sequencing situations with ready times". *Mathematical Programming*, 70: p. 1-13

[29] H. Hamers, F. Klijin, J. Suijs, (1999) "On the balancedness of multi-machine sequencing games". *European Journal of Operational Research*, 119(3): p. 678-691

[30] W Herroelen, B De Reyck, and E Demeulemeester 1998 "Resource-constrained project scheduling A survey of recent developments" *Computers & Operations Research*, 25(4) p 279-302

[31] R Kauffman and B Wang, "New Buyers' Arrival under Dynamic Pricing Market Microstructure The Case of GBS Discounts on the Internet", *Journal of Management Information Systems*, 18 157-188, 2002a

[32] R Kauffman and B Wang, "Bid Together, Buy together On the Efficacy of GBS Business Models in Internet-Based Selling", *Handbook of Electronic Commerce in Business and Society*, Boca Raton, FL CRC Press, 2002b

[33] Kalai, E , Zemel, E (1982) "Totally Balanced Games and Games of Flows" *Mathematics of Operations Research*, 7(3) p 476-478

[34] M Maschler, B Peleg, L Shapley (1979), "Geometric properties of the kernel, nucleolus, and related solution concepts", *Mathematics of Operations Research* 4 p 303 - 338

[35] B Pasternack, 1985 "Optimal Pricing and Returns for Perishable Commodities" *Marketing Science* 4(2) p 166-176

[36] 1000ventures com 2009 Change management Available at http //1000ventures com/business_guide/crosscuttings/change_management html

[37] Owen, G (1975) "On the core of linear production games" *Mathematical Programming*, 9(1) p 358-370

[38] S Nahmias, 1982 "Perishable Inventory Theory a Review" *Operations Research* 30(4) p 680-708

[39] Peleg, B , P Sudholter 2003 *Introduction to the Theory of Cooperative Games* Kluwer, Boston, MA

[40] F Raafat, 1991 "Survey of literature on continuously deteriorating inventory models" *Journal of the Operational Research Society* 42 p 27-37

[41] Samet, D.,Zemel, E. (1984). "On the core and Dual Set of Linear Programming Games". *Mathematics of Operations Research*, 9(2): p. 309-316

[42] L. Shapley (1953). "A Value for n-person Games". *Annals of Mathematical Studies* 28: p. 307 - 317.

[43] L. Shapley (1967). "On balanced sets and cores". *Naval Research Logistics Quarterly* 14: p. 453 - 460.

[44] L. Shapley (1971). "Cores of convex games". *International Journal of Game Theory*, 1: p. 11-26

[45] M. Slikker, (2006). "Relaxed sequencing games have a nonempty core". *Naval Research Logistics*, 53(4): p. 235-242

[46] S. Tijs, T.Parthasarathy, J.Potters, V. Prasad (1984). "Permutation games: another class of totally balanced games". *OR Spectrum*, 6(2): p. 119-123

[47] A. Tsay, 1999. The Quantity Flexibility Contract and Supplier-Customer Incentives. *Management Science.* 45(10): p. 1339-1358.

[48] Turner, J.R. (ed.) 2003. *Contracting for Project Management*, Gower Publishing, Aldershot, U.K.

[49] Value Based Management.net. 2009. Product life cycle - Industry maturity stages. Available at:
http://www.valuebasedmanagement.net/methods_product_life_cycle.html