# COMPETITIONS AND DELEGATIONS ON NETWORK GAMES: APPLICATIONS IN SUPPLY CHAIN AND PROJECT MANAGEMENT

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Tao Jiang

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2018

Purdue University

West Lafayette, Indiana

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF DISSERTATION APPROVAL

Dr. Thành Nguyen, Chair
Krannert School of Management

Dr. Mohit Tawarmalani
Krannert School of Management

Dr. Yanjun Li
Krannert School of Management

Dr. Jitesh Panchal
School of Mechanical Engineering

**Approved by:**
Dr. Yanjun Li
Head of the School Graduate Program

## ACKNOWLEDGMENTS

I want to thank the members of my committee, especially my advisor, Dr. Thành Nguyen, for all of his help, patience, and guidance over the past three years. Ph.D. study is a long journey to me, and I have tried many different problems. I am glad that I eventually found some interesting results.

I would also like to thank the bouldering wall at Purdue CoRec Center and all the friends I have met over there (even though they may never read my thesis). I am not sure what am I going to do ten years later, but I am pretty sure that I will be climbing forever.

TABLE OF CONTENTS

Page

# LIST OF FIGURES

ABSTRACT

Jiang, Tao PhD, Purdue University, December 2018. Competitions and Delegations on Network Games: Applications in Supply Chain and Project Management . Major Professor: Thành Nguyen.

We consider the models of sequential games over supply chain networks and production chain networks. In the supply chain model, we show that in particular, for series-parallel networks, there is a unique equilibrium. We provide a polynomial time algorithm to compute the equilibrium and study the impact of the network structure to the total trade flow at equilibrium. Our results shed light on the trade-off between competition, production cost, and double marginalization.

In the production chain model, we investigated sequential decisions and delegation options over three agents, chain, and tree networks. Our main contribution is showing the value of delegation and how to maximumly leverage the middleman's aligned interests with the principal. In particular, we provide a polynomial time algorithm to find the optimal delegation structure and the corresponding necessary contract payments for the principal. Furthermore, we analyzed the trade-off of the delegation and gave a deeper insight into the value of delegation in different conditions. Several questions are left for future research such as what's the optimal delegation structures in general tree and how to build the model that agents can try multiple times until the task is successful.

# 1. SUPPY CHAIN

## 1.1 Introduction

Supply chain networks in practice are multi-tier and heterogeneous. A firm's decision influences not only other firms within the same tier but also across. The literature on game theoretical models of supply chain networks, however, has largely focused on two extreme cases: heterogeneous 2-tier networks (bipartite graph) [1, 2] and a linear chain of $n$-tier firms [3, 4]. One main reason for this is that most models of sequential decision making in multi-tier supply chain networks are intractable. Sequential decision making is a well-observed phenomenon in supply chains because firms at the top tier typically need to make decisions on the quantity to sell to firms in the next tier and the buying firms then decide how much to buy from which suppliers, and continue to pass on the goods by determining the quantity for firms at the next level.

To study such models, one needs to analyze subgame perfect equilibria in which a firm needs to internalize all the decision of all the firms downstream and compete with all the firms of the same tier at the same time. Another factor that further complicates models of general supply chain networks is that even the basic concept of tiers is ambiguous because there are often multiple routes of different length that goods are traded from the original producers to the consumers. Our paper studies a model of sequential network game motivated by supply chain network applications. Our main goal is to understand the effect of network structure on the efficiency of the system.

When considering the efficiency of a supply chain network, there is a trade-off between the length and the number of trading routes. On the one hand, a large variety of options to trade indicates a high degree of competition, which leads to a more efficient system. On the other hand, along the trading path causes double, triple and higher degree marginalization problems. In this paper, to capture these ideas, we consider a sequential game theoretical model for a special class of networks: series-parallel graphs. We focus our analysis on these networks because they are rich enough for studying the trade-off described above and simple sufficient for characterizing the equilibrium outcomes. In particular, series-parallel networks have a natural decomposition of parallel and serial insertions. A parallel insertion, which merges two different sub-networks at the source and the sink, can capture the increase in competition. A serial insertion, which attaches two sub-networks sequentially, corresponds to the increase in the length of trading paths.

Our first contribution is a result showing that the equilibrium is unique in these networks. Furthermore, we provide a polynomial time algorithm to compute the equilibrium. Our algorithm is nontrivial and combines a dynamic program capturing the backward induction of an equilibrium computation and a convex quadratic programming technique for calculating the flow and price functions.

Our second contribution is a set of comparative analysis on the influence of the network structure and the two operations in series-parallel graphs to the total flow at equilibrium. For example, we show that:

- Parallel insertion increases total flow, while serial insertion decreases total flow.
- Given two networks $N_1$ and $N_2$ the order of serial insertion to obtain $N_1N_2$ or $N_2N_1$ network matters only when the production cost of at least one component is positive. The total flow is larger if the component with a higher production cost is closer to the source.
- In parallel insertion, adding a component to a longer range increases the flow more than adding it to a shorter range. This means increasing competition globally is more beneficial than increasing competition locally.
- An upstream firm that controls all the flow of goods of another downstream firm has a location advantage. The utility of this upstream firm is at least twice as much as the dominated downstream firm.

Finally, we show that extending the series-parallel graph to a slightly more general class of network, series-parallel graphs with multiple producers or markets, the problem may become intractable. With multiple producers and single market, our technique extends to construct the unique equilibrium of the game. However, with multiple producers and markets, there may exist multiple or no pure strategy equilibria.

The paper is organized as follows. In section 1.2 we introduce the model of competition and series-parallel networks together with the composition. In section 1.3 we provide the algorithm to compute the unique equilibrium. Section 1.4 uses the network composition and the algorithm to analyze comparative analysis on how network structure influences the efficiency measured by the total trade flow. Section 1.5 discusses extensions to other classes of networks and shows that pure equilibrium might not exist in general networks.

**Related work:** In our paper, we assume the consuming nodes are Cournot markets. Thus, the structure of the game is closely related to the literature on Cournot games in networks. [2, 5], for example, consider a Cournot game in two-sided markets. [6] study Cournot game in three-tier networks. However, the 2-tier structure of the network in these papers, and the assumption that only the middle tier make the decision in [6] assumes away the complex sequential decision making considered in our paper.

[7] studies a Cournot game in general networks. However, firms are assumed to make simultaneous decisions. As discussed above, simultaneous games are easier to

analyze but do not capture the essential elements of sequential decision making of firms in supply chain networks.

[8] considers assembly network where agents make a sequential decision but assumes a tree network. The analysis for a tree network is substantially simpler, because each firm has a single downstream node that it can sell the products to. In our game, the network is more general, and each firm needs to make the decision of the goods quantity to each firm that it is connected to. As we show, some of the quantities on some of the links can be zero. Such "inactive" links make the analysis more complicated.

Recently, [9] also considered a sequential game and used market clearing prices like our paper. The network considered in this paper is however symmetric, and its structure is linear. The focus of [9] is on the uncertainty of yields, which is different from the motivation in our paper.

More broadly, our paper belongs to the growing literature of network games and their applications in supply chains, including[4, 10, 11, 12]. These papers, however, are different from ours in the main focus as well as the modeling approach. [10] for example, assume a linear structure of supply chains, [11] consider price competition in two-tier networks, and [4, 12] analyze bargaining games in networks with simpler structures. The main contribution of our paper to this line of work is a tractable analysis of sequential competition model in series-parallel graphs, which allows for richer comparative analysis and deeper understanding of how basic network elements influence market outcomes.

### 1.2 Model

In this section, we introduce the sequential decision mechanism in a supply chain network and the definition of series-parallel graph.

#### 1.2.1 Sequential Decisional Game

Let $G = (V \cup \{s, t\}, E)$ be a simple directed acyclic network that represents an economy where $s$ is the producer at the source, $t$ is the sink market and $V$ represents intermediary firms. The edges of $G$ represent the possibility of trade between two agents. The direction of an edge indicates the direction of trade. The outgoing end of the edge corresponds to the seller, and the incoming end is the buyer, while $s$ has only outgoing edges, and $t$ has only incoming edges. The remaining vertices $i \in V$ representing intermediary firms has both incoming and outgoing edges. For a vertex $i$, $B(i)$ (buyer set) and $S(i)$ (seller set) are the sets of agents that can be buyers and sellers in a trade with $i$, respectively.

Assume every agent has full information about the structure of the network. Agents start deciding their order quantities, and selling quantities after the output of their upstream suppliers is determined. Furthermore, the *market clearance price* at $i$ is such that the total demand from $i$ matches the total supply.

Each intermediate firm $i$ decides on how much to buy from each of his sellers and how much to sell to each of his buyers. Specifically, $i$'s decision includes:

- The buying quantity $x_{ki}^{in} \geqslant 0$ for every $k \in S(i)$;
- The selling quantity $x_{ij}^{out} \geqslant 0$ to every $j \in B(i)$.

while the source only initializes the supplying amount and the sink will take all the goods at the market price. Fig. 1.1 shows an example of decisions in the supply chain:

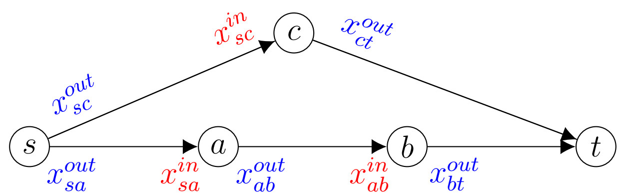



Fig. 1.1.: Decisions in a Supply Chain

For producer $s$, his unit cost of production $p_s$ is given and assumed to be an affine function on $X_s$, the total amount of goods to sell.

$$p_s = a_s + d_s X_s, \text{ where } X_s = \sum_{i \in B(s)} x_{sj}^{out}, d_s \geqslant 0 \text{ and } a_s \geqslant 0.$$

Sink node $t$ does not represent a firm, it corresponds to an end market. The price function $p_t$ at sink node $t$ is given and assumed to be an affine function on the total amount of goods, $X_t$, sold to market $t$.

$$p_t = a_t - b_t X_t, \text{ where } X_t = \sum_{i \in S(t)} x_{it}^{in} = \sum_{i \in S(t)} x_{it}^{out}, a_t > 0, \text{ and } b_t > 0.$$

Note that the market must accept all the goods thus does not have a choice to reject. That is, $x_{it}^{in} = x_{it}^{out}$ for each $i \in S(t)$. Generally, for a trade $ij \in E$, the buyer $j$ cannot obtain more than what the seller $i$ offers, thus $x_{ij}^{in} \leqslant x_{ij}^{out}$. We assume that each intermediary firm $i$ cannot get goods from any other source besides his sellers. Therefore the outflow of $i$ cannot be more than the inflow of $i$,

$$\sum_{j \in B(i)} x_{ij}^{out} \leqslant \sum_{k \in S(i)} x_{ki}^{in}.$$

The price for intermediate goods for each node $i \in V$, denoted as $p_i$, is determined endogenously such that the corresponding intermediate market at $i$ clears.

Furthermore, agents do not get any value from retaining the goods. They incur a processing cost, which we assume to be quadratic in the quantity of goods the agents sell.

The payoff of the source firm $s$ is

$$\Pi_s = \sum_{j \in B(s)} p_j x_{sj}^{in} - p_s \sum_{j \in B(s)} x_{sj}^{out} - \frac{c_s}{2}\Big(\sum_{j \in B(s)} x_{sj}^{out}\Big)^2 \text{ where } c_s \geqslant 0. \tag{1.1}$$

The utility of an intermediate agent $i \in V$ is

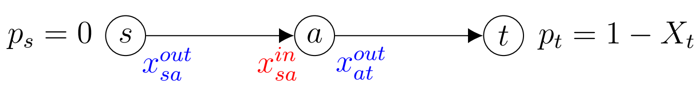
$$\Pi_i = \sum_{j \in B(i)} p_j x_{ij}^{in} - p_i \sum_{k \in S(i)} x_{ki}^{out} - \frac{c_i}{2}\Big(\sum_{k \in S(i)} x_{ki}^{out}\Big)^2 \text{ where } c_i \geqslant 0. \tag{1.2}$$

The formula decomposes the utility function into three terms: the total revenue from $j \in B(i)$, the total cost of materials from $k \in S(i)$, and the processing cost.

The timing of the game is as follows. The producer (source) makes its decision first. A firm makes his decision on the selling quantity to his downstream, once all of his sellers have made decisions.[1] When choosing their order quantities to maximize their expected profits, firm $i$ also needs to take into account the strategies of both the competing firms and the firms downstream. When a firm makes its decision, it only knows the quantities offered by the firms upstream and makes a prediction based on the rational expectation of other firms' strategies.

Here is a toy example for the equilibrium at a supply chain:

EXAMPLE 1
*Assume no processing cost in this example.*




*Suppose source $s$ makes an decision to sell $x_{sa}^{out} = x$ amount of goods to agent $a$. Now for agent $a$, since he has no benefit from unsold goods, his buying amount will be equal to the selling amount, denoted as $x_a = x_{sa}^{in} = x_{at}^{out}$. Meanwhile, the utility function is*

$$\pi_a(x_a) = (1 - x_a)x_a - p_a x_a$$

*where $p_a$ is the market clearance price. And the optimal decision for $a$ ($\frac{\partial \pi_a}{\partial x_a} = 0$) is*

$$x_a = \frac{1 - p_a}{2}$$

[1] By the market clearance price, at the firm's optimal decision, it will consume all the supply from upstream.

*By the definition of the market clearance price, we have $x_a = x_{sa}^{out}$. Thus, the relation between selling amount and market clearance price at agent a is*

$$p_a = 1 - 2x_{sa}^{out}$$

*Now consider the utility function of the source,*

$$\pi_s = p_a x_{sa}^{out} = (1 - 2x_{sa}^{out})x_{sa}^{out}$$

*Finally, we have the optimal supplying amount at the source $x_{sa}^{out} = 1/4$, which will result in a market clearance price $p_a = 1/2$, and processing amount through $a$ is also $1/4$. Note that this is the unique equilibrium flow in this toy supply chain.*

### 1.2.2 Series Parallel Graph

In this paper, we consider the case when $G$ is a *Series Parallel Graph* (SPG). This class of networks is well studied and has several applications in graph theory. (See for example [13]). For completeness, we provide a formal definition as follows.

**Definition 1.2.1 (SPG)** *A* single-source-and-sink SPG *is a graph that may be constructed by a sequence of series and parallel compositions starting from a set of copies of a single-edge graph, where:*

1. Series composition *of $X$ and $Y$: given two SPGs $X$ with source $s_X$ and sink $t_X$, and $Y$ with source $s_Y$ and sink $t_Y$, form a new graph $G = S(X, Y)$ by identifying $s = s_X$, $t_X = s_Y$, and $t = t_Y$.*

2. Parallel composition *of $X$ and $Y$: given two SPGs $X$ with source $s_X$ and sink $t_X$, and $Y$ with source $s_Y$ and sink $t_Y$, form a new graph $G = P(X, Y)$ by identifying $s = s_X = s_Y$ and $t = t_X = t_Y$.*

## 1.3 Equilibrium Characteristics and Computation

In this section, before describing how equilibrium can be computed, we observe some properties of equilibrium and series-parallel graphs.

### 1.3.1 Properties of Equilibrium

First, observe that the best strategy for agent $i$ is always to sell as much as bought since it cannot benefit from paying more for those unsold goods. At the selling side, suppose firm $i$ is willing to offer $x_{ij}^{out}$ quantity of goods to firm $j$, but part of the goods got rejected, i.e. $x_{ij}^{in} < x_{ij}^{out}$. However, this can never happen in equilibrium, because $i$ will be better off by rejecting $x_{ij}^{out} - x_{ij}^{in}$ amount of goods from its upstream at the beginning.

The next proposition lists the properties of supplying quantities at an equilibrium:

**Proposition 1.3.1** *With market clearance price, where $p_s$ and $p_t$ are given, each agent $i \in V$ gets to decide $x_{ij}^{out}$ where $j \in B(i)$ and $x_{ki}^{in}$ where $k \in S(i)$, and $s$ gets to decide $x_{sj}^{out}$ for $j \in B(s)$. The equilibrium satisfies:*

1. $x_{ij}^{out} = x_{ij}^{in}$ *for* $ij \in E$.
2. $\sum_{k\in S(i)} x_{ki}^{in} = \sum_{j\in B(i)} x_{ij}^{out}$, *i.e. inflow is equal to outflow for agent* $i \in V$.

For later notations, at the equilibrium, we will set $x_{ij}$ as the flow along the edge $ij$, i.e. $x_{ij} = x_{ij}^{out} = x_{ij}^{in}$, and no longer use $x_{ij}^{in}$ and $x_{ij}^{out}$. Meanwhile, since each firm accepts all the offers and sells everything they bought, we denote this sum of flow as processing quantity for firm $i$, i.e. $X_i = \sum_{k\in S(i)} x_{ki} = \sum_{j\in B(i)} x_{ij}$. For market $t$, the price is given as $p_t = a_t - b_t X_t$ because $t$ always accepts everything. For example, at equilibrium, the flows of the supply chain in Fig. 1.1 is

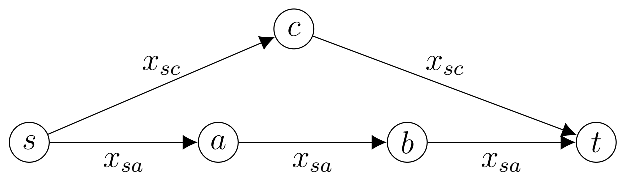



Fig. 1.2.: Flows in a Supply Chain at Equilibrium

With the above new notations, by rewriting equation 1.2, the utility of agent $i$ becomes

$$\Pi_i = \sum_{j\in B(i)} p_j x_{ij} - p_i \sum_{j\in B(i)} x_{ij} - \frac{c_i}{2}(\sum_{j\in B(i)} x_{ij})^2. \tag{1.3}$$

and by rewriting equation 1.1, the utility of source firm $s$ becomes

$$\Pi_s = \sum_{j\in B(s)} p_j x_{sj} - p_s \sum_{j\in B(s)} x_{sj} - \frac{c_s}{2}(\sum_{j\in B(s)} x_{sj})^2. \tag{1.4}$$

For the flow activities along each edge, we define an edge $ij \in E$ is *active* if $x_{ij} > 0$, and *inactive* if $x_{ij} = 0$. Note that for every agent, the buying price should be at most the selling price so that the agent can obtain non-negative utility, thus whenever $ij$ is active, $p_i \leqslant p_j$. Otherwise, $i$ could have been better off by rejecting some goods from upstream and choose not to offer any goods to $j$.

**Proposition 1.3.2** *For each $ij \in E$ that is active, the market clearance price at an equilibrium satisfies $p_i \leqslant p_j$.*

### 1.3.2 Properties of Series Parallel Graphs

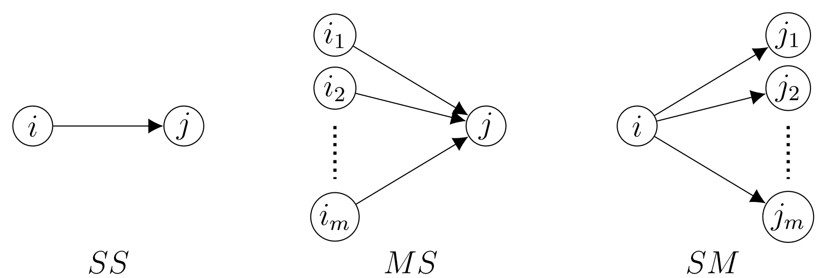
Consider a path $l_{ij} = (i, v_1, ..., v_k, j)$ from $i$ to $j$. If there is an edge $ij \in E$, then we say $ij$ is a *shortcut* of $l_{ij}$. The intuition is $i$ always prefers selling to $j$ directly than through the intermediate agents along the path $l_{ij}$, and we prove it in the following proposition. The proof is provided in Appendix 2.6.1.

**Proposition 1.3.3** *At an equilibrium of a series parallel graph $G$, if $ij \in E$ is a shortcut of a path $l_{ij}$, then there is no trade on $l_{ij}$. Thus, all the edges on the path $l_{ij}$ are inactive.*

By this observation, without loss of generality, we can assume that $G$ does not have any shortcuts.

Here we introduce the node relations in SPG. Node $k$ is called parent node of $i$ if there is a directed path from $k$ to $i$. The set of parent nodes of $i$ is denoted as $P(i)$. By a similar idea, we can define the child node and set of children $C(i)$. If consider the relation between direct parent and child $i \to j$, i.e., $ij \in E$, there are three possibilities in SPG:

- Single seller and single buyer, $|S(j)| = |B(i)| = 1$. (SS)
- Multiple sellers and single buyer, $|S(j)| \geqslant 2, |B(i)| = 1$. (MS)
- Single seller and multiple buyers, $|S(j)| = 1, |B(i)| \geqslant 2$. (SM)




Sometimes there are multiple paths from a parent node to one of its children, and we call these paths *disjoint* if they do not have any common intermediary nodes, that is, all nodes except the starting and the ending ones are different. Base on this definition, we can define the merging nodes with respect to node $i$.

**Definition 1.3.1 (Self-merging Child Node)** *Node $j \in C(i)$ is a* self-merging child node *of $i$ if there are disjoint paths from $i$ to $j$. The set of such nodes $j$ is denoted as $C_S(i)$.*

**Definition 1.3.2 (Parent-merging Child Node)** *Node $j \in C(i)$ is a* parent-merging child node *of $i$, if there exist node $k \in P(i)$, such that there are disjoint paths from $k$ to $j$. The set of such nodes $j$ is denoted as $C_P(i)$.*

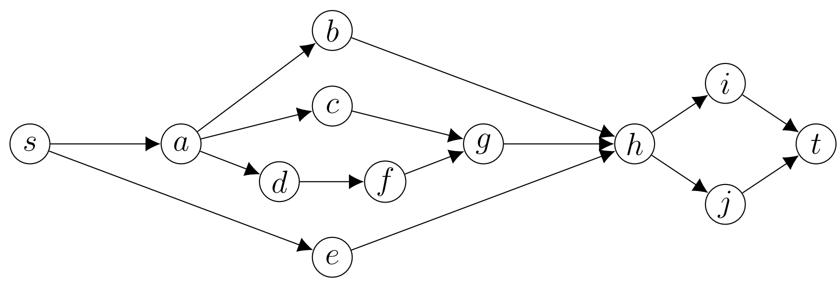
We also introduce the special self-merging child nodes of $i$ and its child $j$ as $C_T(i,j) = C_S(i) \cap C(j) \backslash C_P(i)$. This notation is useful because it helps us capture the "internal" merging nodes that are responsible for the price of $i$ and flow to $j$ later on.

**Proposition 1.3.4** *A series parallel graph has the following properties:*

1. $C_P(s) = C_P(t) = \emptyset$.
2. *In SS case, for* $ij \in E$, $C_P(j) = C_P(i)$.
3. *In SM case, for* $ij \in E$, $C_P(j) = C_P(i) \sqcup C_T(i,j)$.
4. *In MS case, for* $ij \in E$, $C_P(i) = C_P(j) \sqcup \{j\}$.

Note that $\sqcup$ stands for the disjoint set union.

EXAMPLE 2




*In this graph, for node* $a$, $C_S(a) = \{g,h\}$, *because* $\{g,h\} \subset C(a)$ *and there are multiple disjoint paths from* $a$ *to* $g$ *and* $h$, *while* $t \notin C_S(a)$ *because all the paths from* $a$ *to* $t$ *must go through the common node* $h$ *which are not disjoint paths;* $C_P(a) = \{h\}$ *because* $h \in C(a)$, $s \in P(a)$, *and there are multiple disjoint paths from* $s$ *to* $h$; $C_T(a,b) = \emptyset$, *while* $C_T(a,c) = \{g\}$.

*For node* $c$, $C_P(c) = \{g,h\}$, *while* $C_S(c) = \emptyset$; *For node* $g$, $C_P(g) = \{h\}$, *while* $C_S(g) = \emptyset$.

*Since* $a \to c$ *is the SM relation, by Proposition 1.3.4,* $C_P(c) = \{g,h\} = C_P(a) \sqcup C_T(a,c)$. *Also,* $C_P(c) = \{g,h\} = C_P(g) \sqcup \{g\}$, *because* $c \to g$ *belongs to the MS relation.*

### 1.3.3 Equilibrium Computation

In this section, we present an algorithm to compute the equilibrium supplying quantities at every edge. To do that, we first derive a closed-form expression for the market clearance price at each firm through a backward algorithm in section 1.3.3. Then, the unique optimal quantities for each firm can be solved following the decision sequence from source to sink as in section 1.3.3.

**Market Clearance Price Computation.**

A vital characteristic of the equilibrium is that all edges are active. The market clearance prices have closed-form expressions, and quantities can be computed based on the composition of SPG.

Lemma 1.3.1 *At equilibrium, if $a_s < a_t$, then all the edges are active. Also the market clearance price at agent $i$ is an inverse linear function of $X_i$ and flows to its parent-merging children nodes.*

$$p_i = a_t - b_i X_i - \sum_{k \in C_P(i)} b_k X_k$$

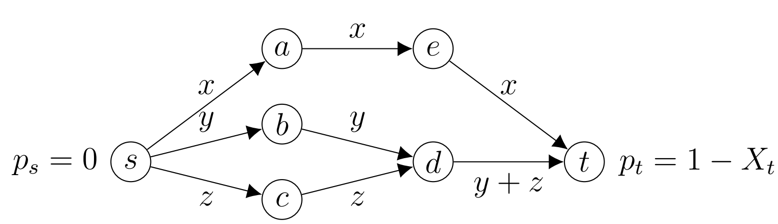
*where $b_i > 0, \forall i \in V$ is a constant that only depends on the structure of $G$ and processing cost.*

The above lemma shows a concise way to present the price function at equilibrium. The last piece of work for price function computation is to find the value of $b_i$ for $i \in V$. This is provided in the proof given in Appendix 2.6.2. By adapting the main equations in that proof, here we introduce a backward Algorithm 1 to compute the market clearance price at equilibrium (starting from $ALG_1(j = t, G)$).

In each iteration, we just compute $b_i$ and this can be done in $O(deg^+(i))$ time where $deg^+(i)$ is the outdegree of $i$. Besides, we also store the convex coefficients of each downstream node $j \in B(i)$. The number of $b_i$ computation is bounded by $O(|V|)$. Therefore, it takes linear time to compute the price functions by Algorithm 1.

Below is an example of the price function computation, for the general form expression as in Algorithm 1, please check Example 15.

Example 3 (Price Function Computation)
*Assume no processing cost in this example.*




*From Proposition 1.3.1, we know that inflow must equal to outflow for each firm at equilibrium. Therefore, we can set $x_{sa} = x_{ae} = x_{et} = x$, $x_{sb} = x_{bd} = y$, $x_{sc} = x_{cd} = z$, and $x_{dt} = y + z$.*

*Consider the utility of $e$,*

$$\Pi_e(x) = p_t x - p_e x = (1 - x - y - z)x - p_e x.$$

[2]If $|C_S(i)| \geqslant 2$, the computation of $b_i$ is more complicated, the detail is provided in Appendix 2.6.2.

**Algorithm 1** : Price Function Computation (Backward)

1: Given the downstream buyer $j$'s clearance price function $p_j$, compute the upstream seller $i$'s clearance price case by case:

- Single seller and single buyer case,

$$b_i = 2b_j + \sum_{k \in C_P(j)} b_k + c_i. \tag{SS}$$

- Multiple sellers and single buyer case, for each seller,

$$b_i = b_j + \sum_{k \in C_P(j)} b_k + c_i. \tag{MS}$$

- Single seller and multiple buyers case ($|C_S(i)| = 1$) [2],

$$b_i = \frac{2}{\sum_{j \in B(i)} \frac{1}{b_j}} + 2b_h + \sum_{k \in C_P(j) \setminus \{h\}} b_k + c_i. \tag{SM}$$

2: Set the price function at seller $i$: $p_i = a_t - b_i X_i - \sum_{k \in C_P(i)} b_k X_k$.
3: **if** seller $i$ is the source **then**
4: Return.
5: **else**
6: Run $ALG_1(j = i, G)$.

*Market clearance price function of* $e$ *can be derived by solving the stable condition of the utility maximization problem:*

$$\frac{\partial \Pi_e(x)}{\partial x_{et}} = 1 - 2x - y - z - p_e = 0 \Rightarrow p_e = 1 - 2x - y - z.$$

*Similarly, we can obtain the following price functions:*

$$\begin{aligned} p_a &= 1 - 4x - y - z, \\ p_d &= 1 - x - 2y - 2z, \\ p_b &= 1 - x - 4y - 2z, \\ p_c &= 1 - x - 2y - 4z. \end{aligned}$$

*Note that the above price functions can be written as the form of*

$$p_i = a_t - b_i X_i - \sum_{k \in C_P(i)} b_k X_k.$$

*As in Lemma 1.3.1, for example,*

$$p_b = 1 - x - 4y - 2z = 1 - b_b X_b - b_d X_d - b_t X_t.$$

*where* $b_b = 2, b_d = b_t = 1, C_P(b) = \{d, t\}$.

*The utility of* $s$ *is*

$$\Pi_s(x, y, z) = p_a x + p_b y + p_c z - p_s(x + y + z).$$

*Let* $p_{s_a}$ *be the price function that has to be satisfied if* $\frac{\partial \Pi_s(x,y,z)}{\partial x} = 0$*, where* $p_{s_b}$ *and* $p_{s_c}$ *are defined similarly. Hence, the following stable condition is obtained:*

$$\begin{aligned}
\frac{\partial \Pi_s(x,y,z)}{\partial x} &= 0 \Rightarrow p_{s_a} = 1 - 8x - 2y - 2z, \\
\frac{\partial \Pi_s(x,y,z)}{\partial y} &= 0 \Rightarrow p_{s_b} = 1 - 2x - 8y - 4z, \\
\frac{\partial \Pi_s(x,y,z)}{\partial z} &= 0 \Rightarrow p_{s_c} = 1 - 2x - 4y - 8z.
\end{aligned}$$

*Note that all above three equations are necessary conditions for* $p_s$*, by using the convex coefficients* $\mu_1 = \frac{2}{5}, \mu_2 = \mu_3 = \frac{3}{10}$*, we write* $p_s$ *as function of total flow* $X_s = x + y + z$*,*

$$\begin{aligned}
p_{s_{abc}} &= \mu_1 p_{s_a} + \mu_2 p_{s_b} + \mu_3 p_{s_c} \\
&= 1 - \frac{22}{5}(x + y + z) \\
&= 1 - \frac{22}{5} X_s
\end{aligned}$$

*Till here, we have the equilibrium price function at every node. Furthermore, we can find the total flow at equilibrium* $X_s$ *at source by solving*

$$p_{s_{abc}} = p_s = 0 \Rightarrow X_s = \frac{5}{22}.$$

Base on the closed-form relation between seller and buyer as in SS, MS, and SM, we can prove a stronger version of Proposition 1.3.2. The proof can be found in Appendix 2.6.3.

**Proposition 1.3.5** *If an edge is active in an SPG, then the price at corresponding seller is strictly less than the price at the buyer.*

**Equilibrium Quantities Computation.**

After having the closed-form of the market clearance price function, we present an algorithm that finds the unique supply quantities at equilibrium. Consider the quantities decision for firm $i$ to its downstream buyers $j \in B(i)$. Suppose there is only a single outflow for firm $i$, i.e., $|B(i)| = 1$, by Proposition 1.3.1, inflow equals outflow at firm $i$, and firm $j$ will take all the supplying quantities from $i$, formally, $x_{ij} = X_i$. Hence, in the following analysis, we focus on the nontrivial case when firm has multiple downstream buyers, i.e., $|B(i)| \geqslant 2$. How to optimally allocate the supplying quantities to different buyers? In particular, firm $i$'s decision $x_{ij}$, where $j \in B(i)$, is to optimize its utility $\Pi_i$. Recall the utility equation 1.3:

$$\Pi_i = \sum_{j \in B(i)} p_j x_{ij} - p_i \sum_{j \in B(i)} x_{ij} - \frac{c_i}{2}(\sum_{j \in B(i)} x_{ij})^2.$$

Note that before firm $i$ makes decision, $p_i$ is determined by upstream flows, but $p_j$ may be affected by $x_{ij}$ where $j \in B(i)$. By Lemma 1.3.1, we can write the price function of seller $j$ as

$$p_j = a_t - b_j x_{ij} - \sum_{k \in C_P(j)} b_k X_k. \tag{1.5}$$

Note that by the property of SPG, $|S(j)| = 1$ when $B(i) \geqslant 2$. Thus, $X_j = x_{ij}$.

To find the optimal supply quantities to downstream firm $j$, take the derivative of the utility function with respect to $x_{ij}$, and obtain

$$\frac{\partial \Pi_i}{\partial x_{ij}} = p_j - \sum_{l \in B(i)} \frac{\partial p_l}{\partial x_{ij}} x_{il} - p_i - c_i X_i. \tag{1.6}$$

Expand the second term of equation 1.6 as

$$\begin{aligned}\sum_{l \in B(i)} \frac{\partial p_l}{\partial x_{ij}} x_{il} &= b_j x_{ij} + \sum_{l \in B(i)} \left(\frac{\partial \sum_{k \in C_P(l)} b_k X_k}{\partial x_{ij}}\right) x_{il} \\ &= b_j x_{ij} + \sum_{l \in B(i)} \Big(\sum_{k \in C_P(l) \cap C(j)} b_k\Big) x_{il} \\ &= b_j x_{ij} + \sum_{h \in C_T(i,j)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i.\end{aligned} \tag{1.7}$$

Plug equation 1.5 and equation 1.7 back into equation 1.6, we get

$$\begin{aligned}\frac{\partial \Pi_i}{\partial x_{ij}} &= a_t - 2b_j x_{ij} - \sum_{k \in C_P(j)} b_k X_k - \sum_{h \in C_T(i,j)} b_h X_h - \sum_{k \in C_P(i)} b_k X_i - c_i X_i - p_i \\ &= a_t - 2b_j x_{ij} - 2 \sum_{h \in C_T(i,j)} b_h X_h - p_i - const.\end{aligned} \tag{1.8}$$

Note that $X_i$ and $X_k$ where $k \in C_P(i)$ are given constant predetermined by upstream supply. By point 3 of Proposition 1.3.4, $C_P(j) = C_P(i) \sqcup C_T(i,j)$ and we have

$$const = (\sum_{k \in C_P(i)} b_k + c_i)X_i + \sum_{k \in C_P(i)} b_k X_k.$$

Observe the utility of firm $i$ (equation 1.3) is concave. At the equilibrium, if $x_{ij} > 0$, then $\frac{\partial \Pi_i}{\partial x_{ij}} = 0$; if $x_{ij} = 0$, then $\frac{\partial \Pi_i}{\partial x_{ij}} \leqslant 0$. This problem is equivalent to the following linear complementary problem (LCP) with variables $x_{ij}$ where $j \in B(i)$.

$$\begin{cases} \frac{\partial \Pi_i}{\partial x_{ij}} x_{ij} = 0, \\ \frac{\partial \Pi_i}{\partial x_{ij}} \leqslant 0, \\ x_{ij} \geqslant 0, \quad \forall j \in B(i). \end{cases} \tag{LCP}$$

To solve the above system of equations LCP, we introduce a convex quadratic program:

$$\begin{aligned} \min_{x_{ij}, X_k} \quad & \sum_{j \in B(i)} b_j x_{ij}^2 + \sum_{k \in C_S(i) \setminus C_P(i)} b_k X_k^2 & \\ \text{subject to} \quad & a_t - 2b_j x_{ij} - \sum_{k \in C_T(i,j)} 2b_k X_k - const \leqslant p_s & \text{for } j \in B(i), \\ & x_{ij} \geqslant 0 & \text{for } j \in B(i). \end{aligned} \tag{CQP}$$

By examining the KKT conditions of the quadratic program, the independent variables $X_k$ satisfy $X_k = \sum_{j:k \in C(j)} x_{ij}$, which fits the definition of $X_k$. Besides, equation LCP also holds. The proof of Lemma 1.3.2 is provided in Appendix 2.6.4.

LEMMA 1.3.2 *Problem LCP is equivalent to the convex optimization problem CQP, and the solution is unique.*

After the market clearance price function is computed by Algorithm 1, by solving CQP directly, we have the optimal decision of each firm in polynomial time. In fact, the algorithm can be sped up by distributing the flow from $i$ to $j \in B(i)$ proportionally to the convex coefficients pre-computed in Algorithm 1. Besides, all the $p_j$'s have the same price value so that $i$ has no preference about whom to sell to. The proof of Lemma 1.3.3 is provided in Appendix 2.6.5.

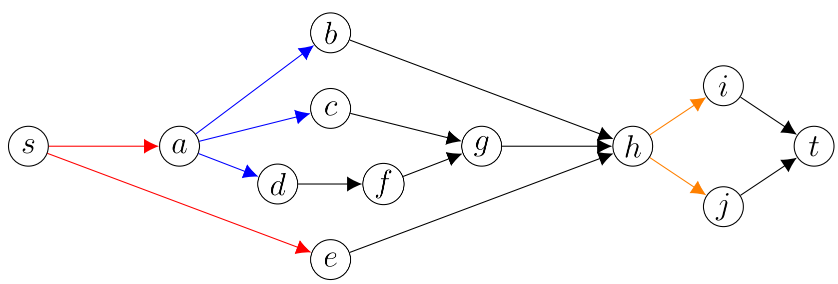
LEMMA 1.3.3 *For the SM case, $\Pi_i$ is maximized by distributing the flow to $j \in B(i)$ proportionally to the convex coefficients pre-computed in Algorithm 1. Besides, all the $p_j$'s have the same price value.*

**Algorithm 2** : SPG Flow Computation (Forward)

(Initialize $X_j = 0, \forall j \in V$. Start with $Alg_2(i = s, p_i = p_s, G)$.)
Distribute the flow $x_{ij}$ where $j \in B(i)$ proportionally to the convex coefficients.
**for** $k \in C_S(i)$ **do**
    **if** $X_k = 0$ **then**
        $X_k = \sum_{j:k \in C_P(j)} x_{ij}$.
**for** $j \in B(i)$ **do**
    **if** $X_j = 0$ **then**
        $X_j = x_{ij}$.
    $p_j = a_t - b_i X_j - \sum_{k \in C_P(j)} b_k X_k$.
    Run $Alg_2(j, p_j, G)$.
Return.

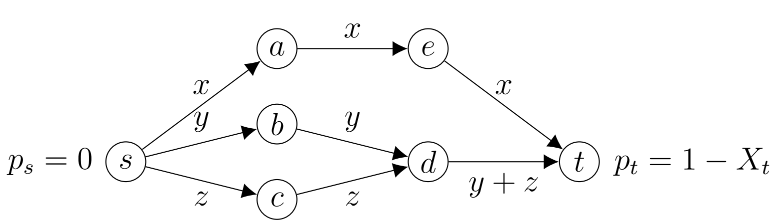
The algorithm starts with solving the equilibrium flow at source, then based on the flow decision, each $j \in B(i)$ is considered as the new source node, and their equilibrium flow decisions were solved along the path to the sink, as demonstrated in the following examples.

Example 4 (Flow Computation Order)
*Consider the same instance as Example 2:*




*Algorithm 2 solves the flow quantities along the red edges first, then those along the blue and orange edges. Note that the flow along the black edge is equal to the total inflow to the upstream firm by the definition of market clearance price (e.g.* $x_{gh} = X_g = x_{cg} + x_{fg}$*).*

Example 5 (Flow Computation)
*Consider the same instance as Example 3:*



*We already have $X_s = \frac{5}{22}$. By distributing the flow proportionally to the convex coefficients $\mu_1 = \frac{2}{5}, \mu_2 = \mu_3 = \frac{3}{10}$, we have $x = \mu_1 X_s = \frac{1}{11}$ and $y = z = \mu_2 X_s = \frac{3}{44}$.*

*We calculate the price values $p_a$, $p_b$, and $p_c$ from the flow values $x$, $y$, and $z$:*

$$\begin{aligned} p_a &= 1 - 4x - y - z = 1 - 4 \times \frac{1}{11} - \frac{3}{44} - \frac{3}{44} = \frac{1}{2}, \\ p_b &= 1 - x - 4y - 2z = 1 - \frac{1}{11} - 4 \times \frac{3}{44} - 2 \times \frac{3}{44} = \frac{1}{2}, \\ p_c &= 1 - x - 2y - 4z = 1 - \frac{1}{11} - 2 \times \frac{3}{44} - 4 \times \frac{3}{44} = \frac{1}{2}. \end{aligned}$$

**Theorem 1.3.1** *For SPG, there exists a linear time algorithm to solve the equilibrium flow and prices, and the equilibrium is unique.*

**Proof** The equilibrium flow and prices can be found by Algorithm 1 and Algorithm 2 in linear time as aforementioned. The uniqueness of equilibrium can be proved by encoding this problem into LCP and its corresponding CQP has a unique solution. ∎

## 1.4 Structural Analysis of Network Pricing Equilibria

In this section, we compare the equilibria and analyze the influence of different operations on SPG, e.g., switching the order of two components in SPG, or inserting a new component to a given SPG. The criterion of the influence is the network efficiency defined as follows:

**Definition 1.4.1 (Efficiency)** *A supply chain network is more efficient if it has a larger total flow value at equilibrium.*

Following are some general results for SPG. The first proposition shows that the direct selling from source to sink is the most efficient supplying network,

**Proposition 1.4.1** *Singe-edge graph is the most efficient SPG supplying network.*

For single-edge graph, let $p_s^0$ be the source price, then $p_s^0 = a_t - (2b_t + c_s)X_s$. For general SPG, by induction, we show that the market clearance price for every firm is higher than $p_s^0$. The induction step is similar to the proof of Lemma 1.3.1, and the proof details can be found in Appendix 2.7.1.

Interpret $a_t$ as the *demand* of the market, the following proposition shows the relation between demand and efficiency.

**Proposition 1.4.2** *The market efficiency increases if the demand at the market increases or material cost at the source decreases.*

**Proof** From Lemma 1.3.1:

$$p_s = a_t - b_s X_s = a_s + d_s X_s \text{ (the given source price)}.$$

It follows that $X_s = \frac{a_t - a_s}{d_s + b_s}$, so the increasing demand at market ($a_t$) or decreasing cost at the source ($a_s$ or $d_s$) will make the supply chain more efficient. ∎

### 1.4.1 Components' Series Order

In this section, we examine the relationship between efficiency and local structure of an SPG, i.e., the order of *components*.

**Definition 1.4.2 (Component)** *$X$ is a component of $G$ if $X$ only contains one node or $X \subseteq G$ is an SPG whose head $s_X$ and tail $t_X$ satisfy $t_X \in C_S(s_X)$. Besides, $X$ contains all the nodes in $P(t_X) \cap C(s_X)$.*

If component $X$'s tail is $Y$'s head (or the reverse), then we say $X$ and $Y$ are series components. Note that we can extend the definition of the component by treating $S(X,Y)$ as a component too, while all the results in this section still hold.

Obviously, the efficiency of a supply chain is highly related to its components, and we define component efficiency as follows.

**Definition 1.4.3 (Component Efficiency)** *Component efficiency of $X$ is $\lambda(X, b_{t_X}) = \frac{b_{s_X}}{b_{t_X}}$.*

We can see measures the changes of slopes by component $X$, and it has high component efficiency if $\lambda(X, b_{t_X})$ is small. Let us first consider the simpler case that the processing cost is absent. As a result, the component efficiency is irrelevant to $b_{t_X}$. The proof is provided in Appendix 2.7.3.

Lemma 1.4.1 *Assume no processing cost in component $X$, then*

$$b_{s_X} = \lambda(X, b_{t_X}) = \lambda(X) b_{t_X}$$

*where $\lambda(X) \geqslant 2$ is a constant only relevant to the graph structure.*

Now consider the efficiency of series components $S(X,Y)$ and assume no processing cost in $X$ and $Y$, by Lemma 1.4.1:

$$\begin{aligned}
\lambda(S(X,Y), b_t) &= \lambda(X, \lambda(Y, b_t)) \\
&= \lambda(X)\lambda(Y) b_t \\
&= \lambda(Y)\lambda(X) b_t \\
&= \lambda(S(Y,X), b_t).
\end{aligned}$$

which means the order of series components does not matter, and we obtain the following theorem (proof detail is provided in Appendix 2.7.4.)

**Theorem 1.4.1** *Assume no processing cost, switching the order of series components does not change the efficiency.*

Now consider the case with processing cost:

$$\Pi_i = \sum_{j \in B(i)} p_j x_{ij} - p_i X_i - \frac{c_i}{2} X_i^2, \text{ where } c_i > 0.$$

If we change the order of series components, the total flow and the slope efficiency may vary as shown in this following example.

EXAMPLE 6 *Consider the price functions of source for the following two graphs, where* $c_a > 0, c_b = 0$, $p_t = a - bX_t$, *and* $p_s = 0$*:*

$$a \xrightarrow{x} b \xrightarrow{x} t$$

$$b \xrightarrow{x} a \xrightarrow{x} t$$

*Every edge is active in both graphs, price functions for the first graph are*

$$\begin{aligned} p_b &= a - 2bx, \\ p_a &= a - (4b + c_a)x = 0. \end{aligned}$$

*As a result, the total flow is* $x_1 = \frac{a - p_s}{4b + c_a}$. *While the price functions for the second graph are:*

$$\begin{aligned} p_a &= a - (2b + c_a)x, \\ p_b &= a - (4b + 2c_a)x = 0. \end{aligned}$$

*As a result, the total flow is* $x_2 = \frac{a - p_s}{4b + 2c_a} < x_1$. *It follows that the first graph is more efficient than the second one, and the series order of* $a$ *and* $b$ *does influence the efficiency.*

In the general case with processing cost, each component has a complex influence on the ratio of $b_s$ to $b_t$, and it is unclear to us what is the efficient algorithm to find the optimal series order of the components. Nevertheless, for some simple cases, we can see the pattern of optimal order.

**Proposition 1.4.3** *For a series composition of components* $X$ *and* $Y$*, suppose there is processing cost in* $X$*, but no processing cost in* $Y$*, then the composition with* $X$ *close to the source is more efficient than the composition with* $X$ *close to the sink.*

The proof is provided in Appendix 2.7.5.

One natural interpretation of the above result is the later the processing cost occurs, the worse the efficiency. At equilibrium, upstream firms will consider the cost from downstream. Therefore, the later cost hinders the incentive of upstream firms to supply more goods.

Suppose the supply chain is a straight line, the pattern is clearer, the processing cost $c_i$ is the only criteria to decide the optimal order. Without loss of generality, denote the optimal order as firm 0, 1, ..., $n-1$, $n$ from source 0 to sink $n$.

**Proposition 1.4.4** *In the most efficient order arrangement of a straight line model, firm $i$ has higher order than firm $j$ if and only if $c_i \leqslant c_j$, and this relation always holds:*

$$a_0 = a_n,$$
$$b_0 = 2^n b_n + \sum_{i=1}^{n} 2^i c_i.$$

The proof is provided in Appendix 2.7.6.

This indicates that it is always better to put the node with a higher cost closer to the source, and the fact is the processing cost will be amplified (exponentially) along the path from sink to source.

### 1.4.2 Series Insertion, Parallel Insertion

This section focuses on in which way and at what location, adding a component to a given supply chain network will change the efficiency. The two operations we are most interested in are series insertion and parallel insertion.

**Definition 1.4.4 (Series Insertion)** *An SPG $X$ is series-inserted into an SPG $G$ at node $i$ by setting $s_X = i$, $t_X = i$.*

**Definition 1.4.5 (Parallel Insertion)** *An SPG $Y$ is parallel-inserted into an SPG $G$ at component $X$ by setting $s_Y = s_X$ and $t_Y = t_X$.*

The intuition is parallel insertion provides another path for the flow in the supply chain, while series insertion just makes the supply chain redundant, and we have the following theorem illustrating our intuition.

**Theorem 1.4.2** *Series insertion always decreases the total flow, while parallel insertion always increases the total flow.*

The proof is provided in Appendix 2.7.7.

Base on the fact that series insertion is always bad, while parallel insertion is always good, the next question is, given components, where is the most efficient location to insert?

To analyze the changes in efficiency from different parallel insertion location, we can start with a special case, where $G$ can be written as a series composition of two components.

Lemma 1.4.2 *Suppose $G = S(X_1, X_2)$, then $P(G, Y)$ is more efficient than parallelly inserting $Y$ at $X_1$ and also more efficient than parallelly inserting $Y$ at $X_2$.*

The proof is provided in Appendix 2.7.8 and it can be extended to general SPG as mentioned in the following theorem.

**Theorem 1.4.3** *Parallel insertion into the entire SPG is more efficient than parallel insertion into a component of the SPG.*

**Proof** Proof by induction, starting from the smallest series of components, it is always better off by parallel insertion at the head and tail nodes by Lemma 1.4.2, and we can repeat this until stopping at the global parallel insertion. ■

This theorem can be interpreted as global parallel insertion will bring more competition to the supply chain network than local parallel insertion. As a result, the network is more efficient after global insertion.

### 1.4.3 Firm Location and Individual Utility

This section focuses on the firm's utility at equilibrium. Specifically, how does the position of a firm in the network influence its utility at equilibrium? To address this question, we first check the result of a simple example.

Example 7 (Firm Utility in Straight Line)

$$p_s = 0 \quad s \xrightarrow{x} a \xrightarrow{x} t \quad p_t = 1 - X_t$$

*Assume processing cost is 0. Price at firm $a$ and $s$ are $p_a = 1-2x$ and $p_s = 1-4x$. Therefore, the utilities are $\Pi_a = (p_t - p_a)x = x^2$ and $\Pi_s = 2x^2 = 2\Pi_a$.*

The above example shows an intuition of the location advantage that the firm closer to the source may have higher utility than its downstream buyers. However, this is not always true in SPG, especially when there is strong competition among upstream buyers (i.e., MS case). To gain a deeper intuition, we would say the upstream firm which controls all the flow of its downstream firm has a relatively better utility at equilibrium. Therefore, we introduce the following new definition.

**Definition 1.4.6 (Dominating Parent)** *$i$ is a dominating parent of $j$ if all the flow from source to $j$ must go through $i$.*

As in Example 4, $a$ is a dominating parent of $b$ and $g$, but neither a dominating parent of $h$ nor a dominating parent of $i$.

For firm $i$, the utility is

$$\begin{aligned}
\Pi_i &= \sum_{j\in B(i)} (p_j - p_i)x_{ij} - \frac{c_i}{2}X_i^2 \\
&= \sum_{j\in B(i)} (b_i X_i + \sum_{k\in C_P(i)} b_k X_k - b_j X_j - \sum_{k\in C_P(j)} b_k X_k)x_{ij} - \frac{c_i}{2}X_i^2.
\end{aligned}$$

By using the coefficient relation between buyer and seller as in equation SS, MS, and SM, we can find the closed-form of the utility. The proof is provided in Appendix 2.7.9.

LEMMA 1.4.3 *The utility at equilibrium can be written as*

$$\Pi_i = \frac{1}{2}(b_i + \sum_{k \in C_P(i)} b_k) X_i^2. \tag{1.9}$$

Based on the utility function, we can prove the following key theorem which shows the location advantage of a dominating parent. Namely, if a firm controls the other firm's flow in the supply chain, then its utility is at least twice as much as its child. The proof is provided in Appendix 2.7.10.

**Theorem 1.4.4** *If firm $i$ is a dominating parent of firm $j$, then firm $i$ has at least twice as much utility as firm $j$.*

The following corollary shows that the seller benefits a lot from the competition among the buyer side, and the proof is provided in Appendix 2.7.11.

COROLLARY 1.4.1 *In the SM case, the utility of the seller is larger than the utility sum of all the buyers.*

To sum up, we proved a dominating parent always has better utility, and the *double utility rule* will hold, which demonstrates the great value of controlling the upstream flows in the real world.

## 1.5 Equilibrium in Generalized Series Parallel Graph

In this section, we discuss the equilibria properties in the extension cases when the series-parallel graph has multiple sources or sinks. In particular, we will show:

- Multiple-sources-and-single-sink SPG: There exists a unique equilibrium, and it can be found in polynomial time.
- Single-source-and-multiple-sinks SPG: Price function of a firm may be piecewise linear under simple settings. Besides, there may exist multiple equilibria.
- Multiple-sources-and-multiple-sinks SPG: There may exist multiple equilibria, or there is no equilibrium.

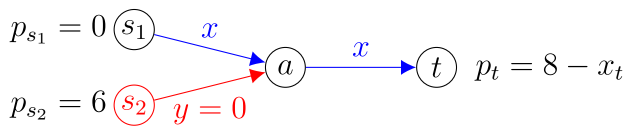
### 1.5.1 Multiple Sources and Single Sink

A series-parallel graph with multiple sources and single sink (MSPG) is defined as follows.

**Definition 1.5.1 (MSPG)** *$G$ is* multiple-source-and-single-sink SPG *if it can be constructed by deleting the source node of an SPG and setting the adjacent nodes of the source as the new source nodes.*

Assume every source producer makes decision simultaneously. In contrast to SPG that all edges are active, there may exist inactive edges in MSPG.

Example 8 (Inactive Edges)




*By Algorithm 1, Price functions at firm a is $p_a = 8 - 2X_a$. By solving the LCP as in section 1.3.3, the equilibrium flow is $x = 2, y = 0$, where firm $s_2$ and edge $x_{s_2a}$ are inactive.*

By the proof 2.6.2 of Lemma 1.3.1 (SM case), if a firm is active, all the sub-flows are active too. Therefore, it is sufficient to identify all the inactive edges by check the seller's activity status, and here is an algorithm to identify all the inactive edges in MSPG:

**Algorithm 3** : Determinate Inactive Edges

1: Similar to Algorithm 1, compute the price function of all nodes.
2: Solve the convex optimization problem CQP at the source nodes, get the equilibrium flow $x_{sj}$ where $j \in B(s)$ for each source node $s$.
3: For any firm $k$, if all of its inflow edges are red, also mark $k$ and its outflow edges as red. Repeat that until no new red firm or edge appears.
4: Firms and edges are inactive if and only if it is marked as red.

Similar to the SPG procedure, we can apply Algorithm 1 and Algorithm 2 to compute the price and quantities at equilibrium.

**Theorem 1.5.1** *For MSPG, there exists a polynomial time algorithm to solve the equilibrium flow and price, and the equilibrium is unique.*

The proof is quite similar to Theorem 1.3.1 and is omitted here. Note that uniqueness is because flow quantity is a solution of CQP (Lemma 1.3.2).

### 1.5.2 Single Source and Multiple Sinks

In this section, we focus on the extension of multiple sinks, and the definition is similar to Definition 1.5.1.

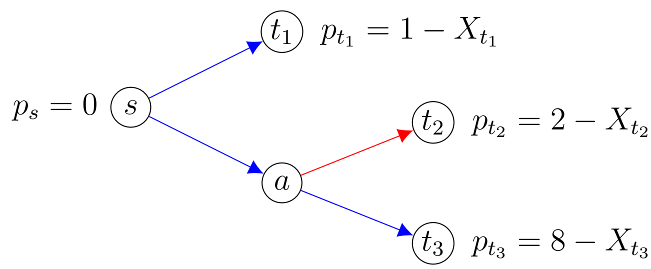
**Definition 1.5.2** *$G$ is* single-source-and-multiple-sinks SPG *if it can be constructed by deleting the sink node of an SPG and setting the adjacent nodes of the sink as the new sink nodes.*

First, we consider a special case that all markets have the same demand $a_t$, then all markets are *active*, i.e., every market has a positive incoming flow. The proof is provided in Appendix 2.8.1.

**Theorem 1.5.2** *If all markets have the same demand, then all markets are active, and there exists a unique equilibrium.*

However, one major difference multiple sinks cast to SPG is that depending on the selling price from upstream, and the ending markets may be *inactive*, that is, the incoming quantity is zero, while the single ending market is always active in SPG. For example,

Example 9 (Markets Activities)




*Since $a_{t_1} > p_s$, it is clear that market $t_1$ is active. Suppose market $t_2$ is active, market clearance price function at $a$ is $p_a = 5 - X_a$. When source $s$ makes decision, note that flow $x_{st_1}$ and $x_{sa}$ can be handled independently, it is easy to see the optimal decision that maximizes the utility $(5 - X_a)X_a$ of $s$ from $a$ is $X_a = 2.5$ and $p_a = 2.5 > a_{t_2}$, contradicting to market $t_2$ is active. Therefore, market $t_2$ is inactive, even though it has higher demand than market $t_1$.*

Note that, the above example is against the intuition that the market with higher demand is more likely to be active ($t_2$ is inactive while $t_1$ is). While the truth is not only market demand, but also the competitors and network structure influence the market activity. Namely, market $t_2$ is inactive because it has a longer supply chain than $t_1$ and a strong competition between $t_3$. As a result, it is less favorable than $t_1$ and $t_3$.

Based on the activity status of the ending markets, we introduce two types of processing strategies for upstream firms.

**Definition 1.5.3 (Low Price Strategy)** *Firm processes relatively large quantity of goods at a relatively low price, such that all the markets are active.*

**Definition 1.5.4 (High Price Strategy)** *Firm processes relatively small quantity of goods at a relatively high price, such that some markets are inactive.*

Note that the firm's decision of strategies only depends on individual utility maximization. Because of various choice of strategies, we will see the price functions are piecewise linear in this case. Furthermore, some counterintuitive results will occur, i.e., the increase of demand may result in the decrease of total flow and social welfare (comparing to Proposition 1.4.2). To understand these differences, it is helpful to consider an example as in Figure 1.3, where the two supply chain networks have identical structure but different market demands.

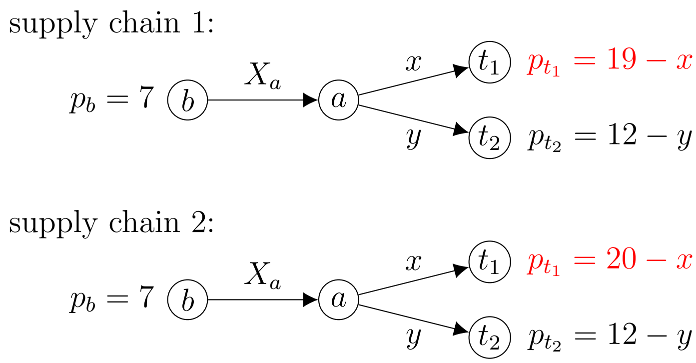



Fig. 1.3.: Multiple Sinks Supply Network

It seems that supply chain 2 with higher market demand should have larger flow and social welfare. However, the truth is supply chain 1 is more efficient. To explain this, let us check the market clearance price at $b$ and $a$ first as in Figure 1.4. Note that the source firm $b$ has two strategies when $p_b = 7$, and both low and high price strategies are feasible. Interestingly, when $a_{t_1} = 20$, the utility of $b$ is maximized by choosing high price strategy and only market $t_1$ is active. However, when demand at market $t_1$ drops, low price strategy is preferred by $b$.

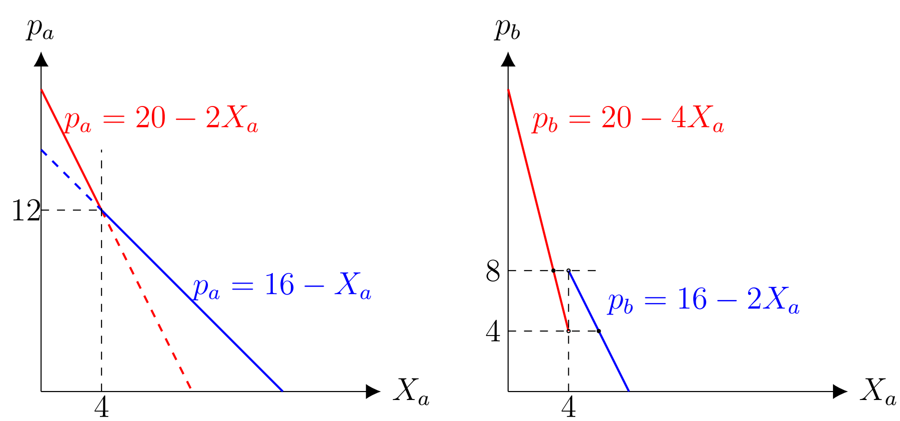



Fig. 1.4.: Piecewise Linear Price Functions of Supply Chain 2

By fixing demand at market 2 and adjusting the demand at market 1 ($a_{t_1}$), Figure 1.5 shows the numerical results of firm $b$'s corresponding surplus, consumer surplus, total flow, and social welfare. Note that the intersecting point at $a_1 \approx 19.5$ shows that increasing demand at market hurts the supply chain efficiency.

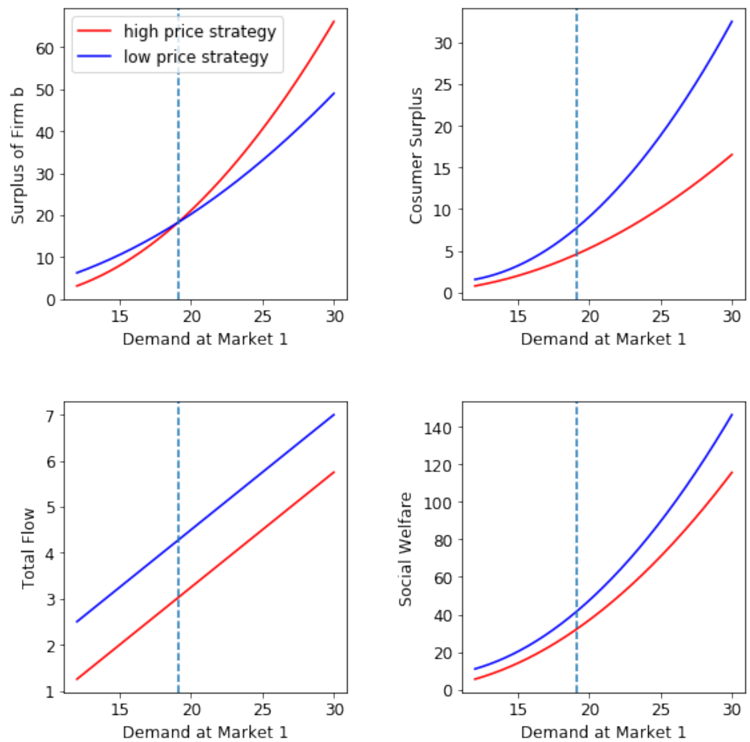



Fig. 1.5.: Price Strategies Simulation

**Remark.** For the supply chain networks in Figure 1.3, we have the following results:

- Supply chain under low price strategy is always more efficient than under high price strategy.
- When the demand difference between two markets is small enough, low price strategy gives better payoff for source firm $b$. If the difference is large enough, high price strategy gives better payoff for source firm $b$.
- Low price strategy always produces a higher total surplus of firms and consumers. Hence, social welfare is also higher.

In short, low price strategy is preferred by $b$ if the demand difference is not significant. Besides, with low price strategy, everyone is usually better off. For more interpretation of these results, please check Appendix 2.8.2.

When upstream chooses the optimal strategy and flow, there may exist multiple equilibria for downstream firms. Details are in Example 2.12.7.

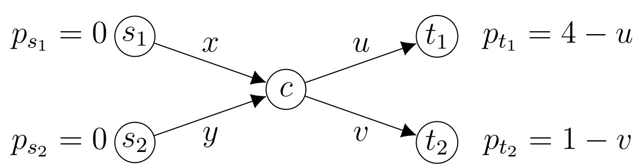
### 1.5.3 Multiple Sources and Multiple Sinks

In the multiple sources and multiple sinks cases, the problem may become intractable as shown in the following examples:

- Multiple pure strategy equilibria exist (Example 10).
- No pure strategy equilibrium exists (Example 11).

Therefore, it is difficult to analyze the behavior of the firms in the supply chain without any further assumption in this case.

Example 10 (Multiple pure strategy equilibria)




*Assume no processing cost.* $\Pi_1^h$ *is the utility of* $s_1$ *with high price strategy and* $\Pi_1^l$ *is the utility of* $s_1$ *with low price strategy. The notations for* $s_2$ *are similar.*

- *Suppose restricted to high price strategy, the optimal quantities are* $x = y = \frac{2}{3}$*, then*

$$\Pi_1^h = \Pi_2^h = \frac{8}{9}.$$

  *If* $s_2$ *increases supply to low price strategy level* ($y' = \frac{11}{12}$)*, his optimal payoff at the new low price strategy is*

$$\Pi_2^{l'} = \frac{121}{144} < \Pi_2^h.$$

  *Thus, exists equilibrium at high price strategy.*

- *Suppose restricted to low price strategy, the optimal quantities are* $x = y = \frac{5}{6}$*, then*

$$\Pi_1^l = \Pi_2^l = \frac{25}{36}.$$

  *If* $s_2$ *decreases supply to high price strategy level* ($y' = \frac{7}{12}$)*, his optimal payoff at the new high price strategy is*

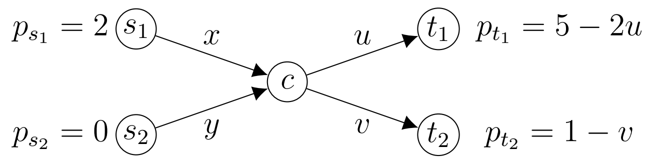
$$\Pi_2^{h'} = \frac{49}{72} < \Pi_2^l.$$

  *Thus, exists equilibrium at low price strategy.*

*In summary, both high and low price strategies are equilibria. Computation details can be found in Appendix 2.12.7*

The following example shows that it is possible that no equilibrium exists in the multiple sources and multiple sinks cases.

Example 11 (No pure strategy equilibrium)




*Assume no processing cost.* $\Pi_1^h$ *is the utility of* $s_1$ *with high price strategy and* $\Pi_1^l$ *is the utility of* $s_1$ *with low price strategy. The notations for* $s_2$ *are similar.*

- *Firm* $s_1$ *never accepts low price strategy, because when market* $t_2$ *is active* $p_c$ *has to be smaller than* 1*, but* $p_{s_1} > 1 > p_c$*.*

- *If firm* $s_1$ *is not active* $(x = 0)$*, firm* $s_2$ *will prefer high price strategy which gives a higher utility,*

$$\Pi_2^l = \frac{49}{32} < \frac{50}{32} = \Pi_2^h.$$

  *while the price function at* $c$ *is greater than the material cost of firm* $s_1$*,*

$$p_c = 2.5 > p_{s_1}.$$

  *Therefore, this is not an equilibrium because firm* $s_1$ *will prefer participating the supply network and* $x > 0$*.*

- *If firm* $s_1$ *is active* $(x > 0)$*, then assume they agrees on a local optimal at high price strategy. However, firm* $s_2$ *will prefer increasing production and switching to low price strategy because*

$$\Pi_2^h = \frac{49}{36} < \frac{50}{36} = \Pi_2^l.$$

  *Thus, it is not an equilibrium either.*

*In summary, neither high nor low price strategy exists equilibrium. Computation details can be found in Appendix 2.12.8.*

## 1.6 Conclusion

We considered a network model of sequential competition in supply chain networks. Our main contribution is that when the network is series-parallel, the model is tractable and allows for a rich set of comparative analysis. In particular, we provide a polynomial time algorithm to compute the equilibrium, and the algorithm helps us to study the influence of the network to the total flow of the equilibrium. Furthermore, we show that slightly extending the network structure beyond series-parallel graphs makes the model intractable. Several questions are left for future research such as extending the model to capture uncertainty, risks, and asymmetric information.

# 2. DELEGATION STRUCTURE

## 2.1 Introduction

The decision of delegation structure affects the cost of the principal and the effort status of every agent in the production chain in different ways. The principal may prefer delegation because that saves the cost of monitoring. However, the conflict interests between the middle agents with the principal may result in an insufficient incentive for downstream agents under delegation. In this article, we investigate and fully characterize the trade-off of the delegation, and provide algorithms to compute the optimal delegation structure for the principal.

Our model has three main features. First, we consider the production chain as a sequential process, where a product is processed from raw material at the initial agent to the final product at the principal. During this process, the agents decide the effort levels sequentially. Second, the effort level is unobservable, but it's possible to monitor the quality of the output product from each agent. Third, and most importantly, the principal could access the intermediate product quality information by signing a contract with the corresponding agent. Therefore, there is no information advantage for the middleman.

In this setting, the principal first designs a delegation structure for the product chain, and start signing contracts sequentially from the top to the down levels. After received contracts, the agents begin exerting efforts sequentially along the production chain. Due to the difficulty of monitoring the efforts, while the observable quality is an aggregation of the effort, the predecessor's output quality, and unknown environmental effect, problems of free riding and moral hazard arise in this context. Hence, how to design an efficient contract structure, at the minimum cost, to induce the effort from every agent becomes the primary concern of the principal.

To find the optimal contract structure, we start our analysis with a three agents model, including the principal, agent 1 and agent 2. A practical instance of this model can be a production chain of building satellites: the power system must be finished by agent 1 first. Then, after knowing the engine's capability and the maximum deliverable mass, agent 2 can start to design the rest of the satellite. In the end, the principal expects a functional satellite ready to launch. To motivate every agent along the production chain, the principal can sign a direct contract with both agents and observes their output signals.

Meanwhile, it is also an option to give agent 2 more power and make him responsible for agent 1's action. Specifically, in the delegation case, agent 2 not only needs to decide his effort but also accountable for the subcontract with the downstream agent. Otherwise, shirking may happen and eventually harms agent 2 task completeness.

One significant difference between our study with the other literature is that we treat the principal and the delegating middle agent at the same fair position when monitoring the downstream agents. For example, both of the principal and agent 2 can observe the same correct output signal from 1, and there is no additional cost for the principal even if to control every agent directly. Under this setup, it may appear to be that the delegation will not help since the middle agent does not have any advantage on inducing the downstream agents' effort but has a personal objective inconsistent with the principal. However, we will show sometimes leaving more responsibilities to the middle agents may help the principal save the cost of incentive when signing contracts.

Our paper studies a model of sequential network game motivated by production chain network applications. We consider the agents are risk neutral and can't be punished. To study such a delegation model, one needs to analyze subgame perfect equilibria. After received a contract from the principal, to make an optimal decision about personal effort and subcontract, agent 2 needs to internalize the decision of agent 1.

Once the subgame equilibrium is solved, we can characterize the threshold for the principal to decide whether direct control or delegation in a three agents model in Section 2.2. Our primary goal is to understand the value of delegation and how to utilize agents' incentive through different contract structures fully. Moreover, our study illustrates the trade-off of using delegation. On the one hand, through delegation, the principal shifts the contract cost of the downstream agents to the middle agents. On the other hand, the principal gives up the ability to observe those intermediate signals and loses control over those delegated agents.

Our main finding in this paper is presenting a new approach to demonstrate the value and trade-offs of the delegation. In contrast, the other work assuming asymmetric information, i.e., [14, 15], we assume the principal can access full information same as the middleman. Furthermore, we provide thresholds for the principal to make the optimal decision between direct control and delegation under different conditions. We explore the influence of various parameters over the delegation decision of the principal in different situations through a comparative study.

After solving the problem in a three agents model in Section 2.2, under some mild assumptions, we extend this model to various complex process structures, including path and tree in Section 2.3 and 2.4. We also developed a polynomial time algorithm to obtain the optimal delegation structure and contract payments.

**Related work:** Our paper assumes the effort is unobservable and concerning the efficient contract structure based on the quality signals, which is closely related to the literature on Moral Hazard on teamwork, including [16, 17, 18, 19]. Comparing to their work, our paper focuses more on the sequential processing and optimal delegation structure along the production chain. However, [17] considers two identical agents in the team and focus more on the benefits from different contract conditions. Meanwhile, [16, 18, 19] stress on different types of agents in the group and the effects of matching between different types. In contrast, the agents in our paper belong to the same type but have different parameters about the success probability.

Our work focus on the value of delegation by using the middleman, [20, 21] also investigate the role of middleman and its effects over the overall network. Our paper's setting is related to [22], which considers uncertainty over the agents preferences and provide an optimal delegation set. However, he restricts the set of feasible delegation sets to intervals. Our work is also related to the papers about relational contracts within and between organizations. For instance, [23] studies the design of self-enforced in the presence of moral hazard and hidden information.

For a similar network structure, [4, 12] analyze bargaining games. However, our paper assumes zero bargaining power when the downstream agent receives the contract from the upstream agent but considers more variation on contract structure by delegation.

## 2.2 General Three Agents

### 2.2.1 Model Description

We consider a three agents sequential working process as in Fig. 2.1, where the work is initiated at the agent 1. After agent 1's task is done, it will be passed over to agent 2, and eventually to the last agent (the principal). During the process, each agent can decide making effort or not, while this effort is costly and **unobservable** to the others. However, a binary signal $s_k \in \{0, 1\}$ which indicates the task completeness is **observable** to the next node, More precisely, agent 2 can observe $s_1$ and the principal can observe $s_2$ after the task of agent 1 or 2 is done.

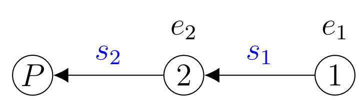



Fig. 2.1.: Process Path

For agent 1, the probability of success, $P(s_1 = 1)$, is related to

- Personal effort $e_1$,
- Environmental random effect $r_1$.

For agent 2, the probability of success, $P(s_2 = 1)$, is related to

- Signals from agent 1, $s_1$,
- Personal effort, $e_2$,
- Environmental random effect, $r_2$.

We assume the success probability in each condition is a public information, and the following values are given to every agents in the production chain,

$$P(s_1 = 1|e_1), \text{where } e_1 \in \{0, 1\}, \tag{2.1}$$
$$P(s_2 = 1|s_1, e_2), \text{where } e_2, s_1 \in \{0, 1\}. \tag{2.2}$$

Without loss of generality, we can rewrite the above probabilities in the following form,

$$P(s_1 = 1|e_1) = \alpha_1 e_1 + \gamma_1, \tag{2.3}$$
$$P(s_2 = 1|s_1, e_2) = \alpha_2 e_2 + \beta_2 s_1 + \tau_2 s_1 e_2 + \gamma_2. \tag{2.4}$$

and the value of the parameters $\alpha, \beta, \tau, \gamma$ are common information. It's also fair to assume that the effort and good signals indeed help complete the task,

$$\begin{aligned}
&P(s_1 = 1|e_1 = 1) > P(s_1 = 1|e_1 = 0),\\
&P(s_2 = 1|s_1, e_2 = 1) > P(s_2 = 1|s_1, e_2 = 0), \forall s_1 \in \{0, 1\},\\
&P(s_2 = 1|s_1 = 1, e_2) > P(s_2 = 1|s_1 = 0, e_2), \forall e_2 \in \{0, 1\}.
\end{aligned}$$

which is equivalent to assume $\alpha, \beta, \tau > 0$.

The priority goal of the principal is to achieve success at the final task, i.e., $s_2 = 1$. Since agents' effort is costly and unobservable, the only way for the principal to induce the effort is by signing contracts based on the output signal $s_1, s_2$. There are two options for the contract structure[1], i.e., either directly control both of agents, or delegate agent 1 to agent 2 as in Fig. 2.2, where the solid black line means contract direction, and the blue dashed line means the process direction:

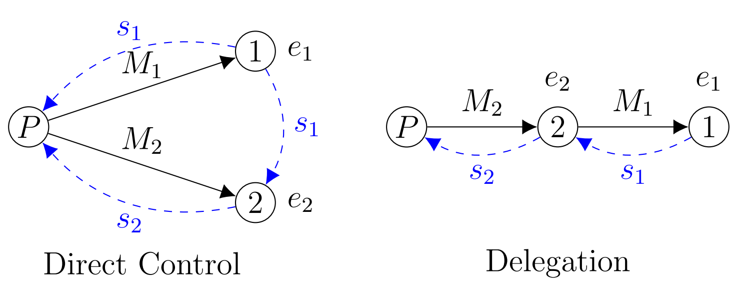



Fig. 2.2.: Two Delegation Structures

As in the Figure 2.2, the principal can direct control (sign contract with) agent 1, 2. In this case, because the contract with agent 1, the principal obtain the ability to

[1]delegate agent 2 to agent 1 is always inefficient, proved in next section.

monitor agent 1's output signal $s_1$. The decision time line of direct control is plotted in Fig. 2.3.

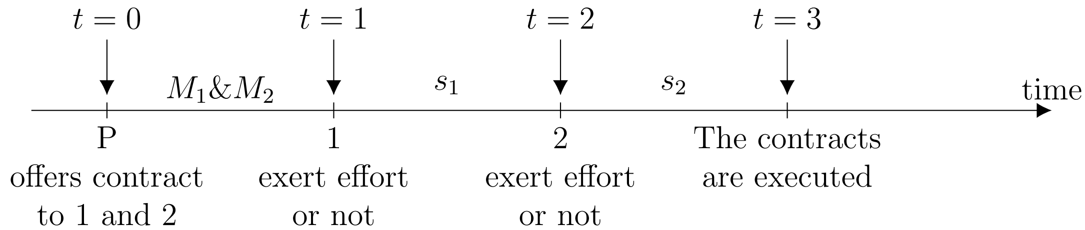



Fig. 2.3.: Timing of Contracting of Direct Control

Another option is only signing contract with agent 2, and agent 2 has the freedom to decide whether signs a subcontract to motivate agent 1. In another words, the principal gives up all the control over agent 1, and cannot observe the contract detail between agent 2 and 1. The decision time line of delegation is plotted in Fig. 2.4.

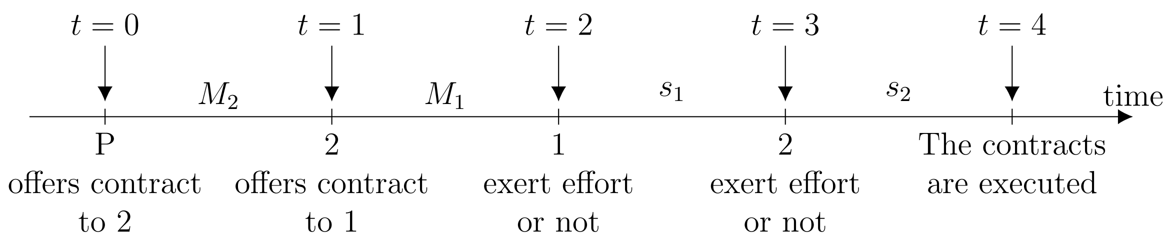



Fig. 2.4.: Timing of Contracting of Delegation

Assume agents are **risk neutral** and have zero liability. In the production chain, agents make decisions sequentially to maximize the individual utility. The goal of the principal is to maximize the success probability with the lowest cost. Namely, the principal wants to minimize the cost under the condition that every single agent in the process tree ha incentive to work. The question is how much is the necessary contract payment that the principal should sign the agents, and what's the best structure, delegation or direct control?

### 2.2.2 Preliminary

Agent 1's expected utility function is

$$\pi_1(e_1|M_1) = P(s_1 = 1|e_1)M_1 - c_1 e_1.$$

Because agent has zero liability, the efficient contract always set the payment to 0 when the task is failed, and the contract to agent 1 has the following form:

$$r_1 = \begin{cases} M_1, & \text{if } s_1 = 1, \\ 0, & \text{if } s_1 = 0. \end{cases} \tag{2.5}$$

We call $M_1$ as the contract payment, and to induce agent 1 to make effort, $M_1$ must satisfy $\pi_1(e_1 = 1|M_1) \geq \pi_1(e_1 = 0|M_1)$. From the this condition, we can compute the minimum contract payment to agent 1,

$$M_1^0 = \frac{c_1}{P(s_1 = 1|e_1 = 1) - P(s_1 = 1|e_1 = 0)} = \frac{c_1}{\alpha_1}. \tag{2.6}$$

When agent 2 is making effort decision, the expected utility function is,

$$\pi_2(e_2|s_1, M_2) = P(s_2 = 1|s_1, e_2)M_2 - c_2e_2.$$

In contrast to agent 1, the utility of agent 2 also depends on the result of $s_1$. The difference of effort is

$$\begin{aligned} \Delta\pi_2(s_1|M_2) &= \pi_2(e_2 = 1|s_1, M_2) - \pi_2(e_2 = 0|s_1, M_2) \\ &= (P(s_2 = 1|s_1, e_2 = 1) - P(s_2 = 1|s_1, e_2 = 0))M_2 - c_2e_2 \\ &= (\alpha_2 + \tau_2 s_1)M_2 - c_2e_2. \end{aligned}$$

By the incentive condition $\Delta\pi_2(s_1|M_2) \geq 0$. When $s_1 = 0$, the minimum contract payment is

$$M_2^+ = \frac{c_2}{\alpha_2}. \tag{2.7}$$

When $s_1 = 1$, the minimum contract payment is

$$M_2^- = \frac{c_2}{\alpha_2 + \tau_2}. \tag{2.8}$$

Note that since we assume $\tau_2 \geq 0$, we have $M_2^- \leq M_2^+$, and the following proposition.

**Proposition 2.2.1** *Minimum payment for agent 2 to make effort is larger when* $s_1 = 0$.

Hence, both $s_1$ and $s_2$ are useful to design the contract with agent 2. In contrast, for contract with agent 1, signal $s_1$ is enough to motivate agent 1, and additional information from $s_2$ doesn't save the expected cost for the principal.

**Proposition 2.2.2** *In the direct control case, signal* $s_1$ *is sufficient for the principal to design the minimum cost contract with agent* 1.

After receiving incentive contract, and suppose everyone makes effort, We denote $P(1), P(2)$ as the success probability under effort,

$$P(1) = P(s_1 = 1|e_1 = 1), \tag{2.9}$$

$$\begin{aligned} P(2) &= P(s_2 = 1|e_1 = 1, e_2 = 1) \\ &= \sum_{s_1} P(s_2 = 1|s_1, e_2)P(s_1|e_1). \end{aligned} \tag{2.10}$$

Recall the delegation options for the principal (Fig 2.2), here we prove, the principal never considers delegating the upstream agent to the downstream agent,

**Proposition 2.2.3** *Delegate agent* 2 *to agent* 1 *is always inefficient.*

The proof is provided in Appendix 2.9.4.

Since the principal knows all the information as the other agents, it seems to be the principal has no benefit to delegate agent 1 under the control of agent 2. However, this is not always true, and the following example shows that delegation may be better than direct control.

Example 12 (Benefit of Delegation)
***Setup:*** *For agent* 1*, the effort cost is* $c_1 = 1$*,* $\alpha_1 = 0.4, \gamma_1 = 0.2$ *and corresponding successful probabilities are,*

$$P(s_1 = 1|e_1 = 0) = 0.2,$$
$$P(s_1 = 1|e_1 = 1) = 0.6.$$

*For agent* 2*, the effort cost is* $c_2 = 2$*,* $\alpha_2 = 0.2, \beta_2 = 0.5, \tau_2 = 0, \gamma_2 = 0.2$ *and corresponding successful probabilities are,*

$$P(s_2 = 1|s_1 = 0, e_2 = 0) = 0.2,$$
$$P(s_2 = 1|s_1 = 1, e_2 = 0) = 0.7,$$
$$P(s_2 = 1|s_1 = 0, e_2 = 1) = 0.4,$$
$$P(s_2 = 1|s_1 = 1, e_2 = 1) = 0.9.$$

***Computation:***

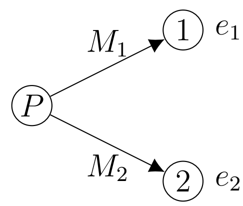
*By Equation 2.6, 2.7, 2.8, the minimum effort payment for agent* 1 *and agent* 2 *is*

$$M_1^0 = 2.5,$$
$$M_2^0 = M_2^+ = M_2^- = 10,$$

*and the successful probability when every agents commits effort by equation 2.9 is,*

$$P(1) = P(s_1 = 1|e_1 = 1) = 0.6,$$
$$P(2) = P(s_2 = 1|e_1 = 1, e_2 = 1) = 0.7.$$

*In the direct control case,*

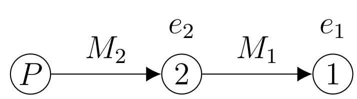



*Direct Control*

*the expected cost of direct control is,*

$$cost = P(1)M_1^0 + P(2)M_2^0 = 8.5.$$

*In the delegation case,*




*Delegation*

*in order to motivate agent* 2*, the payment needs to satisfy* $M_2 \geq M_2^0$*. If agent* 2 *signs subcontract with* 1*, we know the payment is* $M_1 = M_1^0$*, and the following is the agent* 2*'s utility when sign or not sign contract with* 1

$$\begin{aligned}
\pi_2(M_1 = M_1^0|e_2 = 1) \quad &= P(s_2 = 1|e_1 = 1, e_2 = 1)M_2 - P(s_1 = 1|e_1 = 1)M_1 - c_2 \\
&= 0.7M_2 - 0.6M_1 - c_2, \\
\pi_2(M_1 = 0|e_2 = 1) \quad &= P(s_2 = 1|e_1 = 0, e_2 = 1)M_2 - c_2 \\
&= 0.5M_2 - c_2.
\end{aligned}$$

*By the condition that* $\pi_2(e_2 = 1, M_1 = M_1^0) \geq \pi_2(e_2 = 1, M_1 = 0)$*, we have a lower bound for contract payment to agent* 2 *as* $M_2 \geq 7.5$*, which is already satisfied by* $M_2 = M_2^0 = 10$*. Therefore, in the delegation case, if the principal signs contract with payment* $M_2^0$*, the agent* 2 *will sign subcontract with agent* 1 *and exert personal effort. Finally, the expect cost of delegation is*

$$cost' = P(2)M_2^0 = 7 < cost.$$

*Thus, delegation is better than direct control with a lower expected cost for the principal.*

### 2.2.3 Delegation Threshold

In this section, we consider two delegation structures, direct control and delegation as in Fig 2.2. In the delegation case, the principal gives up the control over agent 1 and delegate him to agent 2. Therefore,s compared to the direct control, the principal is only able to observe $s_2$ in the delegation case. Here are a few trade-offs for choosing delegation,

- pro: save the contract cost with agent 1;
- con: may increase the contract cost with agent 2;
- con: less efficient cost with agent 2, due to lack of information about $s_1$.

To compare delegation with direct control, we also consider two direct control case

- with both signals, the principal can sign contract with agent 2 based on $s_1$ and $s_2$;
- with a single signal, the agent 2 only accepts contract conditional on his performance $s_2$.

In the following part, we will analyze the expected cost in each case and the parameters and conditions that influence the principal's decision over different structures.

**Direct control with single signal.**

Suppose the contract payment to agent 2 only depends on $s_2$ as follows:

$$r_2 = \begin{cases} M_2, & \text{if } s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

To cover the worst case scenario $s_1 = 0$, and always induce agent 2's effort, the principal has to set the contract payment as $M_2 = M_2^+$. Thus, the expect cost of the principal is

$$cost_1 = P(1)M_1^0 + P(2)M_2^+. \tag{2.11}$$

Recall that $M_1^0 = \frac{c_1}{\alpha_1}, M_2^+ = \frac{c_2}{\alpha_2}$ are computed at Equation 2.6, 2.7.

**Direct control with both signals.**

Suppose the previous signal $s_1$ can be leveraged to design the contract with agent 2, then agent 2's contract payment is,

$$r_2 = \begin{cases} M_2^-, & \text{if } s_1 = 1, s_2 = 1; \\ M_2^+, & \text{if } s_1 = 0, s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

Recall that $M_2^- = \frac{c_2}{\alpha_2+\tau_2}, M_2^+ = \frac{c_2}{\alpha_2}$ are computed at Equation 2.8, 2.7. Meanwhile, the contract payment to agent 1 is still $M_1^0 = \frac{c_1}{\alpha_1}$.

The expected cost in this case is

$$\begin{aligned} cost_2 &= P(s_1 = 1|e_1 = 1)M_1^0 + P(s_2 = 1, s_1 = 0|e_1 = 1, e_2 = 1)M_2^+ \\ &\quad + P(s_2 = 1, s_1 = 1|e_1 = 1, e_2 = 1)M_2^- \\ &= P(1)M_1 + P(s_1 = 0|e_1 = 1)P(s_2 = 1|s_1 = 0, e_2 = 1)M_2^+ \\ &\quad + P(s_1 = 1|e_1 = 1)P(s_2 = 1|s_1 = 1, e_2 = 1)M_2^-. \end{aligned} \tag{2.12}$$

Since direct control with both signals allows the principal to design more flexible and efficient contracts, we know $cost_2 \leq cost_1$ always holds, and have the following proposition,

**Proposition 2.2.4** *Direct control with both signals is always better than direct control with a single signal.*

**Delegation.**

In the delegation case, since the principal only observes $s_2$, the contract structure is

$$r_2 = \begin{cases} M_2, & \text{if } s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

Because the principal only has contract with agent 2, the expected cost in this case is

$$cost_3 = P(2)M_2. \tag{2.13}$$

The question is what's the minimum contract payment $M_2$ such that agent 2 will be motivated to sign subcontract with agent 1 and exert personal effort. Recall the decision timeline in Fig 2.4, similar to the case of direct control with a single signal, the minimum payment for agent 2 committing personal effort in the worst-case scenario in the effort stage is

$$M_2 \geq M_2^+.$$

Given this is satisfied, in the contract, agent 2 only needs to consider the expected utility in the following two conditions when making subcontract decision to agent 1,

- No subcontract, and commits personal effort later

$$\pi_2(M_1 = 0|e_2 = 1) = P(s_2 = 1|e_1 = 0, e_2 = 1)M_2. \tag{2.14}$$

- Subcontract, and commits personal effort later

$$\pi_2(M_1 = M_1^0|e_2 = 1) = P(s_2 = 1|e_1 = 1, e_2 = 1)M_2 - P(s_1 = 1|e_1 = 1)M_1 - c_2. \tag{2.15}$$

By the condition that $\pi_2(M_1 = M_1^0|e_2 = 1) \geq \pi_2(M_1 = 0|e_2 = 1)$, we get another low bound for the contract payment for agent 2,

$$M_2 \geq \frac{P(1)M_1^0}{\alpha_1(\beta_2 + \tau_2)}.$$

The following lemma gives the minimum contract payment to agent 2 in the delegation case.

Lemma 2.2.1 *In the delegation case, the minimum contract payment to agent 2 is*

$$M_2 = \max\{M_2^+, \frac{P(1)M_1^0}{\alpha_1(\beta_2 + \tau_2)}\}.$$

The proof details is provided in Appendix 2.9.1. Therefore, we have expected cost of delegation as follows,

$$cost_3 = P(2)\max\{M_2^0, \frac{P(1)M_1^0}{\alpha_1(\beta_2 + \tau_2)}\}. \tag{2.16}$$

Given the expected cost in three cases 2.11, 2.12, 2.16, it's clear that the principal prefers delegation to direct control with single signal when $cost_1 \geq cost_3$, and the following theorem gives the specific threshold.

**Theorem 2.2.1** *The principal prefer delegation to direct control with single signal if the following inequality is satisfied.*

$$\frac{(\alpha_2 + (\alpha_1 + \gamma_1)(\beta_2 + \tau_2) + \gamma_2)c_2/\alpha_2}{(\alpha_1 + \gamma_1)c_1/\alpha_1} \geq \frac{\alpha_2 + \gamma_1(\beta_2 + \tau_2) + \gamma_2}{\alpha_1(\beta_2 + \tau_2)}. \tag{2.17}$$

Similarly, if condition $cost_2 \geq cost_3$ is satisfied, then the principal prefers delegation to direct control with both signals, and we derive the thresholds in the following theorem,

**Theorem 2.2.2** *The principal prefer delegation to direct control with both signals if inequality 2.18, 2.19 is satisfied.*

$$\frac{c_1}{\alpha_1\sigma_2} \geq \frac{\tau_2 c_2}{\alpha_2(\alpha_2+\tau_2)}, \tag{2.18}$$

$$(1-\sigma_1)(\alpha_2+\gamma_2)\frac{c_2}{\alpha_2} + \sigma_1\sigma_2\frac{c_2}{\alpha_2+\tau_2} \geq \frac{\sigma_1(\alpha_2+\gamma_2)}{\alpha_1(\beta_2+\tau_2)}\frac{c_1}{\alpha_1}. \tag{2.19}$$

*where* $\sigma_1 = \alpha_1+\gamma_1, \sigma_2 = \alpha_2+\beta_2+\tau_2+\gamma_2$.

### 2.2.4 Comparative Statistics

Given the thresholds from the last section, this section will analysis the impact of each parameters over the principal's choice of delegation structure. Recall the probability functions 2.3,

$$\begin{aligned} P(s_1 = 1|e_1) &= \alpha_1 e_1 + \gamma_1, \\ P(s_2 = 1|s_1, e_2) &= \alpha_2 e_2 + \beta_2 s_1 + \tau_2 s_1 e_2 + \gamma_2. \end{aligned}$$

We first compare the delegation with direct control with single signal, whose contracts to agent 2 both only depend on $s_2$. The threshold 2.17 by Theorem 2.2.1 is,

$$\frac{(\alpha_2+(\alpha_1+\gamma_1)(\beta_2+\tau_2)+\gamma_2)c_2/\alpha_2}{(\alpha_1+\gamma_1)c_1/\alpha_1} \geq \frac{\alpha_2+\gamma_1(\beta_2+\tau_2)+\gamma_2}{\alpha_1(\beta_2+\tau_2)}.$$

If $\alpha_1$, $\beta_2$, $\tau_2$ or $c_2$ increases, the left hand side increases and right hand side decreases. Therefore, the principal prefers delegation if

- agent 1's effort has a large impact on the success rate of the first task ($\alpha_1$);
- the success of the first task has a large impact on the final success ($\beta_2$);
- the success of the first task makes the effort of agent 2 much more valuable: $\tau_2$;
- agent 2 has a high cost of effort ($c_2$).

Meanwhile, if $c_1$, $\gamma_1$, or $\alpha_2$ increases the left hand side decreases and right hand side increases. Therefore, the principal prefers direct control if

- agent 1 has a high cost of effort ($c_1$);
- agent 1 has a high probability of success without effort ($\gamma_1$);

- agent 2's personal effort has a high impact on the final success ($\alpha_2$).

Now let's compare the delegation with direct control with both signals. By Theorem 2.2.2, when $M_2 = M_2^+$ or

$$\frac{c_1}{\alpha_1(\alpha_2+\beta_2+\tau_2+\gamma_2)} \geq \frac{\tau_2 c_2}{\alpha_2(\alpha_2+\tau_2)}. \tag{2.20}$$

only threshold 2.18 is active, and we call this threshold as **efficient contract threshold**,

$$\frac{c_1}{\alpha_1} + (\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2+\tau_2} \geq (\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2}.$$

and when $M_2 = \frac{P(1)M_1^0}{\alpha_1(\beta_2+\tau_2)}$ or

$$\frac{c_2}{\alpha_2} \leq \frac{(\alpha_1+\gamma_1)c_1}{\alpha_1^2(\beta_2+\tau_2)}. \tag{2.21}$$

only threshold 2.19 is active, and we call this threshold as **subcontract incentive threshold**,

$$(1-\alpha_1-\gamma_1)(\alpha_2+\gamma_2)\frac{c_2}{\alpha_2} + (\alpha_1+\gamma_1)(\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2+\tau_2} \geq \frac{(\alpha_1+\gamma_1)(\alpha_2+\gamma_2)}{\alpha_1(\beta_2+\tau_2)}\frac{c_1}{\alpha_1}.$$

In these case, because direct control could sign more efficient contract with agent 2, these parameters don't have consistent impact as previous anymore.

For example, considering cross term $\tau_2$ in the delegation case, a large $\tau_2$ provides more incentive to middle agent 2 to sign subcontract with agent 1, which benefits delegation. However, when the contract payment $M_2^+$ for agent 2's effort is high enough, it already provides enough incentive for subcontracts. Meanwhile, threshold 2.19 becomes inactive, and threshold 2.18 becomes active. As a result, a higher $\tau_2$ decrease the left-hand side but increase the right-hand side, which makes the principal prefers direct control. The intuition is the principal cares more about efficient contract when $M_2 = M_2^+$, and a large $\tau_2$ can help save a lot cost when contract $M_2$ can use additional signal $s_1$.

In summary, for delegation, the advantage of higher $\tau_2$ is a more significant incentive to middle agent 2 to sign subcontract, while the disadvantage is the contract with agent 2 is less efficient by losing the information from $s_1$. On the principal's side, the principal may consider more about how to leverage the aligned the interest and use a middle agent for delegation when the cost of a middle agent is small. However, when the cost of a middle agent is enormous, how to obtain more information and design an efficient contract becomes more important to the principal.

## 2.3 Path

### 2.3.1 Path Model

We consider a principal-agent model with the working process on a directed path, as in Figure 2.5. The work is initiated at the agent 1, after agent 1 exerts an unobservable effort, the task is passed over to agent 2 with an observable output signal $s_1$. After receiving this signal, agent 2 starts to decide the effort and so on so forth until the final task is completed and handed over to the principal.

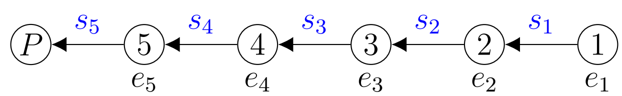



Fig. 2.5.: Production Chain

In this sequential production process, we assume the effort is costly and **unobservable**, and takes two possible values that we normalize respectively as a zero effort level and a positive effort of one: $e \in \{0, 1\}$ [2]. Meanwhile, the output signal after task is completed in each stage is **observable** and binary, which indicate the task completeness, success or failure.

$$s_k = \begin{cases} 1, \text{success}, \\ 0, \text{failure}. \end{cases}$$

Denote the set of agents as $\mathcal{N}$. For the initial agent 1, the output signal $s_1$ is a random variable with probability function

$$P(s_1|e_1) = \alpha_1 e_1 + \gamma_1,$$

where $\alpha_1$ is the positive impact form the effort and $\gamma_1$ is the environmental influence. For intermediate agent $k > 1$, the success of task is also depends on the task status from the previous signal $s_{k-1}$, and the probability function is as follows:

$$P(s_k = 1|e_k, s_{k-1}) = \alpha_k e_k + \beta_k s_{k-1} + \tau_k s_{k-1} e_k + \gamma_k, \tag{2.22}$$

where $\beta_k$ is the positive impact form the previous success, $\tau_2$ is the cross term, and can be interpreted as a previous success will make current effort more valuable to the project. We assume

[2]Our results can be extent to the case with multiple effort levels

$$\alpha_k, \beta_k, \tau_k, \gamma_k \geq 0, \forall k \in \mathcal{N},$$
$$\alpha_k + \beta_k + \tau_k + \gamma_k \leq 1, \forall k \in \mathcal{N},$$

and every coefficient $\alpha_k, \beta_k, \tau_k, \gamma_k, \forall k \in \mathcal{N}$ are common information to everyone in the production chain.

The primary goal of the principal is to maximize the success probability of the final task, $P(s_n = 1)$. In other words, the principal wants every agent to make an effort. Recall that effort are costly and unobservable, thus contracts with payment based on task status can be used to induce agents' effort. Our model assumes the agents has **zero liability**. In other words, the agent cannot be punished in the contract if the task is failed.

About the delegation structure, we have the following assumption,

**Assumption 2.3.1 (Continuous Delegation)** *Every subtree in the delegation tree is an interval in the production chain.*

The motivation is, in practice, if an agent's downstream and upstream supplier are both under the control of another agent $k$, then most likely, this agent also under the influence of agent $k$. To illustrate this assumption better, following are two examples of invalid and valid delegation structure of production chain in Figure 2.5:

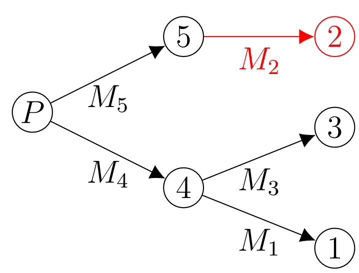



Fig. 2.6.: Example of Invalid Delegation Structure

The above structure is invalid, because agent 2 violate the continuous delegation assumption and should also be delegated to agent 4 instead of 5.

In the Figure 2.7, the principal direct controls (signs contract with) agent $4, 5$ and delegates agents $1, 2, 3$ to agent 4. After signed the contract with principal, agent 4 has the freedom to decide whether signs a subcontract with his children, based on his contract payment $M_4$ from the principal.

In summary, the model contains three stages:

- Design Stage: The production chain and parameters $(\alpha, \beta, \gamma, c)$ are given as common information[3], and the principal designs a delegation tree accordingly.

[3]every agent only needs to know its descendants' information.

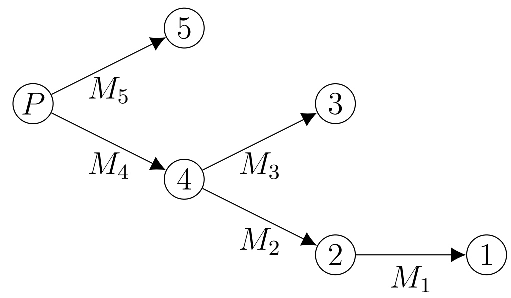



Fig. 2.7.: Example of Valid Delegation Structure

- Contract Stage: **In the delegation tree**, the principal initiates the contract signing, and passing down until the leaves, a node
  - receives contract from his parent;
  - signs contract with his children.
- Effort Stage: **In the production chain**, the leaf starts working on the task, and passing up until the root, a node
  - receives the work and signal from his predecessor;
  - decides to make effort or not (unobservable);
  - passes over his task to the successor.

The goal of the principal is to **maximize the success probability with the lowest cost**. Namely, the principal wants to minimize the cost under the condition that every single agent has the incentive to work.

The utility function of every agent is assumed to be **risk neutral** and the expected utility function includes three parts,

- expected contract reward from the successor or the principal;
- expected contract cost to the children in delegation tree;
- cost of personal effort.

For agent $k \in \mathcal{N}$, the output signal of every descendant under his control may influence his utility. Hence, all of those signals may be used to design the contract with $k$. For simplicity of the contract, we have the following assumption for the contracts,

**Assumption 2.3.2 (Signal Condition)** *The contract payment only depends on the output signal of whom received the contract.*

In the other words, if someone wants to sign a contract with agent $k$, then the contract payment can only depend on signal $s_k$. For example, agent $k$'s contract payment $r_k$ is

$$r_k = \begin{cases} M_k, & \text{if } s_k = 1, \\ 0, & \text{if } s_k = 0. \end{cases} \tag{2.23}$$

Note that for an efficient contract, the payment when project fails is always 0, and we call $M_k$ as the contract payment to agent $k$.

Because of the conflicting interests between the principal and the agents, the question is what's the optimal delegation structure and contract payment the principal should choose? For the delegation structure, does direct control every agent the best choice, or only do the delegation following the production chain structure? In the next section, we will show the answers to the above questions is not fixed but depends on the model parameters. Moreover, we will provide a polynomial time algorithm.

### 2.3.2 Preliminary

Recall the probability of success for agent $k$ (2.22) is

$$P(s_k = 1|e_k, s_{k-1}) = \alpha_k e_k + \beta_k s_{k-1} + \tau_k s_{k-1} e_k + \gamma_k.$$

Because we assume success of the previous signal always has a positive effects over the current task ($\tau_k \geq 0$), agent $k$ has more incentive to exert effort when observed $s_{k-1} = 1$. Therefore, we define three effort status for agent $k$,

- Zero effort, never exert effort;
- Conditional effort, exert effort only when previous signal is positive;
- Full effort, always exert effort regardless the previous signal.

Given the contract payment $M_k$, the corresponding effort status for agent $k$ can be determined by the following theorem.

**Theorem 2.3.1** *For an agent $k$,*

- *the minimum payment for **conditional personal effort** is $M_k(\tilde{k}) = \frac{c_k}{\alpha_k + \tau_k}$.*
- *the minimum payment for **full personal effort** is $M_k(k) = \frac{c_k}{\alpha_k}$.*

If $k$ choose to shirk, his expected payoff is

$$\pi_k(0|s_{k-1}, M_k) = (\beta_k s_{k-1} + \gamma_k)M_k - \sum_{h \in T(k)} s_h M_h.$$

If $k$ makes effort, his expected payoff is

$$\pi_k(1|s_{k-1}, M_k) = (\alpha_k + \beta_k s_{k-1} + \tau_k s_{k-1} + \gamma_k)M_k - \sum_{h\in T(k)} s_h M_h - c_k.$$

Thus, the utility function given $s_{k-1}, M_k$ is a piecewise linear function as in Fig. 2.8,

$$\pi_k(M_k|s_{k-1}) = \max_{e_k\in\{0,1\}} \pi_k(e_k|s_{k-1}, M_k).$$

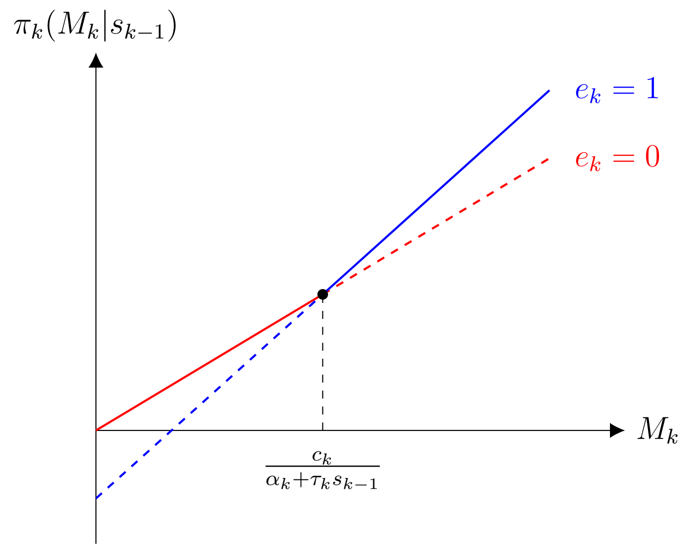



Fig. 2.8.: Piecewise Linear Utility Function of Agent $k$

We know the minimum incentive payment must satisfy $\pi_k(1|s_{k-1}, M_k) \geq \pi_k(0|s_{k-1}, M_k)$, which gives

$$(\alpha_k + \tau_k s_{k-1})M_k \geq c_k. \tag{2.24}$$

By the different values of $s_{k-1}$, we prove the Theorem 2.3.1 by the above inequality 2.24. Note that the effort decision at effort stage is irrelevant to the subcontract decisions in the contract stage.

By Theorem 2.3.1, we know the relation between personal effort $e_k$ and contract payment $M_k$, and we can use it to solve a special case that the principal signs direct contracts with every agent in the production chain. For example, the delegation tree is Fig. 2.9 for a production chain as in Fig 2.5, where the solid black line represents the contract direction, and the blue dashed line represents the original process direction.

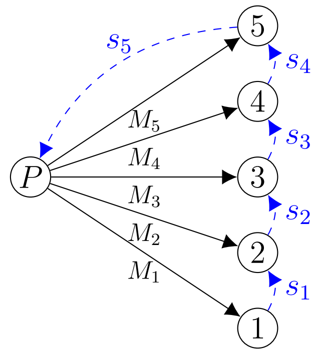



Fig. 2.9.: Direct Control

For the contract payment to $k$, if it's less than $M_k(k)$, it's not enough to provide incentive for full effort. If it's larger than $M_k(k)$, it only increases the expected cost of the principal. Hence, the expected cost in direct control case is

$$cost_p = \sum_{k=1}^{n} P(k)M_k(k).$$

Other than directly controlling every agent, suppose the principal decides to delegate a set of agents to agent $k$, while these agents are the descendants of $k$ in the delegation tree, we call them as the **control set** of $k$[4].

By the Assumption 2.3.1, the control set of $k$ must be a **continuous set** of agents from $i$ to $k$, where $1 \leq i \leq k$. Thus, we use notation $\theta_k^i$ to denote the control set of $k$ including agents from $i$ to $k$. Denote $\Theta(k)$ as the set of all possible control set of $k$. For each control set, there are different delegation structures, denoted as $\eta(\theta_k^i)$.

In the case of delegation, the principal will not sign a direct contract with the control set of $k$, and $k$'s subcontract decision will determine their effort status. The next question is given $M_k$ what would be the effort status of $k$'s descendants in the delegation tree?

We first introduce some notations to describe the effort status of $k$'s control set $\theta_k^i$.

- $\psi_{i,k}^0$, set of agents who make zero effort;
- $\psi_{i,k}^1$, set of agents who make conditional effort (when the previous signal is positive);
- $\psi_{i,k}^2$, set of agents who makes full effort (regardless the previous signal).

[4]including $k$

and we defined the motivated level of every agent in the control set $\theta_k^i$ as the effort status:

$$\psi_{i,k} = \{\psi_{i,k}^0, \psi_{i,k}^1, \psi_{i,k}^2\}.$$

The set of all the possible effort status $\psi_{i,k}$ under $\theta_k^i$ is denoted as $\Psi_{i,k}$.

By offering different contract payments to his children in the delegation tree, an agent can manipulate the effort level $\psi_{i,k}$ in his control set $\theta_k^i$. Meanwhile, for the agents out of his control set, he always believes the others will exert effort,

**Proposition 2.3.1** *Agents believe the agents not in his control set will make an effort.*

**Proof** Since it is common information that the principal's priority is motivating everyone to work. Every agent will believe who are not under his delegation are motivated by the principal's arrangement. ∎

When all of the previous agents are willing to commit effort, the probability of success of agent $k$, denoted as $P(k)$, can be computed recursively as follows,

$$P(k) = \alpha_k + (\beta_k + \tau_k)P(k-1) + \gamma_k \tag{2.25}$$

$$= \sum_{h=1}^{k}(\alpha_h + \gamma_h) \prod_{j=h+1}^{k} (\beta_j + \tau_j), \tag{2.26}$$

and $P(0) = 0$.

Given an effort status $\psi_k$, with the above belief, probability of success $P(s_k = 1|\psi_{i,k})$ (also denoted as $p_k(\psi_{i,k})$) can be decomposed and computed in four parts,

- Impacts from agents before $i$, by Proposition 2.3.1, all of those agents are believed to be making effort,

$$p_k^0 = \beta_{i,k}P(i-1), \tag{2.27}$$

  where $\beta_{i,k} = \prod_{h=i}^{k} \beta_j$;

- Impacts from zero effort agents among $\theta_k^i$,

$$p_k^1 = \sum_{h \in \psi_{i,k}^0} \beta_{h+1,k}\gamma_h; \tag{2.28}$$

- Impacts from conditional effort agents among $\theta_k^i$,

$$p_k^2 = \sum_{h \in \psi_{i,k}^1} \beta_{h+1,k}\big((\alpha_h + \tau_h)p_{h-1}(\psi_k|\theta_k^i) + \gamma_h\big); \tag{2.29}$$

- Impacts from full effort agents among $\theta_k^i$,

$$p_k^3 = \sum_{h \in \psi_{i,k}^2} \beta_{h+1,k}\big(\alpha_h + \tau_h p_{h-1}(\psi_k|\theta_k^i) + \gamma_h\big). \tag{2.30}$$

In summary, the expected successful probability of $k$ given effort status $\psi_k$ over contract set $\theta_k^i$ has the following close form expression,

$$p_k(\psi_{i,k}) = p_k^0 + p_k^1 + p_k^2 + p_k^3. \tag{2.31}$$

For the impact from interval $\theta_k^i$, we denote it as $\Delta p_k(\psi_k|\eta_k)$,

$$\Delta p_k(\psi_{i,k}) = p_k^1 + p_k^2 + p_k^3. \tag{2.32}$$

To measure the size of the possible delegation structure, we introduce the following definition,

**Definition 2.3.1 (Delegation Depth)** *For agent $k$, the distance to $k$'s the furthest delegated agent is the delegation depth of agent $k$.*

By assumption 2.3.1, since delegated agents are continuous on the process path, the delegation depth of an agent is equal to the size of the control set of that agent minus one. For example, the delegation depth of agent $k$ in control set $\theta_k^{k-d}$ is $d$ as in Fig 2.10.

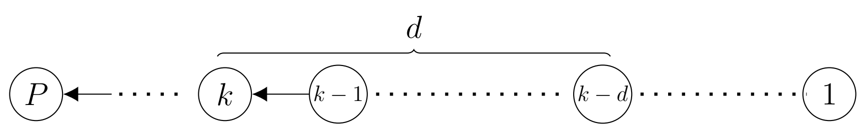



Fig. 2.10.: Delegation with Depth $d$ in a Process Path

Suppose delegate agent $k_0 < k$ under $k$'s control, we can derive lower bound of $M_k$.

**Theorem 2.3.2** *If delegate agent $k_0 < k$ under $k$'s control, then*

$$M_k \geq \frac{M_{k_0}(k_0)}{\prod_{i=k_0+1}^{k}(\beta_i + \tau_i)}. \tag{2.33}$$

Recall that $M_{k_0}(k_0)$ is the minimum payment for $k_0$'s personal effort. The proof is provided in Appendix 2.10.1.

The above Theorem suggests that the contract payment grows exponentially with the depth of delegation. To illustrate that, consider all the $\beta_k$ and $\tau_k, k \in \mathbb{N}$ is equal, we have the following Corollary as a special case of Inequality 2.33, Corollary 2.3.1 *Suppose $\beta = \beta_k, \tau = \tau_k, k \in \mathbb{N}$ is a constant, the lower bound of contract payment to motivate a agent $k$ with delegation depth $d$ is*

$$M_k \geq \frac{M_{k-d}(k-d)}{(\beta+\tau)^d}.$$

Because of the exponentially increasing cost, the principal generally doesn't consider a delegation structure with too large depth. Therefore, in order to analysis the optimal delegation structure in the process path, here we introduce an bounded depth assumption

**Assumption 2.3.3** *The delegation depth for any agent is bounded by $d$.*

With the above assumption, we can prove the delegation structure is bounded.

Lemma 2.3.1 *If an agent has delegated a control set with size $d$, the possible delegation substructure below this agent is $O(2^{d^2})$*

The proof is provided in Appendix 2.10.2. Based on this, we are going to propose a polynomial time algorithm to find the optimal delegation structure in the next section.

### 2.3.3 Dynamic Programming Algorithm

For agent $k$, given delegation structure $\eta_{i,k}$ with control set $\theta_k^i$ and contract $M_k$. The main question in this section is what's the effort status $\psi_{i,k}$? We build the connection between $M_k$ and $\psi_{i,k}$ by induction, suppose $k$ is a leaf agent, by Theorem 2.3.1, the corresponding effort status of $k$ is

- Zero effort, if $M_k < M_k(\tilde{k})$;
- Conditional effort, if $M_k(\tilde{k}) \leq M_k < M_k(k)$;
- Full effort, if $M_k \geq M_k(k)$.

Now consider middle agent $k$, for every child $h \in T(k)$, by induction there is a mapping from minimum contract payment $M_h$ to $\psi_h$, and his expected utility function at the contract stage is

$$\begin{aligned}\pi_k(M_{\vec{h}}|M_k, \eta_k) &= p_k(e_k(M_k), M_{\vec{h}}|\eta_k)M_k - \sum_{h_j \in T(k)} p_h(M_{h_j}|\eta_k)M_h - c_k e_k \\ &= p_k(\psi_{i,k})M_k - \sum_{h_j \in T(k)} p_h(\psi_{h_{j-1}+1,h_j})M_h - c_k e_k,\end{aligned}$$

By Equation 2.22, the expected probability $p_k(\psi_{i,k})$ can be written as

$$\begin{aligned} p_k(\psi_{i,k}) &= \alpha_k e_k(M_k) + \beta_k p_{k-1}(\psi_{i,k-1}) + \tau_k p_{k-1}(\psi_{i,k-1}) e_k(M_k) + \gamma_k \\ &= \alpha_k e_k(M_k) + \sum_{h\in T(k)} \beta_{h+1,k}\Delta p_h(\psi_{i,k}) + \beta_{i,k}P(i-1) + \tau_k p_{k-1}(\psi_{i,k-1}) e_k(M_k) + \gamma_k, \end{aligned} \tag{2.34}$$

recall that $\beta_{h+1,k} = \prod_{j=h+1}^{k} \beta_j$, and the second equality is from Equation 2.31.

For $k$'s personal effort $e_k(M_k)$, it's not a decision variable at the contract stage, but we have its expected value by Theorem 2.3.1,

$$e_k(M_k) = \begin{cases} 0, & M_k \leq \frac{c_k}{\alpha_k+\tau_k}, \\ p_{k-1}(\psi_{i,k-1}), & \frac{c_k}{\alpha_k+\tau_k} \leq M_k \leq \frac{c_k}{\alpha_k}, \\ 1, & M_k \geq \frac{c_k}{\alpha_k}. \end{cases} \tag{2.35}$$

Therefore, we can rewrite the total utility function of agent $k$ at contract stage as

$$\pi_k(M_{\vec{h}}|M_k,\eta_k) = \pi_k^p(\psi_{i,k}|M_k) + \sum_{h\in T(k)} \pi_{k,h}^s(\psi_{i,k}|M_k) + (\beta_{i,k}P(i-1)+\gamma_k)M_k, \tag{2.36}$$

where

$$\pi_k^p(\psi_{i,k}|M_k) = (\alpha_k e_k(M_k) + \tau_k p_{k-1}(\psi_{i,k-1}) e_k(M_k))M_k - c_k e_k \tag{2.37}$$

$$= \begin{cases} 0, & M_k \leq \frac{c_k}{\alpha_k+\tau_k}, \\ \big((\alpha_k+\tau_k)M_k - c_k\big)p_{k-1}(\psi_{i,k-1}), & \frac{c_k}{\alpha_k+\tau_k} \leq M_k \leq \frac{c_k}{\alpha_k}, \\ \alpha_k M_k - c_k + \tau_k p_{k-1}(\psi_{i,k-1})M_k, & M_k \geq \frac{c_k}{\alpha_k}. \end{cases} \tag{2.38}$$

$$\pi_{k,h}^s(\psi_{i,k}|M_k) = \beta_{h+1,k}\Delta p_h(\psi_{i,k})M_k - p_h(M_h|\eta_k)M_h. \tag{2.39}$$

By picking the optimal $\psi_{i,k}$ that maximize the above utility of $k$, we find a mapping between $M_k$ and $\psi_{i,k}$. The following lemma builds the one-to-one mapping and shows the optimal $\psi_{i,k}$ can be found by enumerating polynomial possibilities.

LEMMA 2.3.2 *For any agent $k$, given delegation depth $d$, there is an one-to-one mapping between minimum contract payments and effort status. And the choice of effort status is bounded by $3^d$, i.e.,*

$$|\Psi_{k-d,k}| \leq 3^d.$$

The proof is provided in Appendix 2.10.3. Besides the one-to-one mapping, the effort status of $k$ is monotone by inclusion, and the proof is provided in Appendix 2.10.4.

**Theorem 2.3.3** *The effort status satisfies monotone inclusion with the increasing of the contract payment.*

Denote **the minimum contract payment for effort status** as $M_k(\psi_{i,k})$.

By the computation in the proof, we define the optimal sub-delegation structure as

$$\eta_k^{i*} = \text{argmin}_{\eta_k^i \in H(\theta_k^i)} M_k(\theta_k^i|\eta_k^i),$$

and the optimal full incentive contract payment

$$M_k(\theta_k^i) = \min_{\eta_k^i \in H(\theta_k^i)} M_k(\theta_k^i|\eta_k^i) = M_k(\theta_k^i|\eta_k^*).$$

Define the **minimum expected cost till** $k$ is the minimum expect cost for the principal to motivate agents from 1 to $k$, denoted as $cost_k$. Now we can provide the algorithm to update the minimum cost at each stage.

It starts with $cost_1 = M_1(1)$. Expected cost till $k$, for any $0 \leq k - d \leq i \leq k$,

$$cost_{k,i} = M_k(\theta_k^i) + cost_{i-1},$$

and the minimum cost is

$$cost_k = \min_{k-d \leq i \leq k} cost_{k,i}.$$

The optimal delegation till $k$ is

$$\eta_k^* = \eta_k^{i*} \cup \eta_{i-1}^*$$

In summary, DP along the working process path, while DP stores

- Control set of $k$, $\Theta(k)$.
- For each control set $\theta_k^i \in \Theta(k)$, set of all delegation structures $H(\theta_k^i)$.
- For each delegation structure $\eta_k \in H(\theta_k^i)$, set of all possible effort status $\Psi_k(\eta_k)$.
- For each effort status $\psi_k \in \Psi_k(\eta_k)$, the corresponding minimum contract payment $M_k(\psi_k)$.
- Minimum expected cost till $k$, $cost_k$, and the corresponding optimal structure $\eta_k^*$.

**Theorem 2.3.4** *Time complexity is $O(nd2^{d^2}3^d)$.*

**Proof** As in Algorithm 5. There are $n$ stages ($n$ agents), and at stage $k$, there are at most $d$ control sets. For each control set $\theta_k^i$, there are $O(2^{d^2})$ delegation structure.

For each delegation structure $\eta_k$, we plot the piecewise utility function from the previous DP status and find the mapping between effort status and contract payment. The number of pieces is bounded by $O(3^d)$.

Overall, the time complexity is $O(nd2^{d^2}3^d)$. ■

**Algorithm 4** : Optimal Delegation Structure with Bounded Depth

**for** $k = 1$ to $n$ **do** $\triangleright$ agent $k$
  $cost_k \leftarrow 0$.
  **for** $i = k - d$ to $k$ **do** $\triangleright$ control set $\theta_k^i$
    $\Psi_i(\theta_i^i) = \{\emptyset, \{i\}\}$.
    $M_i(\emptyset) = 0$.
    $M_i(\{i\}|\theta_i^i) = \frac{c_i}{\alpha_i}$.
    **for** $\eta_k \in H(\theta_k^i)$ **do** $\triangleright$ DP from $\theta_j^i$
      Plot $\pi_k(M_k|\theta_k^i, \eta_k)$ by Lemma 2.3.2.
      Intersection points gives $\Psi_k(\theta_k^i, \eta_k)$ and $M_k(\psi_k|\theta_k^i, \eta_k)$.
    $cost_k(\theta_k^i) = P(k)M_k(\theta_k^i) + cost_{i-1}$
    **if** $i = 1$ **or** $cost_k > cost_k(\theta_k^i)$ **then**
      $cost_k \leftarrow cost_k(\theta_k^i)$.
      $i^* \leftarrow i$
  $\eta_k = \eta_{i^*} \cup \{k\}$
**return** Optimal Structure $\eta_n$ and minimum expected cost $cost_n$

Space complexity is $O(d^2 2^{d^2} 3^d)$, since we only need the last $d$ agents' status information to update in the DP algorithm.

### 2.3.4 Properties

**Theorem 2.3.5** *Expected cost till $k$ is monotone increasing.*

**Proof** Consider the expected cost till $k$, suppose the cost is minimized when the principal delegate $i, \dots, k-1$ to $k$, and the minimum expected cost is

$$cost_k = p_k M_k(\theta_k^i) + cost_{i-1},$$

while the utility of agent $k$ is

$$\pi_k = p_k M_k(\theta_k^i) - p_{k-1} M_{k-1}(\theta_{k-1}^i) - c_k \geq 0.$$

Thus, $p_k M_k(\theta_k^i) \geq p_{k-1} M_{k-1}(\theta_{k-1}^i)$, which infers

$$\begin{aligned} cost_{k-1} &\leq p_{k-1} M_{k-1}(\theta_{k-1}^i) + cost_{i-1} \\ &\leq cost_k. \end{aligned}$$

$\blacksquare$

**Proposition 2.3.2** *In the symmetric case with two agents, delegation is better than direct control when*

$$\beta(1+\beta) \geq 1.$$

**Proof** In the symmetric case, where $\alpha_1 = \alpha_2$, $c_1 = c_2$. For the computation simplicity, assume $\gamma = 0$.

$$\frac{p_2 - \alpha_1\beta_2}{\alpha_1\beta_2}\frac{p_1}{p_2}\frac{c_1}{\alpha_1} = \frac{1}{\beta}\frac{1}{1+\beta}\frac{c}{\alpha}.$$

Therefore, delegation will be better if and only if

$$\beta(1+\beta) \geq 1.$$

■

LEMMA 2.3.3 *In the symmetric case, the probability of success when every downstream agent is working converges to* $\frac{\alpha+\gamma}{1-\beta}$.

**Proof** The probability of success of agent $k$ is

$$p_k = f(p_{k-1}) = \alpha + \gamma + \beta p_{k-1}.$$

Since $0 < \beta < 1$, $f(\cdot)$ is a contractive mapping. By Banach Fixed Point Theorem, we know the fix point exists and unique. Therefore, by solving

$$p^* = \alpha + \gamma + \beta p^*.$$

we have $p^* = \frac{\alpha+\gamma}{1-\beta}$. ■

**Proposition 2.3.3** *If every agent and task is identical, when an agent is far from the initial agent on the process path, the principal prefers direct control to delegation.*

The proof is provided in Appendix 2.10.6.

**Proposition 2.3.4** *In the symmetric case, suppose one agent has an effort cost $t$ times larger than the other agents' cost, the delegation depth of this agent is bounded by*

$$d \leq \frac{\log t}{\log(\beta+\tau)^{-1}}.$$

The proof is provided in Appendix 2.10.7.

### 2.3.5 Example

Consider the process chain as in Fig. 2.5 with 6 nodes,

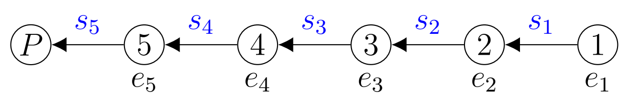



Fig. 2.11.: Process path

and we'll show an example such that the optimal delegation is as follows,

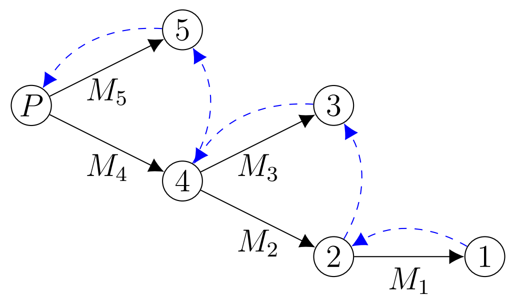



Fig. 2.12.: Optimal Delegation Tree

Let $\tau_i = 0$ for all the agent $i$. We also have $M_i(i) = \frac{c_i}{\alpha_i}$.
Let $c_1 = 1$, $\alpha_1 = 0.4$. Then,

$$M_1(1) = \frac{5}{2}, P(1) = 0.4.$$

Let $c_2 = 2$, $\alpha_2 = 0.2$, $\beta_2 = 0.5$. Therefore,

$$M_2(2) = 10, P(2) = \alpha_2 + P(1)\beta_2 = 0.4.$$

Let $c_3 = 1$, $\alpha_3 = 0.1$, $\beta_3 = 0.25$, $\gamma_3 = 0.4$, then

$$M_3(3) = 10, P(3) = \alpha_3 + P(2)\beta_3 + \gamma_3 = 0.6$$

Let $c_4 = 12$, $\alpha_4 = 0.1$, $\beta_4 = 0.5$,

$$M_4(4) = 120, P(4) = \alpha_4 + P(3)\beta_4 = 0.4.$$

Let $c_5 = 1$, $\alpha_5 = 0.5$, $\beta_4 = 0.5$,

$$M_4(4) = 2, P(5) = \alpha_5 + P(4)\beta_5 = 0.7.$$

For the sub-delegation tree below agent 4, we have the following three options.
**Option 1:**

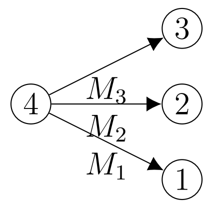



Fig. 2.13.: Option 1

By comparing $\pi_4(1234) \geq \pi_4(234)$, we have a lower bound for $M_4$,

$$\begin{aligned} M_4 &\geq \frac{P(1)M_1(1)}{\alpha_1\beta_2\beta_3\beta_4} + \frac{M_2(2)}{\beta_3\beta_4} + \frac{M_3(3)}{\beta_4} \\ &= 40 + 80 + 20 \\ &= 140. \end{aligned}$$

**Option 2:**

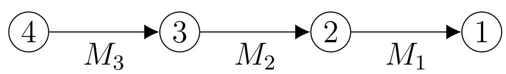



Fig. 2.14.: Option 2

Minimum payment for 2,

$$M_2 \geq \frac{P(1)M_1(1)}{\alpha_1\beta_2} = 5.$$

which is smaller than $M_2(2)$. For the minimum payment for 3,

$$M_3 = \frac{P(2)M_2(2)}{P(2)\beta_3} = 40.$$

By comparing $\pi_4(34) \geq \pi_4(4)$, we have the lower bound for $M_4$ in option 2,

$$M_4 \geq \frac{P(3)M_3}{(\alpha_3 + P(2)\beta_3)\beta_4} = \frac{0.6 * 40}{0.2 * 0.5} = 240.$$

**Option 3:**

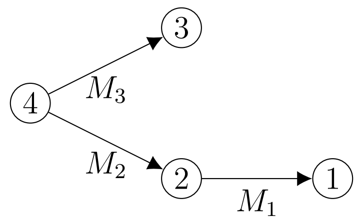



Fig. 2.15.: Option 3

By comparing $\pi_4(234) \geq \pi_4(24)$,

$$M_4 \geq \frac{P(3)M_3(3)}{\alpha_3\beta_4} = \frac{0.6 * 10}{0.1 * 0.5} = 120.$$

By comparing $\pi_4(234) \geq \pi_4(34)$,

$$\begin{aligned} M_4 &\geq \frac{M_2(2)}{\beta_3\beta_4} + \frac{M_3(3)}{\beta_4} \\ &= \frac{10}{0.25 * 0.5} + \frac{10}{0.5} \\ &= 80 + 20 \\ &= 100. \end{aligned}$$

Therefore, the minimum payment for personal incentive $M_4(4)$ is enough to make every agents in option 3 be incentive.

Therefore, the principal will always choose option 3. Now consider the relation between 4 and 5, if delegate 4 under the control of 5, the lower bound of payment to agent 5 is

$$M_5 \geq \frac{P(4)M_4(4)}{P(4)\beta_5} = 240$$

However, the cost of direct control 4 and 5 is only

$$cost_p = M_4(4) + M_5(5) = 122$$

In summary, the optimal delegation structure is the tree as in Fig. 2.12.

## 2.4 Tree

### 2.4.1 Tree Model Description

We consider the principal-agent model with a working process on a directed rooted tree. Denote the set of agents as $\mathcal{N}$. The work is initiated at the leaf agents, and passed over to the parents and so forth, and finally ends at the principal. An example of the process tree is given in Figure 2.16

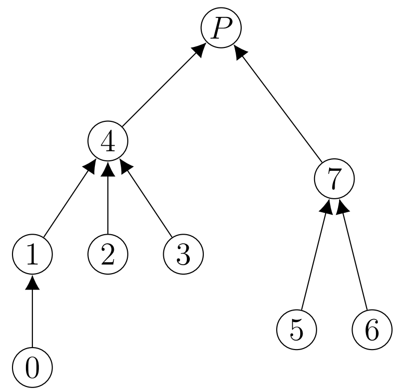



Fig. 2.16.: Process tree

From leaves to the root, after received the completed tasks from his children, each agent $k$ can decide to put effort or not (binary variable) to his task and this effort is **unobservable** to the others. However, after the task of agent $k$ is done, a binary signal $s_k \in \{0, 1\}$ indicating the task status will be observed by **his parent or the principal if he signed contract with** $k$.

$$s_k = \begin{cases} 1, \text{success}, \\ 0, \text{failure}. \end{cases}$$

For agent $k \in \mathbb{M}$, the success probability of his task is related to three elements,

- Personal effort $e_k$;
- Task status from his children $S_k = \{s_i, i \in C(k)\}$;
- Environmental influence $r_k$.

where $C(k)$ is the direct children of $k$ in the process tree. We consider a linear success probability function of task $k$, defined as follows:

$$P(s_k = 1|e_k, S_k) = \alpha_k e_k + \sum_{i \in C(k)} \beta_k^i s_i + \gamma_k, \text{where } e_k, s_i \in \{0, 1\}. \tag{2.40}$$

For any agent $\forall k \in \mathbb{M}$, coefficients $\alpha_k, \beta_k^i, i \in C(k)$ are assumed to be strictly positive. This is a fair assumption. If it doesn't hold, for example, $\beta_k^i = 0$, node $i$'s task won't influence the final project (at principal level), and $i$ and his descendants can be removed from the process tree. To ensure the successful probability ($P(s_k = 1|e_k, S_k)$) always between 0 and 1, we assume $\alpha_k + \sum_{k=1}^n \beta_k^i + \gamma_k \leq 1, \gamma_k \geq 0, \forall k \in \mathbb{M}$.

Personal effort will increase the probability that $s_k = 1$ (Equation 2.40), but it'll also result in an additional effort cost $c_k$ to agent $k$. Therefore, the agents generally choose to shirk unless there is additional payment after the task is successful. To motivate agents to work, the principal can sign contracts with downstream agents. Since we assume the agents can't be punished, an instance of agent $k$'s contract payment $r_k$ can be

$$r_k = \begin{cases} M_k, & \text{if } s_k = 1, \\ M'_k, & \text{if } s_k = 0. \end{cases} \tag{2.41}$$

It's easy to see that an efficient contract always sets $M'_k = 0$, and we call $M_k$ as the contract payment to agent $k$. Meanwhile, it can be proved that for the principal, signal $s_k$ is the only useful information for the contract with agent $k$.

Moreover, not only the principal can sign a contract directly with agents, and every agent can sign a contract with the other agents but restricted to his children set.

**Assumption 2.4.1** *Agents can only sign contracts with their children in the process tree.*

Agents may have the incentive to do that. For example, when the contract payment $M_k$ to agent $k$ is large, $k$ may be better off by sign subcontracts to motivate his children to put effort, and eventually increases his success probability (equation 2.40) and expected payoff.

Besides that, the principal has the power to design the delegation structure (tree), and the agents are only allowed to sign subcontracts with his direct children in the delegation tree. Following is a possible delegation tree of the process tree in Figure 2.16, where the solid black line means contract direction, and the blue and orange dashed line means the process direction:

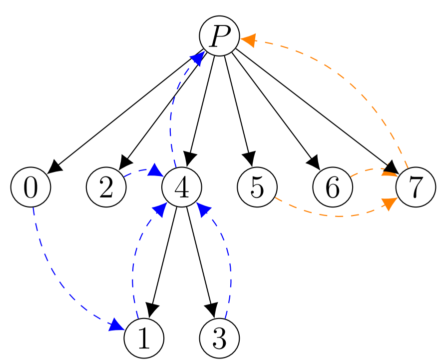



Fig. 2.17.: Delegation tree

An the Figure 2.17, the principal direct controls (signs contract with) agent $0, 2, 4, 5, 6, 7$ and delegates agents $1, 3$ to agent 4. Therefore, agent 4 has the freedom to decide whether signs a subcontract with his children (in the delegation tree), based on his contract payment $M_4$ signed with the principal.

In summary, we break down the model in three stages:

- Stage 1: The process tree and parameters $(\alpha, \beta, \gamma, c)$ are given as common information[5], and the principal designs a delegation tree accordingly.
- Stage 2: **In the delegation tree**, the principal initiates the contract signing, and passing down until the leaves, an intermediate node
  - receives contract from his parent;
  - signs contract with his children.
- Stage 3: **In the process tree**, the leaves start working first, and passing the task up until the root, an intermediate node
  - receives the work and signals from his direct children;
  - decides to spend effort or not (unobservable);
  - passes over his task to the parent.

The goal of the principal is to maximize the success probability with the lowest cost. Namely, the principal wants to minimize the cost under the condition that every single agent in the process tree has the incentive to work. While given the delegation structure and contract payment ($M_k$) from the parent in delegation tree, the agent $k$ can decide

- contract payment to his children $M_i, i \in T(k)$;
- personal effort $e_k$.

where $T(k) \subseteq C(k)$ is set of $k$'s children in the delegation tree.

Assume every agent is **risk neutral**, an agent's utility function includes three parts, contract reward from the parent, contracts payment to the children, and personal effort cost. When every agent makes subcontract decisions, they have the following beliefs, similar to Proposition 2.3.1,

**Proposition 2.4.1** *Every agents $k$ believes the other agents who are not under his delegation are putting effort.*

The question is what's the optimal delegation structure and contract payment the principal should choose? For the delegation structure, does direct control every agent the best choice, or only do the delegation following the process tree structure? In the next section, we will show the answers to the above questions is not fixed but depends on the model parameters. Moreover, we will provide a polynomial time algorithm when the tree depth is bounded.

[5]every agent only needs to know its descendants' information.

### 2.4.2 Preliminary

The goal of the principal is to motivate every agent with the minimum the expected cost. This section will analyze the property of delegation structure that may help save the cost.

Given a delegation tree, the descendant of $k$ is call the **control set** of $k$, denoted as $\theta_k$[6]. Recall Assumption 2.4.1 that agents in the delegation tree are only able to receive contract from his parent, and sign contract with his children. Therefore, the **delegation structure** from $k$ has an one to one mapping to the **control set** of $k$. In the following sections, we'll also use the control set $\theta_k$ to represent the delegation structure of $k$.

Once given the delegation structure $\theta_k$ and contract payment $M_k$, recall that the agent $k$'s decision is

- contract payment to his children $M_i, i \in T(k)$;
- personal effort $e_k$.

while the contract payments to his children will eventually influence the working status of the agents in $k$'s control set. Denote the part of agents making effort as the **effort set** as $\psi(M_i, \theta_k)$, for simplicity, we sometimes use $\psi_k$ denote the effort set directly. Denote the set of all possible effort sets under $k$ as $\Psi(\theta_k)$.

Given the above information we can compute the utility function of $k$, the utility function is a **convex piecewise linear function**, denoted as $\pi_k(\theta_k, M_k)$.

$$\begin{aligned}\pi_k(\theta_k, M_k) &= \max_{\psi_k \in \Psi(\theta_k)} \pi_k(\psi_k, \theta_k, M_k) \\ &= \max_{\psi_k \in \Psi(\theta_k)} p_k(\psi_k|\theta_k)M_k - cost_k(\psi_k|\theta_k).\end{aligned}$$

Furthermore, we can define the **minimum payment over effort set $\psi_k$ under structure** $\theta_k$, denoted as $M_k(\psi_k|\theta_k)$, which is a contract payment to agent $k$ such that all agents in effort set ($\psi_k$) are exerting effort. Based on that, denote $M_k(\theta_k) = M_k(\psi_k = \theta_k|\theta_k)$ as the **minimum full incentive payment under structure** $\theta_k$, which is a minimum contract payment to agent $k$ such that every agent in his control set ($\theta_k$) is incentive. In the other words, $M_k(\theta_k)$ is the minimum $M_k^*$ satisfying the following equation

$$\pi_k(\theta_k, M_k^*) = \pi_k(\psi_k = \theta_k, \theta_k, M_k^*).$$

Delegation structure $\theta_k$ **dominates** $\theta_k'$ if

$$\begin{aligned}\theta_k' &\subset \theta_k, \\ M_k(\theta_k) &\leq M_k(\theta_k').\end{aligned}$$

[6]includes $k$.

Here we introduce the efficiency in the delegation structure,

**Definition 2.4.1 (EDS)** *A delegation structure is efficient if any other delegation structure does not dominate it.*

Similar to Theorem 2.3.1, agent's decisions between personal effort and contracts to each child-branch are independent.

**Theorem 2.4.1** *For any agent $k$, minimum payment for personal effort is $M_k \geq \frac{c_k}{\alpha_k}$.*

**Proof** Agent $k$ makes decisions about effort once the previous tasks are passed to him with signals. By Equation 2.40, given the previous signals, the difference of utilities between effort and shirking is,

$$\begin{aligned}&\pi_k(e_k = 1) - \pi_k(e_k = 0)\\ =&\alpha_k M_k - c_k.\end{aligned}$$

Therefore, the necessary and sufficient condition for agent $k$ to put effort is

$$M_k \geq \frac{c_k}{\alpha_k}.$$

■

Denote the **minimum personal effort** as $M_k^0 = \frac{c_k}{\alpha_k}$.

Because of the linearity of $P(s_k = 1|e_k, s_i, i \in C(k))$, we have the following theorem,

**Theorem 2.4.2** *For any agent $k$, its delegation decision over siblings in process tree is independent.*

**Proposition 2.4.2** *Each sibling in the delegation tree makes the decision independently.*

**Proof** By the beliefs from Proposition 2.3.1. ■

By the above results, we can handle each branch independently, then merge the decision and update the new contract payment, which sheds light to a polynomial time dynamic programming algorithm.

### 2.4.3 Two Layers

This section we consider a process tree with two layers as in Fig. 2.18. The main question is whether $P$ should direct control $k$'s contract workers?

By the Theorem 2.4.1, for each leaf node, the minimum effort payment is

$$M_i(i) = \frac{c_i}{\alpha_i}.$$

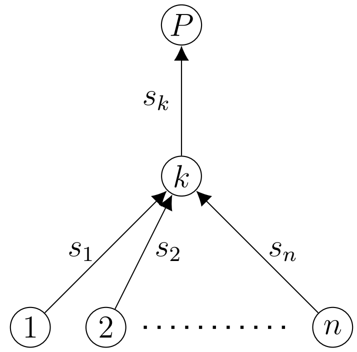



Fig. 2.18.: Two Layers Process tree

By Theorem 2.4.2, for agents $1 \leq i \leq n$, agent $k$ consider the contract with them independently. If $k$ doesn't sign it,

$$\pi_k^0 = p_k M_k - c.$$

If $k$ signs with $i$,

$$\pi_k(i) = (p_k + \alpha_i \beta_k^i) M_k - c - P(i) M_i(i).$$

In order to satisfy $\pi_k(i) \geq \pi_k$, we have the minimum payment for $k$ to motivate $i$,

$$M_k(i) \geq \frac{P(i) M_i(i)}{\alpha_i \beta_k^i}.$$

Therefore, we proved the following results

**Theorem 2.4.3** *The minimum contract payment for $k$ to motivate its children $i \in C(k)$ is*

$$M_k(i) = \frac{p_i M_i(i)}{\alpha_i \beta_k^i}.$$

Without loss of generality, we can rank the children of $k$ by an increasing order of $M_k(i)$, that is $M_k(i) \geq M_k(j), i \geq j$. For an efficient delegation structure, if $M_k \geq M_k(i)$, then we know any agent $j < i$ will be delegated to agent $k$, instead of direct controlled by the principal. We denote $\theta_k^k = \{k\}$, and $\theta_k^i = \{k, 1, 2, \ldots, i\}$ as the control sets of $k$. **The intuition is the principal may principal may delegate the easy to motivate agents to the middle agent $k$, while signs direct contracts with the agents who only exerts effort given high rewards.**

**Theorem 2.4.4** *There are $n+1$ efficient delegation structure for agent $k$, which are $\theta_k^k$ and $\theta_k^i, 1 \leq i \leq n$.*

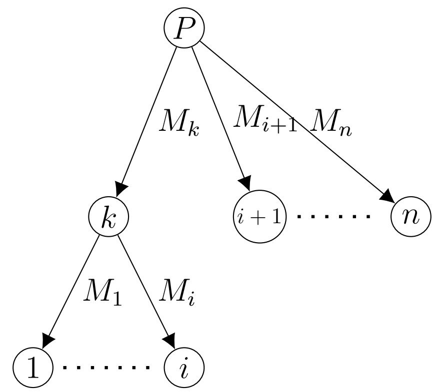



Fig. 2.19.: Example of Efficient Delegation Structure

Furthermore, it gives there is a linear number of efficient delegation structure for the principal, as in Fig. 2.19

To compute the minimum payment $M_k$ given an delegation structure,

$$M_k(\theta_k^i) = \max\{M_k(k), M_k(i)\}.$$

And the expected cost in each delegation structure is

$$cost_p(\theta_k^i) = P(k)M_k(\theta_k^i) + \sum_{j=i+1}^{n} P(j)M_j(j),$$

and minimum expected cost is

$$cost_p = \min_i cost_p(\theta_k^i).$$

Therefore, we can enumerate all the delegation structure and select the optimal one with the minimum expected cost in polynomial time.

### 2.4.4 Three Layers

This section we consider a process tree with three layers as in Fig. 2.20. The main question is what the optimal delegation structure is with the minimum expected cost?

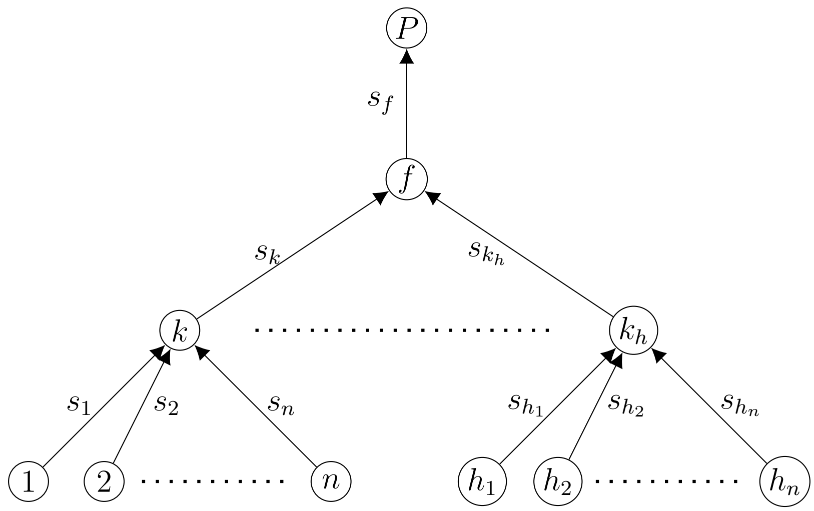



Fig. 2.20.: Three Layers Process tree

Denote the top agent as $f$, and the principal expects task $f$ is successful. By the Theorem 2.4.4 in the previous section, we have the efficient delegation structures from leaves to the second layer agent $k$, $\Theta(k)$. Without loss of generality, we can number the children of $k$ by an increasing order of $M_k(i)$, that is $M_k(i) \geq M_k(j), i \geq j$. To find the efficient delegation structure, we first consider the case that $k$ is the only child of $f$, as in Fig. 2.21.

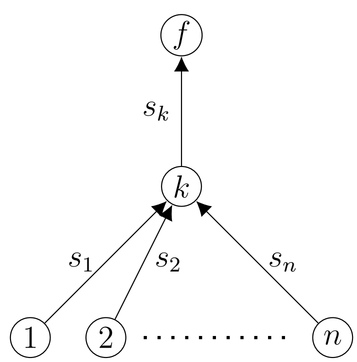



Fig. 2.21.: Update Step

We can construct a set of delegation structure for agent $f$ by

$$\theta_f^i = \theta_k^i, \theta_k^i \in \Theta(k),$$

along with $\theta_f^f = \{f\}$, we denote this set of delegation structure as $\Theta(f, k)$ and claim this set includes all the efficient delegation structures through agent $k$.

Lemma 2.4.1 $\Theta(f,k)$ *includes all the efficient delegation structures for agent $f$ over branch $k$.*

The proof is provided in Appendix 2.11.1.

By Lemma 2.4.1, we restrict the efficient delegation structures to the set $\Theta(f,k)$ with size bounded by $n+2$. For each $\theta_f^i \in \Theta(f,k)$, we compute $M_f(\theta_f^i)$ and remove the inefficient structure if exists any. For simplicity, we still use $\Theta(f,k)$ to denote the set of efficient delegation structure.

For each payment level $M_f^i = M_f(\theta_f^i)$, we use $\theta_f(\cdot)$ denote the inverse function of $M_f(\cdot)$,

$$\theta_f^i = \theta_f(M_f^i).$$

Till now we have all the efficient delegation structures $\Theta(f,k), k \in C(f)$. The next step is to combine all the branches and have the overall EDS.

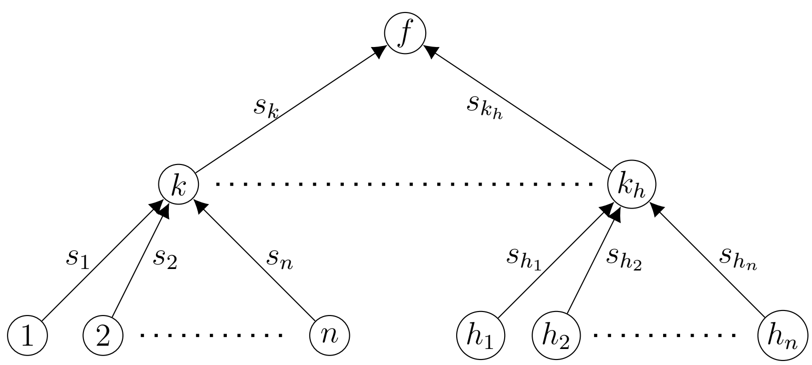



Fig. 2.22.: Combination Step

In order to do that, we first gather all the possible efficient payment level to $f$,

$$\mathcal{M}_f = \{M_f(\theta_f^i) | \theta_f^i \in \Theta(f,k), k \in C(f)\}.$$

Lemma 2.4.2 *The size of efficient payment level is bounded by $O(N)$.*

**Proof** There are at most $|C(k)+2|$ delegation structures in $\Theta(f,k)$ over each branch $k \in C(f)$, while each structure is corresponding to a payment level. Therefore, the total payment level is bounded by $O(n)$. ■

Given $M_f \in \mathcal{M}_f$, the efficient delegation structure is

$$\theta_f(M_f) = \sum_{k \in C(f)} \theta_{f,k}(M_f),$$

and we can construct a delegation set by

$$\Theta(f) = \{\theta_f(M_f) | M_f \in \mathcal{M}_f\}.$$

By Lemma 2.4.2, $|\Theta(f)| = O(N)$.

**Theorem 2.4.5** $\Theta(f)$ *can be ordered by inclusion and is the set of efficient delegation structures of agent* $f$*.*

Denote $cost_i$ as the expected minimum cost for the principal to motivate agent $i$ and his descendant in the process tree, call it as **expected minimum cost till** $i$. Similar to Theorem 2.3.5, we can prove the $cost_i$ is monotone increasing from leaf to the root in the process tree.

For each $M_f \in \mathcal{M}_f$, because $cost_i$ is monotone increasing, the principal would like to delegate as many agents as possible to $f$. Therefore, $k$'s control set is $\theta_f(M_f)$, and it's the largest one because $\Theta(f)$ can be ordered by inclusion. The corresponding minimum cost given $M_f$ is

$$cost_p(M_f) = M_f + \sum_{i \in C(\theta_f(M_f))} cost_i,$$

where $C(\theta_f)$ denotes the agents who don't belong to $\theta_f$ but are the children of nodes in $\theta_f$. Meanwhile, $i$ is either a leaf agent or a second level agent. If $i$ is a leaf agent, $cost_i = M_i(i)$. If $i$ is a second level agent, we can compute minimum cost till $i$ by the algorithm provided in the previous section.

Finally, the minimum expected cost is

$$cost_p = \min_{M_f \in \mathcal{M}_f} cost_p(M_f).$$

In summary, we provide a polynomial time algorithm to find the optimal delegation structure in three layers tree. Moreover, the result in this section also holds when the delegation tree is bounded by three layers, while the process tree can be deeper.

## 2.5 Conclusion

We considered a network model of sequential decisions in production chain networks, specifically chain and tree networks. Our main contribution is showing the value of delegation and how to maximumly leverage the middleman's aligned interests with the principal. In particular, we provide a polynomial time algorithm to find the optimal delegation structure and the corresponding necessary contract payments for the principal. Furthermore, we analyzed the trade-off of the delegation and gave a deeper insight into the value of delegation in different conditions. Several questions are left for future research such as what's the optimal delegation structures in general tree and how to build the model that agents can try multiple times until the task is successful.

# REFERENCES

[1] R. E. Kranton and D. F. Minehart, "A theory of buyer-seller networks," *American economic review*, vol. 91, no. 3, pp. 485–508, 2001.

[2] K. Bimpikis, S. Ehsani, and R. Ilkilic, "Cournot competition in networked markets." in *EC*, 2014, p. 733.

[3] R. Wright and Y.-Y. Wong, "Buyers, sellers, and middlemen: Variations on search-theoretic themes," *International Economic Review*, vol. 55, no. 2, pp. 375–397, 2014.

[4] T. Nguyen, "Local bargaining and supply chain instability," *Operations Research*, vol. 65, no. 6, 2017.

[5] J. Z. Pang, H. Fu, W. I. Lee, and A. Wierman, "The efficiency of open access in platforms for networked cournot markets," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[6] K. Kannan and T. Nguyen, "Welfare implications in intermediary networks," *Working paper*, 2018.

[7] F. Nava, "Efficiency in decentralized oligopolistic markets," *Journal of Economic Theory*, vol. 157, pp. 315–348, 2015.

[8] S. M. Carr and U. S. Karmarkar, "Competition in multiechelon assembly supply chains," *Management Science*, vol. 51, no. 1, pp. 45–59, 2005.

[9] K. Bimpikis, O. Candogan, and S. Ehsani, "Supply disruptions and optimal network structures," 2017.

[10] C. J. Corbett and U. S. Karmarkar, "Competition and structure in serial supply chains with deterministic demand," *Management Science*, vol. 47, no. 7, pp. 966–978, 2001.

[11] A. Federgruen and M. Hu, "Sequential multiproduct price competition in supply chain networks," *Operations Research*, vol. 64, no. 1, pp. 135–149, 2016.

[12] T. Nguyen, V. Subramanian, and R. Berry, "Delay in trade networks," *Operations Research*, vol. 64, no. 3, pp. 646–661, 2016.

[13] R. J. Duffin, "Topology of series-parallel networks," *Journal of Mathematical Analysis and Applications*, vol. 10, no. 2, pp. 303–318, 1965.

[14] R. Alonso and N. Matouschek, “Optimal delegation,” *The Review of Economic Studies*, vol. 75, no. 1, pp. 259–293, 2008.

[15] ——, “Relational delegation,” *The RAND Journal of Economics*, vol. 38, no. 4, pp. 1070–1089, 2007.

[16] G. Vereshchagina *et al.*, “The impact of moral hazard and budget balancing on sorting in partnerships,” working paper, Tech. Rep., 2016.

[17] Y.-K. Che and S.-W. Yoo, “Optimal incentives for teams,” *American Economic Review*, vol. 91, no. 3, pp. 525–541, 2001.

[18] A. Kaya and G. Vereshchagina, “Partnerships versus corporations: Moral hazard, sorting, and ownership structure,” *American Economic Review*, vol. 104, no. 1, pp. 291–307, 2014.

[19] A. M. Franco, M. Mitchell, and G. Vereshchagina, “Incentives and the structure of teams,” *Journal of Economic Theory*, vol. 146, no. 6, pp. 2307–2332, 2011.

[20] T. Nguyen and K. Kannan, “Welfare implications in intermediary networks,” 2018.

[21] T. Nguyen, “Coalitional bargaining in networks,” *Operations Research*, vol. 63, no. 5, pp. 501–511, 2015.

[22] M. Armstrong, “Delegating decision-making to an agent with unknown preferences,” *Mimeograph, University College London*, 1995.

[23] J. Levin, “Relational incentive contracts,” *American Economic Review*, vol. 93, no. 3, pp. 835–857, 2003.

# APPENDIX

## 2.6 Proofs in Section 1.3

### 2.6.1 Proof of Proposition 1.3.3

**Proof** Suppose $ij$ is a shortcut of path $l_{ij} = (i, v_1, ..., v_k, j)$, and assume the path $l_{ij}$ is active, i.e., every edge has postive flow.

Since firms never loss money in the supply chain (otherwise just choose to buy and sell nothing), we know

$$p_{v_1} \leq \cdots \leq p_{v_n} \leq p_j.$$

Considering the case that $p_{v_1} < p_j$ at the equilibrium, by the property of series parallel graph and market clearance price, all the flow from $i$ to $v_1$ will go through firm $j$. If firm $i$ moves all the flow $x_{iv_1}$ to $x_{ij}$, the total flow through $j$ will keep the same, and $p_j$ will remain the same price, too. Therefore, firm $i$ is better off by

$$\pi_i = p_j(x_{ij} + x_{iv_i}) > p_j x_{ij} + p_{v_1} x_{iv_1} = \pi_i^*,$$

which cannot happened at the equilibrium. Thus, $p_{v_1} = p_j$ must hold, and

$$p_{v_1} = \cdots = p_{v_n} = p_j.$$

Now consider the optimal decision for $v_n$, given the market clearance price $p_{v_n}$, if he buys all the goods supplied to him and sell them to $j$, his profit is 0, because $p_{v_n} = p_j$. However, he would make a positive profit if processed less amount of goods. Because this would decrease the flow to $j$ and raise the market price at $j$,

$$p'_j > p_j = p_{v_n},$$

which contradicts to the fact that $p_{v_n}$ is the market clearance price of firm $v_n$. Hence, the path $l_{ij}$ is inactive.

■

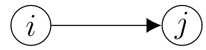
### 2.6.2 Proof of Lemma 1.3.1

**Proof** Suppose $j \in B(i)$ and by induction, assume

$$p_j = a_t - b_j X_j - \sum_{k \in C_P(j)} b_k X_k.$$

Obviously it is true when $j = t$, where $c_P(t) = \emptyset$.

**Case 1 (SS):** $|B(i)| = 1$ and $|S(j)| = 1$:




Utility function of $i$ is

$$\Pi_i = p_j x_{ij} - p_i x_{ij} - \frac{c_i}{2} x_{ij}^2.$$

To compute the price function at $i$, when $X_i > 0$, which means $x_{ij} > 0$, we have $\frac{\partial \Pi_i}{\partial x_{ij}} = 0$ so that $i$ can maximize its utility. Thus:

$$\begin{aligned}
p_i &= p_j + \frac{\partial p_j}{\partial x_{ij}} x_{ij} - c_i x_{ij} \\
&= a_t - b_j x_{ij} - \sum_{k \in C_P(j)} b_k X_k - (b_j + \sum_{k \in C_P(j)} b_k) x_{ij} - c_i x_{ij} \\
&= a_t - (2b_j + \sum_{k \in C_P(j)} b_k + c_i) X_i - \sum_{k \in C_P(i)} b_k X_k
\end{aligned}$$

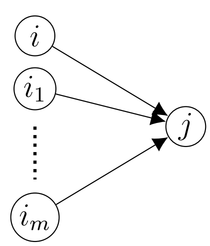
where $X_i = x_{ij}$, $C_P(i) = C_P(j)$ in this case.

$p_i$ is the market clearing price since from above equation, given $p_i$, we can solve the optimal $X_i$ too.

**Summary SS:**

$$\begin{aligned}
b_i &= 2b_j + \sum_{k \in C_P(j)} b_k + c_i, \\
C_P(i) &= C_P(j).
\end{aligned}$$

**Case 2 (MS):** $|B(i)| = 1$ and $|S(j)| \geqslant 1$:




Utility function of $i$ is

$$\Pi_i = p_j x_{ij} - p_i x_{ij} - \frac{c_i}{2} x_{ij}^2.$$

To compute the price function at $i$, when $X_i > 0$, which means $x_{ij} > 0$, we have $\frac{\partial \Pi_i}{\partial x_{ij}} = 0$ so that $i$ can maximize its utility. Thus

$$\begin{aligned}
p_i &= p_j + \frac{\partial p_j}{\partial x_{ij}} x_{ij} - c_i x_{ij} \\
&= a_t - b_j X_j - \sum_{k \in C_P(j)} b_k X_k - (b_j + \sum_{k \in C_P(j)} b_k) x_{ij} - c_i x_{ij} \\
&= a_t - (b_j + \sum_{k \in C_P(j)} b_k + c_i) x_{ij} - b_j X_j - \sum_{k \in C_P(j)} b_k X_k \\
&= a_t - (b_j + \sum_{k \in C_P(j)} b_k + c_i) X_i - \sum_{k \in C_P(i)} b_k X_k
\end{aligned}$$

where $X_i = x_{ij}$, $C_P(i) = C_P(j) \sqcup \{j\}$ in this case.

**Summary MS:**

$$\begin{aligned}
b_i &= b_j + \sum_{k \in C_P(j)} b_k + c_i, \\
C_P(i) &= C_P(j) \sqcup \{j\}.
\end{aligned}$$

**Case 3 (Simple SM):** $|B(i)| \geqslant 2$, $|S(j)| = 1$, and $C_S(i) = \{h\}$:

$$\begin{tikzcd}[row sep=small]
 & j_1 \\
i \arrow[ur] \arrow[dr] & \\
 & j_2
\end{tikzcd}$$

**Remark.** $X_k$ where $k \in C_S(i)$ is a function of $x_{ij}$. This is because market clearance price function ensures downstream firms will buy all the supply from upstream firms. Therefore, $x_{ij}$ is part of $X_k$.

Notice in this simple SM case, $C_P(j_1) = C_P(j_2)$ (by induction based on the compositions of SPG). Thereby, we just denote them as $C_P(j)$ in the following proof, and price functions are

$$\begin{aligned}
p_{j_1} &= a_t - b_{j_1} X_{j_1} - \sum_{k \in C_P(j)} b_k X_k, \\
p_{j_2} &= a_t - b_{j_2} X_{j_2} - \sum_{k \in C_P(j)} b_k X_k.
\end{aligned}$$

and the corresponding derivatives with respect to $x_{ij_1}$ are

$$\frac{\partial p_{j_1}}{\partial x_{ij_1}} = b_{j_1} + \sum_{k \in C_P(j_1)} b_k, \tag{2.42}$$

$$\frac{\partial p_{j_2}}{\partial x_{ij_1}} = \sum_{k \in C_P(j_1)} b_k. \tag{2.43}$$

Utility function of $i$ is

$$\Pi_i = p_{j_1} x_{ij_1} + p_{j_2} x_{ij_2} - p_i X_i - \frac{c_i}{2} X_i^2.$$

Because $i$ has multiple sub-flows and it is possible that some sub-flows are inactive, we will first prove the following claim.

**Claim:** For any firm $i$ in SPG, its sub-flows are all active.

At equilibrium, by $\frac{\partial \Pi_i}{\partial x_{ij_1}} \leqslant 0$, combined with price derivative equations 2.42:

$$\begin{aligned}
p_i &\geqslant p_{j_1} + \frac{\partial p_{j_1}}{\partial x_{ij_1}} x_{ij_1} + \frac{\partial p_{j_2}}{\partial x_{ij_1}} x_{ij_2} - c_i X_i \\
&= a_t - b_{j_1} x_{j_1} - \sum_{k \in C_P(j_1)} b_k X_k - (b_{j_1} + \sum_{k \in C_P(j_1)} b_k) x_{ij_1} - \sum_{k \in C_P(j_1)} b_k x_{ij_2} - c_i X_i \\
&= a_t - 2 b_{j_1} x_{ij_1} - (\sum_{k \in C_P(j_1)} b_k + c_i) X_i - \sum_{k \in C_P(j_1)} b_k X_k \\
&= p_{i_1}.
\end{aligned}$$

Similarly by $\frac{\partial \Pi_i}{\partial x_{ij_2}} \leqslant 0$:

$$\begin{aligned}
p_i &\geqslant a_t - 2 b_{j_2} x_{ij_2} - (\sum_{k \in C_P(j_2)} b_k + c_i) X_i - \sum_{k \in C_P(j_2)} b_k X_k \\
&= p_{i_2}
\end{aligned}$$

where $X_{j_1} = x_{ij_1}$, $X_{j_2} = x_{ij_2}$, and $C_P(j) = C_P(j_1) = C_P(j_2)$ in this case.

To prove both branches are active, first assume $x_{ij_1} > 0$ and $x_{ij_2} = 0$, then $p_i = p_{i_1}$ and

$$p_{i_2} - p_{i_1} = 2 b_{j_1} x_{ij_1} > 0 \Rightarrow p_{i_2} > p_{i_1} = p_i,$$

a contradiction. Same argument leads to a contradiction if we assume $x_{ij_1} = 0$ and $x_{ij_2} > 0$.

Suppose $x_{ij_1} = x_{ij_2} = 0$, then $X_i = x_{ij_1} + x_{ij_2} = 0$. By repeating that, we can prove all the parent nodes including source $s$ have zero flow, a contradiction. Thus, both sub-flows are active, and $p_i = p_{i_1} = p_{i_2}$. So far, the claim above is proved.

We know a convex combination of $p_{i_1}$ and $p_{i_2}$ is a necessary condition of $p_i$. By using the following convex combination coefficients:

$$\alpha_1 = \frac{\frac{1}{b_{j_1}}}{\frac{1}{b_{j_1}} + \frac{1}{b_{j_2}}}; \quad \alpha_2 = \frac{\frac{1}{b_{j_2}}}{\frac{1}{b_{j_1}} + \frac{1}{b_{j_2}}},$$

and $p_i$ can be written as function of $X_i = x_{ij_1} + x_{ij_2}$:

$$\begin{aligned} p_i &= \alpha_1 p_{i_1} + \alpha_2 p_{i_2} \\ &= a_t - b'_i x_{ij_1} - b'_i x_{ij_2} - \sum_{k \in C_P(j)} b_k X_k \\ &= a_t - b'_i X_i - \sum_{k \in C_P(j)} b_k X_k \end{aligned} \tag{2.44}$$

where

$$b'_i = \frac{2}{\frac{1}{b_{j_1}} + \frac{1}{b_{j_2}}} + \sum_{k \in C_P(j)} b_k + c_i.$$

Since $h$ is the only merging node ($C_S(i) = \{h\}$), the flow from $i$ will come through $h$ again, i.e. $X_h = X_i$. Also $C_P(j) = C_P(i) \cup \{h\}$ holds. Hence, coefficient $b_i$ is obtained from $b'_i + b_h$:

$$b_i = \frac{2}{\frac{1}{b_{j_1}} + \frac{1}{b_{j_2}}} + 2b_h + \sum_{k \in C_P(j) \backslash \{h\}} b_k + c_i.$$

Meanwhile, equation 2.44 can be written as the expected format:

$$p_i = a_t - b_i X_i - \sum_{k \in C_P(i)} b_k X_k \tag{2.45}$$

Note that the above argument can be generalized to $B(i) \geqslant 2$ easily. Suppose $B(i) = \{j_1, ..., j_m\}$, $m \geqslant 3$ and $|C_S(i)| = 1$ ($C_P(j_l)$ are all the same for $l = 1, ..., m$). By similar argument as in the previous claim, $ij, j \in B(i)$ must be active. The convex combination coefficient from price $p_{j_l}$ is

$$\alpha_l = \frac{\frac{1}{b_{j_l}}}{\sum_{j \in B(i)} \frac{1}{b_j}}.$$

Eventually, by similar reasoning:

$$b_i = \frac{2}{\sum_{j \in B(i)} \frac{1}{b_j}} + 2b_h + \sum_{k \in C_P(j) \backslash \{h\}} b_k + c_i.$$

**Summary Simple SM:**

$$b_i = \frac{2}{\sum_{j\in B(i)} \frac{1}{b_j}} + 2b_h + \sum_{k\in C_P(j)\setminus\{h\}} b_k + c_i \text{ where } h \text{ is the merging node.}$$

**Case 4 (General SM):** $|B(i)| \geqslant 3$, $|S(j)| = 1$, and $|C_S(i)| \geqslant 2$ (there are multiple self merging child nodes):

At equilibrium, by $\frac{\partial \Pi_i}{\partial x_{ij_1}} \leqslant 0$,

$$\begin{aligned}
p_i &\geqslant p_j + \sum_{l\in B(i)} \frac{\partial p_l}{\partial x_{il}} x_{il} - c_i X_i \\
&= a_t - b_j x_j - \sum_{k\in C_P(j)} b_k X_k - b_j x_{ij} - \sum_{l\in B(i)} \sum_{k\in C_P(l)} b_k x_{il} - c_i X_i \\
&= a_t - 2b_j x_{ij} - \sum_{h\in C_T(i,j)} b_h X_h - (\sum_{k\in C_P(i)} b_k + c_i) X_i - \sum_{k\in C_P(j)} b_k X_k \\
&= a_t - 2b_j x_{ij} - 2\sum_{h\in C_T(i,j)} b_h X_h - (\sum_{k\in C_P(i)} b_k + c_i) X_i - \sum_{k\in C_P(i)} b_k X_k \qquad (2.46) \\
&= p_{i_j}.
\end{aligned}$$

Similarly we can prove every sub-flow is active, and

$$p_i = a_t - 2b_j x_{ij} - 2\sum_{h\in C_T(i,j)} b_h X_h - (\sum_{k\in C_P(i)} b_k + c_i) X_i - \sum_{k\in C_P(i)} b_k X_k. \qquad \text{(GSM-p)}$$

At the same time, we have

$$b_i X_i = 2b_j x_{ij} + 2\sum_{h\in C_T(i,j)} b_h X_h + (\sum_{k\in C_P(i)} b_k + c_i) X_i. \qquad \text{(GSM-b)}$$

To write $p_i$ as in the form of

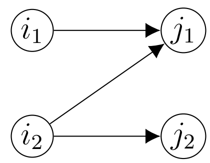
$$p_i = a_t - b_i X_i - \sum_{k\in C_P(i)} b_k X_k,$$

first note that for different $j \in B(i)$, $C_T(i,j)$ in equation 2.46 may be different. Therefore, we cannot merge these flows all together directly as in the previous case. Meanwhile, we can rank the nodes in $C_T(i,j)$ by the parent-child order as $h_1, \cdots, h_n$ where $h_t$ is the parent of $h_{t+1}$. By the property of merging nodes, we know:

- For every $j$, set $C_T(i,j)$ has the common last node $h^*$, and $X_i = X_{h^*}$.
- For every $j$, there exists a set $B_k(i) \subseteq B(i)$ whose nodes share the same $C_T(i,j)$. Denote $B_k(i) = \{h_1^k, h_2^k, \cdots, h^*\}$.

Instead of merging all the flows together, general SM case starts merging flows among each set $B_k(i)$. By similar reasoning to the simple SM case, merging among $B_k(i)$ can be done by using the convex coefficients $\alpha_l = \frac{\frac{1}{b_{j_l}}}{\sum_{j \in B_k(i)} \frac{1}{b_j}}$ for $j_l \in B_k(i)$. We create an *aggregate variable* $b_{B_k(i)} = \frac{1}{\sum_{j \in B_k(i)} \frac{1}{b_j}} + b_{h_1^k}$ to represent the coefficient for flow $X_{h_1^k} = \sum_{j \in B_k(i)} x_{ij}$. Afterwards, we group the new aggregated flows $X_{h_1^k}$ by the same $C_T(i, h_1^k)$, and repeat the above merging operation again for $h_2$, $h_3$, and so on. Once $h^*$ is reached, by applying equation SM, we have the final coefficient $b_i$ for node $i$. Example 15 in the appendix shows the general SM computation.

**Case 5 (MM):** $|B(i)| \geqslant 2$, $|S(j)| \geqslant 2$:




This is impossible in an SPG, proved by induction since any SPG can be constructed by series and parallel insertion:

- Series insertion: it is easy to see MM will not appear after this.
- Parallel insertion: check the merging head and tail, and it is easy to see MM will not appear either.

Therefore, MM never happens in an SPG. ■

### 2.6.3 Proof of proposition 1.3.5

**Proof** Suppose $i$ sells to $j$, we finish the proof by discussion over case by case. For the SS case, by equation SS:

$$p_j - p_i = (b_j + \sum_{k \in C_P(i)} b_k + c_i) X_i.$$

If $X_i = x_{ij} > 0$, then $p_j > p_i$.

For the SM case, by equation GSM-p:

$$p_j - p_i = b_j x_{ij} + \sum_{h \in C_T(i,j)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i + c_i X_i.$$

If $x_{ij} > 0$, then we prove $p_j > p_i$.

For the MS case, if $X_i = x_{ij} > 0$, by equation MS:

$$p_j - p_i = b_i X_i > 0.$$

■

### 2.6.4 Proof of Lemma 1.3.2

**Proof** Consider the Lagrangian function:

$$L(x_{ij}, X_s, X_k, \lambda_{ij}) = \sum_{j \in B(i)} b_j x_{ij}^2 + \sum_{k \in C_S(i) \backslash C_P(i)} b_k X_k^2 - \sum_{j \in B(s)} \lambda_{ij}(a_t - 2b_j x_{ij} - \sum_{k \in C_T(i,j)} 2b_k X_k - const - p_s).$$

**Stationarity condition:**

- Take the derivative with respect to $x_{ij}$:

$$\frac{\partial L(x_{ij}, X_j, X_k, \lambda_{ij})}{\partial x_{ij}} = 2b_j x_{ij} - 2b_j \lambda_{ij} = 0$$

  infers $x_{ij} = \lambda_{ij}$.

- Take derivative with respect to $X_k$ where $k \in C_S(i) \backslash C_P(i)$:

$$\frac{\partial L(x_{ij}, X_j, X_k, \lambda_{ij})}{\partial X_k} = 2b_k X_k - \sum_{j: k \in C_P(j)} 2b_k \lambda_{ij} = 0$$

  infers $X_k = \sum_{j:k \in C(j)} \lambda_{ij} = \sum_{j:k \in C(j)} x_{ij}$, which is exactly the definition of $X_k$ (the total flow through $k$).

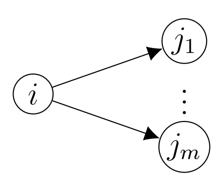
**Complimentary condition:**
$\forall j \in B(s)$ (recall $x_{ij} = \lambda_{ij}$):

$$\lambda_{ij}(a_t - 2b_j x_{ij} - \sum_{k \in C_T(i,j)} 2b_k X_k - const - p_s) = x_{ij} \frac{\partial \Pi_i}{\partial x_{ij}} = 0.$$

Combined with the primal feasibility conditions $\frac{\partial \Pi_i}{\partial x_{ij}} \leqslant 0$ and $x_{ij} \geqslant 0$, we can see the KKT condition of this convex programming is equivalent to the LCP. Meanwhile, this problem is strictly convex, so the solution is unique. ■

### 2.6.5 Proof of Lemma 1.3.3

**Proof** We consider the SM case: $B(i) = \{j_1, ..., j_m\}$ where $m \geqslant 2$.



The decision variables of $i$ are $x_{ij}$'s where $j \in B(i)$. Recall equation 1.8:

$$\frac{\partial \Pi_i}{\partial x_{ij}} = a_t - 2b_j x_{ij} - 2 \sum_{k \in C_T(i,j)} b_k X_k - p_i - const.$$

Notice that $X_k = \sum_{j:j \in B(i) \text{ and } k \in C_P(j)} x_{ij}$ and $\frac{\partial \Pi_i}{\partial x_{ij}} = 0$ for all $j \in B$ because $ij$'s are all active, we can rewrite equation 1.8 as a linear system in the following form:

$$A\vec{x} = (a_t - p_i - const)\vec{1} \tag{2.47}$$

where $\vec{1}$ is a vector of $m$ ones, $\vec{x} = [x_{ij_1}, ..., x_{ij_m}]^T$, and $A \in \mathbb{R}^{m \times m}$.

First we prove that $A$ is symmetric. Consider $A_{l_1 l_2}$ and $A_{l_2 l_1}$ where $l_1 \neq l_2$, we have

$$A_{l_1 l_2} = 2 \sum_{k \in C_S(i) \cup (C(j_{l_1}) \cap C(j_{l_2})) \setminus C_P(i)} b_k = A_{l_2 l_1},$$

so $A$ is symmetric.

Recall that in Algorithm 1, before computing $p_i$, we had $p_j = a_t - b_j X_j - \sum_{k \in C_P(j)} b_k X_k$ for $j \in B(i)$. The utility of $i$ is

$$\Pi_i = \sum_{j \in B(i)} p_j x_{ij} - p_i X_i - \frac{c_i}{2} X_i^2.$$

By Lemma 1.3.1 and equation 1.8, since $ij$'s are all active, we have $\frac{\partial \Pi_i}{\partial x_{ij}} = 0$. Therefore

$$p_i = a_t - 2b_j x_{ij} - 2 \sum_{k \in C_T(i,j)} b_k X_k - [(\sum_{k \in C_P(i)} b_k + c_i) X_i + \sum_{k \in C_P(i)} b_k X_k].$$

Denote the later part, $(\sum_{k \in C_P(i)} b_k + c_i) X_i + \sum_{k \in C_P(i)} b_k X_k$, as $L$. Note that in Algorithm 1, $L$ is some unknown value different from the constant pre-computed in Algorithm 2. However, $L$ will not be effected by the convex coefficients, since we only care about the nodes between $i$ and the last self merging node of $i$.

Let $p_{i_l}$ be the price equation after taking derivative with respect to $x_{ij_l}$. Then in Algorithm 1, we had the convex coefficients $\alpha_1, ..., \alpha_m$ such that $\sum_{l=1}^m \alpha_l = 1$ and

$$p_i = \sum_{l=1}^m \alpha_l p_{i_l} = a_t - \sum_{l=1}^m \alpha_l A_l \vec{x} - L = a_t - b_i X_i - L$$

where $A_l$ is the $l$-th row of $A$ and $X_i = \sum_{j \in B(i)} x_{ij}$.

Note that for any $j \in B(i)$, the coefficient of $x_{ij}$ is $\sum_{l=1}^{m} \alpha_l A_{lj} = b_i$. Since $A$ is symmetric, this can be presented as the following:

$$A^T \vec{\alpha} = A\vec{\alpha} = b_i \vec{1} \tag{2.48}$$

where $\vec{\alpha} = [\alpha_1, ..., \alpha_m]^T$.

By comparing equation 2.47 and equation 2.48, we know $\vec{x}$ is proportional to $\vec{\alpha}$.

To prove that all the price value $p_j$ for $j \in B(i)$ are the same, we can also rewrite equation 1.8 to obtain a relation between $\frac{\partial \Pi_i}{\partial x_{ij}}$ and $p_j$:

$$\begin{aligned}
\frac{\partial \Pi_i}{\partial x_{ij}} &= a_t - 2b_j x_{ij} - 2 \sum_{h \in C_T(i,j)} b_h X_h - p_i - (\sum_{k \in C_P(i)} b_k + c_i) X_i - \sum_{k \in C_P(i)} b_k X_k \\
&= 2(a_t - b_j x_{ij} - \sum_{k \in C_P(j)} b_k X_k) - a_t - (\sum_{k \in C_P(i)} b_k + c_i) X_i + \sum_{k \in C_P(i)} b_k X_k - p_i \\
&= 2p_j - const' - p_i
\end{aligned} \tag{2.49}$$

where $const' = a_t + (\sum_{k \in C_P(i)} b_k + c_i) X_i - \sum_{k \in C_P(i)} b_k X_k$. From equation 2.49 and the fact that all edges are active, we know that

$$0 = 2p_j - const' - p_i.$$

Therefore, $p_j = \frac{p_i + const'}{2}$ for any $j \in B(i)$.

■

## 2.7 Proofs in Section 1.4

### 2.7.1 Proof of Proposition 1.4.1

**Proof** For simplicity, we just consider the case without processing cost, and the proof can be extended to the case with processing cost easily. Suppose the market price function is $p_t = a_t - b_t X_t$, for single-edge graph, the utility is $\Pi_s = p_t x - p_s x$. At equilibrium, $\frac{\partial \Pi_s}{\partial x} = 0$ infers $p_s = a_t - 2b_t X_s$.

For general SPG, proof by induction. From $ij \in E$, it is easy to see for the SS case, $b_i \geqslant 2b_j$, and for the MS case $b_i \geqslant b_j$ by the proof in Appendix 2.6.2. For the simple SM case:

$$b_i = \frac{2}{\sum_{j \in B(i)} \frac{1}{b_j}} + 2b_h + \sum_{k \in C_P(j) \setminus \{h\}} b_k > 2b_h \geqslant 2b_t$$

where $h$ is the merging node.

Meanwhile, it is easy to show it also holds for general SM case. Therefore, it always holds that $b_i \geqslant 2b_t$ if $ij \in E$ is the SS case or SM case. Note that $s$ is the only source so $b_s \geqslant 2b_t$ for general SPG. The total flow satisfies

$$p_s = a_s + d_s X_s = a_t - b_s X_s \Rightarrow X_S = \frac{a_t - a_s}{d_s + b_s}.$$

$b_s = 2b_t$ only holds in the single-edge graph and $b_s \geqslant 2b_t$ in any other SPG. Therefore, the single-edge graph is the most efficient SPG supply chain network. ■

### 2.7.2 Proof of Proposition 1.4.2

**Proof** From Lemma 1.3.1:

$$p_s = a_t - b_s X_s = a_s + d_s X_s \text{ (the given source price).}$$

It follows that $X_s = \frac{a_t - a_s}{d_s + b_s}$, so the increasing demand at market $(a_t)$ or decreasing cost at the source $(a_s$ or $d_s)$ will make the supply chain more efficient. ■

### 2.7.3 Proof of Lemma 1.4.1

**Proof** By Lemma 1.3.1, we know that $p_s = a_t - b_s X_s$. While calculating the price function from sink, $b_i$ where $i \in V$ changes proportionally to $b_t$ since there is no "offset" $c_i$.

By Proposition 1.4.1, the most efficient network is the single-edge graph and $b_s = 2b_t$. For general SPG, $b_s \geqslant 2b_t$ since it is less efficient and the source price is a given value. ■

### 2.7.4 Proof of Theorem 1.4.1

**Proof** Consider series components $X$ and $Y$, and the larger component $G' = P(X, Y)$, where $t_x = s_y$, $s' = s_X$, and $t' = t_y$.

By lemma 1.4.1:

$$\begin{aligned} b_{s'} &= \frac{b_{s_X}}{b_{t_x}} \frac{b_{t_x}}{b_{t_y}} b_{t_y} \\ &= \lambda(X)\lambda(Y) b_{t'}. \end{aligned}$$

Now if we change the order of this components, and let $s_X = t_y$, $s' = s_y$, $t' = t_x$, then

$$\begin{aligned} b_{s'} &= \frac{b_{s_y}}{b_{t_y}} \frac{b_{t_y}}{b_{t_x}} b_{t_x} \\ &= \lambda(Y)\lambda(X) b_{t'}. \end{aligned}$$

Thus, we can consider $X$ and $Y$ as one components and switching the inner order does not change the slope

$$b_{s'} = \lambda(X)\lambda(Y) b_{t'} = \lambda(G') b_{t'}$$

and does not change the price function of the other components. Thus, the total flow remains the same. ■

### 2.7.5 Proof of Proposition 1.4.3

For the case with processing cost, $\lambda(\cdot)$ is a function of $b_t$, and we first prove the following lemma.

LEMMA 2.7.1 *With processing cost, for any $\alpha \leqslant 1$,*

$$\lambda(X, \alpha b_t) \geqslant \alpha \lambda(X, b_t).$$

*For any $\alpha \geqslant 1$,*

$$\lambda(X, \alpha b_t) \leqslant \alpha \lambda(X, b_t).$$

**Proof** For any $\alpha \leqslant 1$, we proved it by induction, starts from $t$, and consider its buyer, which must be SS or MS cases.

For the SS case, by equation SS:

$$b'_i = 2\alpha b_t + \sum_{k \in C_P(t)} b_k + c_t \geqslant \alpha b_i.$$

For the MS case, by equation MS:

$$b'_i = \alpha b_t + \sum_{k \in C_P(t)} b_k + c_t \geqslant \alpha b_i.$$

For the SM case, by induction, $b'_j \geqslant \alpha b_j, j \in B(i)$, by equation SM:

$$\begin{aligned} b'_i &= \frac{2}{\sum_{j\in B(i)} \frac{1}{b'_j}} + 2b'_h + \sum_{k\in C_P(j)\setminus\{h\}} b_k + c_i \\ &\geqslant \frac{2\alpha}{\sum_{j\in B(i)} \frac{1}{b_j}} + 2\alpha b_h + \sum_{k\in C_P(j)\setminus\{h\}} b_k + c_i \\ &\geqslant \alpha b_i. \end{aligned}$$

Similar result applies to the general SM case. Therefore, $\lambda(X, \alpha b_t) = b'_s \geqslant \alpha b_s = \lambda(X, b_t)$.

The proof when $\alpha \geqslant 1$ is very similar thus it is omitted here. ■

Now we begin to prove the proposition.

**Proof** Denote $S(X,Y)$ and $S(Y,X)$ as SPG 1 and SPG 2. By Lemma 1.3.1, let $a_t - b_s^1 X_s$ be the source price of SPG 1 and $a_t - b_s^2 X_s$ be the source price of SPG 2.

We prove $b_s^1 \leqslant b_s^2$ as follows:

$$\begin{aligned} b_s^1 &= \lambda(X, \lambda(Y, b_t)) \\ &= \lambda(X, \lambda(Y) b_t) \\ &\leqslant \lambda(Y)\lambda(X, b_t) \\ &= \lambda(Y, \lambda(X, b_t)) \\ &= b_s^2 \end{aligned}$$

where the second and second last inequality used Lemma 1.4.1, the third inequality used Lemma 2.7.1 with $\lambda(Y) \geqslant 1$.

Then the flow of SPG 1 is $X_s^1 = \frac{a_t - p_s}{b_s^1 + d_s}$, which is larger than the flow of SPG 2 $X_s^2 = \frac{a_t - p_s}{b_s^2 + d_s}$. Hence, SPG 1 is more efficient. ■

### 2.7.6 Proof of Proposition 1.4.4

**Proof** Consider $n$ agents in the straight line model, suppose the firms are labeled by the order as $0, 1, \ldots, n$, where 0 is the source, and $n$ is the sink.

Under market clearance price, every node has the same inflow and outflow, denoted as $x$. The utility function for agent $i$ is

$$\Pi_i = (a_{i+1} - b_{i+1}x)x - p_i x - \frac{c_i}{2}x^2,$$

and its derivative is

$$\frac{\partial \Pi_i}{\partial x} = a_{i+1} - (2b_{i+1} + c_i)x - p_i.$$

Since $x > 0$, $\frac{\partial \Pi_i}{\partial x}$ and we have

$$p_i = a_{i+1} - (2b_{i+1} + c_i)x,$$

the following update rule holds:

$$\begin{aligned} a_i &= a_{i+1}, \\ b_i &= 2b_{i+1} + c_i, \end{aligned}$$

and we can use this to compute the source price function:

$$\begin{aligned} a_0 &= a_n, \\ b_0 &= 2^n b_n + \sum_{i=1}^{n} 2^i c_i. \end{aligned}$$

The coefficient of $c_i$ is $2^i$ with $i$ (closer to the sink). Consequently, putting the node with a higher processing cost $c_i$ closer to source results in better efficiency. ■

### 2.7.7 Proof of Theorem 1.4.2

We need to prove the following two lemmas first, based on the $b_i$ computation from 2.6.2. Let $b'_i$ be the slope coefficient of $i$ after the insertion.

Lemma 2.7.2 *Series insertion on node $i$ always increases the price function slope $b_k$ where $k \in S(i) \cup i$.*

**Proof** After a series insertion on node $i$, we know $b'_i > b_i$ since by Lemma 1.3.1, $b_i > b_j$ if $ij \in E$ for the SS case and the SM case. By induction and the proof of Lemma 1.3.1, we know $b'_k > b_k, \forall k \in S(i)$. Finally, $b'_s > b_s$ infers the total flow decreases. ■

Lemma 2.7.3 *Parallel insertion on $path_{ij}$ always decreases the price function slope $b_k$ where $k \in S(i) \cup i$.*

**Proof** After a parallel insertion on $path_{ij}$, by case SM in Lemma 1.3.1, the new slope $b'_i$ satisfies $b'_i < b_i$. By induction and the proof of Lemma 1.3.1, we know $b'_k < b_k, \forall k \in S(i)$. Finally, $b'_s < b_s$ infers the total flow increases. ■

**Proof** Suppose the original price function at source is $p_s = a_t - b_s X_s$. If the raw material is sold at price $p_s$, then at equilibrium:

$$X_s = \frac{a_t - a_s}{d_s + b_s}.$$

By Lemma 2.7.2, after series insertion, $b'_s > b_s$, then we know the total inflow at equilibrium is decreased:

$$X'_s = \frac{a_t - a_s}{d_s + b'_s} < \frac{a_t - a_s}{d_s + b_s} = X_s.$$

While after parallel insertion, $b'_s < b_s$ by Lemma 2.7.3, thus the total inflow at equilibrium is increased:

$$X'_s = \frac{a_t - a_s}{d_s + b'_s} > \frac{a_t - a_s}{d_s + b_s} = X_s.$$

■

### 2.7.8 Proof of Lemma 1.4.2

**Proof** For global parallel insertion, the only common child of two branches $X, Y$ is $\{t\}$, denote the new coefficient at $s$ as $b_s^G$:

$$\begin{aligned} b_s^G &= \frac{1}{\frac{1}{b_s - c_s - 2b_t} + \frac{1}{b}} + c_s + 2b_t \\ &= f(b_s^0, b)b_s^0 + c_s + 2b_t \end{aligned}$$

where $b_s^0 = b_s - c_s - 2b_t$ and define $f(x, y) = \frac{y}{x+y}$.

- Local insertion on component $X_2$, denote the new coefficient at $s$ as $b_s^{L2}$:

  $$b_2^{L2} = f(b_2^0, b)b_2^0 + c_2 + 2b_t$$

  where $b_2^0 = b_2 - c_2 - 2b_t$.

  Since $b_s - c_s > b_2$, we know $b_s^0 > b_2^0$. Thus, $f(b_s^0, b) < f(b_2^0, b)$, and by induction (similar to the proof of proposition 2.7.5), we can prove

  $$\begin{aligned} b_s^{L2} &\geq f(b_2^0, b)b_s^0 + c_s + 2b_t \\ &\geqslant f(b_s^0, b)b_s^0 + c_s + 2b_t \\ &= b_s^G. \end{aligned}$$

  Therefore, global parallel insertion is more efficient than local parallel insertion $P(X_2, Y)$.

- Local insertion on component $X_1$, denote the new coefficient at $s$ as $b_s^{L1}$:

  $$b_s^{L1} = \frac{1}{\frac{1}{b_s - c_s - 2b_2} + \frac{1}{b'}} + c_s + 2b_2.$$

Because $b_2 > b_t$, we have $b' > b$, thus

$$b_s^{L1} \geqslant \frac{1}{\frac{1}{b_s - c_s - 2b_2} + \frac{1}{b}} + c_s + 2b_2.$$

Furthermore, by the fact that $(t < x)$,

$$\frac{1}{\frac{1}{x-t} + \frac{1}{y}} + t \geq \frac{1}{\frac{1}{x} + \frac{1}{y}}.$$

Again, since $b_t < b_2$,

$$\begin{aligned} b_s^{L1} &\geq \frac{1}{\frac{1}{b_s - c_s - 2b_2} + \frac{1}{b}} + c_s + 2b_2 \\ &\geqslant \frac{1}{\frac{1}{b_s - c_s - 2b_t} + \frac{1}{b}} + c_s + 2b_t \\ &= b_s^G. \end{aligned}$$

Therefore, global parallel insertion is more efficient than local parallel insertion $P(X_1, Y)$. ■

### 2.7.9 Proof of Lemma 1.4.3

**Proof**

- For the SS case, $X_i = X_j = x_{ij}, C_P(i) = C_P(j)$. Consider the utility of $i$, by equation SS:

$$\begin{aligned} \Pi_i &= (p_j - p_i)x_{ij} - \frac{c_i}{2}X_i^2 \\ &= (b_i X_i - b_j X_j)x_{ij} - \frac{c_i}{2}X_i^2 \\ &= (b_i - \frac{b_i - \sum_{k \in C_P(i)} b_k - c_i}{2})X_i^2 - \frac{c_i}{2}X_i^2 \\ &= \frac{1}{2}(b_i + \sum_{k \in C_P(i)} b_k)X_i^2. \end{aligned}$$

- For the SM case, $X_j = x_{ij}$. Consider the utility of $i$, by equation GSM-p:

$$\begin{aligned} \Pi_i &= \sum_{j \in B(i)} (p_j - p_i)x_{ij} - \frac{c_i}{2}X_i^2 \\ &= \sum_{j \in B(i)} (b_j x_{ij} + \sum_{h \in C_T(i,j)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i + c_i X_i)x_{ij} - \frac{c_i}{2}X_i^2. \end{aligned}$$

By equation GSM-b:

$$\begin{aligned}\Pi_i &= \frac{1}{2}\sum_j (b_i X_i + \sum_{k\in C_P(i)} b_k X_i + c_i X_i) x_{ij} - \frac{c_i}{2} X_i^2 \\ &= \frac{1}{2}(b_i + \sum_{k\in C_P(i)} b_k) X_i^2.\end{aligned}$$

- For the MS case, $C_P(i) = C_P(j) \cup j$.

$$\begin{aligned}\Pi_i &= (p_j - p_i)X_i - \frac{c_i}{2}X_i^2 \\ &= (a_t - b_j X_j - \sum_{k\in C_P(j)} b_k X_k - a_t + b_i X_i + \sum_{k\in C_P(i)} b_k X_k) X_i - \frac{c_i}{2} X_i^2 \\ &= b_i X_i^2 - \frac{c_i}{2} X_i^2. \end{aligned} \tag{2.50}$$

By equation MS:

$$\begin{aligned} b_i &= b_j + \sum_{j\in C_P(j)} b_k + c_i \\ &= \sum_{j\in C_P(i)} b_k + c_i. \end{aligned} \tag{2.51}$$

Plug equation 2.51 into equation 2.50:

$$\begin{aligned}\Pi_i &= \frac{1}{2}(b_i + \sum_{k\in C_P(i)} b_k + c_i) X_i^2 - \frac{c_i}{2} X_i^2 \\ &= \frac{1}{2}(b_i + \sum_{k\in C_P(i)} b_k) X_i^2.\end{aligned}$$

■

### 2.7.10 Proof of Theorem 1.4.4

**Proof** For the SS or SM case, since $C_P(i) \subset C_P(j)$, $X_i \geqslant X_j = x_{ij}$. Plug equation GSM-b into the utility function of $i$ as in equation 1.9:

$$\begin{aligned}
\Pi_i &= \frac{1}{2}(b_i X_i + \sum_{k \in C_P(i)} b_k X_i) X_i \\
&\geqslant \frac{1}{2}(2 b_j x_{ij} + 2 \sum_{h \in C_T(i,j)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i + \sum_{k \in C_P(i)} b_k X_i) X_i \\
&\geqslant (b_j X_j + \sum_{k \in C_P(j)} b_k X_j) X_j \\
&= 2\Pi_j
\end{aligned}$$

where the second inequality holds because $X_i \geqslant X_j$ and $C_P(j) = C_P(i) \sqcup C_T(i,j)$.

Now suppose there is MS relation to $j$, consider the the closest dominate parent $i$ of $j$. Let $l \in B(i)$, and $j \in C(l)$. Then

$$C_P(l) = C_T(i,l) \sqcup C_P(i) = C_P(j) \sqcup \{j\}.$$

Combine this with equation 1.9:

$$\begin{aligned}
\Pi_i &= \frac{1}{2}(b_i X_i + \sum_{k \in C_P(i)} b_k X_i) X_i \\
&\geqslant \frac{1}{2}(2 \sum_{h \in C_T(i,l)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i + \sum_{k \in C_P(i)} b_k X_i) X_i \\
&\geqslant (b_j X_j + \sum_{k \in C_P(j)} b_k X_j) X_j \\
&= 2\Pi_j.
\end{aligned}$$

■

### 2.7.11 Proof of Corollary 1.4.1

**Proof** By equation GSM-p:

$$
\begin{aligned}
\Pi_i &= \sum_{j \in B(i)} (p_j - p_i) x_{ij} - \frac{c_i}{2} X_i^2 \\
&= \sum_{j \in B(i)} (b_j x_{ij} + \sum_{h \in C_T(i,j)} b_h X_h + \sum_{k \in C_P(i)} b_k X_i + c_i X_i) x_{ij} - \frac{c_i}{2} X_i^2 \\
&\geqslant \sum_{j \in B(i)} (b_j X_j + \sum_{h \in C_T(i,j)} b_h X_j + \sum_{k \in C_P(i)} b_k X_j) X_j \\
&= \sum_{j \in B(i)} (b_j + \sum_{k \in C_P(j)} b_k) X_j^2 \\
&= \sum_{j \in B(i)} \Pi_j
\end{aligned}
$$

where the second last equality is because $C_P(j) = C_P(i) \sqcup C_T(i,j)$, and the last equality is by equation 1.9. ■

## 2.8 Proofs in Section 1.5

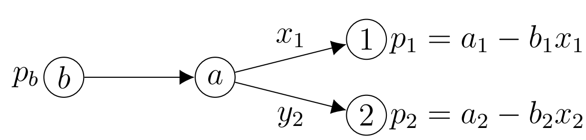
### 2.8.1 Proof of Theorem 1.5.2

**Proof** Proof by contradiction to show all edges are active. Suppose there is an inactive market $t$, then there exists an active firm $i$ such that for any path from $i$ to $t$, the edges in the path are all inactive. Similar to the proof of Lemma 1.3.1, the price of every firm $j$ can be presented as a function like $a_t$ minus the sum of some constants time $X_j$ and $X_k$ where $k \in C_P(j)$. If $ij \in E$ is on the path from $i$ to $t$, then $p_j = a_t$ since $X_j = x_{ij} = 0$ and by the structure of SPG, $X_k = 0$ for any $k \in C_P(j)$. $i$ as an active firm must have sold some goods to another firm $k$ with price less than $a_t$. However, $i$ could have just sold the goods to $j$ with a higher price to increase its utility  a contradiction.

From the fact that every edge is active, we have a unique price function for each firm, and similar to Theorem 1.3.1, we can prove the supply quantities at equilibrium is also unique. ■

### 2.8.2 Proof of Remark 1.5.2

We consider the following supply chain network:



For convenience, we denote the first market price as $p_1$ and the second market price as $p_2$. The production cost is a constant $p_b$. Suppose the two price functions at the markets are:

$$\begin{aligned} p_1 &= a_1 - b_1 x_1, \\ p_2 &= a_2 - b_2 x_2, \end{aligned}$$

where $a_1 \geqslant a_2 \geqslant p_b$.

- Supply chain under low price strategy is always more efficient than under high price strategy.

  **Proof** Optimal flow $X_h$ at high price strategy is

  $$\begin{aligned} p_b &= a_1 - 4b_1 X_h, \\ X_h &= \frac{a_1 - p_b}{4b_1}. \end{aligned}$$

  Optimal flow $X_l$ at low price strategy is

  $$\begin{aligned} p_a &= (a_1/b_1 + a_2/b_2)B - 2BX_l, \\ p_b &= (a_1/b_1 + a_2/b_2)B - 4BX_l, \\ X_l &= \frac{(a_1/b_1 + a_2/b_2)B - p_b}{4B}, \end{aligned}$$

  where $B = \frac{1}{\frac{1}{b_1} + \frac{1}{b_2}}$.

  Then we have the difference of total flow between these two strategies:

  $$\begin{aligned} X_l - X_h &= \frac{(a_1/b_1 + a_2/b_2)B - p_s}{4B} - \frac{a_1 - p_b}{4b_1} \\ &= \frac{a_2}{4b_2} - \frac{p_b}{4B} + \frac{p_b}{4b_1} \\ &= \frac{a_2}{4b_2} - \frac{p_b}{4b_1} - \frac{p_b}{4b_2} + \frac{p_b}{4b_1} \\ &= \frac{a_2}{4b_2} - \frac{p_b}{4b_2} \\ &\geqslant 0. \end{aligned}$$

  ■

- When the demand difference between two markets is small enough, low price strategy gives better payoff for source firm. If the difference is large enough, high price strategy gives better payoff for source firm.

**Proof** Let $CS$ be the consumer surplus, $PS_a$ be the surplus of firm $a$, $PS_b$ be the surplus of firm $b$, $SW$ be the social welfare. At the high price strategy:

$$\begin{aligned}
&CS = \frac{1}{2}b_1 X_h^2,\\
&PS_a = b_1 X_h^2,\\
&PS_b = 2b_1 X_h^2 = \frac{(a_1 - p_b)^2}{8b_1},\\
&SW = CS + PS_a + PS_s = \frac{7}{2}b_1 X_h^2 = \frac{7}{2}b_1(\frac{a_1 - p_b}{4b_1})^2 = \frac{7(a_1 - p_b)^2}{32b_1}.
\end{aligned}$$

For social welfare at the low price strategy, let $x_1$ be the inflow of the first market and $x_2$ be the inflow of the second market. From the flow relation:

$$\begin{aligned}
&a_1 - 2b_1 x_1 = a_2 - 2b_2 x_2,\\
&x_1 + x_2 = X_h,
\end{aligned}$$

infers

$$\begin{aligned}
x_1 &= \frac{a_1 - a_2 + 2b_2 X_h}{2b_1 + 2b_2},\\
x_2 &= \frac{2b_1 X_h - a_1 + a_2}{2b_1 + 2b_2},\\
X_h &= \frac{(a_1/b_1 + a_2/b_2)B - p_b}{4B}.
\end{aligned}$$

Therefore

$$\begin{aligned}
&CS = \frac{1}{2}b_1 x_1^2 + \frac{1}{2}b_2 x_2^2,\\
&PS_a = b_1 x_1^2 + b_2 x_2^2,\\
&PS_b = 2BX_l^2 = 2B[\frac{(a_1/b_1 + a_2/b_2)B - p_b}{4B}]^2 = \frac{[(a_1/b_1 + a_2/b_2)B - p_b]^2}{8B},\\
&SW = CS + PS_a + PS_b = \frac{7}{2}b_1 X_h^2.
\end{aligned}$$

Notice $PS_b \leqslant PS_a$ in this case.

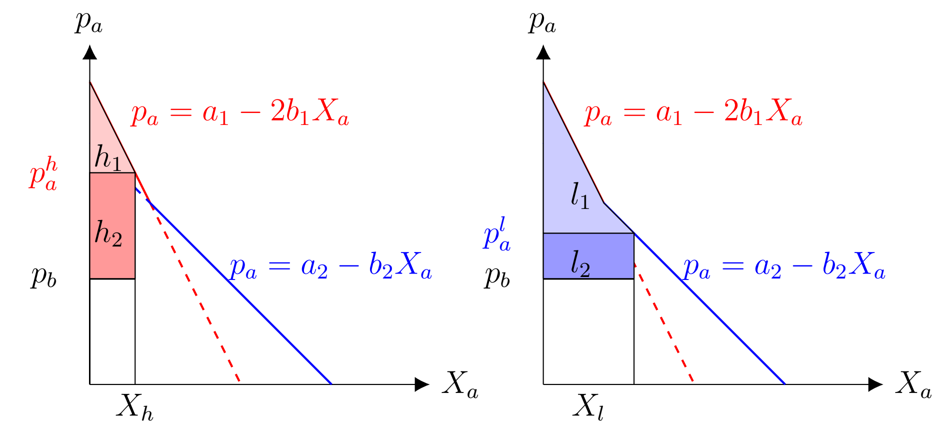
However, to prove this statement, we only need:

$$\frac{PS_b^h}{PS_b^l} = \frac{b_1 X_h^2}{BX_l^2} = \frac{b_1 + b_2}{b_2}\frac{X_h}{X_h + \Delta}$$

where $PS_b^h$ is the surplus of $b$ at high price and $PS_b^l$ is the surplus of $b$ at low price, and $\Delta = \frac{a_2}{4b_2} - \frac{p_b}{4b_2}$ is irrelevant to $a_1$. Therefore, as $a_1$ increases, value $\frac{PS_s^h}{PS_s^l}$ increases from less than 1 to greater than 1. ∎

- Low price strategy always produces a higher total surplus of firms and consumers. Hence, social welfare is also higher.

  **Proof** We will use a proof by picture. Consider the following figure:




  - For high price strategy, the area of the upper triangle $h_1$ is $PS_a^h$, while the area of the lower rectangle $h_2$ is $PS_b^h$.
  - For low price strategy, we can compute $x_1, x_2$ from the intersecting point first. The area of $\frac{1}{2}b_1x_1^2$ and $\frac{1}{2}b_2x_2^2$ is larger than $l_1$, while the area of the lower rectangle $l_2$ is $PS_b^l$.

  Comparing these areas, we can easily see

$$PS_l = PS_a^l + PS_b^l > l_1 + l_2 > h_1 + h_2 = PS_a^h + PS_b^h = PS_h.$$

  For consumer surplus, from the fact that the flow to the first and second market is higher with low price strategy and the market prices are inverse linear, the total market surplus is higher.

  Social welfare is the sum of total firm surplus and total consumer surplus. This is a direct result by the fact that low price strategy always produces a higher total surplus of firms and consumers. ∎

## 2.9 Proofs in Section 2.2

### 2.9.1 Proof of Lemma 2.2.1

**Proof** 1. For the incentive of agent 2 to put personal effort. Note that whether signed the subcontract or not, when $s_1$ is observed, the subcontract cost is a sink cost, and will **not** influence agent 2's decision on personal effort. Therefore, by Equation 2.7, the minimum payment for agent 2 to make effort in any condition is

$$M_2 \geq M_2^+.$$

2. For the incentive to sign subcontract, under the condition that agent 2 always makes effort ($M_2 \geq M_2^+$). Incentive for signing subcontract with agent 1, compare 2.15 with 2.14,

$$\pi_2(M_1 = M_1^0 | e_2 = 1) - \pi_2(M_1 = 0 | e_2 = 1) = \alpha_1(\beta_2 + \tau_2)M_2 - P(1)M_1^0.$$

And the minimum payment to satisfy $\pi_2(e_2 = 1, M_1 = M_1^0) \geq \pi_2(e_2 = 1, M_1 = 0)$ is

$$M_2 \geq \frac{P(1)M_1^0}{\alpha_1(\beta_2 + \tau_2)}.$$

■

### 2.9.2 Proof of Theorem 2.2.1

**Proof** Recall the cost of delegation 2.16, the condition for the principal to prefer delegation is

$$\begin{aligned} cost_1 - P(2)M_2^+ &\geq 0 \\ cost_1 - \frac{P(2)P(1)M_1^0}{\alpha_1(\beta_2 + \tau_2)} &\geq 0 \end{aligned}$$

Note the first inequality always holds, because $cost_1 - P(2)M_2^+ = P(1)M_1^0 \geq 0$. For the second inequality

$$\frac{P(2)M_2^+}{P(1)M_1^0} \geq \frac{P(s_2 = 1 | e_1 = 0, e_2 = 1)}{\alpha_1(\beta_2 + \tau_2)}.$$

and it's equivalent to,

$$\frac{(\alpha_2 + (\alpha_1 + \gamma_1)(\beta_2 + \tau_2) + \gamma_2)c_2/\alpha_2}{(\alpha_1 + \gamma_1)c_1/\alpha_1} \geq \frac{\alpha_2 + \gamma_1(\beta_2 + \tau_2) + \gamma_2}{\alpha_1(\beta_2 + \tau_2)}.$$

■

### 2.9.3 Proof of Theorem 2.2.2

**Proof** If delegation is better than direct control with both signals, then it must satisfy $cost_2 \geq cost_3$, which is equivalent to

$$\begin{aligned} cost_2 - P(2)M_2^+ &\geq 0 \\ cost_2 - \frac{P(2)P(1)M_1^0}{\alpha_1(\beta_2+\tau_2)} &\geq 0 \end{aligned}$$

From the first inequality,

$$\begin{aligned} cost_2 - cost_3 &= P(1)M_1 + (1-P(1))P(s_2=1|s_1=0,e_2=1)M_2^+ \\ &\quad + P(1)P(s_2=1|s_1=1,e_2=1)M_2^- - P(2)M_2^+ \\ &= P(1)M_1 + P(2)M_2^+ + P(1)P(s_2=1|s_1=1,e_2=1)(M_2^- - M_2^+) - P(2)M_2^+ \\ &= P(1)M_1 + P(1)P(s_2=1|s_1=1,e_2=1)(M_2^- - M_2^+). \end{aligned}$$

Therefore, delegation is better, $cost_2 - cost_3 \geq 0$, if

$$\frac{c_1}{\alpha_1} + (\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2+\tau_2} \geq (\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2}.$$

Note that when this threshold is binding, we have $M_2 = M_2^+ \geq \frac{P(1)M_1^0}{\alpha_1(\beta_2+\tau_2)}$, and

$$\frac{c_2}{\alpha_2} \geq \frac{(\alpha_1+\gamma_1)c_1}{\alpha_1^2(\beta_2+\tau_2)}.$$

Equivalent to

$$\frac{c_1}{\alpha_1(\alpha_2+\beta_2+\tau_2+\gamma_2)} \geq \frac{\tau_2 c_2}{\alpha_2(\alpha_2+\tau_2)}.$$

From the second inequality,

$$\begin{aligned} cost_2 - cost_3 &= P(1)M_1 + (1-P(1))P(s_2=1|s_1=0,e_2=1)M_2^+ \\ &\quad + P(1)P(s_2=1|s_1=1,e_2=1)M_2^- - P(2)\frac{P(1)M_1^0}{\alpha_1(\beta_2+\tau_2)}. \end{aligned}$$

Equivalent to

$$(1-\alpha_1-\gamma_1)(\alpha_2+\gamma_2)\frac{c_2}{\alpha_2} + (\alpha_1+\gamma_1)(\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2+\tau_2} \geq \frac{(\alpha_1+\gamma_1)(\alpha_2+\gamma_2)}{\alpha_1(\beta_2+\tau_2)}\frac{c_1}{\alpha_1}.$$

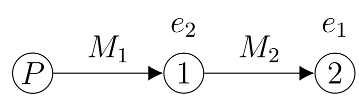
Note that when the above threshold is binding, we have $M_2 = \frac{P(1)M_1^0}{\alpha_1(\beta_2+\tau_2)} \geq M_2^+$, and

$$\frac{c_2}{\alpha_2} \leq \frac{(\alpha_1+\gamma_1)c_1}{\alpha_1^2(\beta_2+\tau_2)}.$$

■

### 2.9.4 Proof of Proposition 2.2.3

**Proof** Consider the principal delegates agent 2 to agent 1 as in the following Fig 2.23, in this case, the principal can't observe $s_2$ because of the delegation.




Delegation

Fig. 2.23.: Two Delegation Structures

If the contract is based on a single signal, then the principal's payment to agent 1 is irrelevant to 2, and there is no incentive for agent 1 to sign any subcontract with agent 2.

If the contract uses both signals, and agent 1's contract payment is in the following form,

$$r_1 = \begin{cases} M_1(1,1), & \text{if } s_1 = 1, s_2 = 1; \\ M_1(1,0), & \text{if } s_1 = 1, s_2 = 0; \\ M_1(0,1), & \text{if } s_1 = 0, s_2 = 1; \\ 0, & \text{if } s_1 = 0, s_2 = 0. \end{cases}$$

While payment to agent 2 is

$$r_2 = \begin{cases} M_2^-, & \text{if } s_1 = 1, s_2 = 1; \\ M_2^+, & \text{if } s_1 = 0, s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

Utility of agent 1 with effort and sufficient subcontract is,

$$\begin{aligned}\pi_1(e_1=1,M_2|M_1) = &P(s_1=1,s_2=1|e_1=1,e_2=1)(M_1(1,1)-M_2^-)\\&+P(s_1=0,s_2=1|e_1=1,e_2=1)(M_1(0,1)-M_2^+)\\&+P(s_1=1,s_2=0||e_1=1,e_2=1)M_1(1,0)-c_1e_1\end{aligned}$$

Utility of agent 1 with zero effort and no subcontract is,

$$\begin{aligned}\pi_1(e_1=0,M_2=0|M_1) = &P(s_1=1,s_2=1|e_1=0,e_2=0)M_1(1,1)\\&+P(s_1=0,s_2=1|e_1=0,e_2=0)M_1(0,1)\\&+P(s_1=1,s_2=0||e_1=0,e_2=0)M_1(1,0)\end{aligned}$$

By the condition that $\pi_1(e_1=1,M_2|M_1)\geq\pi_1(e_1=0,M_2=0|M_1)$, we have

$$\begin{aligned}cost_4 &\geq cost_2-P(1)M_1^0+c_1e_1+\pi_1(e_1=0,M_2=0|M_1)\\&\geq cost_2-\gamma_1\frac{c_1}{\alpha_1}+\gamma_1(\alpha_2+\beta_2+\tau_2+\gamma_2)M_1(1,0)\\&\quad+\gamma_1(1-\alpha_2-\beta_2-\tau_2-\gamma_2)M_1(1,1)\end{aligned}\tag{2.52}$$

where $cost_4$ is the expected cost for this reverse delegation,

$$\begin{aligned}cost_4 = &P(s_1=1,s_2=1|e_1=1,e_2=1)M_1(1,1)\\&+P(s_1=0,s_2=1|e_1=1,e_2=1)M_1(0,1)\\&+P(s_1=1,s_2=0||e_1=1,e_2=1)M_1(1,0),\end{aligned}$$

and $cost_2$ is the expected cost for direct control with both signals 2.12,

$$\begin{aligned}cost_2 = &P(s_1=1,s_2=1|e_1=1,e_2=1)M_2^-\\&+P(s_1=0,s_2=1|e_1=1,e_2=1)M_2^++P(1)M_1^0.\end{aligned}$$

Therefore, if we prove $cost_4\geq cost_2$, then reverse delegation is dominated by direct control with both signals and it's always inefficient. By Equation 2.52, the sufficient condition is to show

$$(\alpha_2+\beta_2+\tau_2+\gamma_2)M_1(1,1)+(1-\alpha_2-\beta_2-\tau_2-\gamma_2)M_1(1,0)\geq\frac{c_1}{\alpha_1}\tag{2.53}$$

To prove the above inequality does hold, we use the condition that $\pi_1(e_1 = 1, M_2|M_1) \geq \pi_1(e_1 = 0, M_2|M_1)$, which gives

$$\begin{aligned} c_1 \leq\ & \alpha_1(\alpha_2 + \beta_2 + \tau_2 + \gamma_2)(M_1(1,1) - M_2^-) \\ & + \alpha_1(1 - \alpha_2 - \beta_2 - \tau_2 - \gamma_2)M_1(1,0) \\ & - \alpha_1(\alpha_2 + \gamma_2)(M_1(0,1) - M_2^+) \\ \leq\ & \alpha_1(\alpha_2 + \beta_2 + \tau_2 + \gamma_2)M_1(1,1) \\ & + \alpha_1(1 - \alpha_2 - \beta_2 - \tau_2 - \gamma_2)M_1(1,0) \end{aligned} \tag{2.54}$$

where the second inequality used the condition that $M_1(0,1) - M_2^+ \geq 0$ (otherwise, agent 1 won't be incentive to sign contract with agent 2 when $s_1 = 0$).

In summary, inequality 2.54 implies inequality 2.53. Combine inequality 2.52 and inequality 2.53, we eventually proves $cost_4 \geq cost_2$, and shows delegate agent 2 to agent 1 is dominated by direct control with both signals. ■

## 2.10 Proofs in Section 2.3

### 2.10.1 Proof of Theorem 2.3.2

**Proof** If $k$ direct control agent $k_0$, then

$$\begin{aligned} M_k &\geq \frac{p_{k_0} M_{k_0}(k_0) + \Delta}{\delta_{k_0}^k} \\ &\geq \frac{p_{k_0} M_{k_0}(k_0)}{\alpha_{k_0} \prod_{i=k_0+1}^k (\beta_i + \tau_i)} \\ &\geq \frac{M_{k_0}(k_0)}{\prod_{i=k_0+1}^k (\beta_i + \tau_i)}. \end{aligned}$$

If agent $k_0$ is not direct controlled by $k$, and denote the agent direct control him as $h_1$, then similar to the above derivation,

$$M_{h_1} \geq \frac{M_{k_0}(k_0)}{\prod_{i=k_0+1}^{h_1} (\beta_i + \tau_i)}.$$

Suppose $h_1$ is directly controlled by $h_2$, utility of $h_2$ with full incentive is

$$\pi_{h_2} = p_{h_2} M_{h_2} - p_{h_1} M_{h_1} - \sum_{i \in T(h_2) \setminus h_1} p_i M_i - c_{h_2}.$$

Now consider the case that $h_2$ choose to shirk the whole branch from $h_1$, the new utility function is

$$\pi'_{h_2} = (p_{h_2} - \delta_{h_1}^{h_2})M_{h_2} - \sum_{i \in T(h_2)\setminus h_1} (p_i - \delta_{h_1}^{i})M_i - c_{h_2}.$$

Note that if $i < h_1$, $\delta_{h_1}^{i} = 0$.

By $\pi_{h_2} \geq \pi'_{h_2}$, we have a lower bound for $M_{h_2}$ as

$$\begin{aligned} M_{h_2} &\geq \frac{p_{h_1}M_{h_1} + \sum_i \delta_{h_1}^{i} M_i}{\delta_{h_1}^{h_2}} \\ &\geq \frac{p_{h_1}M_{h_1}}{\delta_{h_1}^{h_2}} \\ &\geq \frac{p_{h_1}M_{h_1}}{p_{h_1}\prod_{h_1}^{h_2}(\beta_i + \tau_i)} \\ &\geq \frac{M_{h_1}}{\prod_{h_1}^{h_2}(\beta_i + \tau_i)} \\ &\geq \frac{M_{k_0}(k_0)}{\prod_{i=k_0+1}^{h_2}(\beta_i + \tau_i)}. \end{aligned}$$

This can be repeated until the parent node is $k$, thus we have the lower bound of $k$ to motivate $i$ in any delegation,

$$M_k \geq \frac{M_{k_0}(k_0)}{\prod_{i=k_0+1}^{k}(\beta_i + \tau_i)}.$$

■

### 2.10.2 Proof of Lemma 2.3.1

**Proof** Denote $DS(d)$ as the size of possible delegation structures with control set size $d$. The goal is to prove

$$DS(d) = O(2^{d^2}).$$

For agent $k$, given control set $\theta_k^{k-d}$. There are $2^d$ ways to choose it's direct control set of agents $T(k)$, denote the agents controlled by $i \in T(k)$ as $d_i$. Then the number of possible delegation under $i$ is $DS(d_i)$.

$$DS(d) \leq 2^d \sum_{i \in T(k)} DS(d_i), \tag{2.55}$$

where $\sum_i d_i = d - 1$.

Proof by induction. Assume $DS(d_i) = O(2^{d_i^2})$, by Equation 2.55,

$$\begin{aligned} DS(d) &\leq 2^d \sum_{i \in T(k)} O(2^{d_i^2}) \\ &\leq 2^d O(2^{(d-1)^2}) \\ &\leq O(2^{d^2}). \end{aligned}$$

■

### 2.10.3 Proof of Lemma 2.3.2

**Proof** Suppose the control set for agent $k$ is $\theta_k^i$, where $i$ is the last agent under $k$'s control and satisfies $k - i \leq d$ by Assumption 2.3.3.

By induction, for any descendants $h$ of $k$, assume the size of effort status is bounded as follows,

$$|\Psi_h| \leq 3^{DS(h)}.$$

For his descendants, there are at most $\prod_{h \in T(k)} 3^{DS(h)} \leq 3^d$ effort status. Therefore, agent $k$ only need to enumerate $3^d$ to find the optimal subcontract decision. Meanwhile, the function of **optimal benefit from subcontracts** $\pi_k^s(M_k|\eta_k)$ has at most $3^d$ pieces (Fig. 2.24),

$$\pi_k^s(M_k|\eta_k) = \max_{\psi_h \in \Psi_h(\theta_h^i)} \sum_h \pi_k^s(\psi_h|M_k, \eta_k).$$

For agent $k$, the utility function at the contract stage (Eq. 2.36) is

$$\pi_k(M_{\vec{h}}|M_k, \eta_k) = \pi_k^p(\psi_{i,k}|M_k) + \sum_{h \in T(k)} \pi_{k,h}^s(\psi_{i,k}|M_k) + (\beta_{i,k} P(i-1) + \gamma_k) M_k.$$

STEP 1: build one-to-one mapping

Overall, the optimal utility function of $k$ can be decomposed as

$$\begin{aligned} \pi_k(M_k|\eta_k) &= \max_{\psi_k} \pi_k(\psi_{i,k}|M_k) \\ &= \max_{\psi_k} (\pi_k^p(\psi_{i,k}|M_k) + \pi_k^s(\psi_{i,k}|M_k)) + \gamma_k M_k. \end{aligned}$$

Because the personal effort benefit of $k$ in Equation 2.37 is continuous and concavity of function $\pi_k^s(\psi_{i,k}|M_k)$, for two effort status $\psi_h$ and $\psi_h'$, if

$$\pi_k(\psi_h|M_k, \eta_k) \geq \pi_k(\psi_h'|M_k, \eta_k), \text{interval } i.$$

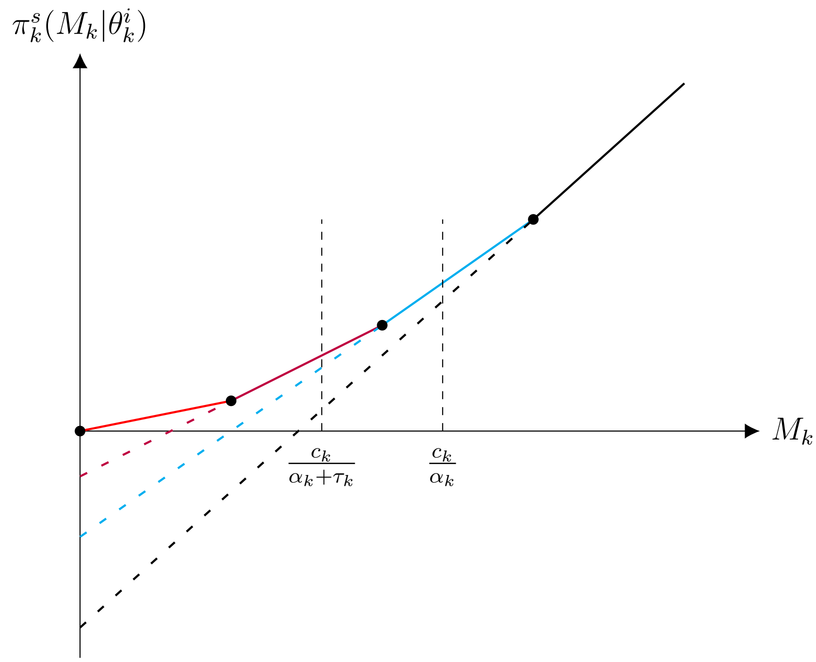



Fig. 2.24.: Expected Subcontract Utility of Agent $k$ at Contract Stage

Then the relation still holds in the next interval,

$$\pi_k(\psi_h|M_k,\eta_k) \geq \pi_k(\psi'_h|M_k,\eta_k), \text{interval } i.$$

Therefore, the utility function of $k$ is a convex piecewise linear function (Fig. 2.25) with pieces bounded by

$$|\Psi_{i,k}| \leq 3 + \prod_{h \in T(h)} 3^{DS(h)} \leq 3 + 3^{d-1} \leq 3^d.$$

The intersection point over $\pi_k(M_k|\eta_k)$ gives the one-to-one mapping between the minimum contract payment and effort status $\psi_{i,k}$.

STEP 2: the last intersection point is the minimum payment for the full effort.

■

### 2.10.4 Proof of Lemma 2.3.3

**Proof** Proof by induction. Easy to check $\Psi_i(\theta_i^i) = \{\emptyset, \{i\}\}$ are monotone inclusion with the increasing $M_i$. Now assume, effort status in $\Psi_{k-1}(\theta_{k-1}^i)$ are monotone inclusion with the increasing $M_{k-1}$.

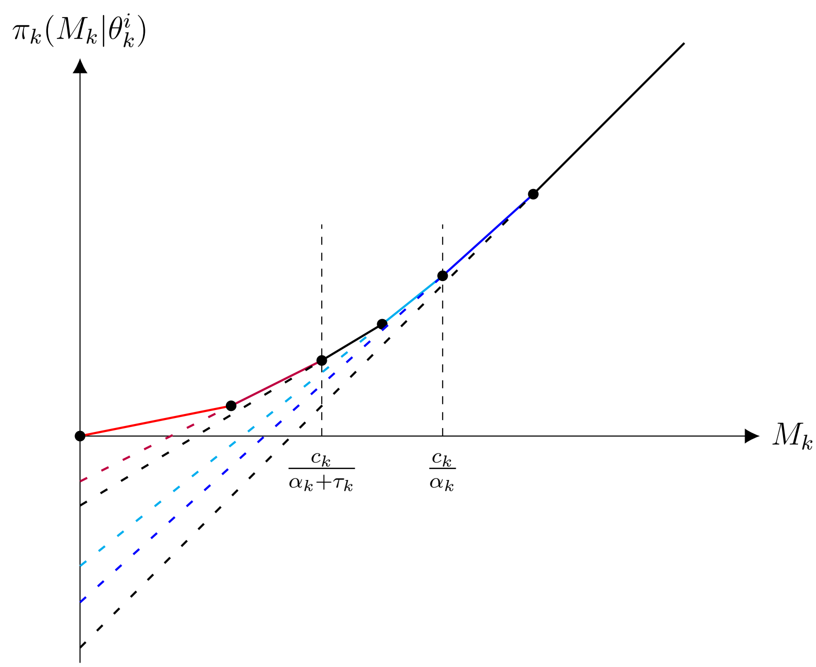



Fig. 2.25.: Expected Utility of Agent $k$ at Contract Stage

For agent $k$, given any $M_k \leq M'_k$, it's equivalent to prove

$$\psi'_k = \psi_k(M'_k|\theta_k^i) \subseteq \psi_k(M_k|\theta_k^i) = \psi_k,$$

under any control set $\theta_k^i$.

First we know

$$\psi_k(M_k|\theta_k^i) = \psi_{k-1}(M_k|\theta_k^i) + I_k(M_k),$$
$$\psi_k(M'_k|\theta_k^i) = \psi_{k-1}(M'_k|\theta_k^i) + I_k(M'_k).$$

where $\psi_{k-1}(M_k|\theta_k^i)$ is the effort status in function $\pi_k^s(M_k|\theta_k^i)$. For simplicity, denote

$$\psi_{k-1} = \psi_{k-1}(M_k|\theta_k^i),$$
$$\psi'_{k-1} = \psi_{k-1}(M'_k|\theta_k^i).$$

Since $M'_k \leq M_k$, we have $I_k(M'_k) \subseteq I_k(M_k)$. Hence, it's sufficient to prove the theorem if we can prove

$$\psi'_{k-1} \subseteq \psi_{k-1}.$$

Because $\pi_k^s(M_k|\theta_k^i)$ is convex, with $M_k' \leq M_k$, we know

$$p_k(\psi_{k-1}') < p_k(\psi_{k-1}),$$

which infers

$$p_{k-1}(\psi_{k-1}') < p_{k-1}(\psi_{k-1}).$$

Now because $\pi_{k-1}(M_{k-1}|\theta_{k-1}^i)$ is also a convex function, we know

$$M_{k-1}(\psi_{k-1}') < M_{k-1}(\psi_{k-1}).$$

By induction assumption that $\Psi_{k-1}(\theta_{k-1}^i)$ are monotone inclusion with the increasing $M_{k-1}$, we have

$$\psi_{k-1}' \subseteq \psi_{k-1}.$$

■

### 2.10.5 Proof of Lemma 2.13.2

**Proof** When control set is $\theta_i^i$, agent $i$ only has two effort status $\{i\}$ or $\emptyset$.

By induction, let's assume it's true for agent $k-1$ with $\theta_{k-1}^i$, that exists an mapping function

$$M = M_{k-1}(\psi_{k-1}|\theta_k^i), \text{where } \psi_{k-1} \in \Psi_{k-1}(\theta_k^i).$$

such that $M$ is the minimum contract payment to make agents in $\psi_{k-1}$ be incentive. And the size of effort statuss of $k-1$, $\Psi_{k-1}(\theta_k^i)$, is bounded by $k-i+1$.

For agent $k$, his expected utility function is the maximum among different decisions over $\psi_{k-1}$ ($M_{k-1}$) and $e_k$, which can be written as

$$\pi_k(e_k, \psi_{k-1}|M_k, \theta_k^i) = (\alpha_k e_k + \beta_k p_{k-1}(\psi_{k-1}|\theta_{k-1}^i) + \gamma_k)M_k - p_{k-1}(\psi_{k-1}|\theta_{k-1}^i)M_{k-1}(\psi_{k-1}) - c_k e_k.$$

where $p_{k-1}$ can be computed by Equation 2.31. The above utility can be divided into three parts,

$$\pi_k(e_k, \psi_{k-1}|M_k, \theta_k) = \pi_k^p(e_k|M_k) + \pi_k^s(\psi_{k-1}|M_k, \theta_k) + \gamma_k M_k.$$

where $\pi_k^p(e_k|M_k)$ is **benefit from making effort**,

$$\pi_k^p(e_k|M_k) = \alpha_k M_k e_k - c_k e_k, \tag{2.56}$$

and $\pi_k^s(M_{k-1}|M_k,\theta_k)$ is **benefit from subcontract**,

$$\pi_k^s(\psi_{k-1}|M_k,\theta_k) = \beta_k p_{k-1}(\psi_{k-1}|\theta_{k-1}^i)M_k - p_{k-1}(\psi_{k-1}|\theta_{k-1}^i)M_{k-1}(\psi_{k-1}). \quad (2.57)$$

1. Decision over $e_k$ is independent to $\psi_{k-1}$, and can be computed by Theorem 2.3.1,

- $k \in \psi_k(M_k,\theta_k^i)$, if $M_k \geq M_k(k)$;
- $k \notin \psi_k(M_k,\theta_k^i)$, if $M_k < M_k(k)$.

and benefit from making effort, $\pi_k^p(M_k) = \max_{e_k} \pi_k^p(e_k|M_k)$, is a piecewise linear function with 2 pieces similar to Fig. 2.8.

2. Denote $\pi_k^s(M_k|\theta_k)$ as the benefit function from subcontract,

$$\pi_k^s(M_k|\theta_k) = \max_{\psi_{k-1}\in\Psi_{k-1}(\theta_{k-1}^i)} \pi_k^s(\psi_{k-1}|M_k,\theta_k),$$

which is the maximum of $|\Psi_{k-1}(\theta_{k-1}^i)| = k-i+1$ linear functions. So the benefit of $k$ from subcontract is a convex piecewise linear function with at most $k-1-i$ pieces (Fig. 2.26).

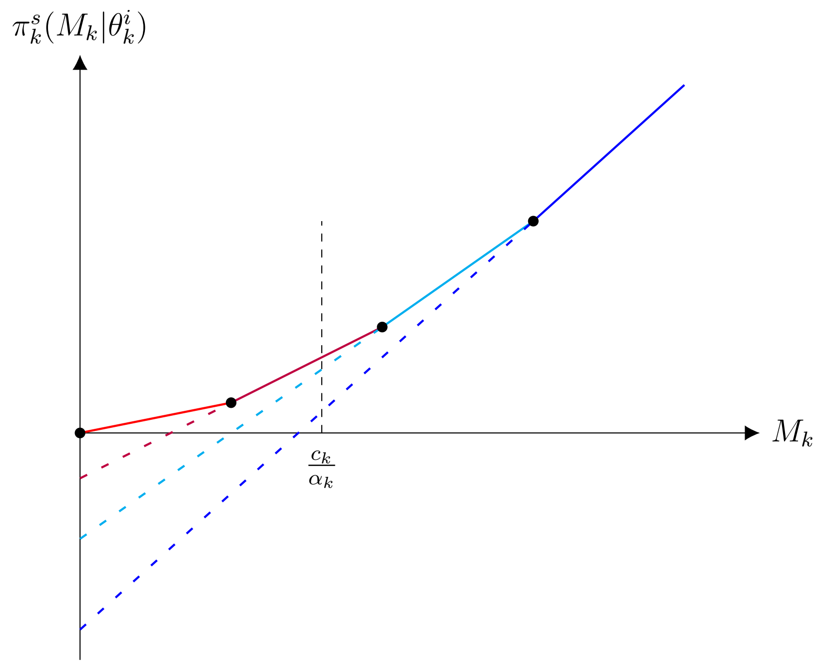



Fig. 2.26.: Subcontract Utility Function of Agent $k$

Overall, the optimal utility function of $k$ can be decomposed as

$$\begin{aligned}\pi_k(M_k|\theta_k) &= \max_{e_k,\psi_{k-1}} \pi_k(e_k,\psi_{k-1}|M_k,\theta_k) \\ &= \max_{e_k} \pi_k^p(e_k|M_k) + \max_{\psi_{k-1}} \pi_k^s(\psi_{k-1}|M_k,\theta_k) + \gamma_k M_k.\end{aligned}$$

By the personal effort condition of $k$, we can rewrite it as,

$$\pi_k(M_k,\theta_k) = \begin{cases} \pi_k^s(M_k|\theta_k) + \gamma_k M_k, & M_k \leq \frac{c_k}{\alpha_k}, \\ \alpha_k M_k - c_k + \pi_k^s(M_k|\theta_k) + \gamma_k M_k, & M_k \geq \frac{c_k}{\alpha_k}. \end{cases} \tag{2.58}$$

Therefore, the utility function of $k$ is a convex piecewise linear function with at most $k - i + 2 = |\Psi_k(\theta_k^i)|$ pieces (Fig. 2.27).

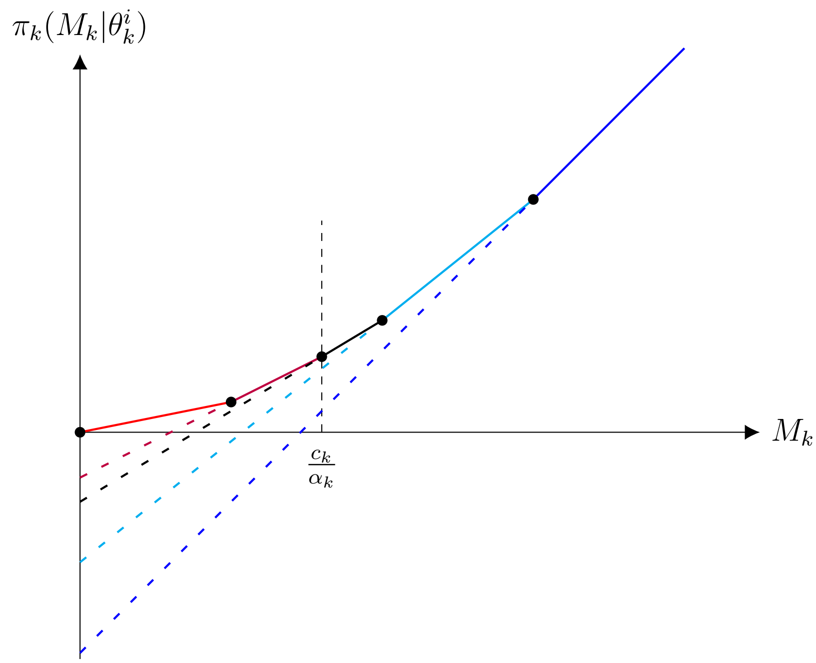



Fig. 2.27.: Utility Function of Agent $k$

To show the one-to-one mapping between $M_k$ and $\psi_k$, we first find the mapping between $M_k$ and $\psi_{k-1}$. Consider the piecewise linear function $\pi_k^s(M_k|\theta_k^i)$, each piece is corresponding to an effort status $\psi_{k-1}$ that maximized the subcontract utility of $k$ at the given $M_k$. Therefore, we can build an **one-to-one mapping** between $M_k$ and effort status $\psi_{k-1}$,

$$M_k^1(\psi_{k-1}|\theta_k^i) = \text{argmin}_{M_k} 0$$
$$\text{s.t. } \pi_k^s(M_k|\theta_k^i) = \pi_k^s(\psi_{k-1}|M_k, \theta_k^i).$$

Note that if the payment is lower than $M_k^1(\psi_{k-1}|\theta_k^i)$, agent $k$ will prefer another effort status $\psi_{k-1}$. While a higher payment only brings an unnecessary cost.

Denote the inverse function of $M_k^1(\cdot|\theta_k^i)$ as $\psi_{k-1}(\cdot|\theta_k^i)$. Given $M_k$, the corresponding effort status $\psi_k$ is

$$\psi_k(M_k|\theta_k^i) = \psi_{k-1}(M_k|\theta_k^i) + I_k(M_k),$$

recall that

$$I_k(M_k) = \begin{cases} \emptyset, & M_k < M_k(k), \\ \{k\}, & M_k \geq M_k(k). \end{cases}$$

Denote all the different $\psi_k$ from $M_k \geq 0$ as the set $\Psi_k(\theta_k^i)$, again easy to see any effort combination other than these is impossible to happen. This is still an one-to-one mapping, and denote minimum contract payment for effort status $\psi_k$, $M_k(\cdot|\theta_k^i)$, as an inverse function of $\psi_k(\cdot|\theta_k^i)$.

■

### 2.10.6 Proof of Proposition 2.3.3

**Proof** Suppose delegate $h$ agents to $k$. We first prove it's better to direct control $k$, and delegate $h-1$ agents to $k-1$. Which is equivalent to prove

$$p^* M(h) \geq p^* M^* + p^* M(h-1), \tag{2.59}$$

where $M^*$ is the minimum payment for personal effort, and $M(h)$ is the minimum payment to an agent to motivate him and his $h$ sub-agents.

Consider agent $k$'s utility function at the optimal decision

$$\pi_k^* = p^* M(h) - p^* M(h-1) - c_k.$$

Consider utility function under effort only enough for $h-1$ agents,

$$\begin{aligned} \pi_k^s &= (p^* - \alpha\beta^h) M(h) - (p^* - \alpha\beta^{h-1}) M(h-1)' - c \\ &\geq (p^* - \alpha\beta^h) M(h) - (p^* - \alpha\beta^{h-1}) M(h-1) - c. \end{aligned}$$

From $\pi_k^* - \pi_k^s \geq 0$, we know

$$\alpha\beta^h M(h) \geq \alpha\beta^{h-1} M(h-1),$$

which means $M(h) \geq \frac{M(h-1)}{\beta}$, and

$$M(h) \geq \frac{M^*}{\beta^h}.$$

Now for the utility function with single personal effort

$$\pi_k^p = (p^* - \sum_{i=1}^{h} \alpha\beta^i)M(h) - c.$$

From $\pi_k^* - \pi_k^p \geq 0$, we know

$$\begin{aligned} p^*M(h) - p^*M(h-1) &\geq (p^* - \sum_{i=1}^{h} \alpha\beta^i)M(h) \\ &\geq (p^* - \sum_{i=1}^{h} \alpha\beta^i)\frac{M^*}{\beta^h}. \end{aligned}$$

To prove Equation 2.59, it'll be sufficient if proved

$$\frac{(p^* - \sum_{i=1}^{h} \alpha\beta^i)}{\beta^h} \geq p^*,$$

which is equivalent to

$$p^*(1 - \beta^h) \geq \sum_{i=1}^{h} \alpha\beta^i, \tag{2.60}$$

and it's true because $p^* \geq \alpha\beta$ and $1 - \beta^h = \sum_{i=0}^{h-1} \beta^i$.

Similarly we can prove

$$p^*M(h-1) \geq p^*M^* + p^*M(h-2),$$

and eventually we have

$$p^*M(h) \geq np^*M^*.$$

Therefore, the principal always prefer direct control than delegation when $k$ is far from the initial agent.

$\blacksquare$

### 2.10.7 Proof of Proposition 2.3.4

**Proof** Denote $M^*$ as the contract payment for personal effort in symmetric case with normal cost. By Corollary 2.3.1, we have a lower bound for delegation payment,

$$M_k \geq \frac{M^*}{(\beta+\tau)^d}.$$

Meanwhile, the cost of direct control is

$$cost_p^d = (t+d)p^* M^*.$$

The necessary condition for delegation is better is

$$\frac{M^*}{(\beta+\tau)^d} \leq (t+d)M^*.$$

A necessary condition for above to hold is

$$\frac{M^*}{(\beta+\tau)^d} \leq tM^*,$$

which gives

$$d \leq \frac{\log t}{\log(\beta+\tau)^{-1}}.$$

■

## 2.11 Proofs in Section 2.4

### 2.11.1 Proof of Lemma 2.4.1

**Proof** Now consider any delegation structure other than $\Theta(f)$, $\theta'_f \notin \Theta(f)$. First look at the most inefficient agent, $i = \max \theta'_f \cap C(k)$, note that $i$ is the leaf agent with the highest $M_k(i)$. Now we are going to prove $\theta^i_f$ dominates $\theta'_f$.

Subcontract benefit of $f$ at contract stage when fully motivated under $\theta^i_f$,

$$\begin{aligned}\pi^s_f(\theta^i_f|M_f,\theta^i_f) &= p_f(\theta^i_f|\theta^i_f)M_f - p_k(\theta^i_k|\theta^i_f)M_k(\theta^i_k|\theta^i_k)\\ &= P(f)M_f - P(k)M_k(i).\end{aligned}$$

Note that

$$\pi^s_f(\theta'_f|M_f,\theta'_f) = \pi^s_f(\theta^i_f|M_f,\theta^i_f).$$

Denote the next effort set of $\theta_f^i$ as $\psi_f^j$, where $j = \max \psi_f^j$. The utility with effort set $\psi_f^j$ is

$$\pi_f^s(\psi_f^j|M_f, \theta_f^i) = p_f(\psi_f^j|\theta_f^i)M_f - p_k(\psi_k^j|\theta_f^i)M_k(j).$$

Denote the next effort set of $\theta'_f$ as $\psi_f^h$, where $h = \max \psi_f^h$. The utility with effort set $\psi_f^h$ is

$$\pi_f^s(\psi_f^h|M_f, \theta'_f) = p_f(\psi_f^h|\theta'_f)M_f - p_k(\psi_k^h|\theta'_f)M_k(h).$$

By comparing $\pi_f^s(\theta'_f|M_f, \theta'_f) \geq \pi_f^s(\psi_f^h|M_f, \theta'_f)$, we have the minimum payment $M_f(\theta'_f)$ for control set $\theta'_f$

$$\begin{aligned} M_f(\theta'_f) &= \frac{P(k)M_k(i) - p_k(\psi_k^h|\theta'_f)M_k(h)}{\delta_f^h - \delta_f^*} \\ &\geq \frac{P(k)M_k(i) - p_k(\psi_k^j|\theta'_f)M_k(j')}{\delta_f^j - \delta_f^*} \\ &\geq \frac{P(k)M_k(i) - p_k(\psi_k^j|\theta'_f)M_k(j)}{\delta_f^j - \delta_f^*}, \end{aligned}$$

where $j' \in \theta'_f$ is the largest numbered leaf agent which satisfies $j' \leq j$, and we have $M_k(j') \leq M_k(j)$.

The agent's direct controlled by the principal will influence $k$'s probability of success too.

$$p_k(\psi_k^j|\theta'_f) = p_k(\psi_k^j|\theta_f^i) + \delta_k^*,$$

where $\delta_k^* = \delta_f^*/\beta_f$, and we have,

$$M_f(\theta'_f) \geq \frac{P(k)M_k(i) - (p_k(\psi_k^j|\theta_f^i) + \delta_k^*)M_k(j)}{\delta_f^j - \delta_f^*}.$$

On the other hand, by $\pi_f^s(\theta_f^i|M_f, \theta_f^i) \geq \pi_f^s(\psi_f^j|M_f, \theta_f^i)$ we have

$$M_k(\theta_k^i) = \frac{P(k)M_k(i) - p_k(\psi_k^j|\theta_f^i)M_k(j)}{\delta_f^j}.$$

To prove $M_f(\theta'_f) \geq M_f(\theta_f^i)$, it will be sufficient if prove

$$\frac{P(k)M_k(i) - p_k(\psi_k^j|\theta_f^i)M_k(j)}{\delta_f^j} \leq \frac{P(k)M_k(i) - (p_k(\psi_k^j|\theta_f^i) + \delta_k^*)M_k(j)}{\delta_f^j - \delta_f^*}.$$

Which is equivalent to

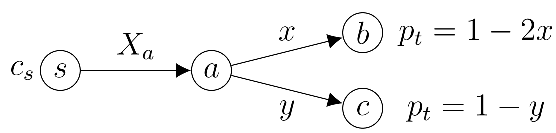
$$
\begin{aligned}
&(\delta_f^j - \delta_f^*)\big(P(k)M_k(i) - p_k(\psi_k^j|\theta_f^i)M_k(j)\big) - \delta_f^j\big(P(k)M_k(i) - (p_k(\psi_k^j|\theta_f^i) + \delta_k^*)M_k(j)\big)\\
=&-\delta_f^* P(k)M_k(i) + \delta_f^* p_k(\psi_k^j|\theta_f^i)M_k(j) + \delta_f^j\delta_k^* M_k(j)\\
\le&\delta_f^* M_k(i)(-P(k) + p_k(\psi_k^j|\theta_f^i) + \delta_f^j/\beta_f^k)\\
=&\delta_f^* M_k(i)(-P(k) + p_k(\psi_k^j|\theta_f^i) + \delta_k^j)\\
=&0.
\end{aligned}
$$

Since $\theta'_f \subset \theta_f^i$, but $M_f(\theta'_f) \geq M_f(\theta_f^i)$, we know $\theta'_f$ is dominated by $\theta_f^i$. Hence, any delegation structure other than $\Theta(f)$ is not efficient. ■

## 2.12 Examples

### 2.12.1 Convex combination

Example 13 (Convex combination in price function computation)




*Aim: show the unique market clearing price function is*

$$
\begin{aligned}
p_a &= 1 - \frac{4}{3}(x+y) = 1 - \frac{4}{3}X_a\\
p_s &= 1 - \frac{8}{3}X_a = 1 - \frac{8}{3}X_s
\end{aligned}
$$

*and* $X_s = X_a$.

*Proof: given* $p_a < 1$, *first we know both* $x, y > 0$, *payoff of* $a$,

$$\Pi_a = (1-2x)x + (1-y)y - p_a(x+y)$$

*Taking the derivative,*

$$
\begin{aligned}
\frac{\partial \Pi_a}{\partial x} &= 1 - 4x - p_a = 0\\
\frac{\partial \Pi_a}{\partial y} &= 1 - 2y - p_a = 0
\end{aligned}
$$

*By convex combination, we know $x+y$ must satisfy,*

$$
\begin{aligned}
p_a &= \frac{1}{3}(1-4x) + \frac{2}{3}(1-2y) \\
&= 1 - \frac{4}{3}(x+y)
\end{aligned}
$$

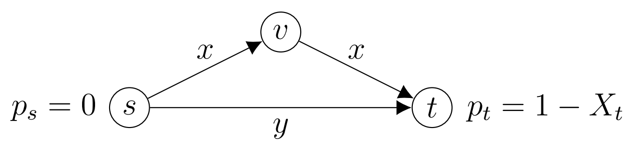
*Also by the assumption, $p_a = f(X_a) = f(x+y)$. Thus,*

$$
p_a = 1 - \frac{4}{3}X_a
$$

*Similarly,*

$$
\begin{aligned}
\frac{\partial \Pi_s}{\partial X_a} = 0 &\Rightarrow p_s = 1 - \frac{8}{3}X_a \\
&\Rightarrow p_s = 1 - \frac{8}{3}X_s
\end{aligned}
$$

EXAMPLE 14 (SPG WITH SHORTCUT) *Consider the following network where path $(s,t)$ is a shortcut of path $(s,v,t)$. Assume no processing and producing cost.*




*The utility of $v$ is:*

$$
\Pi_v = (1-x-y)x - p_v x
$$

*Taking the derivative:*

$$
\frac{\partial \Pi_v}{\partial x} = 1 - 2x - y - p_v = 0 \Rightarrow p_v = 1 - 2x - y
$$

*The utility of $s$ is:*

$$
\begin{aligned}
\Pi_s &= p_v x + p_t y - p_s(x+y) \\
&= (1-2x-y)x + (1-x-y)y
\end{aligned}
$$

*Taking the derivative:*

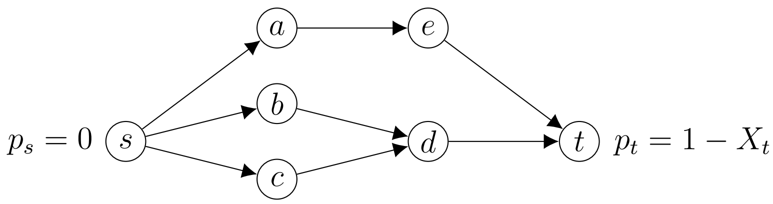
$$
\begin{aligned}
\frac{\partial \Pi_s}{\partial x} &= 1 - 4x - 2y = 0 \\
\frac{\partial \Pi_s}{\partial y} &= 1 - x - 2y = 0
\end{aligned}
$$

*The solution is $x = 0$ and $y = \frac{1}{2}$. $sv$ and $vt$ are inactive.*

### 2.12.2 Price Function Computation

Example 15 (Price Function Computation General Form)

*Assume that each node in the following has no processing or producing cost.*



*Recall the equations in $ALG_1$:*

***SS Case:***

$$b_i = 2b_j + \sum_{k \in C_P(j)} b_k + c_i$$

***MS Case:***

$$b_i = b_j + \sum_{k \in C_P(j)} b_k + c_i$$

***SM Case:***

$$b_i = \frac{2}{\sum_{j \in B(i)} \frac{1}{b_j}} + 2b_h + \sum_{k \in C_P(j) \setminus \{h\}} b_k + c_i \text{ where } h \text{ is the merging node.}$$

*Backward algorithm, MS case ($t$ to its seller $f$, $d$ , and $e$):*

$$p_d = 1 - X_d - X_t$$
$$p_e = 1 - X_e - X_t$$

*MS case ($d$ to its seller $b$ and $c$):*

$$p_b = 1 - 2X_b - X_d - X_t$$
$$p_c = 1 - 2X_c - X_d - X_t$$

*SS case ($f$ to its seller $a$):*

$$p_a = 1 - 3X_a - X_t$$

*To compute the price function at $s$ (SM case), utility function at $s$*

$$\Pi_s = p_a X_a + p_b X_b + p_c X_c - p_s(X_a + X_b + X_c)$$

*Take the derivative with respect to $X_a, X_b$, and $X_c$:*

$$\frac{\partial \Pi_s}{\partial X_a} = 0 \Rightarrow p_{s_a} = 1 - 6X_a - (X_a + X_b + X_c) - X_t = 1 - 6X_a - 2X_t$$
$$\frac{\partial \Pi_s}{\partial X_b} = 0 \Rightarrow p_{s_b} = 1 - 4X_b - 2X_d - (X_a + X_b + X_c) - X_t = 1 - 4X_b - 2X_d - 2X_t$$
$$\frac{\partial \Pi_s}{\partial X_c} = 0 \Rightarrow p_{s_c} = 1 - 4X_c - 2X_d - (X_a + X_b + X_c) - X_t = 1 - 4X_c - 2X_d - 2X_t$$

*Note that $X_t = X_a + X_b + X_c$, so $\frac{\partial X_t}{\partial X_a} = \frac{\partial X_t}{\partial X_b} = \frac{\partial X_t}{\partial X_c} = 1$.*

*For the merging order, note that $C_S(s) = \{d, t\}$, by case 4 in the proof 2.6.2 of Lemma 1.3.1. We start from merging flows with $d$:*

$$\begin{aligned} p_{s_{bc}} &= \frac{1}{2}p_{s_b} + \frac{1}{2}p_{s_c} \\ &= 1 - 2X_{bc} - 2X_d - 2X_t \\ &= 1 - 4X_{bc} - 2X_t \end{aligned}$$

*where $X_{bc}$ is a flow variable considering $b$ and $c$ together.*

*After this, merge flows with $t$:*

$$\begin{aligned} p_{s_{abc}} &= \frac{2}{5}p_{s_a} + \frac{3}{5}p_{s_{bc}} \\ &= 1 - \frac{12}{5}X_{abc} - 2X_t \\ &= 1 - \frac{22}{5}X_s \end{aligned}$$

*Note that since $t \notin C_P(s)$, we substitute $X_t$ by $X_s$.*

*The aforementioned method is based on computing the convex coefficient. The following method applies aggregate variables. First, $C_S(s) = \{d, t\}$ and $d$ is the merging node. Therefore, $b_{bc} = \frac{1}{\frac{1}{b_b} + \frac{1}{b_c}} + b_d = \frac{1}{\frac{1}{4} + \frac{1}{4}} + 2 = 4$. $b_{bc}$ represents the coefficient of nodes $b, c$, and $d$. Next move on to the merging node $t$, this is the simple SM case, so:*

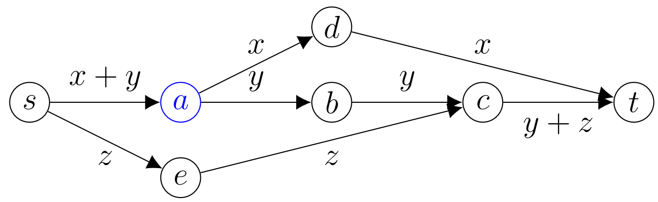
$$b_s = \frac{2}{\frac{1}{b_a} + \frac{1}{b_{bc}}} + 2b_t = \frac{2}{\frac{1}{3} + \frac{1}{4}} + 2 = \frac{22}{5}$$

*Please compare it with the Example 3.*

### 2.12.3 Non-SPG SM

For the parent-child relation, only SS, SM, MS three cases are possible. This example shows the graph restricted to these three relations is not necessary an SPG though.

Example 16 (Non-SPG SM)

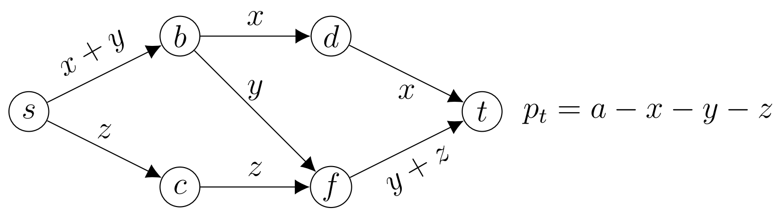



*Note that when computing $p_a$, $C_P(d) = \{t\}$ while $C_P(b) = \{c, t\}$.*

### 2.12.4 Non-SPG MM

The graph is non-SPG, since MM happens at $\{b, c\} \to \{d, f\}$. However, the equilibrium still exists and unique.

Example 17 (Non-SPG MM)




*About parent-merging child nodes, $C_P(b) = C_P(c) = \{f, t\}$.*

$$
\begin{aligned}
p_d &= a - 2x - y - z \\
p_f &= a - x - 2y - 2z \\
p_c &= a - x - 2y - 4z
\end{aligned}
$$

*To compute price function at $b$, utility at $b$ is*

$$
\begin{aligned}
\Pi_b &= p_d x + p_f y - p_b(x + y) \\
&= (a - 2x - y - z)x + (a - x - 2y - 2z)y - p_b(x + y)
\end{aligned}
$$

*Take its derivative with respect to $x$ and $y$:*

$$
\begin{aligned}
\frac{\partial \Pi_b}{\partial x} &= a - 4x - 2y - z - p_b \\
\frac{\partial \Pi_b}{\partial y} &= a - 2x - 4y - 2z - p_b
\end{aligned}
$$

*To write it as a function of inflow $X_b = x + y$,*

$$\begin{aligned} p_b &= 0.5(a - 4x - 2y - z) + 0.5(a - 2x - 4y - 2z) \\ &= a - 3x - 3y - 1.5z \end{aligned}$$

*while $C_P(b) = \{f, t\}$, the above result can't be written as the form unless $b_f = 0$,*

$$\begin{aligned} p_b &= a - b_b X_b - b_f X_f - b_t X_t \\ &= a - b_b(x + y) - b_f(y + z) - b_t(x + y + z) \end{aligned}$$

*Some relations about market clearing price:*

$$\begin{gathered} p_b = a - 4x - 2y - z = a - 2x - 4y - 2z \\ \Rightarrow \\ 2x = 2y + z \end{gathered}$$

*We can rewrite $p_c$ as*

$$\begin{aligned} 4p_c &= 4a - 4x - 8y - 16z \\ 4p_c + 2x &= 4a - 4x - 8y - 16z + 2y + z \\ 4p_c &= 4a - 6x - 6y - 15z \\ p_c &= a - \frac{3}{2}(x + y) - \frac{15}{4}z \end{aligned}$$

*So far, the utility of $s$ can be written as,*

$$\begin{aligned} \Pi_s &= p_b(x + y) + p_c z - p_s(x + y + z) \\ &= (a - \frac{3}{2}X_b - \frac{3}{2}X_s)X_b + (a - \frac{9}{4}X_c - \frac{3}{2}X_s)X_c - p_s X_s \end{aligned}$$

*Similar to the analysis in Section 1.3.3, to maximize the utility, the optimal decision flow of source $s$ is the solution of a system of LCP, and it's equivalent to a convex problem 1.4, which has unique solution.*

*An interesting point about the coefficients that*

$$A \begin{pmatrix} X_b \\ X_c \end{pmatrix} = \begin{pmatrix} a - p_b \\ a - p_c \end{pmatrix}$$

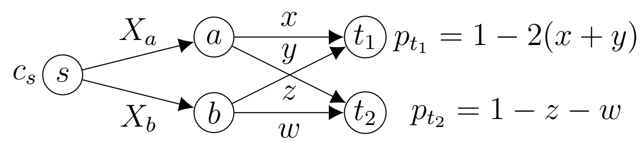
*where*

$$A = \begin{pmatrix} 3 & 3/2 \\ 3/2 & 15/4 \end{pmatrix}$$

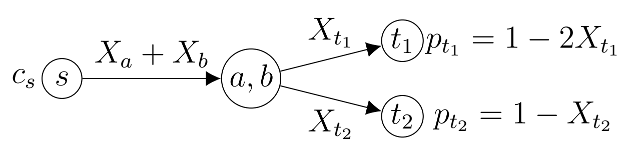
*For the solvable problem, the coefficient matrix $A$ always satisfies*

- *A is positive.*
- *A is invertible.*
- *unique common coefficient (symmetric, eligible to write as a convex problem)*

Example 18 (Non-invertible A)
*An example that A is not invertible,*

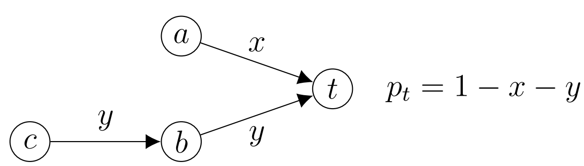



*However, we can imaging this equivalent to*




### 2.12.5 Decision Sequence

Example 19




*Assume raw material cost is 0 at both end. Price functions:*

$$p_a = 1 - 2x - y; p_b = 1 - x - 2y,$$

*and the relation holds at the equilibrium:* $x = \frac{1-y}{2}$. *The total flow is*

$$x + y = \frac{1+y}{2}.$$

*The utility of a is*

$$\Pi_a = p_t x = \frac{(1-y)^2}{4}$$

*1. Suppose c makes decision $p_b, y$ first, then $a, b$ make decision $x, p_t$ based on the belief over each other.*

$$p_b = \frac{1}{2} - \frac{3}{2}y; \quad p_c = \frac{1}{2} - 3y$$

*and the optimal* $y = 1/6$.

*2. Suppose* $c, a$ *makes decision* $p_b, p_t, x, y$ *based on the belief over each other, then* $a$ *make decision (given* $p_b$ *and* $x$*, take* $y$*, accept* $p_t$*).*

$$\begin{aligned} p_b &= 1 - x - 2y \\ p_c &= 1 - x - 4y \\ p_c &= \frac{1}{2} - \frac{7}{2}y \end{aligned}$$

*and the optimal* $y = 1/7$.

*Note that the total flow is higher in the first case, and in the second case, the utility of* $a$ *is higher.*

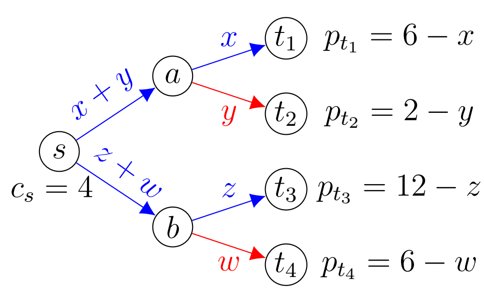
***Summery:*** *in our model*

- *parallel decision is case 2, decide simultaneously based on the* ***belief*** *over each other (assume both act as the unique equilibrium).*
- *multiple branches is case 1, decide each sub-flows by himself, after a decision any combination holds, thus any combination can be treated as the TRUE price function.*

### 2.12.6 Inactive Edges

One of the main difference between MSPG and SPG is the existence of inactive flow; if the inactive edge is mistakenly assumed active, wrong price function will be used for solving the equilibrium.

EXAMPLE 20




*When computing the price function, the challenge is we do not know which edges are active at equilibrium (while in a single source and sink case, we proved every edge is active).*

*Suppose we assume all of them are active,*

$$p_a = 4 - x - y$$

$$p_b = 9 - z - w$$

*Thus, at "equilibrium", s makes decision not selling to a* $(x+y=0)$*, and decision to b is* $p_b = 6.5, x_{sb} = 2.5$*, and edge* $bt_4$ *is inactive* $(w = 0)$*.*

*However, this is not the equilibrium. For a, since a will make a profit by buying items at a higher price than* $p_b$*, and sell them to* $t_1$*, and s will be better off too. For b, by solving the optimal solution at b, we know* $p_b$ *is too low and* $x_{sb}$ *is under demand. Thus s can be better off by raising the price.*

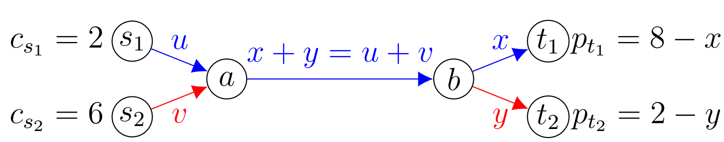
*Actually, at equilibrium edge* $at_2, bt_4$ *are inactive, while* $at_1, bt_3$ *are active. So we should delete edge* $at_2, bt_4$ *before the price computation, and the true income price function at node* $a, b$ *is:*

$$p_a = 6 - 2x$$

$$p_b = 12 - 2z$$

Meanwhile, MSPG may also have inactive flow starts from the source.

Example 21




*Due to the low profit at market* $t_2$*,* $c_{s_1}, c_{s_2} \geq p_{t_2}$*, it's obvious that edge* $bt_2$ *is inactive at equilibrium, so the market clearing price,*

$$p_b = 8 - 2x$$

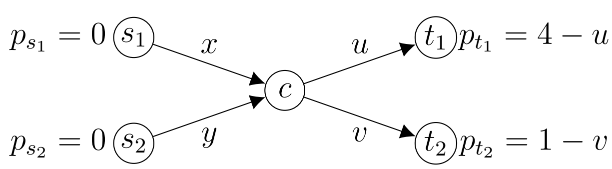
$$p_a = 8 - 4x$$

*Treat a as the market of* $s_1, s_2$*, by solving a standard bipartite Cournot game, we know edge* $s_2a$ *is inactive at equilibrium, while* $s_1a$ *is active.*

*However, we do not need to worry about the inactive edges* $s_2a$ *since it does not influence the price function computation of other branches. In other words, the equilibrium can be solved even though we keep this type of inactive edges in the graph. Notice this is always true by the property of series-parallel graph.*

### 2.12.7 Multiple Equilibria

**Multiple Sources and Multiple Sinks (Computation of Example 10)**



Assume the processing cost is 0. For convenience, denote $p_1 = p_{s_1}$ and $p_2 = p_{s_2}$.

1. High price strategy, since market 2 is inactive, $p_c = 4-2X_c$, and prices function at sources are

$$p_1 = 4 - 4x - 2y = 0,$$
$$p_2 = 4 - 2x - 4y = 0.$$

By solving the above equations, the optimal flows are $x = y = a_1/6 = \frac{2}{3}$, double check the price under the optimal flow:

$$p_c = 4 - \frac{8}{3} = \frac{4}{3} \geq a_2.$$

It is a high price strategy and the payoffs are

$$\Pi_1^h = \Pi_2^h = 2x^2 = \frac{8}{9}.$$

2. Low price strategy, since both markets are inactive, $p_c = \frac{5}{2} - X_c$, and prices function at sources are

$$p_1 = \frac{5}{2} - 2x - y = 0,$$
$$p_2 = \frac{5}{2} - x - 2y = 0.$$

By solving the above equations, the optimal flows are $x = y = \frac{a+1}{6} = \frac{5}{6}$, double check the price under the optimal flow:

$$p_c = \frac{5}{2} - \frac{5}{3} = \frac{5}{6} \leq a_2.$$

It is a low price strategy and the payoffs are

$$\Pi_1^l = \Pi_2^l = x^2 = \frac{25}{36}.$$

Note that the high price strategy gives a higher payoff.

3. High price strategy is an equilibrium.

Recall the optimal flow $x = \frac{2}{3}$ in part 1, let's fix it for firm 1, while consider firm 2 increases $y$ and try low price strategy:

$$p_c = \frac{5}{2} - X_c,$$
$$p_2 = \frac{5}{2} - x - 2y = 0.$$

The new flow is $y = \frac{11}{12}$, double check the price under these flows:

$$p_c = \frac{5}{2} - \frac{3}{2} - \frac{11}{12} = \frac{11}{12} \leqslant a_2$$

It is a low price strategy and the new payoffs for firm 2 is

$$\Pi_2' = y^2 = \frac{121}{144} < \Pi_2^h = \frac{8}{9}$$

Thus, a high price strategy is an equilibrium.

4. Low price strategy is an equilibrium.

Recall the optimal flow $x = \frac{5}{6}$ in part 2, let's fix it for firm 1, while consider firm 2 decreases $y$ and try high price strategy,

$$\begin{aligned} p_c &= 4 - 2X_c, \\ p_2 &= 4 - 2x - 4y = 0. \end{aligned}$$

The new flow is $y = \frac{7}{12}$, double check the price under these flows:

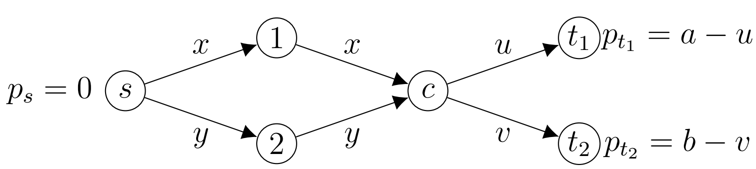
$$p_c = 4 - 2(\frac{5}{6} - \frac{7}{12}) = \frac{7}{6} \geqslant a_2.$$

It is a high price strategy and the new payoffs for firm 2 is

$$\Pi_2' = 2y^2 = \frac{49}{72} < \Pi_2^l = \frac{25}{36}.$$

Thus, a low price strategy is an equilibrium. In summary, both high and low price strategy are equilibria.

**Single Source and Multiple Sources**




1. High price strategy:

$$\begin{aligned} p_c &= a - 2X_c \\ p_1 &= a - 4x - 2y = 0 \\ p_2 &= a - 2x - 4y = 0 \\ p_s &= a - 6x - 6y \Rightarrow X_s = \frac{a}{6} \end{aligned}$$

Utility of $b$ is

$$\Pi_b^h = p_1 X_s = \frac{a}{2}\frac{a}{6} = \frac{a^2}{12}$$

2. Low price strategy:

$$\begin{aligned}
p_c &= \frac{a+b}{2} - X_c \\
p_1 &= \frac{a+b}{2} - 2x - y = 0 \\
p_2 &= \frac{a+b}{2} - x - 2y = 0 \\
p_s &= \frac{a+b}{2} - 3x - 3y \Rightarrow X_s = \frac{a+b}{6}
\end{aligned}$$

Utility of $b$ is

$$\Pi_b^l = p_1 X_s = \frac{a+b}{4}\frac{a+b}{6} = \frac{(a+b)^2}{24}$$

To make low price strategy more preferable:

$$\Pi_b^l > \Pi_b^h \Rightarrow b > (1-\sqrt{2})a$$

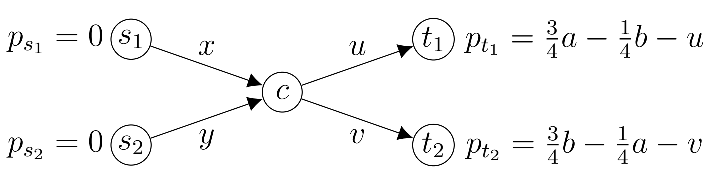
Suppose $b$ chooses low price strategy;

$$\begin{aligned}
p_1 = p_2 &= \frac{a+b}{4} \\
p_c &= \frac{a+b}{3}
\end{aligned}$$

To ensure it is a low price strategy:

$$p_c < a_2 \Rightarrow b > 0.5a$$

Given $p_1$ and $p_2$, for firm 1 and 2's decision, it is equivalent to



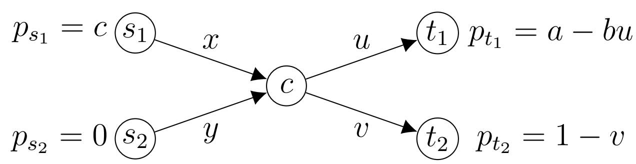

Suppose

$$\frac{3}{4}a - \frac{1}{4}b = 4\left(\frac{3}{4}b - \frac{1}{4}a\right) \Rightarrow b = \frac{7}{13}a$$

which also satisfies the above requirements for $a, b$, and we can apply the previous example's result to show that the equilibrium for $1, 2$ decision is not unique!

In summary, firm $s$ will prefer a low price strategy. However, the decision of downstream firms $1, 2$ will be unpredictable (multiple equilibria) if $s$ choose the "optimal" price for low price strategy.

### 2.12.8 Non-Equilibrium (Computation of Example 11)




Assume:

$$p_{s_1} = c = 2; p_{s_2} = 0; a = 5; b = 2$$

1. High price strategy where market 2 is inactive, and price function at firm $c$ is $p_c = a - 2bX_c$, and price functions at sources are

$$p_1 = a - 4bx - 2by = 0,$$
$$p_2 = a - 2bx - 4by = 0.$$

Solve the above equations and flows at equilibrium

$$x = \frac{a - 2c}{6b}; y = \frac{a + c}{6b}$$

($a \geqslant 2c$ so that $x_h \geqslant 0$)

Double check the price at $c$

$$p_c = \frac{a + c}{3} \geq 1$$

It is a high price strategy and the payoffs are

$$\Pi_2^h = 2y^2 = \frac{(a + c)^2}{18b}$$

2. Low price strategy where both markets are inactive, and price function at firm $c$ is $p_c = \frac{a+b}{b+1} - \frac{2b}{b+1} X_c$, and price functions at sources are

$$p_1 = \frac{a + b}{b + 1} - \frac{4b}{b + 1}x - \frac{2b}{b + 1}y = 0,$$
$$p_2 = \frac{a + b}{b + 1} - \frac{2b}{b + 1}x - \frac{4b}{b + 1}y = 0.$$

By solving the above equations, we got the flows as

$$x = \frac{a+b-2c(b+1)}{6b}; y = \frac{a+c(b+1)}{6b} + \frac{1}{6}$$

$(a+b \geqslant 2c(b+1)$ so that $x_l \geqslant 0)$

Double check the price $c$

$$p_c = \frac{2}{3}\frac{a+b}{b+1} + \frac{c}{3}$$

It is a low price strategy, and the payoffs are

$$\Pi_2^l = y^2$$

Note that the high price strategy gives a higher payoff.

3. High price strategy is NOT an equilibrium.

Suppose firm 2 increases $y$ and try low price strategy:

$$p_2 = \frac{a+b}{b+1} - \frac{2b}{b+1}x_h - \frac{4b}{b+1}y = 0$$

The new flow is

$$y' = \frac{a+c}{6b} + \frac{1}{4}$$

Double check the price $c$

$$p_c = \frac{a+b}{b+1} - \frac{2b}{b+1}X_c \leqslant 1$$

To prove high price strategy is not an equilibrium,

$$\Pi_2^h = 2by^2 = 2b(\frac{a+c}{6b})^2 < b(\frac{a+c}{6b} + \frac{1}{4})^2 = by'^2 = \Pi_2^{h \to l}$$

4. Low price strategy is NOT an equilibrium.

Suppose firm 2 decreases $y$ and try high price strategy:

$$p_2 = \frac{a+b}{b+1} - \frac{2b}{b+1}x - \frac{4b}{b+1}y = 0$$

The new flow is

$$y' = \frac{a+c(b+1)}{6b} - \frac{1}{12}$$

Double check the price at $c$,

$$p_c = a - 2bX_c \geq 1$$

To prove low price strategy is not an equilibrium,

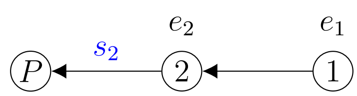
$$\Pi_2^l = by^2 = b(\frac{a+c}{6b} + \frac{c}{6} + \frac{1}{6})^2 < 2b(\frac{a+c}{6b} + \frac{c}{6} - \frac{1}{12})^2 = 2by'^2 = \Pi_2^{l \to h}$$

In summary, neither high nor low price strategy is equilibria.

## 2.13 Supplementary Materials

### 2.13.1 Parallel Model

This is a comparison with the three agents model in Section 2.2. In this section, we consider the case that the first signal $s_1$ is unobservable.




Working Sequence

Actually this case is equivalent to the following parallel model.

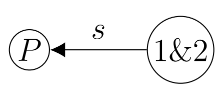
In the parallel model, two agents work simultaneously over the same take, again their effort is unobservable, but there will be an unique task signal indicates the result, i.e., task succeeds or fails. In summary, the given probability information is

$$p(e_1, e_2) = P(s = 1|e_1, e_2), \text{where } e_1, e_2 \in \{0, 1\},$$

or equivalently,

$$P(s = 1|e_1, e_2) = \alpha_1 e_1 + \alpha_2 e_2 + \tau e_1 e_2 + \gamma.$$

and the working sequence,




Parallel Working Sequence

In direct control case, each of them assumes the other is making effort. Denote the effort cost for agent $1, 2$ as $c_1, c_2$. For agent 1, given contract payment $M_1$, the utilities with and without effort are

$$\pi_1(e_1 = 1, e_2 = 1) = P(s = 1|e_1 = 1, e_2 = 1)M_1 - c_1,$$
$$\pi_1(e_1 = 0, e_2 = 1) = P(s = 1|e_1 = 0, e_2 = 1)M_1.$$

Therefore, the minimum effort payment for agent 1 is

$$M_1^0 = \frac{c_1}{P(s=1|e_1=1,e_2=1) - P(s=1|e_1=0,e_2=1)}. \tag{2.61}$$

Similarly, the minimum effort payment for agent 2 is

$$M_2^0 = \frac{c_2}{P(s=1|e_1=1,e_2=1) - P(s=1|e_1=1,e_2=0)}. \tag{2.62}$$

Cost of direct control

$$c_p^c = P(s=1|e_1=1,e_2=1)(M_1^0 + M_2^0).$$

WLOG, only consider delegate agent 1 to agent 2, and the following are the utility functions in different decisions

$$\pi_2(e_2, M_1) = p(e_2, I(M_1 \geq M_1^0))(M_2 - M_1) - c_2 e_2.$$

Specifically,

$$\begin{aligned}
&\pi_2(e_2=0, M_1=0) = P(s=1|e_1=0,e_2=0)M_2,\\
&\pi_2(e_2=0, M_1=M_1^0) = P(s=1|e_1=1,e_2=0)(M_2 - M_1),\\
&\pi_2(e_2=1, M_1=0) = P(s=1|e_1=0,e_2=1)M_2 - c_2,\\
&\pi_2(e_2=1, M_1=M_1^0) = P(s=1|e_1=1,e_2=1)(M_2 - M_1) - c_2.
\end{aligned}$$

Suppose agent 2 decided signing subcontract with 1, the additional benefit from personal effort is

$$\pi_2(e_2=1, M_1=M_1^0) - \pi_2(e_2=0, M_2=M_1^0) = P(s=1|e_1=1,e_2=1)(M_2 - M_1^0) - c_2 - P(s=1|e$$

and the low bound of $M_2$ to make the additional benefit greater than 0 is

$$\begin{aligned}
M_2 &\geq M_1^0 + \frac{c_2}{P(s=1|e_1=1,e_2=1) - P(s=1|e_1=1,e_2=0)}\\
&= M_1^0 + M_2^0.
\end{aligned}$$

Therefore, the cost of delegation is always larger than the cost of direct control

$$c_p^d \geq pM_2 \geq p(M_1^0 + M_2^0) = c_p^c.$$

This result can be generalized when there are $n$ parallel agents,

**Theorem 2.13.1** *Direct control is always better than delegation in the parallel model.*

**Comparison between sequential model and parallel model**

- Valuation of intermediate signal $s_1$

- Sunk cost
- Valuation of middle man

Consider a sequential model without $s_1$, basically a special case of the parallel model. Same as the previous section, the lower bound of contract payment for agent 2 to make effort and sign contract is

$$M_2 \geq M_2^0 + \tilde{M}_1^0.$$

There are two disadvantage of missing $s_1$. First, signal $s_2$ is not a good way to measure effort of agent 1, since the influence is $\alpha_1\beta_2$. Therefore, the minimum payment is $\tilde{M}_1^0 = \frac{c_1}{\alpha_1\beta_2}$. Instead of $\frac{c_1}{\alpha_1}$. Even though $\beta_2 = 1$, observing signal $s_1$ may still help.

Specifically, in the delegation case, agent 2 not only needs to decide his personal effort but also responsible for the subcontract with the downstream agent. Otherwise, shirking may happen and eventually harms agent 2 task completeness. Therefore, delegation gives a way to help the principal shift the cost of contract 1 to agent 2.

In contrast, if $s_1$ is unobservable in the delegation case, it will be hard for 2 to put effort, because higher success probability means a higher expected payment to 1.

While in the original model, note that agent 2's effort decision is made after observing $s_1$ as in Figure 2.4. Therefore, at that time, the subcontract cost $M_1$ is a sink cost,

- No effort: $\pi_2(e_2 = 0, s_1) = P(s_2 = 1|s_1, e_2 = 0)M_2 - cost(M_1)$.
- Effort: $\pi_2(e_2 = 1, s_1) = P(s_2 = 1|s_1, e_2 = 1)M_2 - cost(M_1) - c_2$.

and the benefit of putting effort is

$$\pi_2(e_2 = 1, s_1) - \pi_2(e_2 = 0, s_1) = (P(s_2 = 1|s_1, e_2 = 1) - P(s_2 = 1|s_1, e_2 = 1))M_2 - c_2 \geq 0,$$

which is **not** related with the subcontract. Hence, minimum effort payment to agent 2 is

$$M_2 \geq \frac{c_2}{P(s_2 = 1|s_1, e_2 = 1) - P(s_2 = 1|s_1, e_2 = 1)}.$$

### 2.13.2 Full Information

This is a supplementary material for Section 2.2 In this section, we consider the principal can observe $s_1$ even in the delegation case. The contract to agent 1 will be the same, but the principal can use signal $s_1$ to design a more flexible contract with agent 2,

$$r_2 = \begin{cases} M_2^-, & \text{if } s_1 = 1, s_2 = 1; \\ M_2^+, & \text{if } s_1 = 0, s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

Note that $s_2$ is useless to the contract with agent 1.

The main question will be

- How does the full information influence the principal's decision over direct control or delegation?
- How does it influence the principal and the agents' payoff?

Denote the minimum payment for agent 1 in direct control case is

$$M_1^0 = \frac{c_1}{\alpha_1}.$$

Denote agent 2's contract payment as,

$$r_2 = \begin{cases} \hat{M}_2^-, & \text{if } s_1 = 1, s_2 = 1; \\ \hat{M}_2^+, & \text{if } s_1 = 0, s_2 = 1; \\ 0, & \text{if } s_2 = 0. \end{cases}$$

and this is the decision graph of the delegation case

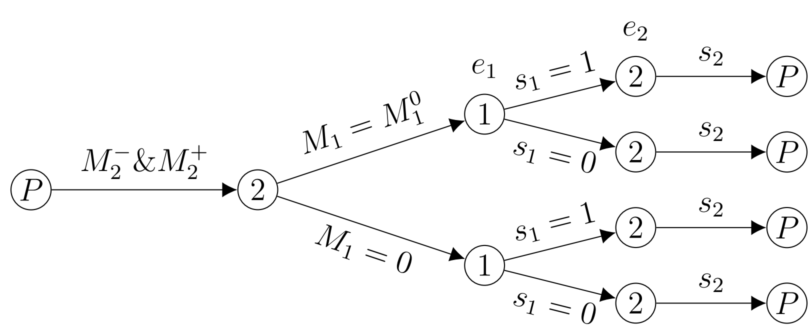



Fig. 2.28.: Decision Tree In Delegation with Flexible Contract

The cost of principal in the delegation case is

$$\hat{cost_3} = (1 - P(1))P(s_2 = 1|s_1 = 0, e_2 = 1)\hat{M}_2^+ + P(1)P(s_2 = 1|s_1 = 1, e_2 = 1)\hat{M}_2^-. \tag{2.63}$$

Following is the two potential decisions of agent 2:

- Effort, no subcontract

$$\pi_2^s = P(s_1 = 1|e_1 = 0)P(s_2 = 1|s_1 = 1, e_2 = 1)\hat{M}_2^- \tag{2.64}$$
$$+ (1 - P(s_1 = 1|e_1 = 0))P(s_2 = 1|s_1 = 0, e_2 = 1)\hat{M}_2^+ - c_2. \tag{2.65}$$

- Effort and subcontract

$$\pi_2^* = P(1)\big(P(s_2 = 1|s_1 = 1, e_2 = 1)\hat{M}_2^- - \hat{M}_1\big) \tag{2.66}$$
$$+ (1 - P(1))P(s_2 = 1|s_1 = 0, e_2 = 1)\hat{M}_2^+ - c_2. \tag{2.67}$$

Lemma 2.13.1 *In the delegation case, the minimum payment to agent* 2 *is*

$$\hat{M}_2^+ = M_2^+,$$
$$\hat{M}_2^- \geq \max\{M_2^-, \frac{P(1)}{\alpha_1 P(s_2 = 1|s_1 = 1, e_2 = 1)}M_1 + \frac{P(s_2 = 1|s_1 = 0, e_2 = 1)}{P(s_2 = 1|s_1 = 1, e_2 = 1)}M_2^+\}.$$

*and the bound is tight.*

**Proof** 1. For the incentive of agent 2 to put personal effort. Note that whether signed the subcontract or not, when $s_1$ is observed, the subcontract cost is a sunk cost, and will **not** influence agent 2's decision on personal effort.

Hence, the minimum payment for agent 2 to make effort in any condition is

$$\hat{M}_2^+ \geq M_2^+,$$
$$\hat{M}_2^- \geq M_2^-,$$

2. For the incentive to sign subcontract with agent 1 (only consider agent 2 always puts effort). Compare 2.66 with 2.64,

$$\pi_2^* - \pi_2^s = \alpha_1(P(s_2 = 1|s_1 = 1, e_2 = 1)\hat{M}_2^- - P(s_2 = 1|s_1 = 0, e_2 = 1)\hat{M}_2^+) - P(1)M_1.$$

In summary the optimization for the principal to solve under the delegation case is,

$$\begin{aligned} \min_{\hat{M}_2^+, \hat{M}_2^-} \quad & c_p \\ \text{subject to} \quad & \pi_2^* - \pi_2^s \geq 0, \\ & \hat{M}_2^+ \geq M_2^+, \\ & \hat{M}_2^- \geq M_2^-. \end{aligned} \tag{2.68}$$

It's easy to see that $\hat{M}_2^+ = M_2^+$ at the optimal. By solving $\pi_2^* - \pi_2^s \geq 0$, we have another lower bound of $\hat{M}_2^-$,

$$\hat{M}_2^- \geq \frac{P(1)}{\alpha_1 P(s_2=1|s_1=1,e_2=1)} M_1 + \frac{P(s_2=1|s_1=0,e_2=1)}{P(s_2=1|s_1=1,e_2=1)} M_2^+.$$

■

**Theorem 2.13.2** *When the contract is conditional on both signals, the principal prefers delegation if and only if,*

$$\frac{\gamma_1 c_1}{\alpha_1^2} \leq \frac{\beta_2+\gamma_2}{\alpha_2+\tau_2} c_2 - \frac{\gamma_2}{\alpha_2} c_2. \tag{2.69}$$

**Proof** Recall the cost of direct control 2.12 is,

$$\begin{aligned} cost_2 = &(1-P(1))P(s_2=1|s_1=0,e_2=1)M_2^+ \\ &+ P(1)P(s_2=1|s_1=1,e_2=1)M_2^- + P(1)M_1. \end{aligned}$$

Cost of delegation is

$$\begin{aligned} \hat{cost}_3 &= (1-P(1))P(s_2=1|s_1=0,e_2=1)\hat{M}_2^+ + P(1)P(s_2=1|s_1=1,e_2=1)\hat{M}_2^- \\ &= (1-P(1))P(s_2=1|s_1=0,e_2=1)M_2^+ + P(1)P(s_2=1|s_1=1,e_2=1)\hat{M}_2^-. \end{aligned}$$

If $\hat{M}_2^- = M_2^-$ is the tight bound, it's easy too see delegation is better. Now only consider the case that

$$\hat{M}_2^- = \frac{P(1)}{\alpha_1 P(s_2=1|s_1=1,e_2=1)} M_1 + \frac{P(s_2=1|s_1=0,e_2=1)}{P(s_2=1|s_1=1,e_2=1)} M_2^+.$$

Delegation is better if $\hat{cost}_3 \leq cost_2$,

$$P(1)P(s_2=1|s_1=1,e_2=1)\hat{M}_2(s_1=1) \leq P(1)P(s_2=1|s_1=1,e_2=1)M_2^- + P(1)M_1,$$

which is equivalent to

$$P(s_2=1|s_1=0,e_2=1)M_2^+ + \frac{P(1)}{\alpha_1} M_1 \leq M_1 + P(s_2=1|s_1=1,e_2=1)M_2^-,$$

which is equivalent to $(P(1) - \alpha_1 = P(s_1=1|e_1=0))$

$$\frac{P(s_1=1|e_1=0)}{\alpha_1} M_1^0 \leq P(s_2=1|s_1=1,e_2=1)M_2^- - P(s_2=1|s_1=0,e_2=1)M_2^+.$$

Equivalently,

$$\frac{\gamma_1 c_1}{\alpha_1^2} \leq (\alpha_2+\beta_2+\tau_2+\gamma_2)\frac{c_2}{\alpha_2+\tau_2} - (\alpha_2+\gamma_2)\frac{c_2}{\alpha_2}$$
$$= \frac{\beta_2+\gamma_2}{\alpha_2+\tau_2}c_2 - \frac{\gamma_2}{\alpha_2}c_2.$$

Otherwise, direct control is better. ■

When the environmental impact $\gamma_1 = \gamma_2 = 0$, threshold 2.69 becomes,

$$0 \leq \frac{\beta_2}{\alpha_2+\tau_2}c_2,$$

which always holds. Therefore, we have the following result.

COROLLARY 2.13.1 *When the environmental impact is zero, delegation always has an expected contract cost than direct control.*

EXAMPLE 22 (BENEFIT OF DELEGATION UNDER FULL INFORMATION)
***Setup:*** *For agent* 1, *the effort cost is* $c_1 = 1$, *and successful probabilities are,*

$$P(s_1 = 1|e_1 = 0) = 0,$$
$$P(s_1 = 1|e_1 = 1) = 0.4.$$

*where* $\alpha_1 = 0.4, \gamma_1 = 0$.

*For agent* 2, *the effort cost is* $c_2 = 2$, *and successful probabilities are,*

$$P(s_2 = 1|s_1 = 0, e_2 = 0) = 0,$$
$$P(s_2 = 1|s_1 = 1, e_2 = 0) = 0.5,$$
$$P(s_2 = 1|s_1 = 0, e_2 = 1) = 0.2,$$
$$P(s_2 = 1|s_1 = 1, e_2 = 1) = 0.7.$$

*where* $\alpha_2 = 0.2, \beta_2 = 0.5, \tau_2 = 0, \gamma_2 = 0$.

***Computation:***

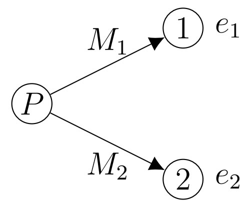
*By Equation 2.7, 2.8, the minimum effort payment for agent* 1 *and agent* 2 *is*

$$M_1 = 2.5,$$
$$M_2^+ = M_2^- = 10,$$

*and the successful probability when all the previous agents have make effort by equation 2.9,*

$$\begin{aligned}
P(1) &= P(s_1 = 1 | e_1 = 1) = 0.4, \\
P(2) &= P(s_2 = 1 | e_1 = 1, e_2 = 1) = 0.4.
\end{aligned}$$

*In the direct control case,*

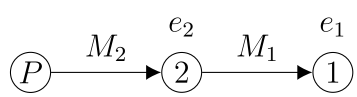



*Direct Control*

*the expected cost of direct control is,*

$$\begin{aligned}
\hat{cost}_3 &= P(1)M_1 + (1 - P(1))P(s_2 = 1 | s_1 = 0, e_2 = 1)M_2^+ + P(1)P(s_2 = 1 | s_1 = 1, e_2 = 1)M_2^- \\
&= P(1)M_1 + P(2)M_2 = 5
\end{aligned}$$

*In the delegation case,*




*Delegation*

*By Lemma 2.13.1,*

$$\begin{aligned}
\hat{M}_2^+ &= M_2^+ = 10, \\
\hat{M}_2^- &\geq \max\{M_2^-, \frac{P(1)}{\alpha_1 P(s_2 = 1 | s_1 = 1, e_2 = 1)} M_1 + \frac{P(s_2 = 1 | s_1 = 0, e_2 = 1)}{P(s_2 = 1 | s_1 = 1, e_2 = 1)} M_2^+\} \\
&= \max\{10, \frac{0.4}{0.4 \times 0.7} 2.5 + \frac{0.2}{0.7} 10\} \\
&= 10.
\end{aligned}$$

*and the expected cost for delegation is,*

$$\begin{aligned}
c_p^{dc} &= (1 - P(1))P(s_2 = 1 | s_1 = 0, e_2 = 1)\hat{M}_2^+ + P(1)P(s_2 = 1 | s_1 = 1, e_2 = 1)\hat{M}_2^- \\
&= P(2) \times \hat{M}_2 \\
&= 4.
\end{aligned}$$

*Thus, delegation is better than direct control with a lower expected cost for the principal.*

### 2.13.3 Continuous Model

This is a supplementary material for the three agents model in Section 2.2
**Question:**

- What's the opt utility of direct control when it's not concave, $\beta^2\xi \leq 4$ (condition 2.70).
- Careful about the boundary of $e$, by the condition that $0 \leq p_i \leq 1$.

In this section, we extend the binary effort level $e$ to a continuous variable, while the task signal is still binary[7].

The main question is how does the principal make the delegation decision in this situation?

For simplicity, we assume $\gamma_1, \gamma_2 = 0$, and

$$\begin{aligned} p_1 &= \alpha_1 e_1, \\ p_2 &= \alpha_1 e_1 + \beta_2 s_2. \end{aligned}$$

Different than the previous question, in order to have a non-trivial result, we use a quadratic effort cost function,

$$cost_1 = \frac{c_1}{2} e_1^2, cost_2 = \frac{c_2}{2} e_2^2.$$

For agent 1, given $M_1$,

$$\pi_1(e_1) = \alpha_1 e_1 M_1 - \frac{c_1}{2} e_1^2.$$

The optimal decision at $\pi_1'(e_1) = 0$ gives

$$\begin{aligned} e_1^* &= \frac{\alpha_1}{c_1} M_1, \\ p_1^* &= \frac{\alpha_1^2}{c_1} M_1, \\ p_1^* M_1 &= \frac{\alpha_1^2}{c_1} M_1^2. \end{aligned}$$

**Boundary condition**,

$$(\alpha_1 M_1 - c_1 e_1)(\frac{1}{\alpha_1} - e_1) = 0$$

[7]If the transform function is linear, continuous task signal may work too.

Given $M_2$, decision on personal effort is independent to contract to agent 1, similarly

$$e_2^* = \frac{\alpha_2}{c_2} M_2,$$
$$p_2^* = \frac{\alpha_2^2}{c_2} M_2 + \beta_2 p_1.$$

If the principal decides direct control,

$$\begin{aligned}
\pi_p(M_1, M_2) &= p_2^*(M_p - M_2) - p_1^* M_1 \\
&= (\frac{\alpha_2^2}{c_2} M_2 + \beta_2 p_1^*)(M_p - M_2) - p_1^* M_1 \\
&= (\frac{\alpha_2^2}{c_2} M_2 + \beta_2 \frac{\alpha_1^2}{c_1} M_1)(M_p - M_2) - \frac{\alpha_1^2}{c_1} M_1^2 \\
&= \frac{\alpha_2^2}{c_2} M_p M_2 + \beta_2 \frac{\alpha_1^2}{c_1} M_p M_1 \\
&- \frac{\alpha_1^2}{c_1} M_1^2 - \frac{\alpha_2^2}{c_2} M_2^2 - \beta_2 \frac{\alpha_1^2}{c_1} M_1 M_2.
\end{aligned}$$

Hessian matrix and condition for it to be concave!

$$4\frac{\alpha_2^2}{c_2} \geq \beta_2^2 \frac{\alpha_1^2}{c_1}. \tag{2.70}$$

Denote

$$\xi = \frac{\alpha_1^2}{c_1} / \frac{\alpha_2^2}{c_2}.$$

and the above condition can be rewrite as

$$\beta^2 \xi \leq 4.$$

Derivative over $M_1$ and $M_2$,

$$\frac{\partial \pi_p(M_1, M_2)}{\partial M_1} = \beta_2 M_p - 2M_1 - \beta_2 M_2 = 0, \tag{2.71}$$
$$\frac{\partial \pi_p(M_1, M_2)}{\partial M_2} = \frac{\alpha_2^2}{c_2} M_p - 2\frac{\alpha_2^2}{c_2} M_2 - \beta_2 \frac{\alpha_1^2}{c_1} M_1 = 0. \tag{2.72}$$

and we can solve the optimal solution

$$M_1^*(M_p) = \frac{\beta_2}{4 - \xi \beta_2^2} M_p. \tag{2.73}$$

And the optimal utility of the principal can be simplified by Equation 2.71,

$$\begin{aligned}
\pi_p^* &= (\frac{\alpha_2^2}{c_2}M_2 + \beta_2\frac{\alpha_1^2}{c_1}M_1)(M_p - M_2) - \frac{\alpha_1^2}{c_1}M_1^2 \\
&= (\frac{\alpha_2^2}{c_2}M_2 + \beta_2\frac{\alpha_1^2}{c_1}M_1)\frac{2}{\beta_2}M_1 - \frac{\alpha_1^2}{c_1}M_1^2 \\
&= \frac{2}{\beta_2}\frac{\alpha_2^2}{c_2}M_2M_1 + \frac{\alpha_1^2}{c_1}M_1^2 \\
&= \frac{1}{\beta_2}M_1(2\frac{\alpha_2^2}{c_2}M_2 + \beta_2\frac{\alpha_1^2}{c_1}M_1) \\
&= \frac{1}{\beta_2}\frac{\alpha_2^2}{c_2}M_pM_1^*.
\end{aligned}$$

By plug in Equation 2.73, we have the optimal utility in direct control case,

$$\pi_p^* = \frac{\alpha_2^2}{c_2}\frac{1}{4 - \xi\beta_2^2}M_p^2. \tag{2.74}$$

Now consider the delegation, optimal effort is still the same,

$$\begin{aligned}
\pi_2(e_2, M_1) &= p_2M_2 - p_1M_1 \\
&= (\frac{\alpha_2^2}{c_2}M_2 + \beta_2p_1)M_2 - \frac{\alpha_1^2}{c_1}M_1^2 \\
&= (\frac{\alpha_2^2}{c_2}M_2 + \beta_2\frac{\alpha_1^2}{c_1}M_1)M_2 - \frac{\alpha_1^2}{c_1}M_1^2 \\
&= \beta_2\frac{\alpha_1^2}{c_1}M_2M_1 - \frac{\alpha_1^2}{c_1}M_1^2 + \frac{\alpha_2^2}{c_2}M_2^2.
\end{aligned}$$

we can have the optimal delegation $M_1$ given $M_2$

$$\tilde{M_1}(M_2) = \frac{\beta_2}{2}M_2.$$

Utility of the principal is

$$\begin{aligned}
\pi_p(M_2) &= p_2(M_p - M_2) \\
&= (\frac{\alpha_2^2}{c_2} M_2 + \beta_2 \tilde{p_1})(M_p - M_2) \\
&= (\frac{\alpha_2^2}{c_2} M_2 + \beta_2 \frac{\alpha_1^2}{c_1} \tilde{M}_1)(M_p - M_2) \\
&= (\frac{\alpha_2^2}{c_2} M_2 + \frac{\beta_2^2}{2} \frac{\alpha_1^2}{c_1} M_2)(M_p - M_2) \\
&= (\frac{\alpha_2^2}{c_2} + \frac{\beta_2^2}{2} \frac{\alpha_1^2}{c_1})(M_p - M_2) M_2.
\end{aligned}$$

The optimal utility is

$$\tilde{\pi_p} = (\frac{\alpha_2^2}{c_2} + \frac{\beta_2^2}{2} \frac{\alpha_1^2}{c_1}) \frac{M_p^2}{4}, \tag{2.75}$$

with $\tilde{M_2}(M_p) = \frac{M_p}{2}$

**Theorem 2.13.3** *In the continuous effort case, the principal prefer delegation condition, if $\beta^2 \xi \geq 2$.*

**Proof** Recall the utility function in direct control 2.74 and delegation 2.75. The principal prefer delegation condition if $\pi_p^* \geq \tilde{\pi_p}$. By comparing them, we have

$$\beta^2 \xi \geq 2,$$

which is equivalent to

$$\frac{\beta^2}{2} \frac{\alpha_1^2}{c_1} \geq \frac{\alpha_2^2}{c_2}.$$

■

Note that the concave condition is $\beta^2 \xi \leq 4$.

### 2.13.4 Unbounded Depth

This is a supplementary material for Section 2.3, when the delegation depth is not bounded. First we know the size of control set is not $d$ anymore, but still linear to the number of agents.

**Proposition 2.13.1** *There are $k$ possible control sets for agent $k$, i.e., $|\Theta(k)| = k$.*

To bound the number of possible delegation structure in each control set, we use an additional assumption as follows.

**Assumption 2.13.1** *Agents can only sign contracts with their children in the process tree.*

With the above assumption, we know the delegation structure is fixed and unique once given the control set, because the sub-structure in the delegation tree has to be the same as the sub-structure in the process path.

Similar to the proof of Lemma 2.3.2, we show the one-to-one mapping still exists by induction,

LEMMA 2.13.2 *In the linear probability model, for any agent $k$, given $\theta_k^i, 1 \le i \le k$, there is an one-to-one mapping between minimum contract payments and effort status. And the choice of effort status is bounded by $k - i + 2$, i.e.,*

$$|\Psi_k(\theta_k^i)| \le k - i + 2.$$

Similar to the Theorem 2.3.3 in the more flexible delegation structure case, we have the same monotone inclusion property for the effort status here.

**Theorem 2.13.4** *The effort status satisfies monotone inclusion with the increasing of the contract payment.*

Since this case fixes the delegation structure, we introduce a new and simplified dynamic programming algorithm along the working process path, and DP stores

- Control set of $k$, $\Theta(k)$.
- For each control set $\theta_k^i \in \Theta(k)$, set of all possible effort status $\Psi_k(\theta_k^i)$.
- For each effort status $\psi_k \in \Psi_k(\theta_k^i)$, the corresponding minimum contract payment $M_k(\psi_k)$.
- Minimum expected cost till $k$, $cost_k$, and the corresponding optimal structure $\eta_k$.

where the **minimum expected cost till** $k$ is the minimum expect cost for the principal to motivate agents from 1 to $k$, denoted as $cost_k$.

While the first three parts are computed in the previous section, now provide the algorithm to update the minimum cost at each stage.

Set $cost_0 = 0$. For the first agent, cost till 1 is simply

$$cost_1 = P(1)M_1(1) = (\alpha_1 + \gamma_1)\frac{c_1}{\alpha_1}.$$

Suppose $cost_i, 1 \le i \le k-1$ at previous stages are all know, and $M_k(\theta_k^i, 1 \le i \le k)$ at current stage are all computed,

As showed in Fig. 2.29, the cost till $k$ with control set $\theta_k^i$ can be updated by

$$cost_k(\theta_k^i) = P(k)M_k(\theta_k^i) + cost_{i-1}.$$

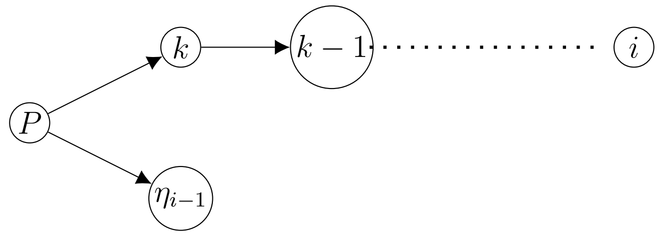



Fig. 2.29.: Control set

And the minimum expected cost till $k$ is

$$\begin{aligned} cost_k &= \max_{1\le i\le k} cost_k(\theta_k^i) \\ &= \max_{1\le i\le k} P(k)M_k(\theta_k^i) + cost_{i-1}. \end{aligned}$$

After find the optimal $\theta_k^{i^*}$. The optimal structure till $k$ (recording the set of agents directed controlled by the principal) is

$$\eta_k = \eta_{i^*} \cup \{k\}.$$

Therefore, we have the following algorithm to find the optimal delegation structure and the corresponding minimum expected cost,

**Algorithm 5** : Optimal Delegation Structure

**for** $k = 1$ to $n$ **do** ▷ agent $k$
  $cost_k \leftarrow 0$.
  **for** $i = 1$ to $k$ **do** ▷ control set $\theta_k^i$
    Given $\Psi_{k-1}(\theta_{k-1}^i)$, and $M_{k-1}(\psi_{k-1}|\theta_{k-1}^i)$.
    Plot $\pi_k(M_k|\theta_k^i)$ by 2.58.
    Intersection points gives $\Psi_k(\theta_k^i)$ and $M_k(\psi_k|\theta_k^i)$.
    $cost_k(\theta_k^i) = P(k)M_k(\theta_k^i) + cost_{i-1}$
    **if** $i = 1$ **or** $cost_k > cost_k(\theta_k^i)$ **then**
      $cost_k \leftarrow cost_k(\theta_k^i)$.
      $i^* \leftarrow i$
  $\eta_k = \eta_{i^*} \cup \{k\}$
**return** Optimal Structure $\eta_n$ and minimum expected cost $cost_n$

**Theorem 2.13.5** *Time complexity is $O(n^3)$.*

**Proof** As in Algorithm 5. There are $n$ stages ($n$ agents), and at stage $k$, there are $k$ control sets.

For each control set $\theta_k^i$, we use the status of $\theta_{k-1}^i$ from previous DP to plot the new piecewise linear function, with pieces bounded by $O(k)$.

Overall, the time complexity is $O(n^3)$. ■

### 2.13.5 Counter Examples

These examples are supplementary materials for Section 2.4. The following one shows smaller control set may have a larger expected cost

EXAMPLE 23

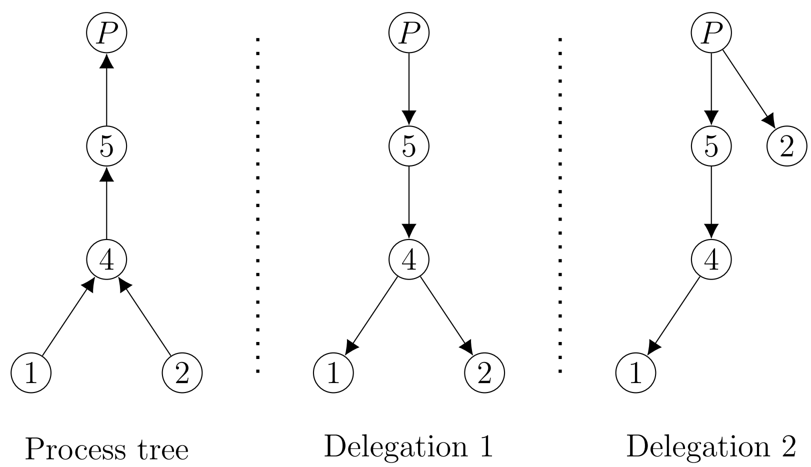



Fig. 2.30.: Example 1

*Assume* $\gamma_i = 0$ *and* $\alpha_i = \beta_i = 1/2$ *for every agent.*

$$M_i^0 = \frac{c_i}{\alpha}.$$

*At agent* 4*'s level,*

$$M_4(1) = \frac{\alpha M_1}{\beta\alpha} = \frac{M_1}{\beta},$$
$$M_4(12) = \frac{M_2}{\beta} \geq M_4(1) \geq M_4(4).$$

*In the delegation 1, at agent 5's level,*

$$\begin{aligned}
\pi_5^1(5412) &= p_5 M_5^1 - p_4 M_4(412), \\
\pi_5^1(541) &= (p_5 - \delta_2^5) M_5^1 - (p_4 - \delta_2^4) M_4(41), \\
\pi_5^1(54) &= (p_5 - \delta_1^5 - \delta_2^5) M_5^1 - (p_4 - \delta_1^4 - \delta_2^4) M_4(4).
\end{aligned}$$

*we have the payment lower bound for* $5$*, from* $\pi_5^1(5412) \geq \pi_5^1(541)$*,*

$$\begin{aligned}
M_5^1(5412) &\geq \frac{p_4 M_4(412) - (p_4 - \delta_2^4) M_4(41)}{\delta_2^5} \\
&= \frac{p_4 2M_2 - (p_4 - \delta_2^4) 2M_1}{\delta_2^5}.
\end{aligned}$$

*and from* $\pi_5^1(5412) \geq \pi_5^1(54)$*,*

$$\begin{aligned}
M_5^1(5412) &\geq \frac{p_4 M_4(412) - (p_4 - \delta_1^4 - \delta^4) M_4(4)}{2\delta^5} \\
&= \frac{p_4 2M_2 - (p_4 - 2\delta^4) M_4}{2\delta^5}.
\end{aligned}$$

*In the delegation 2, at agent 5's level,*

$$\begin{aligned}
\pi_5^2(541) &= p_5 M_5^2 - p_4 M_4(41), \\
\pi_5^2(54) &= (p_5 - \delta_1^5) M_5^2 - (p_4 - \delta_1^4) M_4(4).
\end{aligned}$$

*we have the minimum payment for* $5$

$$\begin{aligned}
M_5^2(541) &= \frac{p_4 M_4(41) - (p_4 - \delta_1^4) M_4(4)}{\delta_1^5} \\
&= \frac{p_4 2M_1 - (p_4 - \delta_2^4) M_4(4)}{\delta_1^5}.
\end{aligned}$$

*By setting* $M_2$ *close to* $M_1$ *but larger than* $M_4$*, we may have* $M_5^2(541) \geq M_5^1(5412)$*.*

The following example shows, control set update step is not enough and may not be linear

Example 24

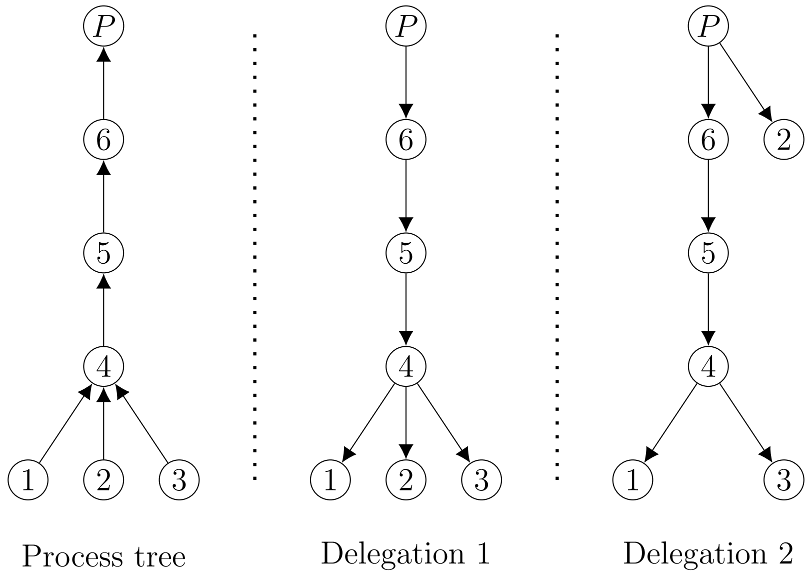



Fig. 2.31.: Example 2

*In the delegation 1, at agent* $6$*'s level,*

$$\pi_6^1(654123) = p_6 M_6^1 - p_5 M_5^1(54123),$$
$$\pi_6^1(65412) = (p_6 - \delta_3^6) M_6^1 - (p_5 - \delta_3^5) M_5^1(5412).$$

*we have the minimum payment for* $5$

$$M_6^1(654123) = \frac{p_5 M_5^1(54123) - (p_5 - \delta_3^5) M_5^1(5412)}{\delta_3^6}.$$

*In the delegation 2, at agent* $6$*'s level,*

$$\pi_6^2(65413) = p_6 M_6^2 - p_5 M_5^2(5413),$$
$$\pi_6^2(6541) = (p_6 - \delta_3^6) M_6^2 - (p_5 - \delta_3^5) M_5^2(541).$$

*We have the minimum payment for* $5$

$$M_6^2(65413) = \frac{p_5 M_5^2(5413) - (p_5 - \delta_3^5) M_5^2(541)}{\delta_3^6}.$$

*We know* $M_5^2(5413) = M_5^1(54123) + \frac{(p_4 - \delta_3^4)(M_4(12) - M_4(1))}{\delta_3^5}$*, but it's possible that* $M_5^2(541) \geq M_5^1(5412)$*.*

# VITA

## TAO JIANG

### PERSONAL INFORMATION

DoB: September, 1993
Place: Fenyi, Jiangxi, China

### EDUCATION

**Ph.D. in Quantitative Methods, 2018**
Krannert School of Management, Purdue University

**M.S. in Computer Science, 2018**
Department of Computer Science, Purdue University

**B.S. in Statistics, 2013**
School for Gifted Young, University of Science and Technology of China

### OUTDOOR ASCENTS (SELECTED)

Super Mario Extension (V6), Stone Fort, TN
Smooth Shrimp (V6), Bishop, CA
Perfectly Chicken (V5), Bishop, CA
Turtle Rock (V5), Boulder, CO
Ketron Classic (V4), Bishop, CA
Super Mario (V4), Stone Fort, TN
Dragon Lady (V4), Stone Fort, TN

### PROJECTS

Optimal Delegation Hierarchy in Project Management
with Thành Nguyen (In Preparation)

Quantity Competition in Multi-tier Supply Chain Networks
with Young-San Lin and Thành Nguyen (In Preparation)

Cocktail Sauce (V8), Bishop, CA
Every Color You Are (V6), Bishop, CA
The Hulk (V6), Bishop, CA