Reducing Interaction Cost: A Mechanism Design Approach

by
Yunqi Zhang

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2007

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this thesis we study the problem of requiring self-interested agents need to interact with some centralized mechanism where this interaction is costly. To improve their utility, agents may choose to interact with *neighbours* in order to coordinate their actions, potentially resulting in savings with respect to total interaction costs for all involved. We highlight the issues that arise in such a setting for the mechanism as well as for the agents.

We use a mechanism-design approach to study this problem and present a model for self-interested agents to form groups with neighbours in order to reduce the total interaction cost. Our model focuses on two aspects: reward-distribution and cost-sharing. We look at two scenarios for reward-distribution mechanisms and proposed a core-stable payoff as well as a fair payoff mechanism. We then propose a cost-sharing mechanism that agents can use to coordinate and reduce their interaction costs. We prove this mechanism to be incentive-compatible, cost-recovery and fair. We also discuss how agents might form groups in order to save on cost. We study how our final outcome (the total percentage of savings as a group) depends on the agents' interaction topology and analyze different topologies. In addition we carry out experiments which further validate our proposal.

Acknowledgments

First I would like to thank my supervisor, Kate Larson. I would like to thank her for all the effort and help she has dedicated to my research. She has provided me full support, as well as flexibility and freedom during my graduate studies. She provided me a model of a top research.

I would also like to thank the people on my thesis committee, Robin Cohen and Pascal Poupart, thank you for your time and valuable feedback on my thesis.

I would like to thank my fellow graduate students who provided me technical advise and friendship: Jie Zhang, Greg Hines and Reid Kerr.

Last but not least, I would like to thank Tony, for all the proof-reading.

Dedication

To my parents, for your love and faith in me. To Tony, for your love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Increasing attention has been given to multiagent systems where there are resource constraints, e.g., when communicating, agents must consume only limited communication bandwidth, or pay a certain amount of accessing fee. A lot of work has been done with respect to teamwork under limited resources [22]. In these settings, agents can benefit by coordinating with each other, possibly forming groups in order to save on the total communication cost incurred. In these situations agents are cooperative. However, in reality, often the agents we interested in are *self-interested*, meaning they will only act in their own self-interest, and are not concerned with the overall benefit of the society or the benefit of other agents. In this situation, agents might have diverging incentives. On one hand, agents may benefit from co-operating and coordinating in order to save on interaction costs. On the other hand, agents might not want to reveal the information that would allow for effective coordination in the first place. We are interested in determining how to design protocols so that agents will coordinate their actions when that is the best thing to do, given their own private costs and preferences.

1.1 Motivating Examples

There are many real world problems which can benefit from the ideas presented in this thesis. To illustrate, let us consider the following examples.

In the first scenario we have a network consisting of self-interested agents, each of whom is capable of monitoring local properties of the network. A centralized network operator wants to gain an overview of the entire network's functionality, and thus requests agents to provide their local information. However, agents wish to be rewarded for the information they provide. Additionally, agents are limited in their communication capabilities and so incur costs when communicating to the centralized network operator. However, some agents may reside in the same sub-network (local network) so the communication between them is virtually minimal. Thus, these agents may benefit by grouping together and gathering all their information together. Then this group will choose one agent to send the overall information to the centralized operator in order to save on the total interaction cost incurred.

In the second scenario there is a retailer auctioning off a set of items. Bidders can bid on any subset of the items, but are responsible for the shipping costs for any items they win in the auction. However, the shipping pricing scheme is such that there may be savings if agents coordinate by transporting a large number of items to one agent's location, and then distributing the items among the neighbouring agents. For example, the auctioneer could be located outside of the country so bidders need to pay a high international shipping rate for the item. But for those bidders in the same city, they can make savings by transporting all the items they want to one bidder first, and then redistribute among the other bidders, which will then only incur a small local shipping cost.

In the third scenario, we have students studying for an exam. If an individual student does not understand some of the course material, then he/she can always visit the professor's office hours to get help. However, if the students form a study group, they can share their knowledge, and only attend office hours when none of them understand something, thus reducing the amount of time and effort spent at office hours, while improving their understanding of the material.

While the above scenarios are motivated by very different applications, they can all fit in our model: one where self-interested agents must interact with some centralized mechanism, but this interaction is costly. Instead, agents may choose to interact with *neighbours* in order to coordinate their actions, potentially resulting in savings with respect to total interaction costs for all involved. At the same time, the center's mechanism is trying to obtain some desirable outcome which depends on the agents. For example, in the network setting the central mechanism wishes to maximize its knowledge about the network performance, while in the auction setting the auctioneer may wish to maximize social welfare by finding an efficient allocation. And in the last setting, the professor wants every student to learn the course material.

1.2 Our Approach

We use a mechanism-design approach to study this problem. In particular, we are interested in understanding under what interaction-conditions should self-interested agents coordinate their actions so as to reduce their overall interaction costs, and what sort of coordination mechanisms should be used. We introduce a model that consists of a center, a set of agents, an interaction cost from an agent to the center and an inter-agent interaction cost. We then study our model from three perspectives, the mechanism design techniques, the grouping strategies and the cost-sharing algorithms. For mechanism design techniques, we choose to study two different applications separately. The two applications we choose are the auction application and the information provision network application. These two are quite different and are both very popular subjects that have been extensively studied. Thus studying the two applications will cover a majority of scenarios that our model fits. The next aspect we study is the grouping strategy. We propose a group-formation technique for agents to form appropriate groups in order to make savings. Some major properties we desire are stability and individual rationality. We also analyze our grouping strategy with respect to different interaction topologies. The last part of our model is the cost-sharing protocol. We provide an incentive compatible cost-sharing scheme for the agents in the group to share the interaction cost. The cost-sharing protocol handles how the total cost is divided among group members. We would like the protocol to guarantee truthfulness and also *fairness*, which we will elaborate in Chapters Four and Five. We argue that this group formation and cost sharing approach does not affect the center's goals.

1.3 Contributions

The key contributions for this thesis are as follows:

A Model of Costly Interaction: We present a model for self-interested agents to form groups with neighbours in order to reduce total interaction cost.

Core-stable and Fair Payoff Algorithms for Reward-Distribution: We propose two reward-distribution mechanisms that feature core-stability and fairness. We prove that it is not possible to guarantee the existence of both properties together.

Cost-Sharing Protocol: We propose a novel cost-sharing protocol that is incentive compatible, cost-recovery and fair.

Analysis on Different Interaction Topologies: We observe that our total percentage of cost savings for the society depends on the agents' interaction topology and analyze different topologies.

1.4 Guide to the Thesis

In this section we outline the chapters for the rest of the thesis:

Chapter 2 - Background: In this chapter we provide the background information for cost-sharing, game theory and mechanism design used for this thesis.

Chapter 3 - Problem Description: In this chapter we describe our model in detail. We discuss two sample scenarios. We point out issues that arise in these settings with respect to the mechanisms and the agents and how we propose to overcome them. We introduce the main issues and components of our model.

Chapter 4 - Mechanism Design Aspects: In this chapter we describe the mechanism design techniques used in our model. In particular, we carry out the analysis under two different scenarios, the combinatorial auction scenario and the information-provision scenario. We propose different solutions under the two scenarios. For the information-provision network we use a regular VCG mechanism with reward-distribution. For the combinatorial auction scenario we look at the problem from a coalitional games perspective and propose two payoff strategies, with one being in the core and the other being *fair*. We also carry out experiments to test the effects of both strategies.

Chapter 5 - Cost-Sharing Mechanism and Grouping Strategy: In this chapter we present our novel cost-sharing protocol and the grouping strategy used in our model. We show that the cost-sharing protocol has all the properties necessary to support our model. We develop two sets of grouping strategies and compare the outcomes as well as the effect on different interaction topologies. We also include experiments for testing total cost-savings using different grouping strategies on various topologies in this chapter.

Chapter 6 - Related Research: In this chapter we discuss other work that has been done in this area and highlight the similarities and differences with our work.

Chapter 7 - Conclusion: In this chapter we conclude our work with a review of our contributions and a discussion of the future directions.

Chapter 2

Background

In this chapter we provide an overview of relevant game theory, mechanism design and coalitional games concepts that we use in this thesis. For a more complete overview on game theory and mechanism design, we direct the reader to Mas-Colell *et al.* [17]. For a more complete overview on coalitional games, please refer to Kahan and Rapoport [10].

2.1 Game Theory

Game theory is a branch of applied mathematics. It provides a mathematical framework for studying and predicting how agents will interact with each other in the decision making process.

A game consists of a set of agents, $N, (|N| = n)$, a set of actions, A_i, for each agent $i \in N$, and a set of outcomes, O. The key concept in game theory is a strategy.

Definition 1 (Strategy) *A strategy for agent i, denoted by s_i, is a contingency plan that specifies the action an agent should take at every point in the game when it has to take an action.*

Strategies can be either pure or mixed. Pure strategies are deterministic plans. A mixed strategy, $s_i \in \triangle(S_i)$ is a probability distribution over the set of all pure strategies, S_i, of agent i. We use the notation $s = (s_i, s_{-i})$ to denote the strategy profile where the strategy of agent i is s_i and $s_{-i} = (s_1, .., s_{i-1}, s_{i+1}, ..., s_n)$. A strategy profile, $s = (s_1, \ldots, s_n)$, is a vector specifying one strategy for each agent in the game. Each agent i tries to play a strategy to achieve its preferred outcome. We assume that agents' preferences are expressed in terms of utility functions.

Definition 2 (Utility Function) *The utility function of agent i, $u_i(\cdot)$, is a mapping from outcomes to the real numbers;*

$$u_i : (O) \mapsto \mathcal{R}$$

An agent prefers outcome o_1 to o_2 if $u_i(o_1) > u_i(o_2)$. For simplicity, when it is clear from the context, we use $u_i(s_i, s_{-i})$ to denote $u_i(o(s_i, s_{-i}))$.

A key goal of game theory is to find the stable solution in the space of strategy profiles. These stable solution are the equilibria of the game. The most well known equilibrium concept is the *Nash equilibria*. A Nash equilibrium is a strategy profile in which each agent is playing its optimal strategy, given the strategies the other agents are playing.

Definition 3 (Nash Equilibrium) *A strategy profile $s^* = (s_1^*, \ldots, s_n^*)$ is a Nash equilibrium if no agent has incentive to deviate from its strategy, given that the other agents do not deviate. Formally,*

$$\forall i, u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*) \forall s_i'.$$

Although the Nash Equilibrium is a fundamental concept in game theory, it does have several weaknesses. First of all, there can be multiple Nash equilibria existing in one game: thus, agents may not know which one to play. Also, the Nash equilibrium assumes agents have perfect information about all agents in the game. This is often not true.

A stronger solution concept is called the *dominant strategy equilibrium*. A strategy is dominant if it is the agent's best strategy against any strategy that other agents may choose.

Definition 4 (Dominant Strategy) *The strategy of agent i, s_i^* , is dominant if*

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i', s_{-i}), \forall s_{-i}, \forall s_i' \neq s_i^*.$$

A dominant strategy equilibrium is an equilibrium in which every agent has a dominant strategy. The dominant strategy equilibrium is a robust solution concept since it makes no assumptions about the information that agents have available to them. It also does not depend on other agents being rational since an agent's dominant strategy stays the same independent of how other agents act. Unfortunately, however, many games do not have dominant strategy equilibria.

2.2 Mechanism Design

Mechanism design is a subfield of game theory and is sometimes called *reverse game theory*. Given a group of rational agents, mechanism design studies how to design rules of a game for the agents so to achieve a specific outcome. It has been applied in a lot of areas including auctions and electronic markets [16, 25, 30].

We assume that there is a set of agents, N, $|N| = n$. Each agent, i, has a *type*, $\theta_i \in \Theta_i$, which represents the private information of the agent that is relevant to the agent's decision making. This may include the agent's payoff function, its beliefs about other agents, its beliefs about other agents' beliefs about it, and so on. In particular, an agent's type determines its preferences over different outcomes. We use the notation $u_i(o, \theta_i)$ to denote the utility of agent i with type θ_i for outcome $o \in \mathcal{O}$ ($\mathcal{O}$ is the space of possible outcomes). The goal of mechanism design is to implement some system-wide solution, captured by a social-choice function $f(\cdot)$ which is a function of the agents' types. The mechanism design problem is to implement a set of "rules" so that the solution to the social choice function is implemented despite agents' acting in their own self-interest.

Definition 5 (Social Choice Function) *A social choice function is a function $f : \Theta_1 \times \ldots \times \Theta_n :\mapsto \mathcal{O}$, that, for each possible profile of agents' types, $\theta = (\theta_1, \ldots, \theta_n)$, it assigns an outcome $f(\theta) \in \mathcal{O}$.*

Definition 6 (Mechanism) *A mechanism $M = (S_1, \ldots, S_n, g(\cdot))$ defines the set of strategies S_i available to each agent and an outcome rule $g : S_1 \times \ldots \times S_n :\mapsto \mathcal{O}$, such that $g(s)$ is the outcome implemented by the mechanism for strategy profile $s = (s_1, \ldots, s_n)$.*

A mechanism *implements* a social choice function $f(\cdot)$ if there is an equilibrium of the game using the mechanism which results in the same outcomes as $f(\cdot)$ for every profile of types, θ.

Definition 7 (Implementation) *A mechanism $M = (S_1, \ldots, S_n, g(\cdot))$ implements social choice function $f(\cdot)$ if there is an equilibrium strategy profile $s^* = (s_1^*, \ldots, s_n^*)$ such that $g(s^*(\theta)) = f(\theta)$ for all θ.*

While the space of possible mechanisms for implementing a particular social choice function can be very large, we can restrict ourselves to *direct mechanisms* which are those mechanisms where agents' strategies can consist only of announcing a type.

Definition 8 (Direct Mechanism) *A direct mechanism is a mechanism in which $S_i = \theta_i$ and $g(\theta) = f(\theta)$ for all θ.*

We will be particularly interested in an important class of direct mechanisms, namely the *incentive-compatible* direct mechanisms, which are those mechanisms where agents are best off truthfully revealing their type.

Definition 9 (Incentive Compatible) *A social choice function is incentive compatible if the direct revelation mechanism $M = (\Theta_1, \ldots, \Theta_n, f(\cdot))$ has an equilibrium $(s_1^*, \ldots, s_n^*)$, where $s_i^*(\theta_i) = \theta_i$ for all $\theta_i \in \Theta_i$ and for all i.*

If the equilibrium is a dominant-strategy equilibrium, then we say that the social choice function is *strategy-proof*.

Another important mechanism property is *individual rationality*.

Definition 10 (Individual rationality) *A mechanism is* individual-rational *if for all types θ_i, it implements a social choice function $f(\theta)$ such that*

$$u_i(f(\theta_i, \theta_{-i})) \geq \overline{u_i}(\theta_i)$$

where $\overline{u_i}(\theta_i)$ is the utility the agent could get when not participating.

In mechanism design problems, agents usually have the freedom to choose whether they wish to participate in the mechanism. If the utility that they can achieve by not participating is greater than what they can obtain through the mechanism, then the mechanism is not individually rational. Sometimes individual rationality is called voluntary participation. To further extend this notion, there is also *group rationality*.

Definition 11 (Group Rationality) *For a group of N players, let $v(N)$ be the amount they jointly receive from a mechanism and $u(i)$ be the utility they receive individually, then we say this mechanism is group-rational if:*

$$v(N) > \sum u(i)$$

An even stronger notion, which requires the above to be true for any subset of a group, is then called *coalitional rationality*. It means no combination of agents should receive a better utility than what they can collectively obtain by forming a coalition.

While mechanisms can be implemented across a wide spectrum of environments, in this paper we restrict ourselves to settings where agents are risk neutral and have *quasi-linear preferences*.

Definition 12 (Quasi-linear Preferences) *A quasi-linear utility function for agent i with type θ_i is of the form:*

$$u_i(o, \theta_i) = v_i(x, \theta_i) + t_i$$

where outcome o defines a choice $x \in \mathcal{K}$ from a discrete choice set $\mathcal{K}$ and a transfer t_i by the agent.

The notation $v_i(x, \theta_i)$ represents the valuation function of agent i, that is, the value the agent places on $x \in \mathcal{K}$. Mechanisms for quasi-linear settings specify both the choice $x(\theta)$ which affects the agents' valuation functions, along with the transfer functions t_i

for each agent. The transfer function is the payment either made to the agent or collected from the agent. It can be either positive or negative.

In quasi-linear settings, we are often interested in whether or not a mechanism is *social-welfare maximizing* (or *efficient*). We say that a mechanism is maximizing social-welfare if it selects an outcome $o = (x, t_1, \ldots, t_n)$ such that $\sum_i v_i(x, \theta_i)$ is maximized.

Definition 13 (Efficient) *A social choice function $f(x(\theta); t(\theta))$ is efficient if for all types $\theta = (\theta_1, \ldots, \theta_n)$:*

$$\sum_i v_i(x, \theta_i) \geq \sum_i v_i(x', \theta_i) \forall x' \in \mathcal{K}.$$

A second property of interest in quasi-linear settings is *budget-balance*. A mechanism is (strongly) budget-balanced if $\sum_{i=0}^{n} t_i = 0$. If a social choice function is budget-balanced then the system does not collect any payment or no net payment is taken out of the system.

Definition 14 (Budget balanced) *A social choice function $f(\theta) = (x(\theta); t(\theta))$ is budget-balanced if for all types $\theta = (\theta_1, \ldots, \theta_n)$:*

$$\sum_{i=0}^{n} t_i(\theta) = 0.$$

An important family of quasi-linear, incentive-compatible, social-welfare maximizing mechanisms are the Vickrey-Clarke-Groves (VCG) mechanisms.[1] In fact, the VCG mechanism is the only class of incentive-compatible and efficient mechanisms. In a VCG mechanism each agent reports a type $\hat{\theta}_i$ to the mechanism. This type is not required to

[1]VCG mechanisms are not necessarily budget-balanced.

be its true type. Given the reported types, the mechanism produces an allocation $k^*(\hat{\theta})$ which is efficient. That is

$$k^*(\hat{\theta}) = \arg\max_{x \in \mathcal{K}} \sum_i v_i(x, \hat{\theta}_i).$$

The transfer rules of the VCG mechanism are defined as

$$t_i(\hat{\theta}) = h_i(\hat{\theta}_{-i}) - \sum_{j \neq i} v_j(k^*, \hat{\theta}_j)$$

where $h_i : \Theta_{-i} \mapsto \mathcal{R}$ is an arbitrary function which does not depend on the declared type of agent i. The pivotal mechanism is a VCG mechanism with:

$$t_i = \sum_{j \neq i} v_j(k'_j) - \sum_{j \neq i} v_j(k^*_j)$$

Where $k' = (k'_1, \ldots, k'_{i-1}, k'_{i+1}, \ldots, k'_n)$ is the allocation that maximizes the sum of all agents' valuations assuming that agent i did not participate. In the rest of this paper when we refer to a VCG mechanism we mean the pivotal mechanism.

2.2.1 Auctions

An important class of mechanisms are auctions, which are well studied examples of resource-allocation mechanisms. If an auctioneer wants to sell an item to the agent with the highest value for the item, it could run a Vickrey auction (also known as the *second-price-sealed-bid* auction). The Vickrey auction, which is an example of a VCG mechanism, works in the following way. Bidders are asked to submit bids to the auctioneer privately. The auctioneer sells the item to the highest bidder, who pays an amount equal to the second highest bid. We know that in a VCG mechanism, the payment is defined as $t_i = \sum_{j \neq i} v_j(k'_j) - \sum_{j \neq i} v_j(k^*_j)$. In an auction scenario, $v_j(k'_j)$, the valuation if the winner is not present, is equal to the bid of the second highest price, and the term $\sum_{j \neq i} v_j(k^*_j)$ equals zero. Thus the Vickrey auction follows the payment scheme of a VCG mechanism and is then incentive compatible since no agent has incentive to submit a bid not equal to its true valuation. It is also efficient since the agent who valued the item the most receives the item.

The Vickrey auction is an example where only one type of item is on sale. There are other auction types where bidder can bid for multiple items as a bundle. This is known as a combinatorial auction. Combinatorial auctions are auctions in which multiple goods are sold simultaneously. In a combinatorial auction, bidders are allowed to place bids on arbitrary combinations of these goods.

Definition 15 (Combinatorial auction) *A combinatorial auction problem is a tuple,* $(N, X, v_1, ..., v_n)$*, where* N *is a set of* n *agents,* X *is a set of* m *goods, and for each agent* $i \in N, v_i : 2^X \rightarrow \mathcal{R}$ *is a valuation function. Most commonly, the combinatorial auction problem is to select an allocation* $a : 2^X \rightarrow N$ *of goods to agents that maximizes some measure such as total revenue to the auctioneer, or efficiency.*

First let us consider a naive implementation of combinatorial auction where the auctioneer simply calculates the valuation that maximizes the social welfare and charges winners their bid.

Example 1 (Naive Combinatorial Auction) *Let there be three agents* A, B, C *and two items* g_1, g_2 *to bid o. Agents can bid on either a single item or on the bundle* $\{g_1, g_2\}$*. An agent's bid is represented by a tuple: (a bid for* $\{g_1\}$*, a bid for* $\{g_2\}$*, a bid for* $\{g_1, g_2\}$ *where the bids are XOR'ed together). Suppose the agents bid as follows:*

- *Agent A's bid: (0,0,100)*

- *Agent B's bid: (75,0,0)*
- *Agent C's bid: (0,45,0)*

Given the bids submitted, the auctioneer will calculate the winner to be bidder B and bidder C and bidder B pays 75 while bidder C pays 45. However, if the auctioneer simply calculates the payment by the winner's bid, agent C would benefit by declaring its valuation of item g_2 to be 26 instead of 45, so that together agent B and C still win but agent C will pay a much lower cost.

To solve this problem, we use a VCG mechanism (also referred to as the GVA - Generalized Vickrey Auction), a direct generalization of the second price sealed bid auction to the combinatorial case, to solve a combinatorial auction. Using the VCG mechanism, the winners are determined by the bids that maximize social welfare. The cost is determined by the social welfare if the winner is not present in the auction minus the social welfare for the rest of agents except the winner. To illustrate, let us consider the following example.

Example 2 (GVA Example) *We now provide an example to see how the Auction works. Let there be two agents, agent A and agent B, and let there be two items, g_1 and g_2. Agents can bid on either item or on the bundle $\{g_1, g_2\}$. An agent's bid is represented by a tuple: (a bid for $\{g_1\}$, a bid for $\{g_2\}$, a bid for $\{g_1, g_2\}$ where the bids are XOR'ed together). Suppose the agents bid as follows*

- *Agent A's bid: (20,5,25)*
- *Agent B's bid: (10,15,30)*

The auctioneer allocates g_1 to agent A and g_2 to agent B since this allocation maximizes the sum of the agents' valuations. The amount that each agent pays is computed as *follows. If agent A did not bid, then* $\{g_1, g_2\}$ *would have been allocated to agent B whose valuation for this bundle is 30. When* g_1 *is allocated to agent A, agent B's valuation is only 15 since it receives* g_2*. Therefore, agent A's payment is calculated as* $30 - 15 = 15$ *and its utility is* $20 - 15 = 5$*. Agent B's payment is* $25 - 20 = 5$ *and its utility is* $15 - 5 = 10$.

2.3 Coalitional Games

An important set of concepts used in this paper are coalition formation and coalitional games. In particular, we study one application of our model from a coalition-formation perspective. We analyze core-stability in the mechanism of our model. We are interested in determining under what conditions can we achieve core-stability using our mechanism design approach. We chose to analyze this concept because of the nice properties it has. Thus here we provide some basic concepts that will be used later on.

Given a set of agents, a coalition game is defined simply by how well each set of agents (or coalition) can do for itself.

Definition 16 (Coalitional Game) *A coalitional game (with transferrable payoffs) is a pair* (N, v) *where*

- $N = 1, ..., n$ *is a set of players*
- $v : 2^N \rightarrow \mathcal{R}$ *is the characteristic function, attaching a real-valued value to each coalition*

Usually, the assumption of *super-additivity* is made:

- If $S \cap T = \emptyset$ then $v(S \bigcup T) \geq v(S) + v(T)$. This means in particular that the value of the entire set of players (the grand coalition) is no less than the value of any other coalition.

Finally, for convenience, we normalize the value of the empty coalition to be zero.

Example 3 (Coalitional Game Example) *Let us consider a game with three players. Three players bargain in pairs to form a deal. The deal is simply to determine how much to reward to the pair of players depending on which pair concludes the deal. If players* A *and player* B *form a pair, then they split* 5.00*. If players* A *and* C *form a pair, then they split* 3.00 *and if players* B, C *form a pair, they split* 6.00*. Any player alone cannot win the game and the three cannot be paired all-together.*

The game described is a 3-person coalitional game. The characteristic functions for this game are defined as follows:

$$v(A) = v(B) = v(C) = v(ABC) = 0, v(AB) = 5.00, v(AC) = 3.00, v(BC) = 6.00$$

If player A and B chose to pair up and split their shares equally, the outcome for this coalitional game is:

$$(2.50, 2.50, 0.00; AB, C)$$

Given the definition of a coalitional game, analyzing coalitional games amounts to deciding how $v(N)$ should be divided among the agents. In particular, we are interested in what *payoff vectors* should be used.

Definition 17 (Payoff Vector) *A payoff vector for a coalition game* (N, v) *is a vector* $(x_1, ..., x_n)$ *such that* $x_i \geq 0$ *and* $\sum_{i=1}^{n} x_i = v(N)$.

Coalitional game theory tries to understand the repercussions of using different payoff vectors. Properties that are looked for are stability and fairness. Two commonly used solution concepts are the *core* and *Shapley value*.

Definition 18 (The Core) *The core of a coalitional game consists of all the payoff vectors such that for all subsets $S \subset N$ it is the case that $\sum_{i \in S} x_i \geq v(S)$.*

In words, the core consists of the set of solutions that satisfy individual, group and coalitional rationality. If a payoff vector is in the core, then under no circumstances will any agent want to deviate from this solution and form or join a different coalition.

For example, consider the Treasure of Sierra Madre game [28].

Example 4 (Treasure of Sierra Madre game) *In the Treasure of Sierra Madre game, a set N of gold prospectors find a treasure of many (more than $2|M|$) gold pieces in the mountains. Each piece can be carried by two people but not by a single person. The situation can be modeled by a game (N, v), where $v(S) = \llcorner \frac{|S|}{2} \lrcorner$. In this game, if $|N| \geq 4$ is even then the core consists of a single payoff vector $(\frac{1}{2}, \frac{1}{2}, .., \frac{1}{2})$. This is because if every two persons can carry a piece of gold, then each of them receives $\frac{1}{2}$ as the payoff. But if $|N| \geq 3$ and is odd, then the core is empty. This is because, since every piece of gold has to be carried by two persons, the payoff for everyone who carries a piece of gold is $\frac{1}{2}$ as in the previous case. However, since $|N|$ is odd, there is always going to be one person who cannot carry anything and has a payoff of zero. Thus for any subset S (*with size s is even*) including this person, the total payoff would be:*

$$\sum_{i \in S} x_i = \frac{1}{2} * (s - 1)$$

And the valuation of the subset is:

$$v(S) = \llcorner \frac{|S|}{2} \lrcorner = \frac{s}{2}$$

As a result, $\sum_{i \in S} x_i < v(S)$*,which contradicts the definition of core.*

From the Treasure of Sierra Madre game, we can see that there are only two possible situations. When $|N| \geq 4$ is even, a core always exists and has the single unique solution of $(\frac{1}{2}, ..., \frac{1}{2})$. On the other hand, when $|N| \geq 3$ and is odd, the core is always empty. There can also be cases where the core is not empty, but is not unique either, as illustrated in the next example:

Example 5 (3-Person Game) *Consider a 3-person game, let there be three agents A,B and C and define*

$$v(AB) = 90, v(AC) = 80, v(BC) = 70, v(N) = 135.$$

Let the payoff vector be (x_A, x_B, x_C)*, we need to solve for:*

$$x_A + x_B \geq 90$$

$$x_B + x_C \geq 70$$

$$x_A + x_C \geq 80$$

and we know $x_A + x_B + x_C = 135$*. Consider solution sets* $(65, 55, 15)$ *and* $(65, 25, 45)$*, we can see both of these solutions satisfy our equations. Thus both of the solutions are in the core, which means the core in this game does exist, but is not unique.*

From the above examples, we can see that the core can be empty or nonempty. When it is nonempty, it can be either unique or not. When the core does exist, it gives us stability. When the core is nonempty, the cooperative demands of every coalition can be granted, thus there is no need for a social mechanism for resolving conflicts and setting priorities. In other words when a core is nonempty, the whole system is stable, since no one would have a desire to leave. Having said that, the existence of empty cores does limit its usefulness. Thus people sometimes use other solution concepts for coalitional games. An important one of them is, the *Shapley value*.

Definition 19 (Shapley Value) *Given a coalitional game (N, v), the Shapley Value of player i is:*

$$\rho_i(v) = \frac{1}{N!} \sum_{S \subseteq N} |S|!(|N| - |S| - 1)![v(S \bigcup i) - v(S)]$$

In words, this can be viewed as capturing the *average marginal contribution* of an agent. For a player i, the Shapley Value $\rho_i(v)$ gives the payoff of each agent. It is a summation over all the subsets of the coalition S where i is a member, with a numerical coefficient multiplying the difference between the value of any subset S and the value of that coalition without i. It equals the weighted sum of the incremental additions made by the player to all coalitions that it is a member of. It thus reflects a player's marginal contribution in a coalition. The Shapley value is interesting because it always exists. It also captures a notion of fairness. The fairness here means that it gives more payoff to agents with a greater contribution. However, this fairness comes with a price of weakening the stability condition as we have guaranteed when the core exists. Furthermore, there is no guarantee that, when a core exists, the Shapley value for the group is in the core.

Chapter 3

Problem Description

In this chapter we provide a detailed description of the problem we study in this thesis, namely how agents should organize in order to reduce interaction costs. We start by giving two motivating examples which illustrate the types of problems we are interested in. We then define our model of the problem, and discuss the issues which we address later on in this thesis.

3.1 Motivating Examples

In this section we provide two motivating examples that we use throughout the thesis. The first example is from an e-commerce domain, whereas the second example is drawn from a network monitoring application. While these two domains initially appear to be quite different, we will show, in the next section, that they have more similarities than differences.

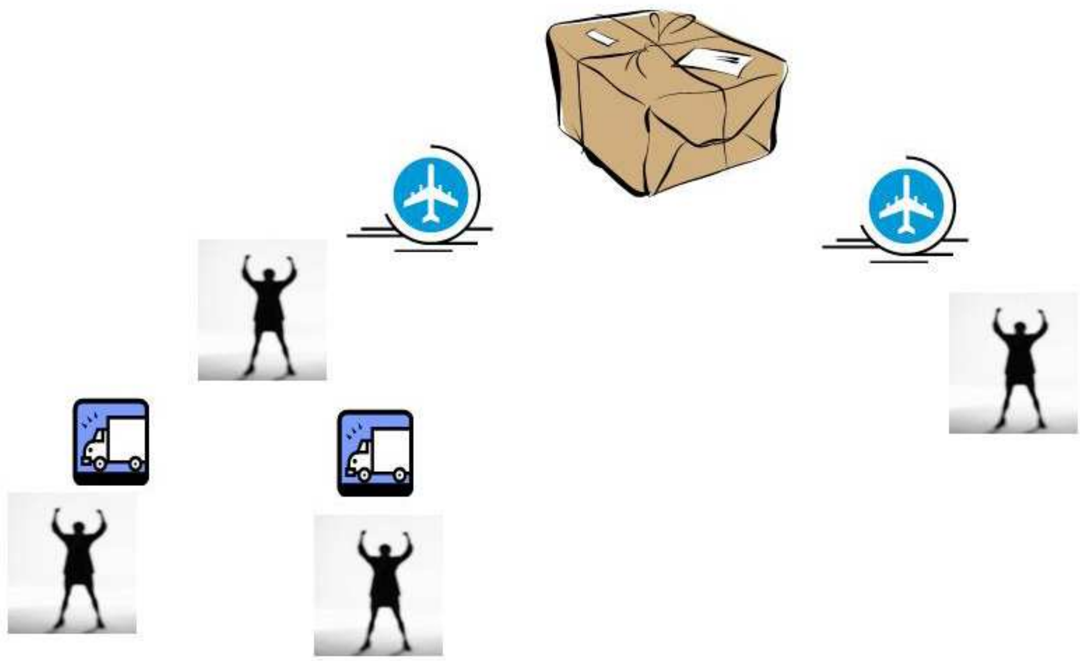

Figure 3.1: A real-world auction setting example

3.1.1 A Combinatorial Auction Problem

In Figure 3.1, an auctioneer is auctioning bundles of items to bidders. However, the problem is that the auctioneer is located quite far (e.g. in another country) from the bidders and a bidder, if it wins an item in the auction, has to pay the shipping cost. However, if agents in the same geographic location (i.e. the same city) coordinate and cooperate, they could submit a joint bid for all items the group is interested in, ship all items to a single location, and then deliver them locally. If they shared the shipping costs, it might be possible to reduce the costs of all agents in the group.

3.1.2 Information Provision Network Example

In the second example, we have a network consisting of self-interested agents, each of whom is capable of monitoring local properties of the network. A centralized network operator wants to gain an overview of the entire network's functionality, and thus requests the agents to provide their local information. However, the agents wish to be rewarded for the information they provide. Additionally, agents are limited in their communication capabilities and so incur costs when communicating to the centralized network operator. For example, there might be a one-time fee charged to contact the network operator, or there might be limited bandwidth so that agents need to pay a per-unit charge to transfer the information. However, some agents may reside in the same sub-network (local network) so the communication between them is minimal. Thus, these agents may benefit by grouping together and gathering all their information together. By grouping together they first choose a communicator (possibly the agent with lowest interaction cost to the network operator) and transfer all information to this communicator. Then the communicator sends the group information up to the network operator, pays the interaction cost and receives a reward. The reward and cost are then shared among group members using some pre-determined protocol independent of the network operator's knowledge. From the network operator's point of view, it treats a group the same as an individual agent.

3.2 The Model

At first glance, the two problems presented in the previous section are quite different. However, after a closer look, we can see there are many similarities between the two. In both applications self-interested agents need to interact with some centralized entity (the seller in the auction problem, and the network operator in the network problem). In

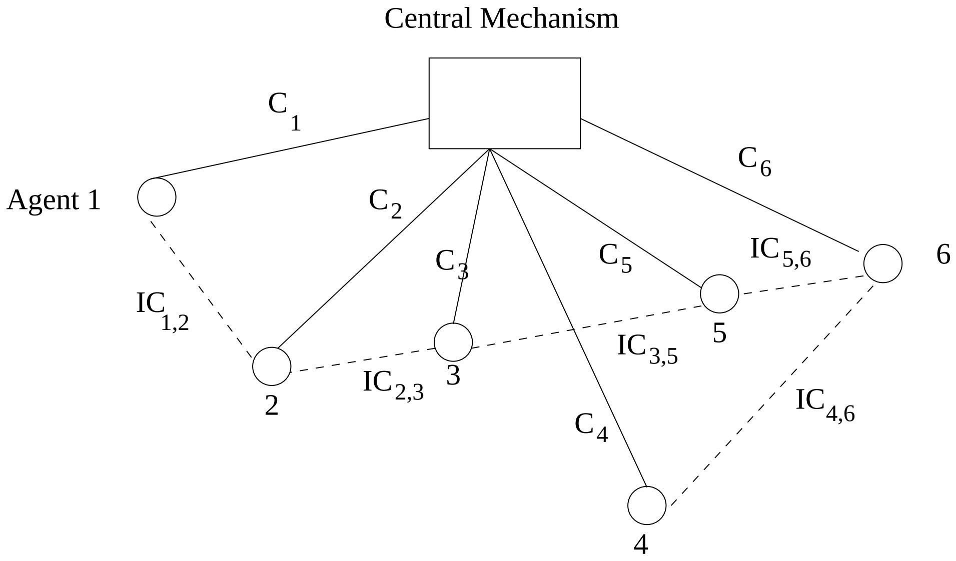

Figure 3.2: An example problem configuration. It costs agent 1 C_1 to communicate to the center. Agent 1 is also able to interact with its neighbour, agent 2, at a cost of $IC_{1,2}$.

both cases, by interacting with the central entity, the agents try to obtain some benefit. However, agents incur a cost (e.g. shipping cost, limited bandwidth). Thus, essentially, both applications need to solve the problem where self-interested agents need to interact with some central entity in exchange for some reward, and want to share and minimize the cost incurred during the interaction.

We present a general model for these problems. First, in our model there is a central entity, denoted by E. This center handles the overall reward/payment distribution. In terms of the combinatorial auction example [Example 3.1.1], the auctioneer is the center. The auctioneer has to determine who wins which items and report the shipping cost to each agent. In terms of the information provision example [Example 3.1.2], the centralized network operator is the central entity who collects information from the agents and rewards them for their efforts.

Second, in our model there are interaction costs from agents to the central entity. We denote this cost by C_i. We assume that this cost is only known to the agent itself and the central entity since in real life this is often the case. For example in the auction setting [Example 3.1.1], the interaction cost for an agent is the shipping cost if it wins an item in the auction.

Third, we assume that agents can interact with their neighbours. We denote an inter-agent interaction cost between agents i and j as $IC_{i,j}$. This is the cost of agents i and j interacting. We refer to this cost as the *inter-agent interaction cost*. In the auction example [Example 3.1.1], this is the local shipping cost between bidders who reside in the same city. In the information-provision network example [Example 3.1.2], this is the communication cost between agents who reside in the same local network. The interaction cost between two agents is known publicly to both agents.

Lastly, we need a set of n self-interested agents, with quasi-linear utility function.

An example of this interaction model is shown in Figure 3.2. Agent 2 can interact with the central mechanism at a cost of C_2. It can also directly interact with its neighbours 1 and 3 at a cost of $IC_{1,2}$ and $IC_{2,3}$ respectively. Without loss of generality we assume that inter-agent interaction costs are symmetric. This is merely to simplify notation, and does not affect our general results. We also assume that agents' interaction costs are additive. That is, for example, if agent 2 wishes to interact with agent 5, then the cost to agent 2 is $IC_{2,3} + IC_{3,5}$. Note that it is possible for an agent to be unable to interact with any other agents. We are particularly interested in settings where $IC_{i,j} << C_i$. This is a reasonable assumption, for example, in the combinatorial auction scenario, the international shipping rate is usually much more expensive (easily exceed a hundred times more for heavy items) than the local shipping rate. We need this assumption because otherwise agents will not benefit by forming groups.

We assume that agents derive some value from the interaction with the central mechanism. For example, if we are in an auction environment, the interaction result would be an allocation of items to the agent along with a price that the agent must pay for those items. In a network information-provision setting, an agent's value from an interaction is the amount it is paid to provide the information minus the cost it incurred from having to collect the information in the first place. We denote the utility derived by agent i from interacting with the central mechanism by

$$u_i^{\text{int}} = v_i(o, \theta_i) + t_i$$

where t_i is the transfer of money that is either made by the agent (i.e. the money agents pay for the item in the auction setting) or is made to the agent (i.e. the reward agents get in the information provision setting), o is the outcome of the interaction, θ_i is the type of the agent, and $v_i(o, \theta_i)$ is its valuation for the result of the interaction.[1]

3.3 Issues

We are interested in settings where the central mechanism is trying to ensure that agents truthfully reveal their types, while maximizing social welfare. In a standard setting, the central mechanism could simply run a VCG mechanism, and every self-interested agent would be best off interacting with the mechanism by truthfully revealing its type. The issue, however, is that in our setting, interactions are costly. An agent's utility function is actually

$$U_i = u_i^{\text{int}} - \mathcal{C}^i$$

[1]The $v_i(o, \theta_i)$ for the auction setting would be the value that the agent places on getting the items it is allocated. The $v_i(o, \theta_i)$ in the information provision setting would be the cost to the agent of collecting information to the center.

where $\mathcal{C}^i$ is the cost to agent i of the interaction. Agents wish to increase u_i^{int} while decreasing $\mathcal{C}^i$. In particular, we are interested in situations where there are savings to be had if *groups* of agents coordinate their interactions and *share* the cost. For example, in Figure 3.2 if C_3 is much less than C_2 and C_5 then agents 2 and 5 might want to coordinate with agent 3 and form a group. The three agents would then negotiate how to share the cost and reward amongst themselves. If an agreement is reached, a group will be formed and agent 3 will then represent the group to collect and send all the information to the central entity. Once a group has been formed, it is acting as a single unit. In other words, from the central entity's perspective, it treats each group as a whole and is not responsible for what happens within a group. A goal of this thesis is to understand how and when agents will coordinate their interactions and what are the aspects that will affect agents' decisions.

In the rest of this chapter, we describe some of the issues and challenges related to this model.

3.3.1 Two Levels of Mechanisms

The first issue we need to solve is how to encourage the self-interested agents to truthfully reveal their information/valuation to the mechanism. Our overall mechanism needs to be strategy-proof so that overall, our agents will not lie about their valuations or information. In addition, the model requires two levels of mechanisms. Inside each group, a mechanism has to exist to handle the reward-sharing for group members. As an example, in Figure 3.2, assuming a combinatorial auction scenario, agents 4, 5 and 6 may be able to form a group. If they decided to form a group, then a mechanism needs to be introduced so that each agent in the group will report its valuation truthfully to it. Then this mechanism will send the group valuation to the central entity. At the center's level, a central mechanism needs to exist to handle the reward-distribution amongst groups. Here the central entity runs a centralized mechanism to calculate which group gets what items and the payment to collect from the group. This centralized mechanism also needs to be incentive compatible so that the mechanism inside each group will not lie about the total group valuation. When rewards are assigned back to each group, the mechanism inside the group needs to determine the shares that each agent would get. A simple solution to guarantee truthfulness is to use the VCG mechanism at both levels, since the VCG mechanism is strategy-proof. However, this solution has some weaknesses. In particular, in this hierarchical approach, it is no longer guaranteed to maximize social welfare if agents are viewed as separate entities, though it is still guaranteed to maximize social welfare if groups are viewed as individual entities. This will be discussed further in Chapter Four.

3.3.2 Cost-Sharing Protocols

One of the motivations of this work is that agents will coordinate in order to share interaction costs. Thus we need to develop protocols that allow agents to share the costs. Since in our model, the interaction cost from the agent to the center is private, known only to the agent itself, we need a cost-sharing protocol that is incentive compatible so that agents will not benefit by lying to the mechanisms about their costs. While many cost-sharing algorithms have been studied previously [20, 19], a common assumption in these protocols is that the total cost is known in advance or can not be manipulated directly by the agents. Since in our model, an agent is chosen by the cost-sharing protocol to interact with the center, costs can be manipulated by the agents. We argue that our cost-sharing algorithm needs to be able to achieve this. Simple ideas such as dividing the shares equally amongst group members will not have this property. Lastly, the mechanism should not collect an excess amount. We want the protocol to collect exactly

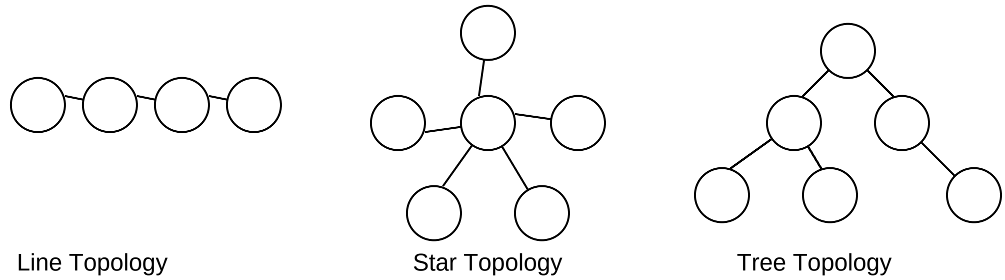

Figure 3.3: An example of three basic interaction topolgoies: Line, Star and Tree

enough to cover the actual interaction cost because we do not want excess money being collected by the mechanism. Therefore, we require a protocol that ensures that all agents are honest about their (potential) costs, and that shares the costs appropriately amongst the agents. We propose desired properties for a cost-sharing mechanism and introduce one that satisfies our required properties in Chapter Five.

3.3.3 Interaction Topology

Another important feature we study in our mechanism is the *interaction topology* of the agents. The interaction topology is a graph where each node represents an agent, and edges between nodes indicate that two agents can interact with each other. We borrow this concept from the concept of *network topology*, the study of arrangement or mappings of elements in a network. Figure 3.3 shows an example of three major topologies. In other words, the interaction topology in this paper describes the layout and neighbour information of the agents. The weight of edge (i, j) is $IC_{i,j}$ or the inter-agent interaction cost. The central entity in our model is not shown in these topologies, and each agent is able to communicate to the central entity alone. Figure 3.4 is an example of the line topology. As we can see, in this example, all agents are able to communicate to the center at a cost $C_i, i \in [1, 2, 3, 4]$, and Agents 1, 2, and 3 form a line topology. The interaction

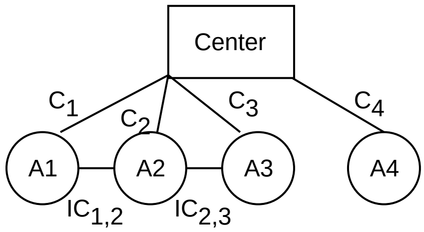

Figure 3.4: An example with line topology

cost between agents in the topology is $IC_{1,2}$ and $IC_{2,3}$.

In this thesis we will study how different interaction topologies affect the group formation process as well as the cost-saving in our model in Chapter Five.

3.3.4 Group Formation

The last aspect of our model is the grouping strategy agents follow. We need to determine how groups will be formed by agents. In particular, there are two requirements for our grouping strategy. First, we want individual rationality, i.e. we want a strategy that only forms groups when all the agents are better off (or at least not worse off) than if they acted alone. Second, we want our grouping strategy to satisfy, at a minimum, a weak stability condition so that agents, once inside a group, do not want to deviate and act on their own. We also analyze if and how grouping strategies are affected by different interaction topologies in Chapter Five.

3.4 Summary

In this chapter, we provided a detailed description of the problem we are studying. We first gave two motivating examples to illustrate the type of problems we are interested in. We then provided a detailed description of the problem, namely how agents should organize in order to reduce interaction costs. Finally, we presented our model, discussed the issues and challenges we face and how we plan to address them in this thesis.

Chapter 4

Reward Distribution Mechanism

This chapter discusses the mechanisms we use to handle reward-distribution among groups and between group members. The role of a reward-distribution algorithm is to allow agents to achieve their original goal while forming groups. For example, in an combinatorial auction setting, agents' goal is to win the item, and they will still want to achieve this while forming groups and save cost at the same time. This concept is thus separated from our cost-sharing and group formation concept and assumes groups have already been formed. As mentioned earlier, there are two levels of mechanisms to consider. The first level is at the center's level, and the second is at the group's level. Our general proposal is that the center simply runs a VCG at the top level to ensure truthful reporting of the valuations. We assume that each group, once formed, submits a single *group type* represented by a group valuation function, v_g, to the central mechanism. For example, in the combinatorial auction example, the group type would be the total items that each agent in the group wants, and in the information-provision example, it will be the valuation of the overall information of agents in the group. The central mechanism uses this information to determine the outcome. In the simplest scenario we assume that the central mechanism simply runs a VCG mechanism based on the v_g's it receives. That is, it chooses the outcome $o^* = \arg\max \sum_g v_g(o)$ and calculates transfers for each group $t_g = h_{-g}(v_{-g}) - \sum_{i \neq g} v_i(o)$. In the combinatorial auction, this will be a negative amount representing the money group pays to get the items, in the information provision setting this will be the amount of communication cost for the group to communicate to the centralized network operator. In order for the groups to determine what v_g is and how to share the transfers, we make each group run a VCG mechanism among its members. If o_g^* is the outcome of the group's VCG mechanism, then we set $v_g = \sum_{i \in g} v_i(o_g^*)$.

Proposition 1 *Under the center's VCG scheme, once a group G has formed a group valuation v_G, it is best off truthfully revealing it to the center and every agent in the group will truthfully reveal its valuation to the group.*

Proof:

Let us assume that agent i in the group declares its valuation to the group mechanism be $\widehat{v}_i$, different from its true valuation v_i. There are two cases. First, an agent will not declare a cost $\widehat{v}_i < v_i$ simply because it will have less chances of winning and will not be better off. Now what happens if it declares $\widehat{v}_i > v_i$? Since the center is running a VCG mechanism, as a group, the group will submit a truthful group valuation, if some agent declares a higher valuation, there must be some other agent who has to declare a lower valuation to make the group valuation unchanged. However, no agent will be willing to declare a valuation lower than its own. Given this situation, if agent i declares $\widehat{v}_i > v_i$, the the group valuation will be greater than its true valuation. Since agent i knows the center is running a VCG mechanism and as a group, they need to submit the true valuation, it will not declare its valuation to be higher. Thus agent i will report its true cost to the group. □

Unfortunately, this approach no longer guarantees that the final allocation among the agents themselves, when viewed from a global perspective, will be efficient.

Example 6 (Inefficiency Example) *Assume there are three agents* $\{A, B, C\}$ *and three items* $\{a, b, c\}$ *for sale. Assume that agent A wants item* a *and is willing to pay up to $3 for it, agent B wants item* b *and is willing to pay $2 for it, and that agent C wants bundle* $\{b, c\}$ *and is willing to pay up to $4 for it. If each agent interacted with the central entity independently, then the social welfare maximizing allocation would be for agent A to get item* a *and for agent C to get bundle* $\{b, c\}$*. However, if agents A and B formed a group and acted as a single agent, then their bid to the central entity would be $5 for bundle* $\{a, b\}$*, thus the final allocation would be that the group of agents A and B would get* $\{a, b\}$ *which they would allocate between themselves. This does not lead to a socially optimal outcome.*

The above example shows possible issues of inefficiency using our proposed mechanism. We are interested in developing methods to overcome this. However, we do not believe that a single approach will work for all applications. As a result, we classify this problem based on the valuation functions of the agents. Specifically, we study two scenarios. In the first scenario, items that agents want (or will be getting rewarded for) are substitutes. That is, an agent does not care which item it gets (or is rewarded for), as long as it gets one of them (The items are interchangeable). For example, if an agent wants one item from a set of N items, it does not care which of the N items it gets, as long as it gets one of them. In the second scenario, items are not substitutes. That is, if an agent wants item x with a certain valuation, then it will not be happy if we give it item y instead of x.

We choose two applications that represent each scenario to study the inefficiency problem. For the first scenario, we choose the information-provision network applica-

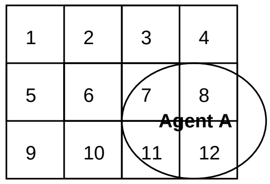

Figure 4.1: Example of the discretized areas

tion. In an information-provision network setting, agents want to be rewarded for providing the information to the centralized network operator. They do not care about which part of information the center rewards them for, as long as they get a reward. In the second scenario, we choose the combinatorial auction application, since in a combinatorial auction, bidders are only happy if they get the items they want. For example, if a bidder has a valuation on a TV set, it does not tell us if he likes a computer or how much he would like to pay for it. Furthermore, he will not be happy if we give him a computer instead of a TV set. Although we are studying the inefficiency problem on an application-specific basis, our results can be transferred to other domains. We are confident that by looking at these two different cases, we cover a wide range of applications.

4.1 Information-Provision Setting

Recall the information provision setting we introduced in the beginning of the thesis. In this setting, we make an assumption about the information structure of the agents. In particular, we assumed that information can be discretized into *areas* and an agent's value of a set of information areas is evenly distributed, and an agent's value of the information areas is additive. As an example, let us look at Figure 4.1. In this figure agent A covers areas $7, 8, 11, 12$. If the overall valuation for agent A over the four areas is V_{total}, then A's valuation for each single area will be $\frac{V_{total}}{4}$.

This assumption allows us to handle information overlap in a consistent and precise manner. The center still runs a VCG mechanism with each agent's type being the information it preserves and the value being the valuation of the information. The idea is that, if the bids of groups can be split so that the central mechanism can pin-point which part of the bid is causing conflicts with other bids, then we can reach better solutions.

Assume that two groups submit bids on bundles which overlap with each other. The central mechanism allocates the items using a standard VCG mechanism, but indicates which items caused a conflict between the bundles. The central mechanism can redistribute some of the money it collects back to all groups, to encourage everyone to participate. In particular, we can use a redistribution scheme as introduced by Cavallo [4] and Bailey [3], which has been proven to be incentive-compatible. Although this scheme does not necessarily affect agents' decision on whether to join the group or not, it does lead to better efficiency.

Definition 20 (Redistribution Scheme) *Assume I agents, $a_1, .., a_i$ have overlapping bids and the value of their overlapping items are $V_1, .., V_i$ respectively, with $V_1 > V_2 > \cdots > V_i$. If $i = 2$, then let $V_3 = V_2 - \beta, \beta > 0$.(We chose $\beta > 0$ to control how much we want to redistribute in the setting where only two agents overlap). Let Z_i be the payment redistributed back to each agent, then:*

$$Z_i = \begin{cases} \frac{V_3}{I} & i = 1,2 \\ \frac{V_2}{I} & i = 3,..,i \end{cases} \tag{4.1}$$

Since in this scenario the center's best interest is to collect maximum information. However, when an overlap occurs, it will collect an extra amount of money which it has no interest to keep. The above scheme thus allows the center to distribute part of this amount back to the agents.

Now let us summarize the overall mechanism design approach used by the center and the mechanism inside each group for this example. After each group is formed, a group-mechanism is assigned and is in charge of the payment and cost distribution amongst group members. Here we focus on reward distribution (cost-sharing is discussed in Chapter Five). After forming groups, the group-mechanism runs a standard VCG inside the group. It keeps a record of the bidding order and collects the sum of the winning bid as the groups submit bids to the center. The center then runs a VCG on these group bids. If no overlap occurs, the case is very simple and no extra step is needed.

Complications arise when there is an overlap of information. There can be two types of overlaps: at the group level or inside individual groups. When overlap occurs inside groups, the central mechanism does not need to be aware of it. The group-mechanism will take care of the overlap by using the redistribution algorithm to compensate the losing agents inside each group. When overlap occurs at the group level, the center will use the redistribution method and send back the information about which agent's bid caused the overlap. Then for the losing groups, the group-mechanism can use this information to charge the appropriate amount for the overlapping agent. In other words, the non-overlapping agents will not pay extra because of the existence of the overlapping agent. Let us use a concrete example to illustrate.

Example 7 (Payment-Redistribution with discretization) *Assume that all the information can be divided into 9 areas,* $1, \ldots, 9$. *Let there be three agents A, B and C with A and B in the same group G. Let agent A cover the information areas 1, 2, 3 and agent B cover 4 and 5. Let agent C cover the areas 5 and 8 Assume the three agents' value of the information is 6, 6 and 6 respectively. First, group G runs its internal VCG mechanism and obtains:* $t_A = t_B = 6 - 6 = 0$, $v_g = 6 + 6 = 12$. *The group then sends along its group bid for areas 1, 2, 3, 4, 5 with the value being 12.*

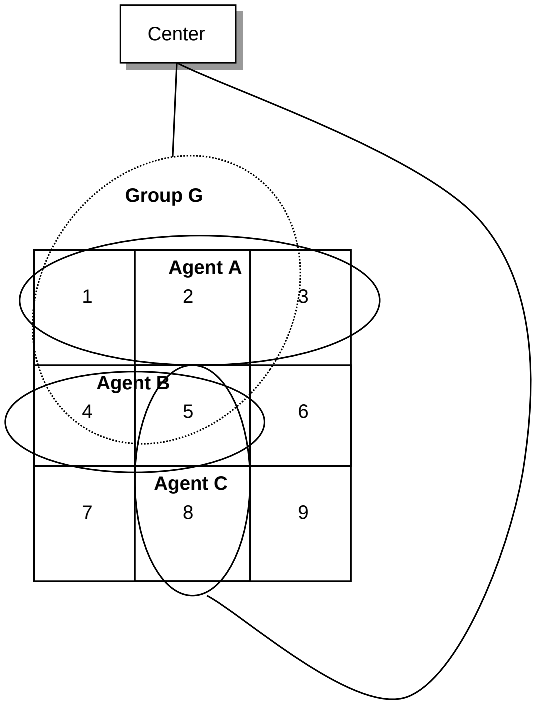

Figure 4.2: Example for reward-distribution in Information Provision Setting

Next, the center runs VCG with group G and agent C. Now we see there is an overlap of area 5. Since in the information provision setting we assume agents' value are evenly distributed, the center can run VCG on areas except the overlapping one first. That is, the center first runs VCG on areas 1, 2, 3, 4 and 8, with $v_G = \frac{12\times4}{5} = 9.6$ *and* $v_C = \frac{6}{2} = 3$ *There are no overlaps here so the payment sent back to the two will be* 9.6 *and* 3 *respectively. Now the center runs VCG on area 5, since* $v_G = 12 - 9.6 = 2.4$ *and* $v_C = 3$*, agent C will win and must pay* 2.4*. Agent C will also get back from the center a value of* 3*. Now using redistribution, each agent will get back from the center* $\frac{2.4-0.5}{2} = 1.05$*. Thus the total payment agent C gets :* $3+3-2.4+1.05 = 4.65$ *The total payment group G gets is :* $9.6 + 1.05 = 10.65$*.*

Now inside group G, the group-mechanism knows it is agent B who has the overlapping information, so agent A should get back 6 dollars and agent B will get $10.65 - 6 =$

4.65.

4.2 Combinatorial Auction Setting

The inefficiency issue we illustrated in the example at the start of the chapter not only shows possible inefficient outcomes, but also brings a factor of instability and unfairness in the combinatorial settings. To try solving these problems, we look at the combinatorial setting in a different way. First of all, we modify the bidding language and let the valuation sent to the center from the group be a permutation of XORs of all the possible bids. For example, if there are two agents (A,B) in the group and agent A wants item $\{x\}$ and agent B wants item $\{y\}$, then the group will send out the bids $\{x\}$ XOR $\{y\}$ XOR $\{x, y\}$, with the value of the combined bid simply being the summation of valuations of each individual bid. We try to analyze this setting from a coalitional game prospective.

4.2.1 Coalitional Games

As introduced in Chapter Two, there are different solution concepts for a coalitional game. The two most common ones are the core and the Shapley value.

The core is a well-known solution concept in coalitional games. Of all the theories to be considered, the core is the simplest to define and perhaps the most intuitively satisfying. If we can show that our payoff vector is in the core, it means the cooperative demands of every coalition can be granted, and coalitional, group and individual rationality are all satisfied. It gives us a stable solution. However, one disadvantage of the core is that it does not necessarily exist. Therefore, people often rely on some other solution concepts such as the Shapley value. The Shapley value has some nice properties. In particular, it always exists and it captures some degree of *fairness* (i.e. the payment of each agent is proportional to its marginal contribution in the group), moreover, its *fairness* comes with a price of sacrificing *stability*. We discuss the notion of fairness in detail in the next section. A solution using Shapley value is not necessarily in the core. Moreover, calculating Shapley value is very expensive. The core can be calculated in polynomial time, but the Shapley value can only be calculated in quasi-polynomial time. Thus, in our solution, we decide to focus on the core concept.

4.2.2 Core Stability

First, let there be N agents in the group, $(a_1, ..., a_n)$. Let the final allocation be A: thus if $a_i \in A$ then agent a_i is in the final allocation. The final allocation represents a mapping with the winning agents in the group and the payment the agents make for the items. Let $v(i), i \in N$ be the valuation of the bundle to each agent. we define the valuation of the group G to be:

$$V(G) = \sum_{i \in A} v(i) - P_G$$

where P_G equals to the total payment for the items the group has to submit to the center (i.e. the payment each agent pays for the item) and $\sum_{i \in A} v(i)$ represents the sum of total valuations for the winning agents. This value is obtained by the group mechanism running a VCG mechanism. For simplicity of notation, let us define $\sum_{i \in A} v(i)$ as $\mathcal{V}(A)$. If an agent is not in the final allocation, then the total payment it pays should equal to zero. Then $V(G)$ represents the earnings in the group and this amount should be distributed across the agents. We then define a payoff vector $x_1, ..., x_n$ such that $\sum_{i \in N} x_i = V(G), x_i \geq 0$. The payoff vector represents the amount of earning that each agent gets back (mathematically, it equals the valuation of the items of the agent minus the payment it makes for them).

Given the above definitions, the following property holds:

Proposition 2 *For each subset $S \subset G$, if we define:*

$$V(S) = \sum_{i \in S \cap A} v(i) - \sum_{i \in S \cap A} P_G * \frac{v(i)}{\sum_{i \in A} v(i)}$$

then the group is in the core if and only if the payoff vector $(x_1, ..., x_n)$ satisfies:

$$x_i = \begin{cases} 0 & i \not\subseteq A \\ V(G) * \frac{v(i)}{\mathcal{V}(A)} & i \in A \end{cases} \tag{4.2}$$

for every subset $S \subset G$.

And each agent's payment for its bid is then:

$$v(i) - x_i$$

Proof: First of all, we declare that if an agent is not in an allocation, then the payoff vector for it should be 0 and it will not make any payment either. We claim this is stable for the agent because if it does not get the final allocation, it means that even if it goes alone or joins another group, it will still not get the final allocation. However, by staying in the group it does make a saving on the total interaction cost incurred. Thus when an agent is not in the allocation, we claim the payoff of this agent is 0. And we then only need to look at the cases where an agent is part of the allocation.

By definition, to show that our payoff vector is in the core, we need to show that for every subset $S \subset G$,

$$\sum_{i \in S} x_i \geq V(S)$$

From our definition of $V(S)$, this means we need to show for every subset,

$$\sum_{i \in S} x_i \geq \sum_{i \in S \cap A} v(i) - \sum_{i \in S \cap A} P_G * \frac{v(i)}{\mathcal{V}(A)}$$

And we also have the constraint that $\sum_{i \in N} x_i = V(G), x_i \geq 0$.

Substituting the x_i's and expanding the left hand side of the equation, we get:

$$\sum_{i \in S} (\mathcal{V}(A) - P_G) * \frac{v(i)}{\mathcal{V}(A)} = \sum_{i \in S} v(i) - \sum_{i \in S} P_G * \frac{v(i)}{\mathcal{V}(A)}$$

And since if an agent is not in the allocation, it is not making any payments, then $\sum_{i \in S} P_G * \frac{v(i)}{\mathcal{V}(A)} = \sum_{i \in S \cap A} P_G * \frac{v(i)}{\mathcal{V}(A)}$. Similarly, $\sum_{i \in S} v(i) = \sum_{i \in S \cap A} v(i)$. This means the left hand side of our equation equals to the right hand side, and by definition, the payoff vector that gives us this solution is in the core.

Next we need to show that our claimed solution is the only solution under this circumstance.

To prove the uniqueness of our allocation vector, let us assume there exists some allocation vector which is not the same as the one given in the core above. Let us call this other solution $y = (y_1, ..., y_n)$. Since this solution is also in the core, the following equations are satisfied:

$$\sum y_i = V(G)$$

By definition, this means:

$$y_1 \geq v_1 - P_G \frac{v_1}{\mathcal{V}(A)} = x_1$$

$$y_2 \geq v_2 - P_G \frac{v_2}{\mathcal{V}(A)} = x_2$$

...

$$\sum_{i \in S} y_i \geq \sum_{i \in S} x_i - P_G \frac{\sum_{i \in S} x_i}{\mathcal{V}(A)}$$

since $x \neq y$, then it must be the case that $\exists y_a > v_a - P_G \frac{v_a}{\mathcal{V}(A)}$. Thus we have:

$$\sum_{i \in N} y = y_a + \sum_{j \neq a} y_j$$

$$\geq y_a + \sum_{j \neq a} v_j - P_G \frac{\sum_{j \neq a} v_j}{\mathcal{V}(A)}$$

$$> v_a - P_G \frac{v_a}{\mathcal{V}(A)} + \sum_{j \neq a} y_j$$

$$\geq y_a + \sum_{j \neq a} v_j - P_G \frac{\sum_{j \neq a} v_j}{\mathcal{V}(A)}$$

$$\geq y_a + \sum_{j \neq a} v_j - P_G \frac{\sum_{j \neq a} v_j}{\mathcal{V}(A)}$$

$$= \sum_{i \in N} v_a - P_G \frac{\sum_{i \in N} v_j}{\mathcal{V}(A)} = V(G)$$

Now we have showed that $\sum_{i \in N} y_i > V(G)$ which leads to a contradiction of the definition of the core. Thus the allocation vector we proved is the only vector that is in the core. □

The proof shows that we are able to get a core-stability solution. However, this particular solution is not *fair*. We discuss this in the next section.

4.2.3 Fairness

The word *fairness* means free from dishonesty, bias or injustice. In mechanism design, this term is used in a lot of models and the meaning of them are usually different. In our model, we use the notion of *coalition fairness* as defined below.

Definition 21 (Coalition Fairness) *Let $P(a_i)$ denote the payment of agent i if we use a standard VCG mechanism and every agent is going alone, and let v_i denote the valuation of agent i. We then say a payoff vector $(x_1, .., x_n)$ of a coalition G in our model is fair if, given the final allocation and the bid of the second highest price:*

- $v_i - x_i \leq P(a_i)$
- *If $P(a_i) = 0$, then there exist $\ell > 0$ such that the payment of the agent is $v_i - x_i - \ell$.*

In words, the above definition means:

- If an agent should win irrespective of whether it is in the group or going alone, he should pay at most the amount as though it is going alone.
- If an agent could have won with a price of zero by going alone, he should be compensated by the group in which it belongs.

To illustrate, let us consider an example to see why the core-stable payoff vector does not guarantee fairness.

Example 8 (Core-stable but not Fair payoff) *Let there be four agents A, B, C, D. where agents A, B and C form a group and agent D is in a group alone. Let agents A, B, C want items x, y, z with prices 4, 4, 2 respectively and agent D would like item x, z with a price of 5. From the VCG mechanism, we know the group will win items x, y, z with a price of 5.*

Now, from the coalitional games, we have the value of the group as following:

$$V(G) = 4 + 4 + 2 - 5 = 5$$

And all the possible subsets:

$$V(A,B) = 4 + 4 - \frac{4}{10} * 5 - \frac{4}{10} * 5 = 4$$

$$V(A,C) = 4 + 2 - \frac{4}{10} * 5 - \frac{2}{10} * 5 = 3$$

$$V(B,C) = 4 + 2 - \frac{4}{10} * 5 - \frac{2}{10} * 5 = 3$$

If we want allocation vector (x_A, x_B, x_C) *to be in the core, we need:*

- $x_A + x_B + x_C = 5$
- $x_A + x_B \geqslant 4$
- $x_A + x_C \geqslant 3$
- $x_B + x_C \geqslant 3$

Solving the above inequalities will give us the unique solution: $(2, 2, 1)$*. The payment each agent will need to submit is then* $(4-2, 4-2, 2-1) = (2, 2, 1)$.

However, let us we look at agent B in the group. If this agent was not inside any group and were to submit the bid to the center on his own, it would still win the item y*, but with a price of 0! This is because no one else is competing on this item with him, and by the definition of a VCG mechanism, the payment it has to pay equals zero. Thus, by being in the group, it is particularly unfair for this agent since now it has to pay a price of 2 dollars!*

In order to get fairness into our model, we need to modify our value function for the subsets as follows:

Let B denote the set of items in the second highest price bid. For each subset $S \subset G$, we define $V(S)$ to be:

$$V(S) = \begin{cases} 0 & S \cap B \neq \emptyset \& \sum_{i \in S \cap A} v(i) < P_G \\ \sum_{i \in S \cap A} v(i) - P_G & S \cap B \neq \emptyset \& \sum_{i \in S \cap A} v(i) > P_G \\ \sum_{i \in S \cap A} v(i) - 0 & S \cap B = \emptyset \end{cases} \quad (4.3)$$

In words, it means that the value for the subset is equal to the sum of the valuations for the agents in the set, minus the price that the group should pay if this subset was competing with and winning over the second highest price bid. If the subset could not win over the winning bid then the value of the subset is zero. If on the other hand, the subset does not contain common items from the second highest price bid, the value of the subset equals to the sum of the valuations of the members.

Let us see how this second definition brings us fairness:

Example 9 (Fair payoff Example) *Consider the same setting as the previous example, we still have the same* $V(G) = 5$ *and the constraint* $x_A + x_B + x_C = 5$*. However, the valuations for the subsets now become:*

$$V(A, B) = 4 + 4 - 5 = 3$$

$$V(A, C) = 4 + 2 - 5 = 1$$

$$V(B, C) = 4 + 2 - 5 = 1$$

Thus, we need:

- $x_A + x_B + x_C = 5$
- $x_A + x_B \geqslant 3$

- $x_A + x_C \geqslant 1$
- $x_B + x_C \geqslant 1$

There are multiple allocations in the case, one possible allocation is then $(1, 3, 1)$*, which gives each agent's payment as* $(4 - 1, 4 - 3, 4 - 1) = (3, 1, 3)$*. As we can see, this solution is fair since agent A and C are not paying more than they should have when they're going alone and agent B gets compensated by being in the group. However, as we can see, there are multiple solutions to this question: the core is not unique.*

4.2.4 An Impossibility Result

Unfortunately, the solution above, although it solved the fairness issue, will not guarantee the existence of a core.

Example 10 (Fair payoff with empty-core example) *Let there be four agents* A, B, C, D *with agents* A, B *and* C *in the same group. Let agents* A, B, C, D *want items* x, y, z, y *with valuations* $4, 2, 1, 4$ *respectively. In this example, agents* A, B *will win items* x, y *with a price of 6. From the fair payoff algorithm, we can calculate* $V(G) = 4{+}2{+}1{-}4 = 3$ *and the constraint* $x_A + x_B + x_C = 2$*. And the valuation of the subsets are:*

$$V(A, B) = 4 + 2 - 4 = 2$$

$$V(A, C) = 4 + 2 - 0 = 6$$

$$V(B, C) = 0$$

Thus, we need:

- $x_A + x_B + x_C = 3$

- $x_A + x_B \geq 2$
- $x_A + x_C \geq 6$
- $x_B + x_C \geq 0$

However, since $x_A + x_B + x_C = 3$*,* $x_A + x_C \geq 6$ *is not possible! Thus in this scenario, a core-stable solution does not exist.*

Proposition 3 *It is not possible to guarantee both fairness and stability for reward-distribution in our coalitions.*

Proof: As we have shown, the core-stable payoff algorithm always gives us an unique payoff vector that is in the core, but it is not *fair*. On the other hand, there is no guarantee that the fairness payoff algorithm always has a core-stable solution. Thus it is impossible to guarantee both fairness and stability in our model. □

Now we have shown that our first set of solutions is in the core, but not *fair*, while our second set of solutions is *fair*, but a core does not necessarily exist. It is intuitive to ask which solution should we use? We think the answer is application-dependent. A lot of time, stability has to be sacrificed to get fairness. For example, the very popular solution concept, Shapley Value, provides fairness at a price of decreasing stability. In a setting where self-interested agents may have access to more information, the fair algorithm should be preferred. For example, if agents in our setting are able to talk to agents in other groups, then they may deviate if we are using the core-stable solution because it is not fair to them. To better understand the difference between these two payoff algorithms, we carry out experiments in the next section.

4.3 Experiments

In order to understand the effects on agents when using a stable protocol compared to a fair protocol, we carry out a series of experiments. We first would like to see the effect on each agent's payment with no grouping compared to the two algorithms (Core-stable v.s. Fairness). Then we would also like to see the differences in each agent's payment using the two algorithms.

4.3.1 Setup

For simplicity, we fix our settings to test a very simple scenario. We have five agents and four items to choose from. Each agent randomly picks one out of the four items. Since there are five agents and four items, this makes an overlap very likely to occur. Each agent has a value for the item, drawn uniformly from the interval [1,10]. We then divided the agents into two groups (a group of three and a group of two) and then ran the regular VCG mechanism, the core-stability algorithm and the fair algorithm respectively. Each experiment is carried out 10000 times. We ignored the cases when the fairness payoff algorithm cannot generate a core-stable solution and only looked at the cases where both algorithms had a core-stable solution.

4.3.2 Results

No Grouping v.s. Core-stable Grouping

The first set of experiments compares the total payment of each agent with no grouping (the regular VCG with 5 agents) and core-stable grouping where we divide the payment by fraction among the winning group members.

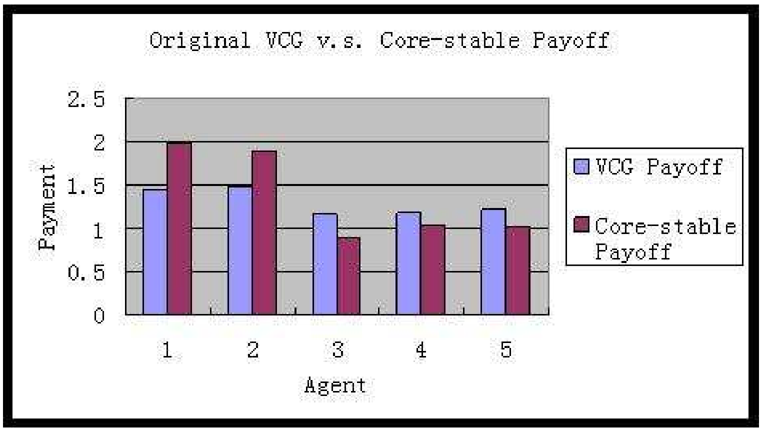

Figure 4.3: Payment comparison for each individual agent using original VCG and our core-stable mechanism

Figure 4.3 suggests that the core-stable grouping method is beneficial only for certain agents, but not every agent. This is expected because forming coalitions changed the final outcome and as discussed before, sometimes by doing this an agent has to pay more than what it would have paid originally. However, if we calculate the total payment the agents had to make for the two methods, they are roughly the same. This means that the algorithm redistributes some of the payments among agents but does not add to the total payment. The total payment that the center collects is the same.

Core-stable Grouping v.s. Fair Grouping

Next we look at the comparison of the core-stable payoff method and the fair payoff method. Of the 10000 runs, we specifically look for an agent who would have paid 0 if it had participated in a standard VCG mechanism, then collected the data on these runs and looked at the payment for each agent. Of the 10000 runs, there are 1652 cases where the agent we picked (Agent 2) falls in this category.

We can see from Figure 4.4, as expected, the total payment agent 2 made in the fair

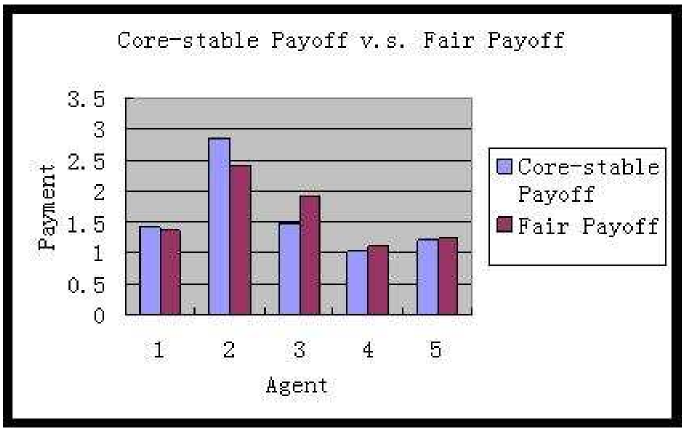

Figure 4.4: Payment comparison for each individual agent using the core-stable and fair mechanisms

scheme is less than in the stable scheme. If we take a closer look, we can see that the total payment still equals roughly the same for the two algorithms. So again, our algorithm mainly redistributes the payment among the agents. From the center's point of view, it collects the same amount no matter which algorithm the agents follow. This is a nice property because we do not want the center to be able to take advantage of the strategies that agents follow.

So far we have experimented using different payoff solutions. It is clear that no matter which payoff we choose, the total amount of payment does not change since the payoff method does not change the final allocation. It is also clear that some agents benefit from the solution while some agents have to pay more. Unfortunately, it is unclear to us at this point as in which agents benefit more from which solution. Clearly, each approach has its own advantages and disadvantages. Thus we think the answer is application dependent. The system designer should decide which property (stability or fairness) is more important in the specific application.

Table 4.1: Highlights of Two Scenarios

Application:	Information Provision Network	Combinatorial Auction
Valuation Function:	Substitutes	Non-substitutes
Group Level Payment:	VCG + Redistribution	VCG + XOR group bids
In-Group Payment:	VCG + Redistribution	Core-Stable or Fair Payoff

4.4 Summary

In this chapter, we provided a detailed discussion on the mechanisms used for reward-distribution. We looked at two applications. We made an assumption in the information-provision network that information could be discretized into *areas* and an agent's value of a set of information areas is evenly distributed. Given this assumption, we could apply the VCG mechanism on the areas with or without overlap separately. For the areas with overlap, we then proposed a redistribution algorithm to redistribute some of the extra transfer that the center collects. We then looked at the problem from a coalitional games perspective for the combinatorial auction setting. We proposed a reward-redistribution method that is core-stable. We showed that this core-stable method is not *fair* and proposed a fair solution. However, the fair solution is not necessarily stable. We then carried out experiments to look at the payment redistribution of each individual agent in the group using different redistribution schemes.

In summary, Table 4.1 highlights the key properties of the two scenarios and our model can be applied to any application that fits into the generalization of either scenario. The valuation function in the table is the key difference of the two scenarios. It is the underlying valuation function of the agents that determines which scenario an application fits in. For example, if we have an auction scenario where there is just one type of item and an agent's utility is associated with the number of items it gets but not which item it gets (i.e. items are substitutes), then this setting would fit into the generalization of

’information provision network’ scenario even though it is an ’auction’. The group level payment entry lists how payment is redistributed at group level and the in-group payment entry lists how payment is redistributed inside groups. When there is a none-substitutes valuation function, there is not a generic answer as to which in-group payment approach to take (core-stable or fair). We think it is up to the system designer to decide which property is more important in a specific application.

Chapter 5

Cost-Sharing Protocol

In this chapter we introduce our cost-sharing protocol and group-formation algorithms. The cost-sharing protocol is held by a trusted third party to whom each agent can reveal its cost. Once a group has been formed, our cost-sharing protocol picks the agent to interact with the center and handles the cost-distribution among group members. Our grouping strategy, on the other hand, allows agents to form groups in the first place. We discuss and prove the properties of these protocols and discuss our findings from series of experiments.

5.1 The Model

As we recall, our model consists of the following parts:

- A central entity, denoted E
- A set of n self-interested agents, with quasi-linear utility functions
- Interaction cost from agent i to the central entity, denoted as C_i

- Interaction cost between agents i, j (*inter-agent interaction cost*), denoted as $IC_{i,j}$

Our goal is to design a protocol that is truthful, individually rational and allows agents to reduce interaction costs by forming groups. In this chapter, we will focus on the last part. Namely, we study how agents can share the interaction cost in order to reduce their costs overall. The rest of the chapter consists of three sections. In the first section we present our cost-sharing protocol and in the second section we discuss the group-formation algorithms. In the last section we discuss our experiments and analyze the results.

5.2 Cost-sharing Protocol

One intuitive question to ask is: Why do we need a cost-sharing protocol? Recall the combinatorial auction example in Chapter Three. Agents located in the same geographic area form groups in order to save on the total interaction cost. In our model, agents have a common goal: *save cost*. Forming groups and sharing total cost allows them to do this. Once they form a group, they must cover the total cost spent as a group, and thus must decide how to divide and share this cost. Since our agents are self-interested utility-maximizers, a simple cost-sharing protocol is not trivial. Thus we need to develop protocols that allow agents to share the costs and not at the expense of others in the system.

5.2.1 Goals

There are several features of our problem setting which make cost-sharing an interesting and challenging problem. First, the agents are self-interested and so will take actions which will benefit themselves, possibly at the expense of others in the system. Second, there are hard constraints faced by the agents. In particular, if the group incurs a cost, then the resources to pay for the cost *must* come from the group. Finally, we would like a cost-sharing protocol to be *fair*, in that the amount that agents contribute to the group somehow reflect the benefit the agents are receiving from the group. In the rest of this section we provide more details about these desired properties.

Since the agents are self-interested, the appropriate incentives must exist so that they will participate in a cost-sharing protocol. Agents need incentives to participate in the first place. This means that the cost they must pay by participating in the protocol must not be higher than the cost they would pay by acting alone. Let C_i be agent i's cost to interact to the center alone and let $\mathfrak{C}_\mathrm{i}$ denote agent i's cost inside a group. The constraint we need is $\mathfrak{C}_\mathrm{i} < C_i$. We call this the participation constraint or the individual rationality constraint.

The next goal is to achieve truthfulness. This property, although not considered in a lot of popular cost-sharing schemes, is very important to us. The reason is in our setting, the interaction cost to the center is an unknown variable and since our agents are self-interested, it is possible and likely that agents will manipulate this cost in order to make a profit. To prevent this situation from happening, we need a cost-sharing algorithm that is truthful, i.e., agents cannot manipulate the interaction cost by themselves and benefit.

Definition 22 (Truthfulness) *Let C_i denote the actual interaction cost to the center for agent i and let C'_i be agent i's reported cost ($C'_i \neq C_i$). Let $\mathcal{C}_i$ denote agent i's share in the group if it declared its cost truthfully and let $\mathcal{C}'_i$ denote agent i's share in the group if it declares its cost to be C'_i. We say a cost-sharing algorithm is truthful (incentive compatible) if an agent will not pay a smaller share of the total cost (and as a result,* *increase its own utility) by lying:*

$$\mathcal{C}'_i \geq \mathcal{C}_i \tag{5.1}$$

The next property we desire is what we call *cost recovery*. This is desirable since we have to collect enough money to cover the cost of the group, but on the other hand we do not want extra money floating around, decreasing overall social welfare.

Definition 23 (Cost-Recovery) *A protocol is a cost-recovery protocol if the amount it collects from the agents is exactly the amount needed to cover the costs. Let $\mathcal{C}$ be the total interaction costs of the group, including the cost incurred from interacting with the central entity. Let $\mathcal{C}_i$ be the amount agent i is charged by the cost-sharing protocol. Then if $\sum \mathcal{C}_i = \mathcal{C}$, the protocol is a cost-recovering protocol.*

The final property we desire is *fairness*, which we define in terms of the benefit a particular agent receives by being in a group. Let $C_i, i \in N$ be the interaction cost from agent i to the central mechanism. Let $\mathfrak{C}_i, i \in N$ denote the share of cost each agent pays inside the group. We denote the *benefit* of an agent as $\mathcal{B}_i = \alpha C_i, \alpha > 0$, i.e. the benefit an agent receives is proportional to its interaction cost to the center. After a group is formed, the interaction agent is chosen to be the one with lowest interaction cost to the center and this cost will be shared among the group members. As we recall, we assume that the interaction cost between agents is much smaller comparing to the interaction cost for an agent to communicate to the center thus we neglect it here. As a result, agents with higher interaction cost originally make more savings than agents with lower interaction cost. This is why our benefit depends on the original interaction cost of an agent, and the higher the original cost is, the larger the benefit is.

Definition 24 (Fairness) *We say a cost-sharing algorithm is fair to an agent i if, for all agent $j \neq i$:*

If $\mathcal{B}_j > \mathcal{B}_i$*, then* $\mathfrak{C}_j \geq \mathfrak{C}_i$

In words, this means that if an agent obtains a *higher* benefit, it cannot pay less than an agent with a lower benefit. We describe this property last because while it is desirable, without it, we will still have enough nice properties for our models, since the cost-recovery and truthfulness properties alone can result in an interesting cost-sharing protocol. Thus, we will try to achieve fairness, but not at the expense of the other properties. We think that in real-world scenarios, fairness can actually enhance stability. For example, if an agent with a very low benefit observes that some other agent in another group with similar benefit is paying a much smaller amount, or some agent in its group with a much higher benefit is paying a similar share, it will give the agent incentive to leave the group. In our model, we use the agent with the lowest interaction cost to the center as the communicating agent. Thus we want to give this agent a degree of fairness so that he will not have incentive to leave the group.

5.2.2 Existing cost-sharing protocols

Cost sharing has been a topic of interest across many areas. Moulin *et al.* has proposed cost-sharing algorithms in economics [19, 21]. Roughgarden *et al.* has looked at the problem in the networks [5, 1]. There have been many proposals for cost-sharing protocols. In this section we describe the two standard ones. One of the simplest cost-sharing protocol is derived from the Shapley value, and is sometimes called *equal charge cost-sharing* [10]. This rule states that given some fixed cost C, every agent pays an equal share (i.e. $\frac{C}{n}$ if there are n agents). While this protocol clearly satisfies our cost-recovery property, it is far from our definition of fairness.

Example 11 (Shapley value cost-sharing) *Let us refer back to our information provision scenario. Suppose there are three agents* (A, B, C) *who can talk to each other. Let* *us assume the agents can talk to the center with a one time fixed-cost of* $6, 10, 50$ *respectively. For simplicity, let us omit the inter-agent interaction cost here. Suppose the three agents decide to group together and send all the information through agent* A*, with a cost of* 6*. Now if the agents use a Shapley value cost-sharing algorithm, then each agent will pay* $\frac{6}{3} = 2$*. Clearly, agent* C *is making a much bigger benefit than the rest of the agents and thus this algorithm is not fair.*

Another standard cost-sharing protocol is the Moulin-Shenker serial cost-sharing algorithm [21]. In this protocol, a fixed group of agents share one input, and one output technology with decreasing returns. For example, a group of people all want to buy cherries through some vendor located far away. They then have to share the cost of shipping cherries. However, they have decreasing returns as in that although they want cherries, as they get more, they are not as happy as before because if they cannot finish them, the fruit will go bad. Each agent announces its demand q_i of output. The cost function is denoted by C so the system must share a total cost of $C(\sum_i q_i)$ among the n agents. The cost-sharing formula goes as follows: Agent 1 with the lowest demand q_1 pays $\frac{1}{n}$th cost of nq_1. Agent 2, with second lowest demand q_2 pays agent 1's cost share plus $\frac{1}{n-1}$th of the incremental cost from nq_1 to $(n-1)q_2 + q_1$. Similarly, agent 3 pays agent 2's share plus $\frac{1}{n-2}$th of the incremental cost from $(n-1)q_2 + q_1$ to $(n-2)q_3 + q_2 + q_1$. Thus the cost share depends anonymously upon demands and an agent's cost share is independent of demands higher than its own. Formally, let $\xi_i(C, q)$ denote the cost for agent i with cost function C and demand q, then:

$$\xi_i(C, q) = \frac{C(q_i)}{n - i + 1} - \sum_{k=1}^{i-1} \frac{C(q_k)}{(n - k + 1)(n - k)}$$

Example 12 (Moulin-Shenker Example) *Let us go back to the cherry example. Assume the cost function for shipping cherry is one for one up to ten pounds of goods and* *ten for one afterwards, i.e.:*

$$C(q) = q, q \leq 10$$

$$C(q) = 10q - 90, q > 10$$

Let there be three agents with demand $q_1 = 3, q_2 = 6, q_3 = 10$*, respectively. Thus, from the definition, agent 1 will pay:*

$$C(\frac{1}{n} * nq_1) = C(\frac{1}{3} * 3 * 3) = C(3) = 3$$

Agent 2 would pay agent 1's cost, plus $\frac{1}{n-1}$*th of the incremental cost from* nq_1 *to* $(n-1)q_2 + q_1$*, which equals:*

$$3 + \frac{1}{3-1} * (C((3-1)*6+3) - C(3*3)) = 3 + \frac{1}{2} * (C(15) - C(9)) = 3 + 25.5 = 28.5$$

Similarly, we can get agent 3's share of cost, which equals to the remaining balance, 68.5.

The Moulin-Shenker protocol recovers cost since the sum of all shares is C. It is also fair according to our definition since agents with higher demand pay more. In addition, it has several other nice properties. First, it is anonymous (the names of the agents do not matter). Second, it is monotonic (an agent's share increases as it demands more output). Third, it is smooth (an agent's cost share is a continuously differentiable function of the vector of demands), and lastly, it is a Nash equilibrium profile (the result of the protocol gives a unique Nash equilibrium). Finally it has been shown that it is the unique cost-sharing protocol with these properties. Unfortunately, it is not well suited to our domain. In its setting, the declared demand q_i are assumed to be given and cannot be manipulated by agents. This is not true in our model since the interaction cost from an agent to the center is unknown. If we were to use this scheme, the demand q_i would be replaced by a variation of the interaction cost to center C_i. However, we could not guarantee that agents report their interaction cost truthfully. If agents were able lie about this cost (e.g, declaring a lower cost), they could make a profit. Also, we cannot guarantee a convex cost function, which is critical in the serial cost-sharing protocol. Another point is that the group is considered fixed in this protocol and the goal is not on saving the total cost incurred. For example, if we look at the cherry shipping example above, although the protocol is fair, the most reasonable solution is for each agent to order and ship alone! Thus, giving the above arguments, we conclude this protocol is not suitable for our model.

Having said the above, we conclude that none of the existing cost-sharing algorithms can best reflect our model, thus we propose our own cost-sharing algorithm in the next section.

5.2.3 Our Cost-Sharing Protocol

In this section we introduce our proposed cost-sharing protocol.

Definition 25 (Cost-Sharing Protocol) *Assume each agent i announces that its interaction cost with the center is C'_i, assume its interaction cost with the group, g, is IC_i^g and this is publicly known by group members. Assume $C'_1 < C'_2 < \ldots < C'_n$. Thus C'_1 is the interaction cost to the center that needs to be covered. Let $IC = \sum_{i=1}^{n} IC_i'^g$. The share that each agent has to pay is:*

$$\mathfrak{C}_i = \mathcal{C}^i + \frac{IC}{n}$$

where

$$\mathcal{C}^i = \begin{cases} \frac{C_1'}{n+1}, & i = 1 \\ \frac{C_1' - \mathcal{C}^1}{n-1}, & 2 \leq i \leq n \end{cases}$$

In words, the cost-share of each agent can be divided into two parts, the share of cost to interact with the center ($\mathcal{C}^i$), and the share of cost to interact with each other ($\frac{IC}{n}$). We declare the share of interaction cost between agents to be divided equally among the members since each agent is involved and has equal importance in this cost. For the interaction cost to the center, we declare the interaction agent to pay a lower cost while all other agents are paying a higher cost. We argue the above cost-sharing algorithm can easily be used with different cost functions. In this thesis we study two basic cost functions. First is the fixed cost function, such as a one-time fee charge, or a flat shipping rate. The other typical cost function we will study is a unit-cost function. In a unit cost function, the total cost = (per unit cost) x (number of units). Our protocol can be used in a fixed-cost scheme without any modification. For a unit-cost scheme, let I denote the total number of units for the group, each agent will pay:

$$\mathcal{C}^i \times I + \frac{IC}{n}$$

In our protocol, the inter-agent interaction cost within a group is equally shared among the agents. For the interaction cost to the center, the interacting agent is paying the lowest amount and all the other agents are paying a higher value. To prove this, we can see that the agent with lowest cost is paying a share of:

$$\frac{C_1'}{n+1}$$

And all the other agents are paying:

$$\frac{C_1' - \frac{C_1'}{n+1}}{n-1}$$

Comparing these two equations, we have:

$$\frac{C_1' - \frac{C_1'}{n+1}}{n-1} - \frac{C_1'}{n+1} = \frac{(n+1)C_1' - C_1' - (n-1)C_1'}{(n+1)(n-1)} = \frac{C_1'}{(n+1)(n-1)} > 0$$

And this shows that every other agent is paying an amount higher than what the agent with lowest interacting cost pays.

To get a better understanding of the protocol, let us present an example:

Example 13 *Let there be three agents (n=3) A, B, C, in a fixed-cost line topology A-B-C. Let the inter-agent interaction cost between AB and BC equal to one. Let* $C_A = 6$, $C_B = 3$ *and* $C_C = 8$. *Since agent B has the lowest cost to the center, it will become the agent that interacts with the center. The total inter-agent interaction cost is 2 (1 each for agents A and C to interact with B). Thus the total cost needs to be spent for the group is* $3 + 2 = 5$. *Now let us calculate each agent's cost using the above algorithm, if everyone truthfully report their costs, we have:*

$$\mathcal{C}^B = \frac{3}{4} + \frac{2}{3} = \frac{17}{12} \approx 1.42$$

$$\mathcal{C}^A = \mathcal{C}^C = \frac{3 - \frac{3}{4}}{2} + \frac{2}{3} = \frac{43}{24} \approx 1.79.$$

The total cost agents pay is $\frac{17}{12} + 2 \times \frac{43}{24} = 5 = 3 + 2$.

5.2.4 Properties

In this section we discuss the properties of our protocol.

Proposition 4 *The cost-sharing protocol is individually rational if the total interaction cost for the group IC satisfies $IC < min(\frac{n^2}{n+1}C_1, nC_i - \frac{n^2}{n^2-1}C_1)$.*

Proof: Let IC denote the total inter-agent interaction cost for a group of n agents. Let the original interaction cost to the center of these agents be $C_1, ..., C_n$, respectively, with $C_1 < ... < C_n$. To interact as a group, the agents will each pay a share of the inter-agent interaction cost as well as a share of the cost to interact to center (C_1, in this case). To get individual rationality, we need the following equations to hold:

$$\frac{C_1}{n+1} + \frac{IC}{n} < C_1 \tag{5.2}$$

$$\frac{C_1 - \frac{C_1}{n+1}}{n-1} + \frac{IC}{n} < C_i \tag{5.3}$$

The first equation needs to hold for agent 1, who is paying the least share, the second equation needs to hold for any other agent. Working out the two equations, we get:

$$IC < \frac{n^2}{n+1}C_1 \tag{5.4}$$

and

$$IC < nC_i - \frac{n^2}{n^2-1}C_1 \tag{5.5}$$

In order for the cost-sharing algorithm to be individually rational, both of the equations have to be satisfied. Thus the total interaction cost IC has to be the minimum of $\frac{n^2}{n+1}C_1$ and $nC_i - \frac{n^2}{n^2-1}C_1$. □

The individual rationality of our cost-sharing protocol does come with a constraint. However, we assume that the inter-agent interaction cost is much smaller than the cost to interact to the center alone in most of the cases, since otherwise it does not make sense for agents to form groups at all. And as we can see, the constraint asks for $IC < min(\frac{n^2}{n+1}C_1, nC_i - \frac{n^2}{n^2-1}C_1)$. For example, when $n = 2$, $\frac{n^2}{n+1}C_1 = \frac{4}{3}C_1$ and $nC_i - \frac{n^2}{n^2-1}C_1) = 2C_i - \frac{4}{3}C_1 > 2C_1 - \frac{4}{3}C_1 > \frac{2}{3}C_1$. So in this case, the minimum constraint is $\frac{2}{3}C_1$. However, from our assumption, IC is usually much less than this. This shows that although there is a constraint, most of the time it will be satisfied. Although the individual rationality for our cost-sharing protocol has a constraint, since our grouping formation algorithm (which we will talk about in the next section) is individually rational, we still get our desired outcome in our model.

Proposition 5 *The cost-sharing protocol is a cost-recovery protocol.*

Proof: From the scheme, we can see that the total cost the agents pay is equal to:

$$(n-1) \times \frac{C_1 - \mathcal{C}^1}{n-1} + \mathcal{C}^1 + \frac{IC}{n} \times n = C_1 + IC$$

Thus the mechanism collects $C_1 + IC$, which is exactly what needs to be spent. The mechanism makes no profit or loss, it covers exactly the amount that needs to be covered. No agent has paid any extra amount. Thus we say this mechanism is 'cost-recovery'. □

Proposition 6 *The cost-sharing scheme is incentive-compatible.*

Proof: To show that our scheme is incentive-compatible, we need to show that agents are best off telling the truth about their interaction costs. Let the actual lowest cost be C_1'. This proof is done in three parts. First of all, we can see that no agent will want to lie by announcing a cost lower than C_1'. This is because even if an agent lies to have a lower cost $C_i' < C_1'$, when interacting, he will incur a cost C_i,which is greater than C_i'. Thus the difference $C_i - C_i'$ has to be paid by this agent itself and it has to do the extra work. Next, it is obvious that the cost-sharing scheme only depends on the number of agents and the agent with lowest cost C_1'. This means for all the other agents, lying by having a cost C_i', such that $C_i' \neq C_i$ and $C_i' > C_1'$ is not going to make a difference. This shows for all agents except agent with cost C_1', lying does not make them a profit.

Now let us look at the agent with the lowest cost, agent 1. We have already showed that by lying to have a lower cost, it is not going to make a profit. Assume that agent 1 announces a cost $\hat{C}_1 > C_2 > C_1$. Then, its share would be

$$\hat{\mathcal{C}}^1 = \frac{C_2 - \mathcal{C}^2}{n-1} + \frac{IC}{n},$$

as opposed to

$$\mathcal{C}^1 = \frac{C_1}{n+1} + \frac{IC}{n}$$

if it has declared its cost truthfully.

We need to show that $\hat{\mathcal{C}}^1 \geq \mathcal{C}^1$. Assume that $\hat{\mathcal{C}}^1 < \mathcal{C}^1$. That is

$$\frac{C_2 - \mathcal{C}^2}{n-1} + \frac{IC}{n} \leq \frac{C_1}{n+1} + \frac{IC}{n}$$

Since C_2 is the lowest cost now, $\mathcal{C}^2$ equals to the share of the communicating agent, i.e.$\mathcal{C}^2 = \frac{C_2}{n+1}$. Thus our inequality becomes:

$$\frac{C_2 - \frac{C_2}{n+1}}{n-1} + \frac{IC}{n} \leq \frac{C_1}{n+1} + \frac{IC}{n}$$

Further simplification gives us:

$$C_2(n+1) - C_2 \leq C_1(n-1)$$

or

$$n(C_2 - C_1) + C_1 \leq 0.$$

This is impossible since $C_1 \geq 0$ and $C_2 > C_1$. Therefore, we have a contradiction, and so agent 1 is worse off declaring an interaction cost that is greater than C_2. Thus now we have showed that none of the agents will make profit by lying, *i.e.* the scheme is incentive-compatible. □

Proposition 7 *Our cost-share protocol is fair to every agent in the group.*

Proof: From the definition of fairness, for a group of n agents, we can write out their benefits by: $\mathfrak{B}_i = \alpha C_i$, where $\alpha > 0, C_i$ is the interaction cost for this agent to communicate to center, and $\mathfrak{C}_i$ is the total cost this agent pays inside the group. Since we assume $C_1 < C_2 < ... < C_n$, this means $\mathfrak{B}_1 < \mathfrak{B}_2 < ... < \mathfrak{B}_n$. From our cost-sharing protocol, we know that $\mathfrak{C}_1 < \mathfrak{C}_2 = \mathfrak{C}_3 = ... = \mathfrak{C}_n$. This gives us that for every $i, j \in (1, ..., n)$, if $\mathfrak{B}_i < \mathfrak{B}_j$, then $\mathfrak{C}_i \leq \mathfrak{C}_j$. This satisfies our definition of fairness, and thus our cost-sharing protocol is fair to everyone.

□

We designed our protocol to distinguish only the interacting agent and the rest of the group on purpose. Our reason is that the absence of any other agent will not result in much difference whereas the absence of the agent with lowest interaction cost may result in changing the whole structure of the group. Since if the lowest cost agent does not exist, it may not be beneficial for the agents to be in that group at all! Thus we made sure in our protocol that the agent with lowest cost is indeed paying the smallest share so that he will not have incentive to leave the group.

5.3 Grouping Strategy

So far in this chapter we have talked about how costs would be shared once a group has been formed. We have not, however, discussed how the group would be formed in the first place. In this section we discuss group formation strategies. We first define some properties we wish our group formation strategy will exhibit. We then propose a group-formation algorithm that satisfies these properties. We analyze its performance on different interaction topologies and then propose heuristics that can improve its performance.

5.3.1 Properties

We propose a greedy group-formation strategy for agents to use when deciding which other agents to coordinate with. There are two properties we desire in our strategy. First, we want a strategy that only forms groups when all the agents are better off (or at least not worse off) than if they acted alone. That is, we want an *individually-rational* grouping strategy. This is essential since our agents are self-interested. Second, we want our groups to satisfy, at minimum, a weak stability property.

Definition 26 (Locally Stable) *A group is locally-stable if no subgroup of agents can benefit by leaving the group and acting on their own.*

Our local-stability property is similar to the notion of *group-strategy-proof* but is weaker since we require only that agents are better inside a group compared to acting independently.

Definition 27 (Group-Strategy-proof) *A protocol is group-strategy-proof if for any subset $S \subset N, u(S)$ is maximized. That is, no subsets of agents in the group will want to deviate and form a group on their own.*

5.3.2 Grouping Algorithm

Our grouping strategy works as follows. First, each agent creates a list of its neighbours and the inter-agent interaction costs. Then the algorithm picks an agent randomly and this agent then scans the list and contacts the neighbours starting with the lowest interaction cost. If there is a mutual advantage to coordinate, the two agents form a group. Let C_i, C_j be the interaction cost from agents i, j to the center respectively, and let $\mathfrak{C}_i, \mathfrak{C}_j$ be the interaction cost agent i and j would pay if these two form a group. Then the two agents will form a group if and only if $\mathfrak{C}_i < C_i$ and $\mathfrak{C}_j < C_j$. That is, they will form a group if the cost shares for both of them are less than their cost of interacting with the center alone. After a group is formed, the group's neighbour agents will automatically update their neighbour list to include the group. The algorithm then randomly picks another agent who will do the same. If the group is contacted by another agent i, the group will let i join only if two conditions are met:

1. The cost shares of the agents do not increase. Agents will not let another agent enter their group, if the cost increases for all agents. This tends to occur whenever agent i's addition to the group would increase the total inter-agent interaction cost, without decreasing the interaction cost with the mechanism.

2. The internal social welfare in the group does not decrease by the introduction of the new agent, as calculated by the group mechanism.

If an agent is rejected by a group, then it must either act on its own or contact another group or individual agent.

Here is the pseudo-code for our algorithm:

```
create agent list with all agents;

while(there exist an agent i inthe agent list) {
    if(i has neighbour)
    {
        Agent* bestChoice = i.neighbours.at(0);
        for(int j =1; j < i.neighbours.size();j++)
        {
            if(calculate_benefit(i,i.neighbours.at(j) >
            calculate_benefit(i,bestChoice))
            {
                bestChoice = i.neighbours.at(j);
            }
            form_group(i,bestChoice);
            update_neighbour_lists;
            remove i,bestChoice from agent list
        }
    }
    else
    {
        i forms a group on its own
        remove i from agent list
    }
}
```

Our proposed grouping strategy has the following properties:

Proposition 8 *The grouping strategy is individually rational.*

Proof: From the strategy, no agent would join a group which would make it worse off compared to if it was acting by itself. This means for every agent in the group, it is getting better utility compared to going alone, thus our grouping strategy is individually rational. □

Proposition 9 *The grouping strategy is locally-stable.*

Proof: First of all, we already proved the strategy is individually rational, meaning once in a group, an agent will not want to deviate by leaving the group and acting on its own. Also, after a group is formed, it will only let new agents join if the social welfare of the existing members does not decrease. Thus in terms of an individual agent, its own utility always improves by the introduction of a new agent. This means that once in a group, an agent will never find itself in a worse situation than when it joined. This results in the group being locally-stable, since once in a group, an agent is always better off staying in the group. □

5.3.3 Interaction Topology

As mentioned in the previous chapter, one aspect that we are interested in studying is the *interaction topology*.

Definition 28 (Interaction Topology) *The interaction topology is a graph where each node represents an agent, and edges between nodes indicate that two agents can interact with each other.*

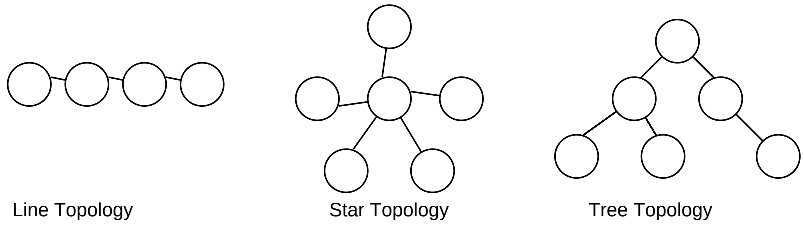

Figure 5.1: An example of three basic interaction topolgoies: Line, Star and Tree

In particular, we are interested in studying if our grouping strategy is topology independent. In other words, is the outcome the same no matter what topologies we use? If not, then why is it happening and is there a common pattern in terms of the cost savings? Having these questions in mind, we analyze the savings of the total interaction cost using our proposed grouping strategy. We start with the three basic topologies; a line topology, a star topology and a tree topology as shown in Figure 5.1.

Line Topology

The line topology has the simplest structure of the three topologies we study in this section. Each agent (except the first/last agent) would have exactly two neighbours that it can communicate with. Thus essentially, each agent has approximately the same importance in the topology. Thus we conclude the current group formation strategy in random order serves the line topology the best and we will focus on improvement with the other topologies.

Star Topology

In a star topology, only the center agent is able to contact other agents. Thus, if our grouping starts with any other agent, the only choice for this agent is to communicate to

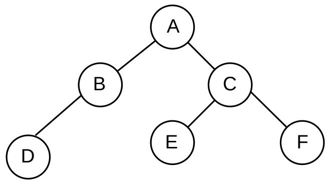

Figure 5.2: An example scenario of Tree Topology

the center agent. This will give the center agent limited choices. Thus it will make more sense if we actually start with the center agent so that this agent will have the full set of choices. Since the neighbour agents do not really have another option, it is very easy for them to make decisions. Additionally, since our goal is to maximize social welfare, it also makes sense to include agents with higher interaction cost to the center in the group rather than having these agents act by themselves. This means we should give priorities to agents with higher interaction cost to the center.

Tree Topology

The tree topology has the most complicated structure among the three topologies we study. Initially, we predict that highest social welfare should be achieved in this topology amongst the three. To illustrate, let us consider the following example:

In Figure 5.2, we have six agents (Agent A - F) in the system. Let us suppose that randomly, we start grouping with agent A. Agent A would go ahead and contact its neighbours (B and C) and would eventually form a group with agent B and agent C. If the grouping algorithm stopped here, it would isolate agents D, E and F, forcing them to act in their own. If on the other hand, we start from the leaf nodes, agent D and E, then we may end up with only two groups, one for agents A, B, D and another for agents C, E, F, and as a result, increase the social welfare. This is the aspect that our current grouping strategy does not take advantage of.

5.3.4 Heuristics

While our grouping algorithm is very general, by adding heuristics that depend on the interaction topology, we can improve performance. First of all, we modify the strategy to have a specific order of communication. We sort the agents from highest interaction cost to lowest and then start the grouping with the first agent, i.e. the agent with highest interaction cost. The reason is that by doing so, we make sure that agents with higher interaction cost are in the group so that we can improve the social welfare. To illustrate, let us consider the following example:

Example 14 *let there be three agents* A, B, C *and let the interaction from these agents to the center be* $2, 3, 100$ *respectively. Let* $IC_{A,B} = 2.5$*,* $IC_{B,C} = 2$.

First of all, if agents A and B form a group, according to our protocol, agent A's total cost is $\frac{2}{4} + 1.25 = 1.75$*. If agent C is added to the group, agent A's total cost would become* $\frac{2}{4} + 1.5 = 2 > 1.75$*. Thus by our grouping strategy, agent C will not be added in the group and will have to act alone. Similarly, if agent A and C grouped together first, adding agent B will also decrease the current social welfare thus the three agents will have to be split into two groups.*

However, agent C has a very high interaction cost to the center. Thus if agent C is in the group, we will save a huge amount on the total interaction cost incurred in the system whereas for agent B, being in the group does not benefit it that much.

From the example, we can see that starting from the agent with highest interaction cost will improve social welfare. Secondly, we also want to take into consideration the different topologies in our strategy. Thus we added in another small justification: if the interaction topology is a star topology, we start grouping from the center agent first, for the rest of the agents, we start from the agent with highest interaction cost to the center.

For a tree topology, we start from the leaves. If there is more than one leaf node, we start with the one with highest interaction cost to the center. Then for the rest of the nodes, we apply the order from highest interaction cost to the lowest.

Let us give an example on how the improved method is better.

Example 15 (Tree Topology Example) *Consider the tree topology in Figure 5.2. Let the inter-agent interaction cost equal 3 for any two agents. (i.e.* $IC_{i,j} = 3$*, for* $i, j \in A, ..., F$*). Assume the interaction cost from each agent to the center is as follows:* $C_A = 3, C_B = 5, C_C = 4, C_D = 4, C_E = 3, C_F = 5$*. Let us assume that using the first grouping strategy, agent A goes first.*

Using the original grouping strategy, agent A will contact its neighbours B and C and eventually, the three will form a group and agent A's total cost equals 2.75. Now if we add another agent to the group, agent A's new cost would be $\frac{3}{5} + \frac{9}{3} = 3.6$*. The social welfare for the current group is going down thus no more agent will be added in the group. As we can see, agent D, E, F have to communicate to the center alone since they do not have any other neighbours. Thus the total interaction cost we saved is* $1 - \frac{15}{24} = 0.58$

Now let us consider the use of heuristics. In this scenario we start communication from the leaf nodes. Thus agent D goes first to communicate to the neighbours (Agent B) and form a group. Agent E then communicates to agent C to form a group. Then agent F will communicate to join the group with agent E and C. From calculation we can get that this group cannot accept any more agents. And agent A will end up joining the same group as agent D. Thus in this scenario, we have two groups in total. And the total interaction cost saved is $1 - \frac{9}{24} = 0.625$.

As we can see, using the improved grouping strategy, we save an extra 14.5 percent on the total interaction cost.

5.4 Experiments

We developed a simulation environment and experimented with our mechanism using different numbers of agents, interaction topologies, and cost function combinations. We first report results which show that our group-formation and cost-sharing protocols greatly reduce the overall interaction cost for agents, but, as expected, is sensitive to the type of cost function. We then report our findings on the impact that different interaction topologies have, both in terms of interaction-cost savings, as well as on the average group size formed.

5.4.1 Experimental Setup

Our simulation environment assumes that agents are providing information, and thus the interaction cost represents the cost of communicating the information. Therefore, we use the term *communication cost* when referring to the interaction overhead. We experimented with two different types of cost function. We first used a constant-cost function, which depended only on the fact that an interaction with the center occurred, and not on the amount of information sent. We assigned cost functions to agents at random, by drawing values from the interval [1,100]. The second type of cost function used was per-unit cost function, which depended on the amount of information transmitted. The cost of transmitting a single unit of information to the center was drawn uniformly from [1,10], and the amount of information each agent had to transmit, in terms of units, was also drawn uniformly from one to ten. For all experiments, we set the inter-agent communication cost to be five, independent of whether the communication cost to the center was a constant, or a function of the number of units of information sent. We note that the inter-agent communication cost was set quite high in order to bias the results so that they were pessimistic in their cost-savings finding. We expect that in many actual applications, the inter-agent communication cost would be much lower than the cost of interacting with the central mechanism.

We studied different interaction topologies in our experiments. We used the simplest topology, the line topology in the experiments comparing different cost functions. In our second set of experiments we compared the line, star, and tree topologies when agents have a fixed communication cost function, as defined above. We ran each experiment 100000 times, and report the average values.

Measurement

We had three goals when running our experiments. We first wanted to measure the cost-savings agents experienced from using our mechanism. Let C_{ns} denote the total cost without our grouping strategy and C_s denote the total cost with the grouping strategy. We calculated a cost-measurement using the following;

$$\text{Savings} = S = 1 - \frac{C_s}{C_{ns}} \tag{5.6}$$

That is, $S \times 100$ is equal to the percentage savings experienced by the agents. Values of S closer to 1.0 are better. Our second goal was to gain an understanding of the size of groups that would form when using our mechanism, and how this was influenced by the types of cost functions agents had. Finally, we also studied the cost savings as a function of the underlying interaction topology.

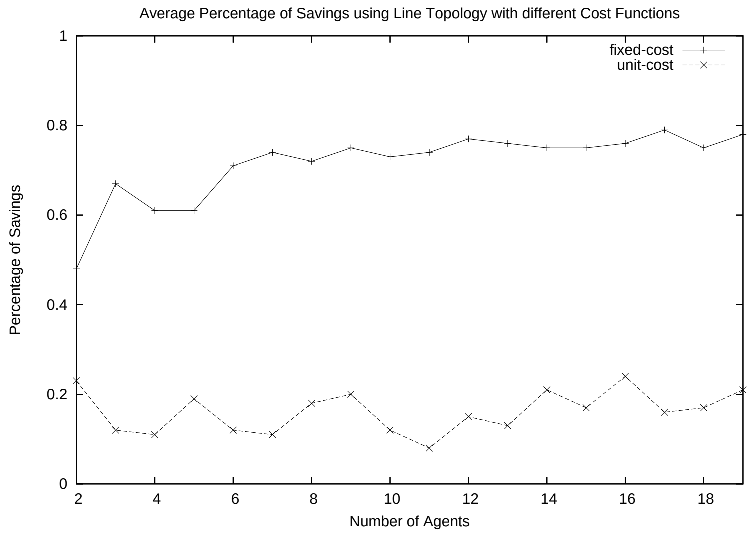

Figure 5.3: Savings using Line Topology, Fixed-cost v.s. Unit-cost

5.4.2 Results

Comparison with different cost functions

Figure 5.3 shows the difference in cost savings between situations where agents had fixed communication cost functions as opposed to unit-cost communication cost functions. We set the interaction topology to be a line. The result shows that the savings with fixed-cost is quite high (up to 80%) and relatively stable as the number of agents increases. Comparing the outcomes between the two cost-functions, we see a higher saving is achieved using the fixed-cost scheme. However, the savings for the unit-cost scheme can still be maintained at an average of 15%.

Next, Figure 5.4 shows the average number of groups formed under both fixed cost functions and unit-cost functions. We see that for the same number of agents, the unit-cost scheme generates a larger number of groups than the fixed-cost scheme. This is to be expected. In the fixed-cost scheme there is a benefit to having larger groups, which

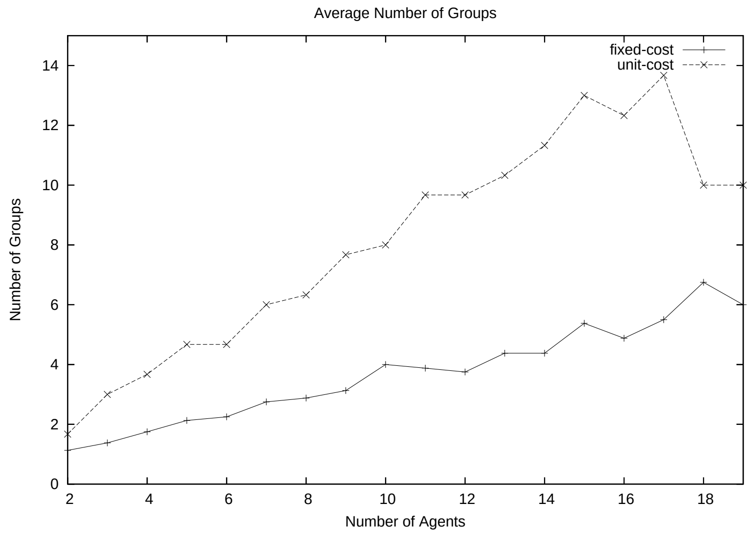

Figure 5.4: Number of Groups, Fixed-cost v.s. Unit-cost

leads to a decrease in the number of groups formed. While a larger group has a larger inter-agent cost, the cost of interacting with the center is fixed, so each agent has to pay a smaller share. However, in a unit-cost scheme, as the number of agents increases in the group, the total number of units also increases, and therefore the total communication cost increases. Thus under the unit-cost scheme, the total cost will increase rapidly to the point where introducing a new agent will not benefit existing agents. As a result, the average group size in a unit-cost scheme is smaller. We also studied the variance in this setting and our resulting variance is around 0.01. This is a small variance and thus a good indication that our algorithm does not fluctuate much.

Comparison with different interaction topologies

The next set of experiments compares the result of a fixed communication cost function on three different topologies – line, star and tree. In a star topology, agents are grouped as a star of size 6. Stars are disjunctive, which means the maximum group size in this

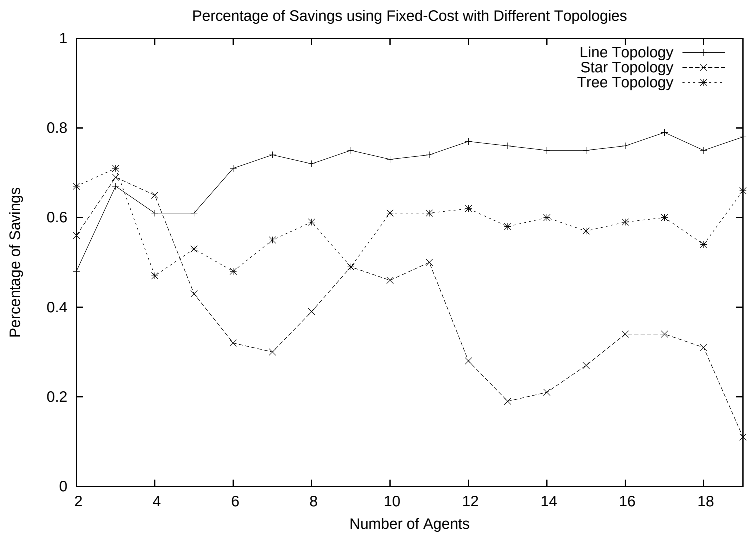

Figure 5.5: Savings using different topologies.

topology is 6. In the tree topology, balanced binary trees are used. Figure 5.5 shows the average percentage of savings using the three topologies.

First, we observe some interesting patterns in the star topology results. In particular, we observe a wave pattern of size six. This particular finding is very interesting and it captures the property of the star topology. Since in our experimental setting each star is disjoint, the maximum group size is six and each group of six is essentially independent from the others. Another important property in the star topology is agents have a very limited number of neighbours. Only one agent is fully connected to other agents. Due to the fact that our inter-agent communication cost is set to be quite high, this means that as the number of agents increase, there is a higher chance that the agent will not be accepted in the group. If an agent is not accepted, it will have to communicate to the center alone and decrease the total savings in the system. This explains the downward trend inside each group.

Next we discuss our findings from using a tree topology. The result looks similar to the line topology. The average percentage of savings in this topology does not fluctuate greatly as the number of agents increases. However, we see that the average saving in this topology is not as great as the line topology, which came as a surprise. Our hypothesis is that our group mechanism does not fully take advantage of the tree topology, and in particular, we are observing situations where parents are forming one group, but leaving all the children isolated, thus increasing the overall communication cost. This raises an interesting question for future investigation. In particular, we are interested in whether it is possible to modify the grouping scheme so that it is aware of topological properties in the interaction network. If this is possible, then we would expect to see an increase in savings.

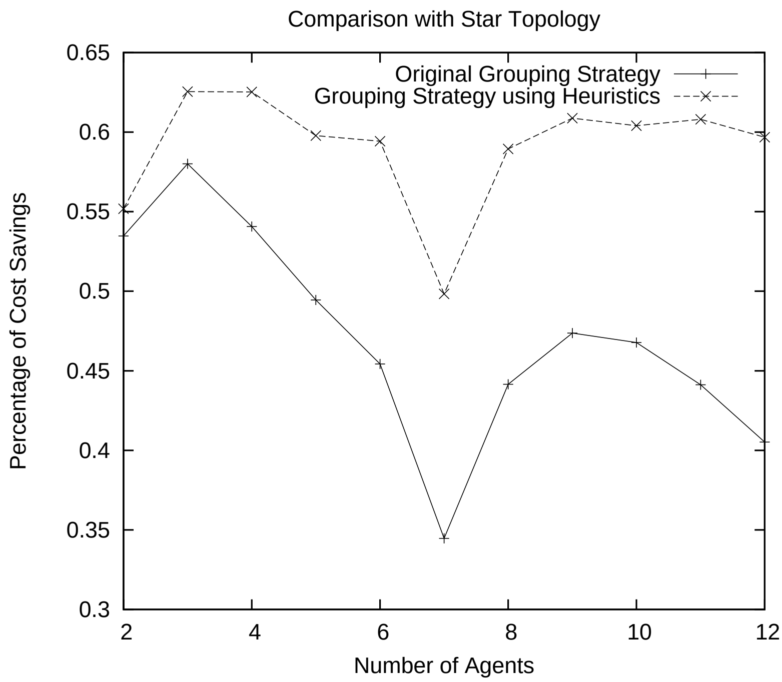

Figure 5.6: Savings in Star Topology using different grouping strategies.

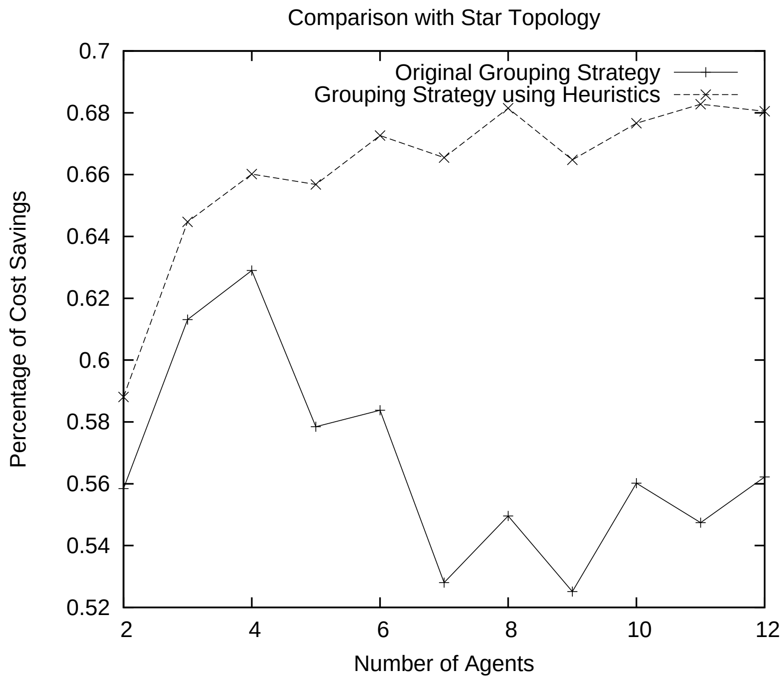

Figure 5.7: Savings in Tree Topology using different grouping strategies.

Comparison with different grouping strategies

Having done the above, we decide to look at possible improvement using our improved grouping strategy. In particular, in the next set of experiments, we compare the result using the original grouping strategy with the improved strategy where we start the grouping from the agent with highest cost. The setup for this set of experiment is exactly the same as the previous ones.

We can make several observations from figures 5.6, 5.7 and 5.8. First of all, our new strategy does improve the overall social welfare. Having said that, the improvement depends heavily on the type of topology. We see a great amount of improvement in the star topology. In a star topology, there can only be one group with multiple agents. If we start from the agent with highest cost, we guarantee that in the final group, the agents

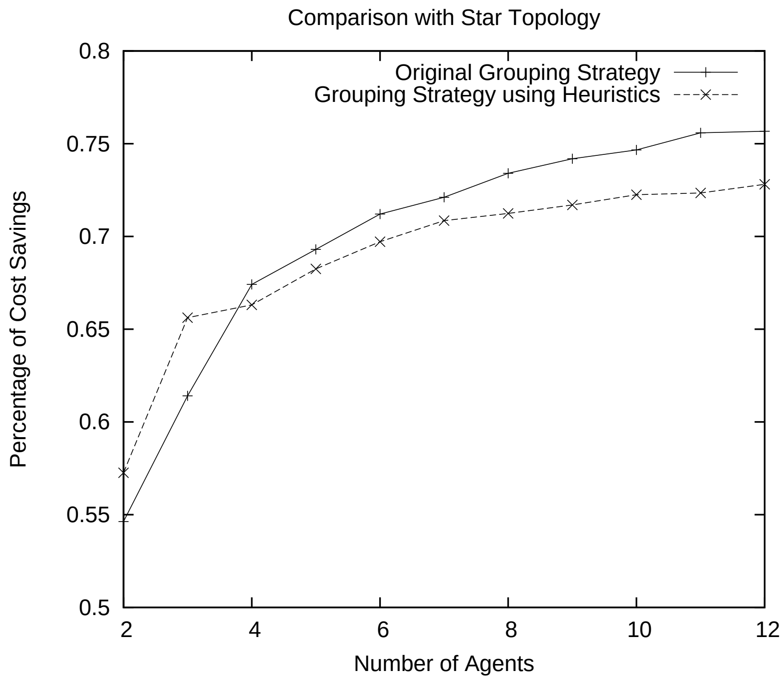

Figure 5.8: Savings in Line Topology using different grouping strategies.

with higher interaction costs are inside the group and thus saves more globally. In the tree topology we also see a reasonably good savings. However, with the line topology, the new scheme does not make much difference(for some number of agents it is even worse than the original). This is acceptable because first of all, the savings originally in a line topology is already quite high and hard to beat. Secondly, due to the particular neighbour limitation in a line topology, starting from the agent with highest interaction cost might not necessarily result in maximum group sizes.

5.5 Summary

This chapter can be summarized in three sections. In the first section, we studied the cost-sharing protocol. We discussed existing protocols and showed that they do not have all the properties we desire. We then presented our protocol in detail and proved all the properties it carries. In the second section we described our grouping strategy. We discussed issues and challenges we face and gave an algorithm for our grouping strategy. We then analyzed the result on three different interaction topologies and proposed improvements. In the last section, we carried out experiments that further studied how much cost we saved using our model and what is the outcome of our grouping strategy in terms of savings on cost share with different interaction topologies.

Chapter 6

Related Work

In this section we introduce literature that is related to our research and discuss the similarities and differences with our model. The ideas in this thesis are related to several concepts, including cost-sharing protocols, mechanism design and coalition formation. We first study the related literature in the cost-sharing algorithms. Then we study the related work in mechanism design and coalition formation. We introduce the application area that motivated our research, sensor networks at the end of the chapter [7, 29, 22].

6.1 Cost-Sharing Strategies

One area closely related to the work presented in this thesis is cost-sharing. There are many different algorithms and protocols for cost-sharing in the literature, but they can be broadly classified into two groups. The first approach focuses on trying to achieve both strategy-proofness and budget-balance properties, while the second approach tries to achieve both strategy-proofness and efficiency. Unfortunately, it has been shown that there is no strategyproof mechanism that is both budget-balanced and efficient [9].

One of the simplest cost-sharing schemes that is in the group focusing on both strategy-proofness and budget-balance is the *Shapley Value* cost-sharing. The Shapley Value cost-sharing refers to a cost-sharing algorithm where each agent pays an equal share of some fixed cost. This rule states that given some fixed cost C and n agents in the group, every agent pays an equal share of $\frac{C}{n}$. It is incentive compatible since an agent's cost does not depend on its own cost, thus manipulating its own cost will not result in any changes in its charged amount in a group. Also this algorithm is budget-balanced (as we can see, $\frac{C}{n} \times n = C$, which is the total cost to be covered). However, this algorithm is not efficient, since no matter what agents' original cost are, they always pay the same amount.

An algorithm that represents the group focusing on both strategy-proofness and efficiency is the *Marginal Cost-sharing* protocol. Marginal Cost-sharing is in the group of VCG mechanisms. The Marginal Cost-sharing algoritm charges each agent an amount depending on the marginal cost of the agent. Given an agent i in a group of agents S, the mechanism charges this agent $C(S) - C(S - i)$, i.e. the marginal cost for this agent when it joined the coalition. Since it is a variation of the VCG mechanism, this algorithm is strategyproof and efficient. However, just like other VCG mechanisms, it is not group-strategyproof, nor is it budget-balanced [20].

There are lots of cost-sharing algorithm that modify and improve on these algorithms (for example, [21, 2]). One of the most famous cost-sharing algorithms was developed by Moulin and Shenker [21], where a fixed group of agents share a single input and single output technology with decreasing returns (the detailed algorithm has been discussed in Chapter Four). This algorithm is budget-balanced and fair meaning an agent with higher demand pays more of the cost. In this algorithm, the cost share depends anonymously upon demands and an agent's cost share is independent of demands higher than its own. The Moulin-Shenker algorithm is nice in several ways. It is anonymous (the names of the agents do not matter), monotonic (an agent's share increases as it demands more output), and the result of the protocol gives a unique Nash Equilibrium, thus following this protocol is the dominant strategy for agents. However, there are significant differences in their settings from ours. The Moulin-Shenker algorithm assumes the quality cannot be manipulated. However in our case our agents may deliberately manipulate declared cost in order to make more profit. Moulin also looked at cost-sharing from a strateyproof and efficiency point of view with indivisible units of output [19]. Each agent may only consume either one or zero unit and there is a specific ordering for agents to make payments. It is a variation of the Marginal-cost sharing algorithm (e.g. agent one pays $C(Q_1)$ and agent two pays $C(Q_1 + Q_2) - C(Q_1)$), with a goal of prioritizing budget-balancing over efficiency. However, the mechanism only guarantees approximate budget-balance, and recent studies show that for some fundamental cost-sharing problems, the algorithm suffers poor budget-balance [23, 18].

The study of cost-sharing also appears in network domains [5, 1, 2]. Roughgarden *et al.* proposed a budget-balanced and group-strategyproof algorithm for Steiner forest cost-sharing problems [5]. A Steiner forest consists of an undirected graph $G = (V, E)$, non-negative costs C_e for all the edges $e \in E$, and a set $R \subseteq V \times V$ of k terminal pairs. Each terminal pair $(s, t) \in R$ is associated with an agent who wishes to establish a connection between nodes s and t. A feasible solution is a forest F that contains a s, t-path for each connection request $(s, t) \in R$. They use an auction-based approach. Given a set of agents who are potential customers of a service provider, each agent has a utility function associated with the service and each agent makes a bid to submit to the service. The service then runs an auction to determine which agents get the service and how much they should pay for the service. Given a set of N agents and a cost-sharing method c, the algorithm works as follows:

1. Step 1: create a set of agents $Q = N$
2. Step 2: for each agent $i \in Q$, check if the cost share $c(i)$ is less or equal to its bid b_i.
    - If true, terminate
    - Otherwise, remove all agent $i \in Q$ with $c(i) > b_i$, go back to Step 1

They showed that this algorithm is strategyproof but only approximately budget-balanced. In our model, incentive-compatibility and budget-balance are equally important; we need a protocol that can achieve both.

Feigenbaum *et al* studied multi-cast cost sharing [2]. Multicast routing is a technique for transmitting a packet from a single source to multiple receivers without wasting network bandwidth. The routing is solved by constructing a directed tree and sending only one copy of the packet over each link of a directed tree. When a packet reaches a branch point of a tree, it is then duplicated and a copy is sent to each link below. The study of multicast cost-sharing is to come up with an algorithm that handles the cost incurred during this transmitting stage. Their work extends the previous result on the Shapley Value (by dividing the cost of each link *l* equally among all members that are downstream to it) and Marginal Cost (selects the receiver set that maximizes total net worth, and charges each receiver its reported utility minus a "bonus" equal to the marginal value it added to the system) approaches. They focus on the computational complexity of the algorithm and come up with a group strategy-proof algorithm which can be computed with exponentially lower worst-case communication than the Shapley Value. They also characterized groups that can strategize successfully against the Marginal Cost mechanism. However, their algorithm may not achieve complete budget-balance.

Although achieving budget-balance is a general goal in most cost-sharing applications, there are also applications who assume a monopoly network operator and work on algorithms that maximizes revenue [8]. We will not elaborate on these works here since they are the opposite of what we want to achieve in our thesis.

6.2 Mechanism Design and Coalition Formation

The ideas in this thesis are related to concepts from both mechanism design and coalition formation. Coalition formation has been a well-studied area of multiagent systems and ecommerce (see, for example Shehory and Kraus [27] and Sandholm *et al.* [24]). For the most part, this work has not focused on the interaction overhead of agents nor does it assume that agents are forming groups or coalitions to reduce their interaction costs with some central mechanism, whereas we explicitly include these features in our model.

Forming coalition serves as a way for multi-agent cooperation. A coalition can be formed for different purposes. For the most intuitive part, agents may form coalitions to perform tasks that they are unable to perform otherwise. With respect to this, several studies have provided algorithms for forming agent coalitions [27, 12, 32]. This work usually focuses on task oriented domains where agents need to form coalitions to perform a specific task. Zlotkin and Rosenschein study this problem on self-interested agents [32]. While they study self-interested agents, they restrict themselves to super-additive environments (where one's utility is strictly increasing) and so are able to achieve an efficient algorithm for coalition formation for this specific environment. In general, the algorithms for coalition-formation in task-oriented domains have a different goal than our work since they are focused on how agents would coordinate in order to perform a specific task. The agents are essentially *cooperative* rather than self-interested. In our self-interested setting, agents only form coalitions if there is a benefit for themselves.

6.2.1 Coalition Formation in Electronic-Marketplace

The work that most closely related to ours studies how groups of agents can form in order to derive deals and discounts from auctions in electronic-marketplace. This particular problem has received excessive attention [16, 15, 30, 25, 26].

Yamamoto and Sycara [30] studied the problem with a goal of reducing computational complexity for very large scale of buyers (e.g, thousands of buyers). They studied the problem of forming *buyer coalitions* to enlarge total quantity of goods purchased in each transaction, so each buyer can obtain a lower price without buying more than its original demand. They used a reversed combinatorial auction approach (which is called a *GroupBuyAuction*). A GroupBuyAuction is formed based on the category of items. For example, a buyer may want to buy either TV A for 300 dollars or lower, or TV B for 400 dollars or higher. It is a form of reverse auction where sellers make bids with volume discount prices (for example, a TV A for 250 dollars if more than 5 are sold, 300 dollars otherwise) and buyers pool their demand to maximize their power. A mechanism is assigned to each group to manage the auction on behalf of the buyers. When an auction closes, the mechanism splits the buyers into subgroups consisting of buyers preferring the same items and also selects the winning seller for this coalition, and at last, calculates the price for each buyer. Buyers wanting the same item may pay differently depending on their original reservation price. They give an efficient algorithm for coalition formation and payoff division for the case where each buyer wants an XOR of items in the same category . A suboptimal coalition (a coalition in each category) is chosen by choosing in each round a coalition with maximum value among all optimal ones, which are formed for each item by the buyers who have not been picked to be in the coalition.

Li *et al.* extended Yamamoto's idea and studied the mechanism design problem of coalition formation and cost sharing in an electronic marketplace [16, 15]. They studied the problem combining coalition formation and combinatorial auction and used a regular auction scheme instead of a reverse auction. They tackled the problem from a core-stability perspective and their work is the first work that studies both stability and incentive compatibility of an economy with incomplete information. They also chose several cost-sharing algorithms (e.g. equal cost-sharing, Clarke's mechanism) to test the average efficiency loss. Their goal is quite different from ours. They focus on volume discounts, i.e. the seller gives a discount when higher volumes are purchased. This requires the quantity of each item in the combinatorial auction to be more than one and buyers who want same items to group together in order to take advantage of the price discounts. In our model, we study the problem of saving interaction cost. Our model can be applied to, but is not limited to combinatorial auctions. Furthermore, Li's and Yamamoto's idea does not take cost into account when forming coalitions. As a result, the grand coalition will always be formed since the more agents in the coalition, the better discount rate they are getting. In our model agents cannot always form the grand coalition due to cost involved.

Sarne and Kraus also studied a similar problem where buyers form coalitions to take advantage of discounts [25]. In addition, they introduced a new element: *costly environment*, which is similar to our model. They assume that agents' search for coalition opportunities is costly, at each stage of a search an agent has to spend resources in locating and interacting with another agent. Thus, each agent, when entering the marketplace, must consider if it is going to execute its task immediately or if it is more beneficial to engage in costly search to extend the coalition. If an agent decides to engage in such a search, at each stage it must decide whether to terminate the process or not. Forming the grand coalition is not applicable anymore due to the cost introduced during formation. They then explore agent's optimal strategies in the equilibrium in this situation. The difference with our model is that the cost during coalition formation is known and cannot be manipulated. While there are similarities between these works and ours, the motivation is different as their focus is on forming coalitions to take advantage of volume-based discounts in e-marketplaces, whereas we study a more general problem domain, and focus on ways of reducing the overall interaction cost for agents in the system.

Kraus *et al* described distributed coalition formation schemes for multi-agent systems mainly focusing on increasing the group's total utility in [26]. They study the case where a single agent cannot perform goals by itself or is inefficient to do so. Furthermore, being in just one coalition may also lead to a waste of resources. Thus they propose algorithms that allow agents to form overlapping coalitions where goals have precedence order. They use a greedy-approach to handle the coalition formation stage. However it requires a pre-calculation of all the permutations of coalitions including up to k agents for each agent, which has a computational complexity of $O(n^k)$. Thus the highest coalition size is limited which means the algorithm is not suitable for large-scale systems.

In another work, Kraus, Shehory and Taase studied the problem of coalition formation with agents with incomplete information under time constraints [13]. In terms of coalition formation, they propose a ranking strategy where each agent ranks the possible coalitions and computes the revenue distribution for the top coalition. It then offers this coalition to others, and accepts only proposals in which its net benefit is at least as high as its net benefit from its computed coalition. In particular, they proposed a compromise strategy for revenue distribution where agents compromise and agree with a payoff lower than their estimated share and showed that under time constraint this strategy is stable and increases social welfare compared to non-compromise schemes. Their focus is on stability and computation complexity while we focus on maximizing social-welfare.

All the work mentioned above study a particular domain of the coalition formation problem. We, on the other hand, look at the problem in a more general setting. Our model is generic and can be applied to all the areas discussed above. The two applications we chose to discuss in the paper can be generalized and represent a wide variety of application areas.

6.2.2 Coalition Formation from Other Perspectives

The work mentioned in the previous section are all from an e-commerce perspective. There are also a lot of game-theoretic work that study coalition formation from other perspectives.

Ketchpel [11] proposed a coalition-formation algorithm that handles coalition generation and payoff distribution simultaneously. The resulting coalitions are groups of size two where one agent acts as the manager of the group and the other agent gets paid a fixed amount. They propose a *Two Agent Auction* scheme that solves the payoff division problem (i.e. how to choose the manager of the group). However, this process can be very complex and thus inefficient. His algorithm does not consider incentives so it cannot be used on self-interested agents.

Zlotkin *et al* [32] analyze payoff division in subadditive task oriented domains. Their algorithm guarantees each agent an expected value equal to the Shapley value. They achieved it by only linear complexity in the number of agents, as opposed to exponential complexity for a naive implementation of the Shapley value. Shapley value does not usually guarantee stability in each group, however, in the setting of a subadditive domain as they study, the solution does guarantee that the group of agents will stay in the same coalition structure. Furthermore, in their work, the agents always form the grand coalition, which is very different from our model.

Yokoo *et al* [31] have looked at the existing solution concepts and pointed out that the existing concepts have limitations when applied to open, anonymous environments such as the Internet. In such environments, an agent may hide its identity and pretend to be multiple agents, or a group of agents can collude and pretend to be a single agent. These manipulations can be very hard to detect. Having identified the issues, Yokoo *et al* have developed a new solution concept called the anonymity-proof core, which is robust against these manipulations.

There is also work that analyze the optimization within each coalition and the computational complexity. Contizer and Sandholm [6] looked at the coalition formation problem from a computational-complexity perspective. They studied the set of superadditive games and provided an efficient method for checking whether a given outcome is in the core. They also showed, however, checking whether the core is nonempty is in fact NP-hard due to the difficulty in determining the collaborative possibilities (the set of outcome possibly for the grand coalition).

Sandholm *et al*. [24] have studied a worst-case bound on the quality of coalition structure while only searching a small fraction of the coalition structures. This is important since searching the optimal coalition structure is an NP-Complete problem. They provided an algorithm that established a tight bound within the minimum amount of search.

6.3 Coordination in Sensor Networks

The motivating application of our research was sensor networks [14]. A sensor network is a network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

Dang *et al.* proposed coalition formation algorithms for sensor network applications with overlapping coalitions [7]. They proposed two algorithms to find the optimal coalition for sensors. Their first algorithm is a polynomial time approximation algorithm that uses a greedy approach. Their second algorithm is based on a branch-and-bound technique (i.e. search through the search space in a depth-first manner). One major contribution of their paper is considering the situation of overlapping coalitions. However their work is quite different from ours as in the sensors in their setting are cooperative and one of their limitations is that the environment is static as opposed to dynamically changing values.

In another paper, Soh *et al.* investigated negotiation strategies for forming coalitions to track moving objects [29]. Their focus is on forming dynamic coalitions for tracking targets in a noisy and uncertain environment. Agents have incomplete information and need to respond in real-time. Thus their focus is on improving the quality of coalition formation process and the quality of coalition in terms of target-tracking. In their algorithm, agents carry out a one-to-one negotiation to its neighbours based on a negotiation protocol. Their goal is for agents to form a satisfying coalition in a timely manner.

Rogers *et al.* [22] looked at information fusion within a bandwidth-limited multi-sensor network. They consider sensors to be self-interested rational agents who work to maximize their own utility. They defined a valuation function based on Kalman filters. Then they extended the generalized VCG mechanism to deal with the valuation functions and proved their mechanism is incentive compatible and individually rational. In this paper, the sensors have incomplete views of the world and need to communicate with other sensors to get a full view of the world and gain utility from it. However the bandwidth is limited so sensors have to decide which neighbour to contact for information. This setting is quite different from ours as in that the agents in our scenario do not care about the overall information. One most important difference is that most literature in sensor networks considers cooperative agents, i.e. agents working together with the same goal in mind. We, on the other hand, are more interested in self-interested agents. Although the sensor network application motivated our research, there are quite a few differences with respect to the model we would like to study and furthermore, we would want to expand our model to a more general domain.

6.4 Summary

In this chapter, we provided an overview of literature in the area of mechanism design, cost-sharing and coalition formation. Our work in this thesis is different from most of the work mentioned above. None of the work on cost-sharing achieves incentive-compatibility, budget-balance and fairness together as we desire. Thus we developed our own cost-sharing protocol which is budget-balanced, prevents agents from manipulating the cost and is also fair. The literature on sensor-networks is a motivation for our model. However, most applications in this area assumes agents are cooperative rather than different stake-holders with a goal of maximizing their own utility. Moreover, our model is not limited to the network-scenario; it covers a wider range of applications. The work that is closest to our research is the work on mechanism design for volume discounts in e-commerce. In this work agents are self-interested and will form coalitions in order to get a lower discounted price. Some literature studied the set of super-additive games, and in these settings agents will always form the grand coalition thus the only question is how to divide payments. In other work, agents incur a cost during the coalition formation stage thus the grand coalition is not always feasible, which is closer to our work. However our goal is different from theirs. They focus on achieving volume discounts (meaning the auction is on a single type of item) while we focus on lowering the interaction cost incurred when communicating to the center, and our work can be applied in both single-item and combinatorial auctions.

Chapter 7

Conclusion

In this thesis we looked at the problem of coalition formation with self-interested agents in order to reduce total interaction cost. We studied a problem where agents are limited in their interactions with a central mechanism by cost, but have the possibility of coordinating with other agents in the system in order to share and reduce these costs. Our goal was to design a model which allows agents to form stable groups to avoid the overhead of each agent interacting independently with the mechanism, and at the same time still promoting truth-telling and ensuring individual rationality throughout the model.

In this chapter, we summarize the contributions of this thesis. We also describe some directions for future work.

7.1 Contributions

The main contributions of this thesis were:

- A model of costly interaction

- Core-stable and fair payoff algorithms for reward-distribution
- Novel cost-sharing protocol
- Analysis on different interaction topologies

7.1.1 A Model of Costly Interaction

We presented a model for self-interested agents to form groups with neighbours in order to reduce total interaction cost. We gave motivating examples that can be applied using our model and discussed issues and difficulties we are facing in Chapter Three. In particular, our model consists of a reward-distribution and a cost-sharing algorithm which were studied separately in Chapter Four and Five.

7.1.2 Core-stable and Fair Payoff Algorithms for Reward-Distribution

We analyzed the reward-distribution mechanisms in Chapter Four. We made the assumption in the information-provision setting that information can be discretized into areas and proposed a redistribution algorithm to handle the excess amount collected by the VCG mechanism. We then observed the inefficiency issue in the combinatorial auction setting when using a regular VCG mechanism. Thus we modified the group bid to be a permutation of XORs of all the possible bids and looked at the issue from a coalitional games perspective. The major contributions in this chapter are the two reward-distribution mechanisms we proposed which features core-stability and fairness. The core-stability payoff solution features a payoff vector proportional to each agent's original valuation. As a result, the outcome is unique and core-stable. The fair payoff mechanism used a different valuation function and was *fair*. However when the core is non-empty using this mechanism, it is not necessarily unique. Then we showed that it is not possible to guarantee both core-stability and fairness in our model. Finally, we carried out experiments to test the effect on agents' payoff using two different reward-distribution schemes.

7.1.3 Cost-Sharing Protocol

We presented our cost-sharing protocol and group formation strategy in Chapter Five. Our cost-sharing protocol features three properties: incentive-compatibility, cost-recovery and fairness. The incentive-compatibility is crucial in our protocol because we are working with self-interested agents, and the interaction cost from an agent to center is known privately to the agent itself and thus may be manipulated. Our protocol also satisfies a cost-recovery property meaning that we are not collecting extra money from the agents. The final property we had is *fairness*. We proposed that the cost agents pay inside groups should reflect the benefit they receive. The more benefit they receive, the higher amount they should pay.

7.1.4 Analysis on Different Interaction Topologies

We proposed a greedy-based grouping strategy for group formation and we observed that our final outcome depends on the agents' interaction topology and analyzed different topologies. We analyzed the total savings in interaction cost using three different interaction topologies: line, star and tree. We discovered that the outcome is different because of the different layouts of topologies. We then proposed a heuristic for group formation which takes different topology patterns into consideration. Experiments showed that the heuristic worked better than the original grouping strategy.

7.2 Directions for Future Work

In this section we outline some direction for future work that arise from this thesis.

7.2.1 Stability and Fairness: The Impossibility Result

In Chapter Four we proposed two reward-distribution mechanisms which features core-stability and fairness respectively. Unfortunately, we showed that there is no guarantee that we can achieve both these properties together.

This observation leads to two directions. First, we would like to study further into the valuation functions and the mechanisms we used for reward-distribution to see if a core-stable and fair algorithm is attainable. This would possibly involve in modifying the mechanism (the VCG mechanism) that we are currently using.

The second direction is studying the tradeoffs between core-stability and fairness. We have conducted preliminary experiments which give us a general idea about the effect of the two mechanisms. However, experiments on larger groups of agents might provide additional insight as in what kind of agents benefit more in which algorithm.

7.2.2 Fair Cost-Sharing Protocol

We proposed an incentive-compatible, cost-recovery and fair cost-sharing protocol in Chapter Five. Some future work on this area is to further study the definition of fairness. Our current definition requires that an agent with higher benefit is not paying less then an agent with lower benefit. We would like to extend this definition so that an agent with higher benefit is always paying a higher cost, to best reflect real-world situations. We would also like to see if we can modify our cost-sharing algorithm so that the extended degree of fairness can be reached.

7.2.3 Grouping Mechanism and Topology Dependency

We observed in Chapter Five that our grouping mechanism is topology dependent. Our heuristic mechanism goes one step further and takes some advantage of different topologies. However, more can be done in this area. For example, we would like to take a further look at possible extensions to our mechanism to take advantage of different topologies. We would like to investigate whether a topology-independent grouping strategy is feasible. We would also like to look at combined, more complicated interaction topologies and conduct further experiments.

7.3 Summary

In this thesis we studied a problem where self-interested agents need to interact with some centralized mechanism where this interaction is costly. We provided a model to allow self-interested agents to form groups in order to save on the total interaction cost. We proposed two different reward-distribution algorithms based on different applications. We analyzed the reward-distribution mechanism from a coalitional games perspective and proposed two solutions featuring core-stability and fairness. We then proposed a novel cost-sharing algorithm which is incentive-compatible, cost-recovery and fair. We provided algorithms for group formation techniques and studied the effect of this technique on different interaction topologies. In the end we conducted preliminary experiments that validate our model. Our reward-distribution algorithm applies to any application that can fall into the two generalized category generalized from information provision network and combinatorial auction setting. Moreover, our cost-sharing algorithm and grouping strategy are generic so they can be applied to any applications.

Bibliography

[1] Elliot Anshelevich, Anirban Dasgupta, Jon M. Kleinberg, Éva Tardos, Tom Wexler, and Tim Roughgarden. The price of stability for network design with fair cost allocation. In *FOCS*, pages 295–304, 2004.

[2] A. Archer, J. Feigenbaum, R. Sami A. Krishnamurthy, and S. Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47, 2004.

[3] Martin Bailey. The demand revealing process:to distribute the surplus. *Public Choice*, 91:107–126, 1997.

[4] Ruggiero Cavallo. Strategyproof redistribution of VCG payments. In *In the Proc. of the 5th Int. Joint Conf. on Autonomous Agents and Multi Agent Systems (AAMAS 2006)*, Hakodate, Japan, 2006.

[5] Shuchi Chawla, Tim Roughgarden, and Mukund Sundararajan. Optimal cost-sharing mechanisms for steiner forest problems. In *WINE*, pages 112–123, 2006.

[6] V. Conitzer and T Sandholm. Complexity of determining nonemptiness of the core. In *the Proc. of the Eighteenth International Joint Conference on Artificial Intelligence(IJCAI)*, pages 613–618, 2003.

[7] Viet Dung Dang, Rajdeep K. Dash, Alex Rogers, and Nicholas Jennings. Overlapping coalition formation for efficient data fusion in multi-sensor networks. In *In the Proc. of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, Boston, USA, 2006.

[8] A. Fiat, A. Goldberg, J. Hartline, and A Karlin. Generalized competitive auctions. In *the Proc. of the 34th Symposium on Theory of Computing, ACM Press*, pages 72–81, 2002.

[9] J. Green and J-J Laffont. *Incentives in Public Decision Making*. North Holland, Amsterdam, 1979.

[10] James P. Kahan and Amnon Rapoport. *Theories of Coalition Formation*. Lawrence Erlbaum Associates, 1984.

[11] S. Ketchpel. Forming coalitions in the face of uncertain rewards. In *the Proc. of AAAI-94, Seattle, WA*, pages 414 – 419, 1994.

[12] Michael Klush and Onn Shehory. A polynomial kernal-oriented coalition formation algorithm for rational information agents. In *In the Proc. of ICMAS*, pages 157–164, 1996.

[13] S. Kraus, O. Shehory, and Gilad Taase. The advantages of compromising in coalition formation with incomplete information. In *In the Proc. of the 4th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, pages 588–595, 2004.

[14] Victor Lesser, Charles Ortiz, and Milind Tambe, editors. *Distributed Sensor Networks: A multiagent perspective*. Kluwer Publishing, 2003.

[15] Cuihong Li, Shuchi Chawla, Uday Rajan, and Katia Sycara. Mechanism design for coalition formation and cost sharing in group-buying markets. *Electronic Commerce Research and Applications*, 3:341–354, 2003.

[16] Cuihong Li and Katia Sycara. Algorithm for combinatorial coalition formation and payoff in electronic markets. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 120–127, Bologna, Italy, 2002.

[17] Andreu Mas-Colell, Michael Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[18] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond moulin mechanisms. In *In the Proc. of EC07*, 2007.

[19] H. Moulin. Incremental cost sharing: Characterization by coalition strategyproofness. *Social Choice and Welfare*, 16:279–320, 1999.

[20] H. Moulin and S Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533.

[21] Herve Moulin and Shenker Scott. Serial cost sharing. *Econometrica*, 60(5):1009–37, 1992.

[22] A. Rogers, R. K. Dash, S. Reece, S. Roberts, and N. R. Jennings. Computational mechanism design for multisensor information fusion. In *In the Proc. of the 5th Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2006)*, pages 1463–1464, Hakodate, Japan, 2006.

[23] T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *In the Proc. of the 38th Annual ACM Symposium on the Theory of Computing (STOC)*, page 79C88, 2006.

[24] Tuomas Sandholm, Kate Larson, Martin Andersson, Onn Shehory, and Fernando Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence*, 111(1-2):209–238, 1999.

[25] D. Sarne and S. Kraus. *The Search for Coalition Formation in Costly Environments,Cooperative Information Agents VII.* Springer-Verlag, 2003.

[26] Onn Shehory and Sarit Kraus. Formation of overlapping coalitions for percedence-ordered task execution among autonomous agents. In *In the Proc. of the 2nd Internaction Conference on Multiagent Systems*, 1996.

[27] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, 1998.

[28] Y. Shoham and K. Leyton-Brown. *Foundations of Multiagent Systems: Algorithmic, Game-Theoretic and Logical Models of Competition and Cooperation*. Book in preparation.

[29] L Soh, C Tsatsoulis, and H Sevay. A satisficing, negotiated, and learning coalition formation architecture. In Victor Lesser, Charles Ortiz, and Milind Tambe, editors, *Distributed Sensor Networks: A multiagent perspective*. Kluwer Publishing, 2003.

[30] Junichi Yamamoto and Katia Sycara. A stable and efficient buyer coalition formation scheme for e-marketplaces. In *In the Proc. of the 5th Internaction Conference on Autonomous Agents*, 2001.

[31] M. Yokoo, V. Contizer, T. Sandholm, N. Ohta, and A. Iwaski. Coalitional games in open anonymous environments. In *the Proc. of the National Conference on Artificial Intelligence(AAAI)*, pages 509–515, 2005.

[32] G. Zlotkin and J. Rosenschein. Coalition, cryptography and stability: mechanisms for coalition-formation in task oriented domains. In *In the Proc. of AAAI*, pages 432–437, 1994.