Coping with Selfish Behavior in Networks using Game Theory

by

Dejun Yang

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved June 2013 by the
Graduate Supervisory Committee:

Guoliang Xue, Chair
Andréa Richa
Arunabha Sen
Junshan Zhang

ARIZONA STATE UNIVERSITY

August 2013

ABSTRACT

While network problems have been addressed using a central administrative domain with a single objective, the devices in most networks are actually not owned by a single entity but by many individual entities. These entities make their decisions independently and selfishly, and maybe cooperate with a small group of other entities only when this form of coalition yields a better return. The interaction among multiple independent decision-makers necessitates the use of game theory, including economic notions related to markets and incentives.

In this dissertation, we are interested in modeling, analyzing, addressing network problems caused by the selfish behavior of network entities. First, we study how the selfish behavior of network entities affects the system performance while users are competing for limited resource. For this resource allocation domain, we aim to study the selfish routing problem in networks with fair queuing on links, the relay assignment problem in cooperative networks, and the channel allocation problem in wireless networks. Another important aspect of this dissertation is the study of designing efficient mechanisms to incentivize network entities to achieve certain system objective. For this incentive mechanism domain, we aim to motivate wireless devices to serve as relays for cooperative communication, and to recruit smartphones for crowdsourcing. In addition, we apply different game theoretic approaches to problems in security and privacy domain. For this domain, we aim to analyze how a user could defend against a smart jammer, who can quickly learn about the user's transmission power. We also design mechanisms to encourage mobile phone users to participate in location privacy protection, in order to achieve *k*-anonymity.

Dedicated to my family

ACKNOWLEDGEMENTS

First, I would like to gratefully and sincerely thank Dr. Guoliang Xue for his guidance, patience, understanding, and most importantly, friendship during my PhD studies at Arizona State University. This dissertation would not have been possible without him and without the academic freedom and encouragement he has given me during the last six years.

I have been honored to have an amazing group of coauthors and collaborators, Satyajayant Misra, Xi Fang, Nan Li, Afsheen Irani, Jian Tang, Pritam Gundecha, Jin Zhang, Dr. Krishnaiyan Thulasiraman, Ying Xiao, and Ravi Gottapu. Their unselfish help, insights, and feedbacks helped me improve my knowledge in the area. I would like to thank my dissertation committee members, Dr. Andréa Richa, Dr. Arunabha Sen, and Dr. Junshan Zhang for their advise and comments. I am also grateful to my colleagues and friends: Seung Dong Hong, Jun Shen, Lingjun Li, Xinxin Zhao, Gabriel Silva, Vishnu Kilari, Ziming Zhao, Yashu Liu, and Xiang Zhang for their friendship and care, which helped me stay sane through these years.

Finally and most importantly, none of this would have been possible without the love and patience of my girlfriend and my family. I would like to thank my beloved girlfriend Jing for her endless support, patience, encouragement, and unwavering love during the last 9 years. I am very thankful to my parents, Shengzhu Cui and Zhiguo Yang, for their unconditional love and support throughout my life. I thank my brother Degang Yang. With him taking care of my parents in China, I could focus on my research and graduate studies here in US.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

Introduction

Network problems have been addressed using a central administrative domain with a single objective. The users in the network are always assumed to be obedient. However, the devices in most networks are not own by a single entity but by many individual entities. These entities make their decisions independently and selfishly, and maybe cooperate with a small group of other entities only when this form of coalition yields a better return. The interaction among multiple independent decision-makers necessitates the use of game theory, including economic notions related to markets and incentives.

Game theory is the study that analyzes the strategic interactions among autonomous decision-makers, whose actions have mutual, probably conflicting, consequences. Originally developed to model problems in the field of economics, game theory has recently been applied to network problems, in most cases to solve the resource allocation problems in a competitive environment. The reason that game theory is an appropriate choice for studying network problems is multifold. First, entities in the network are autonomous agents, making decisions only for their own interests. Game theory provides us sufficient theoretical tools to analyze the network users' behaviors and actions. Second, game theory primarily deals with distributed optimization, which often requires local information only. Thus it enables us to design distributed algorithms. Finally, auction, a market game of incomplete information, allows us to design mechanisms to provide incentives for network entities participating in tasks, which would not be achieved without sufficient participation.

1.1 Game Theory 101

Game theory [46] is a discipline aimed at modeling scenarios where individual decision-makers have to choose specific actions that have mutual or possibly conflict consequences. A game consists of three major components:

- **Players**: The decision makers are called players, denoted by a finite set $\mathcal{N} = \{1, 2, \ldots, n\}$.
- **Strategy**: Each player $i \in \mathcal{N}$ has a non-empty strategy set S_i. Let s_i denote the selected strategy by player i. A strategy profile s consists of all players' strategies, i.e., $s = (s_1, s_2, \ldots, s_n)$. Obviously, we have $s \in S = \times_{i \in \mathcal{N}} S_i$, where $\times$ is the Cartesian product.
- **Utility/Payoff**: The utility of player i is a measurement function, denoted by $u_i : S \mapsto \mathbb{R}$, on the possible outcome determined by the strategies of all players, where $\mathbb{R}$ is the set of real numbers.

The players of the game are assumed to be rational and selfish, which means each player is only interested in maximizing its own utility without respecting others' and the system's performance. Let s_{-i} denote the strategy profile excluding s_i. As a notational convention, we have $s = (s_i, s_{-i})$. We say that player i prefers s_i to s_i' if $u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$. When other players' strategies are fixed, player i can select a strategy, denoted by $b_i(s_{-i})$, which maximizes its utility function. Such a strategy is called a *best response* of player i. A strategy is called a *dominant strategy* of player i if, regardless of what other players do, the strategy earns player i a larger utility than any other strategy. In order to study the interactions among players, the concept of *Nash Equilibrium* (NE) is introduced. A strategy profile constitutes an NE if none of the players can improve its utility by unilaterally deviating from its current strategy.

Definition 1.1. *[Nash Equilibrium] A strategy profile* $s^{ne} = \{s_1^{ne}, s_2^{ne}, \ldots, s_N^{ne}\}$ *is called a Nash Equilibrium (NE), if for every player i, we have:*

$$u_i(s^{ne}) \geq u_i(s_{-1}^{ne}, s_i')$$

for every strategy $s_i' \in S$. □

To characterize and quantify the inefficiency of the system performance due to the lack of cooperation among the players, we use the concept of *price of anarchy* (POA) [77].

Definition 1.2. *[Price of Anarchy] The* price of anarchy *(POA) of a game is the ratio of the total utility achieved in a worst possible NE over that of the social optimum.* □

The POA in game theory is an analogue of the approximation ratio in combinatorial optimization. If a game has a POA lower bounded by $\alpha \leq 1$, it means that for any instance of the game, the system performance in *any* NE is at least α times the system performance in the optimal solution.

Games can be classified into two categories, *strategic form game (or static game)* and *extensive form game (or dynamic game)*. The strategic form game is a one-shot game. In this game, the players make their decisions simultaneously without knowing what others will do. On the contrary, the extensive form game represents the structure of interactions between players and defines possible orders of moves. The *repeat game* is a class of the extensive form game, in which each stage is a repetition of the same strategic game. At the beginning of each stage, players observe the past history of strategies before making decisions. The number of stages may be finite or infinite. The utility of each player is the accumulated utility through all the stages. Therefore, players care not only the current utility but also the future utilities.

The *Stackelberg game* is an extensive form game, which is used to model the competition between one player, called the leader, and a set of players, called the followers. In this game, the leader takes action first and then the followers take actions. The leader knows ex ante that the followers observe its action and take actions accordingly. The NE in the Stackelberg game is called *Stackelberg Equilibrium*.

		Player B	
		L	R
Player A	U	3,2	**6,5**
	D	4,3	8,2

Table 1.1: Utility matrix: the first number in each cell is the utility of Player A, while the second is the utility of Player B.

We illustrate these concepts using a simple example given in Table 1.1. Note that this example is just for the illustration of an SE and not an instance of the problem studied. Assume that Player A is the leader, and Player B is the follower. If A plays strategy U, B would play strategy R, as it gives player B a utility of 5 (as opposed to a utility of 2 should B play strategy L). This leads to a utility of 6 for player A. If A plays strategy D, B would play strategy L, as it gives player B a utility of 3 (as opposed to a utility of 2 should B play strategy R). This leads to a utility of 4 for player A. Hence A would play strategy U, since doing so would result in a utility of 6 compared to 4 by playing strategy D. As explained before, B would play R if A plays U. Therefore the Stackelberg Equilibrium of this game is (U, R).

As game theory studies interactions between rational and intelligent players, it can be applied to the economic world where people interact with each other in the market. The marriage of game theory and economic models yields interesting games and fruitful theoretical results in microeconomics and auction theory. Auction is a decentralized market mechanism for allocating resources. The essence of auction is a game of incomplete information, where the players are the bidders, the strategies are the bids, and both allocations and payments are functions of the bids. In an auction mechanism, each bidder i has some private information t_i, called its *type*, and its strategy is the bid b_i. A mechanism then computes an *output* $o = o(b_1, b_2, \ldots, b_n)$ and a *payment* vector $p = (p_1, p_2, \ldots, p_n)$, where $p_i = p_i(b_1, b_2, \ldots, b_n)$ is the money given to the participating agent i. For each possible output o, bidder i's valuation is $v_i(t_i, o)$. The utility of bidder i is $u_i(t_i, o) = v_i(t_i, o) + p_i$.

Based on the number of objects auctioned on the market, auctions can be categorized into *single-object auction* and *multi-object auction*. Two basic single-object auction schemes are the *first-price auction* and the *second-price auction*. In the first-price auction, the auctioneer grants the item to the highest bidder and charges the highest bid. In the second-price auction, also known as Vickrey auction, the auctioneer grants the item to the highest bidder, but charges the second highest bid. Multi-object auction can be *homogeneous auction* or *heterogeneous auction*, depending on whether the objects are identical.

There are four desirable properties while designing an auction scheme:

- **Computation Efficiency**: The outcome of the auction can be computed in polynomial time.
- **Individual Rationality**: Each agent can expect a non-negative profit.
- **System Efficiency**: An auction is system-efficient if the sum of valuations of all bidders is maximized.
- **Truthfulness**: An auction is truthful if revealing true private valuation is the dominant strategy for each bidder. In other words, no bidder can improve its utility by submitting a bid different from its true valuation, no matter how others submit.

For double auction, one more property is desirable, **Budget Balance**. An auction is budget-balanced if the payment collected from the buyers is at least as much as the payment paid to the sellers.

1.2 Overview and Contributions

This dissertation studies and addresses a number of important network problems. It focuses on three problem domains: resource allocation, incentive mechanism design, and security and privacy. The common thread throughout the research on these different domains is applying game theory to modeling, analyzing, and solving problems caused by the selfish behavior from network entities.

In general, for the resource allocation domain, we analyze how the selfish behavior of users affect the system performance while users are competing against each other for limited resource. In this domain, we consider three problems: 1) Selfish Routing in Networks with Fair Queuing [161, 162]; 2) Relay Assignment for Cooperative Networks [156, 159]; and 3) Channel Allocation in Non-Cooperative Multi-Radio Multi-Channel Wireless Networks [158].

Contributions:

- We model the studied problem as a non-cooperative game.
- We prove the existence of Nash Equilibria. If an NE does not exist, we design a charging scheme to influence users to converge to an NE.
- We quantitatively measure the degradation of the system performance caused by the selfish behavior from users.
- For Problem 2), we design efficient schemes to induce users to converge an NE, which meanwhile is also the optimal solution.

For the second domain, we design incentive mechanisms to stimulate network entities to achieve certain objective, which may not be possible otherwise. In this domain, we consider two problems: 1) Motivating Wireless Devices for Cooperative Communications [157]; and 2) Recruiting Smartphones for Crowdsourcing [163].

Contributions:

- We design auction-based incentive mechanisms to motivate network entities.
- We prove that the design incentive mechanisms satisfy several desirable economic properties.

For the security and privacy domain, we adopt different game theoretic approaches to either help the user defend against the attacker or protect users' privacy. In this domain, we consider two problems: 1) Coping with A Smart Jammer [164]; and 2) Motivating Mobile Users for K-Anonymity Location Privacy [160].

Contributions:

- We derive an optimal strategy for the user to defend against a smart jammer.
- We design mechanisms to incentivize mobile users to achieve k-anonymity.

Part I

Resource Allocation

Chapter 2

Selfish Routing in Networks with Fair Queuing

Routing is the process of selecting paths in a network along which to send data packets. In communication networks, the choice of a route between a source-destination pair has a significant bearing on the resulting bandwidth. For example, in peer-to-peer networks, there may be several pairs of peers sharing volumes of data between each other. The objective of each pair of peers, considered as a user, is to send as many packets as possible through the network while competing for network resources against other users. With this selfish objective, a user will change its path if the new path provides a larger bandwidth value even at the cost of other users. Since multiple users may compete for the bandwidth on the same link, it is necessary to have a congestion control scheme to allocate bandwidth among competing users. Hence max-min fair bandwidth allocation has been widely adopted as a congestion control scheme at the link level [21, 34, 69, 89, 92, 105, 125]. The max-min fair bandwidth allocation scheme treats all paths passing through a link equally and assigning an equal share of bandwidth to each of them unless a path receives less bandwidth at another link.

2.1 Introduction

In this work, we model the network using a directed graph, and present a game theoretic study of non-cooperative routing under max-min fair congestion control, where the goal of each user is to maximize the bandwidth of its chosen path. We call this problem the Maximal-Bandwidth Routing problem. Two questions arise while addressing this problem: *How can a user efficiently find a path with maximum bandwidth under max-min fair congestion control, when the paths of all other users are given?* and *Will the network* *oscillate or converge to a stable state?* The first question is critical to our convergence analysis, since it directly affects the convergence speed. It is also an independent problem to study, as we will point out later that the strong correlation among competing paths makes the calculation of available bandwidth on each link challenging. The second question is important because oscillation among different paths introduces dramatic overhead, consuming network resources. This work answers both questions.

In answering the first question, we introduce the concept of *observed available bandwidth* and prove that it can accurately predict the bandwidth of a path. In answering the second question, we model the routing problem as a non-cooperative game and employ game theoretic tools to analyze the interaction among users. This question boils down to *the existence of Nash Equilibria* and *the convergence of the game*. One major challenge arises while answering these questions. While selecting a new path, the available bandwidth of a link may depend on the bandwidth of existing paths of other users. However, the bandwidths of these paths in turn depend on the bandwidth of the new path. Therefore the problem is significantly more involved than the traditional maximum capacity path problem.

The major contributions of this work are as follows:

- We formulate the Maximal-Bandwidth Routing problem (MAXBAR) as a non-cooperative strategic game where each player makes the routing decision selfishly to maximize its bandwidth. In Section 2.5, we generalize it to the case where each user has a bandwidth demand.
- We prove the existence of Nash Equilibria in the MAXBAR game, where no player has any incentive to deviate from its chosen path. We also prove a lower bound and an upper bound on the *price of anarchy* of the MAXBAR game, which is a

concept quantifying the system degradation due to selfish behavior of users. As a byproduct, this also gives an approximation to the social optimal solution to the MAXBAR problem.

- We introduce a novel concept of observed available bandwidth to compute the available bandwidth on each link. It empowers the efficient computation of the best response strategy for each user. This is non-trivial, as the traditional widest path algorithm cannot be directly applied due to the mutual influence between paths sharing common links [89].
- We investigate the behavior and incentives of the players in the game and present a game based algorithm to compute an NE. We prove that by following the natural game course, the MAXBAR game converges to an NE.

The rest of this work is organized as follows. In Section 2.2, we present a brief overview of related work. In Section 2.3, we describe our system model, present the MAXBAR problem where each user would like to have as much bandwidth as possible, and formulate it as a non-cooperative game. In Section 2.4, we prove the existence of Nash Equilibria and quantify the inefficiency incurred by the lack of cooperation via price of anarchy, present an efficient algorithm to select a path with maximum bandwidth in a max-min fair network with multiple users, and provide a comprehensive analysis of the MAXBAR game and prove the convergence to an NE. In Section 2.5, we study a generalization of the MAXBAR problem where each user has a bandwidth demand, instead of aiming to have as much bandwidth as possible. In Section 2.6, we present numerical results on randomly generated networks. These results show that the game converges to an NE rapidly (within 7 iterations on average and 10 iterations at worst) and achieves better fairness compared with other algorithms. We conclude this work in Section 2.7.

2.2 Related Work

Congestion control is a critical task in communication networks to address the issue of fairly and optimally allocating resources, bandwidth in particular, among multiple competing users. Max-min fair bandwidth allocation has been proposed as one of the congestion control schemes [11, 69]. This scheme was first presented in [69]. The author also proved the optimality and the uniqueness of the allocation. In [34], Demers proposed a fair queuing scheduler, which is employed on each gateway, to implement a max-min fair network. In [89], Ma *et al.* studied how to route in max-min fair networks to improve the total throughput of the network. To calculate the max-min fair bandwidth for each path, they also presented a centralized algorithm. Note that the information used by the routing algorithm is abstract and only an estimate of the accurate available bandwidth. Showing that computing the max-min fair bandwidth requires global information, Mayer *et al.* [92] designed a local distributed scheduling algorithm to approximate max-min fair bandwidth allocation.

Chen and Nahrstedt [21] extended the concept of max-min fairness to the routing level, since the max-min fair bandwidth allocation scheme was proposed to achieve fairness at link level. They defined the *fairness-throughput* and introduced a new set of relational operators to compare two different feasible bandwidth allocations at routing level. The fairness-throughput performance of the bandwidth allocation is maximized if and only if such an allocation is the largest under the relational operator. They also proposed a max-min fair routing algorithm to select a path for the new user to maximize the minimum bandwidth allocated to all users. In [105], Nace considered a model, where the routing is splittable, and gave a linear programming based algorithm to compute the optimal max-min fair bandwidth allocation. Schapira *et al.* [125] and Godfrey *et al.* [54] studied the efficiency and incentive compatibility of different congestion control schemes in the network where users' paths are fixed. They also presented a family of congestion control protocols called Probing Increase Educated Decrease and showed that by following any of these protocols, the network converges to a fixed point.

All the previous works mainly focused on either the case where paths are fixed [34, 69, 92] or the case where routing aims to improve the total performance [21, 54, 89, 105, 125]. In contrast, the objective of our work is to investigate the scenario where each user in the network is able to adapt its routing decision based on the current environment and driven by its own selfish objective. The game formulation of this scenario falls into the category of *bottleneck game* [8]. There are also important works on stable routing in the literature [48, 58, 59]. However, these works do not consider max-min fair bandwidth allocation in their models.

2.3 Network Model and Problem Formulation

We first describe the network model and discuss the well known max-min fair congestion control scheme. We then formulate the problem studied in this work.

2.3.1 Network Model

We model the network by a directed edge-weighted graph denoted by $G = (V, E, b)$, where V is the set of *n nodes*, E is the set of *m links*, and b is a weight function such that $b(e) = b(v, w) > 0$ is the *bandwidth of link* $e = (v, w) \in E$. In the network, there is a collection $\mathscr{U} = \{1, 2, \ldots, N\}$ of *users*. User $i \in \mathscr{U}$ needs to transmit packets from a source node $s_i \in V$ to a destination node $t_i \in V$ over an s_i–t_i path. An s–t path in the network consists of an ordered sequence of vertices s=v_0, v_1, …, v_q=t, where $(v_l, v_{l+1}) \in E$ for $0 \leq l < q$. We denote such a path by v_0-v_1-$\cdots$-v_q. We are only interested in simple paths–for which the nodes in the sequence are distinct. Although there may be multiple s_i–t_i paths, at any given time, user i uses only one path, which is denoted by P_i. We denote the set of paths currently used by the users as $\mathscr{P} = \{P_1, P_2, \ldots, P_N\}$. We denote the set of users currently sharing link e by $\mathscr{U}_e(\mathscr{P})$, i.e., $\mathscr{U}_e(\mathscr{P}) = \{i | i \in \mathscr{U} \text{ and } e \in P_i\}$.

For routing approach, we will use link-state source routing algorithms as in [89]. In such routing schemes, each node knows the network topology and the state information on each link [9, 134]. Thus it is possible for the node to select its path. In this work, we consider best-effort flows [89] and assume that every source node always has sufficient data to transmit.

2.3.2 Congestion Control

Since multiple users are competing for bandwidth resources, congestion control is necessary for the management of bandwidth. The employed congestion control needs to satisfy two requirements: 1) the bandwidth allocation is fair and 2) the bandwidth is fully allocated. A simple way to allocate the bandwidth of a link to multiple competing paths is to share it equally among them. However, some paths can use only less than the equal share (due to some bottlenecks), while some can use more. Hence, equal allocation is not desirable. In this work, we assume that at the link level, *max-min fair bandwidth allocation* (also known as fair queuing) [34, 69] is used for congestion control. Max-min fair bandwidth allocation has been recognized as the optimal throughput-fairness definition [69, 92]. Intuitively, if there are multiple users sharing a common link, each user will get a “fair share” of the link’s bandwidth. If some user cannot use up its fair share bandwidth because it has a lower share assigned on another link, the excess bandwidth is “fairly” split among all other users of this link. Such a network with max-min fair congestion control at the link level is called a *max-min fair network*. We denote the bandwidth allocated to user i in a max-min fair network by $b_i(\mathscr{P})$ (how to compute the value of $b_i(\mathscr{P})$ will be shown later). Since user i will use only one path at any given time, we will say *the bandwidth of user i* instead of *the bandwidth of user i's path* when the path is clear from the context. We use $\mathbf{b}(\mathscr{P}) = (b_1(\mathscr{P}), b_2(\mathscr{P}), \ldots, b_N(\mathscr{P}))$ to denote the Max-min Fair Bandwidth Allocation (*MFBA*) given users' paths $\mathscr{P}$. The uniqueness of *MFBA* has been proved in [125]. While assigning the bandwidth to each path P_i, there must exist at least one link that keeps the path from obtaining more bandwidth. We call such link a *bottleneck* of path P_i. Note that there could be more than one bottleneck for a path. We use $\mathscr{B}_i(\mathscr{P})$ to denote the *set of all bottlenecks* of path P_i. Each bottleneck e of path P_i has two important properties, which can be mathematically expressed as follows:

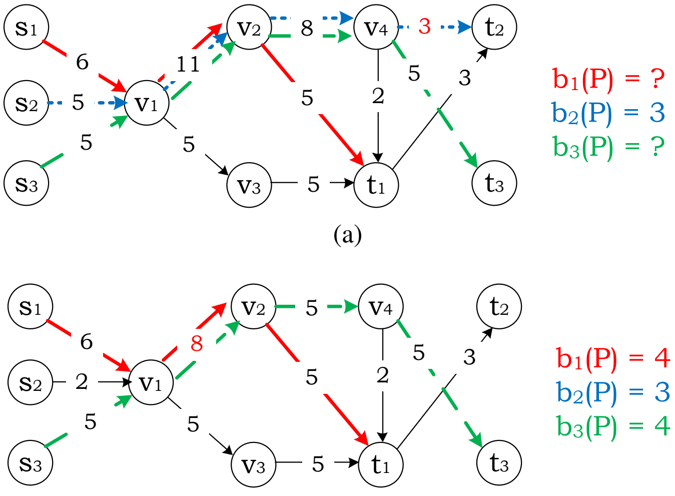
1. $\sum_{j \in \mathscr{U}_e(\mathscr{P})} b_j(\mathscr{P}) = b(e)$,
2. $b_i(\mathscr{P}) \geq b_j(\mathscr{P}), \forall j \in \mathscr{U}_e(\mathscr{P})$.

Property 1) means that link e is saturated. We call a link *saturated* if its bandwidth is fully allocated. This property is obvious as otherwise e is not a link that keeps P_i from obtaining more bandwidth. Property 2) states that there is no path being allocated more bandwidth than P_i on link e. The reason is that if there exists another path P_j allocated more bandwidth, P_i could equally share the bandwidth with P_j due to max-min fair bandwidth allocation and obtain more bandwidth. These two properties have also been proved in Lemma 3 of [21] and Lemma 3 of [69].

Algorithms for calculating the bandwidth allocation for each path in a max-min fair network have been proposed in [69, 89]. To make our work self-contained, we illustrate the pseudo code in Algorithm 1. For detailed description and correctness proof, we refer the readers to [69, 89].

Algorithm 1: $ComB(G, b, \mathscr{P}, \mathscr{U})$

```
input  : Network $G$, path set $\mathscr{P}$ and user set $\mathscr{U}$
output: $b_i(\mathscr{P})$ for all $i \in \mathscr{U}$
$b_i(\mathscr{P}) \leftarrow 0, \forall i \in \mathscr{U}$;
repeat
    Let $\bar{e} := argmin_{e \in E} \frac{b(e)}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}$ in $G(V, E, b)$;
    $b_{temp} \leftarrow \frac{b(\bar{e})}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}$;
    foreach player $i \in \mathscr{U}_{\bar{e}}(\mathscr{P})$ do
        $b_i(\mathscr{P}) \leftarrow b_{temp}$;
        foreach $e \in P_i$ do
            $b(e) \leftarrow b(e) - b_i(\mathscr{P})$;
            if $b(e) = 0$ then $E \leftarrow E \setminus \{e\}$;
        end
        $\mathscr{P} \leftarrow \mathscr{P} \setminus \{P_i\}$;
    end
until $\mathscr{P} = \emptyset$;
return $b_i(\mathscr{P})$ for all $i \in \mathscr{U}$;
```



(a)

(b)

Figure 2.1: Example with 3 users. $P_1 = s_1$-v_1-v_2-t_1 (red solid), $P_2 = s_2$-v_1-v_2-v_4-t_2 (blue dotted), and $P_3 = s_3$-v_1-v_2-v_4-t_3 (green dashed).

The basic idea of Algorithm 1 is that in each iteration, we find a *global bottleneck* $\bar{e}$, which is defined as the link having the least equal share, i.e., $\bar{e} = \arg\min_{e \in E} \frac{b(e)}{|\mathscr{U}_e(\mathscr{P})|}$. We allocate the equal share of $b(\bar{e})$ to all users in $\mathscr{U}_{\bar{e}}(\mathscr{P})$. Then all the paths of users in

$\mathscr{U}_{\bar{e}}(\mathscr{P})$ are removed from the network. The link bandwidths are reduced by the bandwidth consumed by the removed users. The above procedure is repeated until all the paths have been assigned bandwidth and removed from the network.

To illustrate the idea of Algorithm 1, we compute the bandwidth for the example in Figure 2.1. In Figure 2.1(a), (v_4, t_2) is the $\bar{e}$ selected in the first iteration and $\frac{b(v_4,t_2)}{|\mathscr{U}_{(v_4,t_2)}(\mathscr{P})|} = 3$. Since user 2 (blue dotted) is the only one using link (v_4, t_2), we set $b_2(\mathscr{P}) = 3$, remove path P_2 from the network and subtract the bandwidth from all the links along path P_2 (blue dotted). In the resulting network shown in Figure 2.1(b), (v_1, v_2) is selected as $\bar{e}$. There are two paths, P_1 (red solid) and P_3 (green dashed), sharing link (v_1, v_2). Each of them obtains bandwidth $\frac{b(v_1,v_2)}{|\mathscr{U}_{(v_1,v_2)}(\mathscr{P})|} = 4$. We set $b_1(\mathscr{P}) = b_3(\mathscr{P}) = 4$, and remove path P_1 and path P_3. Since there is no more paths left, the algorithm terminates.

2.3.3 Problem Formulation

In this work, we study the problem of routing in a max-min fair network with multiple *selfish* users, where each user selects its path to maximize its bandwidth. We call this problem the **MAX***imal*-**BA***ndwidth* **R***outing* (MAXBAR) problem. We are interested in the following questions:

Q1. How does each user select the path to maximally increase its bandwidth?

Q2. Will the routing oscillate forever or converge to a stable state, where no user can increase its bandwidth by unilaterally changing its path?

Q3. If the answer to Q2 is converging to a stable state, how is the social welfare in the stable state compared to that in the optimal solution with centralized control?

The MAXBAR problem can be formulated as a non-cooperative game, called MAXBAR game, as follows. Each user is a *player* in this game. We define the *strategy* of player i as its path P_i. A strategy profile of all players is then $\mathscr{P}$. We denote the strategies except player i's by $\mathscr{P}_{-i}$. We define the *utility* of player i as the bandwidth $b_i(\mathscr{P})$ of path P_i. Since players are selfish but rational, each player makes independent routing decisions to maximize its own utility. When player i's path is not in the network, we use $\mathbf{b}(\mathscr{P}_{-i})$ to denote the *MFBA* and $b_j(\mathscr{P}_{-i})$ to denote the bandwidth of path P_j, where $b_i(\mathscr{P}_{-i}) = 0$ as a technical convention. Let $\mathscr{P}|^i P_i'$ denote the path profile where player i changes its path to P_i' and others remain the same. When the context is clear, we use $\mathscr{P}|^i$ instead of $\mathscr{P}|^i P_i'$ for notational simplicity. Let $\mathbf{b}(\mathscr{P}|^i)$ and $b_j(\mathscr{P}|^i)$ denote the *MFBA* and the new bandwidth of user j's path. It is clear that $\mathscr{U}_e(\mathscr{P}|^i) = \mathscr{U}_e(\mathscr{P}_{-i}) \cup \{i\}$ if $e \in P_i'$ and $\mathscr{U}_e(\mathscr{P}|^i) = \mathscr{U}_e(\mathscr{P}_{-i})$ otherwise.

An important subproblem of the MAXBAR problem, which is of independent interest, is how to select a path to maximize the allocated bandwidth, given the network and other users' paths. This is known as *best response* in game theory.

Definition 2.1. *[Best Response Routing] Given other users' paths $P_1, \cdots, P_{i-1}, P_{i+1}, \cdots, P_N$, the* best response routing *for user i is a path P_i such that $b_i(\mathscr{P})$ is maximized over all s_i–t_i paths.* □

Finding a best response path for a user is not straightforward. As we learned from previous discussions, the allocated bandwidth for each path can be computed after considering the whole network topology and all path selections. Thus how to compute the available bandwidth on each link before the routing is known has not been solved yet. This problem was also studied in [89]. However, the authors only gave estimated information for each link and their algorithm is approximate. We will present an efficient solution to this problem in Section 2.4.2.

In order to study the strategic interactions of the players, we first introduce the concept of Nash Equilibrium [46].

Definition 2.2. *[Nash Equilibrium] A strategy profile $\mathscr{P}^{ne} = \{P_1^{ne}, P_2^{ne}, \dots, P_N^{ne}\}$ is called a Nash Equilibrium (NE), if for every player i, we have:*

$$b_i(\mathscr{P}^{ne}) \geq b_i(\mathscr{P}^{ne}|^i P_i')$$

for every strategy P_i', where P_i' is an s_i–t_i path. □

In other words, *in an NE, no player can increase its utility by unilaterally changing its strategy.*

The *social optimum* in the MAXBAR game is a strategy profile $\mathscr{P}^*$ such that the total utility, i.e. $\sum_{i \in \mathscr{U}} b_i(\mathscr{P}^*)$, is maximized among all $\mathscr{P}$. We use the concept of price of anarchy defined in [77] to quantify the system inefficiency due to selfishness.

Definition 2.3. *[Price of Anarchy] The* price of anarchy *(POA) of a game is the ratio of the total utility achieved in a worst possible NE over that of the social optimum.* □

Table 2.1 lists frequently used notations.

2.4 Analysis of the MAXBAR Game

2.4.1 Existence of Nash Equilibria

As a crucial step in proving the existence of NE, we show that every time a player changes its path, the minimum bandwidth of the players, whose bandwidths change, increases strictly.

Lemma 2.1. *Assume that player i unilaterally changes its path from P_i to P_i', such that $b_i(\mathscr{P}) < b_i(\mathscr{P}|^i)$. We have $\min_{j \in \mathscr{U}_\downarrow \cup \mathscr{U}_\uparrow} b_j(\mathscr{P}|^i) > \min_{j \in \mathscr{U}_\downarrow \cup \mathscr{U}_\uparrow} b_j(\mathscr{P})$, where $\mathscr{U}_= = \{j \in$*

Table 2.1: Frequently used Notations

Notation	Description
G	graph representing the network
V, E	node set, link set
v, w	node
$e, \bar{e}$	link and global bottleneck
$b(e)$	bandwidth of link e
$\mathscr{U}, N$	user (player) set, number of users (players)
i, j, k, u	user (player)
s_i, t_i	source node and destination node of user i
P_i	path (strategy) of user i
$\mathscr{P}$	path (strategy) set of users
$\mathscr{P}_{-i}$	path (strategy) set of users except i
$\mathscr{P}\|^i P_i'$	path (strategy) set with user i's path changed to P_i'
$\mathscr{P}\|^i$	abbreviation of $\mathscr{P}\|^i P_i'$ when P_i' is clear from the context
$\mathscr{U}_e(\mathscr{P})$	set of users whose paths share link e for given $\mathscr{P}$
$b_i(\mathscr{P})$	bandwidth (utility) of user i for given $\mathscr{P}$
$\mathbf{b}(\mathscr{P})$	bandwidth (utility) vector of all users for given $\mathscr{P}$

$\mathscr{U} | b_j(\mathscr{P}) = b_j(\mathscr{P}|^i)\}$, $\mathscr{U}_\uparrow = \{j \in \mathscr{U} | b_j(\mathscr{P}) < b_j(\mathscr{P}|^i)\}$ and $\mathscr{U}_\downarrow = \{j \in \mathscr{U} | b_j(\mathscr{P}) > b_j(\mathscr{P}|^i)\}$. □

Proof. It is clear that $i \in \mathscr{U}_\uparrow$, since $b_i(\mathscr{P}) < b_i(\mathscr{P}|^i)$. First we claim that, *for any $j \in \mathscr{U}_\downarrow$, there exists $k \in \mathscr{U}_\uparrow$, such that $b_j(\mathscr{P}|^i) \geq b_k(\mathscr{P}|^i)$*. Let $e \in \mathscr{B}_j(\mathscr{P}|^i)$ be a bottleneck of P_j after player i changes its path. By Property 2) of bottleneck, we have $b_j(\mathscr{P}|^i) \geq b_k(\mathscr{P}|^i)$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i)$. Therefore, we only need to prove that there exists a player $k \in \mathscr{U}_e(\mathscr{P}|^i) \cap \mathscr{U}_\uparrow$. If $i \in \mathscr{U}_e(\mathscr{P}|^i)$, then we can take $k = i$. Next, we consider the case where $i \notin \mathscr{U}_e(\mathscr{P}|^i)$. Note that $\mathscr{U}_e(\mathscr{P}) \setminus \{i\} = \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$, since only player i changes its path. Therefore $i \notin \mathscr{U}_e(\mathscr{P}|^i)$ implies that $\mathscr{U}_e(\mathscr{P}|^i) \subseteq \mathscr{U}_e(\mathscr{P})$. Assuming to the contrary that $b_k(\mathscr{P}) \geq b_k(\mathscr{P}|^i)$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i)$, the total bandwidth usage on link e in $\mathbf{b}(\mathscr{P}|^i)$ is

$$b_j(\mathscr{P}|^i) + \sum_{k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}} b_k(\mathscr{P}|^i) < b_j(\mathscr{P}) + \sum_{k \in \mathscr{U}_e(\mathscr{P}) \setminus \{j\}} b_k(\mathscr{P}) \leq b(e),$$

where the first inequality follows from $j \in \mathscr{U}_{\downarrow}$ and $\mathscr{U}_e(\mathscr{P}|^i) \subseteq \mathscr{U}_e(\mathscr{P})$, the second inequality follows from the feasibility of $\mathbf{b}(\mathscr{P})$. This contradicts the fact that e is a bottleneck, and proves the existence of player k in the case where $i \notin \mathscr{U}_e(\mathscr{P}|^i)$.

In summary, for any $j \in \mathscr{U}_{\downarrow}$, there exists $k \in \mathscr{U}_{\uparrow}$ such that

$$b_j(\mathscr{P}) > b_j(\mathscr{P}|^i) \geq b_k(\mathscr{P}|^i) > b_k(\mathscr{P}). \tag{2.1}$$

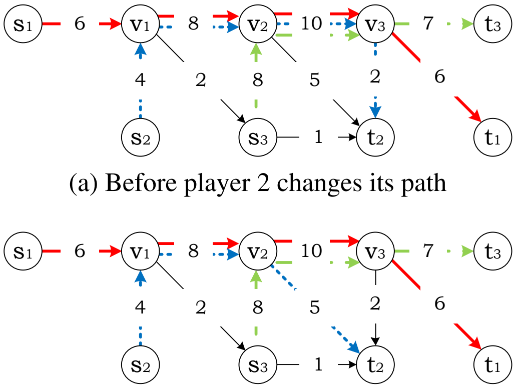
Following inner pair of (2.1), we know that

$$\min_{j \in \mathscr{U}_{\downarrow} \cup \mathscr{U}_{\uparrow}} b_j(\mathscr{P}|^i) = b_{k_1}(\mathscr{P}|^i)$$

for some player $k_1 \in \mathscr{U}_{\uparrow}$. Following outer pair of (2.1), we know that

$$\min_{j \in \mathscr{U}_{\downarrow} \cup \mathscr{U}_{\uparrow}} b_j(\mathscr{P}) = b_{k_2}(\mathscr{P})$$

for some player $k_2 \in \mathscr{U}_{\uparrow}$. Since $k_1 \in \mathscr{U}_{\uparrow}$, we know that $b_{k_1}(\mathscr{P}|^i) > b_{k_1}(\mathscr{P}) \geq b_{k_2}(\mathscr{P})$. Hence this lemma holds. □



(a) Before player 2 changes its path

(b) After player 2 changes its path

Figure 2.2: Example for Lemma 2.1.

We use the example in Figure 2.2 to illustrate the meaning of Lemma 2.1. In this example, we have three players: player 1 (red solid), player 2 (blue dotted) and player 3 (green dashed). From Figure 2.2(a) to Figure 2.2(b), player 2 changes its path from $P_2 = s_2$-v_1-v_2-v_3-t_2 to $P_2' = s_2$-v_1-v_2-t_2. Before the change, $b_1(\mathscr{P}) = 4$, $b_2(\mathscr{P}) = 2$, and $b_3(\mathscr{P}) = 4$. After the change, $b_1(\mathscr{P}|^2) = 4$, $b_2(\mathscr{P}|^2) = 4$, and $b_3(\mathscr{P}|^2) = 6$. In this example, $\mathscr{U}_= = \{1\}$, $\mathscr{U}_\uparrow = \{2,3\}$, and $\mathscr{U}_\downarrow = \emptyset$. We have $\min\{b_2(\mathscr{P}|^2), b_3(\mathscr{P}|^2)\} > \min\{b_2(\mathscr{P}), b_3(\mathscr{P})\}$.

We now prove the existence of NE in the MAXBAR game.

Theorem 2.1. *There exists at least one NE in the MAXBAR game.* □

Proof. At every stage of the game, we arrange the bandwidth values of the paths lexicographically in a non-decreasing order, resulting in a vector $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_N)$. In this vector, the minimum bandwidth b_1 is at the most significant coordinate. We have $b_\kappa \le b_{\kappa+1}$ for $1 \le \kappa < N$. For any two vectors $\vec{\mathbf{b}}_l = (b_1, b_2, \ldots, b_N)$ and $\vec{\mathbf{b}}_l' = (b_1', b_2', \ldots, b_N')$, $\vec{\mathbf{b}}_l < \vec{\mathbf{b}}_l'$ in lexicographic order if and only if:

1) $b_1 < b_1'$, or

2) $\exists\, 1 < \tau \le N$ s.t. $b_\kappa = b_\kappa'$ for $1 \le \kappa < \tau$ and $b_\tau < b_\tau'$.

By Lemma 2.1, we conclude that every time a player changes its path, the ordering $\vec{\mathbf{b}}_l$ increases lexicographically. We know that there are a finite number of paths for each player. Thus the number of different strategy profiles is finite as well. As each strategy profile corresponds to one vector, we pick the one corresponding to the largest vector as the strategies for the players. We conclude that such strategy profile is an NE as no player can improve its utility by unilaterally changing its strategy. □

While we know the existence of NE, there are still open questions to answer. How to *efficiently* find a path with maximum bandwidth in a max-min fair network? Will the

MAXBAR game converge to an NE? We will answer these questions in Sections 2.4.2 and 2.4.3, respectively.

Now, we quantify the worst-case "penalty" incurred by the lack of cooperation among the players in this game using the concept of *price of anarchy* (POA). Recall that POA is the ratio of the total bandwidth of the worst NE to the total bandwidth of the social optimum among all strategies.

Theorem 2.2. *For the MAXBAR game, $\frac{1}{N} \leq POA \leq \frac{2}{N}$.* □

Proof. We prove this theorem by proving the lower bound in Lemma 2.2 and the upper bound in Lemma 2.3. □

Lemma 2.2. *For the MAXBAR game, $POA \geq \frac{1}{N}$.* □

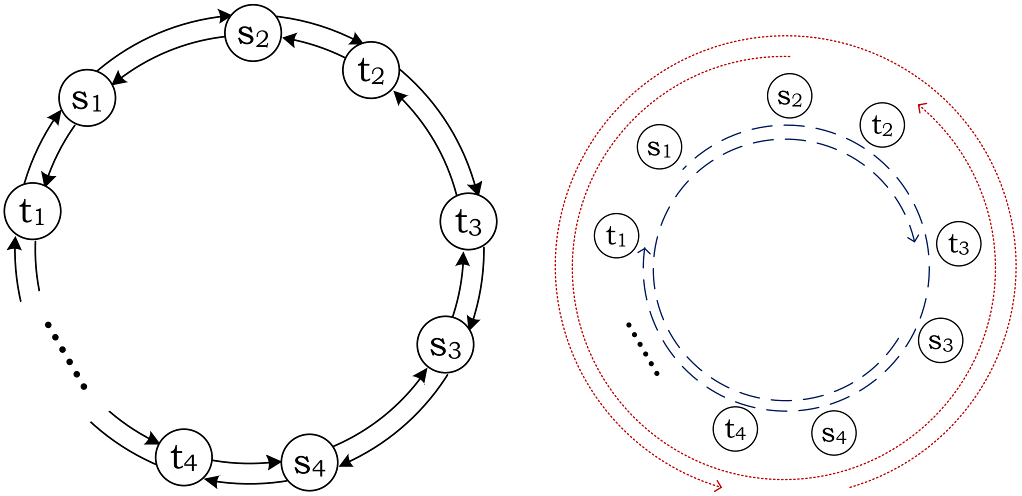
Proof. Let $\mathscr{P}^{ne} = \{P_1^{ne}, P_2^{ne}, \ldots, P_N^{ne}\}$ be any NE of the MAXBAR game. Let $\mathscr{P}^* = \{P_1^*, P_2^*, \ldots, P_N^*\}$ be the social optimum. We first claim that $b_i(\mathscr{P}^{ne}) \geq \frac{b_i(\mathscr{P}^*)}{N}$ *for any player i, where $b_i(\mathscr{P}^*)$ is the bandwidth of P_i^* in the social optimum.* Since $\mathscr{P}^{ne}$ is an NE, no player has any incentive to change its path, i.e.,

$$b_i(\mathscr{P}^{ne}) \geq b_i(\mathscr{P}|^i P_i^*) \geq \frac{b(e^*)}{N}, \tag{2.2}$$

where e^* is a bottleneck of P_i^* after player i unilaterally changes its path from P_i^{ne} to P_i^*. The second inequality follows from the fact that each link can be shared by at most N players. In the social optimum, we have $b_i(\mathscr{P}^*) \leq b(e)$ for any $e \in P_i^*$. Plugging it into (2.2), we proved our claim. Based on the claim, the total utility is

$$\textstyle\sum_{i \in \mathscr{U}} b_i(\mathscr{P}^{ne}) \geq \frac{\sum_{i \in \mathscr{U}} b_i(\mathscr{P}^*)}{N} = \frac{b(OPT)}{N} \tag{2.3}$$

for any NE, where $b(OPT)$ is the total bandwidth in the social optimum. Since (2.3) holds for any NE, we have $POA \geq \frac{1}{N}$. □



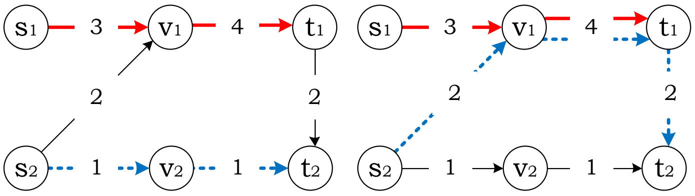
(a) Network topology (b) Players' strategies

Figure 2.3: Example where the POA is $\frac{2}{N}$.

Lemma 2.3. *For the MAXBAR game, POA $\leq \frac{2}{N}$.* □

Proof. We prove this lemma with the help of an example. Figure 2.3 depicts (partly) a network with N players. In this network, the bandwidth of each link is 1. As shown in Figure 2.3(a), all the source-destination pairs with odd indices are located counterclockwise on a ring topology, while those with even indices are located clockwise. The source and destination for the same player are next to each other. Clearly, there are only two s_i–t_i paths for each player i with odd index (resp. even index), the clockwise (resp. counterclockwise) path s_i-s_{i+1}-t_{i+1}-…-s_N-t_N-t_1-s_1-…-t_i and the counterclockwise (resp. clockwise) path s_i-t_i. As shown in Figure 2.3(b), if each player i with odd index chooses the clockwise s_i–t_i path and each player i with even index chooses the counterclockwise s_i–t_i path, the resulting strategy profile is an NE with $b_i(\mathscr{P}) = \frac{2}{N}$ for each player i. Because if any player i deviates from the current strategy and chooses the clockwise s_i–t_i path, it results in the bandwidth $\frac{2}{N+2}$. The total utility in this NE is 2.

Next, we consider the social optimum, where players with odd indices choose the counterclockwise paths and players with even indices choose the clockwise paths. The total utility is N. Hence the POA of the MAXBAR game is at most $2/N$. □



(a) Social optimum (b) Nash Equilibrium

Figure 2.4: A social optimum is not necessarily an NE.

Remark 1. Note that the social optimum in Figure 2.3 is also an NE. However, the example in Figure 2.4 shows that *a social optimum is not necessarily an NE.*

Remark 2. Efficient algorithms to compute a social optimum are still open. Simple brute fore algorithms may take exponential time, since the number of *s*-*t* paths for a single player is exponential in the size of the network.

Remark 3. We do not know whether the bounds for the POA are tight. Either proving the tightness of these bounds or deriving tighter bounds is a topic for future research.

Remark 4. We also studied the MAXBAR game under an undirected model [161]. The POA is proved to be exactly $\frac{1}{N}$.

2.4.2 Best Response Routing in Max-min Fair Networks

An important step in the MAXBAR game is for a player to decide whether it has any incentive to change its strategy unilaterally. Intuitively, it is natural for the player to unilaterally change its strategy to one that would give it the maximum utility. However, the utility of the chosen path depends on other players' strategies due to the competition among players sharing links with this chosen path. Obviously, the player can try all its strategies and pick the one giving it the maximum utility. However, this may take exponential time as the number of strategies of the user may not be polynomially bounded.

In this section, we introduce the novel concept of *observed available bandwidth* (formally defined later in this section) and prove the following facts: 1) the observed available bandwidth on all links can be computed in $O(Nm+N\log N)$ time; 2) the widest s_i-t_i path with regard to the observed available bandwidth is a best response routing for player i. Hence, player i can compute its best response routing in polynomial time. Therefore, *player i has an incentive to change its strategy if and only if the utility corresponding to its best response strategy is larger than that corresponding to its current strategy*. Given the challenges outlined at the beginning of this section, our results are significant. Although the facts are seemingly simple, the proofs are quite involved, which are the subjects of the rest of this section.

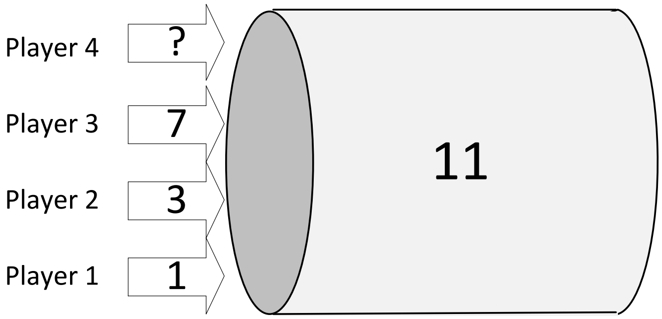


Figure 2.5: Link e with max-min fair bandwidth allocation, where there are three players before player 4 joins.

To get an intuition for calculating the available bandwidth, we take the link in Figure 2.5 as an example. In this example, we assume that player $i=4$ needs to find a path. Further assume that $\mathscr{U}_e(\mathscr{P}_{-i})=\{1,2,3\}$ and $b(e)=11$. Also, $b_1(\mathscr{P}_{-i})=1$, $b_2(\mathscr{P}_{-i})=3$, and $b_3(\mathscr{P}_{-i})=7$. After player i joins, it is clear that player 1 would not lose its bandwidth share, since it has less than the equal share, i.e., $b_1(\mathscr{P}_{-i})=1<\frac{11}{4}$. If player i competes the bandwidth with players 2 and 3 for the residual bandwidth of 10, each of them gets bandwidth of $\frac{10}{3}$. We know before i joins, player 2 only uses bandwidth of 3, which is less than $\frac{10}{3}$. Therefore, only i and 3 will compete for the residual bandwidth of 7 and get bandwidth of $\frac{7}{2}$ each.

To capture the process we conducted above, we introduce the concept of *observed available bandwidth*. Assume that all players except i have their paths chosen. Now player i needs to find a path with maximum bandwidth in the current network. For any link e and player $j \in \mathscr{U}_e(\mathscr{P}_{-i})$, let

$$\hat{\mathscr{U}_e}(\mathscr{P}_{-i}, j) = \{k | k \in \mathscr{U}_e(\mathscr{P}_{-i}) \text{ and } b_k(\mathscr{P}_{-i}) < b_j(\mathscr{P}_{-i})\}$$

denote the set of players who are using less bandwidth than player j on link e. Let

$$\tilde{\mathscr{U}_e}(\mathscr{P}_{-i}) = \{ \quad j | j \in \mathscr{U}_e(\mathscr{P}_{-i}) \text{ and } b_j(\mathscr{P}_{-i}) \geq \frac{b(e) - \sum_{k \in \hat{\mathscr{U}_e}(\mathscr{P}_{-i}, j)} b_k(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - |\hat{\mathscr{U}_e}(\mathscr{P}_{-i}, j)| + 1} \}$$

denote the set of players such that for any player j in this set, the new bandwidth $b_j(\mathscr{P}|^i)$ is at least as large as the bandwidth of the new path P_i' of player i. The *observed available bandwidth $b^o(e)$* of link $e \in E$ is

$$b^o(e) = \frac{b(e) - \sum_{j \in \mathscr{U}_e(\mathscr{P}) \setminus \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})} b_j(\mathscr{P}_{-i})}{|\tilde{\mathscr{U}_e}(\mathscr{P}_{-i})| + 1}. \tag{2.4}$$

If we first sort the paths according to their bandwidth values, then for each link e we can compute $\hat{\mathscr{U}_e}(\mathscr{P}_{-i}, 1), \hat{\mathscr{U}_e}(\mathscr{P}_{-i}, 2), \ldots, \hat{\mathscr{U}_e}(\mathscr{P}_{-i}, N)$, and $\tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$ in $O(N)$ additional time. Thus we can compute $b^o(e)$ for all links $e \in E$ in $O(Nm + N\log N)$ time. Accordingly, the *observed bandwidth* of the new path P_i' is

$$b_i^o(\mathscr{P}|^i) = \min_{e \in P_i'} b^o(e), \tag{2.5}$$

and the set of *observed bottlenecks* of path P_i' is

$$\mathscr{B}_i^o(\mathscr{P}|^i) = \arg\min_{e \in P_i'} b^o(e).$$

Considering the example in Figure 2.5, we have $\hat{\mathscr{U}_e}(\mathscr{P}_{-i},1) = \emptyset$, $\hat{\mathscr{U}_e}(\mathscr{P}_{-i},2) = \{1\}$, and $\hat{\mathscr{U}_e}(\mathscr{P}_{-i},3) = \{1,2\}$. The set $\tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$ is $\{3\}$. Therefore, $b^o(e) = \frac{11-1-3}{1+1} = \frac{7}{2}$.

The properties of the observed available bandwidth are summarized in the following four lemmas, which will be used in later proofs in the rest of this section.

Lemma 2.4. *Assume that $j \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. For all $u \in \mathscr{U}_e(\mathscr{P}_{-i})$, $b_u(\mathscr{P}_{-i}) \geq b_j(\mathscr{P}_{-i})$ implies $u \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$.* □

Proof. It is obvious that if $b_u(\mathscr{P}_{-i}) = b_j(\mathscr{P}_{-i})$, then $u \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. Next, we prove that if $b_u(\mathscr{P}_{-i}) > b_j(\mathscr{P}_{-i})$, then $u \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. Let $\mathscr{K} = \hat{\mathscr{U}_e}(\mathscr{P}_{-i},u) \setminus \hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)$. We have

$$\begin{aligned}
& b_u(\mathscr{P}_{-i}) - \frac{b(e) - \sum_{k \in \hat{\mathscr{U}_e}(\mathscr{P}_{-i},u)} b_k(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - |\hat{\mathscr{U}_e}(\mathscr{P}_{-i},u)| + 1} \\
= \;& b_u(\mathscr{P}_{-i}) - \frac{b(e) - \sum\limits_{k \in \hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)} b_k(\mathscr{P}_{-i}) - \sum\limits_{k \in \mathscr{K}} b_k(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - (|\hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)| + |\mathscr{K}|) + 1} \\
> \;& b_j(\mathscr{P}_{-i}) - \frac{b(e) - \sum\limits_{k \in \hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)} b_k(\mathscr{P}_{-i}) - |\mathscr{K}| b_j(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - (|\hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)| + |\mathscr{K}|) + 1} \qquad (2.6) \\
\geq \;& 0, \qquad (2.7)
\end{aligned}$$

where (2.6) follows from the fact that $k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \hat{\mathscr{U}_e}(\mathscr{P}_{-i},j)$ implies $b_k(\mathscr{P}_{-i}) \geq b_j(\mathscr{P}_{-i})$, and (2.7) follows from the fact that $j \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. Hence we have $u \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. □

Lemma 2.5. *If $j \in \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$, then $b_j(\mathscr{P}_{-i}) \geq b^o(e)$. If $j \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$, then $b_j(\mathscr{P}_{-i}) < b^o(e)$.* □

Proof. Let x be the player whose path has the minimum bandwidth in $\tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. Thus we have $\hat{\mathscr{U}_e}(\mathscr{P}_{-i},x) \subseteq \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}_e}(\mathscr{P}_{-i})$. For all $j \in \mathscr{U}_e(\mathscr{P}_{-i})$, if $b_j(\mathscr{P}_{-i}) \geq b_x(\mathscr{P}_{-i})$, it follows from Lemma 2.4 that $j \in \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. Thus we have $\hat{\mathscr{U}}_e(\mathscr{P}_{-i},x) \supseteq \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. Therefore $\hat{\mathscr{U}}_e(\mathscr{P}_{-i},x) = \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. Since $x \in \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$, we have

$$\begin{aligned} b_x(\mathscr{P}_{-i}) &\geq \frac{b(e) - \sum\limits_{j \in \hat{\mathscr{U}}_e(\mathscr{P}_{-i},x)} b_j(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - |\hat{\mathscr{U}}_e(\mathscr{P}_{-i},x)| + 1} && (2.8) \\ &= \frac{b(e) - \sum\limits_{j \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_j(\mathscr{P}_{-i})}{|\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})| + 1} \\ &= b^o(e). && (2.9) \end{aligned}$$

Therefore $b_x(\mathscr{P}_{-i}) \geq b^o(e)$. This implies the first part of the lemma, since $b_j(\mathscr{P}_{-i}) \geq b_x(\mathscr{P}_{-i})$ for any $j \in \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$.

Next, we prove the second part of the lemma. If $j \notin \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$, we know that $b_j(\mathscr{P}_{-i}) < b_x(\mathscr{P}_{-i})$. Now assume that y is the player whose path has the maximum bandwidth in $\mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. Then, we have $b_j(\mathscr{P}_{-i}) = b_y(\mathscr{P}_{-i})$, $\forall j \in \hat{\mathscr{U}}_e(\mathscr{P}_{-i},x) \setminus \hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)$. Let $\mathscr{J} = \hat{\mathscr{U}}_e(\mathscr{P}_{-i},x) \setminus \hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)$. We have

$$\begin{aligned} & b_y(\mathscr{P}_{-i}) - b^o(e) \\ =\ & b_y(\mathscr{P}_{-i}) - \frac{b(e) - \sum_{j \in \hat{\mathscr{U}}_e(\mathscr{P}_{-i},x)} b_j(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - |\hat{\mathscr{U}}_e(\mathscr{P}_{-i},x)| + 1} && (2.10) \\ =\ & b_y(\mathscr{P}_{-i}) - \frac{b(e) - (\sum\limits_{j \in \hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)} b_j(\mathscr{P}_{-i}) + \sum\limits_{j \in \mathscr{J}} b_j(\mathscr{P}_{-i}))}{|\mathscr{U}_e(\mathscr{P}_{-i})| - (|\hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)| + |\mathscr{J}|) + 1} \\ =\ & b_y(\mathscr{P}_{-i}) - \frac{b(e) - \sum_{j \in \hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)} b_j(\mathscr{P}_{-i}) - |\mathscr{J}| b_y(\mathscr{P}_{-i})}{|\mathscr{U}_e(\mathscr{P}_{-i})| - (|\hat{\mathscr{U}}_e(\mathscr{P}_{-i},y)| + |\mathscr{J}|) + 1} \\ <\ & 0, && (2.11) \end{aligned}$$

where (2.10) follows from (2.8) and (2.9), (2.11) follows from the fact that $y \notin \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. In addition, we know that $b_j(\mathscr{P}_{-i}) \leq b_y(\mathscr{P}_{-i}) < b^o(e)$, $\forall j \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. □

We now prove that the observed available bandwidth defined above accurately calculates the bandwidth on each link in the sense that after we choose a path with the maximum observed bandwidth and reallocate the bandwidth for each path using Algorithm 1, the new allocated bandwidth of the path is equal to its observed bandwidth.

We use proof by contradiction. The sketch of our proof is as follows. If the new allocated bandwidth of the path is not equal to its observed bandwidth, two cases may happen: 1) the path is allocated more bandwidth than the observed bandwidth, or 2) the path is allocated less bandwidth than the observed bandwidth. For each case, we show that it will lead to a chain reaction, which results in a contradiction. We analyze two phenomena that may occur and cause the chain reaction after a player chooses its new path based on the observed available bandwidth. In Lemma 2.6 (resp. Lemma 2.7), we show that the decrease (resp. increase) of the bandwidth of one path must be directly related to the increase (resp. decrease) of that of another path. More importantly, the relation between new bandwidth values of these two paths satisfies certain rules. In order to facilitate the understanding of these lemmas, an example is presented in Figure 2.6.

Lemma 2.6. *Let P_i' be the new s_i-t_i path chosen by player i based on the observed available bandwidth. We have the following:*

1. *If $b_i(\mathscr{P}|^i) < b_i^o(\mathscr{P}|^i)$, then $\exists k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$, such that*

   ***1**a) $b_k(\mathscr{P}|^i) > b_k(\mathscr{P}_{-i})$ and **1**b) $b_k(\mathscr{P}|^i) \leq b_i(\mathscr{P}|^i)$,*

   where $e \in \mathscr{B}_i(\mathscr{P}|^i)$ is a bottleneck of path P_i'.

2. *If $b_j(\mathscr{P}|^i) < b_j(\mathscr{P}_{-i})$ for some $j \in \mathscr{U}$, then $\exists k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}$, such that*

   ***2**a) $b_k(\mathscr{P}|^i) > b_k(\mathscr{P}_{-i})$ and **2**b) $b_k(\mathscr{P}|^i) \leq b_j(\mathscr{P}|^i)$,*

   where $e \in \mathscr{B}_j(\mathscr{P}|^i)$ is a bottleneck of path P_j after player i changes its path. □

Proof. We prove 1) and 2) separately:

We first prove 1). Assume that $b_i(\mathscr{P}|^i) < b_i^o(\mathscr{P}|^i)$.

By Property 2) of bottleneck, we know that $b_i(\mathscr{P}|^i) \geq b_j(\mathscr{P}|^i)$, $\forall j \in \mathscr{U}_e(\mathscr{P}|^i)$. Thus it suffices to prove that $\exists k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$, such that **1**a) holds. We prove this by contradiction. Assume that $b_j(\mathscr{P}|^i) \leq b_j(\mathscr{P}_{-i})$, $\forall j \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$. The total bandwidth usage on link e in $\mathbf{b}(\mathscr{P}|^i)$ is

$$\begin{aligned}
& \textstyle\sum_{j\in\mathscr{U}_e(\mathscr{P}|^i)} b_j(\mathscr{P}|^i) \\
= \;& b_i(\mathscr{P}|^i) + \sum_{j\in\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_j(\mathscr{P}|^i) + \sum_{j\in\mathscr{U}_e(\mathscr{P}_{-i})\setminus\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_j(\mathscr{P}|^i) \\
\leq \;& (|\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})|+1)b_i(\mathscr{P}|^i) + \sum_{j\in\mathscr{U}_e(\mathscr{P}_{-i})\setminus\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_j(\mathscr{P}|^i) \qquad (2.12) \\
< \;& (|\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})|+1)b_i^o(\mathscr{P}|^i) + \sum_{j\in\mathscr{U}_e(\mathscr{P}_{-i})\setminus\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_j(\mathscr{P}_{-i}) \qquad (2.13) \\
\leq \;& b(e), \qquad (2.14)
\end{aligned}$$

where (2.12) follows from Property 2) of bottleneck, (2.13) follows from the condition $b_i(\mathscr{P}|^i) < b_i^o(\mathscr{P}|^i)$ and the assumption $b_j(\mathscr{P}|^i) \leq b_j(\mathscr{P}_{-i})$, and (2.14) follows from (2.5) and (2.4). This contradicts the fact that $e \in \mathscr{B}_i(P_i')$, because e should be saturated in $\mathbf{b}(\mathscr{P}|^i)$ according to Property 1) of bottleneck. This completes the proof of 1).

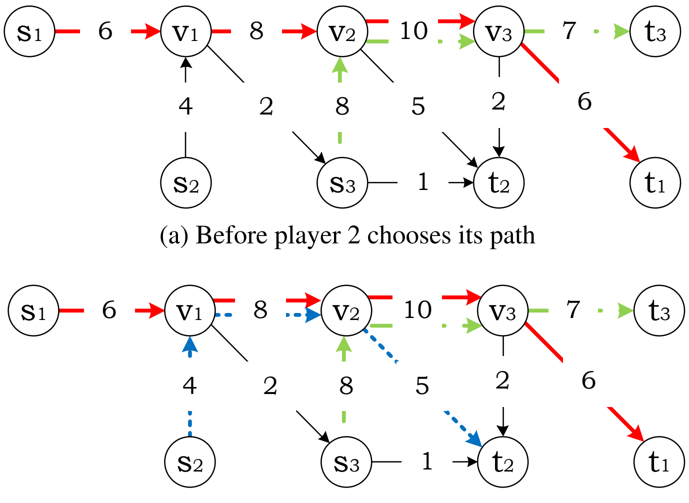
We now prove 2). Assume that $b_j(\mathscr{P}|^i) < b_j(\mathscr{P}_{-i})$.

By Property 2) of bottleneck, we know that $b_j(\mathscr{P}|^i) \geq b_k(\mathscr{P}|^i)$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i)$. Thus it suffices to prove that $\exists k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}$ such that **2**a) holds. The condition $b_j(\mathscr{P}|^i) < b_j(\mathscr{P}_{-i})$ implies that $i \neq j$. If $i \in \mathscr{U}_e(\mathscr{P}|^i)$, we can take $k = i$ and $b_i(\mathscr{P}|^i) > 0 = b_i(\mathscr{P}_{-i})$. Next, we consider the case where $i \notin \mathscr{U}_e(\mathscr{P}|^i)$. We prove **2**a) by contradiction. Assume that $b_k(\mathscr{P}|^i) \leq b_k(\mathscr{P}_{-i})$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}$. Note that $i \notin \mathscr{U}_e(\mathscr{P}|^i)$ implies $\mathscr{U}_e(\mathscr{P}_{-i}) = \mathscr{U}_e(\mathscr{P}|^i)$. The total bandwidth usage on e in $\mathbf{b}(\mathscr{P}|^i)$ is

$$b_j(\mathscr{P}|^i) + \sum_{k\in\mathscr{U}_e(\mathscr{P}|^i)\setminus\{j\}} b_k(\mathscr{P}|^i) < b_j(\mathscr{P}_{-i}) + \sum_{k\in\mathscr{U}_e(\mathscr{P}_{-i})\setminus\{j\}} b_k(\mathscr{P}_{-i}) \leq b(e),$$

where the first inequality follows from $b_j(\mathscr{P}|^i) < b_j(\mathscr{P}_{-i})$ and the assumption $b_k(\mathscr{P}|^i) \leq b_k(\mathscr{P}_{-i})$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}$, and the second inequality follows from the feasibility of $\mathbf{b}(\mathscr{P}_{-i})$. This contradicts the fact that $e \in \mathscr{B}_j(\mathscr{P}|^i)$. Therefore, 2) holds.

We have finished the proof of this lemma. □



(a) Before player 2 chooses its path

(b) After player 2 chooses its path

Figure 2.6: Example for Lemma 2.6 and Lemma 2.7.

Figure 2.6 illustrates Part 2) of Lemma 2.6 with $i = k = 2$ and $j = 1$. From Figure 2.6(a), we observe that $b_1(\mathscr{P}_{-2}) = 5$. From Figure 2.6(b), we observe that $b_1(\mathscr{P}|^2) = 4$ and $b_2(\mathscr{P}|^2) = 4$. We note that the bandwidth of player 1 (red solid) decreases from 5 to 4 and the bandwidth player 2 (blue dotted) increases from 0 to 4. We also note that $b_2(\mathscr{P}|^2) \leq b_1(\mathscr{P}|^2)$.

Lemma 2.7. *Let P_i' be the new s_i-t_i path chosen by player i based on the observed available bandwidth. We have the following:*

1. *If $b_i(\mathscr{P}|^i) > b_i^o(\mathscr{P}|^i)$, then $\exists k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$, such that*

   ***1**a) $b_k(\mathscr{P}|^i) < b_k(\mathscr{P}_{-i})$ and **1**b) $b_k(\mathscr{P}|^i) < b_i(\mathscr{P}|^i)$,* *where $e \in \mathscr{B}_i^o(\mathscr{P}|^i)$ is an observed bottleneck of path P_i'.*

2. *If $b_j(\mathscr{P}|^i) > b_j(\mathscr{P}_{-i})$, then $\exists k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \{j\}$, such that*

   ***2a)** $b_k(\mathscr{P}|^i) < b_k(\mathscr{P}_{-i})$ and **2b)** $b_k(\mathscr{P}|^i) < b_j(\mathscr{P}|^i)$,*

   where $e \in \mathscr{B}_j(\mathscr{P}_{-i})$ is a bottleneck of path P_j when player i's path is not in the network. □

Proof. We prove Part 1) and 2) separately:

We first prove 1). Assume that $b_i(\mathscr{P}|^i) > b_i^o(\mathscr{P}|^i)$.

We prove 1) by contradiction. Assuming to the contrary that $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$ or $b_k(\mathscr{P}|^i) \geq b_i(\mathscr{P}|^i)$, $\forall k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{i\}$, we have the following two claims:

Claim 1: *For all $k \in \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$, we have $b_k(\mathscr{P}|^i) \geq b_i^o(\mathscr{P}|^i)$.*

When $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$ is true, we have

$$b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i}) \geq b^o(e) = b_i^o(\mathscr{P}|^i),$$

where the second inequality follows from Lemma 2.5 and the equality follows from the fact that $e \in \mathscr{B}_i^o(\mathscr{P}|^i)$. When $b_k(\mathscr{P}|^i) \geq b_i(\mathscr{P}|^i)$ is true, we have $b_k(\mathscr{P}|^i) \geq b_i(\mathscr{P}|^i) > b_i^o(\mathscr{P}|^i)$, due to the condition $b_i(\mathscr{P}|^i) > b_i^o(\mathscr{P}|^i)$.

Claim 2: *For all $k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$, we have $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$.*

Let k be any player in $\mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})$. We need to prove that $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$. According to the contrary assumption at the beginning of this proof, we only need to prove for the case where $b_k(\mathscr{P}|^i) \geq b_i(\mathscr{P}|^i)$ is true. In this case, we have

$$b_k(\mathscr{P}|^i) \geq b_i(\mathscr{P}|^i) > b_i^o(\mathscr{P}|^i) = b^o(e) > b_k(\mathscr{P}_{-i}),$$

where the last inequality follows from Lemma 2.5.

Note that $\mathscr{U}_e(\mathscr{P}|^i) = \mathscr{U}_e(\mathscr{P}_{-i}) \cup \{i\}$. The total bandwidth usage on link e in $\mathbf{b}(\mathscr{P}|^i)$ is

$$
\begin{aligned}
& \sum_{k \in \mathscr{U}_e(\mathscr{P}|^i)} b_k(\mathscr{P}|^i) && (2.15)\\
= \; & b_i(\mathscr{P}|^i) + \sum_{k \in \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_k(\mathscr{P}|^i) + \sum_{k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_k(\mathscr{P}|^i) \\
> \; & (|\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})| + 1) b_i^o(\mathscr{P}|^i) + \sum_{k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_k(\mathscr{P}_{-i}) && (2.16)\\
= \; & (|\tilde{\mathscr{U}}_e(\mathscr{P}_{-i})| + 1) b^o(e) + \sum_{k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \tilde{\mathscr{U}}_e(\mathscr{P}_{-i})} b_k(\mathscr{P}_{-i}) \\
= \; & b(e), && (2.17)
\end{aligned}
$$

where (2.16) follows from the condition of 1) and the two claims, and (2.17) follows from (2.4). This contradicts the feasibility of $\mathbf{b}(\mathscr{P}|^i)$. Thus we have proved 1).

We now prove 2). Assume that $b_j(\mathscr{P}|^i) > b_j(\mathscr{P}_{-i})$.

We prove 2) by contradiction. Assume to the contrary that $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$ or $b_k(\mathscr{P}|^i) \geq b_j(\mathscr{P}|^i)$, $\forall k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \{j\}$. When $b_k(\mathscr{P}|^i) \geq b_j(\mathscr{P}|^i)$ is true, we have

$$b_k(\mathscr{P}|^i) \geq b_j(\mathscr{P}|^i) > b_j(\mathscr{P}_{-i}) \geq b_k(\mathscr{P}_{-i}),$$

where we used the condition of 2) and the fact that $e \in \mathscr{B}_j(\mathscr{P}_{-i})$. Thus we have $b_k(\mathscr{P}|^i) \geq b_k(\mathscr{P}_{-i})$, $\forall k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \{j\}$. Then, considering the fact that $\mathscr{U}_e(\mathscr{P}_{-i}) \subseteq \mathscr{U}_e(\mathscr{P}|^i)$, the total bandwidth usage on link e in $\mathbf{b}(\mathscr{P}|^i)$ is

$$b_j(\mathscr{P}|^i) + \sum_{k \in \mathscr{U}_e(\mathscr{P}|^i) \setminus \{j\}} b_k(\mathscr{P}|^i) > b_j(\mathscr{P}_{-i}) + \sum_{k \in \mathscr{U}_e(\mathscr{P}_{-i}) \setminus \{j\}} b_k(\mathscr{P}_{-i}) = b(e),$$

where the equality follows from the fact that $e \in \mathscr{B}_j(\mathscr{P}_{-i})$. This violates the feasibility of $\mathbf{b}(\mathscr{P}|^i)$. We have proved 2). □

Figure 2.6 illustrates Part 2) of Lemma 2.7 with $i = 2$, $j = 3$, and $k = 1$. From Figure 2.6(a), we observe that $b_3(\mathscr{P}_{-2}) = 5$ and $b_1(\mathscr{P}_{-2}) = 5$. From Figure 2.6(b), we observe that $b_3(\mathscr{P}|^2) = 6$ and $b_1(\mathscr{P}|^2) = 4$. We note that the bandwidth of player 3 (green dashed) increases from 5 to 6, but the bandwidth of player 1 (red solid) decreases from 5 to 4. We also note that $b_1(\mathscr{P}|^2) < b_3(\mathscr{P}|^2)$.

Based on Lemma 2.6 and Lemma 2.7, we prove in the following an important theorem, which states that the bandwidth of the new path is equal to its observed bandwidth.

Theorem 2.3. *Let P_i' be the new s_i-t_i path chosen by player i based on the observed available bandwidth. Then $b_i(\mathscr{P}|^i) = b_i^o(\mathscr{P}|^i)$.* □

Proof. First, we prove that $b_i(\mathscr{P}|^i) \geq b_i^o(\mathscr{P}|^i)$. To the contrary, assume that $b_i(\mathscr{P}|^i) < b_i^o(\mathscr{P}|^i)$. We will derive a contradiction. By Part 1) of Lemma 2.6, we know that

$$\exists j, s.t., b_j(\mathscr{P}|^i) > b_j(\mathscr{P}_{-i}) \text{ and } b_j(\mathscr{P}|^i) \leq b_i(\mathscr{P}|^i). \tag{2.18}$$

By the first inequality of (2.18) and Part 2) of Lemma 2.7, we know that

$$\exists k, s.t., b_k(\mathscr{P}|^i) < b_k(\mathscr{P}_{-i}) \text{ and } b_k(\mathscr{P}|^i) < b_j(\mathscr{P}|^i). \tag{2.19}$$

By the first inequality of (2.19) and Part 2) of Lemma 2.6, we know that

$$\exists j_1, s.t., b_{j_1}(\mathscr{P}|^i) > b_{j_1}(\mathscr{P}_{-i}) \text{ and } b_{j_1}(\mathscr{P}|^i) \leq b_k(\mathscr{P}|^i). \tag{2.20}$$

By the first inequality of (2.20) and Part 2) of Lemma 2.7, we know that

$$\exists k_1, s.t., b_{k_1}(\mathscr{P}|^i) < b_{k_1}(\mathscr{P}_{-i}) \text{ and } b_{k_1}(\mathscr{P}|^i) < b_{j_1}(\mathscr{P}|^i). \tag{2.21}$$

Repeating (2.20) and (2.21), we obtain a sequence $i, k, k_1, k_2, \ldots$, such that $b_i(\mathscr{P}|^i) > b_k(\mathscr{P}|^i) > b_{k_1}(\mathscr{P}|^i) > b_{k_2}(\mathscr{P}|^i) > \cdots$. Since the number of users is finite, there must be a user that is repeated an infinite number of times in the above sequence of users. This is a contradiction, since the corresponding sequence of bandwidth values is strictly decreasing. This contradiction proves that $b_i(\mathscr{P}|^i) \geq b_i^o(\mathscr{P}|^i)$.

Using a similar logic, we can prove that $b_i(\mathscr{P}|^i) \leq b_i^o(\mathscr{P}|^i)$. This implies that $b_i(\mathscr{P}|^i) = b_i^o(\mathscr{P}|^i)$. □

Remark 5. As a direct consequence of Theorem 2.3, player i has an incentive to change its strategy if and only if $b_i^o(\mathscr{P}|^i) > b_i(\mathscr{P})$. Also, P_i' is the best response strategy for player i.

2.4.3 Converging to Nash Equilibrium

In this section, we present a game based algorithm, listed in Algorithm 2, to compute an NE of the MAXBAR game. The idea of the algorithm is as follows. In the initialization stage (Line 2), each player i chooses an initial s_i–t_i path regardless of the paths of other players. Without loss of generality, each player chooses a path with maximum bandwidth using an algorithm denoted by $WP(G, s_i, t_i, b)$. Then Algorithm 2 proceeds in a round-robin fashion. At every stage, there can be only one player changing its path. Such assumption is common in game theory and essential to avoid oscillation.

When a player plans to change its path, it follows the following steps:

1. Compute its current bandwidth (Line 5).
2. Calculates the observed available bandwidth for each link in the resulting network (Lines 6 and 7).
3. Finds a path with the maximum observed bandwidth (Line 8).

4. If the observed bandwidth of the new path is greater than its current bandwidth, it switches to the new path; otherwise, it keeps the same path (Line 9).

The process stops when no player can improve its bandwidth by changing to another path.

Algorithm 2: Game Based Algorithm

```
   input  : Network $G(V,E,b)$ and set $\mathscr{U}$ of players $\{1,\dots,N\}$
   output : A Nash Equilibrium $\mathscr{P}$
1  $\mathscr{P} \leftarrow \emptyset$;
2  $P_i \leftarrow WP(G,s_i,t_i,b)$, $\mathscr{P} \leftarrow \mathscr{P} \cup \{P_i\}, \forall i \in \mathscr{U}$;
3  repeat
4      foreach player $i \in \mathscr{U}$ do
5          $(b_1(\mathscr{P}),\dots,b_N(\mathscr{P})) \leftarrow ComB(G,b,\mathscr{P},\mathscr{U})$;
6          $(b_1(\mathscr{P}_{-i}),\dots,b_N(\mathscr{P}_{-i})) \leftarrow ComB(G,b,\mathscr{P}_{-i},\mathscr{U})$;
7          Compute $b^o(e)$ for all $e \in E$ using (2.4);
8          $P_i' \leftarrow WP(G,s_i,t_i,b^o)$;
9          if $b^o(P_i') > b_i(\mathscr{P})$ then $\mathscr{P} \leftarrow \mathscr{P}|^i P_i'$;
10     end
11 until there is no path changed;
12 return $\mathscr{P}$;
```

In Algorithm 2, $WP(G,s_i,t_i,b)$ returns a path with maximum bandwidth from s_i to t_i in graph G with bandwidth function b. The basic idea of Algorithm 2 is as follows. First (Line 2), each player i chooses an initial s_i–t_i path regardless of other players. Next, in a round-robin fashion (Lines 3-11), each player changes its path to improve its utility, when possible. This is referred to as the *best-response move* in [106]. The process stops when no player can improve its bandwidth by changing to another path.

The correctness and an upper on the convergence speed of Algorithm 2 are captured in the following theorem.

Theorem 2.4. *For every instance of the MAXBAR game, Algorithm 2 converges to a set $\mathscr{P}$ of paths in $O((Nm+n\log n+N\log N)(Nm)^N)$ time, where N is the number of players, m is* *the number of links, and n is the number of nodes. Moreover, $\mathscr{P}$ is an NE of the MAXBAR game.* □

To prove this theorem, we need the following lemma, which shows an important property of the global bottleneck.

Lemma 2.8. *Let $\mathscr{P}$ be a path set of the users and $\boldsymbol{b}(\mathscr{P})$ be the corresponding* MFBA. *Let $\bar{e}$ be a global bottleneck. We then have $b_j(\mathscr{P}) = \frac{b(\bar{e})}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}$, $\forall j \in \mathscr{U}_{\bar{e}}(\mathscr{P})$.* □

Proof. First, we claim that *for any $e \in \mathscr{B}_i(\mathscr{P})$ for some i, we have $b_i(\mathscr{P}) \geq \frac{b(e)}{|\mathscr{U}_e(\mathscr{P})|}$*. Considering both Properties 1) and 2) of bottleneck e, we have

$$b(e) = \sum_{j \in \mathscr{U}_e(\mathscr{P})} b_j(\mathscr{P}) \leq |\mathscr{U}_e(\mathscr{P})| \cdot b_i(\mathscr{P}).$$

Thus the claim is proved. Based on this claim and the fact that $\bar{e}$ is a global bottleneck, we have

$$b_j(\mathscr{P}) \geq \frac{b(\bar{e})}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}, \forall j \in \mathscr{U}_{\bar{e}}(\mathscr{P}). \tag{2.22}$$

Assume that $\exists k \in \mathscr{U}_{\bar{e}}(\mathscr{P})$ such that $b_k(\mathscr{P}) > \frac{b(\bar{e})}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}$. The total bandwidth usage on $\bar{e}$ is $\sum_{j \in \mathscr{U}_{\bar{e}}(\mathscr{P})} b_j(\mathscr{P}) > b(\bar{e})$, contradicting the feasibility of $\mathbf{b}(\mathscr{P})$. Hence we have proved that $b_j(\mathscr{P}) = \frac{b(\bar{e})}{|\mathscr{U}_{\bar{e}}(\mathscr{P})|}$, $\forall j \in \mathscr{U}_{\bar{e}}(\mathscr{P})$. □

Proof of Theorem 2.4: By Lemma 2.1, we conclude that every time a player changes its path, the ordering $\vec{\mathbf{b}}_l$ increases lexicographically. Now we prove an upper bound on the number of times the ordering can increase. By Lemma 2.8, we know that a global bottleneck must be equally shared by all paths using it. As a result, the number of different possible values of b_1 is bounded by $O(Nm)$. For each possible value of b_1, there are at most N players whose paths correspond to this value. If the value of b_1 and the corresponding path P_i stay the same, the number of different possible values of b_2 is $O(Nm)$. The reason is that we can subtract b_1 from the bandwidth of each link along P_i, and remove i from the player set. This resulting graph is a smaller instance and all the lemmas still hold. Repeating this analysis for all the coordinates, we conclude that the number of times that the lexicographic ordering can increase is bounded by $O((Nm)^N)$. The time complexity of Algorithm 1 is $O(Nm)$. Recall that computing $b^o(e)$ for all $e \in E$ takes $O(Nm + N\log N)$ time. In addition, the time complexity of $WP(G, b, \mathscr{P})$ is $O(m + n\log n)$ by using a variant of Dijkstra's shortest path algorithm [36, 44]. Therefore the time complexity of Algorithm 2 is $O((Nm + n\log n + N\log N)(Nm)^N)$. By Theorem 2.3, the returned $\mathscr{P}$ is an NE, since no player can improve its utility by changing its path unilaterally.

Remark 6. Our extensive simulations in Section 2.5 show that the MAXBAR game converges to an NE within 10 iterations. This indicates that our theoretical bound $O((Nm)^N)$ on the number of iterations is quite conservative.

Remark 7. As shown in the example in Section 2.4, there could be more than one NE. If the initial set of strategies were different from the one computed in Line 2 of Algorithm 2, Lines 3–12 may lead to a *different* NE. However, Lines 3–12 of the algorithm will always lead to some NE.

Remark 8. In our algorithm, we require that only one player can change its path each time. This is essential to the convergence of the algorithm. We use an example to show that oscillation may occur when this requirement is violated. As shown in Figure 2.7, assume that player 1's path is s_1-v_1-v_2-v_3-t_1 and player 2's path is s_2-v_1-v_2-v_3-t_2 at certain point of the game. If the players are allowed to change their paths simultaneously, player 1 and player 2 would change their paths to s_1-v_1-v_4-v_3-t_1 and s_2-v_1-v_4-v_3-t_2, respectively. Because both of them expect that they can increase their bandwidth from 1 to 2. Since two players change their paths simultaneously, the allocated bandwidth for each player is actually 1.5. Now both players would change their paths back to the previous ones because they expect to increase their bandwidth from 1.5 to 2. Therefore the network will oscillate between Figure 2.7(a) and Figure 2.7(b) if simultaneous path change is allowed.

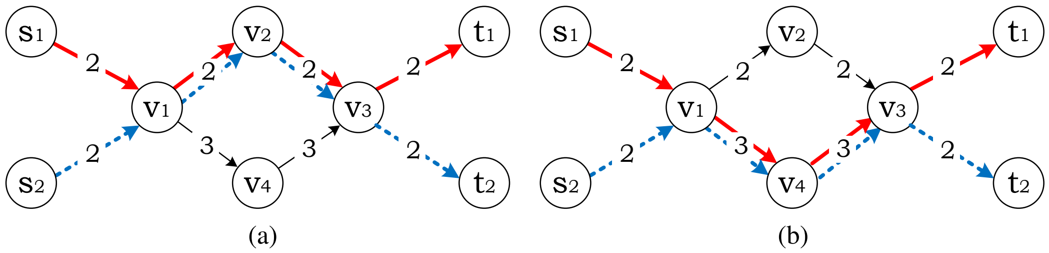


Figure 2.7: Oscillation when simultaneous path change is allowed

One way to enforce the users in the network to follow the game course is to use a token-based protocol, where a token is circulated among the users in a round-robin fashion– only the user with the token has the opportunity to change its path. This token-based protocol can guarantee the convergence of Algorithm 2. A distributed implementation of $ComB(G,b,\mathscr{P},\mathscr{U})$ were proposed by [19, 69]. The information needed by (2.4) to compute the observed available bandwidth is sent to each user by the link-state algorithm for determining the new path.

2.5 Generalization of MAXBAR

We have studied the MAXBAR problem where users have infinite bandwidth demand. In this section, we generalize the MAXBAR problem and consider the case where each user has a bandwidth demand of $\gamma_i > 0$. We denote this generalized problem as MAXBAR$_\gamma$. The difference between the MAXBAR$_\gamma$ problem and the MAXBAR problem is that we need to consider user's bandwidth demand while allocating bandwidth. Each user i will only use up to γ_i bandwidth and is not interested in switching to a path with more bandwidth as long as its bandwidth demand is met. The MAXBAR problem is a special case of the MAXBAR$_\gamma$ problem, as we can consider that $\gamma_i = \infty$ in the MAXBAR problem. It is seemingly necessary for us to redesign the ComB algorithm, and analyze the existence of NEs and convergence of routing again. However, we will show that we can transform any instance of the MAXBAR$_\gamma$ problem to a corresponding instance of the MAXBAR problem, and study the MAXBAR problem using the algorithms and analysis in previous sections.

Let $\mathscr{I}_\gamma = ((V,E,b),\mathscr{U},\gamma)$ be an instance of the MAXBAR$_\gamma$ problem, where $G = (V,E,b)$ is the edge-weighted graph for the network. We build a corresponding instance $\mathscr{I} = ((V',E',b'),\mathscr{U}')$ of the MAXBAR problem (where $G' = (V',E',b')$ is the edge-weighted graph for the corresponding network) as follows. Corresponding to each node $v \in V$, V' contains a node v. Corresponding to each link $(v,w) \in E$, E' contains a link (v,w) and $b'(v,w) = b(v,w)$. Corresponding to each source $s_i \in V$, V' contains an *additional node* s'_i and E' contains an *additional link* (s'_i,s_i) with bandwidth $b'(s'_i,s_i) = \gamma_i$. Corresponding to each user $i \in \mathscr{U}$, $\mathscr{U}'$ contains a user i, who needs to transmit packets from s'_i to t_i in G'. Figure 2.8 illustrates this transformation.

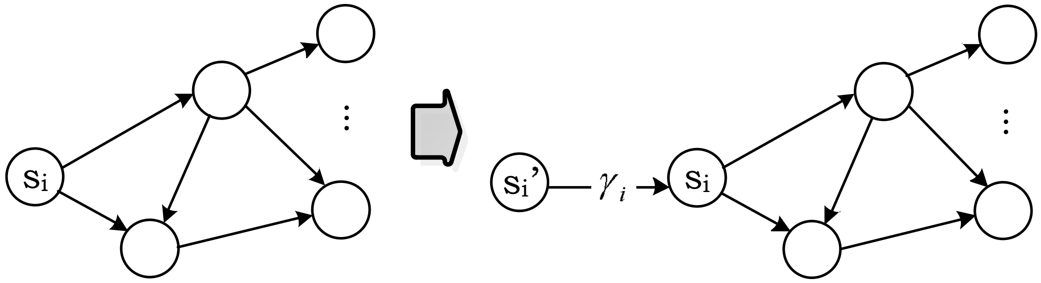
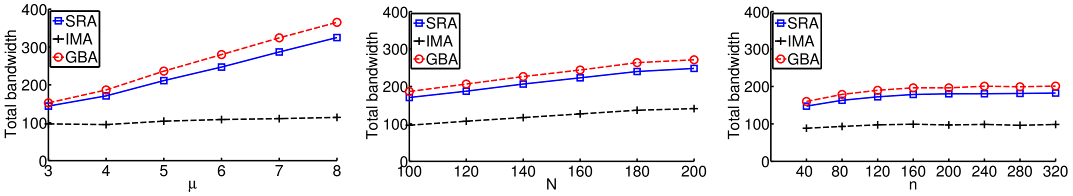


Figure 2.8: Transforming an instance of the MAXBAR$_\gamma$ problem to a corresponding instance of the MAXBAR problem

Note that although we allow users to have as much bandwidth as possible in the MAXBAR problem, the special link (s'_i,s_i) ensures that user i will only compete for bandwidth up to the demand γ_i. It is clear that the *MFBA* for $\mathscr{I}_\gamma$ can be obtained by computing the *MFBA* for $\mathscr{I}$. Therefore all the lemmas and theorems for the MAXBAR problem still hold for the MAXBAR$_\gamma$ problem.

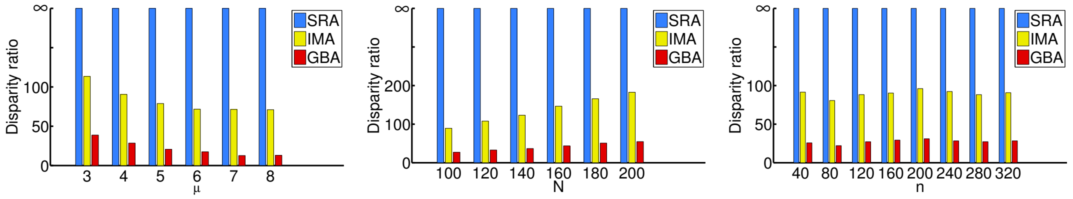
2.6 Performance Evaluation

In this section, we evaluate the performance and verify the convergence analysis of Algorithm 2 (denoted as GBA) on network topologies generated by BRITE [15].



(a) Impact of μ ($n = 120, N = 100$)
(b) Impact of N ($n = 120, \mu = 4$)
(c) Impact of n ($\mu = 4, N = 100$)

Figure 2.9: Total bandwidth

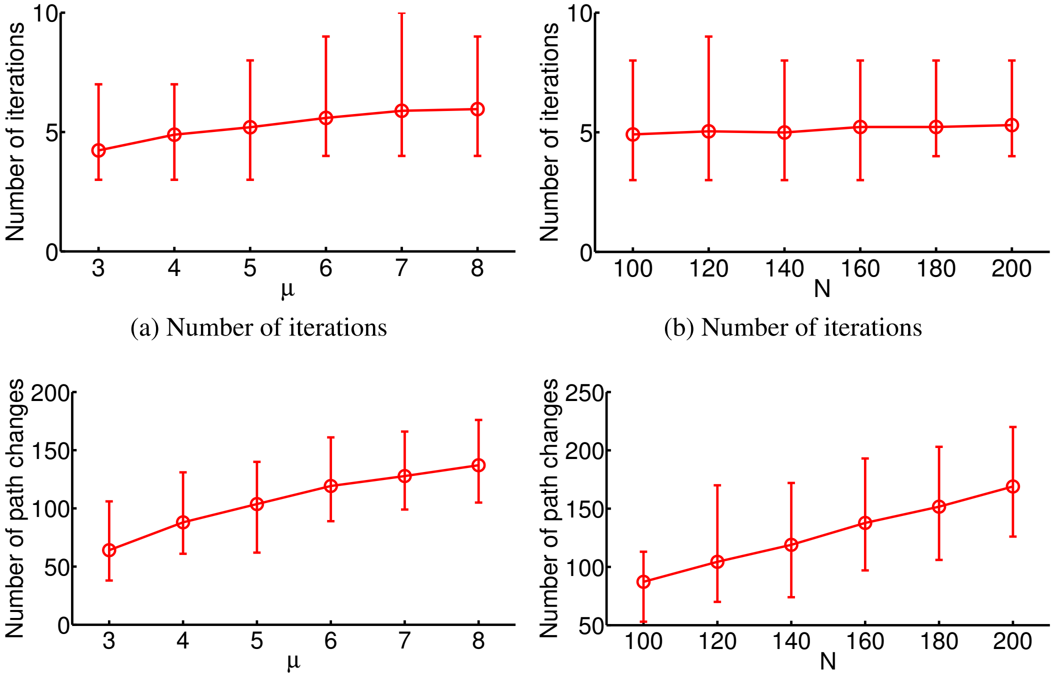


(a) Impact of μ ($n = 120, N = 100$)
(b) Impact of N ($n = 120, \mu = 4$)
(c) Impact of n ($\mu = 4, N = 100$)

Figure 2.10: Disparity ratio

Simulation Setup

We compared GBA with two other routing algorithms. In the first algorithm, each user acts independently and attempts to maximize its bandwidth as much as possible. We denote this algorithm by IMA (Independent Maximization Algorithm). In the second algorithm, the bandwidth allocation for the users is done sequentially. A user is chosen randomly from the set of users that have not been allocated bandwidth. It then chooses a widest path in the residual network, and has a bandwidth equal to that of the chosen



(a) Number of iterations

(b) Number of iterations

(c) Number of path changes

(d) Number of path changes

Figure 2.11: Convergence speed. For (a) and (c), $n = 120$ and $N = 100$. For (b) and (d), $n = 120$ and $\mu = 4$.

path. This procedure is repeated until all users are considered for bandwidth allocation. This technique is similar to the Resource reSerVation Protocol (RSVP) [119], with the difference being that each user is allocated the maximum possible bandwidth in the residual network. We denote this scheme by SRA (Sequential Reservation Algorithm).

BRITE [15] is a widely used Internet topology generator. We used the Waxman model [145] with default values for $\alpha = 0.15$ and $\beta = 0.2$. According to the Waxman model, if d_{vw} denotes the Euclidean distance between two nodes v and w, the probability of having a directed link (v, w) from v to w is given by $\beta \times \exp\left(\frac{d_{vw}}{\alpha \cdot L}\right)$, where L is the maximum distance between two nodes. The nodes of the graph were deployed randomly in a square region of size 1000×1000 m^2. We varied the number of nodes n from 40 to 320 with increment of 40 and set the number of links to $m = \mu n$, where μ is the link density and was varied from 3 to 8. We varied the number of users N from 100 to 200 with increment of 20. For each network size, we used BRITE to generate different network topologies, where the link bandwidth was drawn from a uniform distribution in the range $[1,10]$. For each setting, we randomly generated 100 test cases and averaged the results.

Performance Metrics:

- Total bandwidth: the sum of the bandwidth of all users.
- Bandwidth disparity ratio: the ratio of the highest bandwidth over the lowest bandwidth among the users.
- Convergence speed: the number of the round-robin iterations (Lines 3–11 in Algorithm 2) or the number of path changes (Line 9 in Algorithm 2).

Results Analysis

Total Bandwidth: Figure 2.9 shows the total bandwidth obtained by SRA, IMA and GBA. We observe that GBA always outperforms IMA. This is as expected, because IMA uses less information in decision making. SRA and GBA have similar performance, because some users can reserve most of the bandwidth resources in SRA. We also notice that the total bandwidth in Figure 2.9(c) increases first and almost remains the same after $n = 240$. This is because the bandwidth of some users has reached the maximum value at $n = 240$.

Disparity Ratio: Figure 2.10 shows the bandwidth disparity ratio obtained by SRA, IMA and GBA. We observe that GBA is the fairest. SRA has the worst disparity ratio with the value of ∞ for all settings. This is because some users will be blocked and have zero bandwidth in SRA, as other users have reserved all the bandwidth on the links connecting their sources and destinations. We also see that the disparity ratios of IMA and GBA are independent of n, as shown in Figure 2.10(c), but decrease when the user density, $\frac{N}{m}$, becomes lower, as shown in Figure 2.10(a) and Figure 2.10(b). The reason is that when the user density is low, users have a low probability of sharing common links and hence competing the bandwidth. These results are not unexpected, as SRA and IMA are not designed to achieve small disparity ratios.

Convergence Speed: Figure 2.11(a) and Figure 2.11(b) show the number of iterations before GBA converges. We observe that the number of iterations is within 10 in all cases. Figure 2.11(c) and Figure 2.11(d) show the number of path changes before GBA converges. The theoretical bound on the number of path changes is $O((Nm)^N)$ in Theorem 2.4. However, as we can see, the number of path changes in the simulations is significantly less than the theoretical bound. Another observation is that GBA converges slower when the link density μ is high, as shown in Figure 2.11(c). The reason is that when each node has more links, a user is highly likely to find a path with higher bandwidth if the current path results in low bandwidth, due to the competition from newly joined paths. According to Theorem 2.4, the number of path changes is independent of n. Our simulation results also confirm this proof and thus are omitted due to the space limitations.

To summarize, extensive simulations show that our algorithm converges to an NE rapidly and achieves very good fairness as well as total bandwidth.

2.7 Conclusions

In this work, we formulated the problem of routing in networks with max-min fair bandwidth allocation as a non-cooperative game, where each user aims to maximize its own bandwidth. We proved the existence of Nash Equilibria, where no user has any incentive to unilaterally change its path. We derived both a lower bound and an upper bound of the system degradation, due to the selfish behavior of users. Finding a path with maximum bandwidth in the max-min fair network is both a key step for our main analysis and of independent interest. To this end, we introduced a novel concept of *observed available bandwidth* to accurately predict the available bandwidth on each link. We next presented a game based algorithm to compute an NE and proved that the network converges to an NE if all users follow the natural game course. Note that the theoretical convergence speed proved in this work does not change even when an approximate Nash Equilibrium [31] is considered. Deriving a tighter bound on the time complexity of the convergence speed is a future research direction. Through extensive simulations, we showed that the network can converge to an NE within 10 iterations and also achieve better fairness compared with other algorithms.

Chapter 3

HERA: An Optimal Relay Assignment Scheme for Cooperative Networks

Through cooperative relaying from wireless devices (generally called relay nodes), cooperative communication (CC) [83] has been shown to have the potential to increase the channel capacity between two wireless devices. The essence of CC is to exploit the nature of broadcast and the relaying capability of other nodes to achieve spatial diversity. Two primary CC modes have been commonly used, *Amplify-and-Forward* (AF) and *Decode-and-Forward* (DF) [83], depending on how the relay node processes the received signal and transmits to the destination. Because an improper choice of the relay node for a source-destination pair can result in an even smaller capacity than that under direct transmission, the assignment of relay nodes plays a critical role in the performance of CC [12, 17, 37, 127, 168].

3.1 Introduction

In this work, we consider the following scenario. In a wireless network, there are a number of source nodes and corresponding destination nodes. Other wireless devices can function as relay nodes. We are interested in designing a relay assignment scheme, such that the total capacity under the assignment is maximized.

We call the network with CC the *cooperative network*. Designing a relay assignment scheme for cooperative networks is very challenging for the following reasons.

- *System Performance*: A relay assignment scheme should provide a relay assignment algorithm, which appropriately assigns relay nodes to source nodes such that the system capacity is maximized. The system capacity is the sum of the capacity of all source nodes.

- *Selfish Behavior*: Usually, wireless devices in cooperative networks are not owned by a single entity, but by many profit-maximizing independent entities. Therefore, even if an optimal relay assignment algorithm is developed, an individual source node may not want to follow the assignment, given the fact that it can improve its own capacity by selecting a different relay node. This selfish behavior can result in system performance degradation.
- *Potential Cheating*: As to relay nodes, most of the protocols in cooperative networks assume that all the wireless devices are cooperative, and in particular willing to participate in cooperative communications as relay nodes. However, the voluntary cooperativeness assumption may not be true in reality as relaying data for other network nodes can consume energy and other resources of the relay node. A naive solution is to make payments to the participating relay nodes as an incentive. The question arising from this naive solution is how much a relay node should be paid for helping with the cooperative communication. A simple payment mechanism is vulnerable to the dishonest behavior of relay nodes, in the sense that a relay node can profit from lying about its true relaying capability, e.g. transmission power.

In this work, we design an integrated optimal relay assignment scheme for cooperative networks, called HERA, named after the Goddess of Marriage in Greek Mythology. **T**o the best of our knowledge, HERA is the first relay assignment scheme for cooperative networks, which considers both selfish and cheating behavior of network entities while guaranteeing socially optimal system performance. HERA is composed of three components: 1) an optimal relay assignment algorithm, 2) a payment mechanism for source nodes, and 3) a payment mechanism for relay nodes. HERA is a centralized scheme, where a system administrator is responsible for collecting the payment from the source nodes and paying the relay nodes.

HERA provides the following key features:

- HERA guarantees to find a relay assignment for the source nodes, such that the total capacity is maximized. The system model considered in this paper allows a relay node to be shared by multiple source nodes. Hence it is more general compared with the model in [127], where each relay node is restricted to be assigned to only one source node. Our assignment algorithm works regardless of which CC mode is used in the network. It is also independent of the relation between the number of source nodes and that of the relay nodes. In addition, our algorithm can guarantee that the achieved capacity of each source node under the assignment is no less than that achieved by direct transmission.

- HERA provides a payment mechanism to charge source nodes for using relaying service from the relay nodes. To cope with the selfish behavior of source nodes, our payment mechanism is designed in a way such that the system possesses a *Strictly Dominant Strategy Equilibrium* (SDSE), where each selfish source node plays the strategy that brings the maximum utility regardless of others' strategies. Furthermore, the SDSE achieves the socially optimal system capacity.

- HERA also provides a payment mechanism to pay relay nodes for providing relaying service. To prevent relay nodes from lying about their relaying ability (e.g. transmission power) to gain profits, the payment mechanism uses a VCG-based payment formula to calculate the payment. Under this payment mechanism, reporting true relaying ability is the dominant strategy for each relay node. In other words, the relay node can maximize its payment received from the system administrator by reporting its true relaying ability.

- Finally, from the perspective of the system administrator, HERA assures that the system administrator will not run the system with any loss. In other words, the total payment collected from source nodes is at least as much as the total payment paid to relay nodes.

The remainder of this paper is organized as follows. In Section 3.2, we give a brief review of the related work in the literature. In Section 3.3, we describe the system model considered in this paper. In Section 3.4, we present a polynomial time optimal algorithm to solve the relay assignment problem, study the selfish behavior of source nodes and design a payment mechanism to charge source nodes for using relaying service, and consider the potential cheating relay nodes in the system, design another payment mechanism to pay relay nodes for providing relaying service and prove the desired properties of the designed mechanism. We present our extensive experimental results in Section 3.5. We conclude this paper in Section 3.6.

3.2 Background

In [12], Bletsas *et al.* proposed a novel scheme to select the best relay node for a single source node from a set of available relays. However, this cannot be extended to a network consisting multiple source nodes, which is the model studied in this work.

Some efforts have been made on the relay assignment or relay selection problem in cooperative networks. In [17], Cai *et al.* studied the problem of relay selection and power allocation for AF wireless relay networks. They first considered a simple network with only one source node, and then extended it to the multiple-source case. The proposed algorithm is an effective heuristic, but offers no performance guarantee. Xu *et al.* [152] studied a similar problem with a different objective, which is to minimize the total power consumption of the network. In [109], Ng and Yu jointly considered the relay node selection, cooperative communication and resource allocation for utility maximization in a cellular network. However, the algorithm is heuristic and not polynomial, as pointed out by Sharma *et al.* [127].

In [127], Sharma *et al.* studied the relay assignment problem in a network environment, such that the minimum capacity among all source nodes is maximized. Following this work, Zhang *et al.* [167] considered the relay assignment problem with interference mitigation. In both models in [127] and [167], a relay node is restricted to be assigned to at most one source node. In contrast, our model is more general in the sense that it allows multiple source nodes to share the same relay node. In addition, different from [127], our objective is to maximize the total capacity of all pairs. Although Zhang *et al.* [167] had the same objective as ours, they only provided a heuristic algorithm.

There are few studies on the scheme design for cooperative communications in the networking literature, among which the works in [66, 128, 141, 157] are most related to our work. In [128], Shastry and Adve proposed a pricing-based system to stimulate the cooperation via payment to the relay nodes. The goal in their scheme is to ensure both the access point and the relay nodes benefit from cooperation. In [141], Wang *et al.* employed a buyer/seller Stackelberg game, where a single buyer tries to buy services from multiple relays. The buyer announces its selection of relays and the required transmission power, then the relays ask proper prices to maximize their profits. In [66], Huang *et al.* proposed two auction mechanisms, which are essentially repeated games. In each auction mechanism, each user iteratively updates its bid to maximize its own utility function with the knowledge of others' previous bids. With a common drawback, none of the above works guarantees the optimal system capacity or considers truthfulness of relay nodes. In [157], Yang *et al.* designed a truthful auction scheme for cooperative communications, which satisfies truthfulness, individual rationality, and budget balance properties. Similarly, the scheme cannot guarantee the optimal system capacity.

3.3 System Model

We consider a static wireless network. There is a set $\mathscr{S} = \{s_1, s_2, \cdots, s_n\}$ of n source nodes and a set $\mathscr{D} = \{d_1, d_2, \cdots, d_n\}$ of corresponding destination nodes, where s_i transmits to d_i. Other nodes in the network function as relay nodes. We assume that there is a collection $\mathscr{R} = \{r_1, r_2, \ldots, r_m\}$ of m relay nodes. As in [127], we assume that orthogonal channels are available in the network (e.g. using OFDMA) to mitigate interference. We further assume that each node is equipped with a single transceiver and can either transmit or receive at a time. Let P_i^s denote the transmission power of source node s_i and P_j^r denote the transmission power of relay node r_j. Let $P^s = (P_1^s, P_2^s, \ldots, P_n^s)$ and $P^r = (P_1^r, P_2^r, \ldots, P_m^r)$. When node u transmits a signal to node v with power P_u, the signal-to-noise ratio (SNR) at node v, denoted as SNR_{uv}, is $SNR_{uv} = \frac{P_u}{N_0 \cdot ||u,v||^\alpha}$, where N_0 is the abient noise, $||u,v||$ is the Euclidean distance between u and v, and α is the path loss exponent which is between 2 and 4 in general.

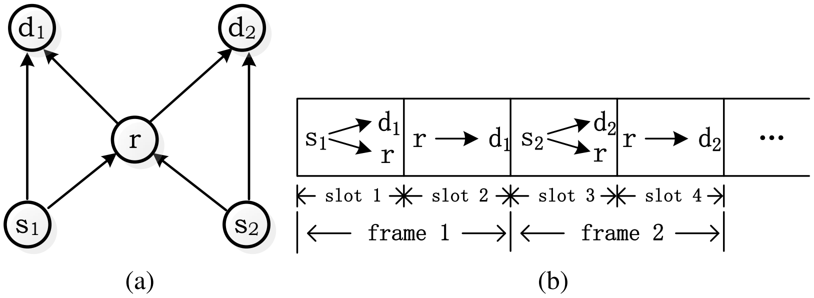
For the transmission model, we assume that each source node has an option to use cooperative communication (CC) with the help of a relay node. A recent work by Zhao *et al.* [168] showed that it is sufficient for a source node to choose the best relay node even when multiple relay nodes are available to achieve full diversity. Therefore, it is reasonable to assume that each source node will either transmit directly or use CC with the help of only one relay node. When source node s transmits to destination node d directly, the achievable capacity is $c_{DT}(s,d) = W \log_2(1 + SNR_{sd})$, where W is the bandwidth of the channel. There are two different CC modes, *Amplify-and-Forward* (AF) and *Decode-and-Forward* (DF) [83]. Let r denote the relay node and P_r be the transmission power of r. The achievable capacity from s to d under the AF mode is

$$c_{AF}(s,r,d) = \frac{W}{2} \log_2 \left(1 + SNR_{sd} + \frac{SNR_{sr} \cdot SNR_{rd}}{SNR_{sr} + SNR_{rd} + 1} \right).$$

The achievable capacity from s to d under the DF mode is

$$c_{DF}(s,r,d) = \frac{W}{2}\min\{\log_2(1+SNR_{sr}), \log_2(1+SNR_{sd}+SNR_{rd})\}.$$

Note that, for given s and d, both c_{AF} and c_{DF} are functions of P_r, $||s,r||$ and $||r,d||$. Thus whether a source node can obtain larger capacity by using CC than it can by transmitting directly depends on the relay node assigned. The scheme designed in this work is *independent* of the CC mode. We use c_R to denote the achievable capacity under CC. Let $\bar{\mathscr{S}} = \mathscr{S} \cup \{s_0\}$ and $\bar{\mathscr{R}} = \mathscr{R} \cup \{r_0\}$, where s_0 is a *virtual source node* and r_0 is a *virtual relay node*. Let $\mathscr{A} = \{(s_1, r_{j_1}), (s_2, r_{j_2}), \ldots, (s_n, r_{j_n})\} \subseteq \mathscr{S} \times \bar{\mathscr{R}}$ denote a relay assignment. If $(s_i, r_j) \in \mathscr{A}$, relay node r_j is *assigned* to source node s_i under assignment $\mathscr{A}$. If $(s_i, r_0) \in \mathscr{A}$, s_i transmits to d_i directly under the relay assignment $\mathscr{A}$. Note that it is possible to have $(s_i, r_j), (s_k, r_j) \in \mathscr{A}$, for $s_i \neq s_k$. This is a major difference between our model and the model in [127], where a relay node is assigned to at most one source node. Since we do not enforce such constraints, our model is more general.



(a) (b)

Figure 3.1: Multi-source cooperative communication

Now let us consider the case where the same relay node is assigned to multiple source nodes. In this case, we use $\mathscr{S}_j$ to denote the set of source nodes being assigned r_j, i.e., $\mathscr{S}_j = \{s_i | (s_i, r_j) \in \mathscr{A}\}$. Note that $\mathscr{S}_j$ is dependent on relay assignment $\mathscr{A}$. We assume that r_j equally provides service to all the source nodes employing it. This can be achieved for example by using a reservation-based TDMA scheduling. The relay node serves each source node in a round-robin fashion. Each frame is dedicated to a single source node for CC. Each source node gets served every n_j frames, where $n_j = |\mathscr{S}_j|$. Therefore, the average achievable capacity for each source node $s_i \in \mathscr{S}_j$ is $\frac{c_R(s_i,r_j,P^r_j,d_i)}{n_j}$. Let $c(s_i, r_j, \mathscr{A}, P^r)$ denote the achievable capacity of s_i under relay assignment $\mathscr{A}$, where $(s_i, r_j) \in \mathscr{A}$. *Hereafter we also omit d_i in the capacity expression.* Thus we have

$$c(s_i, r_j, \mathscr{A}, P^r) = \begin{cases} \frac{c_R(s_i,r_j,P^r_j)}{n_j}, & \text{if } r_j \neq r_0, \\ c_{DT}(s_i), & \text{if } r_j = r_0. \end{cases}$$

In the expressions above, we take P^r (or P^r_j) as a parameter, because a relay node may lie about its transmission power. We will explain it in detail later. We define the *system capacity*, denoted by $C(\mathscr{S}, \mathscr{R}, \mathscr{A}, P^r)$, corresponding to relay assignment $\mathscr{A}$ and transmission power P^r, as the total capacity of all the source nodes in $\mathscr{S}$, i.e., $C(\mathscr{S}, \mathscr{R}, \mathscr{A}, P^r) = \sum_{s_i \in \mathscr{S}, (s_i, r_j) \in \mathscr{A}} c(s_i, r_j, \mathscr{A}, P^r)$.

The ultimate goal in the design of the relay assignment scheme can be defined as the following optimization problem.

Definition 3.1. *(Relay Assignment Problem (RAP)): Given $\mathscr{S}$, $\mathscr{D}$, $\mathscr{R}$, and P^r, the* Relay Assignment Problem *seeks for a relay assignment $\mathscr{A}$ such that $C(\mathscr{S}, \mathscr{R}, \mathscr{A}, P^r)$ is maximized among all possible relay assignments.* □

RAP is different from the problem studied in [127], whose objective is to maximize the minimum capacity among all source nodes. Let $\mathscr{A}^*(\mathscr{S}, \mathscr{D}, \mathscr{R}, P^r)$ be the optimal solution to RAP. For notational simplicity, we use $\mathscr{A}^*$ to denote $\mathscr{A}^*(\mathscr{S}, \mathscr{D}, \mathscr{R}, P^r)$ and C to denote $C(\mathscr{S}, \mathscr{R}, \mathscr{A}, P^r)$ when the context is clear. Correspondingly, C^* denotes $C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*, P^r)$.

3.4 Design of HERA

In this section, we design an integrated optimal relay assignment scheme for cooperative networks, called HERA, named after the Goddess of Marriage in Greek Mythology. **To the best of our knowledge, HERA is the first relay assignment scheme for cooperative networks, which considers both selfish and cheating behavior of network entities while guaranteeing socially optimal system performance.** HERA is composed of three components: 1) an optimal relay assignment algorithm, 2) a payment mechanism for source nodes, and 3) a payment mechanism for relay nodes. HERA is a centralized scheme, where a system administrator is responsible for collecting the payment from the source nodes and paying the relay nodes. HERA provides the following key features:

- HERA guarantees to find a relay assignment for the source nodes, such that the total capacity is maximized. The system model considered in this work allows a relay node to be shared by multiple source nodes. Hence it is more general compared with the model in [127], where each relay node is restricted to be assigned to only one source node. Our assignment algorithm works regardless of which CC mode is used in the network. It is also independent of the relation between the number of source nodes and that of the relay nodes. In addition, our algorithm can guarantee that the achieved capacity of each source node under the assignment is no less than that achieved by direct transmission.

- HERA provides a payment mechanism to charge source nodes for using relaying service from the relay nodes. To cope with the selfish behavior of source nodes, our payment mechanism is designed in a way such that the system possesses a *Strictly Dominant Strategy Equilibrium* (SDSE), where each selfish source node plays the strategy that brings the maximum utility regardless of others' strategies. Furthermore, the SDSE achieves the socially optimal system capacity.

- HERA also provides a payment mechanism to pay relay nodes for providing relaying service. To prevent relay nodes from lying about their relaying ability (e.g. transmission power) to gain profits, the payment mechanism uses a VCG-based payment formula to calculate the payment. Under this payment mechanism, reporting true relaying ability is the dominant strategy for each relay node. In other words, the relay node can maximize its payment received from the system administrator by reporting its true relaying ability.

- Finally, from the perspective of the system administrator, HERA assures that the system administrator will not run the system with any loss. In other words, the total payment collected from source nodes is at least as much as the total payment paid to relay nodes.

Due to the possibility of sharing a common relay node among multiple source nodes, solving RAP becomes a challenging task. Nonetheless, we can design a polynomial time optimal algorithm to solve RAP by exploiting some special properties of the problem.

3.4.1 Optimal Relay Assignment

We start with an example consisting of 5 source-destination pairs and 2 relay nodes. Each of the five subtables in Table 3.1 represents a relay assignment. More specifically, the number in the cell of column s_i and row r_j is the achievable capacity for source node s_i, when relay node r_j is exclusively assigned to it. The symbol ϕ represents direct transmission to the corresponding destination. For example, in Table 3.1(a), the numbers in the first column represent $c_R(s_1, r_1, P_1^r) = 10$, $c_R(s_1, r_2, P_2^r) = 4$ and $c_{DT}(s_1) = 4$. Each highlighted cell represents the current assigned relay node for the corresponding source node. For example, in Table 3.1(b), $\mathscr{A}_2 = \{(s_1, r_1), (s_2, r_2), (s_4, r_2), (s_5, r_2)\}$. From Table 3.1(a) to 3.1(d), we illustrate an iterative procedure to improve the total capacity. Let $C_i = C(\mathscr{S}, \mathscr{R}, \mathscr{A}_i, P^r)$. In each iteration, we change the relay assignment of the underlined source node from its currently assigned relay node to transmitting directly. For example, in Table 3.1(b), we change the relay assignment of s_3 from r_1 to ϕ and improve the total capacity from 17 to 20. Note that during this procedure, it seems that we can improve the total capacity by changing the assignment of the source node with minimum c_R among all the source nodes sharing the same relay node and letting it transmit directly to its destination. Later, we will prove that this is not a coincidence but an inherent property.

	s_1	s_2	s_3	s_4	s_5
r_1	**10**	7	**6**	6	8
r_2	4	**8**	4	**10**	**9**
ϕ	4	2	1	3	1

(a) $C_1 = \frac{10+6}{2} + \frac{8+10+9}{3} = 17$

	s_1	s_2	$\underline{s_3}$	s_4	s_5
r_1	**10**	7	6	6	8
r_2	4	**8**	4	**10**	**9**
ϕ	4	2	**1**	3	1

(b) $C_2 = 10 + \frac{8+10+9}{3} + 1 = 20$

	s_1	$\underline{s_2}$	s_3	s_4	s_5
r_1	**10**	7	6	6	8
r_2	4	8	4	**10**	**9**
ϕ	4	**2**	**1**	3	1

(c) $C_3 = 10 + \frac{10+9}{2} + 2 + 1 = 22.5$

	s_1	s_2	s_3	s_4	$\underline{s_5}$
r_1	**10**	7	6	6	8
r_2	4	8	4	**10**	9
ϕ	4	**2**	**1**	3	**1**

(d) $C_4 = 10 + 10 + 2 + 1 + 1 = 24$

	s_1	s_2	s_3	s_4	s_5
r_1	10	7	6	6	**8**
r_2	4	8	4	**10**	9
ϕ	**4**	**2**	**1**	3	1

(e) $C^* = 8 + 10 + 4 + 2 + 1 = 25$

Table 3.1: Example with 5 source-destination pairs and 2 relays

The design of the optimal algorithm for RAP is based on Lemma 3.1 and Lemma 3.2.

Lemma 3.1. *Let $\mathscr{A}$ be a relay assignment, where relay node $r_j \in \mathscr{R}$ is assigned to $n_j > 1$ source nodes. Let $s_i \in \mathscr{S}_j$ be the source node with the minimum c_R, i.e., $c_R(s_i, r_j, P_j^r) = \min_{s_k \in \mathscr{S}_j} c_R(s_k, r_j, P_j^r)$. If we let s_i transmit to the destination d_i directly, instead of using r_j,* *while keeping others the same, the total capacity will be increased. That is* $C(\mathscr{S},\mathscr{R},\mathscr{A}',P^r) > C(\mathscr{S},\mathscr{R},\mathscr{A},P^r)$*, where* $\mathscr{A}' = \mathscr{A} \setminus \{(s_i,r_j)\} \cup \{(s_i,r_0)\}$. □

Proof. Let $\mathscr{S}_j' = \mathscr{S}_j \setminus \{s_i\}$. If s_i transmits to d_i directly, we have

$$\begin{aligned}
& C(\mathscr{S},\mathscr{R},\mathscr{A}',P^r) > C(\mathscr{S},\mathscr{R},\mathscr{A},P^r) \\
= \; & c_{DT}(s_i) + \sum_{s_k \in \mathscr{S}_j'} \left(\frac{c_R(s_k,r_j,P_j^r)}{n_j - 1} - \frac{c_R(s_k,r_j,P_j^r)}{n_j} \right) - \frac{c_R(s_i,r_j,P_j^r)}{n_j} \\
= \; & c_{DT}(s_i) + \left(\sum_{s_k \in \mathscr{S}_j'} \frac{c_R(s_k,r_j,P_j^r)}{n_j(n_j-1)} - \frac{c_R(s_i,r_j,P_j^r)}{n_j} \right) \\
\geq \; & c_{DT}(s_i) + \left(c_R(s_i,r_j,P_j^r) \sum_{s_k \in \mathscr{S}_j'} \frac{1}{n_j(n_j-1)} - \frac{c_R(s_i,r_j,P_j^r)}{n_j} \right) \\
= \; & c_{DT}(s_i) + \left(c_R(s_i,r_j,P_j^r)(n_j-1) \cdot \frac{1}{n_j(n_j-1)} - \frac{c_R(s_i,r_j,P_j^r)}{n_j} \right) \\
= \; & c_{DT}(s_i) > 0.
\end{aligned}$$

Therefore, we complete the proof. □

According to Lemma 3.1, we can always improve the system capacity if there exists a relay node shared by more than one source node in the current relay assignment. Unfortunately, the example in Table 3.1 shows that this procedure may lead to a *local optimum*. Nonetheless, Lemma 3.1 implies a nice property pertaining to the optimal relay assignment for RAP.

Lemma 3.2. *Let* $\mathscr{A}^*$ *be an optimal solution to RAP. Each relay node is assigned to at most one source node in* $\mathscr{A}^*$. □

Proof. Assume to the contrary that there exists a relay node $r_j \in \mathscr{R}$ that is assigned to more than one source nodes, i.e. $n_j > 1$. By Lemma 3.1, we can obtain a new relay assignment with strictly higher total capacity by changing one of the source nodes in $\mathscr{S}_j$ to transmit to the destination node directly. This contradicts the optimality of the relay assignment $\mathscr{A}^*$. Therefore, each relay node is assigned to at most one source node in the optimal solution. □

Surprisingly, although our model allows multiple source nodes to share a common relay node, an optimal relay assignment preferably assigns a relay node to at most one source node to achieve the maximum system capacity. On the other hand, we know that each source node will either employ a relay node for CC or transmit to the destination directly, but not both at the same time. This one-to-one matching relation in the optimal solution indicates that we can transform any instance of RAP into that of the *Maximum Weighted Bipartite Matching* (MWBM) problem [146] and solve it using corresponding algorithms.

Now we are ready to present our optimal algorithm for RAP. The pseudo-code is illustrated in Algorithm 3.

The correctness and the computational complexity of Algorithm 3 are guaranteed by Theorem 3.1.

Theorem 3.1. *Algorithm 3 guarantees to find an optimal relay assignment $\mathscr{A}^*(\mathscr{S}, \mathscr{R}, \mathscr{D}, P^r)$ for RAP in time bounded by $O(n^2 m)$.*

Proof. We prove the correctness and the running time separately.

Correctness Analysis: First, Lemma 3.2 assures that each relay node is assigned to at most one source node in the optimal relay assignment. In other words, each source node either transmits directly to its destination or use cooperative communication with the help of a relay node that is not shared with any other source nodes. Therefore, each optimal relay

Algorithm 3: ASGMNT($\mathscr{S},\mathscr{R},\mathscr{D},P^r$)

```
1  Construct a set $\mathscr{U}$ of $n$ vertices corresponding to $\mathscr{S}$;
2  Construct a set $\mathscr{V}$ of $n+m$ vertices corresponding to $\mathscr{D}\cup\mathscr{R}$;
3  Construct a set $\mathscr{E}$ of edges, where $(s_i,v)\in\mathscr{E}$ if $v=d_i$ or $v\in\mathscr{R}$;
4  for $i=1$ to $n$ do
5      $w(s_i,d_i)\leftarrow c_{DT}(s_i)$;
6  end
7  for $\forall s_i\in\mathscr{U}$ and $\forall r_j\in\mathscr{R}$ do
8      $w(s_i,r_j)\leftarrow c_R(s_i,r_j,P_j^r)$;
9  end
10 Apply an MWBM algorithm to find a maximum weighted matching $\mathscr{M}^*$ in graph
   $G=(\mathscr{U},\mathscr{V},w)$;
11 $\mathscr{A}^*\leftarrow\emptyset$;
12 for $(s_i,v)\in\mathscr{M}^*$ do
13     if $v\in\mathscr{R}$ then $\mathscr{A}^*\leftarrow\mathscr{A}^*\cup\{(s_i,v)\}$;
14     else $\mathscr{A}^*\leftarrow\mathscr{A}^*\cup\{(s_i,r_0)\}$;
15 end
16 return $\mathscr{A}^*$.
```

assignment can be mapped to a matching in the graph G constructed from Line 1 to Line 9. Assume that there exists another relay assignment $\mathscr{A}'$ resulting in a higher capacity than $\mathscr{A}^*$ returned by Algorithm 3. If we map it back to a matching in graph G, we obtain a matching $\mathscr{M}'$ with higher weight than that of $\mathscr{M}^*$ corresponding to $\mathscr{A}^*$. It contradicts the fact that $\mathscr{M}^*$ is a maximum weighted matching in G. Hence, $\mathscr{A}^*$ is an optimal relay assignment.

Running Time: Note that the most time consuming component in Algorithm 3 is the MWBM algorithm in Line 10. Thus we focus our analysis on this algorithm. The MWBM problem can be solved using a modified shortest path search in augmenting path algorithm. If the Dijkstra algorithm with Fibonacci heap is used, the running time is $O(\min\{|\mathscr{U}|,|\mathscr{V}|\}\cdot((|\mathscr{U}|+|\mathscr{V}|)\log(|\mathscr{U}|+|\mathscr{V}|)+|\mathscr{E}|))$ [25]. Since $|\mathscr{U}|=n$, $|\mathscr{V}|=n+m$ and $|\mathscr{E}|=nm+n=O(nm)$, the running time of Algorithm 3 is bounded by $O(n(n+m)\log(n+m)+n^2m)=O(n^2m)$. □

3.4.2 Mechanism Design for Selfish Users

System capacity maximization is only desirable from a global point of view, not from the point of view of an individual selfish user. Yet most wireless devices in the network are owned by independent profit-maximizing entities. In this section, we use the term *selection* instead of *assignment*, because selection is from the user's point of view while assignment is from the system's point of view. When selfish users have their own preferences on relay selection, several questions may arise: Is there a *stable state*, where no user has the incentive to deviate from its current selection? How can users reach such a state? If the system performance is not optimized in the stable state, how can the system administrator exert influence on the relay selection to achieve social optimum? These questions will be the focus of this part.

To study the relay selection problem with selfish entities, we model it by a game, called *Relay Selection Game* (RSG). In this game, the source nodes are *players*, because they make relay selections. The *strategy* of each player is its relay selection $\gamma_i \in \bar{\mathscr{R}}$. The strategy profile $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ is a vector of all players' strategies. Let $\gamma_{-i} = (\gamma_1, \ldots, \gamma_{i-1}, \gamma_{i+1}, \ldots, \gamma_n)$ denote the strategy profile excluding player s_i's strategy. Hence, $\gamma = (\gamma_i, \gamma_{-i})$ is a strategy profile where s_i plays γ_i and others play γ_{-i}. Given a strategy profile γ, we can construct the corresponding relay assignment $\mathscr{A} = \{(s_1, \gamma_1), (s_2, \gamma_2), \ldots, (s_n, \gamma_n)\}$. Given a relay assignment $\mathscr{A} = \{(s_1, r_{j_1}), (s_2, r_{j_2}), \ldots, (s_n, r_{j_n})\}$, we have the corresponding strategy profile γ, where $\gamma_i = r_{j_i}$ for each $s_i \in \mathscr{S}$. In this game, each player s_i selects a strategy γ_i to maximize its own *utility*, which is defined as its achieved capacity $u_i^s(\gamma) = c(s_i, \gamma_i, \mathscr{A}, P^r)$. If $u_i^s(\gamma_i, \gamma_{-i}) > u_i^s(\gamma_i', \gamma_{-i})$, we say that player s_i prefers γ_i to γ_i' when others play γ_{-i} .

As a motivation to the design of our payment mechanism, we show that an NE of

RSG is not necessarily social optimal. Note that RSG is closely related to the *Congestion Game* introduced by Rosenthal [123]. Specifically, RSG can be reduced to the *Congestion Game with Player-specific Payoff Function*, which was studied by Milchtaich [96]. Due to space limitations, we make reference to [96] for the existence proof of NE and the algorithm for computing an NE.

Recall that C^* is the optimal system capacity. A simple example in Table 3.2 shows that the selfishness of players can degrade the system performance by half.

	s_1	s_2	s_3	$\cdots$	s_n
r_0	1	1	1	$\cdots$	1
r_1	10	1	1	$\cdots$	5
r_2	10	10	1	$\cdots$	1
r_3	1	5	10	$\cdots$	1
r_4	1	1	5	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_m	1	1	1	$\cdots$	10

(a) $C^{ne} = 10 + 5(n-1)$

	s_1	s_2	s_3	$\cdots$	s_n
r_0	1	1	1	$\cdots$	1
r_1	10	1	1	$\cdots$	5
r_2	10	10	1	$\cdots$	1
r_3	1	5	10	$\cdots$	1
r_4	1	1	5	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_m	1	1	1	$\cdots$	10

(b) $C^* = 10n$

Table 3.2: Example with $POA = \frac{10+5(n-1)}{10n} \approx \frac{1}{2}$

To achieve the optimal relay assignment, we need to exert influence on players' selection of relay nodes. Here we require players to make payments for using relaying service.

As in many existing scheme designs, we assume *virtual currency* exists in the system. Each source node (player) needs to pay certain amount of currency to the administrator based on its relay node selection. In particular, given the strategy profile γ and the corresponding $\mathscr{A}$, we define the payment of player s_i as

$$p_i^s = \begin{cases} c(s_i,\gamma_i,\mathscr{A},P^r) + \left(g(\gamma_i,\gamma_i^*) - \frac{1}{n-1}\sum\limits_{k\neq i} g(\gamma_k,\gamma_k^*) \right), & \text{if } \gamma_i \neq r_0, \\ g(\gamma_i,\gamma_i^*) - \frac{1}{n-1}\sum_{k\neq i} g(\gamma_k,\gamma_k^*), & \text{if } \gamma_i = r_0. \end{cases}$$

Here $g(\gamma_i,\gamma_i^*) = l\cdot|x-y|$, where $\gamma_i = r_x$ and $\gamma_i^* = r_y$, $l = \max_{s_i\in\mathscr{S}} c_{DT}(s_i)+\varepsilon$ and $\varepsilon > 0$ is a constant. In other words, $g(\gamma_i,\gamma_i^*)$ is equal to l times the difference between the indices of the relay node selected by s_i and the relay node assigned in the optimal solution. Intuitively, a source node needs to pay for using relaying service if it selects a relay node. Each source node also pays (or receives) a penalty (resp. bonus) depending on how much more (resp. less) it deviates from the optimal strategy γ^* than others. Here γ^* is the strategy profile corresponding to the optimal solution $\mathscr{A}^*$ of RAP computed by Algorithm 3. The utility of player s_i is then defined as

$$u_i^s(\gamma_i,\gamma_{-i}) = c(s_i,\gamma_i,\mathscr{A},P^r) - p_i^s. \tag{3.1}$$

A similar payment mechanism was also used by Wu *et al.* to solve a different problem [150]. We call the Relay Selection Game with utility function (3.1) the *Incentive-added Relay Selection Game* (IRSG). Next we prove that γ^* is an SDSE in IRSG.

Theorem 3.2. *Let γ^* be the strategy profile corresponding to the optimal solution $\mathscr{A}^*$ of RAP. Then γ^* is an SDSE for IRSG. Therefore, γ^* is the unique NE of IRSG.* □

Proof. To prove this theorem, it suffices to prove that $\forall s_i \in \mathscr{S}$, $\forall\gamma_{-i},\forall\gamma_i \neq \gamma_i^*$, we must have $u_i^s(\gamma_i^*,\gamma_{-i}) > u_i^s(\gamma_i,\gamma_{-i})$. Plugging the payment p_i^s into (3.1), we have

$$u_i^s(\gamma_i,\gamma_{-i}) = \begin{cases} \frac{1}{n-1}\sum_{k\neq i} g(\gamma_k,\gamma_k^*) - g(\gamma_i,\gamma_i^*), \text{if } \gamma_i \neq r_0, \\ c_{DT}(s_i) - \left(g(\gamma_i,\gamma_i^*) - \frac{1}{n-1}\sum\limits_{k\neq i} g(\gamma_k,\gamma_k^*) \right), \text{if } \gamma_i = r_0. \end{cases}$$

Assume player s_i plays strategies γ_i^* and $\gamma_i \neq \gamma_i^*$, respectively. We consider all the possible cases:

Case 1: $\gamma_i^* \neq r_0$ and $\gamma_i \neq r_0$.

$$u_i^s(\gamma_i^*, \gamma_{-i}) - u_i^s(\gamma_i, \gamma_{-i}) = g(\gamma_i, \gamma_i^*) - g(\gamma_i^*, \gamma_i^*) = g(\gamma_i, \gamma_i^*) > 0,$$

where the second equality and the last inequality follow from the definition of $g(\cdot,\cdot)$ and the assumption that $\gamma_i \neq \gamma_i^*$.

Case 2: $\gamma_i^* \neq r_0$ and $\gamma_i = r_0$.

$$u_i^s(\gamma_i^*, \gamma_{-i}) - u_i^s(\gamma_i, \gamma_{-i}) = g(\gamma_i, \gamma_i^*) - c_{DT}(s_i) - g(\gamma_i^*, \gamma_i^*) = g(\gamma_i, \gamma_i^*) - c_{DT}(s_i) > 0,$$

where the last inequality follows from the definition of $g(\cdot,\cdot)$.

Case 3: $\gamma_i^* = r_0$ and $\gamma_i \neq r_0$.

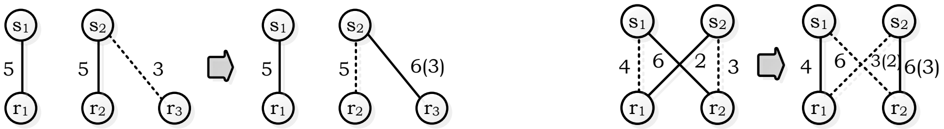
$$u_i^s(\gamma_i^*, \gamma_{-i}) - u_i^s(\gamma_i, \gamma_{-i}) = c_{DT}(s_i) + g(\gamma_i, \gamma_i^*) - g(\gamma_i^*, \gamma_i^*) = c_{DT}(s_i) + g(\gamma_i, \gamma_i^*) > 0.$$

We have proved that γ^* is an SDSE of IRSG. Hence, γ^* is the unique NE of IRSG.

□

3.4.3 Mechanism Design to Prevent Relay Nodes from Cheating

Relay nodes involved in the final assignment help source nodes with cooperative communications at the cost of their own energy and other resources. Without an attractive incentive, a relay node may not be willing to participate in cooperative communications. A naive solution to this problem is to pay each relay node the achieved capacity of cooperative communications involving it (while the relay assignment $\mathscr{A}$ is computed based on the reported transmission power, the achieved capacity is computed based on the true transmission power and the relay assignment $\mathscr{A}$). However, such a simple payment mechanism could result in relay nodes' lying about their transmission power. For example, a relay node would not be selected if it reports its transmission power honestly, but could be selected if it reports a larger transmission power instead. Likewise, a relay node would be assigned to cooperate with a source node, resulting in a small capacity, if it reports its transmission power honestly. But it could cooperate with another source node by lying, resulting in a larger capacity. These two examples are shown in Figure 3.2. In both examples, the relay node receives a larger payment by lying about its transmission power.



(a) r_2 increases its payment from 0 to 3. The system capacity is decreased from 5 to 3.

(b) r_2 increases its payment from 2 to 3. The system capacity is decreased from 8 to 7.

Figure 3.2: Examples showing that a relay node can increase its payment by lying. Solid links represent the relay assignment. The numbers beside the links represent the achieved capacities calculated based on reported transmission power (outside the parentheses) and based on the true transmission power (inside the parentheses) if it is different from the reported transmission power.

Obviously, the dishonest behavior of relay nodes may influence the relay assignment and further degrade the system performance. Hence it is essential to design a payment mechanism such that every relay node will report its transmission power truthfully to maximize its payment.

In our design, we assume that each relay node r_j is an agent and the type of r_j is its transmission power P^r_j. Before the relay assignment, each relay node r_j reports a transmission power T_j, which may or may not be equal to P^r_j. Let $P^r = (P^r_1, P^r_2, \ldots, P^r_m)$ be the *true transmission power* profile and $T = (T_1, T_2, \ldots, T_m)$ the *reported transmission power* profile. **ASGMNT**$(\mathscr{S}, \mathscr{D}, \mathscr{R}, T)$ (illustrated in Algorithm 3) is then applied to compute an optimal relay assignment $\mathscr{A}^*(T)$, which is optimal with respect to T. According to Lemma 3.2, each relay node r_j is assigned to at most one source node under $\mathscr{A}^*(T)$. Under $\mathscr{A}^*(T)$, let $\sigma_j(T) \in \bar{\mathscr{S}}$ denote the source node, to which r_j is assigned. If $\sigma_j(T) = s_0$, it indicates that r_j is not assigned to any source node. Let $\sigma(T) = (\sigma_1(T), \sigma_2(T), \ldots, \sigma_m(T))$ be the source nodes corresponding to all the relay nodes in $\mathscr{R}$. Let $\Psi(\mathscr{S}, \mathscr{R}, T)$ denote the optimal capacity of the system consisting of $\mathscr{S}$ and $\mathscr{R}$ based on T, i.e., $\Psi(\mathscr{S}, \mathscr{R}, T) = C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T), T)$. Let $\mathscr{S}_{-s_i} = \mathscr{S} \setminus \{s_i\}$, $\mathscr{R}_{-r_j} = \mathscr{R} \setminus \{r_j\}$, and $T_{-j} = (T_1, \ldots, T_{j-1}, T_{j+1}, \ldots, T_m)$. We define the payment to relay node r_j (for a given T) by the following

$$p_j^r(T) = \begin{cases} 0, & \sigma_j(T) = s_0, \\ c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r) - (\Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) & \\ -\Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j})), & o/w, \end{cases} \tag{3.2}$$

where $c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r)$ is the achieved capacity in the cooperative communication, and $\Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) - \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j})$ is a charge determined by the system administrator, based on T.

Before proving the properties of the designed payment mechanism, we note the fact that

$$c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j}) = C(\mathscr{S}, \mathscr{R}, \mathscr{A}(T'), T'), \tag{3.3}$$

where $T' = (P_j^r, T_{-j})$ and $\mathscr{A}(T')$ is *some* relay assignment for the RAP instance given by $(\mathscr{S}, \mathscr{D}, \mathscr{R}, T')$. The intuition behind this fact is that, after the optimal assignment $\mathscr{A}^*(T)$ is computed, the values of $c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r)$ and $\Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j})$ are independent of T_j, and only dependent on T'. In addition, their sum is the system capacity under $\mathscr{A}(T')$, which may be different from $\mathscr{A}^*(T')$. Similarly, if relay node r_j reports its true transmission power P_j^r and other relay nodes report T_{-j}, we have

$$c(\sigma_j(T'), r_j, \mathscr{A}^*(T'), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T')}, \mathscr{R}_{-r_j}, T_{-j}) = C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T'), T'). \tag{3.4}$$

Theorem 3.3. HERA *is individually rational.* □

Proof. Let r_j be any relay node and $T' = (P^r_j, T_{-j})$. Then the payment to relay node r_j is

$$\begin{align}
& p^r_j(T') \notag \\
= \;& c(\sigma_j(T'), r_j, \mathscr{A}^*(T'), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T')}, \mathscr{R}_{-r_j}, T_{-j}) - \Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) \notag \\
= \;& C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T'), T') - \Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) \tag{3.5} \\
= \;& C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T'), T') - C(\mathscr{S}, \mathscr{R}_{-r_j}, \mathscr{A}^*(T_{-j}), T_{-j}) \notag \\
\geq \;& 0, \notag
\end{align}$$

where (3.5) follows from (3.4). This completes the proof. □

Theorem 3.4. HERA *is truthful.* □

Proof. Assume r_j reports a transmission power $T_j \neq P^r_j$. Let $T' = (P^r_j, T_{-j})$. Then the difference between its received payment and that when reporting truthfully is

$$\begin{align}
& p^r_j(T') - p^r_j(T) \notag \\
= \;& c(\sigma_j(T'), r_j, \mathscr{A}^*(T'), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T')}, \mathscr{R}_{-r_j}, T_{-j}) \notag \\
& - \Big(c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j}) \Big) \notag \\
= \;& C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T'), T') - \Big(c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r) + \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j}) \Big) \tag{3.6} \\
= \;& C(\mathscr{S}, \mathscr{R}, \mathscr{A}^*(T'), T') - C(\mathscr{S}, \mathscr{R}, \mathscr{A}(T'), T') \tag{3.7} \\
\geq \;& 0, \tag{3.8}
\end{align}$$

where (3.6) follows from (3.4), (3.7) follows from (3.3), and (3.8) follows from the optimality of $\mathscr{A}^*(T')$. □

We have designed a payment mechanism to charge source nodes for using relaying service. Now we designed another payment mechanism to pay relay nodes for providing relaying service. A question arising naturally is whether HERA is *budget-balanced*. That is, whether the payment collected from all the source nodes is enough to pay all the relay nodes. The following theorem confirms the budget-balance property of HERA.

Theorem 3.5. HERA *is budget-balanced.* □

Proof. By Theorem 3.2, we know that all the source nodes will follow the optimal relay assignment. Let $T = (T_1, T_2, \dots, T_m)$ be the reported transmission power profile. Let γ^* be the strategy profile of source nodes corresponding to $\mathscr{A}^*(T)$. Therefore, the total payment collected from all source nodes is

$$p^s = \sum_{s_i \in \mathscr{S}} p_i^s = \sum_{s_i \in \mathscr{S}, \gamma_i^* \neq r_0} c(s_i, \gamma_i^*, \mathscr{A}^*(T), P^r). \tag{3.9}$$

The total payment paid to all the relay nodes is

$$\begin{aligned} p^r &= \sum_{r_j \in \mathscr{R}} p_j^r \\ &= \sum_{r_j \in \mathscr{R}, \sigma_j(T) \neq s_0} c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r) \\ &\quad - \sum_{r_j \in \mathscr{R}, \sigma_j(T) \neq s_0} \Big(\Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) - \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j})\Big), \end{aligned}$$

where the second equality follows the fact that

$$\textstyle\sum_{s_i \in \mathscr{S}, \gamma_i^* \neq r_0} c(s_i, \gamma_i^*, \mathscr{A}^*(T), P^r) = \sum_{r_j \in \mathscr{R}, \sigma_j(T) \neq s_0} c(\sigma_j(T), r_j, \mathscr{A}^*(T), P^r).$$

The profit of the administrator is

$$p^s - p^r = \sum_{r_j \in \mathscr{R}, \sigma_j(T) \neq s_0} \Big(\Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) - \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j})\Big) \geq 0,$$

where the inequality follows from the fact that

$$\Psi(\mathscr{S}, \mathscr{R}_{-r_j}, T_{-j}) - \Psi(\mathscr{S}_{-\sigma_j(T)}, \mathscr{R}_{-r_j}, T_{-j}) \geq 0.$$

We finished the proof of the theorem. □

We considered a wireless network, where wireless nodes are uniformly randomly distributed in a $1000m \times 1000m$ square. We followed the same parameter settings as in [127]. The only exception was the transmission power, which in our setting is uniformly distributed over $(0,1]$, i.e., $P_i^s, P_j^r \in (0,1]$ Watt for all $s_i \in \mathscr{S}$ and $r_j \in \mathscr{R}$. We set the bandwidth W to 22 MHz for all channels. For the transmission model, we assumed that the path loss exponent $\alpha = 4$ and the abient noise $N_0 = 10^{-10}$. In most of the experiments, we varied both n and m from 50 to 400 with increment of 50. For each setting, we randomly generated 100 instances and averaged the results.

Assignment Algorithm

Since this is the first work on the design of relay assignment scheme for cooperative networks with the objective to maximize the total capacity, we compared our algorithm with the algorithms listed below.

- *Greedy Assignment Algorithm (Greedy):* This algorithm proceeds iteratively. In each iteration, it greedily assigns a relay node to the source node or lets the source node transmit directly, such that the system capacity under the current assignment is maximized.
- *Direct Transmission Algorithm (DT):* In this algorithm, each source node transmits to its destination directly. The system capacity under this assignment is $C = \sum_{s_i \in \mathscr{S}} c_{DT}(s_i)$. *DT* serves as a lower bound of the system capacity of the network under any relay assignment.
- *ORA [127]:* The basic idea of *ORA* is to adjust the assignment iteratively, starting from any arbitrary initial assignment. In each iteration, *ORA* identifies the source

node with currently minimum capacity among all the source nodes and searches a better relay node for it. Although *ORA* is not intentionally designed for RAP, we include it in the comparison for the sake of completeness.

Cheating Report Distribution

We assume that a relay node can cheat by reporting a transmission power larger than its true transmission power. If P_j^r is the transmission power of relay node r_j, then its reported transmission power is $P_j^r + \delta$, where δ is a random number uniformly distributed over $(\Delta, \Delta + 1]$ and Δ is a parameter.

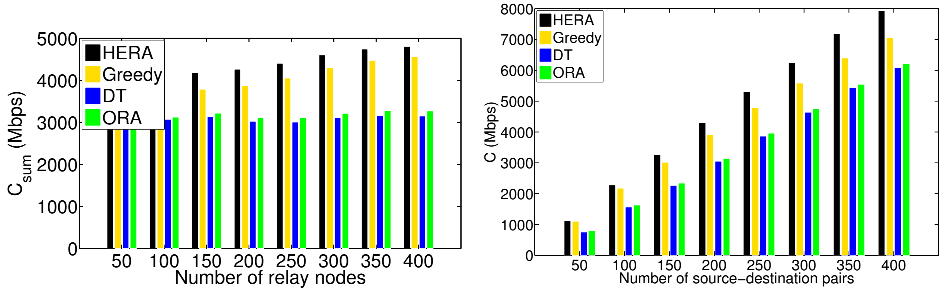
The performance metrics in the experiments include the system capacity and the number of cooperative communications.

Evaluation of Assignment Algorithms

Figure 3.3 shows the system capacity under the assignments returned by different algorithms. As expected, HERA has the best performance while *DT* has the worst. Surprisingly, the performance of *Greedy* is only slightly worse than that of HERA, especially when $m > n$. The reason is that some source nodes may not need to compete with other source nodes for their best relay nodes. Therefore, we may have the same assignment for these source nodes in both HERA and *Greedy*. Another observation is that when the number of relay nodes exceeds that of the source nodes, the system capacity tends to keep the same.

Impact of Selfishness on System Performance

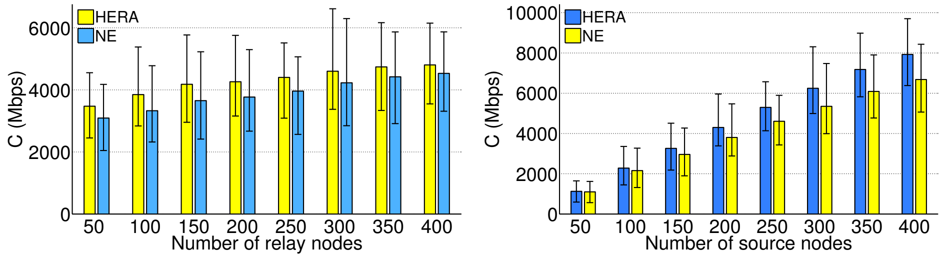
We have shown in an example in Table 3.2 that the *POA* of the Relay Selection Game can be as small as $\frac{1}{2}$. We turn to evaluate how the selfish behavior of source nodes affects the system performance in randomly generated networks. Figure 3.4 plots the ca-



(a) Varying m

(b) Varying n

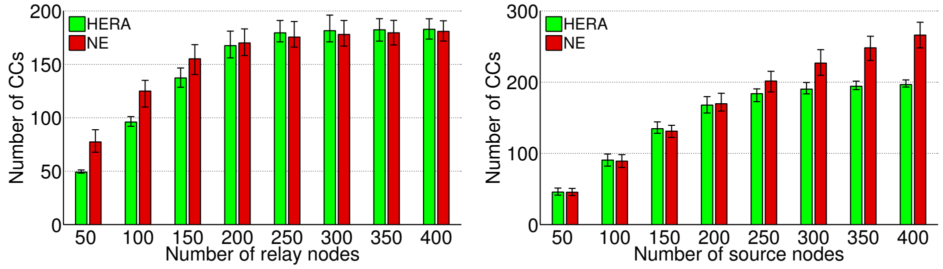
Figure 3.3: Comparison among relay assignment algorithms. For (a), $n = 200$. For (b), $m = 200$.



(a) Varying m

(b) Varying n

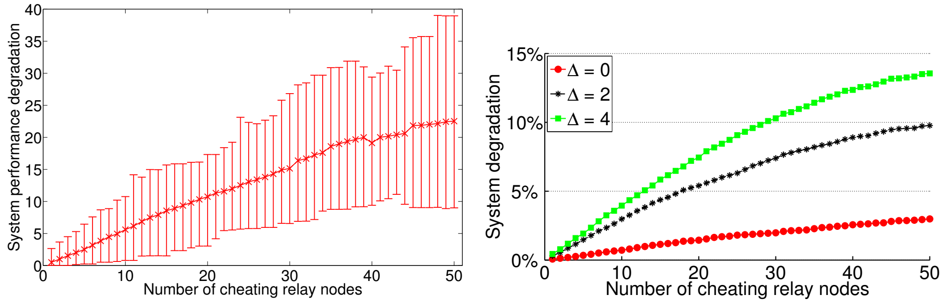
Figure 3.4: Impact of selfish behavior of source nodes on system capacity. For (a), $n = 200$. For (b), $m = 200$. The maximum and minimum values among 100 random instances are shown as error bars.



(a) Varying m

(b) Varying n

Figure 3.5: Number of source nodes using CC. For (a), $n = 200$. For (b), $m = 200$. The maximum and minimum values among 100 random instances are shown as error bars.



(a) System capacity

(b) Impact of Δ

Figure 3.6: Impact of cheating behavior of relay nodes on system capacity where $n = m = 50$. For (a), $\Delta = 4$.

pacities of the systems when HERA is applied and when it is not. We note that the degradation of NE over HERA decreases with the increase of m, as shown in Figure 3.4(a). This is because source nodes do not need to compete with each other for relay nodes when there are enough relay nodes. Another observation is that the degradation becomes worse with the increase of n, as shown in Figure 3.4(b). This can be explained by the same reason above, as source nodes sharing the same relay node can improve the system capacity if one of them changes to direct transmission. Figure 3.5 illustrates the number of source nodes using CC in both HERA and NE. We observe that, when the number of relay nodes is more than that of the source nodes, there are more source nodes competing relay nodes for CC in NE than there are in HERA. This verifies our analysis on the results shown in Figure 3.4.

Impact of Cheating on System Performance

Next we focus on the impact of cheating behavior of relay nodes on system performance. Figure 3.6 shows the system degradation due to the cheating behavior in a network consisting of 50 source nodes and 50 relay nodes. In Figure 3.6(a), we set $\Delta = 4$. Our first observation is that when the number of cheating relay nodes is small, the system performance is not affected significantly. This is because a small number of cheating relay nodes will unlikely affect the matching process in the algorithm. Another observation is that the degradation increases with the increase of the number of cheating relay nodes, which is as expected.

We then evaluate the impact of parameter Δ. Intuitively, the larger Δ is, the more a relay node can untruthfully report its transmission power. Figure 3.6(b) shows the system performance degradation in the networks with different values of Δ. We see that the degradation increases when the value of Δ increases. The reason is that a relay node reporting a large transmission power has a high probability to be selected in the relay assignment. However, its true transmission power may be very small. Hence the final system capacity is degraded.

3.6 Conclusion

In this section, we designed HERA, an integrated optimal relay assignment scheme for cooperative networks. It is composed of three components: an optimal relay assignment algorithm, a payment mechanism to charge source nodes for using relaying service, and a payment mechanism to pay relay nodes for proving relaying service. HERA induces selfish source nodes to converge to the optimal assignment and prevents relay nodes from reporting transmission power untruthfully to gain profit. In addition, HERA satisfies budget-balance property, which means the payment collected from source nodes is no less than the payment paid to relay nodes.

Chapter 4

Channel Allocation in Non-Cooperative Wireless Networks

The development of the IEEE 802.11a/b/g standards has spurred the emergence of broadband wireless networks. Due to the common transmission media shared by communication devices, interference arises if communication devices are operating on the same frequency. It has been shown that interference severely limits the network capacity [62]. Frequency Division Multiple Access (FDMA) is a widely used technique to enable multiple devices to share a communication medium. In FDMA, the available bandwidth is divided into multiple sub-bands, named channels. Using multiple channels in multi-radio wireless networks can greatly alleviate the interference and improve the network throughput [117]. Ideally, if there are a sufficient number of channels and each device assigns different channels to its radios, there would be no interference in the network at all. However, since the spectrum is a scarce resource, we are only allowed to divide the available bandwidth into a limited number of channels. For example, there are 3 and 12 non-overlapping channels for the IEEE 802.11b/g standards in 2.4 GHz and the IEEE 802.11a standard in 5 GHz, respectively. A fundamental problem in multi-radio multi-channel (MR-MC) wireless networks is how to allocate channels to radios, which is commonly referred to as the *channel allocation* problem (also known as the channel assignment problem).

While tremendous efforts have been made on the channel allocation problem, most of them are on cooperative networks where devices are assumed to be cooperative and unselfish; however this assumption may not hold in practice. Usually, a wireless device is owned by an independent individual, who is only interested in selfishly maximizing its own profit without respecting the system performance. There are a few works considering non-cooperative networks [42, 49, 149, 150]. However, all of these works only consider the problem in a single collision domain, which means all the transmissions will interfere with each other if they are on the same channel.

4.1 Introduction

In this section, we study the channel allocation problem in non-cooperative MR-MC networks with *multiple collision domains*. To characterize the network with multiple collision domains, we introduce interference models into the network. The results in this work are independent of the interference model adopted as long as the model is defined on pairs of communications, for example, the *protocol interference model* is used in this work. We model the channel assignment problem in non-cooperative MR-MC wireless networks as a strategic game. We show that the game may oscillate indefinitely when there are no exogenous factors to influence players' behavior. This possible oscillation can result in significant communication overhead and the degradation of the system performance. To avoid this undesirable outcome, we develop a charging scheme to induce players' behavior. The design of the charging scheme ensures the convergence to a Nash Equilibrium (NE). Players are in an NE if no player can improve its utility by changing its strategy unilaterally. Although NE is usually not social optimal, we can prove that the system performance in an NE is guaranteed to be at least a factor of the system performance in the optimal solution.

We summarize our main contributions as follows:

- To the best of our knowledge, we are the *first* to study the channel allocation problem in non-cooperative MR-MC wireless networks with *multiple collision domains*. We model the problem as a strategic game, called *ChAlloc*.
- We show that the ChAlloc game can result in an oscillation, where players keep changing their strategies back and force trying to improve their utilities.

- To avoid the possible oscillation, we design a charging scheme to influence players' behavior. We prove that, under the charging scheme, the ChAlloc game converges to an NE. We also prove that the system performance in an NE is guaranteed to be at least $(1 - \frac{\bar{r}}{h})$ of the system performance in the optimal solution, where $\bar{r}$ is the maximum number of radios equipped on wireless devices and h is the number of available channels.

- We design a localized algorithm for players to find an NE and prove that it takes $O(\bar{r}hn^3(n + \log h))$ time for the ChAlloc game to converge to an NE, where n is the number of players.

- In order to verify our proof of the system performance in an NE, we give an LP-based algorithm to derive efficiently computable upper bounds on the optimal solution.

- Through extensive experiments, we validate our analysis of the possible oscillation in the ChAlloc game lacking the charging scheme and confirm the proof of the convergence of the ChAlloc game with the charging scheme. The results also show that the system performance in an NE is very close to the optimal solution and thus verify our proof of the system performance.

The remainder of this section is organized as follows. In Section 4.2, we review the current literature on the channel allocation problem. In Section 4.3, we present the system model considered in our work and formulate the channel allocation problem as a game, called *ChAlloc*. In Section 4.4, we first use an example to show that it is possible for the ChAlloc game to oscillate endlessly when there are no exogenous factors to influence players' behavior. Then we design a charging scheme to induce players to converge to an NE, compute the price of anarchy of the ChAlloc game, and develop a localized algorithm for players. In addition, we give an LP-based algorithm to find an upper bound on the

Table 4.1: Relation to previous work on channel allocation

	Single Collision Domain	Multiple Collision Domains
Cooperative	none	[29, 75, 91, 114, 115, 130, 132, 135]
Non-cooperative	[42, 49, 149, 150]	our work

optimal solution. In Section 4.5, we evaluate the performance of the ChAlloc game through extensive experiments. Finally, we form our conclusion in Section 4.6.

4.2 Relation to Previous Work on Channel Allocation

Most previous works on channel allocation can be categorized into two categories, *channel allocation in cooperative networks* [29, 75, 91, 114, 115, 130, 132, 135] and *channel allocation in non-cooperative networks* [42, 49, 149, 150]. We summarize the related works in Table 4.1.

Channel Allocation in Cooperative Networks: There is a considerable amount of works on the channel allocation problem in Wireless Mesh Networks (WMNs). In [29], Das *et al.* presented two mixed integer linear programming (ILP) models to solve the channel allocation problem in WMNs with the objective to maximize the number of simultaneously transmitting links. However, it is known that solving an ILP is NP-hard. In [114], Ramachandran *et al.* proposed a centralized channel allocation algorithm utilizing a novel interference estimation technique in conjunction with an extension to the conflict graph model, called the multi-radio conflict graph. In [132], Sridhar *et al.* proposed a localized channel allocation algorithm called LOCA, which is a heuristic algorithm. Subramanian *et al.* [135] and Marina *et al.* [91] studied the channel allocation problem where each link is assigned a channel with the constraint that the number of different channels assigned to the links incident on any node is at most the number of radios on that node. Subramanian *et al.* [135] developed a centralized algorithm based on Tabu search and a distributed algorithm based on the Max-K-cut problem. Marina *et al.* [91] proposed a greedy heuristic channel allocation algorithm, termed CLICA. In [75], Ko *et al.* studied the channel allocation problem with a different objective function and proposed a distributed algorithm without any performance guarantee. In [130], Shin *et al.* considered the channel allocation problem to maximize the throughput or minimize the delay, and presented the an allocation scheme, called SAFE, which is a distributed heuristic.

All the above related works are based on the assumption that wireless devices in the network cooperate to achieve a high system performance. However, this assumption might not hold in practice. Usually, a wireless device is owned by an independent individual, who is only interested in selfishly maximizing its own profit without respecting the system performance or considering others' profits.

Channel Allocation in Non-cooperative Networks: Game theory has been widely used to solve problems in non-cooperative wireless networks, for instance, Aloha networks [90] and CSMA/CA networks [16, 76]. Based on a graph coloring game model, Halldórsson *et al.* [64] provided bounds on the *price of anarchy* of the channel allocation game. However, their model does not apply to multi-radio networks.

In an earlier work, Félegyházi *et al.* [42] formulated the channel allocation problem in non-cooperative MR-MC wireless networks as a game, analyzed the existence of Nash Equilibria and presented two algorithms to achieve an NE. Along this line, Wu *et al.* [150] introduced a payment formula to ensure the existence of a *strongly dominant strategy equilibrium* (SDSE). Furthermore, when the system converges to an SDSE, it also achieves global optimality in terms of system throughput. In [49], Gao *et al.* extended the problem to multi-hop networks and also addressed coalition issues. Most recently, Wu *et al.* [149] studied the problem of adaptive-width channel allocation in non-cooperative MR-MC wireless networks, where contiguous channels may be combined to provide a better utilization of the available channels. However, all the above results can only be applied to a single collision domain, without considering multiple collision domains. In this work, we fill this void and study the channel allocation problem in non-cooperative MR-MC networks with multiple collision domains.

4.3 Network Model and Game Formulation

The network model in this work closely follows the models in [42, 49, 149, 150]. We consider a static wireless network consisting of a set $\mathscr{L} = \{L_1, L_2, \ldots, L_n\}$ of n communication links. Each link L_i is modeled as an undirected link between two nodes v_i and u_i, where v_i and u_i denote two wireless devices communicating with each other. The use of the undirected link model reflects the fact that the IEEE 802.11 DCF requires the sender to be able to receive the acknowledgement message from the receiver for every transmitted packet. Since links are undirected, two nodes are able to coordinate to select the same channels for communication. As in [42, 49, 149, 150], we assume that the links are backlogged and always have packets to transmit. Each wireless device is equipped with multiple radio interfaces. We further assume that each transmission must be between two radios, of which one functions as a transmitter and the other as a receiver. Thus, it is reasonable to assume that both nodes of L_i have the same number of radios, denoted by r_i. We assume the wireless devices have the same maximum transmission power, but each of the devices adjusts its actual data transmission power according to the length, denoted by l_i, of the transmission link. Let R denote the transmission range under the maximum transmission power. Furthermore, there are $h > 1$ orthogonal channels available in the network, e.g. 12 orthogonal channels in the IEEE 802.11a protocol. We denote the set of channels by $\mathscr{C} = \{c_1, c_2, \ldots, c_h\}$.

To communicate, two nodes of a link have to tune at least one of their radios to the same channel(s). Parallel communications are allowed between two nodes if they share multiple channels on radios. In order to avoid the co-radios interference in a device [49], we assume that different radios on a node should be tuned to different channels. Therefore, it is reasonable to assume that $r_i < h$ for all $L_i \in \mathscr{L}$ as it would be straightforward to allocate channels otherwise. Allocating each channel to at most one radio has also been proved to be a necessary condition to maximize the device's transmission data rate [42, 49, 149].

Due to the common transmission medium, wireless transmission along a communication link may interfere with the transmissions along other communication links, especially those within its vicinity. While existing works [42, 49, 149, 150] have studied the channel allocation problem in non-cooperative networks for both single-hop and multi-hop models, all the results can only be applied to a single collision domain. In other words, they assume that all transmissions interfere with each other if they share at least one channel. However, the strength of a wireless transmission signal decays exponentially with respect to the distance it travels from the transmitter. Therefore the signal from a distant transmission is, if not negligible, not destructive enough to prevent another transmission from succeeding. In this work, we generalize the channel allocation problem to networks with multiple collision domains.

To characterize networks with multiple collision domains, an appropriate interference model is necessary. Various interference models have been proposed in the literature, for example, the primary interference model [63], the protocol interference model [62, 70], and the physical interference model (a.k.a SINR interference model) [62, 70]. The results in this work are independent of the specific interference model used *as long as the interference model is defined on pairs of communication links*. For the sake of presentation, the *protocol interference model* is adopted throughout this work. This model has been used by most of the works on channel allocation problems [1, 6, 91, 117, 135, 136]. In this model,

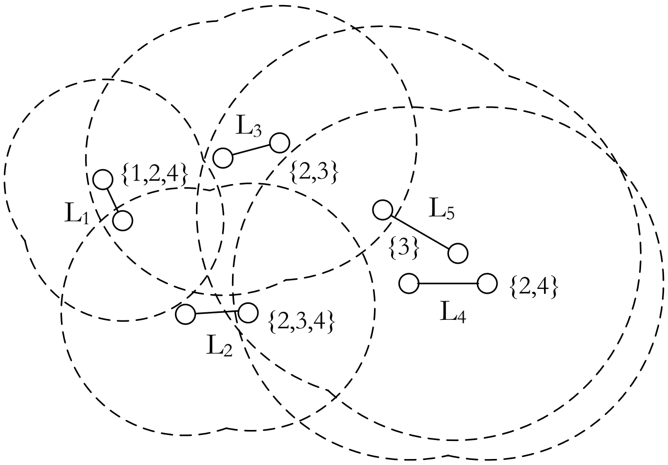


Figure 4.1: Network of 5 links, where $\mathscr{L} = \{L_1, L_2, L_3, L_4, L_5\}$, $r_1 = 3, r_2 = 3, r_3 = 2, r_4 = 2, r_5 = 1$ and $\mathscr{C} = \{1, 2, 3, 4\}$.

each node has an interference range γl_i, which is at least as large as the transmission range (equal to l_i), i.e., $\gamma \geq 1$. We assume that $l_i \leq \frac{R}{\gamma}$, for any L_i. Any node u will be interfered by node v if u is within v's interference range. We can imagine that, associated with each L_i, there is an interference disk D_{u_i} centered at u_i, and an interference disk D_{v_i} centered at v_i. The union of D_{u_i} and D_{v_i}, denoted by $D_{u_i} \cup D_{v_i}$, constitutes the *interference area* of L_i. Link L_i *interferes* with link L_j if and only if either of v_j and u_j is in $D_{u_i} \cup D_{v_i}$, and two links share at least one common channel. Before the channels on the radios are known, we can only say that L_i *potentially interferes* with L_j. As an illustrating example, Figure 4.1 shows a 5-link network, where dashed peanut-shaped curves represent the boundaries of interference areas (numbers in parentheses will be explained later). In this example, L_3 potentially interferes with L_1 while L_1 cannot interfere with L_3.

Conflict graphs are widely used to facilitate the design of channel allocation algorithms [70, 91, 98, 114, 135]. We use a similar concept, called *potential interference graph* (PING), to characterize the interfering relationships among the links. Different from the conflict graph, the edges in the PING are directed due to the heterogeneity of the in-

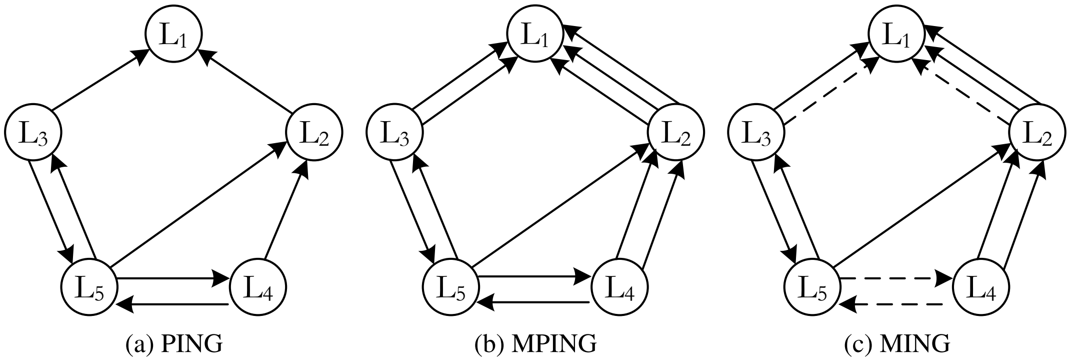


Figure 4.2: The PING, MPING and MING for the example in Figure 4.1

terference range. In a PING, $G_p = (V_p, A_p)$, nodes correspond to communication links. Hereafter, we also use L_i to denote the corresponding node in G_p. There is an arc from L_i to L_j if L_i potentially interferes with L_j. Figure 4.2(a) shows a PING of the example in Figure 4.1. Unfortunately, the above defined PING does not accurately model the devices with multiple radios. For example, if L_i potentially interferes with L_j and both links have two radio pairs, there should be two interference arcs. Therefore, we extend the PING to model multi-radio networks and call the new model *multi-radio potential interference graph* (MPING). An MPING is a directed multigraph, $G_m = (V_m, A_m)$, where nodes still represent transmission links, arcs represent potential interference between links, and parallel directed arcs may exist between two vertices. There are $\min\{r_i, r_j\}$ arcs from L_i to L_j if $(L_i, L_j) \in A_p$. Let $A_m^-(L_i)$ and $A_m^+(L_i)$ be the set of in-arcs and the set of out-arcs, respectively. The in-arc set $A_m^-(L_i)$ of L_i is the set of arcs going into L_i and the out-arc set $A_m^+(L_i)$ of L_i is the set of arcs going from L_i. The in-arcs in $A_m^-(L_i)$ are called the *potential interference arcs* of L_i. Let $N_m^-(L_i)$ be the set of in-neighbors and $N_m^+(L_i)$ be the set of out-neighbors in the MPING G_m. The in-neighbor set $N_m^-(L_i)$ of L_i is the set of vertices, which are the tails of in-arcs and the out-neighbor set $N_m^+(L_i)$ of L_i is the set of vertices, which are the heads of out-arcs. The MPING corresponding to Figure 4.1 is shown in Figure 4.2(b).

We formulate the channel allocation problem in non-cooperative MR-MC wireless networks as a game, called *ChAlloc*. In this game, each transmission link is a player, whose strategy space is the set of all the possible channel allocations on its radios. We assume that players are selfish, rational and honest. We leave the case where players can cheat for our future work. Throughout the rest of this work, we will use link and player interchangeably. The channel allocation of L_i is defined to be a vector $s_i = (s_{i1}, s_{i2}, \ldots, s_{ih})$, where $s_{ik} = 1$ if L_i assigns channel c_k to one radio pair and $s_{ik} = 0$ otherwise. To sufficiently utilize the channel resource, we require that $\sum_{k=1}^{h} s_{ik} = r_i$, which is also proved to be optimal for each player for the single-collision domain case [42]. The strategy profile s is then an $n \times h$ matrix defined by all the players' strategies, $s = (s_1, s_2, \ldots, s_n)^T$.

Although previous works in the literature [42, 49, 149, 150] have used achievable data rate as the utility function, they assume that all the links are in a single collision domain. Because of the hidden terminal problem, it is unlikely to have a closed-form expression to calculate the achievable data rate for each player in the network with multiple collision domains. This difficulty has also been discussed in [42, 150]. An alternative is to use the interference as a performance metric. As shown in [166, Eq.(4)], the data rate is approximately a linear function of the interference that the link can overhear. The use of the interference as a performance metric can also be found in [114, 135, 136].

Given a strategy profile s, we say a communication radio pair *interferes* with L_i if this radio pair belongs to a link interfering with L_i and has been tuned to a channel that is also allocated by L_i. We define the *interference number* of L_i, denoted by $I_i(s)$, to be the number of communication radio pairs interfering with L_i. Mathematically, we have

$$I_i(s) = \sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j,$$

where the symbol $\cdot$ is the dot product between two vectors. Note that $I_i(s) \leq |A_m^-(L_i)|$

for all $L_i \in \mathscr{L}$. When s is given, we can construct the *multi-radio interference graph* (MING), $G_m(s) = (V_m(s), A_m(s))$, from the MPING by removing corresponding potential interference arcs. The number of arcs from L_i to L_j is equal to $s_i \cdot s_j$.

In this work, we define the utility function of a player to be a function of its interference number. More specifically, the utility function $u_i(s)$ of player L_i is defined as

$$u_i(s) = |A_m^-(L_i)| - I_i(s). \tag{4.1}$$

In other words, the objective of L_i is to remove as many of the potential interference arcs as possible from $N_m^-(L_i)$ by allocating channels to its radios. When the network is given, $|A_m^-(L_i)|$ is a constant. Hence maximizing (4.1) can achieve the goal of minimizing $I_i(s)$, which is the interference suffered by L_i under the strategy profile s.

Intuitively, the system performance function is defined as

$$U(s) = |A_m| - \sum_{L_i \in \mathscr{L}} I_i(s), \tag{4.2}$$

which is the total potential interference removed from the MPING under allocation profile s. Likewise, $|A_m|$ is a constant, hence maximizing (4.2) can achieve the goal of minimizing $\sum_{L_i \in \mathscr{L}} I_i(s)$, which is the overall network interference.

Use the example in Figure 4.1 for illustration. The numbers in the parentheses associated to each link represent the allocated channels. The corresponding channel allocation vectors are $s_1 = (1,1,0,1)$, $s_2 = (0,1,1,1)$, $s_3 = (0,1,1,0)$, $s_4 = (0,1,0,1)$, and $s_5 = (0,0,1,0)$. The interference graph under s is shown in Figure 4.2(c). Under this strategy profile, we have $u_1(s) = 2$, $u_2(s) = 0$, $u_3(s) = 0$, $u_4(s) = 1$ and $u_5(s) = 1$. The system performance is $U(s) = 4$.

4.4 ChAlloc Game

4.4.1 Possible Oscillation

We first show that players might not converge to any stable status, i.e. NE, according to the current defined utility function. Consider the network illustrated in Figure4.3(a). Obviously, we have the MPING as shown in Figure4.3(b). Assume that each link is equipped only one radio pair and there are two channels $\{c_1, c_2\}$ available. Due to the special topology and the dependency relation, we have the following conclusions.

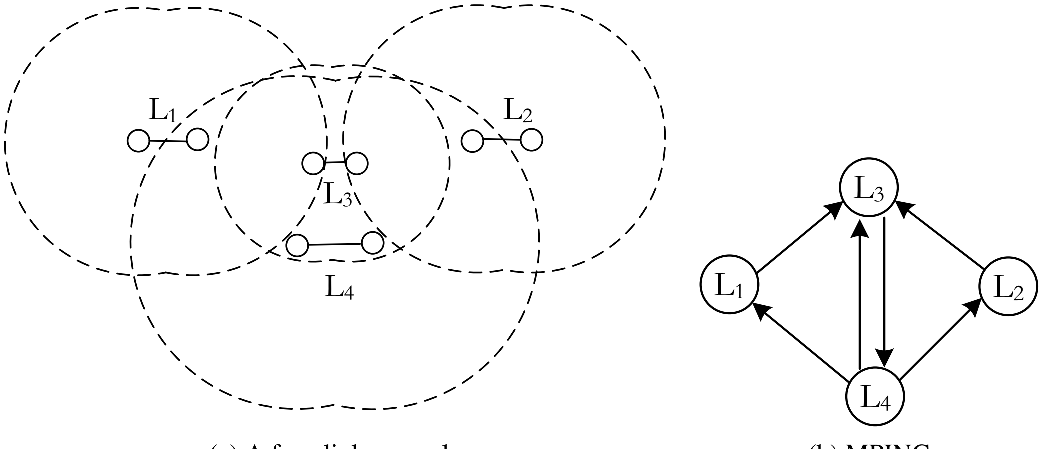
- If L_4 uses c_1, both L_1 and L_2 will use c_2.
- If both L_1 and L_2 use c_2, L_3 will use c_1.
- If L_3 uses c_1, L_4 will use c_2.
- If L_4 uses c_2, both L_1 and L_2 will use c_1.
- ...

This process turns into an infinite loop.

This possible oscillation is definitely undesirable for two reasons: 1) The channel switching delays can be in the order of milliseconds [18], an order of magnitude higher than typical packet transmission time (in microseconds). 2) It can introduce a significant amount of communication overhead, as two devices need to coordinate to switch channels.

4.4.2 Game Analysis

As we have discussed above, the ChAlloc game can run into an oscillation problem, which is undesirable from the system's perspective. In order to induce players to converge to an NE, we design a charging scheme to influence the players.



(a) A four-link example

(b) MPING

Figure 4.3: Example where players oscillate forever

Charging Scheme Design: Similar approaches have also been used in [149, 150]. Their charging functions are designed based on the globally optimal channel allocation. Essentially, players who deviate from the optimal channel allocation will be punished according to the charging function. Unfortunately, it has been proved that the optimization problem of maximizing the system performance function (4.2) is NP-hard [135]. Different variations of the channel allocation problem have also been shown to be NP-hard [1, 91, 114, 117]. Therefore, we focus on designing a charging scheme, which can make the ChAlloc game converge to an NE and achieve guaranteed system performance.

As in [149, 150, 157], we assume that there exists a virtual currency in the system. Each player needs to pay certain amount of virtual money to the system administrator based on the strategy profile s. We define the charge p_i of player L_i as

$$p_i(s) = \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j, \tag{4.3}$$

which is the total interference player L_i imposes on the others. The charge can be considered to be the fee for accessing the channels. We then redefine the utility function for each player $L_i \in \mathscr{L}$ as

$$u_i(s) = \left|A_m^-(L_i)\right| - I_i(s) - p_i(s), \tag{4.4}$$

which is equal to the original utility minus its payment to the system administrator.

Existence of Nash Equilibria: Having defined a new utility function for the player, we next prove the existence of Nash Equilibria with the help of the concept of *potential game* [101].

Definition 4.1. *[Potential Game] A function $\Phi : \Pi \mapsto Z^*$ is an* exact potential function *for a game if the change of any player's utility can be exactly expressed in the function. Formally, Φ should satisfy*

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}),$$

for all s_{-i} and $s_i, s_i' \in \Pi_i$. A game is called a potential game *if it admits an exact potential function.* □

A nice property of being a potential game is that if Φ is bounded, we can prove that the game possesses an NE and any *improvement path* leads to an NE. An improvement path is a sequence of strategy profiles, each of which (except the first one) is formed from the previous one by changing a unique player's strategy to improve the player's utility. Therefore we first prove that the ChAlloc game is a potential game and then prove that its corresponding potential function is bounded.

Lemma 4.1. *The ChAlloc game is a potential game.* □

Proof. We prove this lemma by constructing an exact potential function Φ. Define Φ as

$$\Phi(s) = \frac{1}{2} \sum_{L_i \in \mathscr{L}} u_i(s).$$

We next prove that $\Phi(s_i, s_{-i}) - \Phi(s'_i, s_{-i}) = u_i(s_i, s_{-i}) - u_i(s'_i, s_{-i})$ for all s_{-i} and $s_i, s'_i \in \Pi_i$. First, we have

$$\begin{aligned}\Phi(s) &= \frac{1}{2}\sum_{L_i \in \mathscr{L}} \left(\left|A_m^-(L_i)\right| - \sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j - \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j \right) \\ &= \frac{|A_m|}{2} - \frac{1}{2}\sum_{L_i \in \mathscr{L}} \left(\sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j \right), \end{aligned} \tag{4.5}$$

where the second equality follows from the fact that $|A_m| = \sum_{L_i \in \mathscr{L}} |A_m^-(L_i)|$. We then have

$$\begin{aligned} & \Phi(s_i, s_{-i}) - \Phi(s'_i, s_{-i}) \\ = \; & \frac{1}{2}\left(\sum_{L_j \in N_m^-(L_i)} s_j \cdot s'_i - \sum_{L_j \in N_m^-(L_i)} s_j \cdot s_i + \sum_{L_j \in N_m^+(L_i)} s_j \cdot s'_i - \sum_{L_j \in N_m^+(L_i)} s_j \cdot s_i \right. \\ & \left. + \sum_{L_j \in N_m^-(L_i)} s'_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s'_i \cdot s_j - \left(\sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j \right) \right) \\ = \; & \left(\sum_{L_j \in N_m^-(L_i)} s'_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s'_i \cdot s_j - \left(\sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j \right) \right) \\ = \; & u_i(s_i, s_{-i}) - u_i(s'_i, s_{-i}), \end{aligned}$$

where the first equality follows from the fact that the change of player L_i's strategy only affects players in $N_m^-(L_i)$ and $N_m^+(L_i)$, and the last equality follows from

$$\begin{aligned} & u_i(s_i, s_{-i}) - u_i(s'_i, s_{-i}) \\ = \; & \sum_{L_j \in N_m^-(L_i)} s'_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s'_i \cdot s_j - \left(\sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j + \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j \right). \end{aligned}$$

We have proved that $\Phi(s)$ is an exact potential function (Definition 4.1) of the ChAlloc game. Hence the ChAlloc game is a potential game. □

The bound of $\Phi(s)$ is given in the following lemma.

Lemma 4.2. *For any $s \in \Pi$, $\Phi(s)$ is bounded by $O(\bar{r}n^2)$.* □

Proof. By (4.5), we have

$$\Phi(s) \leq \frac{|A_m|}{2} \leq \frac{\sum_{L_i \in \mathscr{L}} r_i(n-1)}{2} \leq \frac{1}{2}\bar{r}n^2.$$

This completes the proof. □

Now we give the main theorem.

Theorem 4.1. *The ChAlloc game possesses an NE.* □

Proof. Combining Lemma 4.1 and Lemma 4.2, this theorem directly follows from Corollary 2.2 in [101], which states that every finite potential game possesses a Nash Equilibrium. □

Price of Anarchy: Although we have proved that there exist Nash Equilibria in the ChAlloc game, we know that NE is usually not socially efficient in the sense that the system performance in an NE is not optimized. Nevertheless, we prove that the POA of the ChAlloc game is independent of the number of players involved in the game and is lower bounded by a constant when the number of channels and the number of radios equipped on devices are fixed.

Theorem 4.2. *In the ChAlloc game, POA $\geq \left(1 - \frac{\bar{r}}{h}\right)$. Recall that $\bar{r}$ is the maximum number of radios equipped on wireless devices and h is the number of available channels.* □

Proof. Before proving the POA of the ChAlloc game, we first find a lower bound of the utility of any player in an NE. Let $s^{ne} = (s_1^{ne}, s_2^{ne}, \ldots, s_n^{ne})^T$ be any NE of the ChAlloc game.

Let $s^{opt} = (s_1^{opt}, s_2^{opt}, \ldots, s_n^{opt})^T$ be a social optimum. We have

$$\begin{aligned}
& u_i(s^{ne}) \\
= \ & \left|A_m^-(L_i)\right| - \sum_{L_j \in N_m^-(L_i)} s_i^{ne} \cdot s_j^{ne} - \sum_{L_j \in N_m^+(L_i)} s_i^{ne} \cdot s_j^{ne} \\
\geq \ & \sum_{s_i \in \Pi_i} \left(\left|A_m^-(L_i)\right| - \sum_{L_j \in N_m^-(L_i)} s_i \cdot s_j^{ne} - \sum_{L_j \in N_m^+(L_i)} s_i \cdot s_j^{ne} \right) / |\Pi_i| \qquad (4.6) \\
= \ & \left|A_m^-(L_i)\right| - \frac{r_i}{h} \left(\left|A_m^-(L_i)\right| + \left|A_m^+(L_i)\right| \right) \qquad (4.7) \\
\geq \ & \left|A_m^-(L_i)\right| - \frac{\bar{r}}{h} \left(\left|A_m^-(L_i)\right| + \left|A_m^+(L_i)\right| \right), \qquad (4.8)
\end{aligned}$$

where (4.6) follows from the definition of NE, (4.7) follows from the fact that each arc is counted $\binom{h-1}{r_i-1}$ times and $|\Pi_i| = \binom{h}{r_i}$, next (4.8) follows from $\bar{r} = \max_{L_i \in \mathscr{L}} r_i$.

Then the system performance is

$$\begin{aligned}
& U(s^{ne}) \\
= \ & \sum_{L_i \in \mathscr{L}} \left(u_i(s^{ne}) + \sum_{L_j \in N_m^+(L_i)} s_i^{ne} \cdot s_j^{ne} \right) \qquad (4.9) \\
\geq \ & \sum_{L_i \in \mathscr{L}} \left(\left|A_m^-(L_i)\right| - \frac{\bar{r}}{h} \left(\left|A_m^-(L_i)\right| + \left|A_m^+(L_i)\right| \right) \right) + \sum_{L_i \in \mathscr{L}} \sum_{L_j \in N_m^+(L_i)} s_i^{ne} \cdot s_j^{ne} \qquad (4.10) \\
= \ & |A_m| - \frac{2\bar{r}}{h}|A_m| + \sum_{L_i \in \mathscr{L}} \sum_{L_j \in N_m^+(L_i)} s_i^{ne} \cdot s_j^{ne} \qquad (4.11) \\
= \ & |A_m| - \frac{2\bar{r}}{h}|A_m| + |A_m| - U(s^{ne}), \qquad (4.12)
\end{aligned}$$

where (4.9) follows from (4.2) and (4.4), (4.10) follows from (4.8), and (4.12) follows from the fact that $|A_m| = U(s^{ne}) + \sum_{L_i \in \mathscr{L}} \sum_{L_j \in N_m^+(L_i)} s_i^{ne} \cdot s_j^{ne}$.

Considering the obvious fact that $U(s^{opt}) \leq |A_m|$, we have

$$U(s^{ne}) \geq \left(1 - \frac{\bar{r}}{h}\right) U(s^{opt}). \qquad (4.13)$$

Since (4.13) holds for any NE of the ChAlloc game, it is straightforward to prove that $POA \geq \left(1 - \frac{\bar{r}}{h}\right)$. □

A Localized Algorithm for the ChAlloc Game: Since the ChAlloc game is a potential game (Lemma 4.1), any improvement path leads to an NE [101]. To form an improvement path, we need to require the sequential action of the players. Each player takes its best response strategy upon its turn. Note that $b_i(s_{-i})$ can be computed just based on the strategies of the players in $N_m^-(L_i)$ and $N_m^+(L_i)$. Therefore we can design a localized algorithm for players to find an NE. A localized algorithm needs no information to propagate through the whole network. Thus it is scalable to the network size and robust to the topology change.

Algorithm 4: A Localized Algorithm for L_i

```
Randomly allocate $r_i$ channels as $s_i$;
$W_i \leftarrow i$, $ctr \leftarrow 0$;
while true do
    if $W_i = 0$ then
        Get the current channel allocation;
        $s'_i \leftarrow b_i(s_{-i})$;
        if $s'_i = s_i$ then
            if $ctr = n$ then break; else $ctr \leftarrow ctr + 1$;
        else $s_i \leftarrow s'_i$, $ctr \leftarrow 0$;
        $W_i \leftarrow n$;
    else $W_i \leftarrow W_i - 1$;
end
```

The localized algorithm is illustrated in Algorithm 4. The implementation issue will be discussed later. To avoid the simultaneous change in channel allocations of different players, we let each player L_i have a counter W_i, which is initially set to i. At the beginning of the algorithm, each player L_i randomly picks r_i channels as its initial strategy. In every iteration, L_i checks the value of W_i. If W_i is equal 0, L_i gets the current channel allocations and calculates its best response strategy $b_i(s_{-i})$. If the best response strategy is the same with its current strategy, it increases another counter ctr by 1. Otherwise, it updates its strategy and resets ctr to 0. The value of ctr indicates how many times its current strategy has been the best response strategy consecutively. The use of ctr is to avoid early termination before the ChAlloc game converges to an NE. If the counter W_i has not reached 0, L_i decreases its value by 1.

Lemma 4.3. *The best response strategy $b_i(s_{-i})$ can be computed in $O(h(n+\log h))$ time.* □

Proof. We prove this lemma by giving an algorithm to compute $b_i(s_{-i})$. For each channel $c_k \in \mathscr{C}$, we compute the value of $\sum_{L_j \in N_m^-(L_i)} e_k \cdot s_j + \sum_{L_j \in N_m^+(L_i)} e_k \cdot s_j$, where e_k denotes the vector with a 1 in the kth coordinate and 0's elsewhere. This can be finished in $O(hn)$ time. Sort the channels in a nondecreasing order, which can be finished in $O(h \log h)$ time. Since channels are independent, L_i selects the first r_i channels as $b_i(s_{-i})$. Hence the above algorithm can be finished in time bounded by $O(h(n+\log h))$. □

Theorem 4.3. *For any instance of the ChAlloc game, if all the players follow Algorithm 4, it takes $O(\bar{r}hn^3(n+\log h))$ time to converge to an NE.* □

Proof. According to Lemma 4.1 and Lemma 4.2, every time a player changes its strategy (to one introducing better utility), the potential function $\Phi(s)$ will be increased accordingly and the value of $\Phi(s)$ is bounded by $O(\bar{r}n^2)$. Therefore the number of strategy updates is bounded by $O(\bar{r}n^2)$. Since there will be at least one strategy update in each round, the number of rounds is also bounded by $O(\bar{r}n^2)$. Using Lemma 4.3 and the fact that n players take actions sequentially in each round, we can prove that it takes $O(\bar{r}hn^3(n+\log h))$ time for the ChAlloc game to converge to an NE. □

Implementation Issue: Existing works [65, 75, 130, 132] on distributed or localized channel allocation algorithms all assume that the interference sets are given, but do not discuss how to find them in a distributed manner. We assume that during the channel allocation stage, all the players are using the same channel on one of their radios, which is called the control channel, and using the maximum transmission power to send packets. Because of the assumption that $l_i \leq \frac{R}{\gamma}$, it is guaranteed that all the links in the interference range of link L_i can overhear the packets. The packet to be exchanged during the channel allocation stage is of form (i, s_i). For each player L_i, upon its turn, it sends out the packet. During other time periods, it listens to the control channel and receives packets from others. For the packet received from L_j, L_i computes its distance from L_j according to the received signal strength, puts L_j in $N_m^+(L_i)$ if L_j is within its interference range, and puts L_j in $N_m^-(L_i)$ if it is within L_j's interference range.

4.4.3 Upper Bounds on Optimal Channel Allocation

In this section, we derive efficiently computable upper bounds on the channel allocation problem, which will be used in Section 4.5 to evaluate the system performance of the ChAlloc game. We first formulate the channel allocation problem as an integer linear program (ILP) and then relax the constraints to achieve an upper bound on the optimal solution.

Let $s_{ik} \in \{0,1\}$ denote L_i's allocation on c_k, where $s_{ik} = 1$ if L_i allocates c_k to one of its radios and $s_{ik} = 0$ otherwise. Let $x_{ijk} \in \{0,1\}$ denote the interference from L_i to L_j via c_k. Our ILP can be formulated as follows,

$$\max \quad |A_m| - \sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} \sum_{k=1}^{h} x_{ijk}$$

$$\text{s.t.} \quad \sum_{k=1}^{h} s_{ik} = r_i \quad (L_i \in \mathscr{L}) \tag{4.14}$$

$$x_{ijk} \geq s_{ik} + s_{jk} - 1 \quad ((L_i, L_j) \in A_m, c_k \in \mathscr{C}) \tag{4.15}$$

$$s_{ik} \in \{0,1\} \quad (L_i \in \mathscr{L}, c_k \in \mathscr{C})$$

$$x_{ijk} \in \{0,1\} \quad (L_i, L_j \in \mathscr{L}, L_i \neq L_j, c_k \in \mathscr{C}),$$

where Constraints (4.14) follow our system model, and Constraints (4.15) guarantee that $x_{ijk} = 1$ if and only if both L_i and L_j have one radio tuned to channel c_k.

Unfortunately, solving an ILP is in general NP-hard [51], which means it may take exponential time to find the optimal solution. Hence we relax the above ILP to an LP by allowing s_{ik} and x_{ijk} to be real values between 0 and 1. The LP has been shown to be solvable in polynomial time [78]. As the constraints are relaxed, the LP only gives an *upper bound* on the ILP's optimal solution.

4.5 Performance Evaluation

Experiment Setup: In the simulations, links were randomly distributed in a $1000m \times 1000m$ square. The length of each link was uniformly distributed over $[1, 30]$. The interference range of the node was set to 2 times of the link length. The number of available channels was varied from 5 to 12 with increment of 1. The number of links was varied from 10 to 100 with increment of 10. The number of radio pairs on each link was uniformly selected over $[1, r]$, where $r \in \{2, 3, 4, 5\}$. Note that $\bar{r} = r$ in most cases. For every setting, we randomly generated 100 instances and averaged the results.

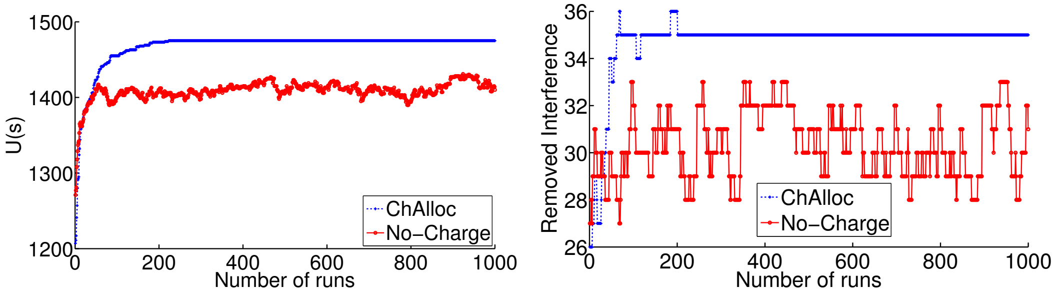
Channel Allocation Algorithms: To evaluate the system performance of the ChAlloc game, we compare the ChAlloc game with other two algorithms listed as below.

- *LP-based Algorithm (LP)*: This algorithm is based on the LP formulation in Section 4.4.3.
- *Random Allocation Algorithm (Rand)*: In this algorithm, each link L_i randomly select r_i channels out of the h channels. To certain extent, *Rand* serves as a lower bound of the system performance in any channel allocation.

Performance Metric: The performance metrics include the *system performance* defined by (4.2), the player's *removed interference* defined by (4.1), and the *convergence speed* defined as the number of rounds before an NE is reached.

Convergence of the ChAlloc Game

We first verify that the ChAlloc game with the charging scheme, denoted by ChAlloc, does converge while the one without the charging scheme, denoted by No-Charge, may oscillate. In this set of simulations, we set n to 50, r to 3, and h to 8. The x-axis represents the number of runs, each of which is an iteration of the while-loop in Algorithm 4. Using runs can show the results in at a more granular level than using rounds. Although the algorithm will terminate when the game reaches an NE, we let it keep running for the sake of comparison. We have the results for 10000 runs, but only show the first 1000 runs due to the space limitation. Figure 4.4(a) shows the system performance of these two difference game settings. As expected, ChAlloc converges to an NE after 222 runs, while No-Charge still oscillates even after 1000 runs. The factors affecting the convergence speed will be investigated later.



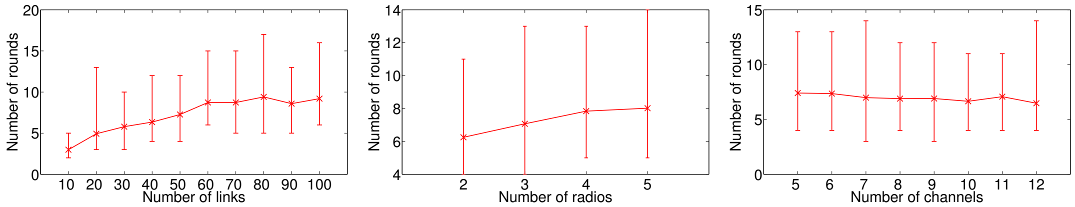
(a) System performance

(b) Player's performance

Figure 4.4: Convergence of the ChAlloc game

Figure 4.4(b) shows the removed interference for a random player (player 44). We observe that the removed interference of the player stays the same after about 200 runs in ChAlloc, but oscillates even after 1000 runs in No-Charge. Note that other players have the similar results.

Convergence Speed



(a) The impact of the number of links

(b) The impact of the number of radios

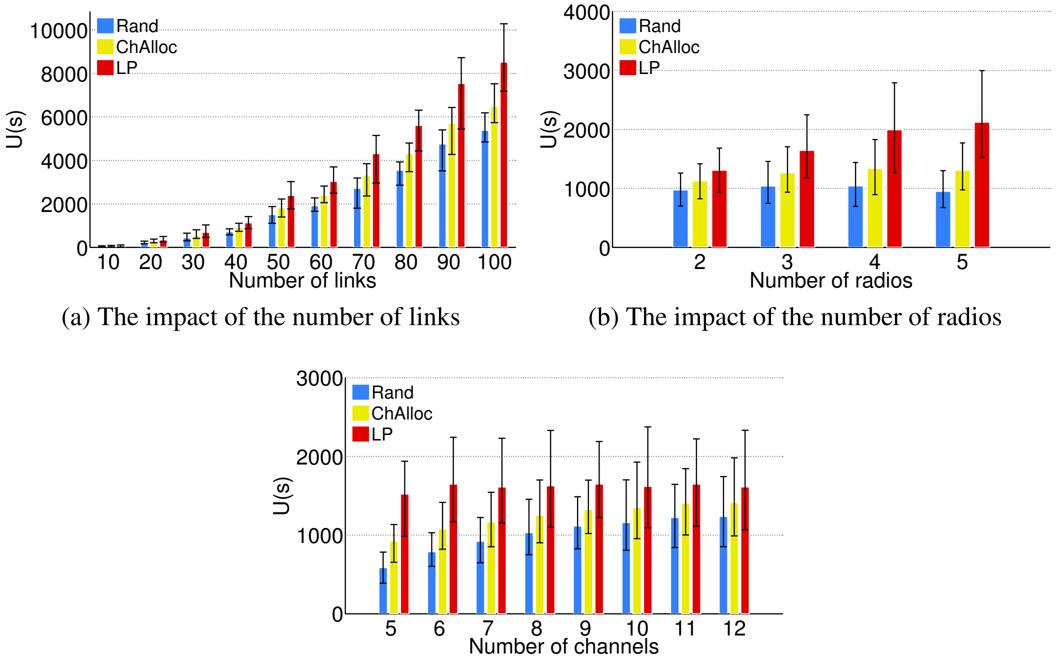
(c) The impact of the number of channels

Figure 4.5: Convergence speed

We next verify our analysis of the convergence speed, measured by the number of rounds. According to Theorem 4.3, the theoretical value is $O(\bar{r}n^2)$. We set h to 8 and r to 3 in Figure 4.5(a). We set n to 50 and r to 3 in Figure 4.5(b). We set n to 50 and h to 8 in Figure 4.5(c). Figure 4.5(a) and Figure 4.5(b) show the impact of n and the impact of r on the convergence speed, respectively. We observe that the convergence speed is much faster than the theoretical speed, and that all the instances can converge within 10 rounds on average. Figure 4.5(c) shows that the convergence speed is almost independent of h, with the varying range of the average being less than 1.

System Performance

We now compare the system performance of the ChAlloc game with those of other algorithms. Figure 4.6 shows all the results. As expected, *LP* has the best performance while *Rand* has the worst. The first observation is that all the results are consistent with our performance analysis in Theorem 4.2. In particular, Figure 4.6(a), Figure 4.6(b) and Figure 4.6(c) show the impact of n, r and h on the system performance, respectively. The results confirm that the system performance of ChAlloc compared to *LP* is independent of n. The less radios or the more channels there are, the closer the performance of ChAlloc



(a) The impact of the number of links

(b) The impact of the number of radios

(c) The impact of the number of channels

Figure 4.6: Comparison on the system performance

is to the performance of *LP*. Another observation is that the gap between the performance of *Rand* and the performance of ChAlloc gets narrower when the number of channels increases or the number of radios decreases. The reason is that when the value of $\frac{\bar{r}}{h}$ decreases, the probability that two interfering links share the same channels decreases as well.

4.6 Conclusion

In this work, we have studied the channel allocation problem in non-cooperative MR-MC networks. Compared with existing works, we removed the single collision domain assumption and considered networks with multiple collision domains. We modeled the problem as a strategic game, called ChAlloc. Via an example, we showed that ChAlloc may result in an oscillation when no exogenous factors exist. To avoid this possible oscillation, we design a charging scheme to influence players' behavior. We then proved that

ChAlloc will converge to an NE in polynomial number of steps. We further proved that the system performance in any NE is guaranteed to be at least $(1-\frac{\bar{r}}{h})$ of that in the optimal solution, where $\bar{r}$ is the maximum number of radios equipped on wireless devices and h is the number of available channels. We also developed a localized algorithm for players to find an NE strategy. Through extensive experiments, we validated our analysis of the possible oscillation and the convergence when there is and is not the charging scheme. Finally, the experiment results also confirmed our proof on the system performance compared to the upper bounds returned by an LP-based algorithm.

Part II

Incentive Mechanisms

Chapter 5

Truthful Auction for Cooperative Communications

Cooperative communication [83] has been shown to have great potential to increase the channel capacity between two wireless devices. It essentially exploits the nature of broadcast and the relaying capability of other nodes to achieve spatial diversity. Yet the applications of cooperative communication technology are rarely seen in reality, even in some scenarios where capacity demand continually grows. The cellular network is one of such examples. The demands for bandwidth-hungry multimedia applications have pushed the system designers to develop more and more innovative network solutions. This fact is mirrored by the exponentially fast growth of 3G/4G wireless networks. Cell phone carrier companies spend billions of dollars on building the infrastructures. As the second largest cell phone carrier company in the U.S., AT&T plans to invest 19 billion dollars on the improvement of 3G networks next year [4]. In contrast to 3G/4G wireless networks, cooperative communication technology does not require extra infrastructure and offers the advantage of flexibility. Although this technology is promising, a main obstacle lying between the potential capability of channel capacity improvement and the wide adoption of cooperative communication is the lack of incentives for the participating wireless nodes to serve as relay nodes. *Why would a cell phone carrier be willing to relay the traffic of another carrier at the cost of its own resource?* One answer to this question is to let the relay node have monetary value in return. Therefore there must be a trade between the wireless node requesting relay service and the one providing such service. Auction is one of the most popular trading form [79], as it allows competitive price discovery and fair and efficient resource allocation.

An auction involving both buyers and sellers is called a *double auction*. To be more

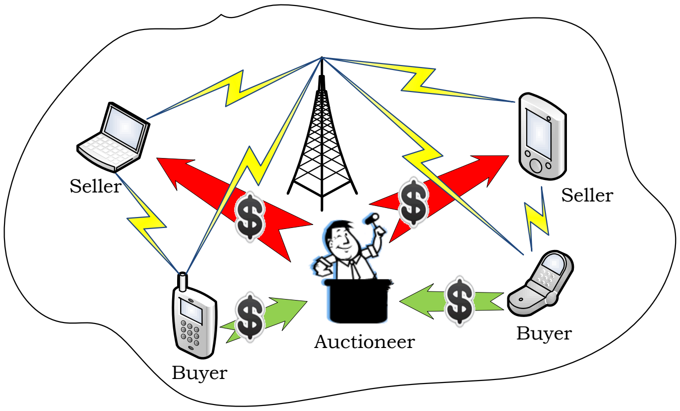


Figure 5.1: Auction for cooperative communications. Relay nodes (sellers) offer prices to sell their relay services. Source nodes (buyers) bid these services for cooperative communication. The base station (auctioneer) determines winners and clearing prices.

specific, the double auction scheme designed in this this falls into the category of *single-round multi-item double auction*. In this auction scheme, as shown in Figure 5.1, n buyers are interested in multiple items from m sellers. However, each buyer needs at most one item at the end of the auction and each seller can sell its item to at most one buyer. The whole auction procedure processes in a single round fashion. Fairly surprisingly, little work has been done in either networking literature or economics literature. Existing double auction schemes [33, 67, 94, 110, 113] cannot be directly applied to the cooperative communication auction. We will give a brief review on the related work in Section 5.2.

The auction design is a crucial aspect of trading market, because the auction scheme not only directly defines the trading rules, but also implicitly defines the behaviors of participating agents. Specifically, truthfulness (also called strategy-proofness) is the most critical property of auction scheme. An auction scheme is truthful if revealing the truthful private value is every participating agent's dominant strategy no matter what strategies other agents are doing. It has been shown both theoretically and practically that an auction could be vulnerable to market manipulation and produce very poor outcomes if this property is not guaranteed [74]. Besides truthfulness, the following properties are also desirable when designing an auction scheme: 1) *Individual Rationality*: each agent participating in the auction can expect a non-negative profit; 2) *Budget Balance*: the auctioneer should finish the auction with no profit loss; 3) *System Efficiency*: the sum of valuations of all agents is optimized, e.g. the total capacity in this work. Unfortunately, the well-known result from [104] shows that no double auction mechanism can achieve truthfulness, budget-balance, and efficiency at the same time, even putting individual rationality to aside. As our goal of this work is to stimulate the participation of wireless nodes in relay services, we focus our design on satisfying truthfulness, individual rationality and budget balance.

5.1 Introduction

In this work, we design a **T**ruthful **A**uction **S**cheme for **C**ooperative communications (TASC). The main contributions of this work are as follows. Firstly, we are the *first* to design a *truthful* auction scheme for cooperative communications, named TASC. TASC implicitly makes it the dominant strategy to bid or ask truthfully for participating agents, thereby eliminating the fear of market manipulation and the overhead of strategizing over others. Secondly, besides being truthful, TASC is also *individually rational* and *budget-balanced*. To the best of our knowledge, this is also the first truthful multi-item double auction scheme even in the economic literature. We hope our study can incite more attention on this type of auction from other researchers. Thirdly, TASC allows the auctioneer to choose any relay assignment algorithm based on its performance requirement. For example, the maximum weighted matching algorithm can be used to maximize the total capacity; Algorithm ORA [129] can be used to maximize the minimum capacity; and the maximum matching algorithm can be used to maximize the number of successful trades. Last but not least, extensive experiments confirm the truthfulness of TASC, and show that TASC achieves all the required properties with limited degradation on the system efficiency.

The remainder of this section is organized as follows. First, we briefly review existing related double auction schemes in the economic literature in Section 5.2. Then we provide an overview of the necessary preliminaries and formulate the problem in Section 5.3. In Section 5.4, we first discuss the challenges of designing a truthful double auction scheme with required economic properties. Next we give the detailed design of our auction scheme TASC. Extensive experiment results are presented in Section 5.5. Finally we conclude this work in Section 5.6.

5.2 Existing Auction Schemes

We categorize the related work into two groups, of which one is the related auction schemes from economics literature and the other is specifically for cooperative communications from networking literature.

Although auction theory has been extensively studied in the economics literature, the existing auction designs cannot fully satisfy the required properties stated in Section 1.1. We summarize the most related works in Table 5.1. In this table, we list the major differences between the existing works and the auction scheme designed in this work. Here the heterogeneity of trading items plays an important role in the auction design. It makes the design more challenging as each buyer has preferences on different items from different sellers. Besides the difference listed in the table, some of the existing schemes are also multi-round auctions [5, 33, 99]. Multi-round auction is unsuitable for the cooperative communications, where timeliness is a necessary requirement and large communication overhead is unfavorable.

There are few studies on the auction design for cooperative communications in networking literature, among which the works in [66, 128, 141] are most related to our work. In [128], Shastry and Adve proposed a pricing-based system to stimulate the cooperation

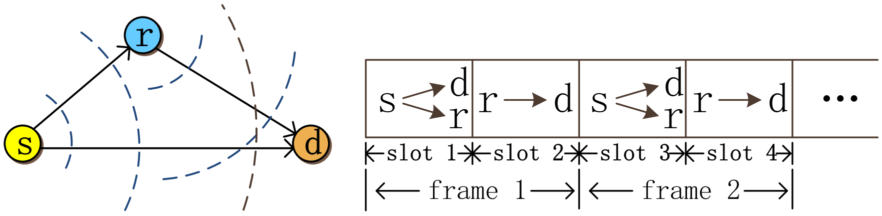
Existing Work	Heter. Item	Double Auction	Truthful
[33]	✓	✗	✓
[113]	✗	✓	✗
[94]	✗	✓	✓
[7]	✗	✓	✓
[110]	✓	✓	✗
[35]	✓	✓	✗
[67]	✗	✓	✓
[5]	✓	✗	–
[99]	✓	✗	–
This work	✓	✓	✓

Table 5.1: Existing auction schemes. "–" means that the corresponding property is unknown.

via payment to the relay node. In [141], Wang *et al.* employed a buyer/seller Stackelberg game, where a single buyer tries to buy services from multiple relays. The buyer announces its selection of relays and the required transmission power, then the relays ask proper prices to maximize their profits. In [66], Huang *et al.* proposed two auction mechanisms, which are essentially repeated games. In each auction mechanism, each user iteratively updates its bid to maximize its own utility function with the knowledge of others' previous bids. With a common drawback, none of the above works considered truthfulness, which is critical to the auction scheme.

5.3 System Model

We use a well-known three-node example in Figure 5.2 to describe the essence of cooperative communications (CC). In this example, s is the source node that transmits information, d is the destination node that receives information and r is the relay node that both receives and transmits information to enhance the communication between the source and the destination. CC proceeds in a frame-by-frame fashion. Each frame is divided into two time slots. The source s transmits data to the destination d in the first time slot. Due to the broadcast nature, relay node r can overhear this transmission. In the second time slot, r forwards the data to d using different techniques depending on different CC modes. There are two CC modes, *Amplify-and-Forward* (AF) and *Decode-and-Forward* (DF) [83]. For details about AF and DF, we refer interested readers to [83]. We use $c_R(s,r,d)$ to denote the achievable capacity under CC and $c_D(s,d)$ to denote the achievable capacity without CC.



(a) (b)

Figure 5.2: Three-node example for CC

In this work, we consider a static ad hoc wireless network consisting of n source-destination pairs $\{s_1,d_1;s_2,d_2;\dots;s_n,d_n\}$ and a set $\mathscr{R}=\{r_1,r_2,\dots,r_m\}$ of m relay nodes. We use $\mathscr{S}=\{s_1,s_2,\dots,s_n\}$ to denote the set of source nodes and $\mathscr{D}=\{d_1,d_2,\dots,d_n\}$ to denote the set of destination nodes. We assume that there is a base station acting as a central control and an auctioneer in the auction scheme, e.g. the base station in the cellular networks, where d_i is the base station for all s_i's, as shown in Figure 5.1.

We design the cooperative communication auction as a *single-round multi-item double auction*. In this auction, source nodes are *buyers*, relay nodes are *sellers*, and the base station is the *auctioneer*. Throughout this work, we may use source node and buyer, relay node and seller, and base station and auctioneer interchangeably. For narration convenience, we call both buyers and sellers *agents* in general. Buyers bid for relay services for cooperative communication, while sellers offer cooperative services at the cost of resources, e.g. energy, and receive monetary payment in return. For each buyer, it has different valuations of the relay nodes as it can achieve different capacities by cooperating with different relay nodes. Let V_i^j be buyer s_i's true valuation of relay service from seller r_j, which describes the true price that s_i is willing to pay for the relay service. Let $\mathbf{V}_i = (V_i^1, V_i^2, \ldots, V_i^m)$ be the true valuation vector of buyer s_i. Obviously, we have $V_i^j > 0$ if $c_R(s_i, r_j, d_i) > c_D(s_i, d_i)$ and $V_i^j = 0$ otherwise. A buyer has no incentive to buy the relay service which cannot provide higher capacity than transmitting directly. Similarly, let C_j be seller r_j's true cost of providing relay service, which is for example related to the energy consumption. A seller does not differentiate among buyers as it uses the same transmission power. We assume that each buyer wants at most one relay to facilitate the cooperative communication. A recent work by Zhao *et al.* [168] shows that it is sufficient for a source node to choose the best relay node even when multiple are available to achieve full diversity. We also assume that each relay node can be shared by at most one source node as it would provide different capacity from what the buyer expects otherwise.

The auction is a sealed-bid auction. Following the terminology in auction theory, we refer the price submitted by a buyer and a seller as *bid* and *ask*, respectively. Each buyer (resp. seller) submits its private bid (resp. ask) to the auctioneer and has no knowledge about others. We assume that both asks and bids are static and will not change during the auction. At the beginning of the auction, each buyer s_i submits a bid vector $\mathbf{B}_i = (B_i^1, B_i^2, \ldots, B_i^m)$, where B_i^j is the bid for seller r_j. $\mathbf{B}_i$ may or may not be the same as its true valuation vector $\mathbf{V}_i$. Each seller r_j submits its ask A_j, which may or may not be its true cost C_j. Let $\mathbb{B} = (\mathbf{B}_1; \mathbf{B}_2; \ldots; \mathbf{B}_n)$ represent the bid matrix consisting of bid vectors submitted by all buyers. Similarly, let $\mathbf{A} = (A_1, A_2, \ldots, A_m)$ represent the set of asks submitted by all sellers. Let $\mathbf{B}_i^{-j} = (B_i^1, \ldots, B_i^{j-1}, B_i^{j+1}, \ldots, B_i^m)$ denote the bid vector of buyer s_i with bid B_i^j removed. Let $\mathbb{B}_{-i} = (\mathbf{B}_1; \ldots; \mathbf{B}_{i-1}; \mathbf{B}_{i+1}; \ldots; \mathbf{B}_n)$ denote the bid matrix with s_i's bid vector $\mathbf{B}_i$ removed. Let $\mathbb{B}|^i\mathbf{B}$ denote the bid matrix with s_i's bid vector changed to $\mathbf{B}$. We have $\mathbf{A}_{-j}$ and $\mathbf{A}|^jA$ defined in similar ways. Given $\mathscr{S}$, $\mathscr{R}$, $\mathscr{D}$, $\mathbb{B}$ and $\mathbf{A}$, the auctioneer decides the winners, including both winning buyers and winning sellers, allocates the relay nodes to the source nodes, and determines the *clearing price* for both winning buyers and winning sellers according to the designed auction scheme. Let $\mathscr{S}_w \subseteq \mathscr{S}$ be the set of winning buyers and $\mathscr{R}_w \subseteq \mathscr{R}$ be the set of winning sellers. Let $\sigma : \{i : s_i \in \mathscr{S}_w\} \to \{j : r_j \in \mathscr{R}_w\}$ be the relay node assignment decided by the auctioneer. Note that $\sigma(\cdot)$ is actually a one-to-one mapping from the indices of winning buyers to those of winning sellers. Therefore, $\sigma^{-1}(j)$ is the index of the source node that relay node r_j is assigned to. Let P_i^b be the price that the winning buyer s_i needs to pay. Let P_j^s be the payment the auctioneer pays the winning seller r_j. Then the *utility* of buyer $s_i \in \mathscr{S}$ is defined as

$$U_i^b = \begin{cases} V_i^{\sigma(i)} - P_i^b & \text{if } s_i \in \mathscr{S}_w, \\ 0 & \text{otherwise.} \end{cases} \tag{5.1}$$

Accordingly, the *utility* of seller $r_j \in \mathscr{R}$ is defined as

$$U_j^s = \begin{cases} P_j^s - C_j & \text{if } r_j \in \mathscr{R}_w, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

5.4 TASC

5.4.1 Challenge of the Auction Design

In this section, we illustrate the challenges of designing a truthful cooperative communication auction. To better understand these challenges, we show the failures of existing double auction schemes when directly applied to the cooperative communication auction. There are two existing double auction schemes, VCG-based double auction and McAfee double auction. We analyze each of them in Section 5.4.1 and Section 5.4.1, respectively.

VCG-based Double Auction

The most well-known auction scheme is the Vickrey-Clarke-Groves (VCG) scheme [23, 60, 138], which can guarantee the truthfulness. In the VCG-based double auction scheme [110], the winners and the assignment between buyers and sellers are determined in a way such that the social welfare $W = \sum_{s_i \in \mathscr{S}_w} (B_i^{\sigma(i)} - A_{\sigma(i)})$ is maximized. Intuitively, this can be achieved by finding the maximum weighted matching in the bipartite graph $G = (\mathscr{S}, \mathscr{R}, \mathscr{E}, \delta)$, where $(s_i, r_j) \in \mathscr{E}$ if $B_i^j > 0$ and $\delta(s_i, r_j) = B_i^j - A_j$ is the weight on edge (s_i, r_j). Let W^* be the optimal value. Let $W^*_{-s_i}$ be the optimal value when buyer s_i is removed from the auction. Let $W^*_{-r_j}$ be the optimal value when seller r_j is removed from the auction. The price each buyer $s_i \in \mathscr{S}_w$ needs to pay is

$$P_i^b = B_i^{\sigma(i)} - (W^* - W^*_{-s_i}). \tag{5.3}$$

The payment each seller $r_j \in \mathscr{R}_w$ receives is

$$P_j^s = A_j + (W^* - W^*_{-r_j}). \tag{5.4}$$

Obviously, both $W^* - W^*_{-s_i}$ and $W^* - W^*_{-r_j}$ are non-negative for all $s_i \in \mathscr{S}_w$ and $r_j \in \mathscr{R}_w$. Therefore, the VCG-based double auction satisfies the individual rationality property. Furthermore, it has been shown that VCG-based auctions are truthful [110]. The proof closely follows standard Vickrey auction proofs. However, a counter example in Figure 5.3 shows that the VCG-based double auction scheme is not budget balanced. In this example, $W^* = 9$, $W^*_{-s_1} = 4$, $W^*_{-s_2} = 7$, $W^*_{-r_1} = 7$, and $W^*_{-r_2} = 3$. Hence $P_1^b = 10 - (9 - 4) = 5$, $P_2^b = 4 - (9 - 7) = 2$, $P_1^s = 2 + (9 - 7) = 4$ and $P_2^s = 3 + (9 - 3) = 9$. The auctioneer finishes the auction with a loss of $(4 + 9) - (5 + 2) = 6$.

McAfee Double Auction

In the McAfee double auction [94], items for auction are homogeneous. Buyers have no preference on these items. Therefore each buyer s_i only submits one bid B_i and each seller r_j offers one ask A_j. The auctioneer starts by sorting the bids in non-increasing order and the asks in non-decreasing order: $B_{i_1} \geq B_{i_2} \geq \ldots B_{i_n}$ and $A_{j_1} \leq A_{j_2} \leq \ldots A_{j_m}$. The

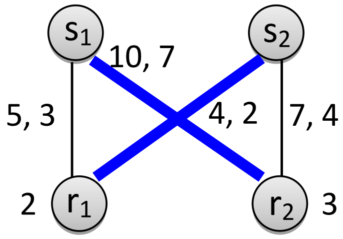


Figure 5.3: Example showing the budget imbalance of the VCG-based double auction. The two numbers associated with link (s_i, r_j) are bid B_i^j and $\delta(s_i, r_j)$. The thick blue lines represent the maximum weighted matching. The number besides each seller is its ask.

auctioneer then finds the largest k such that $B_{i_k} \geq A_{j_k}$ and $B_{i_{k+1}} < A_{j_{k+1}}$. Let $t = \frac{B_{i_{k+1}} + A_{j_{k+1}}}{2}$. The clearing prices are determined as follows:

$$\begin{cases} P^b = P^s = t & \text{if } A_{j_k} \leq t \leq B_{i_k}, \\ P^b = B_{i_k}, P^s = A_{j_k} & \text{otherwise,} \end{cases}$$

where P^b is the price charged to each winning buyer and P^s is the payment that each winning seller receives. Although McAfee double auction satisfies all three properties desired in this work [94], the homogeneity of auction items makes it unsuitable for the cooperative communication auction without further development.

5.4.2 Design of TASC

Now, we present TASC, a truthful and computationally efficient auction scheme for cooperative communications. We start by giving a brief overview of the design rationale. We then describe the detailed design consisting of two main stages. Next we show that TASC satisfies the three properties listed in Section 1.1. Finally, we prove that TASC has a polynomial time complexity of $O(\mathscr{T} + l^2)$, where $\mathscr{T}$ is the time complexity of the relay assignment algorithm and $l = \min\{n, m\}$.

Overview: Although the VCG-based double auction is superficially closest to the auction we aim to design, the imbalance of the budget is unacceptable to the auctioneer. In contrast, TASC is inspired by the design of McAfee double auction. TASC consists of two stages: *Assignment* and *Winner-Determination &Pricing*. To overcome the limitation of McAfee double auction, we apply an assignment algorithm to find a relay assignment in the assignment stage. In the second stage, we apply McAfee double auction to determine the clearing price for both sellers and buyers. The auctioneer charges all winning buyers the same price and pays all winning sellers the same payment.

Design: We now describe the design of TASC in detail. In the assignment stage, we need to design a new relay assignment algorithm or apply the existing relay assignment algorithms with the requirement of being independent of the buyers' bids and the sellers' asks. The relay assignment algorithm's dependency on either the bids or the asks could make the auction vulnerable to market manipulation. Our procedure for the assignment stage is shown in Algorithm 5.

```
Algorithm 5: TASC-Asgmnt($\mathscr{S}$, $\mathscr{R}$, $\mathscr{D}$)
Construct a set $\mathscr{U}$ of vertices corresponding to $\mathscr{S}$;
Construct a set $\mathscr{V}$ of vertices corresponding to $\mathscr{R}$;
$\mathscr{E} \leftarrow \emptyset$;
forall the $s_i \in \mathscr{S}$, $r_j \in \mathscr{R}$ do
    if $c_R(s_i, r_j, d_i) > c_D(s_i, d_i)$ then
        $\mathscr{E} \leftarrow \mathscr{E} \cup \{(s_i, r_j)\}$;
    end
end
$(\mathscr{S}_c, \mathscr{R}_c, \sigma) \leftarrow \Phi(\mathscr{U}, \mathscr{V}, \mathscr{E}, c_R)$;
return $(\mathscr{S}_c, \mathscr{R}_c, \sigma)$;
```

Depending on the specific scenario, the auctioneer can choose different assignment algorithms ($\Phi(\cdot)$) for different purposes. For example, to maximize the total capacity, the maximum weighted matching algorithm can be applied; to maximize the minimum capacity among all source nodes, Algorithm ORA in [129] can be applied; to maximize the number of trades, the maximum matching algorithm fulfills the mission. The return values include the candidate winning buyers $\mathscr{S}_c$, the candidate winning sellers $\mathscr{R}_c$ and the assignment σ.

In the winner-determination & pricing stage, we tightly integrate the winner determination and the pricing operation. With the assignment obtained in the previous stage, we can apply McAfee double auction to determine the winners and the clearing prices. The detailed algorithm is shown in Algorithm 6.

```
Algorithm 6: TASC-WD&Pricing($\mathscr{S}_c, \mathscr{R}_c, \sigma, \mathbb{B}, \mathbf{A}$)
$\mathscr{S}_w \leftarrow \emptyset$, $\mathscr{R}_w \leftarrow \emptyset$;
Sort all the buyers in $\mathscr{S}_c$ to get an ordered list $\mathbb{S} =< s_{i_1}, s_{i_2}, \ldots >$ such that
  $B_{i_1}^{\sigma(i_1)} \geq B_{i_2}^{\sigma(i_2)} \ldots$;
Sort all the sellers in $\mathscr{R}_c$ to get an ordered list $\mathbb{R} =< r_{j_1}, r_{j_2}, \ldots >$ such that
  $A_{j_1} \leq A_{j_2} \ldots$;
Find the largest $k$, such that $B_{i_k}^{\sigma(i_k)} \geq A_{j_k}$;
if $k < 2$ then return $(\mathscr{S}_w, \mathscr{R}_w, 0, 0)$;
$(x, y) \leftarrow (i_k, j_k)$;
// Determine the price and the payment
$P^b \leftarrow B_x^{\sigma(x)}$, $P^s \leftarrow A_y$;
// Sacrifice one buyer and one seller to ensure the truthfulness
$\mathscr{S}_w \leftarrow \mathbb{S}_x \setminus \{s_x\}$, $\mathscr{R}_w \leftarrow \mathbb{R}_y \setminus \{r_y\}$;
// Determine the final winners
for $s_i \in \mathscr{S}_w$ do
    if $r_{\sigma(i)} \notin \mathscr{R}_w$ then $\mathscr{S}_w \leftarrow \mathscr{S}_w \setminus \{s_i\}$;
end
for $r_j \in \mathscr{R}_w$ do
    if $s_{\sigma^{-1}(j)} \notin \mathscr{S}_w$ then $\mathscr{R}_w \leftarrow \mathscr{R}_w \setminus \{r_j\}$;
end
return $(\mathscr{S}_w, \mathscr{R}_w, P^b, P^s)$;
```

For ease of illustration, we introduce more notations and concepts.

- $\mathbb{S}$ denotes an ordered list of buyers sorted in non-increasing order according to their bids on the assigned relay nodes.

- $\mathbb{R}$ denotes an ordered list of sellers sorted in non-decreasing order according to their asks.

- $\mathbb{S}_k$ denotes the sublist of the first k buyers in $\mathbb{S}$.

- $\mathbb{R}_k$ denotes the sublist of the first k sellers in $\mathbb{R}$.

- $\Sigma(\mathbb{S},\mathbb{R})$ denotes the set of matchings induced by $\mathbb{S}$ and $\mathbb{R}$, i.e., $\Sigma(\mathbb{S},\mathbb{R}) = \{(s_i, r_j) : s_i \in \mathbb{S}, r_j \in \mathbb{R}, j = \sigma(i)\}$.

- We call the buyer-seller pair, according to which the auctioneer determines the winners and clearing prices, the *boundary pair*. In McAfee double auction, s_{i_k}–r_{j_k} is such a pair.

- Since the winning buyer and the winning seller are pairwise determined, we call $(s_i, r_{\sigma(i)})$ or $(s_{\sigma^{-1}(j)}, r_j)$ the *winning pair*.

The main algorithm of TASC is illustrated in Algorithm 7.

Algorithm 7: TASC($\mathscr{S}$, $\mathscr{R}$, $\mathscr{D}$, $\mathbb{B}$, $\mathbf{A}$)

```
$(\mathscr{S}_c, \mathscr{R}_c, \sigma) \leftarrow$ TASC-Asgmnt($\mathscr{S}$, $\mathscr{R}$, $\mathscr{D}$);
$(\mathscr{S}_w, \mathscr{R}_w, P^b, P^s) \leftarrow$ TASC-WD&Pricing($\mathscr{S}_c, \mathscr{R}_c, \sigma, \mathbb{B}, \mathbf{A}$);
return $(\mathscr{S}_w, \mathscr{R}_w, \sigma, P^b, P^s)$;
```

Properties of TASC: Having given the detailed design of TASC, we now prove the properties mentioned in Section 1.1.

Theorem 5.1. *TASC is individually rational.* □

Proof. For each winning buyer $s_i \in \mathscr{S}_w \subseteq \mathbb{S}_x$, we know that $B_i^{\sigma(i)} \geq B_x^{\sigma(x)} = P^b$. The same claim also holds for each seller. This completes our proof. □

Theorem 5.2. *TASC is budget-balanced.* □

Proof. Note that we have $|\mathscr{S}_w| = |\mathscr{R}_w|$ based on the assignment assumption. For each winning buyer $s_i \in \mathscr{S}_w$ and its assigned winning seller $r_{\sigma(i)} \in \mathscr{R}_w$, we have $P_i^b = P^b = B_x^{\sigma(x)} \geq A_y = P^s = P_j^s$. Therefore we have $\sum_{s_i \in \mathscr{S}_w} P_i^b - \sum_{r_j \in \mathscr{R}_w} P_j^s = |\mathscr{S}_w|(P^b - P^s) \geq 0$, which completes the proof. □

Theorem 5.3. *TASC is truthful.* □

Before proving Theorem 5.3, we need to prove a series of lemmas. We show that the auction result of each buyer is partially independent of its bid in Lemma 5.1, the winner-determination is bid-monotonic [7] (resp. ask-monotonic) for the buyer in Lemma 5.2 (resp. the seller in Lemma 5.3), the pricing is bid-independent for buyers in Lemma 5.4 (resp. ask-independent for the sellers in Lemma 5.5) and TASC is truthful for buyers in Lemma 5.6 (resp. sellers in Lemma 5.7).

Hereafter, we use tilde to differentiate notations with the same meaning, but different values, e.g., $\tilde{\mathbf{B}}_i$ and $\mathbf{B}_i$ are two different bid vectors of s_i. In addition, we define several comparison operators: $>_j, =_j, <_j$. We say $\tilde{\mathbf{B}}_i >_j \mathbf{B}_i$ if $\tilde{B}_i^j > B_i^j$, $\tilde{\mathbf{B}}_i =_j \mathbf{B}_i$ if $\tilde{B}_i^j = B_i^j$ and $\tilde{\mathbf{B}}_i <_j \mathbf{B}_i$ if $\tilde{B}_i^j < B_i^j$.

Lemma 5.1. *If buyer s_i is assigned relay $r_{\sigma(i)}$ in the assignment stage, then the auction result for s_i is independent of its bids $\boldsymbol{B}_i^{-\sigma(i)}$. In other words, the results of $\Psi = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}|^i \boldsymbol{B}_i, \boldsymbol{A})$ and $\tilde{\Psi} = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}|^i \tilde{\boldsymbol{B}}_i, \boldsymbol{A})$ are the same, if $\boldsymbol{B}_i =_{\sigma(i)} \tilde{\boldsymbol{B}}_i$.* □

Proof. The assignment stage is independent of bids and asks. In the winner-determination and pricing stage (Algorithm 6), it is clear that both the winner determination and the price charged to the buyer are only dependent on the bid the buyer bids on $r_{\sigma(i)}$ and the asks $\mathbf{A}$ of sellers. Therefore, our lemma holds. □

Due to the space limitation, we prove the properties for buyers in the following lemmas and only prove one for sellers. Other properties for sellers can be proved in similar ways as we prove for buyers.

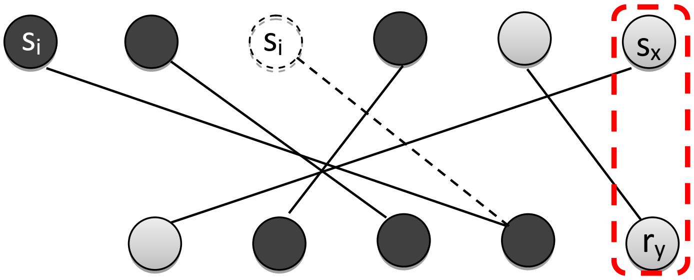


Figure 5.4: Illustration for Lemma 5.2

Lemma 5.2. *If s_i wins $\Psi = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}|^i\boldsymbol{B}_i, \boldsymbol{A})$ by bidding $\boldsymbol{B}_i$, it can also win $\tilde{\Psi} = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}|^i\tilde{\boldsymbol{B}}_i, \boldsymbol{A})$ by bidding $\tilde{\boldsymbol{B}}_i >_{\sigma(i)} \boldsymbol{B}_i$.* □

Proof. In the assignment stage, since the relay assignment algorithm is independent of bids and asks, if s_i is assigned a relay node $r_{\sigma(i)}$ in Ψ, it is assigned the same relay node in $\tilde{\Psi}$ as well. Let p_i and $\tilde{p}_i$ be s_i's positions in $\mathbb{S}$ and $\tilde{\mathbb{S}}$, respectively. An illustration is shown in Figure 5.4. Because $\tilde{B}_i^{\sigma(i)} > B_i^{\sigma(i)}$, the orders in $\mathbb{S}$ and $\tilde{\mathbb{S}}$ after p_i are exactly the same. In addition, we know that the values of k (Line 4) are the same in both $\mathbb{S}$ and $\tilde{\mathbb{S}}$. Therefore, during the winner determination stage, s_x and r_y are still selected as the boundary pair in $\tilde{\Psi}$, which implies that buyer s_i is also a winner in $\tilde{\Psi}$. □

Lemma 5.3. *If r_j wins $\Psi = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}, \boldsymbol{A}|^jA_j)$ by asking A_j, it can also win $\tilde{\Psi} = (\mathscr{S}, \mathscr{R}, \mathscr{D}, \mathbb{B}, \boldsymbol{A}|^j\tilde{A}_j)$ by asking $\tilde{A}_j < A_j$.* □

Proof. Since the relay assignment algorithm is independent of bids and asks, r_j is assigned to the same buyer $s_{\sigma^{-1}(j)}$ in both Ψ and $\tilde{\Psi}$. Let q_i and $\tilde{q}_i$ be r_j's positions in $\mathbb{R}$ and $\tilde{\mathbb{R}}$, respectively. An illustration is shown in Figure 5.5. Because $\tilde{A}_j < A_j$, the orders in $\mathbb{R}$ and

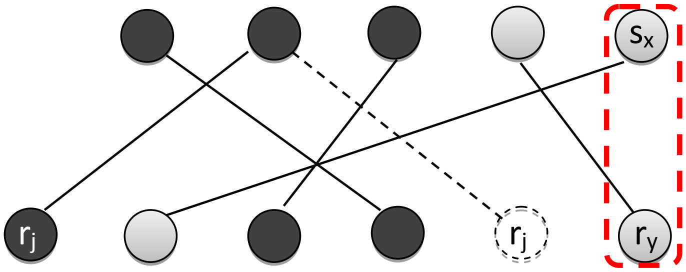


Figure 5.5: Illustration for Lemma 5.3

$\tilde{\mathbb{R}}$ after q_i are exactly the same. In addition, we know that the values of k (Line 4) are the same in both $\mathbb{R}$ and $\tilde{\mathbb{R}}$. Therefore, during the winner determination stage, s_x and r_y are still selected as the boundary pair in $\tilde{\Psi}$, which implies that seller r_i is also a winner in $\tilde{\Psi}$. □

Lemma 5.4. *If s_i wins both $\Psi = (\mathscr{S},\mathscr{R},\mathscr{D},\mathbb{B}|^i\boldsymbol{B}_i,\boldsymbol{A})$ and $\tilde{\Psi} = (\mathscr{S},\mathscr{R},\mathscr{D},\mathbb{B}|^i\tilde{\boldsymbol{B}}_i,\boldsymbol{A})$ by bidding $\boldsymbol{B}_i$ and $\tilde{\boldsymbol{B}}_i$, it is charged the same price, i.e., $P^b = \tilde{P}^b$.* □

Proof. By Lemma 5.1, we know that buyer s_i's bid $\mathbf{B}_i^{-\sigma(i)}$ will not change the auction result nor the charged price if it wins the auction. Hence, without loss of generality, we assume that $\tilde{\mathbf{B}}_i >_{\sigma(i)} \mathbf{B}_i$. As mentioned in the proof of Lemma 5.2, s_x and r_y are selected as the boundary pair in both Ψ and $\tilde{\Psi}$. According to the pricing strategy, we know that, buyer s_i is charged the same price, $P^b = \tilde{P}^b = B_x^{\sigma(x)}$, in both Ψ and $\tilde{\Psi}$. □

Lemma 5.5. *If r_j wins both $\Psi = (\mathscr{S},\mathscr{R},\mathscr{D},\mathbb{B},\boldsymbol{A}|^jA_j)$ and $\tilde{\Psi} = (\mathscr{S},\mathscr{R},\mathscr{D},\mathbb{B},\boldsymbol{A}|^j\tilde{A}_j)$ by asking A_j and $\tilde{A}_j$, it is paid the same payment, i.e., $P^s = \tilde{P}^s$.* □

Lemma 5.6. *TASC is truthful for buyers.* □

Proof. We prove this theorem by showing that no buyer s_i can improve its utility by bidding $\mathbf{B}_i \neq \mathbf{V}_i$, i.e. $\tilde{U}_i^b \leq U_i^b$ for any $\mathbf{B}_i \neq \mathbf{V}_i$, where $\tilde{U}_i^b$ and U_i^b are the utilities of s_i when bidding $\mathbf{B}_i$ and $\mathbf{V}_i$, respectively. We examine all the possible cases one by one as shown in Table 5.2.

Case		Result
$\mathbf{B}_i{=}_{\sigma(i)}\mathbf{V}_i$		$\tilde{U}_i^b = U_i^b$
$\mathbf{B}_i{>}_{\sigma(i)}\mathbf{V}_i$	$\mathbf{B}_i$: w, $\mathbf{V}_i$: w	$\tilde{U}_i^b = U_i^b$
	$\mathbf{B}_i$: w, $\mathbf{V}_i$: l	$\tilde{U}_i^b \leq U_i^b$
	$\mathbf{B}_i$: l, $\mathbf{V}_i$: l	$\tilde{U}_i^b = U_i^b$
$\mathbf{B}_i{<}_{\sigma(i)}\mathbf{V}_i$	$\mathbf{B}_i$: w, $\mathbf{V}_i$: w	$\tilde{U}_i^b = U_i^b$
	$\mathbf{B}_i$: l, $\mathbf{V}_i$: w	$\tilde{U}_i^b \leq U_i^b$
	$\mathbf{B}_i$: l, $\mathbf{V}_i$: l	$\tilde{U}_i^b = U_i^b$

Table 5.2: Proof logic of Lemma 5.6. *w* means it wins and *l* means it loses.

- **C**ase 1: $\mathbf{B}_i{=}_{\sigma(i)}\mathbf{V}_i$

  By Lemma 5.1, we know that buyer s_i is charged the same price P^b by bidding $\mathbf{B}_i$ and $\mathbf{V}_i$ if $\mathbf{B}_i{=}_{\sigma(i)}\mathbf{V}_i$. Therefore, we have $\tilde{U}_i^b = V_i^{\sigma(i)} - P^b = U_i^b$.

- **C**ase 2: $\mathbf{B}_i{>}_{\sigma(i)}\mathbf{V}_i$

  By Lemma 5.2, we know that it is impossible that buyer s_i wins the auction by bidding $\mathbf{V}_i$ but loses by bidding $\mathbf{B}_i$. Hence there are three subcases: 1) s_i wins by bidding both $\mathbf{V}_i$ and $\mathbf{B}_i$; 2) s_i wins by bidding $\mathbf{B}_i$ but loses by bidding $\mathbf{V}_i$; and 3) s_i loses by bidding both $\mathbf{V}_i$ and $\mathbf{B}_i$. For subcase 1), buyer s_i is charged the same price P^b according to Lemma 5.4. Hence, we have $\tilde{U}_i^b = U_i^b = V_i^{\sigma(i)} - P^b$. For subcase 3), we have $\tilde{U}_i^b = U_i^b = 0$ because s_i loses in both auctions. Now we focus on subcase 2). Since s_i wins by bidding $\mathbf{B}_i$ and loses by bidding $\mathbf{V}_i$, we know that $\tilde{P}^b = B_{\tilde{x}}^{\sigma(\tilde{x})} \geq V_i^{\sigma(i)}$, where $\tilde{x}$ is the index of buyer selected in Line 6 in $\tilde{\Psi}$. Hence, we have $\tilde{U}_i^b = V_i^{\sigma(i)} - \tilde{P}^b \leq 0 = U_i^b$.

- **C**ase 3: $\mathbf{B}_i{<}_{\sigma(i)}\mathbf{V}_i$

  By Lemma 5.2, we know that it is impossible that the buyer wins the auction by bidding $\mathbf{B}_i$ but loses by bidding $\mathbf{V}_i$. Hence there are three subcases: 1) s_i wins by bidding both $\mathbf{V}_i$ and $\mathbf{B}_i$; 2) s_i loses by bidding $\mathbf{B}_i$ but wins by bidding $\mathbf{V}_i$; and 3) s_i loses by bidding both $\mathbf{V}_i$ and $\mathbf{B}_i$. For subcases 1) and 3), we can prove that $\tilde{U}_i^b = U_i^b$ following the same analysis as in Case 2. For subcase 2), it is clear that $\tilde{U}_i^b = 0$ and $U_i^b \geq 0$.

We have proved that a buyer cannot improve its utility by submitting a bid vector other than its true valuation vector. This completes the proof. □

Lemma 5.7. *TASC is truthful for sellers.* □

Proof of Theorem 5.3: Lemma 5.6 and Lemma 5.7 together prove that TASC is truthful.

Theorem 5.4. *The time complexity of TASC is $O(\mathscr{T} + l^2)$, where $\mathscr{T}$ is the time complexity of relay assignment algorithm and $l = \min\{n,m\}$.* □

Proof. For the assignment stage, the time complexity depends on the relay assignment algorithm used. For example, the maximum weighted matching algorithm has time complexity of $O((n+m)^2 \log(n+m) + (n+m)nm)$ [26], Algorithm ORA [129] takes $O(nm^2)$ time, and the maximum matching algorithm has time complexity of $O(\sqrt{n+m} \cdot nm)$. We denote the time complexity of the relay assignment algorithm by $\mathscr{T}$ in general. In the winner determination & pricing stage, since the input is the assignment result, we have the number of buyers equal to the number of sellers. Obviously, this number, denoted as l, is not greater than $\min\{n,m\}$. Sorting both sellers and buyers takes $O(l \log l)$ time (Lines 2 and 3). Finding the boundary pair takes $O(l)$ time (Line 4). Determining the final winning pairs takes $O(l^2)$ time (Lines 10 to 17). The time complexity of this stage is thus $O(l^2)$. Therefore the overall time complexity of TASC is $O(\mathscr{T} + l^2)$. □

Theorem 5.5. *The time complexity of TASC is $O(\mathscr{T} + l^2)$, where $\mathscr{T}$ is the time complexity of relay assignment algorithm and $l = \min\{n,m\}$.* □

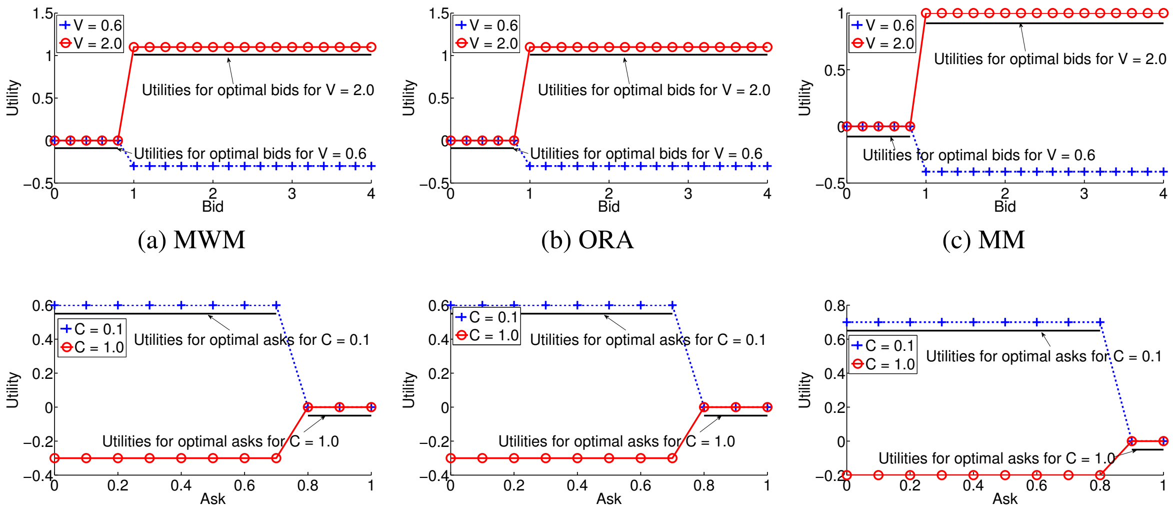
Proof. For the assignment stage, the time complexity depends on the relay assignment algorithm used. For example, the maximum weighted matching algorithm has time complexity of $O((n+m)^2 \log(n+m) + (n+m)nm)$ [26], Algorithm ORA [129] takes $O(nm^2)$ time, and the maximum matching algorithm has time complexity of $O(\sqrt{n+m} \cdot nm)$. We denote the time complexity of the relay assignment algorithm by $\mathscr{T}$ in general. In the winner determination & pricing stage, since the input is the assignment result, we have the number of buyers equal to the number of sellers. Obviously, this number, denoted as l, is not greater than $\min\{n,m\}$. Sorting both sellers and buyers takes $O(l \log l)$ time (Lines 2 and 3). Finding the boundary pair takes $O(l)$ time (Line 4). Determining the final winning pairs takes $O(l^2)$ time (Lines 10 to 17). The time complexity of this stage is thus $O(l^2)$. Therefore the overall time complexity of TASC is $O(\mathscr{T} + l^2)$. □

5.5 Performance Evaluation of TASC

In this section, we present extensive experiments to evaluate the performance of TASC and study the economic impact on the system efficiency.

Experiment Setup: We considered a wireless network where nodes are randomly distributed in a 1000×1000 square. We followed the same parameter settings as in [129]. Let the bandwidth be 22 MHz for all channels. The transmission power is 1 Watt for all wireless nodes. For the transmission model, we assume that the path loss exponent is 4 and the noise is 10^{-10}. For cooperative communication, *DF* mode was used. We fixed the number of buyers (n) at 100 and varied the number of sellers (m) from 50 to 150 with increment of 10. For each setting, we randomly generated 1000 instances and averaged the results. All the tests were run on a Linux PC with 2.00 GHz Intel Pentium CPU and 1.5 GB memory.

For the auction, we assume the buyers' bids are randomly distributed over $(0, V_{max}]$, where V_{max} is set to 4 in most of experiments and varied in the experiments showing the



(a) MWM

(b) ORA

(c) MM

(d) MWM

(e) ORA

(f) MM

Figure 5.6: The utilities of a buyer ((a)-(c)) and a seller ((d)-(f)) in auctions with different assignment algorithms, where $n = m = 100$. In each auction, V is the true valuation of the buyer and C is the true cost of the seller. Two different values are tested for both buyer's true valuation V and seller's true cost C. For each different true valuation (resp. cost), the buyer (resp. the seller) cannot improve its utility by submitting bid (resp. offering ask) different from its true valuation (resp. cost).

impact of bid distribution on the system efficiency. Similarly we assume the sellers' asks are randomly distributed over $(0, 1]$.

The performance metrics in the experiments include agents' utilities, auctioneer's profit, total capacity, number of successful trades and the minimum capacity among all buyers. Throughout all the experiments, we denote the maximum weighted matching algorithm by MWM and the maximum matching algorithm by MM.

Truthfulness of TASC

To verify the truthfulness of TASC, we randomly pick one buyer and one seller, and examine how their utilities change when they bid or ask different values. The results are shown in Figure 5.6(a)-(c) for the buyer and in Figure 5.6(d)-(f) for the seller. We note that

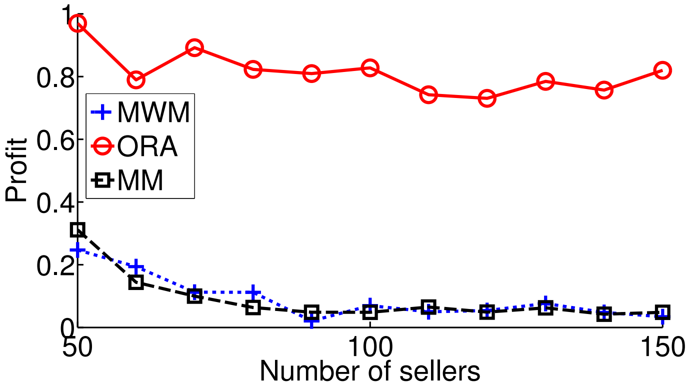


Figure 5.7: Profit of the auctioneer

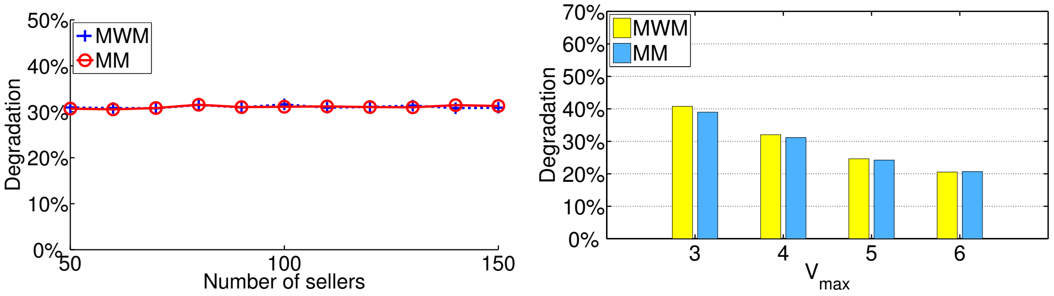
no buyer (resp. seller) can improve its utility by biding (resp. asking) untruthfully.

Impact on Profit

Although making profit is not the goal of designing TASC, it is still necessary to study the impacts of different assignment algorithms on the profit. Figure 5.7 plots the profits of the auctioneer when different relay assignment algorithms are applied. The first observation is that the profits are low for all three different relay assignment algorithms, with the maximum profit less than 2. Due to the way we determine the price and the payment, we have $P^b = P^s$ for many instances. Therefore, the profits are 0 in these instances. Another observation is that the profit decreases with the increase of the number of sellers. This is because as more and more sellers are involved in the auction, the probability that $P^b = P^s$ is becoming higher.

Impact on System Efficiency

Depending on the system performance requirement, the system efficiency could be the total capacity, the number of successful trades, and the minimum capacity among all the participating buyers. Clearly, since the auctioneer cannot allow all the participating agents to be winners, it is inevitable to have degradation over the pure relay assignment, except for the auction with ORA. When the minimum capacity is the system efficiency, TASC will not degrade the performance since the winner decision is made upon the optimal results. To capture the economic impact on the system efficiency, we plot the degradation of TASC over pure relay assignment algorithms in Figure 5.8(a). Surprisingly, the degradation is independent of the number of sellers for both MWM and MM.



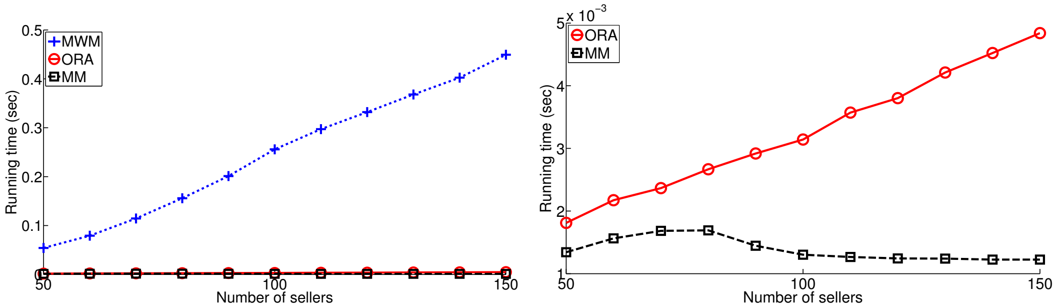
(a) (b)

Figure 5.8: System degradation of TASC over pure relay assignment algorithms

Next we study the impact of the bid distribution on the system efficiency. Figure 5.8(b) illustrates the degradation of TASC with both MWM and MM for different values of V_{max}. We observe that when the maximum bid value V_{max} increases, the degradation of TASC over the pure relay assignment algorithms decreases. In other words, when buyers have higher true valuations on the services from relay nodes, TASC can achieve all the required economic properties without degrading the system efficiency significantly.

Running Time

To confirm our time complexity analysis in Section 5.4.2, we illustrate the running time of TASC with different assignment algorithms in Figure 5.9. We note that the running time increases with the increase of the number of sellers for both MWM and ORA. However, for MM, the running time increases first and then becomes stable after $m = n$. This is because the maximum number of matching is limited by n even when m keeps increasing.



(a)

(b)

Figure 5.9: The running time of TASC, where $n = 100$ and m is varying from 50 to 150. (a) and (b) use the same set of results, while (b) shows the results without MWM for clarity.

5.6 Conclusion

In this work, we have designed TASC, a truthful auction scheme for cooperative communications. To stimulate the participation of wireless devices in relaying traffic for others, TASC allows potential relay nodes to offer prices on their relay services and requires interested source nodes to bid on them. With a careful design, TASC explicitly enforce both sellers and buyers to submit their true valuations, thereby eliminating the fear of market manipulation and the overhead of strategizing over others for them. Meanwhile, TASC also satisfies individual rationality and budget balance properties. In addition, TASC can use any relay assignment algorithm to achieve different system performance requirements. Extensive experiment results confirm our theoretic analysis of TASC and show that TASC can achieve all the required economic properties with limited system efficiency degradation.

Chapter 6

Recruiting an Army of Smartphones for Crowdsourcing

The past few years have witnessed the proliferation of smartphones in people's daily lives. With the advent of 4G networks and more powerful processors, the needs for laptops in particular have begun to fade. Smartphone sales passed PCs for the first time in the final quarter of 2010 [24]. This inflection point occurred much quicker than predicted, which was supposed to be 2012 [95]. According to the International Data Corporation (IDC) Worldwide Quarterly Mobile Phone Tracker, it is estimated that 982 million smartphones will be shipped worldwide in 2015 [68].

Nowadays, smartphones are programmable and equipped with a set of cheap but powerful embedded sensors, such as accelerometer, digital compass, gyroscope, GPS, microphone, and camera. These sensors can collectively monitor a diverse range of human activities and surrounding environment. Smartphones are undoubtedly revolutionizing many sectors of our life, including social networks, environmental monitoring, business, healthcare, and transportation [82].

If all the smartphones on the planet together constitute a mobile phone sensing network, it would be the largest sensing network in the history. One can leverage millions of personal smartphones and a near-pervasive wireless network infrastructure to collect and analyze sensed data far beyond the scale of what was possible before, without the need to deploy thousands of static sensors.

Realizing the great potential of the mobile phone sensing, many researchers have developed numerous applications and systems, such as Sensorly [126] for making cellular/WiFi network coverage maps, Nericell [100] and VTrack [137] for providing traffic information, PIER [102] for calculating personalized environmental impact and exposure, and Ear-Phone [116] for creating noise maps. For more details on mobile phone sensing, we refer interested readers to the survey work [82].

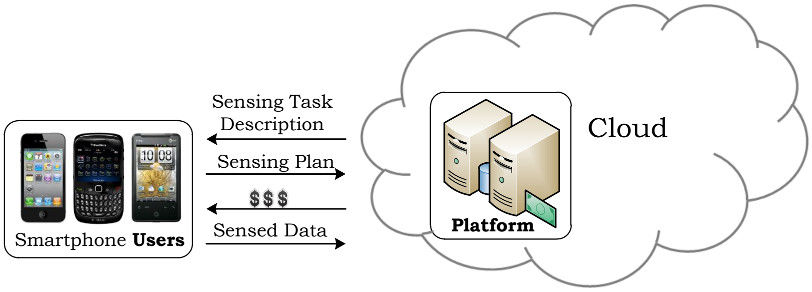


Figure 6.1: Mobile phone sensing system

As shown in Figure 6.1, a mobile phone sensing system consists of a mobile phone sensing *platform*, which resides in the cloud and consists of multiple sensing servers, and many smartphone *users*, which are connected with the platform via the cloud. These smartphone users can act as sensing service providers. The platform recruits smartphone users to provide sensing services.

Although there are many applications and systems on mobile phone sensing [100, 102, 116, 126, 137], most of them are based on voluntary participation. While participating in a mobile phone sensing task, smartphone users consume their own resources such as battery and computing power. In addition, users also expose themselves to potential privacy threats by sharing their sensed data with location tags. Therefore *a user would not be interested in participating in mobile phone sensing, unless it receives a satisfying reward to compensate its resource consumption and potential privacy breach*. Without adequate user participation, it is impossible for the mobile phone sensing applications to achieve good service quality, since sensing services are truly dependent on users' sensed data. While many researchers have developed different mobile phone sensing applications [28, 84], they either do not consider the design of incentive mechanisms or have neglected some critical properties of incentive mechanisms.

6.1 Introduction

To fill the void caused by the lack of incentive mechanisms, we will design several incentive mechanisms to motivate users to participate in mobile phone sensing applications.

We consider two types of incentive mechanisms for a mobile phone sensing system: *platform-centric incentive mechanisms* and *user-centric incentive mechanisms*. In a platform-centric incentive mechanism, the platform has the absolute control over the total payment to users, and users can only tailor their actions to cater for the platform. Whereas in a user-centric incentive mechanism, the roles of the platform and users are reversed. To assure itself of the bottom-line benefit, each user announces a reserve price, the lowest price at which it is willing to sell a service. The platform then selects a subset of users and pay each of them an amount that is no lower than the user's reserve price.

The main contributions of this work include:

- We design incentive mechanisms for mobile phone sensing, a new sensing paradigm that takes advantage of the pervasive smartphones to scale up the sensed data collection and analysis to a level of what was previously impossible.
- We consider two system models from two different perspectives: the platform-centric model where the platform provides a fixed reward to participating users, and the user-centric model where users can have their reserve prices for the sensing service.
- For the platform-centric model, we design an incentive mechanism using a Stackelberg game. We present an efficient algorithm to compute the *unique Stackelberg Equilibrium*, at which the utility of the platform is maximized, and none of the users can improve its utility by unilaterally deviating from its current strategy.

- For the user-centric model, we design an auction-based incentive mechanism, which is computationally efficient, individually-rational, profitable and, more importantly, truthful.

The remainder of this work is organized as follows. In Section 6.2, we briefly review the state of the art on the incentive mechanism design for mobile phone sensing systems. In Section 6.3, we describe the mobile phone sensing system model including both the platform-centric model and the user-centric model. We then study, design and analyze the incentive mechanisms for these two models in Section 6.4.1 and Section 6.4.2, respectively. Next we evaluate the performance of proposed incentive mechanisms through extensive simulations in Section 6.5. Finally, we conclude this work in Section 6.6.

6.2 Related Work

In [118], Reddy *et al.* developed recruitment frameworks to enable the platform to identify well-suited participants for sensing services. However, they focused only on the user selection, not the incentive mechanism design. To the best of our knowledge, there are few research studies on the incentive mechanism design for mobile phone sensing [28, 84]. In [28], Danezis *et al.* developed a sealed-bid second-price auction to motivate user participation. However, the utility of the platform was neglected in the design of the auction. In [84], Lee and Hoh designed and evaluated a reverse auction based dynamic price incentive mechanism, where users can sell their sensed data to the service provider with users' claimed bid prices. However, the authors failed to consider the truthfulness in the design of the mechanism.

The design of the incentive mechanism was also studied for other networking problems, such as spectrum trading [50, 142, 170] and routing [169]. However none of them can be directly applied to mobile phone sensing applications, as they all considered properties specifically pertain to the studied problems.

6.3 System Model

We use Figure 6.1 to aid our description of the mobile phone sensing system. The system consists of a mobile phone sensing *platform*, which resides in the cloud and consists of multiple sensing servers, and many smartphone *users*, which are connected to the platform via the cloud. The platform first publicizes the sensing tasks. Assume that there is a set $\mathscr{U} = \{1, 2, \ldots, n\}$ of smartphone users interested in participating in mobile phone sensing after reading the sensing task description, where $n \geq 2$. A user participating in mobile phone sensing will incur a *cost*, to be elaborated later. Therefore it expects a *payment* in return for its service. Taking cost and return into consideration, each user makes its own sensing plan, which could be the sensing time or the reserve price for selling its sensed data, and submits it to the platform. After collecting the sensing plans from users, the platform computes the payment for each user and sends the payments to the users. The chosen users will conduct the sensing tasks and send the sensed data to the platform. This completes the whole mobile phone sensing process.

The platform is only interested in maximizing its own utility. Since smartphones are owned by different individuals, it is reasonable to assume that users are *selfish* but *rational*. Hence each user only wants to maximize its own utility, and will not participate in mobile phone sensing unless there is sufficient incentive. The focus of this work is on the design of incentive mechanisms that are simple, scalable, and have provably good properties. Other issues in the design and implementation of the whole mobile phone sensing system is out of the scope of this work. Please refer to MAUI [27] for energy saving issues, PRISM [30] for application developing issues, and PEPSI [32] and TP [122] for privacy issues.

We study two models: *platform-centric* and *user-centric*. In the platform-centric model, the sensing plan of an interested user is in the form of its sensing time. A user participating in mobile phone sensing will earn a payment that is no lower than its cost. However, it needs to compete with other users for a fixed total payment. In the user-centric model, each user asks for a price for its service. If selected, the user will receive a payment that is no lower than its asked price. Unlike the platform-centric model, the total payment is not fixed for the user-centric model. Hence, the users have more control over the payment in the user-centric model.

Platform-Centric Model:

In this model, there is only one sensing task. The platform announces a total reward $R > 0$, motivating users to participate in mobile phone sensing, while each user decides its level of participation based on the reward.

The *sensing plan of user i* is represented by t_i, the number of time units it is willing to provide the sensing service. Hence $t_i \geq 0$. By setting $t_i = 0$, user i indicates that it will not participate in mobile phone sensing. The *sensing cost of user i* is $\kappa_i \times t_i$, where $\kappa_i > 0$ is its *unit cost*. Assume that the reward received by user i is proportional to t_i. Then the *utility of user i* is

$$\bar{u}_i = \frac{t_i}{\sum_{j \in \mathscr{U}} t_j} R - t_i \kappa_i, \tag{6.1}$$

i.e., reward minus cost. The *utility of the platform* is

$$\bar{u}_0 = \lambda \log \left(1 + \sum_{i \in \mathscr{U}} \log(1 + t_i) \right) - R, \tag{6.2}$$

where $\lambda > 1$ is a system parameter, the $\log(1 + t_i)$ term reflects the platform's diminishing return on the work of user i, and the outer log term reflects the platform's diminishing return on participating users.

Under this model, the objective of the platform is to decide the optimal value of R so as to maximize (6.2), while each user $i \in \mathscr{U}$ selfishly decides its sensing time t_i to maximize (6.1) for the given value of R. Since no rational user is willing to provide service for a negative utility, user i shall set $t_i = 0$ when $R \leq \kappa_i \sum_{j \neq i \in \mathscr{U}} t_j$.

User-Centric Model:

In this model, the platform announces a set $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_m\}$ of *tasks* for the users to select. Each $\tau_j \in \Gamma$ has a *value* $\nu_j > 0$ *to the platform*. Each user i selects a subset of tasks $\Gamma_i \subseteq \Gamma$ according to its preference. Based on the selected task set, user i also has an *associated cost* c_i, which is private and only known to itself. User i then submits the task-bid pair (Γ_i, b_i) to the platform, where b_i, called user i's *bid*, is the *reserve price* user i wants to sell the service for. Upon receiving the task-bid pairs from all the users, the platform selects a subset $\mathscr{S}$ of users as *winners* and determines the *payment* p_i for each winning user i. The *utility of user i* is

$$\tilde{u}_i = \begin{cases} p_i - c_i, & \text{if } i \in \mathscr{S}, \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

The *utility of the platform* is

$$\tilde{u}_0 = v(\mathscr{S}) - \sum_{i \in \mathscr{S}} p_i, \tag{6.4}$$

where $v(\mathscr{S}) = \sum_{\tau_j \in \cup_{i \in \mathscr{S}} \Gamma_i} \nu_j$.

Our objective for the user-centric model is to design an incentive mechanism satisfying the three desirable properties in Section 1.1. Note that in this specific incentive mechanism design, we also consider **Profitability**. The platform should not incur a deficit. In other words, the value brought by the winners should be at least as large as the total payment paid to the winners.

6.4 Incentive Mechanism Design

6.4.1 Design for the Platform-Centric Model

We model the platform-centric incentive mechanism as a *Stackelberg game* [46], which we call the MSensing game. There are two stages in this mechanism: In the first stage, the platform announces its reward R; in the second stage, each user strategizes its sensing time to maximize its own utility. Therefore the platform is the *leader* and the users are the *followers* in this Stackelberg game. Meanwhile, both the platform and the users are *players*. The *strategy of the platform* is its reward R. The *strategy of user i* is its working time t_i. Let $t = (t_1, t_2, \ldots, t_n)$ denote the *strategy profile* consisting of all users' strategies. Let t_{-i} denote the strategy profile excluding t_i. As a notational convention, we write $t = (t_i, t_{-i})$.

Note that the second stage of the MSensing game itself can be considered a non-cooperative game, which we call the Sensing Time Determination (STD) game. Given the MSensing game formulation, we are interested in answering the following questions:

Q1: For a given reward R, is there a set of stable strategies in the STD game such that no user has anything to gain by unilaterally changing its current strategy?

Q2: If the answer to Q1 is yes, is the stable strategy set unique? When it is unique, users will be guaranteed to select the strategies in the same stable strategy set.

Q3: How can the platform select the value of R to maximize its utility in (6.2)?

The stable strategy set in Q1 corresponds to the concept of *Nash Equilibrium* (NE) in game theory [46].

Definition 6.1 (Nash Equilibrium)**.** *A set of stra- tegies* $(t_1^{ne}, t_2^{ne}, \ldots, t_n^{ne})$ *is a Nash Equilib-*

rium of the STD game if for any user i,

$$\bar{u}_i(t_i^{ne}, t_{-i}^{ne}) \geq \bar{u}_i(t_i, t_{-i}^{ne}),$$

for any $t_i \geq 0$, where $\bar{u}_i$ is defined (6.1).

The *existence* of an NE is important, since an NE strategy profile is stable (no player has an incentive to make a unilateral change) whereas a non-NE strategy profile is unstable. The *uniqueness* of NE allows the platform to predict the behaviors of the users and thus enables the platform to select the optimal value of R. Therefore the answer to Q3 depends heavily on those to Q1 and Q2. The optimal solution computed in Q3 together with the NE of the STD game constitutes a solution to the MSensing game, called *Stackelberg Equilibrium*.

User Sensing Time Determination:

Based on the definition of NE, every user is playing its best response strategy in an NE. From (6.1), we know that $t_i \leq \frac{R}{\kappa_i}$ because $\bar{u}_i$ will be negative otherwise. To study the best response strategy of user i, we compute the derivatives of $\bar{u}_i$ with respect to t_i:

$$\frac{\partial \bar{u}_i}{\partial t_i} = \frac{-Rt_i}{(\sum_{j\in\mathscr{U}} t_j)^2} + \frac{R}{\sum_{j\in\mathscr{U}} t_j} - \kappa_i, \tag{6.5}$$

$$\frac{\partial^2 \bar{u}_i}{\partial t_i^2} = -\frac{2R\sum_{j\in\mathscr{U}\setminus\{i\}} t_j}{(\sum_{j\in\mathscr{U}} t_j)^3} < 0. \tag{6.6}$$

Since the second-order derivative of $\bar{u}_i$ is negative, the utility $\bar{u}_i$ is a *strictly concave function* in t_i. Therefore given any $R > 0$ and any strategy profile t_{-i} of the other users, the best response strategy $\beta_i(t_{-i})$ of user i is unique, if it exists. If the strategy of all other user $j \neq i$ is $t_j = 0$, then user i does not have a best response strategy, as it can have a utility arbitrarily close to R, by setting t_i to a sufficiently small positive number. Therefore we are only interested in the best response for user i when $\sum_{j\in\mathscr{U}\setminus\{i\}} t_j > 0$. Setting the first derivative of $\bar{u}_i$ to 0, we have

$$\frac{-Rt_i}{(\sum_{j\in\mathscr{U}} t_j)^2} + \frac{R}{\sum_{j\in\mathscr{U}} t_j} - \kappa_i = 0. \tag{6.7}$$

Solving for t_i in (6.7), we obtain

$$t_i = \sqrt{\frac{R\sum_{j\in\mathscr{U}\setminus\{i\}} t_j}{\kappa_i}} - \sum_{j\in\mathscr{U}\setminus\{i\}} t_j. \tag{6.8}$$

If the RHS (right hand side) of (6.8) is positive, is also the best response strategy of user i, due to the concavity of $\bar{u}_i$. If the RHS of (6.8) is less than or equal to 0, then user i does not participate in the mobile sensing by setting $t_i = 0$ (to avoid a deficit). Hence we have

$$\beta_i(t_{-i}) = \begin{cases} 0, & \text{if } R \le \kappa_i \sum_{j\in\mathscr{U}\setminus\{i\}} t_j; \\ \sqrt{\frac{R\sum_{j\in\mathscr{U}\setminus\{i\}} t_j}{\kappa_i}} - \sum\limits_{j\in\mathscr{U}\setminus\{i\}} t_j, & \text{otherwise.} \end{cases} \tag{6.9}$$

These analyses lead to the following algorithm for computing an NE of the SDT game.

Algorithm 8: Computation of the NE

```
Sort users according to their unit costs, $\kappa_1 \le \kappa_2 \le \cdots \le \kappa_n$;
$\mathscr{S} \leftarrow \{1,2\}$, $i \leftarrow 3$;
while $i \le n$ and $\kappa_i < \frac{\kappa_i + \sum_{j\in\mathscr{S}} \kappa_j}{|\mathscr{S}|}$ do
    $\mathscr{S} \leftarrow \mathscr{S} \cup \{i\}$, $i \leftarrow i+1$;
end
foreach $i \in \mathscr{U}$ do
    if $i \in \mathscr{S}$ then $t_i^{ne} = \frac{(|\mathscr{S}|-1)R}{\sum_{j\in\mathscr{S}} \kappa_j}\left(1 - \frac{(|\mathscr{S}|-1)\kappa_i}{\sum_{j\in\mathscr{S}} \kappa_j}\right)$;
    else $t_i^{ne} = 0$;
end
return $t^{ne} = (t_1^{ne}, t_2^{ne}, \ldots, t_n^{ne})$
```

Theorem 6.1. *The strategy profile $t^{ne} = (t_1^{ne}, \ldots, t_n^{ne})$ computed by Algorithm 8 is an NE of the STD game. The time complexity of Algorithm 8 is $O(n \log n)$.*

Proof. We first prove that the strategy profile t^{ne} is an NE. Let $n_0 = |\mathscr{S}|$. We have the following observations based on the algorithm: $1)\kappa_i \geq \frac{\sum_{j\in\mathscr{S}}\kappa_j}{n_0-1}$, for any $i \notin \mathscr{S}$; $2)\sum_{j\in\mathscr{S}} t_j^{ne} = \frac{(n_0-1)R}{\sum_{j\in\mathscr{S}}\kappa_j}$; and $3)\sum_{j\in\mathscr{S}\setminus\{i\}} t_j^{ne} = \frac{(n_0-1)^2 R\kappa_i}{\left(\sum_{j\in\mathscr{S}}\kappa_j\right)^2}$ for any $i \in \mathscr{S}$. We next prove that for any $i \notin \mathscr{S}$, $t_i^{ne} = 0$ is its best response strategy given t_{-i}^{ne}. Since $i \notin \mathscr{S}$, we have $\kappa_i \sum_{j\in\mathscr{U}\setminus\{i\}} t_j^{ne} = \kappa_i \sum_{j\in\mathscr{S}} t_j^{ne}$. Using 1) and 2), we have $\kappa_i \sum_{j\in\mathscr{S}} t_j^{ne} \geq R$. According to (6.9), we know that $\beta_i(t_{-i}^{ne}) = 0$.

We then prove that for any $i \in \mathscr{S}$, t_i^{ne} is its best response strategy given t_{-i}^{ne}. Note that $\kappa_i < \frac{\sum_{j=1}^{i}\kappa_j}{i-1}$ according to Algorithm 8. We then have

$$(n_0-1)\kappa_i = (i-1)\kappa_i + (n_0-i)\kappa_i < \sum_{j=1}^{i}\kappa_j + \sum_{j=i+1}^{n_0}\kappa_j,$$

where $\kappa_i \leq \kappa_j$ for $i+1 \leq j \leq n_0$. Hence we have $\kappa_i < \frac{\sum_{i\in\mathscr{S}}\kappa_i}{n_0-1}$. Furthermore, we have

$$\kappa_i \sum_{j\in\mathscr{U}\setminus\{i\}} t_j^{ne} = \kappa_i \sum_{j\in\mathscr{S}\setminus\{i\}} t_j^{ne} = \kappa_i \frac{(n_0-1)^2 R\kappa_i}{\left(\sum_{j\in\mathscr{S}}\kappa_j\right)^2} < R.$$

According to (6.9),

$$\begin{aligned} \beta_i(t_{-i}^{ne}) &= \sqrt{\frac{R\sum_{j\in\mathscr{U}\setminus\{i\}} t_j^{ne}}{\kappa_i}} - \sum_{j\in\mathscr{U}\setminus\{i\}} t_j^{ne} \\ &= \frac{(n_0-1)R}{\sum_{j\in\mathscr{S}}\kappa_j} - \frac{(n_0-1)^2 R\kappa_i}{\left(\sum_{j\in\mathscr{S}}\kappa_j\right)^2} = t_i^{ne}. \end{aligned}$$

Therefore t^{ne} is an NE of the STD game.

We next analyze the running time of the algorithm. Sorting can be done in $O(n\log n)$ time. The while-loop (Lines 3-5) requires a total time of $O(n)$. The for-loop (Lines 6-9) requires a total time of $O(n)$. Hence the time complexity of Algorithm 8 is $O(n\log n)$. □

The next theorem shows the uniqueness of the NE for the STD game.

Theorem 6.2. *Let $R > 0$ be given. Let $\bar{t} = (\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n)$ be the strategy profile of an NE for the STD game, and let $\bar{\mathscr{S}} = \{i \in \mathscr{U} \,|\, \bar{t}_i > 0\}$. We have*

1) $|\bar{\mathscr{S}}| \geq 2$.
2) $\bar{t}_i = \begin{cases} 0, & \textit{if } i \notin \bar{\mathscr{S}}; \\ \frac{(|\bar{\mathscr{S}}|-1)R}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}\left(1 - \frac{(|\bar{\mathscr{S}}|-1)\kappa_i}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}\right), & \textit{otherwise.} \end{cases}$
3) *If $\kappa_q \leq \max_{j\in\bar{\mathscr{S}}}\{\kappa_j\}$, then $q \in \bar{\mathscr{S}}$.*
4) *Assume that the users are ordered such that $\kappa_1 \leq \kappa_2 \leq \cdots \leq \kappa_n$. Let h be the largest integer in $[2,n]$ such that $\kappa_h < \frac{\sum_{j=1}^{h}\kappa_j}{h-1}$. Then $\bar{\mathscr{S}} = \{1, 2, \ldots, h\}$.*

These statements imply that the STD game has a unique NE, which is the one computed by Algorithm 8.

Proof. We first prove 1). Assume that $|\bar{\mathscr{S}}| = 0$. User 1 can increase its utility from 0 to $\frac{R}{2}$ by unilaterally changing its sensing time from 0 to $\frac{R}{2\kappa_1}$, contradicting the NE assumption. This proves that $|\bar{\mathscr{S}}| \geq 1$. Now assume that $|\bar{\mathscr{S}}| = 1$. This means $\bar{t}_k > 0$ for some $k \in \mathscr{U}$, and $\bar{t}_j = 0$ for all $j \in \mathscr{U} \setminus \{k\}$. According to (6.1) the current utility of user k is $R - \bar{t}_k\kappa_k$. User k can increase its utility by unilaterally changing its sensing time from $\bar{t}_k$ to $\frac{\bar{t}_k}{2}$, again contradicting the NE assumption. Therefore $|\bar{\mathscr{S}}| \geq 2$.

We next prove 2). Let $n_0 = |\bar{\mathscr{S}}|$. Since we already proved that $n_0 \geq 2$, we can use the analysis at the beginning of this section (6.7), with t replaced by $\bar{t}$, and $\mathscr{S}$ replaced by $\bar{\mathscr{S}}$. Considering that $\sum_{j\in\mathscr{U}}\bar{t}_j = \sum_{j\in\bar{\mathscr{S}}}\bar{t}_j$, we have

$$\frac{-R\bar{t}_i}{(\sum_{j\in\bar{\mathscr{S}}}\bar{t}_j)^2} + \frac{R}{\sum_{j\in\bar{\mathscr{S}}}\bar{t}_j} - \kappa_i = 0, \quad i \in \bar{\mathscr{S}}. \tag{6.10}$$

Summing up (6.10) over the users in $\bar{\mathscr{S}}$ leads to $n_0R - R = \sum_{j\in\bar{\mathscr{S}}}\bar{t}_j \cdot \sum_{j\in\bar{\mathscr{S}}}\kappa_j$. Therefore we have

$$\sum_{j\in\bar{\mathscr{S}}}\bar{t}_j=\frac{(n_0-1)R}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}. \tag{6.11}$$

Substituting (6.11) into (6.10) and considering $\bar{t}_j=0$ for any $j\in\mathscr{U}\setminus\bar{\mathscr{S}}$, we obtain the following:

$$\bar{t}_i=\frac{(n_0-1)R}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}\left(1-\frac{(n_0-1)\kappa_i}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}\right) \tag{6.12}$$

for every $i\in\bar{\mathscr{S}}$. This proves 2).

We then prove 3). By definition of $\bar{\mathscr{S}}$, we know that $\bar{t}_i>0$ for every $i\in\bar{\mathscr{S}}$. From (6.12), $\bar{t}_i>0$ implies $\frac{(n_0-1)\kappa_i}{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}<1$. Therefore we have

$$\kappa_i<\frac{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}{|\bar{\mathscr{S}}|-1},\forall i\in\bar{\mathscr{S}}. \tag{6.13}$$

(6.13) implies that

$$\max_{i\in\bar{\mathscr{S}}}\kappa_i<\frac{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}{|\bar{\mathscr{S}}|-1}. \tag{6.14}$$

Assume that $\kappa_q\le\max_{j\in\bar{\mathscr{S}}}\{\kappa_j\}$ but $q\notin\bar{\mathscr{S}}$. Since $q\notin\bar{\mathscr{S}}$, we know that $\bar{t}_q=0$. The first-order derivative of $\bar{u}_q$ with respect to t_q when $t=\bar{t}$ is

$$\frac{R}{\sum_{j\in\bar{\mathscr{S}}}\bar{t}_j}-\kappa_q=\frac{\sum_{j\in\bar{\mathscr{S}}}\kappa_j}{n_0-1}-\kappa_q>\max_{i\in\bar{\mathscr{S}}}\{\kappa_i\}-\kappa_q\ge 0. \tag{6.15}$$

This means that user q can increase its utility by unilaterally increasing its sensing time from $\bar{t}_q$, contradicting the NE assumption of $\bar{t}$. This proves 3).

Finally, we prove 4). Statements 1) and 3) imply that $\bar{\mathscr{S}}=\{1,2,\dots,q\}$ for some integer q in $[2,n]$. From (6.13), we conclude that $q\le h$. Assume that $q<h$. Then we have $\kappa_{q+1}<\frac{\sum_{j=1}^{q+1}\kappa_j}{q}$, which implies $\frac{\sum_{j=1}^{q}\kappa_j}{q-1}-\kappa_{q+1}>0$. Hence the first order derivative of $\bar{u}_{q+1}$ with respect to t_{q+1} when $t=\bar{t}$ is $\frac{\sum_{j=1}^{q}\kappa_j}{q-1}-\kappa_{q+1}>0$. This contradiction proves $q=h$. Hence we have proved 4), as well as the theorem. □

Platform Utility Maximization

According to the above analysis, the platform, which is the leader in the Stackelberg game, knows that there exists a unique NE for the users for any given value of R. Hence the platform can maximize its utility by choosing the optimal R. Substituting (6.12) into (6.2) and considering $t_i = 0$ if $i \notin \mathscr{S}$, we have

$$\bar{u}_0 = \lambda \log\left(1 + \sum_{i \in \mathscr{S}} \log(1 + X_i R)\right) - R, \tag{6.16}$$

where $X_i = \frac{(n_0-1)}{\sum_{j\in\mathscr{S}} \kappa_j}\left(1 - \frac{(n_0-1)\kappa_i}{\sum_{j\in\mathscr{S}} \kappa_j}\right)$.

Theorem 6.3. *There exists a unique Stackelberg Equilibrium (R^*, t^{ne}) in the MSensing game, where R^* is the unique maximizer of the platform utility*

$$\bar{u}_0 = \lambda \log\left(1 + \sum_{i \in \mathscr{S}} \log(1 + X_i R)\right) - R,$$

over $R \in [0, \infty)$, $\mathscr{S}$ and t^{ne} are given by Algorithm 8 with the total reward set to R^*.

Proof. The second order derivative of $\bar{u}_0$ is

$$\frac{\partial^2 \bar{u}_0}{\partial R^2} = -\lambda \frac{\sum_{i\in\mathscr{S}} \frac{X_i^2}{(1+X_iR)^2} Y + \left(\sum_{i\in\mathscr{S}} \frac{X_i}{(1+X_iR)}\right)^2}{Y^2} < 0, \tag{6.17}$$

where $Y = 1 + \sum_{i\in\mathscr{S}} \log(1 + X_i R)$. Therefore the utility $\bar{u}_0$ defined in (6.16) is a strictly concave function of R for $R \in [0, \infty)$. Since the value of $\bar{u}_0$ in (6.16) is 0 for $R = 0$ and goes to $-\infty$ when R goes to ∞, it has a unique maximizer R^* that can be efficiently computed using either bisection or Newton's method [13]. □

6.4.2 Design for the User-Centric Model

Auction theory [80] is the perfect theoretical tool to design incentive mechanisms for the user-centric model. We propose a reverse auction based incentive mechanism for the user-centric model. An *auction* takes as input the *bids* submitted by the users, selects a subset of users as *winners*, and determines the *payment* to each winning user.

Auctions Maximizing Platform Utility

Our first attempt is to design an incentive mechanism maximizing the utility of the platform. Now designing an incentive mechanism becomes an optimization problem, called *User Selection* problem: Given a set $\mathscr{U}$ of users, select a subset $\mathscr{S}$ such that $\tilde{u}_0(\mathscr{S})$ is maximized over all possible subsets. In addition, it is clear that $p_i = b_i$ to maximize $\tilde{u}_0(\mathscr{S})$. The utility $\tilde{u}_0$ then becomes

$$\tilde{u}_0(\mathscr{S}) = v(\mathscr{S}) - \sum_{i \in \mathscr{S}} b_i. \tag{6.18}$$

To make the problem meaningful, we assume that there exists at least one user i such that $\tilde{u}_0(\{i\}) > 0$.

Unfortunately, as the following theorem shows, it is NP-hard to find the optimal solution to the User Selection problem.

Theorem 6.4. *The User Selection problem is NP-hard.*

Proof. We prove the NP-hardness of the optimization problem by giving a polynomial time reduction from the NP-hard *Set Cover* problem:

INSTANCE: A universe $Z = \{z_1, z_2, \ldots, z_m\}$, a family $\mathscr{C} = \{C_1, C_2, \ldots, C_n\}$ of subsets of Z and a positive integer s. QUESTION: Does there exist a subset $\mathscr{C}' \subseteq \mathscr{C}$ of size s, such that every element in Z belongs to at least one member in $\mathscr{C}'$?

We construct a corresponding instance of the **User Selection** problem as follows: Let Γ be the task set corresponding to Z, where there is a task $\tau_j \in \Gamma$ for each $z_j \in Z$. Corresponding to each subset $C_i \in \mathscr{C}$, there is a user $i \in \mathscr{U}$ with task set Γ_i, which consists of the tasks corresponding to the elements in C_i. We set ν_j to n for each task τ_j and c_i to 1 for each user $i \in \mathscr{U}$. We prove that *there exists a solution to the instance of the* Set Cover *problem if and only if there exists a subset $\mathscr{S}$ of users such that $\tilde{u}_0(\mathscr{S}) \geq nm - s$.*

We first prove the forward direction. Let $\mathscr{C}'$ be a solution to the *Set Cover* instance. We can select the corresponding set $\mathscr{S}$ of users as the solution to the mechanism design instance. Clearly, $\tilde{u}_0(\mathscr{S}) = nm - |\mathscr{S}| \geq nm - s$. Next we prove the backward direction. Let $\mathscr{S}$ be a solution to mechanism design instance. We then have $\tilde{u}_0(\mathscr{S}) \geq nm - s$. The only possibility that we have such a value is when the selected user set covers all the tasks, because $s \leq m$. Therefore the corresponding set $\mathscr{C}'$ is a solution to the *Set Cover* instance. □

Since it is unlikely to find the optimal subset of users efficiently, we turn our attention to the development of approximation algorithms. To this end, we take advantage of the *submodularity* of the utility function.

Definition 6.2 (Submodular Function)**.** *Let $\mathscr{X}$ be a finite set. A function $f : 2^{\mathscr{X}} \mapsto \mathbb{R}$ is submodular if*

$$f(\mathscr{A} \cup \{x\}) - f(\mathscr{A}) \geq f(\mathscr{B} \cup \{x\}) - f(\mathscr{B}),$$

for any $\mathscr{A} \subseteq \mathscr{B} \subseteq \mathscr{X}$ and $x \in \mathscr{X} \setminus \mathscr{B}$, where $\mathbb{R}$ is the set of reals.

We now prove the submodularity of the utility $\tilde{u}_0$.

Lemma 6.1. *The utility $\tilde{u}_0$ is submodular.*

Proof. By Definition 6.2, we need to show that

$$\tilde{u}_0(\mathscr{S} \cup \{i\}) - \tilde{u}_0(\mathscr{S}) \geq \tilde{u}_0(\mathscr{T} \cup \{i\}) - \tilde{u}_0(\mathscr{T}),$$

for any $\mathscr{S} \subseteq \mathscr{T} \subseteq \mathscr{U}$ and $i \in \mathscr{U} \setminus \mathscr{T}$. It suffices to show that $v(\mathscr{S} \cup \{i\}) - v(\mathscr{S}) \geq v(\mathscr{T} \cup \{i\}) - v(\mathscr{T})$, since the second term in $\tilde{u}_0$ can be subtracted from both sides. Considering $v(\mathscr{S}) = \sum_{\tau_j \in \cup_{i \in \mathscr{S}} \Gamma_i} v_j$, we have

$$\begin{aligned} v(\mathscr{S} \cup \{i\}) - v(\mathscr{S}) &= \sum_{\tau_j \in \Gamma_i \setminus \cup_{j \in \mathscr{S}} \Gamma_j} v_j && (6.19) \\ &\geq \sum_{\tau_j \in \Gamma_i \setminus \cup_{j \in \mathscr{T}} \Gamma_j} v_j && (6.20) \\ &= v(\mathscr{T} \cup \{i\}) - v(\mathscr{T}). && (6.21) \end{aligned}$$

Therefore $\tilde{u}_0$ is submodular. *As a byproduct, we proved that v is submodular as well.* □

When the objective function is submodular, monotone and non-negative, it is known that a greedy algorithm provides a $(1-1/e)$-approximation [108]. Without monotonicity, Feige *et al.* [41] have also developed constant-factor approximation algorithms. Unfortunately, $\tilde{u}_0$ can be negative.

To circumvent this issue, let $f(\mathscr{S}) = \tilde{u}_0(\mathscr{S}) + \sum_{i \in \mathscr{U}} b_i$. It is clear that $f(\mathscr{S}) \geq 0$ for any $\mathscr{S} \subseteq \mathscr{U}$. Since $\sum_{i \in \mathscr{U}} b_i$ is a constant, $f(\mathscr{S})$ is also submodular. In addition, maximizing $\tilde{u}_0$ is equivalent to maximizing f. Therefore we design an auction mechanism based on the algorithm of [41], called Local Search-Based (LSB) auction, as illustrated in Algorithm 9. The mechanism relies on the local-search technique, which greedily searches for a better solution by adding a new user or deleting an existing user whenever possible. It was proved that, for any given constant $\varepsilon > 0$, the algorithm can find a set of users $\mathscr{S}$ such that $f(\mathscr{S}) \geq (\frac{1}{3} - \frac{\varepsilon}{n}) f(\mathscr{S}^*)$, where $\mathscr{S}^*$ is the optimal solution [41].

How good is the LSB auction? In the following we analyze this mechanism using the four desirable properties described in Section 1.1 as performance metrics.

- *Computational Efficiency*: The running time of the Local Search Algorithm is $O(\frac{1}{\varepsilon} n^3 m \log m)$

Algorithm 9: LSB Auction

```
1  $\mathscr{S} \leftarrow \{i\}$, where $i \leftarrow \arg\max_{i \in \mathscr{U}} f(\{i\})$;
2  while there exists a user $i \in \mathscr{U} \setminus \mathscr{S}$ such that $f(\mathscr{S} \cup \{i\}) > (1+\frac{\varepsilon}{n^2})f(\mathscr{S})$ do
3      $\mathscr{S} \leftarrow \mathscr{S} \cup \{i\}$;
4  end
5  if there exists a user $i \in \mathscr{S}$ such that $f(\mathscr{S} \setminus \{i\}) > (1+\frac{\varepsilon}{n^2})f(\mathscr{S})$ then
6      $\mathscr{S} \leftarrow \mathscr{S} \setminus \{i\}$; go to Line 2;
7  end
8  if $f(\mathscr{U} \setminus \mathscr{S}) > f(\mathscr{S})$ then $\mathscr{S} \leftarrow \mathscr{U} \setminus \mathscr{S}$;
9  foreach $i \in \mathscr{U}$ do
10     if $i \in \mathscr{S}$ then $p_i \leftarrow b_i$;
11     else $p_i \leftarrow 0$;
12 end
13 return $(\mathscr{S}, p)$
```

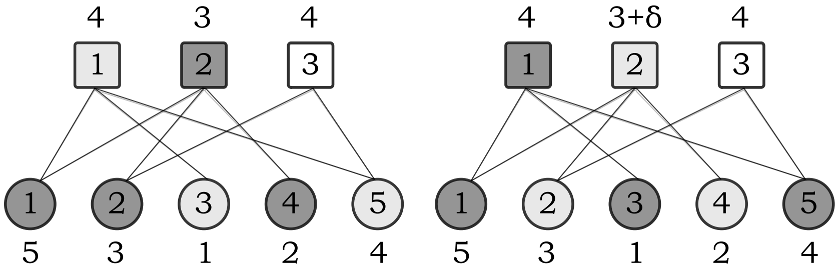
[41], where evaluating the value of f takes $O(m)$ time and $|\mathscr{S}| \leq m$. Hence our mechanism is computationally efficient.

- *Individual Rationality*: The platform pays what the winners bid. Hence our mechanism is individually rational.
- *Profitability*: Due to the assumption that there exists at least one user i such that $\tilde{u}_0(\{i\}) > 0$ and the fact that $f(\mathscr{S})$ strictly increases in each iteration, we guarantee that $\tilde{u}_0(\mathscr{S}) > 0$ at the end of the auction. Hence our mechanism is profitable.
- *Truthfulness*: We use an example in Figure 6.2 to show that the LSB auction is *not* truthful. In this example, $\mathscr{U} = \{1,2,3\}$, $\Gamma = \{\tau_1, \tau_2, \tau_3, \tau_4, \tau_5\}$, $\Gamma_1 = \{\tau_1, \tau_3, \tau_5\}$, $\Gamma_2 = \{\tau_1, \tau_2, \tau_4\}$, $\Gamma_3 = \{\tau_2, \tau_5\}$, $c_1 = 4$, $c_2 = 3$, $c_3 = 4$. Squares represent users, and disks represent tasks. The number above user i denotes its bid b_i. The number below task τ_j denotes its value ν_j. For example, $b_1 = 4$ and $\nu_3 = 1$. We also assume that $\varepsilon = 0.1$.

  We first consider the case where users bid truthfully. Since $f(\{1\}) = \nu(\Gamma_1) - b_1 + \sum_{i=1}^{3} b_i = (5+1+4) - 4 + (4+3+4) = 17$, $f(\{2\}) = 18$ and $f(\{3\}) = 14$, user 2 is first selected. Since $f(\{2,1\}) = \nu(\Gamma_2 \cup \Gamma_1) - (b_2 + b_1) + \sum_{i=1}^{3} b_i = 19 > \left(1 + \frac{0.1}{3^2}\right) f(\{2\}) =$

18.2, user 1 is then selected. The auction terminates here because the current value of f cannot be increased by a factor of $(1+\frac{0.1}{9})$ via either adding a user (that has not been selected) or removing a user (that has been selected). In addition, we have $p_1 = b_1 = 4$ and $p_2 = b_2 = 3$.

We now consider the case where user 2 lies by bidding $3+\delta$, where $1 \leq \delta < 1.77$. Since $f(\{1\}) = 17+\delta$, $f(\{2\}) = 18$ and $f(\{3\}) = 14+\delta$, user 1 is first selected. Since $f(\{1,2\}) = 19 > \left(1+\frac{0.1}{9}\right) f(\{1\})$, user 2 is then selected. The auction terminates here because the current value of f cannot be increased by a factor of $(1+\frac{0.1}{9})$ via either adding a user or removing a user. Note that user 2 increases its payment from 3 to $3+\delta$ by lying about its cost.



(a) Users bid truthfully.

(b) User 2 lies by bidding $3+\delta$, where $1 \leq \delta < 1.77$.

Figure 6.2: Example showing the untruthfulness of the Local Search-Based Auction mechanism, where $\mathscr{U} = \{1,2,3\}$, $\Gamma = \{\tau_1,\tau_2,\tau_3,\tau_4,\tau_5\}$, $\Gamma_1 = \{\tau_1,\tau_3,\tau_5\}$, $\Gamma_2 = \{\tau_1,\tau_2,\tau_4\}$, $\Gamma_3 = \{\tau_2,\tau_5\}$. Squares represent users. Disks represent tasks. The number above user i denotes its bid b_i. The number below task τ_j denotes its value ν_j. We also assume that $\varepsilon = 0.1$.

MSensing Auction

Although the LSB auction mechanism is designed to approximately maximize the platform utility, the failure of guaranteeing truthfulness makes it less attractive. Since our ultimate goal is to design an incentive mechanism that motivates smartphone users to participate in mobile phone sensing while preventing any user from rigging its bid to manipulate the market, we need to settle for a trade off between utility maximization and truthfulness. Our highest priority is to design an incentive mechanism that satisfies all of the four desirable properties, even at the cost of sacrificing the platform utility. One possible direction is to make use of the off-the-shelf results on the *budgeted mechanism* design [20, 131]. The budgeted mechanism design problem is very similar with ours, with the difference that the payment paid to the winners is a constraint instead of a factor in the objective function. To address this issue, we can intuitively plug different values of the budget into the budgeted mechanism and select the one giving the largest utility. However, this can potentially destroy the truthfulness of the incentive mechanism.

Now we present a novel auction mechanism that satisfies all four desirable properties. The design rationale relies on Myerson's well-known characterization [103].

Theorem 6.5. *([131, Theorem 2.1]) An auction mechanism is truthful if and only if:*

- *The selection rule is monotone: If user i wins the auction by bidding b_i, it also wins by bidding $b'_i \leq b_i$;*
- *Each winner is paid the critical value: User i would not win the auction if it bids higher than this value.*

Based on Theorem 6.5, we design our auction mechanism, which is called MSensing auction. Illustrated in Algorithm 10, the MSensing auction mechanism consists of two phases: the *winner selection* phase and the *payment determination* phase.

The winner selection phase follows a greedy approach: Users are essentially sorted according to the difference of their marginal values and bids. Given the selected users $\mathscr{S}$, the marginal value of user i is $v_i(\mathscr{S}) = v(\mathscr{S} \cup \{i\}) - v(\mathscr{S})$. In this sorting the $(i+1)$th user is the user j such that $v_j(\mathscr{S}_i) - b_j$ is maximized over $\mathscr{U} \setminus \mathscr{S}_i$, where $\mathscr{S}_i = \{1, 2, \ldots, i\}$

Algorithm 10: MSensing Auction

```
// Phase 1:  Winner selection
$\mathscr{S} \leftarrow \emptyset$, $i \leftarrow \arg\max_{j \in \mathscr{U}} \left(v_j(\mathscr{S}) - b_j\right)$;
while $b_i < v_i$ and $\mathscr{S} \neq \mathscr{U}$ do
    $\mathscr{S} \leftarrow \mathscr{S} \cup \{i\}$;
    $i \leftarrow \arg\max_{j \in \mathscr{U} \setminus \mathscr{S}} \left(v_j(\mathscr{S}) - b_j\right)$;
end
// Phase 2:  Payment determination
foreach $i \in \mathscr{U}$ do $p_i \leftarrow 0$;
foreach $i \in \mathscr{S}$ do
    $\mathscr{U}' \leftarrow \mathscr{U} \setminus \{i\}$, $\mathscr{T} \leftarrow \emptyset$;
    repeat
        $i_j \leftarrow \arg\max_{j \in \mathscr{U}' \setminus \mathscr{T}} \left(v_j(\mathscr{T}) - b_j\right)$;
        $p_i \leftarrow \max\{p_i, \min\{v_i(\mathscr{T}) - (v_{i_j}(\mathscr{T}) - b_{i_j}), v_i(\mathscr{T})\}\}$;
        $\mathscr{T} \leftarrow \mathscr{T} \cup \{i_j\}$;
    until $b_{i_j} \geq v_{i_j}$ or $\mathscr{T} = \mathscr{U}'$;
    if $b_{i_j} < v_{i_j}$ then $p_i \leftarrow \max\{p_i, v_i(\mathscr{T})\}$;
end
return $(\mathscr{S}, p)$
```

and $\mathscr{S}_0 = \emptyset$. We use v_i instead of $v_i(\mathscr{S}_{i-1})$ to simplify the notation. Considering the submodularity of v, this sorting implies that

$$v_1 - b_1 \geq v_2 - b_2 \geq \cdots \geq v_n - b_n. \tag{6.22}$$

The set of winners are $\mathscr{S}_L = \{1, 2, \ldots, L\}$, where $L \leq n$ is the largest index such that $v_L - b_L > 0$.

In the payment determination phase, we compute the payment p_i for each winner $i \in \mathscr{S}$. To compute the payment for user i, we sort the users in $\mathscr{U} \setminus \{i\}$ similarly,

$$v'_{i_1} - b_{i_1} \geq v'_{i_2} - b_{i_2} \geq \cdots \geq v'_{i_{n-1}} - b_{i_{n-1}}, \tag{6.23}$$

where $v'_{i_j} = v(\mathscr{T}_{j-1} \cup \{i_j\}) - v(\mathscr{T}_{j-1})$ denotes the marginal value of the jth user and $\mathscr{T}_j$ denotes the first j users according to this sorting over $\mathscr{U} \setminus \{i\}$ and $\mathscr{T}_0 = \emptyset$. The marginal value of user i at position j is $v_{i(j)} = v(\mathscr{T}_{j-1} \cup \{i\}) - v(\mathscr{T}_{j-1})$. Let K denote the position of the last user $i_j \in \mathscr{U} \setminus \{i\}$, such that $b_{i_j} < v'_{i_j}$. For each position j in the sorting, we compute the maximum price that user i can bid such that i can be selected instead of user at jth place. We repeat this until the position after the last winner in $\mathscr{U} \setminus \{i\}$. In the end we set the value of p_i to the maximum of these $K+1$ prices.

We will prove the computational efficiency (Lemma 6.2), the individual rationality (Lemma 6.3), the profitability (Lemma 6.4), and the truthfulness (Lemma 6.5) of the MSensing auction in the following.

Lemma 6.2. *MSensing is computationally efficient.*

Proof. Finding the user with maximum marginal value takes $O(nm)$ time, where computing the value of v_i takes $O(m)$ time. Since there are m tasks and each winner should contribute at least one new task to be selected, the number of winners is at most m. Hence, the while-loop (Lines 3–6) thus takes $O(nm^2)$ time. In each iteration of the for-loop (Lines 9–17), a process similar to Lines 3–6 is executed. Hence the running time of the whole auction is dominated by this for-loop, which is bounded by $O(nm^3)$. □

Note that the running time of the MSensing Auction, $O(nm^3)$, is very conservative. In addition, m is much less than n in practice, which makes the running time of the MSensing Auction dominated by n.

Before turning our attention to the proofs of the other three properties, we would like to make some critical observations: 1) $v_{i(j)} \geq v_{i(j+1)}$ for any j due to the submodularity of v; 2) $\mathscr{T}_j = \mathscr{S}_j$ for any $j < i$; 3) $v_{i(i)} = v_i$; and 4) $v'_{i_j} > b_{i_j}$ for $j \leq K$ and $v'_{i_j} \leq b_{i_j}$ for $K+1 \leq j \leq n-1$.

Lemma 6.3. *MSensing is individually rational.*

Proof. Let i_i be user i's replacement which appears in the ith place in the sorting over $\mathscr{U} \setminus \{i\}$. Since user i_i would not be at ith place if i is considered, we have $v_{i(i)} - b_i \geq v'_{i_i} - b_{i_i}$. Hence we have $b_i \leq v_{i(i)} - (v'_{i_i} - b_{i_i})$. Since user i is a winner, we have $b_i \leq v_i = v_{i(i)}$. It follows that $b_i \leq \min\left\{v_{i(i)} - (v'_{i_i} - b_{i_i}), v_{i(i)}\right\} \leq p_i$. If i_i does not exist, it means i is the last winner in $\mathscr{U}$. We then have $b_i \leq v_i(\mathscr{U} \setminus \{i\}) \leq p_i$, according to Line 16. □

Lemma 6.4. *MSensing is profitable.*

Proof. Let L be the last user $j \in \mathscr{U}$ in the sorting (6.22), such that $b_j < v_j$. We then have $\tilde{u}_0 = \sum_{1 \leq i \leq L} v_i - \sum_{1 \leq i \leq L} p_i$. Hence it suffices to prove that $p_i \leq v_i$ for each $1 \leq i \leq L$. Recall that K is the position of the last user $i_j \in \mathscr{U} \setminus \{i\}$ in the sorting (6.23), such that $b_{i_j} < v'_{i_j}$. When $K < n-1$, let r be the position such that

$$r = \arg \max_{1 \leq j \leq K+1} \min\left\{v_{i(j)} - (v'_{i_j} - b_{i_j}), v_{i(j)}\right\}.$$

If $r \leq K$, we have

$$\begin{aligned} p_i &= \min\left\{v_{i(r)} - (v'_{i_r} - b_{i_r}), v_{i(r)}\right\} \\ &= v_{i(r)} - (v'_{i_r} - b_{i_r}) < v_{i(r)} \leq v_i, \end{aligned}$$

where the penultimate inequality is due to the fact that $b_{i_r} < v'_{i_r}$ for $r \leq K$, and the last inequality relies on the fact that $\mathscr{T}_{j-1} = \mathscr{S}_{j-1}$ for $j \leq i$ and the decreasing marginal value property of v. If $r = K+1$, we have

$$p_i = \min\left\{v_{i(r)} - (v'_{i_r} - b_{i_r}), v_{i(r)}\right\} = v_{i(r)} \leq v_i.$$

Similarly, when $K = n-1$, we have

$$p_i \leq v_i(r) \leq v_i,$$

for some $1 \leq r \leq K$. Thus we proved that $p_i \leq v_i$ for each $1 \leq i \leq K$. □

Lemma 6.5. *MSensing is truthful.*

Proof. Based on Theorem 6.5, it suffices to prove that the selection rule of MSensing is monotone and the payment p_i for each i is the critical value. The monotonicity of the selection rule is obvious as bidding a smaller value can not push user i backwards in the sorting.

We next show that p_i is the critical value for i in the sense that bidding higher p_i could prevent i from winning the auction. Note that

$$p_i \quad = \quad \max\left\{\max_{1\leq j\leq K}\left(v_{i(j)}-(v'_{i_j}-b_{i_j})\right), v_{i(K+1)}\right\}.$$

If user i bids $b_i > p_i$, it will be placed after K since $b_i > v_{i(j)} - (v'_{i_j} - b_{i_j})$ implies $v'_{i_j} - b_{i_j} > v_{i(j)} - b_i$. At the $(K+1)$th iteration, user i will not be selected because $b_i > v_{i(K+1)}$. As $K+1$ is the position of the first loser over $\mathscr{U} \setminus \{i\}$ when $K < n-1$ or the last user to check when $K = n-1$, the selection procedure terminates. □

The above four lemmas together prove the following theorem.

Theorem 6.6. *MSensing is computationally efficient, individually rational, profitable, and truthful.* □

6.5 Evaluation of Incentive Mechanisms

To evaluate the performance of our incentive mechanisms, we implemented the incentive mechanism for the platform-centric model, the Local Search-Based auction, denoted by *LSB*, and the MSensing auction, denoted by *MSensing*. In order to reduce the running time of *MSensing*, we applied the *lazy update algorithm* [97]. We denote the *MSensing* auction with the lazy update as *MSensing-L*. The idea of the lazy update algorithm is that we revaluate the marginal value of each user only when necessary. We keep a list of users sorted in a non-increasing order according to the difference between their marginal values and bids. Before searching for the user with the largest difference between its marginal value and bid, we revaluate the marginal value of the first user in the list. If the new value is the same as the old one, it is guaranteed that the first user still provides the largest difference of the marginal value and the bid, due to the submodularity of v. Otherwise, we revaluate all the users in the list and sort them according to the new value.

Performance Metrics: The performance metrics include running time, platform utility, and user utility in general. For the platform-centric incentive mechanism, we also study the number of participating users.

Simulation Setup

We varied the number of users (n) from 100 to 1000 with the increment of 100. For the platform-centric model, we assumed that the cost of each user was uniformly distributed over $[1, \kappa_{max}]$, where κ_{max} was varied from 1 to 10 with the increment of 1. We set λ to 3, 5, and 10. For the user-centric model, we varied the number of tasks (m) from 100 to 500 with the increment of 100. We set ε to 0.01 for *LSB*. We also made the following assumptions. The value of each task is uniformly distributed over $[1,5]$. The number of tasks of each user is uniformly distributed over $[1,10]$. The cost c_i is $\rho|\Gamma_i|$, where ρ is uniformly distributed over $[1,5]$.

All the simulations were run on a Linux machine with 3.2 GHz CPU and 16 GB memory. Each measurement is averaged over 100 instances.

Evaluation of the Platform-Centric Incentive Mechanism

Running Time: We first evaluate the running time of the incentive mechanism and show the results in Figure 6.3. We observe that the running time is almost linear in the number of users and less than 2×10^{-4} seconds for the largest instance of 1000 users. As soon as the users are sorted and $\mathscr{S}$ is computed, all the values can be computed using closed-form expressions, which makes the incentive mechanism very efficient.

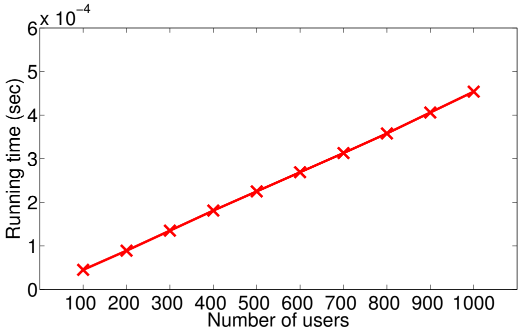


Figure 6.3: Running time

Number of Participating Users: Figure 6.4 shows the impact of κ_{max} on the number of participating users, i.e., $|\mathscr{S}|$, when n is fixed at 1000. We can see that $|\mathscr{S}|$ decreases as the costs of users become diverse. The reason is that according to the while-loop condition, if all users have the same cost, then all of them would satisfy this condition and thus participate. When the costs become diverse, users with larger costs would have higher chances to violate the condition.

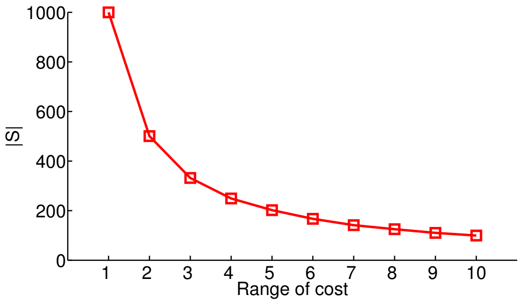
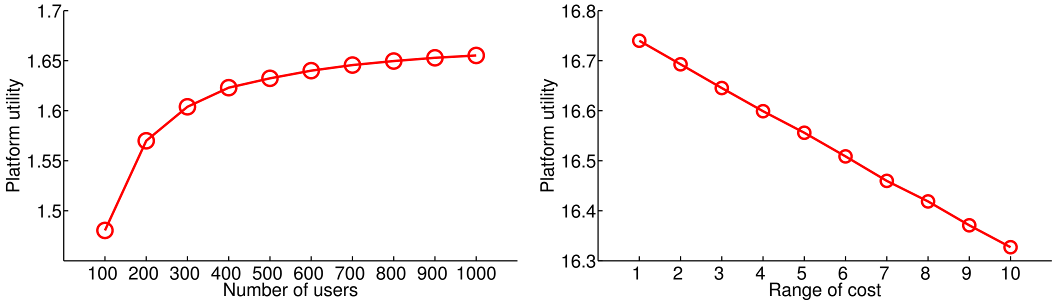


Figure 6.4: Impact of κ_{max} on $|\mathscr{S}|$

Platform Utility: Figure 6.5 shows the impact of n and κ_{max} on the platform utility. We set λ to 3, 5, and 10. In Figure 6.5(a), we fixed $\kappa_{max} = 5$. We observe that the platform utility is almost linear in n and the slope becomes higher as λ is larger. Also, we have higher utility when λ is larger, which is expected. In Figure 6.5(b), we fixed $n = 1000$. With the results in Figure 6.4, it is expected that the platform utility decreases as the costs of users become more diverse.



(a) Impact of n on $\bar{u}_0$ (b) Impact of κ_{max} on $\bar{u}_0$

Figure 6.5: Platform utility

User Utility: We randomly picked a user (ID = 31) and plot its utility in Figure 6.6. We observe that as more and more users are interested in mobile phone sensing, the utility of the user decreases since more competitions are involved.

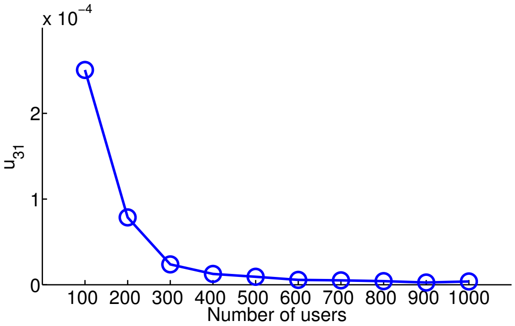
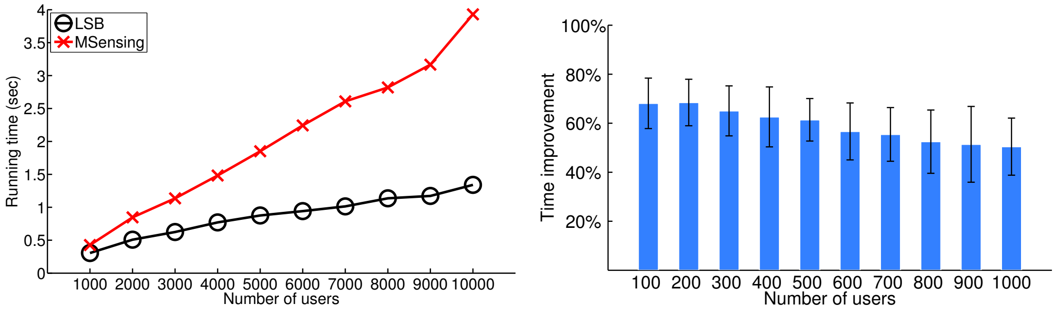


Figure 6.6: Impact of n on $\bar{u}_i$

Evaluation of the User-Centric Incentive Mechanism

Running Time: Figure 6.7(a) shows the running time of different auction mechanisms proposed in Section 6.4.2. We fixed $m = 200$. We can see that *LSB* has the best efficiency while *MSensing* has the worst. Both *MSensing* and *MSensing-L* are linear in n, as we proved in

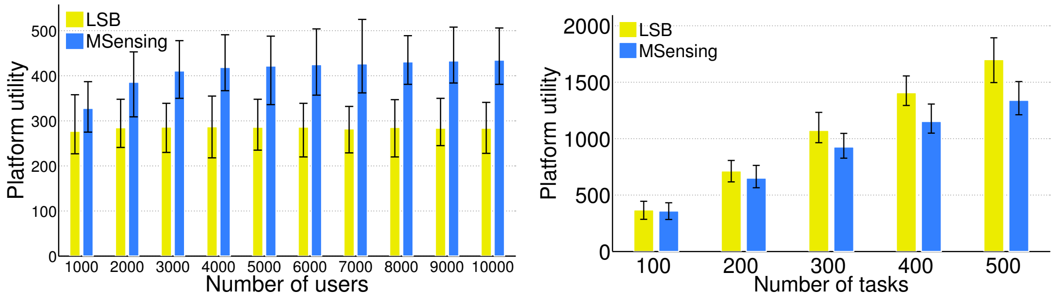
Lemma 6.2. After using lazy update, we can dramatically improve the time efficiency of *MSensing*. As shown in 6.7(b), the time improvement of *MSensing-L* over *MSensing* is roughly around 60%.



(a) Impact of *n* (b) Improvement using lazy greedy.

Figure 6.7: Running time

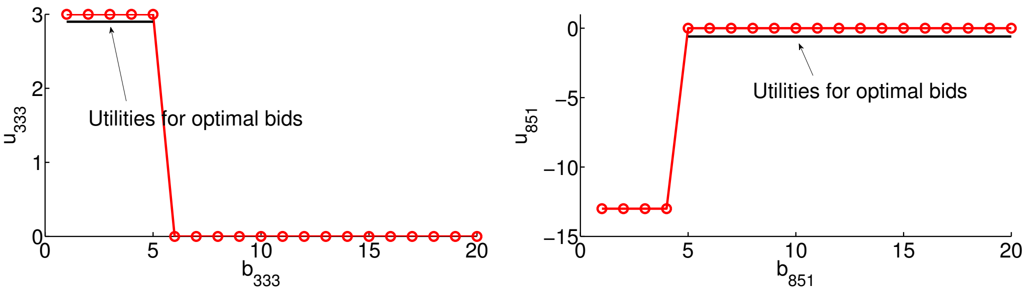
Platform Utility: Now we show how much platform utility we need to sacrifice to achieve the truthfulness compared to *LSB*. As shown in Figure 6.8, we can observe that the platform utility sacrifice is not severe compared to the mechanism with the best-known platform utility approximation. We also observe that when the number of users becomes larger, the gap between *MSensing-L* and *LSB* becomes smaller.



(a) Impact of *n* (b) Impact of *m*

Figure 6.8: Platform utility

Truthfulness: We also verified the truthfulness of *MSensing* by randomly picking two users (ID = 944 and ID = 427) and allowing them to bid prices different from their true costs. We illustrate the results in Figure 6.9. As we can see, user 944 achieves its optimal utility if it bids truthfully ($b_{944} = c_{944} = 10$) in Figure 6.9(a) and user 427 achieves its optimal utility if it bids truthfully ($b_{427} = c_{427} = 15$) in Figure 6.9(b).



(a) $c_{944} = 10$

(b) $c_{427} = 15$

Figure 6.9: Truthfulness of MSensing

6.6 Conclusion and Discussion

In this work, we have designed incentive mechanisms that can be used to motivate smartphone users to participate in mobile phone sensing, which is a new sensing paradigm allowing us to collect and analyze sensed data far beyond the scale of what was previously possible. We have considered two different models from different perspectives: the platform-centric model where the platform provides a reward shared by participating users, and the user-centric model where each user can ask for a reserve price for its sensing service.

For the platform-centric model, we have modeled the incentive mechanism as a Stackelberg game in which the platform is the leader and the users are the followers. We have proved that this Stackelberg game has a unique equilibrium, and designed an efficient mechanism for computing it. This enables the platform to maximize its utility while no user can improve its utility by deviating from the current strategy.

For the user-centric model, we have designed an auction mechanism, called MSensing. We have proved that MSensing is 1) computationally efficient, meaning that the winners and the payments can be computed in polynomial time; 2) individually rational, meaning that each user will have a non-negative utility; 3) profitable, meaning that the platform will not incur a deficit; and more importantly, 4) truthful, meaning that no user can improve its utility by asking for a price different from its true cost. Our mechanism is scalable because its running time is linear in the number of users.

Discussion

For the Platform-Centric model, we assumed that the users have the knowledge of other users' cost. As a future research topic, we are interested in the case where the cost information of other users is not available. Instead, each user knows only the cost distribution function. This distribution function can be obtained based on the statistical data in the smartphone market and the technical specifications of smartphones. For example, the market share of iPhone is 19.4% in 2011 according to Gartner [52]. Hence it is very reasonable to assume that the probability that a users cost is equal to that of an iPhone is 0.194. With the cost distribution function, we will reformulate the utility functions for both users and the platform. Under the new Stackelberg game formulation, we will study the existence and the uniqueness, if possible, of the NE in the WTDG. Our ultimate goal is to compute the Stackelberg Equilibrium of the Stackelberg game with incomplete information.

Part III

Security and Privacy

Chapter 7

Being Smarter: How to Cope with a Smart Jammer

Wireless networks are highly vulnerable to jamming attacks, since jamming attacks are easy to launch. Attacks of this kind usually aim at the physical layer and are realized by means of a high transmission power signal that corrupts a communication channel, as shown in Figure 7.1.

7.1 Introduction

In this work, we are interested in defending against smart jammers, who can quickly learn the transmission pattern of the users and adjust their jamming strategies so as to exacerbate the damage. Since jammers need to consider transmission cost, transmitting with the maximum power may not be the optimal strategy. As a first step along this line, we study the battle between a single user (a transmitter-receiver pair) and a single smart jammer (a malicious transmitter). This problem arises, for example, in military operations, where one radio station transmits data to another in a hostile environment. In this paper, we aim to derive the optimal power control for the user in the presence of a smart jammer.

Game theory is a natural tool to model and address this problem. Jamming defense can be considered a game, where both the user and the jammer are players. Previous works [2, 3] have been done on this topic by proving the existence of Nash Equilibria and computing a Nash Equilibrium. A *Nash Equilibrium* (NE) is the status where no player has an incentive to change its strategy unilaterally so as to increase its own utility. However, Nash Equilibrium is not the best solution to the problem studied in this paper, because the rationality of Nash Equilibrium is based on the assumption that all players take actions simultaneously. In our model, the jammer is intelligent in the sense that it can quickly learn

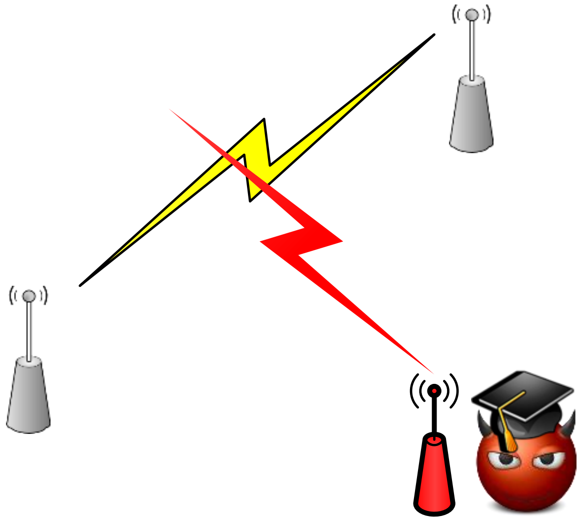


Figure 7.1: Jamming in wireless networks

the user's transmission power and adjust its transmission power accordingly. Stackelberg game serves the purpose of modeling this scenario. In this game, players, including one *leader* and one *follower*, are in a hierarchical structure [1]. The leader takes actions first, and then the follower takes actions accordingly. Similar to the Nash Equilibrium in the standard game, there is *Stackelberg Equilibrium* in this game. Different from the Nash Equilibrium, Stackelberg Equilibrium is the optimal strategy of the leader, given the fact that the follower would take actions according to the leader's strategy, together with the optimal strategy of the follower corresponding to the leader's optimal strategy.

To the best of our knowledge, this paper is the first to study the power control problem in the presence of a smart jammer. As an initial step, we consider a single user and a single jammer (the more challenging scenario with multi users/jammers is a subject of future research). Both the user and the jammer can adjust their transmission power levels. We consider both the single-channel model and the multi-channel model. We model the power control problem with a smart jammer as a Stackelberg game [133], called *Power*

[1] There could be more than one follower. Since we only consider one follower in this paper, we refer the readers to [46] for more details.

Control with Smart Jammer (PCSJ) game. In this game, the user is the *leader* and the jammer is the *follower*. The user is aware of the jammer's existence and has the knowledge of jammer's intelligence, based on which the user chooses an optimal strategy so as to maximize its own utility, while the jammer plays its best response strategy given the user's strategy. For the single-channel model, we derive closed-form expressions for the jammer's best response strategy and the user's optimal strategy, which together constitute the unique Stackelberg Equilibrium (SE). For the multi-channel model, we design an algorithm for computing the jammer's best response strategy, given the user's strategy. We also develop two algorithms to approximate the user's optimal strategy and thus the SE strategies.

The rest of this paper is organized as follows: In Section 7.2, we briefly describe the related works. In Section 7.3, we introduce the system model and the Stackelberg game formulation. In Section 7.5.1, we study the PCSJ game under the single-channel model. In Section 7.5.2, we study the PCSJ game under the multi-channel model. In Section 7.6, we present numerical results. We conclude this paper in Section 7.7.

7.2 Related Work

Due to the importance of jamming defense, wireless network jamming has been extensively studied in the past few years. Many jamming defense mechanisms have been proposed on both the physical layer [86, 87, 107, 154, 155] and the MAC layer [120, 121] to detect jamming, as well as to avoid it. Spread spectrum technologies have been shown to be very effective to avoid jamming. With enough bandwidth or widely spread signals, it becomes harder to detect the start of a packet quickly enough in order to jam it.

Since jamming activities can be considered as a player (the jammer) playing against another player (the user), game theory is an appropriate tool to deal with this kind of problem. Many previous works have studied jamming defense with game theory formulations [2, 3, 85, 124, 151, 171]. In [2], Altman *et al.* studied the jamming game in wireless networks with transmission cost. In this game, both the user and the jammer take the power allocation on channels as their strategies. The utility of the user is the weighted capacity minus transmission cost. The utility of the jammer is the negative of the user's weighted capacity minus transmission cost. The authors proved the existence and uniqueness of Nash Equilibrium. In addition, they provided analytical expressions for the equilibrium strategies. In [3], the same group of authors extended the jamming problem to the case with several jammers. The difference from [2] is that they did not consider transmission cost and they considered SINR and $-$SINR as the utility values for the user and the jammers, respectively. They showed that the jammers equalize the quality of the best sub-carriers for transmitter on as low level as their power constraint allows, meanwhile the user distributes its power among these jamming sub-carriers. In [124], Sagduyu *et al.* considered the power-controlled MAC game, which includes two types of players, selfish and malicious transmitters. Each type of user has a different utility function depending on throughput reward and energy cost. They also considered the case where the transmitters have incomplete information regarding other transmitter's types, modeled as probabilistic beliefs. They derived the Bayesian Nash Equilibrium strategies for different degrees of uncertainty, and characterized the resulting equilibrium throughput of selfish nodes.

The jamming problems have also been studied in cognitive radio networks [85, 151, 171]. The anti-jamming game in this scenario is often modeled as a (stochastic) zero-sum game, where the sum of the utility values of the jammer(s) and the secondary user is zero. In [171], Zhu *et al.* assumed the transition between idle and busy states of the channel to be Markovian. They considered a single secondary user and a single jammer in the cognitive radio system. The strategy of the user is the channel selected to transmit on, while the strategy of the jammer is the channel selected to jam. The utility of the user is 1 if the selected channel is not occupied by the primary user and not jammed by the jammer. They considered mixed strategies and proved the conditions for the uniqueness of the Nash Equilibrium. They also showed that the secondary user can either improve its sensing capability to confuse the jammer or choose to communicate under states where the available channels are less prone to jamming, in order to improve its utility value. In [85], Li and Han studied the problem of defending primary user emulation attack, which is similar to the jamming attack in wireless networks. There is only one jammer and one or multiple secondary users in their models. The strategy of each secondary user is the channel selected to transmit on, while the strategy of the jammer is the channel selected to jam. The utility of each secondary user is a reward if it senses a channel and the jammer is not jamming. They computed the unique Nash Equilibrium and analyzed the efficiency. In [151], Wu *et al.* first investigated the case where a secondary user can access only one channel at a time and then extended to the scenario where secondary users can access all the channels simultaneously. For the former case, the secondary user uses channel hopping as its defense strategy. The utility of the secondary user is equal to a communication gain, if the transmission is successful, minus cost and a significant loss when jammed. They found an approximation to the Nash Equilibrium by letting the user and jammers iteratively update their strategies against each other. For the latter case, the secondary user could allocate power to several channels. The utility of the secondary user is equal to the total number of successful transmissions. They showed that the defense strategy from the Nash Equilibrium is optimal.

In all the previous works on jamming defense, the authors assumed that the users and the jammers take actions simultaneously. In this paper, we study the power control problem in the presence of a smart jammer, which has more power compared to the jammer model studied before. To the best of our knowledge, we are the first to address this problem.

7.3 System Model

In this section, we present the system model and formulate the problem to be studied.

System Model

Our system consists of a *user* (i.e., a transmitter-receiver pair), and a *jammer* (i.e., a malicious transmitter), as illustrated in Figure 7.1. The user (jammer, respectively) has control over its own transmission power. This problem arises, for example, in military operations, where one radio station transmits data to another radio station in a hostile environment. We consider two models in this paper: single-channel model and multi-channel model.

Single-channel model: Let P denote the transmission power of the user and J denote the transmission power of the jammer. In addition, we assume that the user and the jammer transmit with cost E and C *per unit power*. As in [3, 124], we adopt SINR as the reward of the user in our model. Hence, the *utility of the user* is

$$u_s(P,J) = \frac{\alpha P}{N + \beta J} - EP, \tag{7.1}$$

and the *utility of the jammer* is

$$v_s(P,J) = -\frac{\alpha P}{N + \beta J} - CJ, \tag{7.2}$$

where N is the background noise on the channel, and $\alpha > 0$ and $\beta > 0$ are fading channel gains of the user and the jammer, respectively. Note that we have omitted power constraint in this model. If power constraint is added, we can still derive closed-form expressions for the SE. However, the corresponding analysis is more complicated, involving ad-hoc discussions of many cases. Hence, we choose to concentrate on this model, which allows us to to emphasize the main contributions without excessive ad-hoc analysis.

Multi-channel model: We assume that there are n available channels. Let $\alpha_i \in (0,1]$ and $\beta_i \in (0,1]$ denote the fading channel gains of the user and the jammer on channel i, respectively. Let $\hat{P} > 0$ and $\hat{J} > 0$ denote the *total transmission power* of the user and the jammer, respectively. Let P_i and J_i denote the transmission power allocated to channel i by the user and the jammer, respectively. Let $\boldsymbol{P} = (P_1, P_2, \ldots, P_n)$ and $\boldsymbol{J} = (J_1, J_2, \ldots, J_n)$ denote the *transmission power vectors* of the user and the jammer, respectively. $\boldsymbol{P}$ is feasible if $\sum_{i=1}^{n} P_i \leq \hat{P}$, and $\boldsymbol{J}$ is feasible if $\sum_{i=1}^{n} J_i \leq \hat{J}$. Let $\mathscr{P} = \{(P_1, P_2, \ldots, P_n) | P_i \geq 0, \sum_{i=1}^{n} P_i \leq \hat{P}\}$ and $\mathscr{J} = \{(J_1, J_2, \ldots, J_n) | J_i \geq 0, \sum_{i=1}^{n} J_i \leq \hat{J}\}$ denote the sets of feasible power vectors of the user and the jammer, respectively. Similarly to the single-channel model, we assume that the user and the jammer transmit with cost E and C per unit power. The *utility of the user* is

$$u_m(\boldsymbol{P}, \boldsymbol{J}) = \sum_{i=1}^{n} \frac{\alpha_i P_i}{N_i + \beta_i J_i} - E \sum_{i=1}^{n} P_i. \tag{7.3}$$

The *utility of the jammer* is

$$v_m(\boldsymbol{P}, \boldsymbol{J}) = -\sum_{i=1}^{n} \frac{\alpha_i P_i}{N_i + \beta_i J_i} - C \sum_{i=1}^{n} J_i. \tag{7.4}$$

In this paper, we deal with a *smart* jammer, who can quickly learn the user's transmission power and adjust its transmission power accordingly to maximize its utility. The user's transmission power can be accurately learned using physical carrier sensing and location knowledge. We are interested in determining the transmission power of the user such that its utility is maximized, in the presence of a smart jammer. We call this problem *the power control problem with a smart jammer*.

7.4 Stackelberg Game Formulation

In our model, the jammer is smart and can adjust its transmission power based on the user's transmission power. Based on this fact, we model the power control problem in the presence of a smart jammer as a Stackelberg game, called *Power Control with Smart*

Jammer (PCSJ) game. In this game, both the user and the jammer are *players*, of which the user is the leader, and the jammer is the follower. The *strategy* of each player is its transmission power. The *utility* of the user (resp. the jammer) is defined in (7.1) (resp. (7.2)) for the single-channel model and (7.3) (resp. (7.4)) for the multi-channel model.

7.5 PCSJ Game

7.5.1 PCSJ under Single-Channel Model

In this section, we study the PCSJ game under the single-channel model. First, we compute the best response strategy of the jammer, for a given strategy of the user. Then we compute the optimal strategy of the user, based on the knowledge of the best response strategy of the jammer.

Jammer's Best Response Strategy

Assume that the user's strategy P is given. Then the jammer's best response strategy can be computed by solving the following optimization problem.

$$\max_{J\geq 0} v_s(P,J) = -\frac{\alpha P}{N+\beta J} - CJ. \tag{7.5}$$

Thus we have the following lemma.

Lemma 7.1. *Let P be a given strategy of the user. Then the corresponding optimal strategy of the jammer is*

$$J(P) = \begin{cases} 0, & P \leq \frac{CN^2}{\alpha\beta}, \\ \frac{\sqrt{\frac{\alpha\beta P}{C}}-N}{\beta}, & P > \frac{CN^2}{\alpha\beta}. \end{cases} \tag{7.6}$$

Proof. To find the maximum value of $v_s(P,J)$, we differentiate $v_s(P,J)$ with respect to J

and set the resulting derivative equal to 0,

$$0 = \frac{\partial v_s(P,J)}{\partial J} = \frac{\alpha\beta P}{(N+\beta J)^2} - C. \tag{7.7}$$

Considering the constraint $J \geq 0$, we have jammer's optimal strategy in (7.6). □

User's Optimal Strategy

The user is aware of the existence of the jammer and knows that the jammer will play its best response strategy to maximize its own utility. Therefore, the user can derive the jammer's strategy based on Lemma 7.1. To compute the user's optimal strategy, we solve the following optimization problem.

$$\max_{P\geq 0} u_s(P,J(P)) = \frac{\alpha P}{N+\beta J(P)} - EP, \tag{7.8}$$

where $J(P)$ is given in (7.6).

The optimal strategy of the user is given in the following Lemma.

Lemma 7.2. *The optimal strategy of the user is*

$$P^{SE} = \begin{cases} \frac{\alpha C}{4\beta E^2}, & E \leq \frac{\alpha}{2N}, \\ \frac{CN^2}{\alpha\beta}, & \frac{\alpha}{2N} < E \leq \frac{\alpha}{N}, \\ 0, & E > \frac{\alpha}{N}. \end{cases} \tag{7.9}$$

Proof. Plugging (7.6) into the objective function (7.8), we have

$$u_s(P,J(P)) = \begin{cases} (\frac{\alpha}{N} - E)P, & P \leq \frac{CN^2}{\alpha\beta}, \\ \sqrt{\frac{\alpha CP}{\beta}} - EP, & P > \frac{CN^2}{\alpha\beta}. \end{cases} \tag{7.10}$$

Hence $u_s(P,J(P))$ is a linear function in P for $0 \leq P \leq \frac{CN^2}{\alpha\beta}$, and is a strictly concave function in P for $P > \frac{CN^2}{\alpha\beta}$. Note that the derivative of $u_s(P,J(P))$ with respect to P in the range

$P > \frac{CN^2}{\alpha\beta}$ is given by

$$\frac{\partial u_s(P,J(P))}{\partial P} = \frac{1}{2}\sqrt{\frac{\alpha C}{\beta P}} - E. \tag{7.11}$$

Setting equation (7.11) to 0, we obtain $P = \frac{\alpha C}{4\beta E^2}$.

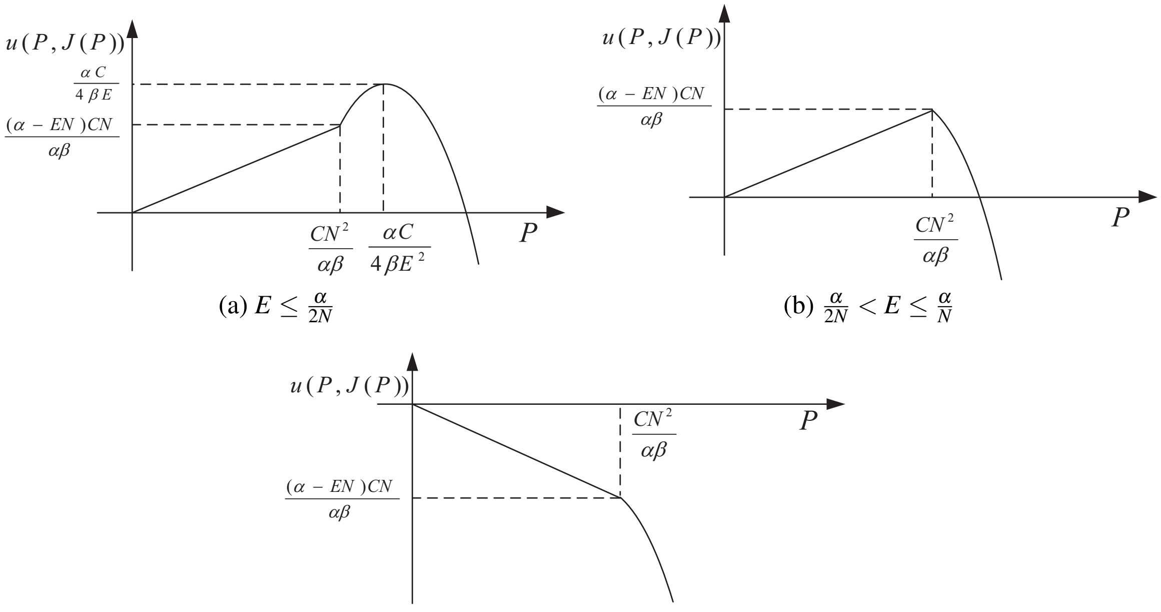
To compute the maximum value of (7.10), we consider three disjoint cases.

Case-1: $E \leq \frac{\alpha}{2N}$. We can verify that $\frac{CN^2}{\alpha\beta} \leq \frac{\alpha C}{4\beta E^2}$. As illustrated in Figure 7.2(a), $u_s(P,J(P))$ achieves its maximum value of $\frac{\alpha C}{4\beta E}$ when $P = \frac{\alpha C}{4\beta E^2}$.

Case-2: $\frac{\alpha}{2N} < E \leq \frac{\alpha}{N}$. We can verify that $\frac{CN^2}{\alpha\beta} > \frac{\alpha C}{4\beta E^2}$. As illustrated in Figure 7.2(b), $u_s(P,J(P))$ achieves its maximum value of $\frac{(\alpha - EN)CN}{\alpha\beta}$ when $P = \frac{CN^2}{\alpha\beta}$.

Case-3: $E > \frac{\alpha}{N}$. In this case, we also have $\frac{CN^2}{\alpha\beta} > \frac{\alpha C}{4\beta E^2}$. As illustrated in Figure 7.2(c), $u_s(P,J(P))$ achieves its maximum value of 0 when $P = 0$.

This proves the lemma. □



(a) $E \leq \frac{\alpha}{2N}$

(b) $\frac{\alpha}{2N} < E \leq \frac{\alpha}{N}$

(c) $E > \frac{\alpha}{N}$

Figure 7.2: User's utility function for different values of E

Lemmas 7.1 and 7.2 lead to the following theorem.

Theorem 7.1. *The strategy pair* (P^{SE}, J^{SE}) *is the Stackelberg Equilibrium of the PCSJ game, where*

$$P^{SE} = \begin{cases} \frac{\alpha C}{4\beta E^2}, & E \leq \frac{\alpha}{2N}, \\ \frac{CN^2}{\alpha\beta}, & \frac{\alpha}{2N} < E \leq \frac{\alpha}{N}, \\ 0, & E > \frac{\alpha}{N}, \end{cases}$$

and

$$J^{SE} = \begin{cases} \frac{\frac{\alpha}{2E} - N}{\beta}, & E \leq \frac{\alpha}{2N}, \\ 0, & E > \frac{\alpha}{2N}. \end{cases}$$

Remark. Note that the user needs to have the knowledge of β to compute P^{SE}. This can be achieved as follows: The user randomly selects its initial transmission power $P^{[0]} > 0$. It then keeps increasing its transmission power to $P^{[i]}$ until the received jamming signal is non-zero. For example, it can set $P^{[i]} = 2^i \times P^{[0]}$ for $i > 0$. The received jamming signal can be measured by taking advantage of the delay in jamming's decision making. According to (7.6), we have $\beta J(P^{[i]}) = \sqrt{\frac{\alpha\beta P^{[i]}}{C}} - N$. Hence we have $\beta = \frac{C\left(\beta J(P^{[i]}) + N\right)^2}{\alpha P^{[i]}}$, where $\beta J(P^{[i]})$ is the received jamming signal.

Comparison with Nash Equilibrium

Now we study the impact of the jammer's intelligence on the utility values of both players.

We show that *the utility values of both the user and the jammer at the SE of the PCSJ game are at least as high as those at the NE of the PCRJ game.*

Lemma 7.3. *There exists a unique NE* (P^{NE}, J^{NE}) *in the PCRJ game when the jammer does not have intelligence to learn the user's strategy. In addition,*

$$(P^{NE}, J^{NE}) = \begin{cases} (\frac{\alpha C}{\beta E^2}, \frac{\alpha/E-N}{\beta}), & E \leq \frac{\alpha}{N}, \\ (0,0), & E > \frac{\alpha}{N}, \end{cases} \tag{7.12}$$

$$u_s(P^{NE}, J^{NE}) = 0, \tag{7.13}$$

and

$$v_s(P^{NE}, J^{NE}) = \begin{cases} \frac{C}{\beta}\left(N - \frac{2\alpha}{E}\right), & E \leq \frac{\alpha}{N}, \\ 0, & E > \frac{\alpha}{N}. \end{cases} \tag{7.14}$$

Proof. We consider two disjoint cases:

Case-1: $E \leq \frac{\alpha}{N}$. If $J = \frac{\alpha/E-N}{\beta}$, the value of (7.1) is 0, for any P. However, in order to have $J = \frac{\alpha/E-N}{\beta}$, we must have $P = \frac{\alpha C}{\beta E^2}$ according to (7.7). Thus the NE is $(P^{NE}, J^{NE}) = (\frac{\alpha C}{\beta E^2}, \frac{\alpha/E-N}{\beta})$.

We now prove the uniqueness of NE in this case. Assume to the contrary that there exists another NE (P', J'). We first use contradiction to prove that $J' = J^{NE}$. If $J' > J^{NE}$, $P' = 0$ according to (7.1). However, if $P' = 0$, we must have $J' = 0$ according to (7.2), contradicting to the assumption that $J' > J^{NE} > 0$. If $J' < J^{NE}$, (7.1) becomes a strictly increasing function of P. Hence the user can increase its utility by unilaterally increase its transmission power, contradicting the NE assumption (P', J'). Thus we have proved that $J' = J^{NE}$. Since J' is a function of P' according to (7.7), we have $P' = P^{NE}$.

Case-2: $E > \frac{\alpha}{N}$. The derivative of (7.1) with respect to P is

$$\frac{\alpha}{N + \beta J} - E < 0,$$

for any $J \geq 0$. Hence $P = 0$ is the unique optimal strategy for the user. Since $v_s(0,J) = -CJ$, $J = 0$ is the unique optimal strategy for the jammer. The NE is then $(P^{NE}, J^{NE}) = (0,0)$.

Combining (7.12), (7.1) and (7.2), we get (7.13) and (7.14). □

Lemma 7.3 leads to the following important theorem, as stated in the 3rd paragraph of this section.

Theorem 7.2. *Let (P^{SE}, J^{SE}) be the SE of the PCSJ game, and (P^{NE}, J^{NE}) be the NE of the PCRJ game. Then we have*

$$\begin{aligned} u_s(P^{SE}, J^{SE}) &\geq u_s(P^{NE}, J^{NE}), & (7.15) \\ v_s(P^{SE}, J^{SE}) &\geq v_s(P^{NE}, J^{NE}). & (7.16) \end{aligned}$$

Proof. It is clear that $u_s(P^{SE}, J^{SE}) \geq u_s(P^{NE}, J^{NE})$. For the jammer, it suffices to prove that $v_s(P^{SE}, J^{SE}) \geq v_s(P^{NE}, J^{NE})$ when $\frac{\alpha}{2N} < E \leq \frac{\alpha}{N}$. According to (7.14), we have

$$\begin{aligned} v_s(P^{NE}, J^{NE}) &= \frac{C}{\beta}\left(N - \frac{2\alpha}{E}\right) \leq \frac{C}{\beta}(N - 2N) \\ &= -\frac{CN}{\beta} = v_s(P^{SE}, J^{SE}) \end{aligned}$$

The theorem is proved. □

7.5.2 PCSJ under Multi-Channel Model

In this section, we study the PCSJ game under the multi-channel model.

Jammer's Best Response Strategy

Given the user's strategy $\boldsymbol{P}$, the problem of power allocation for the jammer can be formulated as a convex optimization problem as follows.

$$\max_{\boldsymbol{J} \in \mathscr{J}} v_m(\boldsymbol{P},\boldsymbol{J}) = -\sum_{i=1}^{n} \frac{\alpha_i P_i}{N_i + \beta_i J_i} - C \sum_{i=1}^{n} J_i. \tag{7.17}$$

In the following, Theorem 7.3 guarantees the existence of the jammer's best response strategy, and Theorem 7.4 computes the jammer's best response strategy, when the user's strategy is given.

Theorem 7.3. *Let $\boldsymbol{P}$ be user's strategy. There exists a unique $\boldsymbol{J}(\boldsymbol{P})$ such that $v_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$ is maximized.*

Proof. Since $v_m(\boldsymbol{P},\cdot)$ is a continuous function on the compact set $\mathscr{J}$, it can achieve its maximum value at some $\boldsymbol{J} \in \mathscr{J}$ [140].

Without loss of generality, we assume that $P_i > 0$ for $1 \le i \le k$ and $P_i = 0$ for $k < i \le n$. It is obvious that, in any optimal solution of the optimization problem (7.17), we have $J_i(\boldsymbol{P}) = 0$ for $k < i \le n$. Otherwise, we can increase the value of $v_m(\boldsymbol{P},\boldsymbol{J})$ by setting $J_i(\boldsymbol{P}) = 0$, contradicting the optimality of $\boldsymbol{J}$. The optimization problem then becomes

$$\max_{\boldsymbol{J}} v_m(\boldsymbol{P},\boldsymbol{J}) = -\sum_{i=1}^{k} \frac{\alpha_i P_i}{N_i + \beta_i J_i} - C \sum_{i=1}^{k} J_i \tag{7.18}$$

s.t.

$$\sum_{i=1}^{k} J_i \le \hat{J}, J_i \ge 0, \text{for all } i \in [1,k].$$

The first order partial derivative of $v_m(\boldsymbol{P},\boldsymbol{J})$ with respective to J_i, for $i \in [1,k]$, is

$$\frac{\partial v_m(\boldsymbol{P},\boldsymbol{J})}{\partial J_i} = \frac{\alpha_i \beta_i P_i}{(N_i + \beta_i J_i)^2} - C, \tag{7.19}$$

and the second order partial derivatives of $v_m(\boldsymbol{P},\boldsymbol{J})$ are

$$\frac{\partial^2 v_m(\boldsymbol{P},\boldsymbol{J})}{\partial J_i \partial J_j} = \begin{cases} -\frac{2\alpha_i \beta_i^2 P_i}{(N_i + \beta_i J_i)^3}, & i = j, \\ 0, & i \ne j. \end{cases} \tag{7.20}$$

The Hessian matrix is negative definite [14]: $\bigtriangledown^2 v_m(\boldsymbol{P},\boldsymbol{J}) \prec 0$, implying that the objective function (7.18) is strictly concave, and there is a unique solution to the optimization problem. □

Theorem 7.4. *Let $\boldsymbol{P}$ be user's strategy. We define $\pi(\lambda) = \sum_{i=1}^{k} \left[\frac{\sqrt{\frac{\alpha_i \beta_i P_i}{C+\lambda}} - N_i}{\beta_i} \right]^+$ for $\lambda \in [0,\infty)$, where $[x]^+ = \max\{x,0\}$. Then the best response strategy of the jammer is $\boldsymbol{J}(\boldsymbol{P}) = (J_1(\boldsymbol{P}), J_2(\boldsymbol{P}), \ldots, J_n(\boldsymbol{P}))$, where*

$$J_i(\boldsymbol{P}) = \begin{cases} \left[\frac{\sqrt{\frac{\alpha_i \beta_i P_i}{C+\lambda_0}} - N_i}{\beta_i} \right]^+, & 1 \le i \le k, \\ 0, & k+1 \le i \le n, \end{cases} \tag{7.21}$$

and

$$\lambda_0 = \begin{cases} 0, & \pi(0) < \hat{J}, \\ \textit{the unique root of } \pi(\lambda) = \hat{J}, & \textit{otherwise}. \end{cases} \tag{7.22}$$

In addition, $\boldsymbol{J}(\boldsymbol{P})$ can be computed in $O(n \log n)$ time. □

Proof. We convert the optimization problem (7.18) into a standard form of convex optimization problem [14]:

$$\min_{\boldsymbol{J}} f(\boldsymbol{J}) = \sum_{i=1}^{k} \frac{\alpha_i P_i}{N_i + \beta_i J_i} + C \sum_{i=1}^{k} J_i \tag{7.23}$$

s.t.

$$\sum_{i=1}^{k} J_i - \hat{J} \le 0,$$
$$-J_i \le 0, \forall i \in [1,k].$$

The first order partial derivative of $f(\boldsymbol{J})$ with respective to J_i, for $i \in [1,k]$, is

$$\frac{\partial f(\boldsymbol{J})}{\partial J_i} = -\frac{\alpha_i \beta_i P_i}{(N_i + \beta_i J_i)^2} + C, \tag{7.24}$$

and the second order partial derivatives of $f(\boldsymbol{J})$ are

$$\frac{\partial^2 f(\boldsymbol{J})}{\partial J_i \partial J_j} = \begin{cases} \frac{2\alpha_i \beta_i^2 P_i}{(N_i+\beta_i J_i)^3}, & i = j, \\ 0, & i \neq j. \end{cases} \tag{7.25}$$

The Hessian matrix is positive definite [14]: $\bigtriangledown^2 f(\boldsymbol{J}) \succ 0$, implying that the objective function (7.23) is strictly convex.

Since the constraints of the optimization problem are also convex, we know that the Karush-Kuhn-Tucker (KKT) conditions [14] are necessary and sufficient for optimality.

We define the Lagrangian as

$$L_J(\boldsymbol{J},\lambda) = v_m(\boldsymbol{P},\boldsymbol{J}) + \lambda_0 \left(\sum_{i=1}^{k} J_i - \hat{J} \right) + \sum_{i=1}^{k} \lambda_i J_i, \tag{7.26}$$

where $\lambda_i \geq 0$, $0 \leq i \leq k$, are the Lagrange multipliers. The KKT conditions for the optimal solution of (7.23) are given by

$$\begin{aligned} \frac{\partial L_J(\boldsymbol{J},\lambda)}{\partial J_i} &= 0, \forall i \in [1,k], && (7.27) \\ \sum_{i=1}^{k} J_i - \hat{J} &\leq 0, && (7.28) \\ -J_i &\leq 0, \forall i \in [1,k], && (7.29) \\ \lambda_i &\geq 0, \forall i \in [0,k], && (7.30) \\ \lambda_0 \left(\sum_{i=1}^{k} J_i - \hat{J} \right) &= 0, && (7.31) \\ -\lambda_i J_i &= 0, \forall i \in [1,k]. && (7.32) \end{aligned}$$

Combining (7.27), (7.28), and (7.32), we have (7.21) and (7.22).

Our algorithm for computing $\boldsymbol{J}(\boldsymbol{P})$ is given in Algorithm 11. Lines 1–8 compute the value of λ_0 satisfying (7.22). Line 9 computes $\boldsymbol{J}(\boldsymbol{P})$ according to (7.21). When $\pi(0) < \hat{J}$, Line 2 of Algorithm 11 computes $\lambda_0 = 0$, which is consistent with the first case in (7.22).

Algorithm 11: Computation of $\boldsymbol{J}(\boldsymbol{P})$

```
   input : The power vector $\boldsymbol{P}$ of the user
1  if $\pi(0) < \hat{J}$ then
2      $\lambda_0 \leftarrow 0$;
3  else
4      $\lambda_0^i \leftarrow \frac{\alpha_i \beta_i P_i}{N_i^2} - C$, for $i \leftarrow 1$ to $k$ ;
5      Sort $\{\lambda_0^i\}_{i=1}^k$ such that $\lambda_0^{i_1} \leq \lambda_0^{i_2} \leq \cdots \leq \lambda_0^{i_k}$;
6      Find $r \in [1,k]$ such that $\pi(\lambda_0^{i_{r-1}}) \geq \hat{J} > \pi(\lambda_0^{i_r})$, where $\lambda_0^{i_0} = 0$;
7      $\lambda_0 \leftarrow \left(\sum_{j=r}^k \sqrt{\frac{\alpha_{i_j} P_{i_j}}{\beta_{i_j}}} \Big/ \left(\hat{J} + \sum_{j=r}^k \frac{N_{i_j}}{\beta_{i_j}}\right)\right)^2 - C$;
8  end
9  $J_i(\boldsymbol{P}) \leftarrow \begin{cases} \left[\frac{\sqrt{\frac{\alpha_i \beta_i P_i}{C+\lambda_0}} - N_i}{\beta_i}\right]^+, & 1 \leq i \leq k, \\ 0, & k+1 \leq i \leq n, \end{cases}$ ;
10 return $\boldsymbol{J}(\boldsymbol{P})$;
```

When $\pi(0) \geq \hat{J} > 0$, we have $\left[\frac{\sqrt{\frac{\alpha_i \beta_i P_i}{C+\lambda}} - N_i}{\beta_i}\right]^+ > 0$ for at least one $i \in [1,k]$. Line 4 computes the values $\{\lambda_0^i\}_{i=1}^k$, such that $\frac{\sqrt{\frac{\alpha_i \beta_i P_i}{C+\lambda}} - N_i}{\beta_i} > 0$ if and only if $\lambda < \lambda_0^i$. Line 5 sorts these values such that $\lambda_0^{i_1} \leq \lambda_0^{i_2} \leq \cdots \leq \lambda_0^{i_k}$. Hence $\pi(\lambda) = \sum_{j=l}^k \frac{\sqrt{\frac{\alpha_{i_j} \beta_{i_j} P_{i_j}}{C+\lambda}} - N_{i_j}}{\beta_{i_j}}$ for $\lambda \in [\lambda_0^{i_{l-1}}, \lambda_0^{i_l})$. This also implies that $\pi(\lambda_0^{i_k}) = 0$, and $\pi(\lambda)$ is strictly decreasing for $\lambda \in [0, \lambda_0^{i_k}]$. Hence there is a unique $\lambda_0 \in (0, \lambda_0^{i_k})$ such that $\pi(\lambda_0) = \hat{J}$. Lines 6 and 7 compute this value.

Line 5 in Algorithm 11 takes $O(k \log k)$ time. The rest of the algorithm takes $O(k)$ time. Since $k \leq n$, the running time of Algorithm 11 is $O(n \log n)$. □

User's Optimal Strategy

We first rigorously prove the existence of the user's optimal strategy, which implies the existence of SEs of the PCSJ game, and then design algorithms for approximating the user's optimal strategy.

Lemma 7.4. *Let $\{\boldsymbol{P}^{[\kappa]}\}$ be a sequence in $\mathscr{P}$ converging to a point $\bar{\boldsymbol{P}}$ in $\mathscr{P}$. Then the sequence $\{\boldsymbol{J}(\boldsymbol{P}^{[\kappa]})\}$ converges to $\boldsymbol{J}(\bar{\boldsymbol{P}})$.*

Proof. To the contrary, assume that $\{\boldsymbol{J}(\boldsymbol{P}^{[\kappa]})\}$ does not converge to $\boldsymbol{J}(\bar{\boldsymbol{P}})$. Since $\{\boldsymbol{J}(\boldsymbol{P}^{[\kappa]})\}$ is contained in the compact set $\mathscr{P}$, it must have a sub-sequence $\{\boldsymbol{J}(\boldsymbol{P}^{[s_\kappa]})\}$ converging to a point $\boldsymbol{J}' \neq \boldsymbol{J}(\bar{\boldsymbol{P}})$. Clearly $\{\boldsymbol{P}^{[s_\kappa]}\}$ converges to $\bar{\boldsymbol{P}}$ since $\{\boldsymbol{P}^{[\kappa]}\}$ converges to $\bar{\boldsymbol{P}}$. Hence $\{(\boldsymbol{P}^{[s_\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[s_\kappa]}))\}$ converges to $(\bar{\boldsymbol{P}}, \boldsymbol{J}')$. Without loss of generality, we assume that $\{(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[\kappa]}))\}$ converges to $(\bar{\boldsymbol{P}}, \boldsymbol{J}')$.

Since $\boldsymbol{J}(\bar{\boldsymbol{P}})$ is the unique optimal strategy of the jammer for the strategy $\bar{\boldsymbol{P}}$ of the user, we have

$$v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}})) - v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}') > 0. \tag{7.33}$$

Define

$$3\varepsilon = v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}})) - v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}'). \tag{7.34}$$

Since $v_m(\boldsymbol{P}, \boldsymbol{J})$ is a continuous function on $\mathscr{P} \times \mathscr{J}$, it is continuous at points $(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}}))$ and $(\bar{\boldsymbol{P}}, \boldsymbol{J}')$. Since $\{(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\bar{\boldsymbol{P}}))\}$ converges to $(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}}))$, and $\{(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[\kappa]}))\}$ converges to $(\bar{\boldsymbol{P}}, \boldsymbol{J}')$, there exists an integer K such that

$$|v_m(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\bar{\boldsymbol{P}})) - v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}}))| < \varepsilon, \text{ when } \kappa \geq K, \tag{7.35}$$

and

$$|v_m(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[\kappa]})) - v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}')| < \varepsilon, \text{ when } \kappa \geq K. \tag{7.36}$$

Therefore, for all $\kappa \geq K$, we have (using (7.35), (7.34), and (7.36))

$$\begin{aligned}
v_m(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\bar{\boldsymbol{P}})) &> v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}(\bar{\boldsymbol{P}})) - \varepsilon && (7.37) \\
&= (v_m(\bar{\boldsymbol{P}}, \boldsymbol{J}') + 3\varepsilon) - \varepsilon && (7.38) \\
&> v_m(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[\kappa]})) - \varepsilon + 3\varepsilon - \varepsilon && (7.39) \\
&= v_m(\boldsymbol{P}^{[\kappa]}, \boldsymbol{J}(\boldsymbol{P}^{[\kappa]})) + \varepsilon. && (7.40)
\end{aligned}$$

This is in contradiction with the assumption that $\boldsymbol{J}(\boldsymbol{P}^{[\kappa]})$ is the best response strategy of the jammer. This proves the lemma. □

Lemma 7.5. *$u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is a continuous function in $\boldsymbol{P}$.*

Proof. By (7.3), $u_m(\boldsymbol{P}, \boldsymbol{J})$ is continuous in the variables $(\boldsymbol{P}, \boldsymbol{J})$. From Lemma 7.4, $\boldsymbol{J}(\boldsymbol{P})$ is continuous in $\boldsymbol{P}$. Hence $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is continuous in $\boldsymbol{P}$. □

Theorem 7.5. *There exists $\boldsymbol{P}^{SE} \in \mathscr{P}$ such that $(\boldsymbol{P}^{SE}, \boldsymbol{J}(\boldsymbol{P}^{SE}))$ is a Stackelberg Equilibrium of the PCSJ game.*

Proof. We know that $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is a continuous function in $\boldsymbol{P}$. Since the set $\mathscr{P}$ is compact, $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ achieves its maximum at some point $\boldsymbol{P}^{SE} \in \mathscr{P}$ [140]. This proves the theorem.

□

Based on the analytical results of the jammer's best response strategy given user's strategy, the user can optimize its strategy $\boldsymbol{P}$ to maximize its utility $u_m(\boldsymbol{P}, \boldsymbol{J})$, being aware that its decision will affect the jammer's strategy. From the user's prospective, its objective is to solve the following optimization problem.

$$\max_{\boldsymbol{P} \in \mathscr{P}} u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P})) = \sum_{i=1}^{n} \frac{\alpha_i P_i}{N_i + \beta_i J_i(\boldsymbol{P})} - E \sum_{i=1}^{n} P_i, \tag{7.41}$$

where $J_i(\boldsymbol{P})$ is derived from Theorem 7.4.

Although Theorem 7.5 proves the existence of an SE of PCSJ, computing an SE is challenging. The reason is that the objective function in (7.41) is not concave.

Non-Concavity of $u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$: Define $g(\boldsymbol{P}) = u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$. We use an example to show that there exists $\boldsymbol{P}^{[1]}$ and $\boldsymbol{P}^{[2]}$ such that

$$\frac{g(\boldsymbol{P}^{[1]}) + g(\boldsymbol{P}^{[2]})}{2} > g\left(\frac{\boldsymbol{P}^{[1]} + \boldsymbol{P}^{[2]}}{2}\right).$$

In this example, $n = 2$, $\alpha_1 = \alpha_2 = 0.6$, $\beta_1 = 0.5$, $\beta_2 = 0.2$, $N_1 = N_2 = 0.2$, $\hat{P} = 10$, $\hat{J} = 4$, $E = 0.1$, and $C = 1$. We set $\boldsymbol{P}^{[1]} = (4,3)$ and $\boldsymbol{P}^{[2]} = (5,4)$. Using Algorithm 11 and the definition of $u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$, we have $g(\boldsymbol{P}^{[1]}) = 4.49089$, $g(\boldsymbol{P}^{[2]}) = 5.57603$, and $g\left(\frac{\boldsymbol{P}^{[1]}+\boldsymbol{P}^{[2]}}{2}\right) = 4.93331$. Hence we show that $\frac{g(\boldsymbol{P}^{[1]})+g(\boldsymbol{P}^{[2]})}{2} > g\left(\frac{\boldsymbol{P}^{[1]}+\boldsymbol{P}^{[2]}}{2}\right)$.

Algorithm 12: SE-SA

```
input : Algorithm parameters $I$, $T$, $\sigma$, and $\delta$
Randomly initialize $\boldsymbol{P}$, $\boldsymbol{P}_{best} \leftarrow \boldsymbol{P}$;
repeat
    for $i \leftarrow 1$ to $I$ do
        $\boldsymbol{P}_{new} \leftarrow neighbor(\boldsymbol{P})$;
        Randomly select $r$ from $(0,1)$;
        if $u_m(\boldsymbol{P}_{new},\boldsymbol{J}(\boldsymbol{P}_{new})) \geq u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$ or $r \leq e^{(u_m(\boldsymbol{P}_{new},\boldsymbol{J}(\boldsymbol{P}_{new}))-u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P})))/T}$ then
            $\boldsymbol{P} \leftarrow \boldsymbol{P}_{new}$;
            if $u_m(\boldsymbol{P}_{new},\boldsymbol{J}(\boldsymbol{P}_{new})) > u_m(\boldsymbol{P}_{best},\boldsymbol{J}(\boldsymbol{P}_{best}))$ then
                $\boldsymbol{P}_{best} \leftarrow \boldsymbol{P}_{new}$;
            end
        end
    end
    $T \leftarrow \sigma T$;
until $T \leq 1$;
```

We propose two algorithms to approximate the optimal strategy of the user. The first is simulated annealing [73], denoted by SE-SA and presented in Algorithm 12. The second is a mesh-based hill-climbing algorithm, denoted by SE-MESH and presented in Algorithm 13.

Since simulated annealing has been widely used in the literature, we do not give detailed description of SE-SA, and refer the readers to [57]. The algorithm finds a global optimal solution with probability 1 when the number of iterations goes to infinity [57]. The algorithm parameters are as follows. $T > 0$ is the initial temperature. $\sigma \in (0,1)$ is the annealing parameter. I is the number of iterations to be performed at each temperature. δ is the parameter used for generating perturbations. For any feasible power vector $P \in \mathscr{P}$, the function *neighbor*$(\boldsymbol{P})$ generates a perturbation $P' \in \mathscr{P}$ in the following way. For each i, let $P'_i = [P_i + \delta_i]^+$, where δ_i is a random number uniformly drawn from $[-\delta, \delta]$. If $\sum_{i=1}^n P'_i > \hat{P}$, set $P'_i = \frac{\hat{P}P'_i}{\sum_{i=1}^n P'_i}$.

In SE-MESH, we first narrow down the searching space $\mathscr{P}$ to the points $(\delta_1\varepsilon, \delta_2\varepsilon, \ldots, \delta_n\varepsilon)$ on a mesh with space ε between lines. We then select the top t points that have highest values of $u(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$. Starting from each of these t points, we apply a searching strategy similar to that used in SE-SA (Lines 3–12) except that only the point resulting in a higher $u(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is accepted in SE-MESH. In addition, for each point $\boldsymbol{P}$, the searching process terminates if we could not find a neighbor yielding higher $u(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ after I iterations.

Remark 1. The value of β_i for $1 \leq i \leq n$ to compute $\boldsymbol{P}^{SE}$ can be computed similarly using the method in Section 7.5.1 for each channel and compute β_i.

Remark 2. Since $\mathscr{P}$ is compact and $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is a continuous function on $\mathscr{P}$ by Lemma 7.5, $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ is uniformly continuous on $\mathscr{P}$. Therefore, there exists a Lipschitz constant $L > 0$, such that $|u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P})) - u_m(\boldsymbol{P}', \boldsymbol{J}(\boldsymbol{P}'))| \leq L||\boldsymbol{P} - \boldsymbol{P}'||$ [140]. Therefore, as ε approaches zero, the solution computed by SE-MESH converges to the optimal solution. More importantly, we have $u_m(\boldsymbol{P}_\varepsilon, \boldsymbol{J}(\boldsymbol{P}_\varepsilon)) \geq u_m(\boldsymbol{P}_{opt}, \boldsymbol{J}(\boldsymbol{P}_{opt})) - \varepsilon n L$, where $\boldsymbol{P}_\varepsilon$ is the transmit power computed by SE-MESH, $\boldsymbol{P}_{opt}$ is optimal transmission power of the user, and n is the dimension of $\boldsymbol{P}$ vector.

Algorithm 13: SE-MESH

```
input : Algorithm parameters $\varepsilon$, $t$, $I$, and $\delta$
$\boldsymbol{P}_{best} \leftarrow \boldsymbol{0}$;
Let $\mathscr{P}' \leftarrow \{(\delta_1\varepsilon, \delta_2\varepsilon, \ldots, \delta_n\varepsilon) | \delta_i \in \mathbb{Z}^*, 1 \leq i \leq n\} \cap \mathscr{P}$, where $\mathbb{Z}^*$ is the set of
  nonnegative integers;
Compute $u(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ for each $\boldsymbol{P} \in \mathscr{P}'$;
Let $\mathscr{P}'[1], \mathscr{P}'[2], \ldots, \mathscr{P}'[t]$ denote the top $t$ power transmission vectors with highest
  $u(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$;
for $i \leftarrow 1$ to $t$ do
    $\boldsymbol{P} \leftarrow \mathscr{P}'[i]$;
    if $u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P})) > u_m(\boldsymbol{P}_{best}, \boldsymbol{J}(\boldsymbol{P}_{best}))$ then
        $\boldsymbol{P}_{best} \leftarrow \boldsymbol{P}$;
    end
    $cnt \leftarrow 0$;
    while $cnt < I$ do
        $\boldsymbol{P}_{new} \leftarrow neighbor(\boldsymbol{P})$;
        if $u_m(\boldsymbol{P}_{new}, \boldsymbol{J}(\boldsymbol{P}_{new})) \geq u_m(\boldsymbol{P}, \boldsymbol{J}(\boldsymbol{P}))$ then
            $\boldsymbol{P} \leftarrow \boldsymbol{P}_{new}$;
            $cnt \leftarrow 0$;
            if $u_m(\boldsymbol{P}_{new}, \boldsymbol{J}(\boldsymbol{P}_{new})) > u_m(\boldsymbol{P}_{best}, \boldsymbol{J}(\boldsymbol{P}_{best}))$ then
                $\boldsymbol{P}_{best} \leftarrow \boldsymbol{P}_{new}$;
            end
        else
            $cnt \leftarrow cnt + 1$;
        end
    end
end
```

7.6 Performance Evaluation

In this section, we validate the theoretical insights of the PCSJ game through extensive simulations.

Simulation Setup

For the single-channel model, five variables determine the players' strategies and their utility values, which are N, α, β, C, and E. Among these five variables, only α and

β, i.e., fading channel gains of the user and the jammer, may vary significantly due to the change of players' physical locations. Hence, we explore the relations of user and jammer's utility values with respect to different values of α and β. We set α and β to be in the range of $[0.1, 0.9]$. Moreover, let $C = E = 1$ (as in [2]), and $N = 0.2$.

For the multi-channel model, we have $n \in [2, 12]$ and $\hat{P} = \hat{J} = 10$. We assume that α_i is randomly distributed over $(0, 1]$ and β_i is randomly distributed over $(0, 0.5]$, for all $1 \leq i \leq n$. Same as the single-channel model, we have $C = E = 1$ and $N_i = 0.2$ for all $1 \leq i \leq n$. For the parameters of SE-SA, we set $I = 1000$, $T = 100$, $\sigma = 0.6$, and $\delta = 0.25$. For the parameters of SE-MESH, we set $\varepsilon = 1$, $t = 100$, $I = 1000$, and $\delta = 0.25$.

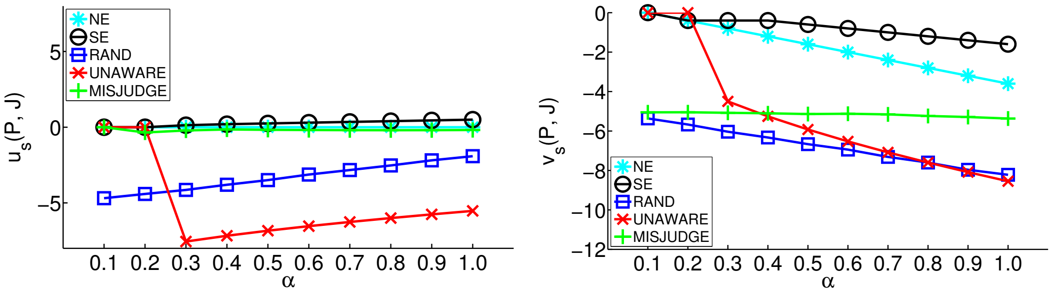
We compare the SE of the PCSJ game with the following scenarios:

- Power Control with Standard Jammer (NE) [165]: The jammer set its power without knowing the user's. Thus both the user and the jammer set their power simultaneously.
- Random Power Control (RAND): Both the user and the jammer randomly set their power, regardless of the existence of the other, as long as the power allocation is feasible.
- Power Control While Being Unaware of Jammer's Existence (UNAWARE): The user maximizes its utility, without the knowledge of the smart jammer's existence. The smart jammer still maximizes it utility with its intelligence.
- Power Control with Misjudgement (MISJUDGE): The user assumes the intelligence of the jammer, while the jammer is just a regular one using random transmission power.

Result Analysis

Figs. 7.3 and 7.4 show the results of the single-channel model. Specifically, Figs. 7.3(a) and 7.3(b) show the impact of α on the players' utility values with $\beta = 0.5$, for different scenarios. We observe that SE leads to the highest utility values for the user. The fact that the utility at SE is higher than that at NE is consistent with the results in [165]. Recall that α is the fading channel gain of the user. Therefore the larger α is, the closer the transmitter is from the receiver. Hence, as α increases, user's utility increases, as shown in Figure 7.3(a), while jammer's utility decreases, as shown in Figure 7.3(b). For the user, both NE and MISJUDGE result in higher utility than both RAND and UNAWARE. It is because the user prepares for the worst case where the jammer has intelligence. In RAND, the user randomly sets its power, which results in a negative utility when $\alpha = 0.1, 0.2$ even without the jammer. Therefore, the utility of the user in RAND is lower than that in UNAWARE when $\alpha = 0.1, 0.2$. However, when $\alpha > 0.2$, the user's utility in RAND is always higher than that in UNAWARE, due to the unawareness of the jammer's existence in UNAWARE. For the jammer, SE leads to the highest utility. Again, the higher utility in SE compared to NE is consistent with the results in [165]. Compare to RAND, jammer's utility is higher in UNAWARE where jammer has intelligence. In addition, MISJUDGE is higher than RAND, because the user assumes the existence of the jammer in MISJUDGE, but sets power randomly in RAND. Another observation is that MISJUDGE results in higher utility than UNAWARE when $\alpha \geq 0.4$. It is because good channel condition (i.e. large value of α) makes the user transmit with the maximum power in UNAWARE.

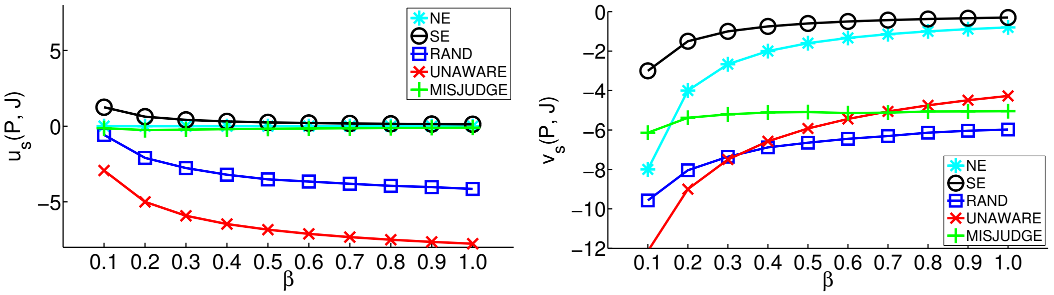
Figs. 7.4(a) and 7.4(b) show the impact of β on the players' utility values with $\alpha = 0.5$, for different scenarios. Again, SE leads to the highest utility values for the user. Note that the jammer's SE utility increases while the user's SE utility decreases due to the fact that the jammer's influence on the receiver gets stronger as β increases. We also have the similar observations about the relationship among different scenarios as in Figs. 7.3 and



(a) User's utility (b) Jammer's utility

Figure 7.3: Impact of α on players' utility values

7.4.



(a) User's utility (b) Jammer's utility

Figure 7.4: Impact of β on players' utility values

Figs. 7.5 through 7.8 show the results of the multi-channel model. For SE-SA, three parameters I, T, and σ need to be decided. Figure 7.5 shows $u_m(\boldsymbol{P},\boldsymbol{J}(\boldsymbol{P}))$ as a function of the number of iterations with our parameter settings. Plugging I, T, and σ into Algorithm 12, we know that there are $\lceil \log_\sigma \frac{1}{T} \rceil * I = 10000$ iterations in total. We observe that the algorithm stops making improvement after 3000 iterations.

Figure 7.6 shows the comparison between SE-SA and SE-MESH. In particular, Figure 7.6(a) shows the user's utility and Figure 7.6(b) shows the running time. Although SE-MESH performs a little better than SE-SA, the running time of SE-MESH grows exponentially in $1/\varepsilon$. Actually, the running time of SE-MESH is dominated by the number

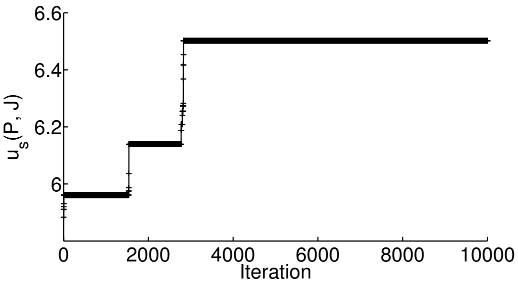
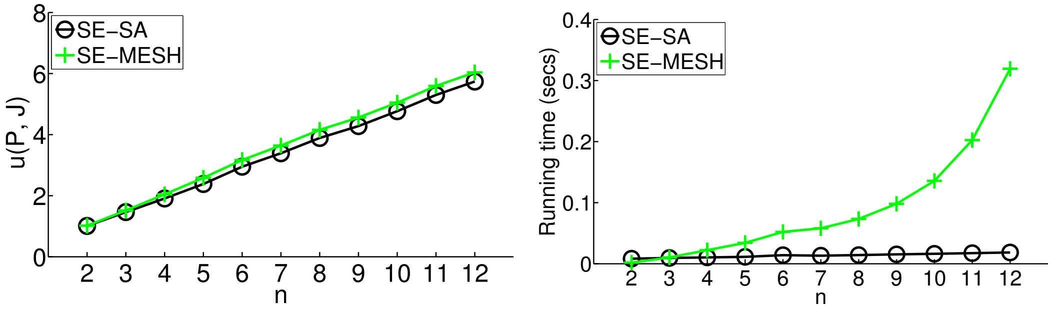


Figure 7.5: Convergence of the simulated annealing algorithm

of mesh points we evaluate in Line 3 in Algorithm 13, which is $\Theta((n+1)^{\hat{P}/\varepsilon})$. To have a better idea on how this scales, we plot it as a function of n and ε in Figure 7.7. As we can see, when $n = 12$, the number of mesh points is more than 10^{100} for $\varepsilon = 0.1$, which is beyond the computability of current PC machines. Hence SE-SA is the recommended approach.



(a) User's utility

(b) Running time

Figure 7.6: Comparison between SE-SA and SE-MESH

Figure 7.8 shows the impact of n on players' utility values under the multi-channel model. We observe that SE has the best performance for the user, followed by RAND at the second and UNAWARE at the bottom. In general, the user's utility increases when there are more channels. The reason is that the user has a better chance to allocate power to channels with better channel gains, i.e., α_i. Another observation is that the jammer has lower utility values in UNAWARE than SE when there are less than 7 channels, while

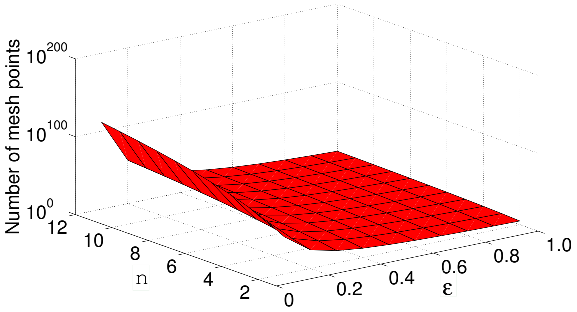
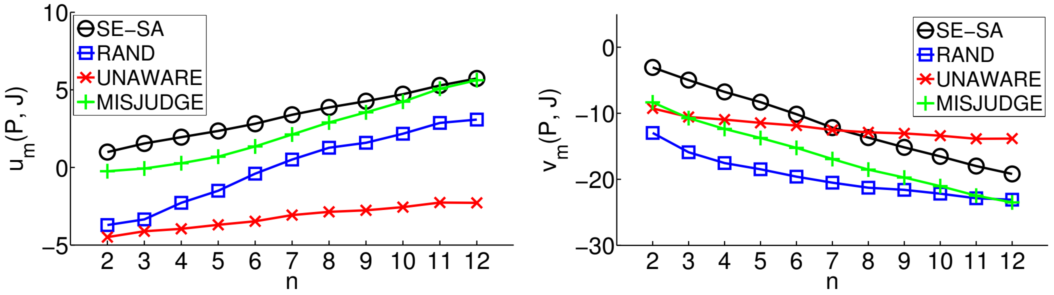


Figure 7.7: Scalability of SE-MESH in log-scale

higher utility values when there are more than 7 channels. This is because the number of channel does not affect the jammer's utility as much in UNAWARE as it does in SE. The user will always only use the channel(s) with the best channel gain(s) if it is unaware of the smart jammer's existence. In contrast, the user has more flexible where there are more channels in SE, with the knowledge of the jammer's intelligence. Regarding the user's utility in MISJUDGE compared to other scenarios, we have the similar observations as in the single-channel model. For the jammer, it has higher utility in UNAWARE than it does in MISJUDGE. It is because unlike the single-channel model, where the user can transmit with the maximum power at one channel, the user allocates power equally to channels with the same condition under the multi-channel in UNAWARE. The even power distribution allows the jammer to attack the channels with better channel conditions for the jammer and thus to improve its utility. This advantage of the jammer is enhanced when the number of channels increases.

7.7 Conclusion

In this paper, we have studied the problem of optimal power control in the presence of a smart jammer, who can quickly learn the transmission power of the user and adjust its transmission power to maximize the damaging effect. We have considered both the single-



(a) User's utility

(b) Jammer's utility

Figure 7.8: Impact of *n* on players' utility values

channel and the multi-channel models. We modeled the problem as a Stackelberg game, called PCSJ game. For the single-channel model, we proved the existence and the uniqueness by giving the closed-form expressions for the Stackelberg Equilibrium (SE). For the multi-channel model, we proved the existence and designed algorithms for approximating an SE.

Chapter 8

Motivating Mobile Users for K-Anonymity Location Privacy

The past few years have witnessed a surge of location-based services (LBSs), including Foursquare [43], Google Latitude [56], and Where [147]. LBS is an information and entertainment service based on the geographical position of mobile devices. Take Foursquare, one of the most popular LBSs, as an example. Users can check in at the participating businesses in exchange for coupons or gaming rewards such as badges and mayorships. A recent report finds that 74% of smartphone owners use LBSs as of February 2012, up from 55% in May 2011 [112]. With this proliferation comes a serious concern about location privacy of mobile users. Beresford and Stajano [10] defined *location privacy* as *the ability to prevent other parties from learning one's current or past location*. LBS providers collect information not only about where we go but what we do, who we know, and who we are. As the Electronic Privacy Information Center reports [39], a third party may have access to users' location history without their explicit consents.

Several approaches have been proposed to protect the location privacy of mobile users [81]. One of the most widely adopted approaches is k-anonymity [61]. The basic idea of k-anonymity is to have a trusted proxy relay the communication between mobile users and LBS providers. After receiving the location information from a mobile user, the proxy adjusts the resolution of the location by returning a cloaking area containing at least $k-1$ other mobile users. As a result, the adversary cannot distinguish one mobile user from any other mobile user in the same cloaking area. Meanwhile, Beresford and Stajano [10] proposed to frequently change pseudonyms for each user to protect its location privacy. To prevent the adversary from linking the old and new pseudonyms of a mobile user, they also introduced a new concept, called *mix zone*. A mix zone is a spatiotemporal region, inside which the mobile users change their pseudonyms and do not communicate with LBS providers. Since a pseudonym change by an isolated mobile user can be trivially traced by the adversary, the pseudonym change should occur with other mobile users.

To achieve *k*-anonymity, there must be at least *k* mobile users in the *anonymity set*. However, not all mobile users are seriously concerned about their location privacy [47, 71, 81]. Iachello *et al.* found through a survey [47] that privacy concerns were quite light for people with a mobile, location-sensitive message service. In an interview with 55 people, Kassinen was surprised by the fact that the interviewees were not worried about privacy issues with location-aware services [71]. In [81], Krumm easily convinced over 250 people from their institution to give up their GPS data and learnt that only 20% out of 97 do not want to share their location data outside their institution.

One way to supplement *k*-anonymity is to introduce dummy users [72]. But this approach has many side effects including waste of resources, significant communication overhead, and difficulty in constructing dummy users' movement [10]. Another solution is to make other mobile users inside the mix zone join the anonymity set. However, mix zones induce a cost for mobile users in the anonymity set, because mobile users in a mix zone cannot communicate [45]. In addition, they need to consume their own resource, including CPU computation, memory, and batter power. Therefore it is necessary to provide incentives for the mobile users to participate.

8.1 Introduction

In this work, we design auction-based incentive mechanisms for *k*-anonymity location privacy. It is desirable for an auction to satisfy the following critical properties: 1) Computational Efficiency: the auction can determine the winners and payments in polynomial time; 2) Individual Rationality: each mobile user can expect a non-negative utility by participating in the auction; 3) Budget Balance: the auctioneer can run the auction without deficit, i.e., the payment collected from the winning buyers is at least as large as the payment paid to the winning sellers; and 4) Truthfulness: no mobile user can benefit from cheating about its true valuation on the *k*-anonymity protection or its cost of participation.

The main contributions of this work are as follows:

- We are the first to design incentive mechanisms for motivating mobile users to assist others achieving *k*-anonymity location privacy.
- We first consider the case where all mobile users have the same privacy degree requirement and design an auction-based incentive mechanism.
- We then generalize the problem to the case where users' degree requirements are different and design a corresponding incentive mechanism.
- We also design an incentive mechanism for a more challenging case where mobile users can also cheat about their requirements.
- We rigorously prove that these incentive mechanisms are computationally efficient, individually rational, budget-balanced, and truthful.

The remainder of this work is organized as follows. In Section 8.2, we briefly review the literature on location privacy protection, with the focus on the anonymity-based approaches. In Section 8.3, we introduce *k*-anonymity and mix-zone location privacy protection techniques, formulate the incentive mechanism design as an auction, and present the desirable properties that an auction should possess. In Section 8.4, we design an incentive mechanism for users with the same privacy degree requirement and an incentive mechanism for users with different requirements and then another incentive mechanism to cope with the scenario where users can cheat about their requirements. In Section 8.5, we evaluate the designed incentive mechanisms through extensive simulations. We conclude this work in Section 8.6.

8.2 Related Work

Location privacy concerns arise as more and more people give away their location information to location-based service providers, without being aware of the risks if location data leaks to an unscrupulous third party. There are many countermeasures proposed to enhance location privacy. They can be generally classified into four categories: *regulatory strategies*, *privacy policies*, *anonymity*, and *obfuscation* [81]. Since the objective of this work is to design incentive mechanisms for anonymity-based approaches, we mainly focus on the related work on anonymity-based location privacy protection.

Inspired by the concept of k-anonymity for data privacy, Gruteser and Grunwald [61] introduced k-anonymity for location privacy. Instead of its real location, each mobile user reports a *cloaking area*, which includes the mobile user itself and at least $k-1$ other mobile users. To prevent the adversary from inferring the identity by learning the pattern of a user's activity, Beresford and Stajano [10] proposed to frequently change the pseudonyms assigned to each user. They further introduced the concept of *mix zone* to make the old and new pseudonyms unlinkable. In [72], Kido *et al.* proposed a new anonymous communication technique to protect the location privacy of users. The idea is that each mobile user generates dummy users and sends them along with its true location to the LBS provider. This approach introduces significant communication overhead. In addition, it is difficult and costly for users to construct dummies with realistic behaviors. In [53], Gedik and Liu designed a framework and algorithms to allow mobile users to specify their degree requirements. Chow *et al.* [22] developed the first distributed algorithm for k-anonymity location privacy. Xu and Cai [153] extended *k*-anonymity to continuous LBS and proposed to use entropy to measure the anonymity degree.

Since the size, shape, and location of the mix zone or the cloaking area directly affect the size of the anonymity set and thus the degree of anonymity, there have been considerable efforts on developing techniques and algorithms to enhance location privacy [45, 88, 139, 144]. In [45], Freudiger *et al.* studied the mix zone placement problem to maximize the achieved location privacy, under cost constraint. In [88], Liu *et al.* proposed a new metric to quantify the system's resilience to privacy attacks and designed heuristic algorithms to deploy mix zones to maximize the new metric. In [139], Vu *et al.* designed a mechanism based on locality-sensitive hashing to compute the cloaking areas, which preserves both locality and *k*-anonymity. In [144], Wang *et al.* introduced the concept of *Location-aware Location Privacy Protection* and developed algorithms to compute cloaking areas with minimum sizes such that the privacy requirements of users are satisfied.

Incentive mechanisms have been adopted to provide incentives for mobile users to participate in different protocols and schemes at the cost of their own resources, including P2P networks [55], routing [143, 148], cooperative communication [157], and mobile sensing [38, 163]. To the best of our knowledge, this work is the first to design incentive mechanisms for *k*-anonymity-based location privacy protection.

8.3 System Model

K-Anonymity and Mix Zone: A straightforward method to protect location privacy is to hide the real identity and associate the location data with a pseudonym. To further reduce the probability of being identified by an adversary, Beresford and Stajano [10] proposed to have each user frequently change its pseudonyms. However they also pointed out that if the spatial and temporal information is sufficient, an adversary can still link the old and new pseudonyms, which defeats the purpose of the frequent pseudonym change. Meanwhile, inspired by the concept of k-anonymity for data privacy, Gruteser and Grunwald [61] considered k-anonymity for location privacy. The location information is called *k-anonymous*, if and only if it is indistinguishable from the location information of at least $k-1$ other users. Instead of its real location, the mobile user reports a two-dimensional area, called *cloaking area*, which contains the mobile user itself and at least $k-1$ other mobile users.

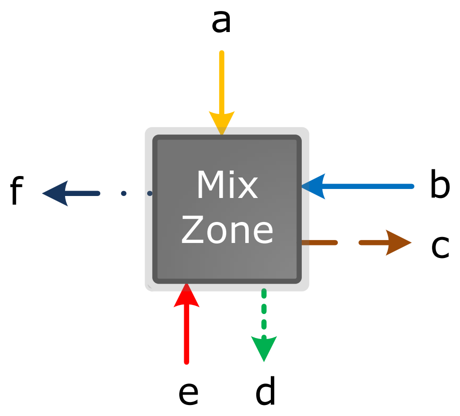


Figure 8.1: Mix zone: after entering mix zone, a, b, and e change their pseudonyms such that an adversary can observe only c, d, and f existing the mix zone.

Combining the ideas of frequently changing pseudonyms and k-anonymity, Beresford and Stajano [10] introduced the concept of *mix zone*, as shown in Figure 8.1. In a mix zone, users receive new, unused pseudonyms and do not report their location information. These users together form an *anonymity set*. In this case an adversary cannot distinguish a user from any other user, and thus cannot link users entering the mix zone with those exiting. There are several techniques to create such a mix zone: 1) Turning off the transceivers of mobile nodes; 2) Encryption; 3) Using a proxy to relay communication; and 4) Exploiting regions that are out of the adversary's coverage [45].

Auction Formulation: Assume that there are a set $\mathscr{U}^b = \{U_1^b, U_2^b, \dots, U_n^b\}$ of $n \geq 1$ mobile users concerned about their location privacy and a set $\mathscr{U}^s = \{U_1^s, U_2^s, \ldots, U_m^s\}$ of $m \geq 1$ mobile users interested in joining the anonymity set. Each mobile user $U_i^b \in \mathscr{U}^b$ desires k_i-anonymity and has a valuation $v_i \geq 0$ on the location privacy protection. Here k_i is the *location privacy degree requirement* of U_i^b, which is the requirement of U_i^b on the size of the anonymity set. Each seller $U_j^s \in \mathscr{U}^s$ has a cost $c_j \geq 0$ of participating in the anonymity set, which is for example related to the energy consumption, data usage, and temporary service connection lost inside the mix-zone.

We model the k-anonymity auction as a *single-round sealed-bid double auction*. In this auction, users in $\mathscr{U}^b$ are *buyers*, users in $\mathscr{U}^s$ are *sellers*, and the central authority is the *auctioneer*, e.g., the middleware proposed in [10] and the wireless carrier in cellular networks. For ease of exposition, we refer both buyers and sellers as *agents* when we do not distinguish them specifically. Buyers offer prices for the desired k-anonymity privacy, while sellers offer prices for participating in the anonymity set. Following the terminology in auction theory, the price offered by a buyer is referred as *bid*, and the price offered by a seller is referred as *ask*. Since the auction is sealed-bid, the price offered by each agent is private to the agent itself, and no agent is aware of the prices offered by others. Let $b_i \geq 0$ denote the bid of U_i^b. Note that b_i is not necessarily equal to v_i. Let $a_j \geq 0$ denote the ask of U_j^s. Similarly, a_j is not necessarily equal to c_j. At the beginning of the auction, buyers submit their bids and sellers submit their asks to the auctioneer. Given as input the bids from the buyers and the asks from the sellers, the auctioneer determines the winning buyer set $\mathscr{W}^b$ and the winning seller set $\mathscr{W}^s$, such that $|\mathscr{W}^b| + |\mathscr{W}^s| \geq k_i$ for each $U_i^b \in \mathscr{W}^b$. In addition, the auctioneer decides the payment p_i^b charged to each buyer $U_i^b \in \mathscr{U}^b$ and the payment p_j^s paid to each seller $U_j^s \in \mathscr{U}^s$. Obviously, $p_i^b = 0$ for each $U_i^b \in \mathscr{U}^b \setminus \mathscr{W}^b$ and

$p^s_j = 0$ for each $U^s_j \in \mathscr{U}^s \setminus \mathscr{W}^s$. The *utility* of buyer U^b_i is defined as

$$u^b_i = \begin{cases} v_i - p^b_i, & \text{if } U^b_i \in \mathscr{W}^b, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, the utility of seller U^s_j is defined as

$$u^s_j = \begin{cases} p^s_j - c_j, & \text{if } U^s_j \in \mathscr{W}^s, \\ 0, & \text{otherwise.} \end{cases}$$

We assume that agents are rational and try to maximize their own utilities.

8.4 Auction Design

Auction with Same Degree Requirement

In this section, we assume that the privacy degree requirements of all the buyers are the same, i.e., $k_i = k$ for all $U^b_i \in \mathscr{U}^b$. We design an auction, called *KASD* (*K*-Anonymity Auction with Same Degree Requirement).

Algorithm 14: *KASD*

```
$p^b_i \leftarrow 0$ for $i \leftarrow 1$ to $n$; $p^s_j \leftarrow 0$ for $j \leftarrow 1$ to $m$;
if $n + m \geq k + 2$ then
    if $n \geq k + 1$ then
        $\mathscr{W}^b \leftarrow \{U^b_1, U^b_2, \ldots, U^b_{n-1}\}$;
        $p^b_i \leftarrow b_n$ for $i \leftarrow 1$ to $n - 1$;
    else if $a_{k-n+2} \leq \frac{(n-1)b_n}{k-n+1}$ then
        $\mathscr{W}^b \leftarrow \{U^b_1, U^b_2, \ldots, U^b_{n-1}\}$;
        $p^b_i \leftarrow b_n$ for $i \leftarrow 1$ to $n - 1$;
        $\mathscr{W}^s \leftarrow \{U^s_1, U^s_2, \ldots, U^s_{k-n+1}\}$;
        $p^s_j \leftarrow a_{k-n+2}$ for $j \leftarrow 1$ to $k - n + 1$;
    end
end
```

The pseudocode of *KASD* is illustrated in Algorithm 14. For ease of exposition, we assume that $b_1 \geq b_2 \geq \cdots \geq b_n$ and $a_1 \leq a_2 \leq \cdots \leq a_m$. The auctioneer first checks whether the total number of buyers and sellers is at least $k+2$. Here we need two more agents than the degree requirement, because one of the buyers and one of the sellers need to be sacrificed to guarantee the truthfulness. If the number of buyers is more than k, the buyers themselves can achieve k-anonymity without the help of the sellers. Otherwise, the auctioneer checks whether it is possible to find $k-n+1$ sellers as winners. In either case, the first $n-1$ buyers become the winning buyers and each of them pays a payment of b_n. In the latter case, the first $k-n+1$ sellers become the winning sellers and each of them receives a payment of a_{k-n+2}.

We now prove that *KASD* satisfies four properties mentioned in Section 1.1. Specifically, we prove that *KASD* is computationally efficient (Lemma 8.1), individually rational (Lemma 8.2), budget-balanced (Lemma 8.3), and truthful (Lemmas 8.4 and 8.5).

Lemma 8.1. KASD *is computationally efficient.* □

Proof. To ease the exposition, we have assumed that buyers are sorted in a nonincreasing order according to their bids and sellers are sorted in a nondecreasing order according to their asks. Sorting the buyers and the sellers takes $O(n\log n+m\log m)$ time. The rest of *KASD* takes linear time. Hence *KASD* takes $O(n\log n+m\log m)$ time. □

Lemma 8.2. KASD *is individually rational.* □

Proof. Each winning buyer $U_i^b \in \mathscr{W}^b$ pays $p_i^b = b_{min}(\mathscr{U}^b)$. When it bids truthfully, i.e., $b_i = v_i$, its utility is $u_i^b = v_i - p_i^b \geq v_i - b_i = 0$. Each winning seller $U_j^s \in \mathscr{W}^s$ receives $p_j^s = a_{k-n+2}$. When it asks truthfully, i.e., $a_j = c_j$, its utility is $u_j^s = p_j^s - c_j \geq a_j - c_j = 0$. Therefore *KASD* is individually rational for both buyers and sellers. □

Lemma 8.3. KASD *is budget-balanced.* □

Proof. If no winner is selected, the auctioneer does not need to pay any seller. If $n \geq k+1$, the total payment collected from the winning buyers is $p^b = \sum_{i=1}^{n-1} p_i^b = (n-1)b_n \geq 0$. Since there is no winning seller, the profit of the auction is nonnegative. If $n < k+1$, the total payment collected from the winning buyers is still $p^b = (n-1)b_n$. The total payment paid to the winning sellers is $p^s = (k-n+1)a_{k-n+2}$. The profit of the auctioneer is $p^b - p^s = (n-1)b_n - (k-n+1)a_{k-n+2} \geq (n-1)b_n - (k-n+1)\frac{(n-1)b_n}{k-n+1} = 0$. □

Lemma 8.4. KASD *is truthful for the buyers.* □

Proof. For any buyer U_i^b, let p_i^b and u_i^b be its payment and utility, respectively, when it bids truthfully, i.e., $b_i = v_i$. Let $\tilde{p}_i^b$ and $\tilde{u}_i^b$ be its payment and utility, respectively, when it cheats, i.e., $b_i \neq v_i$. Throughout the proof, we assume that U_i^b is in front of others after the tie-breaking to simplify the description. We prove that $u_i^b \geq \tilde{u}_i^b$ for any $b_i \neq v_i$. Due to the assumption that the buyers are sorted, we have $b_{min}(\mathscr{U}^b) = b_n$ and $U_{min}^b(\mathscr{U}^b) = U_n^b$. Assume that $n+m \geq k+2$. Consider the following two cases:

- Case 1: $n \geq k+1$

  There are two possible outcomes when U_i^b bids truthfully, *winning* or *losing*. If it wins, we have $p_i^b = b_{min}(\mathscr{U}^b)$ and $u_i^b = v_i - p_i^b \geq 0$ by Lemma 8.2. Note that bidding $b_i > v_i$ will not affect the outcome of the auction and thus result in $\tilde{u}_i^b = v_i - \tilde{p}_i^b = v_i - p_i^b = u_i^b$. We have the same conclusion when U_i^b bids $b_{min}(\mathscr{U}^b \setminus \{U_i^b\}) \leq b_i < v_i$. By bidding $b_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, U_i^b will become the only loser, which makes its utility $\tilde{u}_i^b = 0 \leq u_i^b$. If it loses by bidding truthfully, its utility is $u_i^b = 0$. This also implies that $v_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$. It is obvious that bidding $b_i < v_i$ or $v_i < b_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$ will not affect the outcome of the auction. By bidding $b_i \geq b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, U_i^b can become a winner. However, its utility becomes $\tilde{u}_i^b = v_i - p_i^b = v_i - b_{min}(\mathscr{U}^b \setminus \{U_i^b\}) < 0 = u_i^b$.

- Case 2: $n < k+1$

  If U_i^b wins by bidding its true valuation, we have a similar argument as in Case 1. If it loses by bidding its true valuation, there could be two subcases. Subcase 1: $v_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$. We then have a similar argument as in Case 1. Subcase 2: $v_i \geq b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, but $a_{k-n+2} > \frac{(n-1)b_n}{k-n+1}$. In this case, bidding $b_i \geq b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$ will not affect the outcome of the auction. If it bids $b_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, the condition (Line 6) will not hold either. It still loses the auction, and its utility is $\tilde{u}_i^b = 0$.

Based on the analyses of Cases 1 and 2, we conclude that U_i^b maximizes its utility by bidding truthfully. □

Lemma 8.5. KASD *is truthful for the sellers.* □

Proof. For any seller U_j^s, let p_j^s and u_j^s be its payment and utility, respectively, when it asks truthfully, i.e., $a_j = c_j$. Let $\tilde{p}_j^s$ and $\tilde{u}_j^s$ be its payment and utility, respectively, when it cheats, i.e., $a_j \neq c_j$. Throughout the proof, we assume that U_j^s is in front of others after the tie-breaking to simplify the description. We prove that $u_j^s \geq \tilde{u}_j^s$ for any $a_j \neq c_j$. Note that sellers will be involved in the auction only when $n < k+1$.

If U_j^s wins by asking its true cost, we have $p_j^s = a_{k-n+2}$ and $u_j^s = p_j^s - c_j \geq 0$ by Lemma 8.2. It is obvious that asking $a_j < c_j$ or $c \leq a_j \leq a_{k-n+2}$ will not affect the outcome of the auction. By asking $a_j > a_{k-n+2}$, U_j^s will become a loser, and its utility is $\tilde{u}_j^s = 0 \leq u_j^s$.

If U_j^s loses by asking its true cost, its utility is $u_j^s = 0$. There could be two subcases. Subcase 1: $c_j > a_{k-n+2}$. In this case, it is obvious that asking $a_j > c_j$ or $a_{k-n+2} < a_j < c_j$ will not affect the outcome of the auction. By asking $a_j \leq a_{k-n+2}$, U_j^s may become a winner, but its utility is $\tilde{u}_j^s = \tilde{p}_j^s - c_j = a_{k-n+2} - c_j < 0 = u_j^s$. Subcase 2: $c_j \leq a_{k-n+2}$, but $a_{k-n+2} > \frac{(n-1)b_n}{k-n+1}$. In this case, asking $a_j \leq a_{k-n+2}$ will not affect the outcome of the auction. If it asks $a_j > a_{k-n+2}$, the condition (Line 6) will not hold either. It still loses the auction, and its utility is $\tilde{u}_i^s = 0 = u_j^s$. □

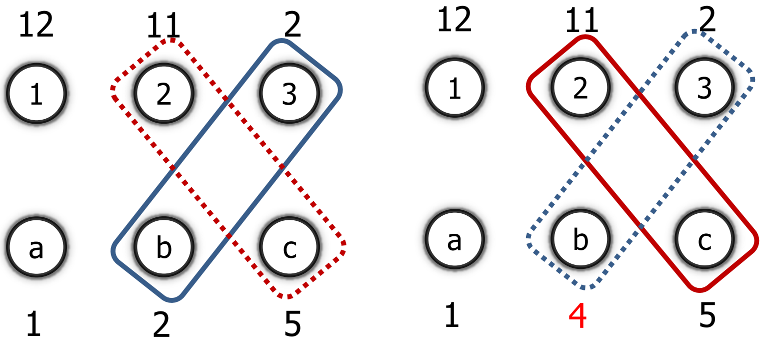
We are ready to give the main theorem in this section.

Theorem 8.1. KASD *is computationally efficient, individually rational, budget balanced, and truthful.* □

Proof. Lemmas 8.1–8.5 together prove this theorem. □

At first sight of *KASD*, one may argue that two possible changes can improve the outcome of the auction, in terms of the auctioneer's profit and the number of total winners. First, instead of only checking whether $a_{k-n+2} \leq \frac{(n-1)b_n}{k-n+1}$ is satisfied, we keep searching for the value of l, $0 \leq l \leq \min\{n-2, m-k+n-2\}$, such that the auctioneer's profit is maximized under the constraint $a_{k-n+l+2} \leq \frac{(n-l-1)b_{n-l}}{k-n+l+1}$. Second, if the condition (Line 6) is not satisfied, we should keep searching for the value of l until $a_{k-n+l+2} \leq \frac{(n-l-1)b_{n-l}}{k-n+l+1}$. Unfortunately, making any of these two changes to *KASD* can devastate its truthfulness, as shown in the following examples.

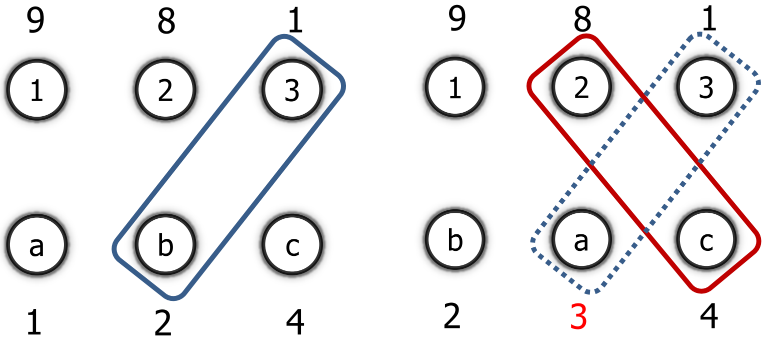
Assume we make the first change. An example is given in Figure 8.2. In this example, there are 3 buyers $\{1,2,3\}$ and 3 sellers $\{a,b,c\}$. In addition, $v_1 = 12$, $v_2 = 11$, $v_3 = 2$, $c_a = 1$, $c_b = 2$, and $c_c = 5$. Assume that $k = 3$. When all agents report truthfully, as shown in Figure 8.2(a), we compare two pairs $(3,b)$ and $(2,c)$. We choose the pair $(3,b)$, since it results in a profit of 2 compared to a profit of 1 by choosing pair $(2,c)$. Thus seller b is one of the losers, and its utility is 0. Now assume that seller b asks 4 instead of 2, as shown in Figure 8.2(b). In this case, we will choose the pair $(2,c)$, since it results in a profit of 1 compared to a profit of 0 by choosing pair $(3,b)$. Seller b now becomes a winner and



(a) Agents report truthfully (b) Seller *b* asks 4 instead of 2

Figure 8.2: Profit maximization may devastate the truthfulness

receives a payment of 5. Thus its utility is $5-2=3$. Seller 2 increases its utility from 0 to 3 by cheating about its ask. This shows that the change we made to *KASD* allows agents to manipulate the outcome of the auction.



(a) Agents report truthfully (b) Seller *a* asks 3 instead of 1

Figure 8.3: Increasing the number of winners may devastate the truthfulness

We then examine the second change. A corresponding example is shown in Figure 8.3. In this example, there are 3 buyers $\{1,2,3\}$ and 3 sellers $\{a,b,c\}$. In addition, $v_1=9$, $v_2=8$, $v_3=1$, $c_a=1$, $c_b=2$, and $c_c=4$. Assume that $k=3$. When all agents report truthfully, as shown in Figure 8.3(a), the pair $(3,b)$ already satisfies the condition (Line 6). As a result, seller *a* is the only winner and receives a payment of 2. Thus its utility is $2-1=1$. Now assume that seller a asks 3 instead of 1, as shown in Figure 8.3(b). The condition (Line 6) does not hold for the pair $(3,a)$. We thus move on to the pair $(2,c)$, which satisfies the condition. As a result, seller a is still a winner, but receives a payment of 4. Its utility becomes $4-1=3$. Seller a increases its utility from 1 to 3 by cheating about its ask. This shows that *KASD* with the above change is not truthful.

The above two examples show that our incentive mechanism *KASD* cannot be trivially improved along the lines discussed above. While it is impossible to satisfy the system efficiency without sacrificing the four properties [104], the design of incentive mechanisms satisfying these four properties but with larger total valuations of the agents remains an important future research topic.

Auction with Different Degree Requirements

In this section, we design an auction for the case where the privacy degree requirements of buyers are different. We call this auction *KADD* (*K*-Anonymity Auction with Different Degree Requirements).

The pseudocode of *KADD* is illustrated in Algorithm 15. For ease of exposition, we assume that $a_1 \leq a_2 \leq \cdots \leq a_m$. The auctioneer divides the buyers into q groups, $\mathscr{U}^b_{\lambda_1}, \mathscr{U}^b_{\lambda_2}, \ldots, \mathscr{U}^b_{\lambda_q}$, according to their degree requirements, such that $\lambda_g < \lambda_{g'}$ for any $1 \leq g < g' \leq q$. For any buyer $U^b_i \in \mathscr{U}^b_{\lambda_g}$, $k_i = \lambda_g$. The auctioneer first checks how many groups can be self-supporting without the help of sellers. It computes the maximum value of x, $1 \leq x \leq q$, such that $|\cup^x_{g=1} \mathscr{U}^b_{\lambda_g}| \geq \lambda_x + 1$. These self-supporting buyers are always winning buyers, and each of them needs to pay a payment of $b_{min}(\cup^x_{g=1}\mathscr{U}^b_{\lambda_g})$. With the value of x computed, the auctioneer then decides which of the remaining groups should be considered potential winning buyers. The intuition is to select the groups such that the possible payment to each necessary seller is maximized. This selection maximizes the probability of the auction's success. If the payment satisfies the sellers, who are necessary to achieve k_i-anonymity for each potential winning buyer, these potential winning buyers win the auction. Each winning buyer U_i^b pays a payment of $b_{min}(\mathscr{U}_{k_i}^b)$.

If the payment could not satisfy the sellers, they become losers.

If there is a tie while determining $U_{i_x}^b$, the buyer with the minimum ID would be chosen.

We now prove that *KADD* satisfies four properties mentioned in Section 1.1: computational efficiency (Lemma 8.6), individual rationality (Lemma 8.7), budget balance (Lemma 8.8), and truthfulness (Lemmas 8.9 and 8.10).

Lemma 8.6. KADD *is computationally efficient.* □

Proof. The initialization phase (Line 1) can be finished in $O(n+m)$ time. Dividing the buyers into groups and sorting the groups take $O(n\log n)$ time. Sorting the sellers takes $O(m\log m)$ time. Finding the largest x in Line 3 takes $O(n)$ time. If x does not exist, finding y (Line 5) takes $O(n)$ time if we keep track of $b_{min}(\mathscr{U}_{\lambda_g}^b)$ for $1 \leq g \leq q$ while grouping the buyers. Similarly, determining the winning buyers (Line 8) can be done in $O(n)$ time if we keep track of $U_{min}^b(\mathscr{U}_{\lambda_g}^b)$ while grouping the buyers. It is obvious that Lines 10 and 11 can be finished in $O(m)$ time once $a_{\pi+1}$ is known. If $x < q$, the analysis is similar as above. Therefore *KADD* takes $O(n\log n + m\log m)$ time. □

Lemma 8.7. KADD *is individually rational.* □

Proof. Each winning buyer $U_i^b \in \mathscr{W}^b$ pays $p_i^b = b_{min}(\mathscr{U}_{k_i}^b)$ or $p_i^b = b_{min}(\cup_{g=1}^x \mathscr{U}_{\lambda_g}^b)$ if it is in the self-supporting groups. When it bids truthfully, i.e., $b_i = v_i$, its utility is $u_i^b = v_i - p_i^b \geq v_i - b_i = 0$. Each winning seller $U_j^s \in \mathscr{W}^s$ receives $p_j^s = a_{\pi+1}$. When it asks truthfully, i.e.,

Algorithm 15: *KADD*

```
$p_i^b \leftarrow 0$ for $i \leftarrow 1$ to $n$; $p_j^s \leftarrow 0$ for $j \leftarrow 1$ to $m$;
Group $\mathscr{U}^b$ into $\mathscr{U}^b_{\lambda_1}, \mathscr{U}^b_{\lambda_2}, \dots \mathscr{U}^b_{\lambda_q}$, such that $\lambda_g < \lambda_{g'}$ for any $1 \le g < g' \le q$, and let
  $n_{\lambda_g} \leftarrow |\mathscr{U}^b_{\lambda_g}|$ for $1 \le g \le q$;
Find the largest $x$ such that $\sum_{g=1}^{x} n_{\lambda_g} \ge \lambda_x + 1$;
if $x$ does not exist then
    $y \leftarrow \arg\max_{1 \le y \le q} \frac{\sum_{g=1}^{y} \left(n_{\lambda_g} - 1\right) b_{min}(\mathscr{U}^b_{\lambda_g})}{\lambda_y - \sum_{g=1}^{y}(n_{\lambda_g} - 1)}$;
    $\pi \leftarrow \lambda_y - \sum_{g=1}^{y}(n_{\lambda_g} - 1)$; $\tau \leftarrow \frac{\sum_{g=1}^{y}\left(n_{\lambda_g} - 1\right) b_{min}(\mathscr{U}^b_{\lambda_g})}{\pi}$;
    if $\pi + 1 \le m$ and $a_{\pi+1} \le \tau$ then
        $\mathscr{W}^b \leftarrow \cup_{g=1}^{y} \left\{ \mathscr{U}^b_{\lambda_g} \setminus \{U^b_{min}(\mathscr{U}^b_{\lambda_g})\} \right\}$;
        $p_i^b \leftarrow b_{min}(\mathscr{U}^b_{\lambda_g})$ for $U_i^b \in \mathscr{U}^b_{\lambda_g} \setminus \{U^b_{min}(\mathscr{U}^b_{\lambda_g})\}$ and $1 \le g \le y$;
        $\mathscr{W}^s \leftarrow \left\{U_1^s, U_2^s, \dots, U_\pi^s\right\}$;
        $p_j^s \leftarrow a_{\pi+1}$ for $j \leftarrow 1$ to $\pi$;
    end
else
    $\mathscr{W}^b \leftarrow \left\{ \cup_{g=1}^{x} \mathscr{U}^b_{\lambda_g} \right\} \setminus \left\{ U^b_{min}(\cup_{g=1}^{x} \mathscr{U}^b_{\lambda_g}) \right\}$;
    $p_i^b \leftarrow b_{min}(\cup_{g=1}^{x} \mathscr{U}^b_{\lambda_g})$ for $U_i^b \in \mathscr{W}^b$;
    if $x < q$ then
        $y \leftarrow \arg\max_{x+1 \le y \le q} \frac{X+Y}{\lambda_y - \left(\sum_{g=1}^{y} n_{\lambda_g} - y + x - 1\right)}$, where
          $X = b_{min}(\cup_{g=1}^{x} \mathscr{U}^b_{\lambda_g}) \left(\sum_{g=1}^{x} n_{\lambda_g} - 1\right)$ and $Y = \sum_{g=x+1}^{y} \left(n_{\lambda_g} - 1\right) b_{min}(\mathscr{U}^b_{\lambda_g})$;
        $\pi \leftarrow \lambda_y - \left(\sum_{g=1}^{y} n_{\lambda_g} - y + x - 1\right)$; $\tau \leftarrow \frac{X+Y}{\pi}$;
        if $\pi + 1 \le m$ and $a_{\pi+1} \le \tau$ then
            $\mathscr{W}^b \leftarrow \mathscr{W}^b \cup \left\{ \cup_{g=x+1}^{y} \mathscr{U}^b_{\lambda_g} \setminus \{U^b_{min}(\mathscr{U}^b_{\lambda_g})\} \right\}$;
            $p_i^b \leftarrow b_{min}(\mathscr{U}^b_{\lambda_g})$ for $U_i^b \in \mathscr{U}^b_{\lambda_g} \setminus \{U^b_{min}(\mathscr{U}^b_{\lambda_g})\}$ and $x + 1 \le g \le y$;
            $\mathscr{W}^s \leftarrow \left\{U_1^s, U_2^s, \dots, U_\pi^s\right\}$;
            $p_j^s \leftarrow a_{\pi+1}$ for $j \leftarrow 1$ to $\pi$;
        end
    end
end
```

$a_j = c_j$, its utility is $u_j^s = p_j^s - c_j \geq a_j - c_j = 0$. Therefore *KADD* is individually rational for both buyers and sellers. □

Lemma 8.8. KADD *is budget-balanced.* □

Proof. When no winner is selected, the proof is trivial. If $x = q$ or $x < q$ but the condition in Line 19 does not hold, the total payment collected from the winning buyers is $p^b = \sum_{U_i^b \in \mathscr{W}^b} p_i^b = \Big(|\cup_{g=1}^{x} \mathscr{U}_{\lambda_g}^b| - 1\Big) b_{min}(\cup_{g=1}^{x} \mathscr{U}_{\lambda_g}^b) \geq 0$. Since there is no winning seller, the profit of the auction is nonnegative. Otherwise, the total payment collected from the winning buyers is $p^b = \pi\tau$. The total payment paid to the winning sellers is $p^s = \pi a_{\pi+1}$. The profit of the auctioneer is $p^b - p^s = \pi(\tau - a_{\pi+1}) \geq 0$. □

Lemma 8.9. KADD *is truthful for the buyers.* □

Proof. For any buyer U_i^b, let p_i^b and u_i^b be its payment and utility, respectively, when it bids truthfully, i.e., $b_i = v_i$. Let $\tilde{p}_i^b$ and $\tilde{u}_i^b$ be its payment and utility, respectively, when it cheats, i.e., $b_i \neq v_i$. Throughout the proof, we assume that U_i^b is in front of others after the tie-breaking to simplify the description. We prove that $u_i^b \geq \tilde{u}_i^b$ for any $b_i \neq v_i$.

We first prove the case where U_i^b is not in any of the self-supporting groups. Assume that U_i^b wins by bidding $b_i = v_i$. By Lemma 8.7, we have $u_i^b \geq 0$. It implies that $v_i \geq b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$. It is clear that bidding $b_i > v_i$ or $b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\}) \leq b_i < v_i$ will not affect the outcome of the auction. Thus we have $\tilde{u}_i^b = u_i^b$. If U_i^b bids $b_i < b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$, it will lose the auction, which makes its utility $\tilde{u}_i^b = 0 \leq u_i^b$.

Assume that U_i^b loses by bidding $b_i = v_i$ and its utility $u_i^b = 0$. We consider two cases $v_i < b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$ and $v_i \geq b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$. For the former case, it is clear that bidding $b_i < v_i$ or $v_i < b_i < b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$ will not affect the outcome of the auction. Thus we have $\tilde{u}_i^b = u_i^b$. Even if U_i^b wins the auction by bidding $b_i \geq b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$, its utility is $\tilde{u}_i^b = v_i - \tilde{p}_i^b = v_i - b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\}) < 0 = u_i^b$. For the latter case, it is clear that bidding $b_i > v_i$ or $b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\}) \leq b_i < v_i$ will not affect the outcome of the auction. Thus we have $\tilde{u}_i^b = u_i^b$. If U_i^b bids $b_i < b_{min}(\mathscr{U}_{k_i}^b \setminus \{U_i^b\})$, it will certainly lose the auction. Thus we have $\tilde{u}_i^b = u_i^b$ as well.

The case where U_i^b is in one of the self-supporting groups can be proved similarly by replacing $\mathscr{U}_{k_i}^b$ with $\cup_{g=1}^{x} \mathscr{U}_{\lambda_g}^b$. Therefore U_i^b maximizes its utility by bidding truthfully. □

Lemma 8.10. KADD *is truthful for the sellers.* □

Proof. For any seller U_j^s, let p_j^s and u_j^s be its payment and utility, respectively, when it asks truthfully, i.e., $a_j = c_j$. Let $\tilde{p}_j^s$ and $\tilde{u}_j^s$ be its payment and utility, respectively, when it cheats, i.e., $a_j \neq c_j$. Throughout the proof, we assume that U_j^s is in front of others after the tie-breaking to simplify the description. We prove that $u_j^s \geq \tilde{u}_j^s$ for any $a_j \neq c_j$. Note that sellers will be involved in the auction only when $\pi + 1 < m$ and $a_{\pi+1} \leq \tau$. Since the first condition is independent of sellers' asks, we focus only on the case where it always holds.

Assume that U_j^s wins the auction by asking $a_j = c_j$. It implies that $c_j \leq a_{\pi+1}$ and $u_j^s \geq 0$ by Lemma 8.7. It is clear that asking $a_j < c_j$ or $c_j < a_j \leq a_{\pi+1}$ will not affect the outcome of the auction. Thus we have $\tilde{u}_j^s = u_j^s$. If U_j^s asks $a_j > a_{\pi+1}$, then U_j^s will be placed at the position after π, which makes U_j^s one of the losers. Thus we have $\tilde{u}_j^s = 0 \leq u_j^s$.

Assume that U_j^s loses the auction by asking $a_j = c_j$. One of two possible reasons is that $a_{\pi+1} > \tau$. If $c_j \leq a_{\pi+1}$, U_j^s cannot win the auction no matter how much its ask is. If $c_j > a_{\pi+1}$, U_j^s may win the auction by asking $a_j < a_{\pi+1}$. However, its payment will be $\tilde{p}_j^s \leq a_{\pi+1}$ and utility will be $\tilde{u}_j^s = \tilde{p}_j^s - c_j \leq a_{\pi+1} - c_j < 0 = u_j^s$. The other reason for U_j^s's lost is that $a_{\pi+1} \leq \tau$ and $c_j > a_{\pi+1}$. In this case, we have a similar argument as in the second case for the first reason. □

Theorem 8.2. KADD *is computationally efficient, individually rational, budget balanced, and truthful.* □

Proof. Lemmas 8.6–8.10 together prove this theorem. □

We next consider a more challenging scenario, where buyers can also cheat about their privacy degree requirements. This type of auction is called *multidimensional private type auction* in auction theory [111]. The multidimensional private type gives the buyers more possibilities to manipulate the outcome of the auction. There are few papers concerning about multidimensional private type auction design [40, 93], and they are all essentially single-sided auctions. To the best of our knowledge, there is no related work on the design of multidimensional private type double auction. The fact that our auction has the constraint on the total number of winning buyers and sellers makes the design even more difficult.

Algorithm 16: KADD$^+$

```
$p_i^b \leftarrow 0$ for $i \leftarrow 1$ to $n$; $p_j^s \leftarrow 0$ for $j \leftarrow 1$ to $m$;
$k_{max} \leftarrow \max_{U_i^b \in \mathscr{U}^b} k_i$;
if $n + m \geq k_{max} + 2$ then
    if $n \geq k_{max} + 1$ then
        $\mathscr{W}^b \leftarrow \{U_1^b, U_2^b, \ldots, U_{n-1}^b\}$;
        $p_i^b \leftarrow b_n$ for $i \leftarrow 1$ to $n - 1$;
    else if $a_{k-n+2} \leq \frac{(n-1)b_n}{k_{max}-n+1}$ then
        $\mathscr{W}^b \leftarrow \{U_1^b, U_2^b, \ldots, U_{n-1}^b\}$;
        $p_i^b \leftarrow b_n$ for $i \leftarrow 1$ to $n - 1$;
        $\mathscr{W}^s \leftarrow \left\{U_1^s, U_2^s, \ldots, U_{k_{max}-n+1}^s\right\}$;
        $p_j^s \leftarrow a_{k_{max}-n+2}$ for $j \leftarrow 1$ to $k_{max} - n + 1$;
    end
end
```

To overcome these difficulties, we need to decouple buyers' degree requirements and bids as much as possible. In addition, we also need to make buyers have little control over the degree requirement considered by the auctioneer. We illustrate our auction designed specifically for this scenario in Algorithm 16, which is called KADD$^+$. In order to weaken the buyers' control over the privacy degree requirement considered by the auctioneer, we select the maximum k_i among the buyers and proceed the auction in a way similar to *KASD*. We assume that buyers are single-minded. For any $U_i^b \in \mathscr{W}^b$, if $|\mathscr{W}^b| + |\mathscr{W}^s| < k_i$, its utility is $u_i^b = 0$.

We now prove the properties of KADD$^+$.

Theorem 8.3. *KADD$^+$ is computationally efficient, individually rational, budget balanced, and truthful.* □

Proof. The computational efficiency, individual rationality, and budget balance of KADD$^+$ can be proved similarly as in Lemmas 8.1 to 8.3. We focus on truthfulness.

For any buyer U_i^b, let p_i^b and u_i^b be its payment and utility, respectively, when it bids truthfully, i.e., $b_i = v_i$. Let $\tilde{p}_i^b$ and $\tilde{u}_i^b$ be its payment and utility, respectively, when it cheats, i.e., $b_i \neq v_i$. Throughout the proof, we assume that U_i^b is in front of others after the tie-breaking to simplify the description. We prove that $u_i^b \geq \tilde{u}_i^b$ for any $b_i \neq v_i$ and $k_i' \neq k_i$.

Assume that U_i^b wins by bidding $b_i = v_i$. We then have $u_i^b \geq 0$, since KADD$^+$ is individually rational. It also implies that $v_i \geq b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$. Obviously, if U_i^b bids $b_i > v_i$ or $b_{min}(\mathscr{U}^b \setminus \{U_i^b\}) \leq b_i < v_i$, the outcome of the auction will not be affected. Thus we have $\tilde{u}_i^b = u_i^b$ in this case. If U_i^b bids $b_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, it will lose the auction and have utility $\tilde{u}_i^b = 0 \leq u_i^b$. If U_i^b reports $k_i' < k_i$ or $k_i < k_i' \leq k_{max}$, the outcome of the auction will not change. If U_i^b reports $k_i' > k_{max}$ and the condition in Line 7 still holds, then $\tilde{p}_i^b = p_i^b$ and thus $\tilde{u}_i^b = u_i^b$. If the condition does not hold, then $\tilde{u}_i^b = 0 \leq u_i^b$.

Assume that U_i^b loses by bidding $b_i = v_i$. We then have $u_i^b = 0$. There are two possible cases: $v_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$ and the condition in Line 7 does not hold. For the former case, bidding $b_i < v_i$ or $v_i < b_i < b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$ will not affect the outcome of the auction. By bidding $b_i \geq b_{min}(\mathscr{U}^b \setminus \{U_i^b\})$, U_i^b may win the auction, but its utility is $\tilde{u}_i^b = v_i - \tilde{p}_i^b < 0 = u_i^b$. In addition, reporting any value of k_i cannot make U_i^b a winner in this case. For the latter case, the only possible way that U_i^b can make the condition hold is to report $k_i' < k_{max}$, under the condition that U_i^b is the only player for whom $k_i = k_{max}$. However, even U_i^b wins the auction, the total winners will be $k_i' < k_i$, which makes $\tilde{u}_i^b = 0 = u_i^b$.

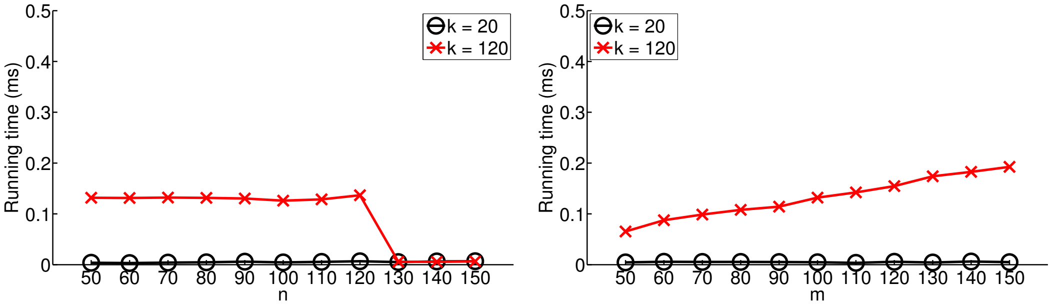
Therefore each buyer maximizes its utility by bidding and reporting degree requirement truthfully. The truthfulness for sellers can be proved similarly as in Lemma 8.5. □

8.5 Performance Evaluation

Simulation Setup

In the simulations, we varied both n and m from 50 to 150 with an increment of 10. The valuation v_i of buyer U_i^b and the cost c_j of seller U_j^s were uniformly distributed over $(0,1]$. For *KASD*, k was varied from 20 to 120 with an increment of 20. For *KADD* and KADD$^+$, k_i was uniformly distributed over $[2,k]$, where k was varied from 20 to 120 with an increment of 20. For each setting we randomly generated 10000 instances and averaged the results. All the simulations were run on a Linux machine with 3.2 GHz CPU and 16 GB memory.

The performance metrics include *running time*, *number of winning buyers*, and *profit of the auctioneer*. The reason we picked the number of winning buyers as a performance metric is that winning buyers are the agents who really desire location privacy after all. For each of these performance metrics, we evaluated the impact of n, m, and k.



(a) Impact of *n*

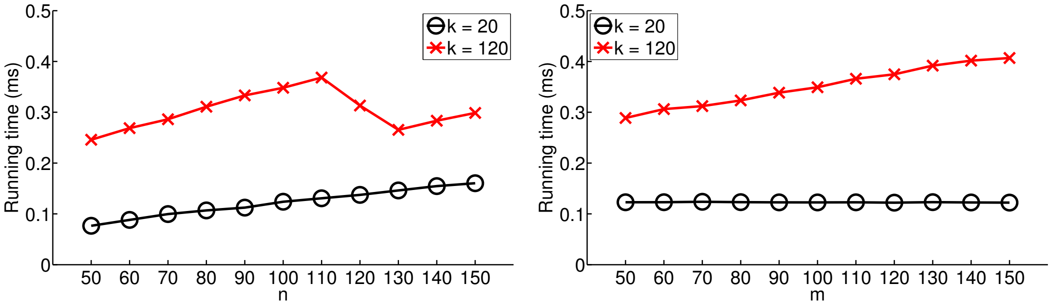
(b) Impact of *m*

Figure 8.4: Running time of *KASD*

While evaluating the impact of n (resp. m), we fixed $m = 100$ (resp. $n = 100$) and chose k to be 20 and 120. While evaluating the impact of k, we fixed $n = 50$ and $m = 100$.

Simulation Results

Running Time: Figure 8.4 and Figure 8.5 plot the running time of *KASD* and *KADD*, respectively. Since KADD$^+$ has very similar curves as *KASD*, we omit the illustration to save space. Specifically, Figure 8.4(a) and Figure 8.5(a) show the impact of n, and Figure 8.4(b) and Figure 8.5(b) show the impact of m. For *KASD*, we observe that the running time of $k = 120$ drops dramatically to the level of $k = 20$ when $n > 120$. This is because the auctioneer needs not consider the sellers to satisfy the winning buyers' degree requirements. Thus the two cases where $k = 20$ and $k = 120$ are essentially equivalent when $n > 120$. When $k = 20$, the auctioneer needs only to spend time determining winning buyers, since n is always greater than k. Thus the auction finishes extremely fast. For *KADD*, we observe that the running time of $k = 120$ starts to drop down when $n = 120$. The reason is that the auctioneer needs not consider the sellers to satisfy all the buyers in some instances. Comparing *KADD* to *KASD*, we note that *KADD* takes more time than *KASD*. This is as expected, because *KADD* has more calculations than *KASD*, although both of them have the same theoretical time complexity.

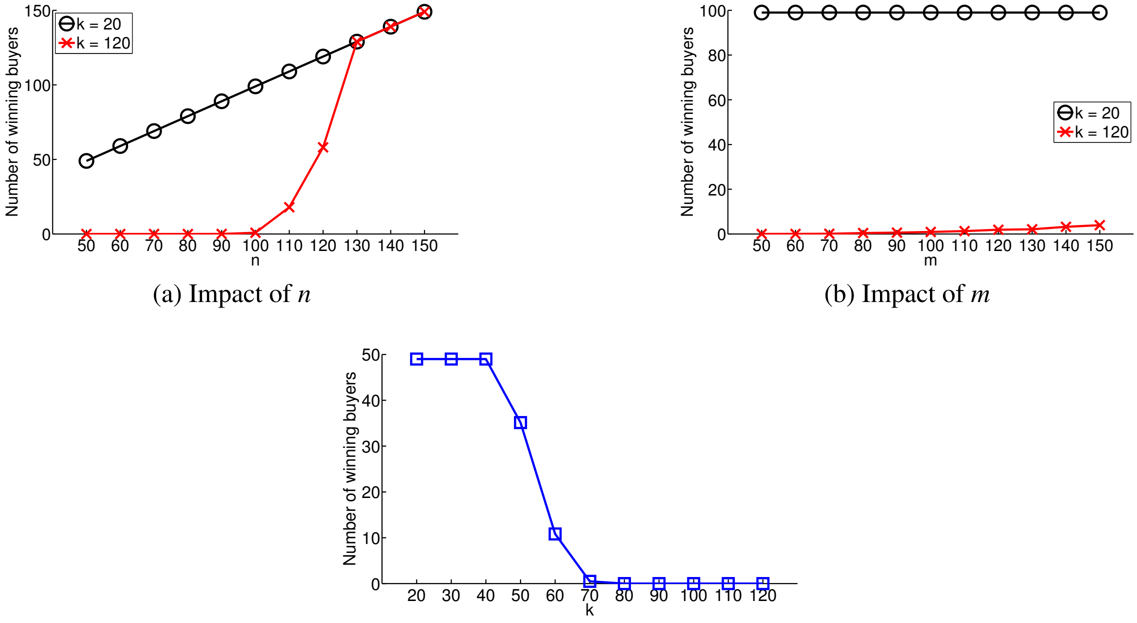


(a) Impact of *n* (b) Impact of *m*

Figure 8.5: Running time of *KADD*

Number of Winning Buyers: Figure 8.6 and Figure 8.7 show the number of winning buyers returned by *KASD* and *KADD*, respectively. One obvious observation is that when $n > k$ the number of winning buyers is always $n-1$. This is consistent with our theoretical analysis. Another observation is that the number of winning buyers increases when there are more sellers participating in the auction. This is because the probability that the conditions (Line 6 in Algorithm 14 and Lines 7 or 19 in Algorithm 15) are satisfied becomes higher. The last observation is that the curves for the impact of k have an upside down S shape. The reason is that the probability that the conditions are satisfied drops dramatically when $k \geq n$.

Profit: Figure 8.8 and Figure 8.9 show the profit of different auctions. In particular, Figure 8.8 shows the profit of *KASD*, and Figure 8.9 shows the profit of *KADD* and KADD$^+$ side by side. For *KASD*, we first observe that the results of the impact of m and the impact of k are consistent with the results in Figure 8.6(b) and Figure 8.6(c). However, the results of the impact of n are quite different from the results in Figure 8.6(a), especially when $k = 20$. The reason is that when n grows, the value of b_n becomes smaller on average, since all the values of b_i are randomly distributed in a fixed range $(0, 1]$. The rate of the decrease in the value of b_n outweighs the increase in the value of $n-1$. For $k = 120$, the profit is

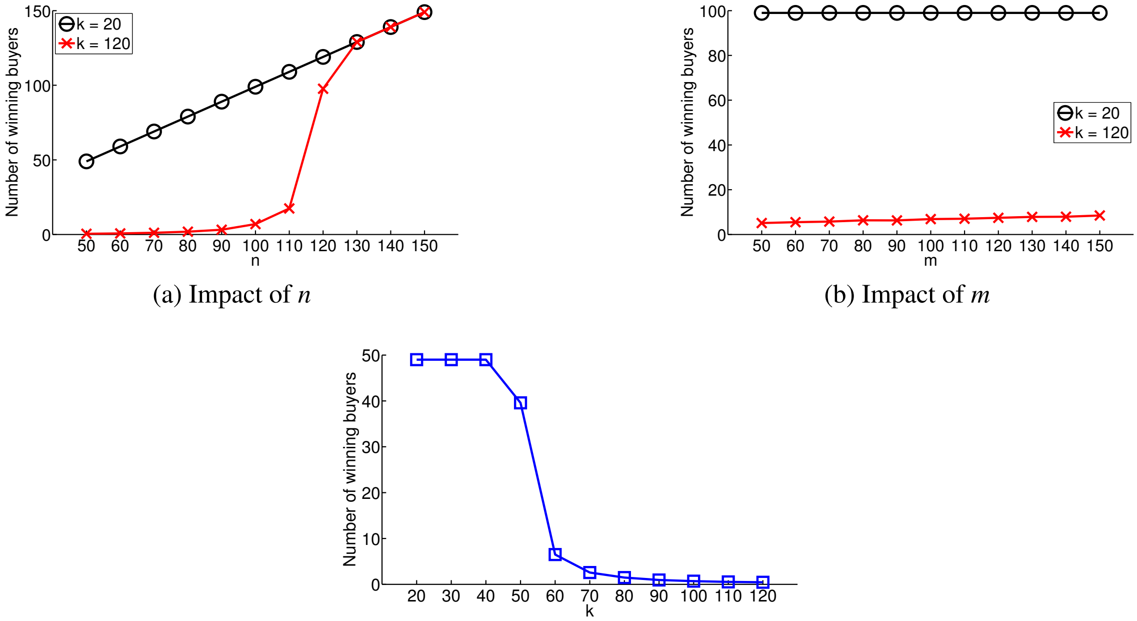


(a) Impact of *n*

(b) Impact of *m*

(c) Impact of *k*

Figure 8.6: Number of winning buyers of *KASD*

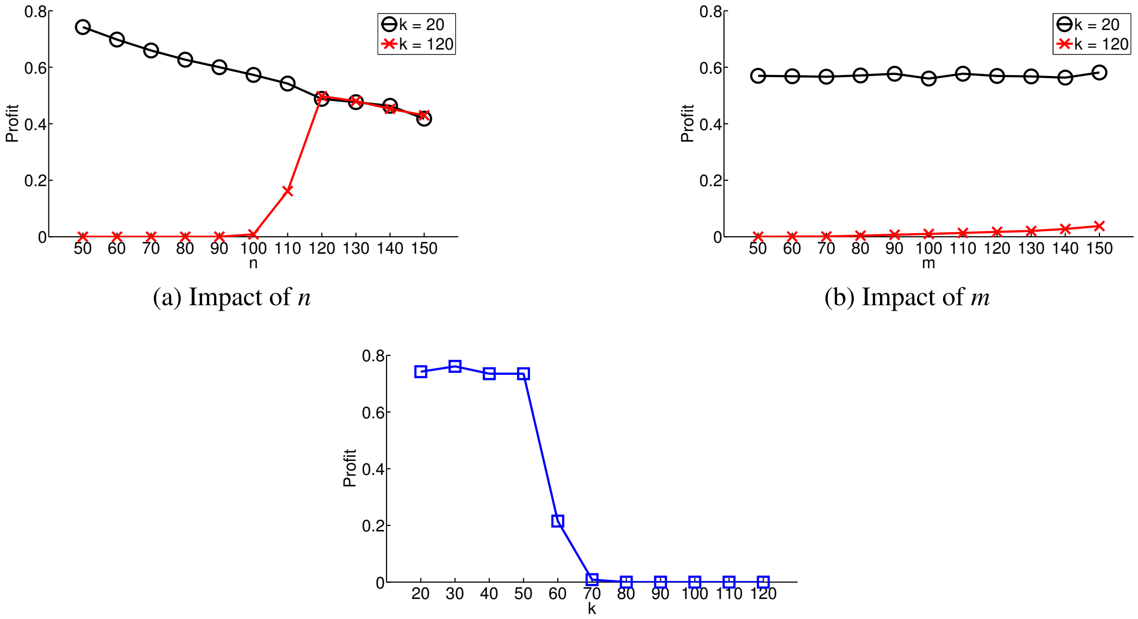


(a) Impact of *n*

(b) Impact of *m*

(c) Impact of *k*

Figure 8.7: Number of winning buyers of *KADD*



(a) Impact of *n*

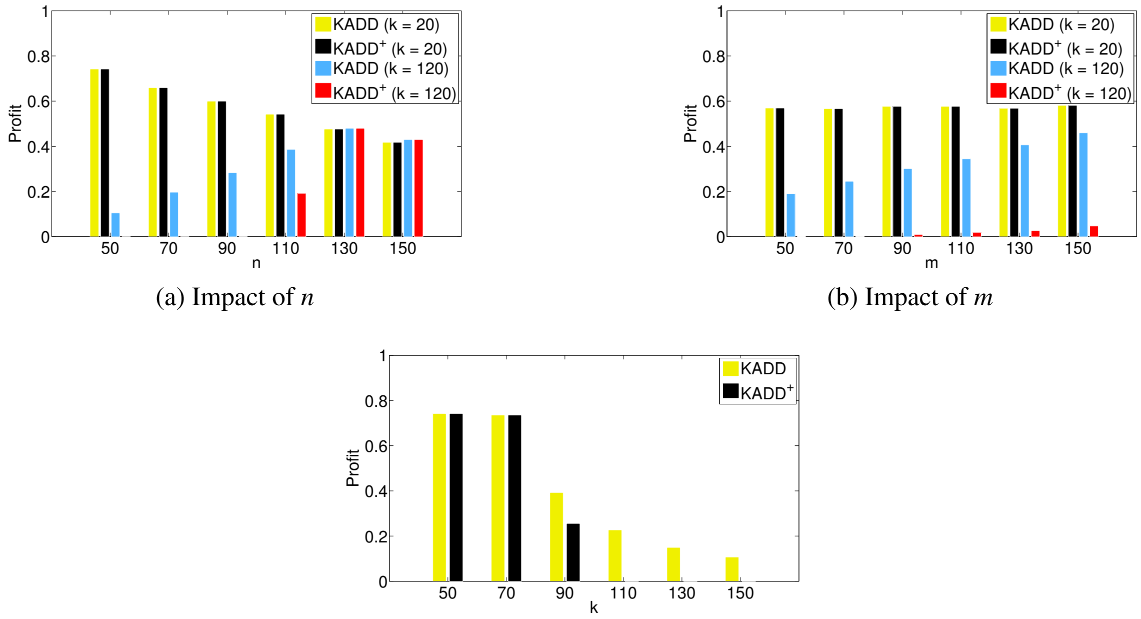
(b) Impact of *m*

(c) Impact of *k*

Figure 8.8: Profit of *KASD*

close to 0 when $n \leq 100$, dramatically increases when $100 < n \leq k$, and then decreases when $n > k$. The reason for the first segment is that the condition (Line 6 in Algorithm 14) can hardly be satisfied. When $100 < n \leq k$, fewer and fewer sellers are needed, but more and more buyers pay their payments. However, when $n > k$, the situation is equivalent to that of $k = 20$.

For the case where buyers have different degree requirements, we compared the profit returned by *KADD* and KADD$^+$. We aimed to evaluate how much profit will be sacrificed in order to guarantee that buyers have no incentives to cheat on k_i. As shown in Figure 8.9, *KADD* and KADD$^+$ almost generate the same profit as long as $n \geq k$. However, when $n < k$, KADD$^+$ barely generates any profit while the profit generated by *KADD* is relatively large and increases with the number of sellers. The reason for this difference is that we consider buyers in a finer granularity for *KADD*.



(a) Impact of *n*

(b) Impact of *m*

(c) Impact of *k*

Figure 8.9: Profit comparison of *KADD* and KADD$^+$

8.6 Conclusion

In this work, we have designed incentive mechanisms for motivating mobile users to assist others achieving *k*-anonymity location privacy. As the first step, we have considered the case where mobile users have the same privacy degree requirement. We have further generalized it to the case where the degree requirements are different. We have also studied a more challenging case where mobile users can cheat about not only their valuations but also their degree requirements. We have designed an auction-based incentive mechanism for each of these cases and proved that all the auctions are computational efficient, individually rational, budget-balanced, and truthful.

Chapter 9

Conclusion

Game theory provides a fruitful set of tools for examining strategic interactions among two or more entities. Network problems can be well modeled and addressed using game theory. Devices in networks, for instance, computers and mobile phones, usually belong to different individuals, who make their decisions independently and selfishly. In this dissertation, we applied game theory to coping with selfish behavior in networks. In general, we studied three problem domains: 1) resource allocation, 2) incentive mechanism, and 3) security.

Specifically, for the domain of resource allocation, we first studied the selfish routing problem in networks with fair queuing on links. We modeled the problem as a game, and proved the existence of Nash Equilibria. In addition, we designed an algorithm following the nature of game course, and rigorously bounded the converge speed to a Nash Equilibrium. The second problem we studied is the relay assignment problem in cooperative networks. We designed an integrated scheme for optimally assigning relays to maximize the total capacity. To avoid system performance degradation due to the selfish relay selections by the source nodes, we propose a payment mechanism for charging the source nodes to induce them to converge to the optimal assignment. To prevent relay nodes from manipulating the relay assignment by reporting transmission power untruthfully, we propose a payment mechanism to pay them for providing relaying service. We also show that HERA is budget-balanced, meaning that the payment collected from source nodes is no smaller than the payment paid to relay nodes. The last problem studied in this domain is the channel allocation problem in non-cooperative networks. We first observed that the network may result in an oscillation, where nodes keep changing their channel allocations back and force trying to improve their utilities. To avoid the possible oscillation, we designed a charging scheme to influence players' behavior. We proved that, under the charging scheme, the network is guaranteed to converge to an NE. We also proved that the system performance in an NE is at least $(1-\frac{\bar{r}}{h})$ of the system performance in the optimal solution, where $\bar{r}$ is the maximum number of radios equipped on wireless devices and h is the number of available channels. Finally, we designed a localized algorithm for players to find an NE.

For the domain of incentive mechanism design, we designed incentive mechanisms for encouraging wireless devices to serve as relays in cooperative networks, and recruiting smartphones for crowdsourcing. For the former problem, we designed an auction-based scheme, which is proved to be efficient, individual-rational, budget-balanced, and, most importantly, truthful. For the latter problem, we considered two system models from two different perspectives: the platform-centric model where the platform provides a fixed reward to participating users, and the user-centric model where users can have their reserve prices for the sensing service. For the platform-centric model, we designed an incentive mechanism using a Stackelberg game. We presented an efficient algorithm to compute the *unique Stackelberg Equilibrium*, at which the utility of the platform is maximized, and none of the users can improve its utility by unilaterally deviating from its current strategy. For the user-centric model, we designed an auction-based incentive mechanism, which is computationally efficient, individually-rational, profitable and, more importantly, truthful.

For the domain of security, we analyzed how a user can defend against a smart jammer, who can quickly learn about the user's transmission power and adaptively adjust its own transmission to maximize the damage. We considered both the single-channel model and the multi-channel model. For the single-channel model, we derived closed-form expressions for the jammer's best response strategy and the user's optimal strategy, which together constitute the unique Stackelberg Equilibrium (SE). For the multi-channel model, we designed an algorithm for computing the jammer's best response strategy, given the user's strategy. We also developed two algorithms to approximate the user's optimal strategy and thus the SE strategies. Another problem we studied is motivating mobile users to participate in k-anonymity location privacy protection. We first considered different cases, including the case where all mobile users have the same privacy degree requirement, the case where users' degree requirements are different, and the case where mobile users can also cheat about their requirements. We carefully design an auction-based incentive mechanism for each case and rigorously proved that these incentive mechanisms are computationally efficient, individually rational, budget-balanced, and truthful.

REFERENCES

[1] M. Alicherry, R. Bhatia, and L. Li. Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks. In *Proc. of MobiCom*, pages 58–72, 2005.

[2] E. Altman, K. Avrachenkov, and A. Garnaev. A jamming game in wireless networks with transmission cost. In *Proc. of NET-COOP*, pages 1–12, 2007.

[3] E. Altman, K. Avrachenkov, and A. Garnaev. Jamming in wireless networks: The case of several jammers. In *Proc. of GameNets*, pages 585–592, 2009.

[4] AT&T. AT&T investor update, 2010. `http://www.att.com/Investor/Financial/Earning_Info/docs/4Q_09_slide_c.pdf`.

[5] L. M. Ausubel. An efficient dynamic auction for heterogeneous commodities. *The American Economic Review*, 96:602–629, June 2006.

[6] S. Avallone and I. Akyildiz. A channel assignment algorithm for multi-radio wireless mesh networks. *Comput. Commun.*, 31(7):1343–1353, May 2008.

[7] M. Babaioff and N. Nisan. Concurrent auctions across the supply chain. In *ACM Conference on Electronic Commerce Proceedings*, pages 1–10. ACM, 2001.

[8] R. Banner and A. Orda. Bottleneck routing games in communication networks. *IEEE J. Sel. Areas Commun.*, 25:1173–1179, Aug. 2007.

[9] J. Behrens and J. Garcia-Luna-Aceves. Distributed, scalable routing based on link-state vectors. In *Proc. ACM SIGCOMM*, pages 136–147, 1994.

[10] A. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[11] D. Bertsekas, R. Gallager, and P. Humblet. *Data networks*. Prentice-hall New York, 1987.

[12] A. Bletsas, A. Khisti, D. Reed, and A. Lippman. A simple cooperative diversity method based on network path selection. *IEEE J. Sel. Areas Commun.*, 24(3):659–672, Mar. 2006.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[14] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[15] BRITE. `http://www.cs.bu.edu/brite/`.

[16] M. Čagalj, S. Ganeriwal, I. Aad, and J. Hubaux. On selfish behavior in csma/ca networks. In *Proc. of INFOCOM*, volume 4, pages 2513–2524, 2005.

[17] J. Cai, X. Shen, J. Mark, and A. Alfa. Semi-distributed user relaying algorithm for amplify-and-forward wireless relay networks. *IEEE Trans. Wireless Commun.*, 7(4):1348–1357, Apr. 2008.

[18] R. Chandra and P. Bahl. MultiNet: Connecting to multiple ieee 802.11 networks using a single wireless card. In *Prof. INFOCOM*, pages 882–893, 2004.

[19] A. Charny, D. Clark, and R. Jain. Congestion control with explicit rate indication. In *Proc. IEEE ICC*, pages 1954–1963, 1995.

[20] N. Chen, N. Gravin, and P. Lu. On the approximability of budget feasible mechanisms. In *Proc. of ACM-SIAM SODA*, pages 685–699, 2011.

[21] S. Chen and K. Nahrstedt. Maxmin fair routing in connection-oriented networks. In *Proc. Euro-Parallel and Distributed Systems Conf.*, 1998.

[22] C.-Y. Chow, M. F. Mokbel, and X. Liu. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In *Proc. of ACM GIS*, pages 171–178, 2006.

[23] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.

[24] CNN Fortune. Industry first: Smartphones pass PCs in sales. `http://tech.fortune.cnn.com/2011/02/07/idc-smartphone-shipment-numbers-passed-pc-in-q4-2010/`.

[25] W. Cook, W. Cunningham, W. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. John Wiley and Sons, 1998.

[26] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver. *Combinatorial Optimization*. Wiley-Interscience, 1 edition, Nov. 1997.

[27] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: making smartphones last longer with code offload. In *Proc. of MobiSys*, pages 49–62, 2010.

[28] G. Danezis, S. Lewis, and R. Anderson. How much is location privacy worth? In *Proc. of WEIS*, 2005.

[29] A. Das, H. Alazemi, R. Vijayakumar, and S. Roy. Optimization models for fixed channel assignment in wireless mesh networks with multiple radios. In *Proc. of SECON*, pages 463–474, 2005.

[30] T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma. PRISM: platform for remote sensing using smartphones. In *Proc. of ACM MobiSys*, pages 63–76, 2010.

[31] C. Daskalakis, P. Goldberg, and C. Papadimitriou. The complexity of computing a nash equilibrium. In *Proc. ACM STOC*, pages 71–78, 2006.

[32] E. De Cristofaro and C. Soriente. Short paper: Pepsi—privacy-enhanced participatory sensing infrastructure. In *Proc. of WiSec*, pages 23–28, 2011.

[33] G. Demange, D. Gale, and M. Sotomayor. Multi-item auctions. *Journal of Political Economy*, 94(4):863–872, 1986.

[34] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *Proc. ACM SIGCOMM*, pages 1–12, 1989.

[35] K. Deshmukh, A. V. Goldberg, J. D. Hartline, and A. R. Karlin. Truthful and competitive double auctions. In *European Symposium on Algorithms Proceedings*, pages 361–373, 2002.

[36] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1:269–271, 1959.

[37] L. Ding, T. Melodia, S. Batalama, and J. Matyjas. Distributed routing, relay selection, and spectrum allocation in cognitive and cooperative ad hoc networks. In *Proc. of IEEE SECON*, pages 1–9, 2010.

[38] L. Duan, T. Kubo, K. Sugiyama, J. Huang, T. Hasegawa, and J. Walrand. Incentive mechanisms for smartphone collaboration in data acquisition and distributed computing. In *Proc. of INFOCOM*, pages 1701–1709, 2012.

[39] Electronic Privacy Information Center. Locational privacy.

[40] H. Fang and S. Morris. Multidimensional private value auctions. *Journal of Economic Theory*, 126(1):1–30, Jan. 2006.

[41] U. Feige, V. S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. *SIAM J. on Computing*, 40(4):1133–1153, 2011.

[42] M. Félegyházi, M. Čagalj, S. Bidokhti, and J. Hubaux. Non-cooperative multi-radio channel allocation in wireless networks. In *Proc. of INFOCOM*, pages 1442–1450, 2007.

[43] Foursquare. https://foursquare.com/.

[44] M. Fredman and R. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *JACM*, 34:596–615, 1987.

[45] J. Freudiger, R. Shokri, and J. Hubaux. On the optimal placement of mix zones. In *Proc. of PETS*, pages 216–234, 2009.

[46] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.

[47] G. Iachello *et al.* . Control, deception, and communication: Evaluating the deployment of a location-enhanced messaging service. *Ubiquitous Computing*, pages 903–903, 2005.

[48] L. Gao and J. Rexford. Stable internet routing without global coordination. *IEEE/ACM Trans. Netw.*, 9:681–692, 2001.

[49] L. Gao and X. Wang. A game approach for multi-channel allocation in multi-hop wireless networks. In *Proc. of MobiHoc*, pages 303–312, 2008.

[50] L. Gao, Y. Xu, and X. Wang. Map: Multiauctioneer progressive auction for dynamic spectrum access. *IEEE Trans. Mobile Comput.*, 10(8):1144–1161, Aug. 2011.

[51] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman, 1979.

[52] Gartner. Gartner: Microsoft windows phone market share to surpass Apple's iOS in 2015. `http://windowsphone7reviews.net/gartner-microsoft-windows-phone-market-share-to-surpass-apple's-ios-in-2015`.

[53] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Trans. Mobile Comput.*, 7(1):1–18, Jan. 2008.

[54] P. Godfrey, M. Schapira, A. Zohar, and S. Shenker. Incentive compatibility and dynamics of congestion control. In *Proc. ACM SIGMETRICS*, pages 95–106, 2010.

[55] P. Golle, K. Leyton-Brown, and I. Mironov. Incentives for sharing in peer-to-peer networks. In *Proc. of EC*, pages 264–267, 2001.

[56] Google Latitude. https://www.google.com/latitude/.

[57] V. Granville, M. Krivanek, and J.-P. Rasson. Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):652–656, June 1994.

[58] T. Griffin, F. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Trans. Netw.*, 10:232–243, 2002.

[59] T. Griffin and G. Wilfong. A safe path vector protocol. In *Proc. IEEE INFOCOM*, pages 490–499, 2000.

[60] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973.

[61] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of ACM MobiSys*, pages 31–42, 2003.

[62] P. Gupta and P. Kumar. The capacity of wireless networks. *IEEE Trans. Inf. Theory*, 46(2):388–404, Mar. 2000.

[63] B. Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Trans. Inf. Theory*, 34(5):910–917, Sept. 1988.

[64] M. Halldórsson, J. Halpern, L. Li, and V. Mirrokni. On spectrum sharing games. In *Proc. of PODC*, pages 107–114, 2004.

[65] B. Han, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Distributed strategies for channel allocation and scheduling in software-defined radio networks. In *Proc. of INFOCOM*, pages 1521–1529, 2009.

[66] J. Huang, Z. Han, M. Chiang, and H. Poor. Auction-based resource allocation for cooperative communications. *IEEE J. Sel. Areas Commun.*, 26(7):1226–1237, Sept. 2008.

[67] P. Huang, A. Scheller-wolf, and K. Sycara. Design of a multi-unit double auction e-market. *Computational Intelligence*, 18:596–617, 2002.

[68] IDC. Worldwide smartphone market expected to grow 55% in 2011 and approach shipments of one billion in 2015, according to IDC. `http://www.idc.com/getdoc.jsp?containerId=prUS22871611`.

[69] J. Jaffe. Bottleneck flow control. *IEEE Trans. Commun.*, 29:954–962, 1981.

[70] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *Proc. of MobiCom*, pages 66–80, 2003.

[71] E. Kaasinen. User needs for location-aware mobile services. *Personal and Ubiquitous Computing*, 7(1):70–79, 2003.

[72] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *Proc. of IEEE ICPS*, pages 88–97, 2005.

[73] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[74] P. Klemperer. What really matters in auction design. *Journal of Economic Perspectives*, 16(1):169–189, 2002.

[75] B. Ko, V. Misra, J. Padhye, and D. Rubenstein. Distributed channel assignment in multi-radio 802.11 mesh networks. In *Proc. of WCNC*, pages 3978–3983, 2007.

[76] J. Konorski. Multiple access in ad-hoc wireless LANs with noncooperative stations. *NETWORKING*, 2345:1141–1146, 2002.

[77] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proc. ACM STACS*, pages 404–413, 1999.

[78] M. Kozlov, S. Tarasov, and L. Khachiyan. Polynomial solvability of convex quadratic programming. $\{$*USSR*$\}$ *Computational Mathematics and Mathematical Physics*, 20(5):223–228, 1980.

[79] V. Krishna. *Auction Theory*. Academic Press, March 2002.

[80] V. Krishna. *Auction theory*. Academic press, 2009.

[81] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[82] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. *IEEE Commun. Mag.*, 48:140–150, 2010.

[83] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: efficient protocols and outage behavior. *IEEE Trans. Inf. Theory*, 50:3062–3080, Dec. 2004.

[84] J. Lee and B. Hoh. Sell your experiences: A market mechanism based incentive for participatory sensing. In *Proc. of IEEE PerCom*, pages 60–68, 2010.

[85] H. Li and Z. Han. Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part ii: Unknown channel statistics. *IEEE Trans. Wireless Commun.*, 10(1):274–283, Jan. 2011.

[86] A. Liu, P. Ning, H. Dai, and Y. Liu. USD-FH: Jamming-resistant wireless communication using frequency hopping with uncoordinated seed disclosure. In *Proc. of MASS*, 2010.

[87] X. Liu, G. Noubir, R. Sundaram, and S. Tan. SPREAD: Foiling smart jammers using multi-layer agility. In *Proc. of INFOCOM*, pages 2536–2540, 2007.

[88] X. Liu, H. Zhao, M. Pan, H. Yue, X. Li, and Y. Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *Proc. of IEEE INFOCOM*, pages 972–980, 2012.

[89] Q. Ma, P. Steenkiste, and H. Zhang. Routing high-bandwidth traffic in max-min fair share networks. In *Proc. ACM SIGCOMM*, pages 206–217, 1996.

[90] A. MacKenzie and S. Wicker. Stability of multipacket slotted aloha with selfish users and perfect information. In *Proc. of INFOCOM*, pages 1583–1590, 2003.

[91] M. Marina, S. Das, and A. Subramanian. A topology control approach for utilizing multiple channels in multi-radio wireless mesh networks. *Comput. Netw.*, 54(2):241–256, Feb. 2010.

[92] A. Mayer, Y. Ofek, and M. Yung. Approximating max-min fair rates via distributed local scheduling with partial information. In *Proc. IEEE INFOCOM*, pages 928–936, 1996.

[93] R. McAfee and J. McMillan. Multidimensional incentive compatibility and mechanism design. *Journal of Economic Theory*, 46(2):335–354, 1988.

[94] R. P. McAfee. A dominant strategy double auction. *Journal of Economic Theory*, 56(2):434–450, Apr. 1992.

[95] M. Meeker. Mary Meeker: Smartphones will surpass PC shipments in two years. `http://techcrunch.com/2010/11/16/meeker-smartphones-pcs/`.

[96] I. Milchtaich. Congestion games with player-specific payoff functions. *Games and Economic Behavior*, 13:111–124, Mar. 1996.

[97] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, 7:234–243, 1978.

[98] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A client-driven approach for channel management in wireless LANs. In *Proc. of INFOCOM*, pages 1–12, 2006.

[99] D. Mishra and R. Garg. Descending price multi-item auctions. *Journal of Mathematical Economics*, 42(2):161–179, 2006.

[100] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *Proc. of SenSys*, pages 323–336, 2008.

[101] D. Monderer and L. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.

[102] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. PIER, the personal environmental impact report, as a platform for participatory sensing systems research. In *Proc. of ACM MobiSys*, pages 55–68, 2009.

[103] R. Myerson. Optimal auction design. *Math. of Operations Research*, 6:58–73, 1981.

[104] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2):265–281, Apr. 1983.

[105] D. Nace, N. Doan, E. Gourdin, and B. Liau. Computing optimal max-min fair resource allocation for elastic flows. *IEEE/ACM Trans. Netw.*, 14:1272–1281, 2006.

[106] A. Nahir, A. Orda, and A. Freund. Topology design and control: a game-theoretic perspective. In *Proc. IEEE INFOCOM*, pages 1620–1628, 2009.

[107] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein. Using channel hopping to increase 802.11 resilience to jamming attacks. In *Proc. of INFOCOM*, pages 2526–2530, 2007.

[108] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

[109] T. C.-Y. Ng and W. Yu. Joint optimization of relay strategies and resource allocations in cooperative cellular networks. *IEEE J. Sel. Areas Commun.*, 25(2):328–339, Feb. 2007.

[110] D. C. Parkes, J. Kalagnanam, and M. Eso. Achieving budget-balance with vickrey-based payment schemes in exchanges. In *International Joint Conference on Artificial Intelligence Proceedings*, pages 1161–1168, 2001.

[111] S. Parsons, J. Rodriguez-Aguilar, and M. Klein. Auctions and bidding: A guide for computer scientists. *ACM Computing Surveys (CSUR)*, 43(2):10:1–10:59, 2011.

[112] Pew Internet. Three-quarters of smartphone owners use location-based services, 2012.

[113] C. R. Plott and P. Gray. The multiple unit double auction. *Journal of Economic Behavior & Organization*, 13(2):245–258, Mar. 1990.

[114] K. Ramachandran, E. Belding, K. Almeroth, and M. Buddhikot. Interference-aware channel assignment in multi-radio wireless mesh networks. In *Proc. of INFOCOM*, pages 1–12, 2006.

[115] B. Raman. Channel allocation in 802.11-based mesh networks. In *Proc. of INFOCOM*, pages 1–10, 2006.

[116] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu. Earphone: An end-to-end participatory urban noise mapping. In *Proc. of ACM/IEEE IPSN*, pages 105–116, 2010.

[117] A. Raniwala, K. Gopalan, and T. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 8(2):50–65, 2004.

[118] S. Reddy, D. Estrin, and M. B. Srivastava. Recruitment framework for participatory sensing data collections. In *Proc. of Pervasive*, pages 138–155, 2010.

[119] RFC. `Http://tools.ietf.org/html/rfc2205`.

[120] A. Richa, C. Scheideler, S. Schmid, and J. Zhang. A jamming-resistant mac protocol for multi-hop wireless networks. In *Proc. of DISC*, 2010.

[121] A. Richa, C. Scheideler, S. Schmid, and J. Zhang. Competitive and fair medium access despite reactive jamming. In *Proc. of ICDCS*, 2011.

[122] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez. For sale: your data by: you. In *Proc. of ACM HotNets*, pages 13:1–13:6, 2011.

[123] R. W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.

[124] Y. Sagduyu, R. Berry, and A. Ephremides. Mac games for distributed wireless network security with incomplete information of selfish and malicious user types. In *Proc. of GameNets*, pages 130–139, 2009.

[125] M. Schapira and A. Zohar. Congestion-control games. `http://leibniz.cs.huji.ac.il/tr/1060.pdf`, 2008. A working paper.

[126] Sensorly. Sensorly. http://www.sensorly.com.

[127] S. Sharma, Y. Shi, Y. T. Hou, and S. Kompella. An optimal algorithm for relay node assignment in cooperative ad hoc networks. *IEEE/ACM Trans. Netw.*, 19(3):879–892, June 2010.

[128] N. Shastry and R. Adve. Stimulating cooperative diversity in wireless ad hoc networks through pricing. In *Proc. of IEEE ICC*, pages 3747–3752, 2006.

[129] Y. Shi, S. Sharma, Y. T. Hou, and S. Kompella. Optimal relay assignment for cooperative communications. In *Proc. of ACM MobiHoc*, pages 3–12, 2008.

[130] M. Shin, S. Lee, and Y. Kim. Distributed channel assignment for multi-radio wireless networks. In *Proc. of MASS*, pages 417–426, 2006.

[131] Y. Singer. Budget feasible mechanisms. In *Proc. of IEEE FOCS*, pages 765–774, 2010.

[132] K. Sridhar, C. Casetti, and C. Chiasserini. A localized and distributed channel assignment scheme for wireless mesh networks. In *Proc. of LCN*, pages 45–52, 2009.

[133] V. Stackelberg. *Marketform und Gleichgewicht*. Oxford University Press, 1934.

[134] M. Steenstrup. Inter-domain policy routing protocol specification: Version 1, 1993.

[135] A. Subramanian, H. Gupta, S. Das, and J. Cao. Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Trans. Mobile Comput.*, 7(12):1459–1473, Dec. 2008.

[136] J. Tang, G. Xue, and W. Zhang. Interference-aware topology control and QoS routing in multi-channel wireless mesh networks. In *Proc. of MobiHoc*, pages 68–77, 2005.

[137] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson. Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones. In *Proc. of ACM SenSys*, pages 85–98, 2009.

[138] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16:8–37, 1961.

[139] K. Vu, R. Zheng, and J. Gao. Efficient algorithms for k-anonymous location privacy in participatory sensing. In *Proc. of IEEE INFOCOM*, pages 2399–2407, 2012.

[140] W. R. Wade. *An Introduction to Analysis*. Pearson, fourth edition, 2010.

[141] B. Wang, Z. Han, and K. Liu. Distributed relay selection and power control for multiuser cooperative communication networks using buyer/seller game. In *Proc. of IEEE INFOCOM*, pages 544–552, 2007.

[142] W. Wang, B. Li, and B. Liang. District: Embracing local markets in truthful spectrum double auctions. In *Proc. of IEEE SECON*, pages 521–529, 2011.

[143] W. Wang, X.-Y. Li, and Y. Wang. Truthful multicast routing in selfish wireless networks. In *Proc. of ACM MobiCom*, pages 245–259, 2004.

[144] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and B. Xu. L2P2: Location-aware location privacy protection for location-based services. In *Proc. of IEEE INFOCOM*, pages 1996–2004, 2012.

[145] B. Waxman. Routing of multipoint connections. *IEEE J. Sel. Areas Commun.*, 6(9):1617–1622, 1988.

[146] D. B. West. *Introduction to Graph Theory (2nd Edition)*. Prentice Hall, 2 edition, Sept. 2001.

[147] Where. http://where.com.

[148] F. Wu, T. Chen, S. Zhong, L. E. Li, and Y. R. Yang. Incentive-compatible opportunistic routing for wireless networks. In *Proc. of ACM MobiCom*, pages 303–314, 2008.

[149] F. Wu, N. Singh, N. Vaidya, and G. Chen. On adaptive-width channel allocation in non-cooperative, multi-radio wireless networks. In *Proc. of INFOCOM*, pages 2790–2798, 2011.

[150] F. Wu, S. Zhong, and C. Qiao. Globally optimal channel assignment for non-cooperative wireless networks. In *Proc. of IEEE INFOCOM*, pages 1543–1551, 2008.

[151] Y. Wu, B. Wang, K. Liu, and T. Clancy. Anti-jamming games in multi-channel cognitive radio networks. *IEEE J. Sel. Areas Commun.*, 30(1):4–15, Jan. 2012.

[152] H. Xu, L. Huang, G. Wang, T. Xu, and G. Liu. Joint relay assignment and power allocation for cooperative communications. *Wirel. Netw.*, 16(8):2209–2219, Nov. 2010.

[153] T. Xu and Y. Cai. Location anonymity in continuous location-based services. In *Proc. of ACM GIS*, pages 39:1–39:8, 2007.

[154] W. Xu, W. Trappe, Y. Zhang, and T. Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *Proc. of MobiHoc*, pages 46–57, 2005.

[155] W. Xu, T. Wood, and Y. Zhang. Channel surfing and spatial retreats: defenses against wireless denial of service. In *Proc. of Workshop on Wireless Security*, 2004.

[156] D. Yang, X. Fang, and G. Xue. OPRA: Optimal relay assignment for capacity maximization in cooperative networks. In *Proc. of IEEE ICC*, 2011.

[157] D. Yang, X. Fang, and G. Xue. Truthful auction for cooperative communications. In *Proc. of ACM MobiHoc*, pages 89–98, 2011.

[158] D. Yang, X. Fang, and G. Xue. Channel allocation in non-cooperative multi-radio multi-channel wireless networks. In *Proc. of IEEE INFOCOM*, pages 882–890, 2012.

[159] D. Yang, X. Fang, and G. Xue. HERA: An optimal relay assignment scheme for cooperative networks. *IEEE J. Sel. Areas Commun.*, 3:245–253, Feb. 2012.

[160] D. Yang, X. Fang, and G. Xue. Truthful incentive mechanisms for k-anonymity location privacy. In *Proc. of IEEE INFOCOM*, pages 3094–3102, 2013.

[161] D. Yang, G. Xue, X. Fang, S. Misra, and J. Zhang. Routing in max-min fair networks: A game theoretic approach. In *Proc. IEEE ICNP*, pages 1–10, 2010.

[162] D. Yang, G. Xue, X. Fang, S. Misra, and J. Zhang. A game-theoretic approach to stable routing in max-min fair networks. *IEEE/ACM Trans. Netw.*, PP(99), 2013.

[163] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing. In *Proc. of ACM MobiCom*, pages 173–184, 2012.

[164] D. Yang, G. Xue, J. Zhang, A. Richa, and X. Fang. Coping with a smart jammer in wireless networks: A stackelberg game approach. *IEEE Trans. Wireless Commun.*, PP(99), 2013.

[165] D. Yang, J. Zhang, X. Fang, G. Xue, and A. Richa. Optimal transmission power control in the presence of a smart jammer. In *Proc. of Globecom*, pages 5506–5511, 2012.

[166] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun. Multi-channel assignment in wireless sensor networks: a game theoretic approach. In *Proc. of INFOCOM*, pages 1–9, 2010.

[167] P. Zhang, Z. Xu, F. Wang, X. Xie, and L. Tu. A relay assignment algorithm with interference mitigation for cooperative communication. In *Proc. of IEEE WCNC*, pages 1–6, Apr. 2009.

[168] Y. Zhao, R. Adve, and T. Lim. Improving amplify-and-forward relay networks: optimal power allocation versus selection. *IEEE Trans. Wireless Commun.*, 6(8):3114–3123, Aug. 2007.

[169] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques. In *Proc. of MobiCom*, pages 117–131, 2005.

[170] X. Zhou, S. Gandhi, S. Suri, and H. Zheng. eBay in the sky: strategy-proof wireless spectrum auctions. In *Proc. of ACM MobiCom*, pages 2–13, 2008.

[171] Q. Zhu, H. Li, Z. Han, and T. Bas andar. A stochastic game model for jamming in multi-channel cognitive radio systems. In *Proc. of ICC*, pages 1–6, 2010.