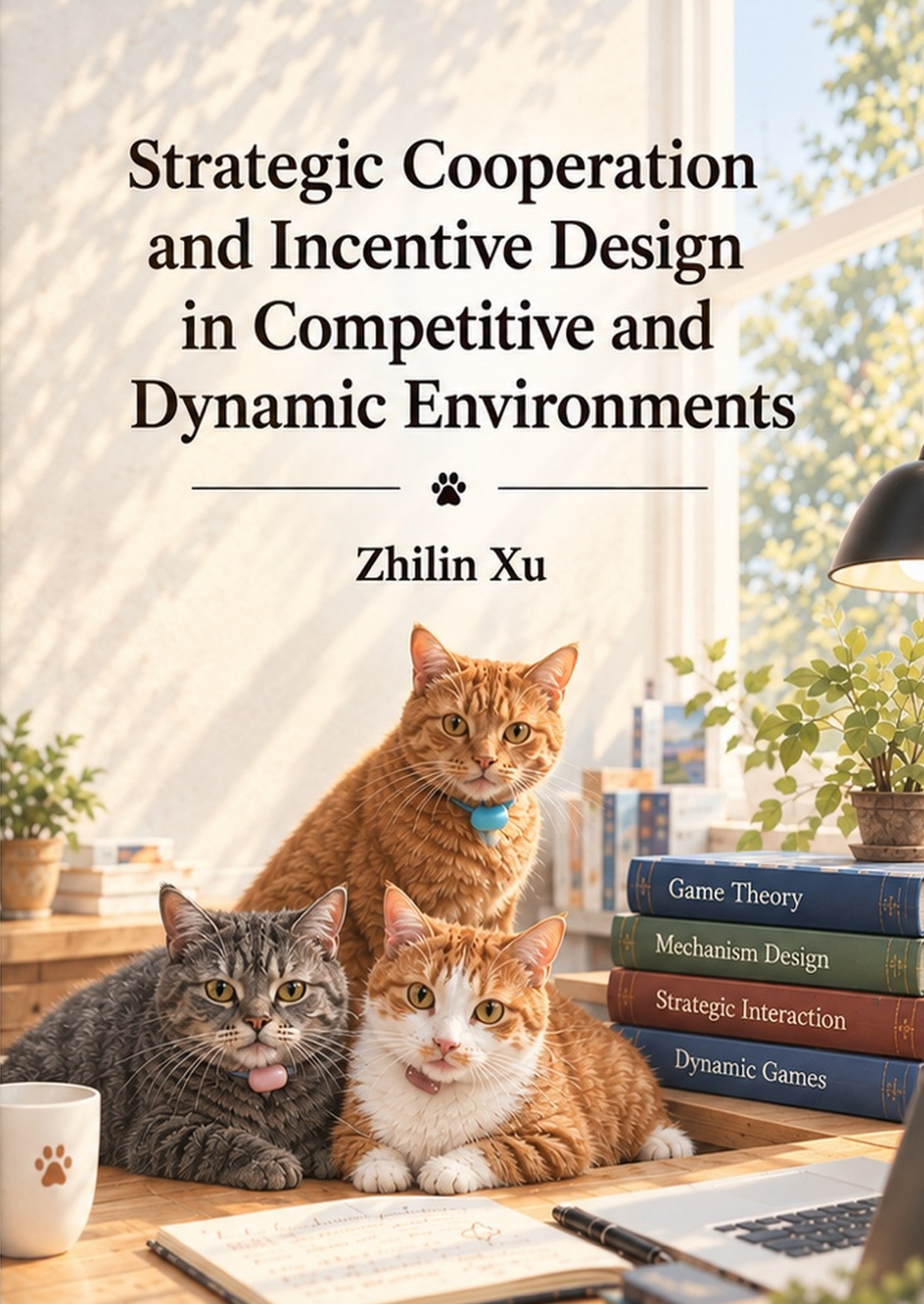
Strategic Cooperation and Incentive Design in Competitive and Dynamic Environments

Zhilin Xu



Propositions accompanying the PhD-thesis

The Maastricht Thesis Template

Zhilin Xu, 28 october 2026

1. In the Bertrand model with collusion, the size of a stable coalition has an upper bound of six. [Chapter 2]
2. In the Bertrand model with collusion, the cartel firms have higher profits than in the Bertrand model without collusion. However, fringe firms profit even more. [Chapter 2]
3. The existence of a Stackelberg equilibrium does not imply an efficient allocation of resources. [Chapter 3]
4. The stability of cooperation depends not only on the benefits of cooperation but also on how these benefits are distributed. [Chapter 3]
5. The most difficult task in optimization is not the search for an optimal solution, but to decide which problem to solve.
6. The pursuit of individual interests can sometimes provide a basis for cooperation rather than prevent it.
7. In strategic decision-making, the timing of a decision can be as important as the decision itself.
8. More information does not always lead to better decisions.

Doctoral thesis

STRATEGIC COOPERATION AND INCENTIVE DESIGN IN COMPETITIVE AND DYNAMIC ENVIRONMENTS

Zhilin Xu

2026

STRATEGIC COOPERATION AND INCENTIVE DESIGN IN COMPETITIVE AND DYNAMIC ENVIRONMENTS

Dissertation

To obtain the degree of Doctor at Maastricht University,
on the authority of the Rector Magnificus, Prof. Dr. Jan Smits
and the degree of Doctor of Philosophy at the Northwestern Polytechnical University,
on the authority of the President, Prof. Dr. Baowei Song,
in accordance with the decision of the Board of Deans,
to be defended in public on Wednesday 28th of October 2026, at 10.00 hours

by Zhilin Xu

Supervisors

Prof. Dr. Dries Vermeulen, Maastricht University
Prof. Dr. Hao Sun, Northwestern Polytechnical University, Xi'an, P.R.C.

Co-supervisor

Dr. Christopher Kops, Maastricht University

Assessment Committee

Prof. Dr. Stan van Hoesel, Maastricht University

Prof. Dr. Peter Borm, Tilburg University

Dr. Iwan Bos, Maastricht University

Prof. Dr. Jörg Oechsler, University of Heidelberg

Prof. Dr. Hui Zhang, Northwestern Polytechnical University, Xi'an, P.R.C.

© Zhilin Xu, Maastricht 2026.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior written permission of the author.

Cover Zhilin Xu using AI image generation
Production ProefschriftMaken
ISBN 978-94-6534-727-1

To my parents, Yaping and Wu and my grandparents, Zhigang and Lihua

Acknowledgments

I would like to express my sincere gratitude to all those who have supported and assisted me throughout my academic journey and the completion of this thesis.

Foremost, I am deeply grateful to my supervisor, Dries, for your patient and dedicated guidance. When my calculations repeatedly went wrong and I felt anxious about the research process, you always encouraged me to relax, remain confident, and move forward step by step. Your continuous support and understanding were crucial to the completion of this work.

I am also deeply grateful to my co-supervisor, Chris, for your meticulous guidance on my academic writing. You carefully corrected my English and patiently taught me every detail, which greatly improved the clarity and rigor of my research presentation.

I would like to express my profound appreciation to my supervisor, Sun, who laid the foundation of my academic training and guided me onto the path of academic research. Your influence extends far beyond this thesis and will continue to shape my future work.

My sincere thanks also go to my project supervisor, Xu, who provided me with this valuable opportunity to study abroad. This experience has been instrumental to my academic growth and personal development.

I am also grateful to the China Scholarship Council (CSC) for the financial support that made my overseas study possible.

I would like to thank all my fellow students and colleagues in the KE department for their companionship, support, and stimulating discussions throughout my studies.

Anh, thank you for welcoming me to Maastricht, and showing me this beautiful city. Your companionship and support made my life far from lonely during this journey. You are like the sunshine in winter, making my life here warm and happy. I really enjoy our brunch and the long, cold city walk.

In particular, I would like to thank Guogai, Juzi and Qianqian. In a strange city and an unfamiliar country, they made me feel a genuine sense of belonging. I would like to thank my friend, Shasha, who welcomed me when I was almost left without a place to stay and generously shared invaluable "survival experience" for living here. Sha, who encouraged me to step out of my comfort zone, meet new people, and embrace life beyond academic work. We went out for hot pot in the rain, took spontaneous trips without much planning, and shared many moments that made everyday life meaningful. My heartfelt thanks also go to Sisi for always keeping an eye on my health and reminding me to take care of myself, and to Yingfu for the spontaneous trips that added freedom and excitement to my life. These moments of care, laughter, and shared experiences made my time here not only manageable, but truly memorable.

Zhilin Xu
Maastricht
2026-01-01

Contents

Acknowledgments vii

1 Introduction 1

2 Persistent Collusion in the Bertrand Model 13
- 2.1 Introduction . . . 15
- 2.2 Model . . . 18
- 2.3 Unique Stable Coalition . . . 20
- 2.4 Proof for Theorem 2.3.5 . . . 27
- 2.5 Discussion . . . 40
- 2.6 Conclusion . . . 47

3 Collaboration-Driven Mechanisms 49
- 3.1 Introduction . . . 51
- 3.2 MCS model . . . 53
- 3.3 Stackelberg MCS model . . . 56
- 3.4 Simulation results on the incomplete information . . . 79
- 3.5 Conclusion . . . 87

4 Time Variability 89
- 4.1 Introduction . . . 91
- 4.2 Stage 0: matching participants to requesters . . . 94
- 4.3 Stage t: updated match between requesters and participants 105
- 4.4 Simulations . . . 120
- 4.5 Conclusion . . . 125

5 Irregular Time Variability 127
5.1 Introduction 129
5.2 Participants-centered matching algorithm in the initial state 132
5.3 Updated participants-centered matching algorithm 145
5.4 Simulation results on the matching algorithm 160
5.5 Conclusion 170

6 Time-sensitive MCS systems 171
6.1 Introduction 173
6.2 Competition for time-sensitive MCS system stability 176
6.3 Cooperation to improve time-sensitive MCS system task completion rate 186
6.4 Simulation results on the performance of time-sensitive MCS system 201
6.5 Conclusion 206

Bibliography 209

Summary 219

Samenvatting 223

Impact of the thesis 227

About the author 231

1

Introduction

Darwin (1859) famously noted that a trait cannot be favored if it exists solely for the benefit of others. Abstracting from its biological context, this observation captures a general principle that is equally relevant for economic analysis: a specific type of behavior, for example, cooperation, is difficult to sustain when it conflicts with individual incentives.

Cooperation appears in many settings, from coalition formation between firms in a particular industry to increase market shares to agents sharing data through information systems to enhance service performance. It is well known that, in settings with repeated interactions, cooperation can be sustained as a noncooperative equilibrium without further external enforcement.

However, endogenous cooperation is not always socially desirable and sometimes even illegal. See for example Igami and Sugaya (2022) on vitamin cartels, Genesove and Mullin (2001) on sugar cartels, and Starc and Wollmann (2025) on pharmaceutical cartels.

On the other hand, there are also settings where cooperation is socially desirable, but hampered by for example free-riding. In information systems, data are publicly available, but the provision of data requires personal cost. Thus free-riding may occur which means that a user of the data profits from its availability but it is costly for a user to provide data.

Overall, there are situations where cooperation occurs but is socially undesirable, while in other cases socially beneficial forms of cooperation are not sustainable and require external incentivization. This contrast highlights the need to understand strategic behavior as well as the design of appropriate incentivization mechanisms. The implementation of such mechanisms is then the task of a centralized authority.

This thesis addresses two questions in this context. First, how can collusion emerge from individual incentives. Second, which mechanisms may help to sustain cooperation. The thesis consists of five essays that are organized into two parts that try to answer these two questions separately.

Part I: Endogenous collusion in Bertrand markets

In Part I we focus on coalition formation in settings where collusion among firms is not beneficial to consumers or possibly even illegal. Chapter 2 tries to answer the question of how collusion can emerge from individual incentives.

The model We look at the Bertrand model in which a finite number of symmetric firms compete in price in a market for a homogeneous good. Within this model, we allow firms to form a coalition in which firms within the coalition agree to set the same price.

We assume that this game is played repeatedly over time. Firms are free to decide whether they want to join the coalition or not one at a time. These decisions are made myopically, meaning that firms only compare their payoffs in the current time period with the payoffs in the next time period after the entry or exit decision has occurred. We analyze the resulting dynamic model.

The solution concept Given a fixed coalition within a given time, we assume that firms play a Bertrand equilibrium. A Bertrand equilibrium is a price vector such that the price set by each non-coalition firm is a best response to the prices of all other firms and the price set by coalition firms is a best response to the prices of non-coalition firms.

Given a fixed coalition, the Bertrand equilibrium is the outcome within one time period. The coalition is stable if for each firm the current payoff is at least as high as the payoff in the next time period if that firm changes its current status (if it is a coalition member, it changes to non-coalition member and vice versa).

Results Given any number of firms, a stable coalition of size no more than 6 forms in equilibrium. To be precise, when the market has fewer than 5 firms, the grand coalition is formed, when the size of the market is between 6 and 11, the stable coalition size remains at 5, and when there are more than 12 firms, the stable collusion size remains fixed at 6.

Consequences We discuss two consequences. The first one is the "number effect", which is how the total number of firms in a market endogenously determines the possibility of forming a stable coalition among them. The second one is the "effectiveness", which means the coalition members achieve higher payoffs than under pure competition.

Number Effect It is important to detect and deter illegal acts of collusion. Classical economic theory (Chamberlin Edward 1933; Selten 1973) suggests that collusion is more likely to arise in markets with only a small number of firms. When the market size is large, cooperation tends to break down. However, Chapter 2 reveals a different possibility: collusive arrangements may also persist in large markets as small, stable coalitions that remain largely undetected due to their limited scale. Precisely because of their small size and low visibility, identifying and regulating such cartels becomes especially challenging. This suggests that implementing leniency programs for these markets may be an especially effective strategy.

Effectiveness In studies such as Selten (1973) and Deneckere and Davidson (1985), although collusive coalitions are shown to be stable, they are not necessarily effective. From an economic perspective, sustaining such ineffective coalitions appears counterintuitive, as they offer limited incentives for participation. In contrast, our results establish that whenever a coalition forms in equilibrium, it is necessarily effective: all participating members are strictly better off relative to the competitive benchmark. This effectiveness ensures that coalition formation is both individually rational and self-enforcing, thereby providing a strong foundation for the persistence of collusive behavior.

Part II: Exogenous mechanisms for Mobile Crowdsensing

In the second part, we analyze collusion among service providers in Mobile Crowdsensing (MCS). MCS is a setting in which cooperation between service providers, when collecting data, is beneficial to both them and endusers.

MCS is the designation of an environment in which participants upload data from mobile devices such as smartphones and smartwatches to an electronic platform on which these data can then be made available to service providers, so-called requesters in our model. The participants act both as data providers and endusers of the service provided by the requesters. The requesters use the available data to develop applications such as smart city management, environmental monitoring, and autonomous driving. Thus, the sensing platform acts as an intermediary between participants and requesters.

A recurring feature of MCS systems is data overlap: a single data point collected by a participant may simultaneously satisfy the requirements of multiple requesters. This characteristic makes coordination among requesters socially desirable. By coordinating their data acquisition, requesters can economize on data collection, reduce costs, and improve social welfare. However, since requesters are typically self-interested, and participants have an incentive to provide sensing data to each requester individually, such cooperation may not arise automatically. Therefore, centralized design of data acquisition and distribution on the platform is needed to coordinate data processing and thereby improving social welfare.

In Chapter 3, we analyze the above general model by means of several variations of this general model. In one variation we study the model where requesters have complete information regarding the costs of collecting and

uploading data for participants. In the second model we study the incomplete information case.

In the first model, the complete information case, we use a Stackelberg game to analyze the interactions between requesters and participants. The requesters as the leaders in the game first place requests for data together with a proposal for a payment scheme. Then the participants as followers decide the data accuracy level based on the proposed payment scheme. The accuracy level of the data describes how closely the collected data matches the true situation. The platform does not play an active role in this game, it simply collects, stores and distributes data.

We define the Stackelberg equilibrium in this setting. However, with multiple requesters, it is difficult to compute equilibrium strategies. In order to address this issue, we introduce the concept of "team". Each team represents one specific type of data. Based on their data requirements, requesters then choose which teams they want to have information from.

To address data overlap and improve the MCS system's efficiency, requesters within the same team request data jointly. We argue that, if we want to compute the Stackelberg equilibrium, we can treat each team separately. Each team now functions as a single leader in the Stackelberg game.

Thus the equilibrium in the new game is easier to compute. However, the equilibrium is the result of cooperation and cannot guarantee the requesters' willingness to cooperate. In the new game, the team pays the total payment to the participants, and each requester within the team divides the payment using a cost allocation method inspired by the Separable Costs Remaining Benefits Principle (Driessen 2013). This ensures that cooperation is individu-

ally rational for the requesters, meaning that it is beneficial for them to cooperate.

Next we analyze the incomplete information case. In order to solve this variation of the model, we assume that requesters use Q-learning, a model-free reinforcement learning algorithm, to estimate participants' costs. The simulation results show that even under incomplete information, the outcome of the Q-learning algorithm converges to the Stackelberg equilibrium of the complete information model.

In Chapter 4, we now assume that teams, which are called data types in Chapter 4, can have different quality levels, where a higher quality value indicates a better quality. For example, a better resolution, or better indication of position in GPS systems. Each requester has a per-unit data value for each data type, provided that the quality of the data satisfies a minimum quality requirement, and the amount of data of at least minimum quality does not exceed the maximum amount of data required for the services provide by that requester. Each participant can provide a quantity of each data type of any level of quality to any requester, but the costs for collecting such data increases in both quantity and quality. Requesters aim to collect a sufficient amount of data that meets their quality requirements. Participants aim to maximize their utility.

We also assume that data are provided over time by participants. Time is assumed to be discrete, and at each time period, requesters can ask for an amount of data of a certain minimum level of quality at a per-unit price, where this price is also chosen by the requester.

In the first time period, requesters request data individually while in the remaining time periods, requesters can form a coalition and thus choose to re-

quest data together. Participants can choose to offer data of certain quality level to a certain requester for the price issued by the requester.

We model the relationship between requesters and participants in the first time period as a Stackelberg game, where requesters as leaders first announce their data request and payment schemes. Then participants as followers choose which data type at which quality level to provide to which requester.

We solve for the equilibrium of the Stackelberg model. Since each participant has a unique optimal strategy, each requester is able to choose a strategy that maximizes its utility. Thus we find a unique Stackelberg equilibrium. Based on requesters' equilibrium strategies, each participant will choose one requester to match with to maximize its own utility.

In each later time period, we allow requesters to form a coalition. At each given later time period, requesters who were not matched with any participant in the previous period are called weak requesters in this period. The coalition consists of those requesters whose per-unit data value is at most the maximum per-unit data value of all weak requesters.

Requesters within the coalition will keep their required total data amount, but update the minimum data quality requirements and the payment schemes. The minimum data quality requirement of the coalition is chosen in such way that data collected by the coalition can be used by each coalition member. The coalition's payment scheme is the minimum payment which ensures that the coalition can always be matched with participants. The payments are shared among the coalition members proportionally to per-unit data values.

We show that the coalition thus formed is stable. The simulation results show that the coalition formation can reduce requesters' costs by up to 90%. The mechanism guarantees that each requester receives data within the first two stages in which it participates.

Chapter 5 is a robustness check for Chapter 4. In Chapter 5, each data type has multiple quantity levels instead of quality levels. This way requesters can prevent participants to provide many small or scattered contributions. Through simulations, we find that even when the number of participants can vary over time, the results from Chapter 4 still hold.

In Chapter 6, we study MCS systems where tasks can be time-sensitive. We assume that requesters need to collect their data within a given time window. Data collected before or after the effective time window are invalid and increase requester costs, reducing efficiency and utility.

We also assume that the sensing platform aims to maximize the completion rate of requester tasks in order to attract more requesters. This creates a tension between requesters' goals and the sensing platforms' goals since these goals may not be perfectly aligned.

To resolve the tension, we design an algorithm in which, in each time period, requesters will first compete to maximize their own utility, and then cooperate to share data. During the competitive phase, we focus on maximizing requesters' utilities. The competitive phase in this chapter is identical to the first time period in Chapter 4. The platform does not participate in this phase, it just passes data through.

Although the setting is the same in both chapters, the solution in this chapter is slightly different. We endow each requester with a budget. Requesters are then sorted in a fixed order to select participants. If a requester has enough

budget to recruit a candidate participant, a match is made and the participant provides data and gets paid. The budget of the requester is then updated.

From the simulation results of the competitive phase, we observe that some requesters may obtain too much data, while some requesters may not get enough data. In the worst case, the task completion rate is only 8%, and the data waste rate, which is the proportion of requesters who collected a surplus of data, is 100%.

In the cooperative phase, the centralized sensing platform classifies the requesters into different categories based on the matching results in the competitive phase. Requesters who get more data than needed in the competitive phase are classified as strong requesters with redundant data. Requesters who fail to get enough data in the competitive phase are classified as weak requesters with insufficient data. Middle requesters are those who get exactly the amount of data they required. To increase the task completion rate and reduce the data waste rate, the centralized MCS platform reallocates excess data from strong requesters to weak requesters. We formulate the data reallocation process as a cooperative game. The characteristic function of the game is the task completion margin, which is the difference between the amount of data actually obtained by the requester and the requested amount. A data reallocation algorithm among requesters is proposed to find an efficient solution to the cooperation game. The algorithm maximizes the sensing platform's task completion rate.

In simulation, we compare the task completion rate of the time-sensitive MCS system under different requester orders (budgets, emergency degree, and payments) and different parameter sets. We guarantee a task completion rate of at least 94%. If the requesters are ordered in the same order in both the competitive phase and the cooperative phase, the task completion rate is

higher. Moreover, when all requesters are sorted by emergency degree (prioritized by the least remaining time), the task completion rate can reach 99% through our dynamic coopetition incentive mechanism.

2

Persistent Collusion in the Bertrand Model

This Chapter is adapted from our working paper: Zhilin Xu, Christopher Kops, Dries Vermeulen. "Persistent Collusion in the Bertrand Model, Where Six is the Upper Bound" (working paper).

We analyze a Bertrand model with collusion. Regardless of the number of firms in the market, some firms collude in equilibrium, forming a stable coalition which maximizes joint profits. For a given number of firms, the size of a stable coalition is unique and has a hard upper bound of six. Our results identify this size for any finite number of firms in the market. Consistent with Selten (1973), the unique stable coalition in a market with four or fewer firms is the grand coalition. But, while in Selten (1973) the probability of collusion becomes negligible when more firms compete, the equilibrium in our model always features a stable coalition of size at most six.

2.1 Introduction

In his seminal paper, Selten (1973) analyzed the connection between the number of firms in a market and the tendency of these firms to collude. His analysis shows that, with four or fewer firms in Cournot competition, firms always collude, while the probability of collusion becomes negligible when more firms compete. Some experimental oligopoly games (Huck, Normann, and Oechssler 2004) confirm these results, others (Devetag and Ortmann 2007) draw them into question. Indeed, it is conceivable that a small cartel in a real-world market with many firms manages to steer below the radar of competition authorities. This begs the question whether there is a theoretical foundation that can be given to such collusive practices. This chapter provides an affirmative answer to this question.

The literature on collusion is vast, running from experimental, to empirical and theoretical work. Most closely related to the model in this chapter are the ones in Selten (1973) and Deneckere and Davidson (1985). In the following list, we lay out the key differences between these three approaches.

[1] Bertrand versus Cournot. Selten (1973) considers a Cournot model with collusion. Deneckere and Davidson (1985) and this chapter analyze instead a Bertrand model with collusion.

[2] Tacit collusion vs bargaining stage. Selten (1973) explicitly models a pre-game bargaining stage, wherein proposals are made simultaneously. As in Deneckere and Davidson (1985), we do not really distinguish between tacit and explicit collusion. But, to the extent that there is no bargaining stage and the decision to collude is not a simultaneous one, our model is closer to tacit than to explicit collusion.

[3] Deterministic, unique coalition size. The model in Selten (1973) is probabilistic. With five or more firms, a stable coalition of colluding firms is unlikely to form and can be of various sizes if it forms. In contrast, our results and those in Deneckere and Davidson (1985) are deterministic. In Deneckere and Davidson (1985), when goods are close enough substitutes it is always the grand coalition with the potential exception of some small firms forming as the stable coalition. In our model, a stable coalition always forms in equilibrium, it is always unique in size, but not its membership.

[4] Hard upper bound. Whereas there neither is a hard upper bound on the coalition size in Selten (1973) nor in Deneckere and Davidson (1985), such an upper bound exists in the Bertrand model with collusion analyzed in this chapter. Indeed, our results provide a theoretical foundation for there always being a small number of firms colluding in equilibrium. The size of the stable coalition is always unique and never exceeds six.

[5] Myopic dynamics. A stable coalition in Selten (1973) is the outcome of a sub-game perfect equilibrium. In this chapter and also in Deneckere and Davidson (1985), a stable coalition of colluding firms can be viewed as the outcome of a myopic dynamic formulation process.[1]

[6] Robustness to industry size. For markets with six or more firms, the probability of a stable coalition of colluding firms is close to zero in Selten (1973). In Deneckere and Davidson (1985), with goods being close enough substitutes the stable coalition is the grand coalition with the potential exception of some small firms. Instead, our model always has a stable coalition of at most six colluding firms forming in equilibrium irrespective of the number of firms in the market.

[1] A coalition is stable in our sense if it is robust to unilateral deviation. No coalition firm has an incentive to leave it and no non-coalition firm has an incentive to enter it.

[7] Effectiveness. With five or more firms, collusion in Selten, 1973 is unlikely and ineffective. A stable coalition of colluding firms only forms with a probability close to zero and a coalition firm's payoff does not exceed the one from Cournot competition. In Deneckere and Davidson (1985) it also not a given that all coalition firms benefit from the coalition. In our model, however, a stable coalition of colluding firms always forms and profits of coalition and non-coalition firms increase in coalition size.

Besides Selten Selten, 1973 and Deneckere and Davidson (1985), the early theoretical literature on collusion also includes d'Aspremont et al. (1983), Marie-Paule Donsimoni (1985), and M.-P. Donsimoni, Economides, and Polemarchakis (1986). Rather than assuming a Cournot model with collusion, these papers draw on the collusive price-leadership concept of Markham (1951). While Diamantoudi (2005) is a more recent contribution in this vein, there are also studies on stable collusion in the framework of infinitely repeated games (Compte, Jenny, and Rey 2002; Vasconcelos 2005; Bos and Harrington 2010; Bos and Harrington 2015; Bos and Marini 2022; Bos, Cesi, and Marini 2025). Complementing this literature is theoretical work on the role that communication (Harrington and Skrzypacz 2011; Awaya and Krishna 2016) and private information (D. A. Miller 2011) play for stable collusion, as well as theoretical and non-theoretical work on the impact that leniency programs have on discovery and deterrence of cartels (Apesteguia, Dufwenberg, and Selten 2007; N. H. Miller 2009; Bos, Letterie, and Vermeulen 2015).

Expanding the scope of these theoretical studies is experimental and empirical work on the topic. Huck, Normann, and Oechssler, 2004 find collusion in a laboratory experiment with just two firms, but never with four or five firms, confirming a number effect in the general sense of Selten (1973). Other work (Devetag and Ortmann 2007) points towards ways that allow to suc-

cessfully engineer collusion in the laboratory when there are more than two firms, with one such way lying in one-off communication (Fonseca and Normann 2012; Fonseca and Normann 2014). Studies on known real-world cartels run from vitamin cartels (Igami and Sugaya 2022), to cartels in the sugar industry (Genesove and Mullin 2001) and pharmaceutical cartels (Starc and Wollmann 2025).

Section 2.2 defines a Bertrand model with collusion. Section 2.3 infers the equilibrium outcome of a stable coalition of colluding firms. Section 2.4 is the detailed proof process of finding the unique stable coalition size. Section 2.5 discusses this result in view of implications and effectiveness in light of cartel detection and deterrance. Section 2.6 concludes.

2.2 Model

We consider a set $N = \{1, 2, \ldots, n\}$ of firms, competing over prices in a Bertrand model with collusion. We assume $n \geq 3$, since $n = 3$ is the minimum market size for which collusion and competition can coexist. Firms supply differentiated products. As such, demand is not just driven towards the lowest price. Instead, each firm $i \in N$ simultaneously sets its price p_i and produces to meet its demand[2],

$$D_i(p) = A - np_i + \sum_{j \neq i} p_j \quad where \quad p = (p_i)_{i \in N}.$$

The demand for firm i's product, which we assume to be non-negative, decreases in firm i's own price p_i and increases in other firms' prices p_{-i}. If firm

[2]This particular demand function is an n-firm variation of the specification by Shubik and Levitan (1980), p.69

i increases this price by one unit, it incurs a loss of n customers. The higher price induces $n-1$ customers to switch to the other $n-1$ firms (one to each), and one customer to exit the market. Since $n > 1$, the demand for a firm's product is more responsive to changes in the price of this product than to changes in the prices of other firms' products.

The goal of each firm i is to maximize its profits

$$\pi_i = (p_i - c)D_i(p)$$

where c denotes the marginal production cost which is assumed the same for all firms, satisfying $A > c$. To maximize its profits, a firm can enter a price competition with all other firms or it can collude on a price. When k firms collude they form a coalition Γ, setting a collusive price p^d at which all coalition members supply their products. Each non-coalition member j continues to choose its own price p_j.

As is standard in the literature[3], we assume that at most a single coalition forms. Now, assume a coalition $\Gamma \subseteq N$ of size k forms. Then, each coalition firm $i \in \Gamma$ sets the collusive price p^d and, thus, faces the demand

$$D_i(p) = A - np^d + (k-1) \cdot p^d + \sum_{j \notin \Gamma} p_j$$

In turn, every non-coalition member $l \notin \Gamma$ then faces the demand

$$D_l(p) = A - np_l + k \cdot p^d + \sum_{j \notin \Gamma, j \neq l} p_j$$

[3]Recent exceptions which consider multiple cartels are Bos and Marini (2022), Khan and Peeters (2024), Khan and Peeters (2025).

2.3 Unique Stable Coalition

This section presents our main result. It starts by identifying, for a given coalition size, what prices and profits characterize the Bertrand equilibrium in this case. It then continues by defining the notion of a stable coalition. Building on this notion, it shows that a stable coalition always forms in equilibrium and identifies the size of this coalition for any number of firms in the market.

2.3.1 Bertrand Competition with collusion

By agreeing on a collusive price p^d, each coalition firm always plays the same strategy. Each non-coalition firm i charges its own individual price p_i and is thus free to play its own strategy. Given a fixed coalition, a Bertrand equilibrium is then a strategy profile in the form of a price vector p such that non-coalition firms best respond to the prices of all other firms and coalition firms best respond to the prices of non-coalition firms.

The next result shows that in a Bertrand equilibrium with fixed coalition all non-coalition firms also agree charge the same price for their products.

Theorem 2.3.1. *Given a coalition Γ of size k, in the Bertrand equilibrium firms charge the following prices*

$$p^d - c = \frac{2n+1}{Z_n(k)} \cdot (A-c) \quad \textit{and} \quad p^c - c = \frac{2n-k+2}{Z_n(k)} \cdot (A-c),$$

where p^d is the price of firms in the coalition Γ, p^c the price of firms not in this coalition and $Z_n(k) = 2n^2 + 4n - nk - k^2 + 2$.

Proof. Given a coalition Γ of size k and the demand structure from Section 2.2, all firms inside the coalition choose a joint price p^d so as to maximize the profits π_k^d given by

$$\pi_k^d = (p^d - c) \cdot (A - np^d + (k-1)p^d + \sum_{\ell \notin \Gamma} p_\ell)$$

The first order condition with respect to p^d then reads

$$A - (n-k+1) \cdot p^d + \sum_{\ell \notin \Gamma} p_\ell - (n-k+1) \cdot (p^d - c) = 0.$$

which can be simplified to

$$2 \cdot (n-k+1) \cdot p^d = A + \sum_{\ell \notin \Gamma} p_\ell + (n-k+1) \cdot c. \tag{2.1}$$

Next, any firm j outside of the coalition chooses a price p_j to maximize profits π_j given by

$$\pi_j = (p_j - c) \cdot (A - np_j + kp^d + \sum_{\ell \notin \Gamma, \ell \neq j} p_\ell)$$

The first order condition with respect to p_j then reads

$$A - np_j + kp^d + \sum_{\ell \notin \Gamma, \ell \neq j} p_\ell - n \cdot (p_j - c) = 0$$

which can be simplified to

$$(2n+1) \cdot p_j = A + n \cdot c + k \cdot p^d + \sum_{\ell \notin \Gamma} p_\ell \tag{2.2}$$

For the best response correspondences, we subtract Equation 2.1 from Equa-

tion 2.2, arriving at

$$2 \cdot (n - k + 1) \cdot p^d - (2n + 1) \cdot p_j = -k \cdot p^d + (1 - k) \cdot c.$$

This simplifies to

$$(2n - k + 2) \cdot p^d + (k - 1) \cdot c = (2n + 1) \cdot p_j.$$

Subtracting $(2n - k + 2) \cdot c$ from both sides and setting $E^c = p^c - c$ and $E^d = p^d - c$, we have

$$(2n - k + 2) \cdot E^d = (2n + 1) \cdot E^c.$$

Substituting this back into Equation 2.1 and Equation 2.2, we arrive at

$$\begin{aligned} E^d &= \frac{2n + 1}{2n^2 + 4n - nk - k^2 + 2} \cdot (A - c) \\ \text{and} \quad E^c &= \frac{2n - k + 2}{2n^2 + 4n - nk - k^2 + 2} \cdot (A - c). \end{aligned}$$

where we used that $2(n - k + 1)(2n + 1) - (n - k)(2n - k + 2) = 2n^2 + 4n - nk - k^2 + 2$.

This concludes the proof. □

Theorem 2.3.1 implies that in the Bertrand equilibrium with fixed coalition, the coalition firms never charge a lower price than non-coalition firms do, i.e., $p^d \geq p^c$. To see this, observe that $Z_n(k) > 0$, for all k. Then, it holds that $p^d = p^c$ for $k = 1$ and $p^d > p^c$ for $k > 1$.

Given the equilibrium prices for collusion and non-collusion firms from Theorem 2.3.1, it is straightforward to determine firms' equilibrium profits. The

next result does exactly that.

Theorem 2.3.2. *Given a coalition Γ of size k, in the Bertrand equilibrium firms' profits are*

$$\pi_k^d = (n-k+1)\cdot(p^d-c)^2 \quad \textit{and} \quad \pi_k^c = n\cdot(p^c-c)^2,$$

where π_k^d is a coalition member's profit, and π_k^c is the profit for each non-coalition member.

Proof. From Theorem 2.3.1 we know that

$$p^d - c = \frac{2n+1}{Z_n(k)}\cdot(A-c) \quad \text{and} \quad p^c - c = \frac{2n-k+2}{Z_n(k)}\cdot(A-c).$$

So, the demand function for a coalition member $i \in \Gamma$ is

$$\begin{aligned}
D_i(p) &= A-(n-k+1)\cdot p^d + (n-k)\cdot p^c \\
&= A-(n-k+1)\cdot(p^d-c) - (n-k+1)\cdot c + (n-k)\cdot p^c \\
&= A-c-(n-k+1)\cdot(p^d-c) + (n-k)\cdot(p^c-c) \\
&= A-c-(n-k+1)\cdot\frac{2n+1}{Z_n(k)}\cdot(A-c) + (n-k)\cdot\frac{2n-k+2}{Z_n(k)}\cdot(A-c) \\
&= \frac{Z_n(k)-(n-k+1)\cdot(2n+1)+(n-k)\cdot(2n-k+2)}{Z_n(k)}\cdot(A-c) \\
&= \frac{(2n+1)\cdot(n-k+1)}{Z_n(k)}\cdot(A-c) = (n-k+1)\cdot(p^d-c) = D^d.
\end{aligned}$$

So,

$$\pi^d = (p^d-c)\cdot D_i(p) = (n-k+1)\cdot(p^d-c)^2.$$

The demand for a non-coalition member $j \notin \Gamma$ is

$$D_j(p) = A-n\cdot p^c + k\cdot p^d + (n-k-1)\cdot p^c = A-(k+1)\cdot p^c + k\cdot p^d$$

$$
\begin{aligned}
&= A - c - (k+1) \cdot (p^c - c) + k \cdot (p^d - c) \\
&= A - c - (k+1) \cdot \frac{2n-k+2}{Z_n(k)} \cdot (A-c) + k \cdot \frac{2n+1}{Z_n(k)} \cdot (A-c) \\
&= \frac{2n^2 + 2n - nk}{Z_n(k)} \cdot (A-c) = n \cdot (p^c - c) = D^c.
\end{aligned}
$$

So,

$$\pi_k^c = (p^c - c) \cdot D_j(p) = n \cdot (p^c - c)^2.$$

This concludes the proof. □

With equilibrium prices and profits determined,[4] we can address the question of how coalition member profits compare to those of non-coalition members. Of course, this is still under the assumption that the size of the coalition is fixed.

Proposition 2.3.3. *Given a coalition Γ of size $k > 1$, the Bertrand equilibrium profits of a non-coalition member exceed those of a coalition member, i.e., $\pi_k^c > \pi_k^d$.*

Proof. For $k > 1$, it holds that $nk^2 > n$ and, thus, $0 < nk^2 - n + k - 1$. Furthermore,

$$
\begin{aligned}
& 0 < nk^2 - n + k - 1 \\
\Leftrightarrow\ & n - k + 1 < nk^2 \\
\Leftrightarrow\ & 4n^3 + 4n^2 + n - 4n^2k - 4nk - k + 4n^2 + 4n + 1 \\
& < 4n^3 - 4n^2k + 8n^2 + nk^2 - 4nk + 4n \\
\Leftrightarrow\ & (n - k + 1) \cdot \left(4n^2 + 4n + 1\right) < 4n^3 - 4n^2k + 8n^2 + nk^2 - 4nk + 4n
\end{aligned}
$$

[4] The case of $k = 1$ equals the standard oligopolistic competition outcome, with each firm charging the price $p_i - c = \frac{A-c}{n+1}$ and receiving the profits $\pi_i = n \cdot \left(\frac{A-c}{n+1}\right)^2$

$$
\begin{aligned}
&\Leftrightarrow \quad (n-k+1)\cdot(2n+1)^2 < n\cdot(2n-k+2)^2 \\
&\Leftrightarrow \quad (n-k+1)\cdot\left(\frac{2n+1}{Z_n(k)}\cdot(A-c)\right)^2 < n\cdot\left(\frac{2n-k+2}{Z_n(k)}\cdot(A-c)\right)^2 \\
&\Leftrightarrow \quad (n-k+1)\cdot(p^d-c)^2 < n\cdot(p^c-c)^2 \\
&\Leftrightarrow \quad \pi_k^d < \pi_k^c
\end{aligned}
$$

where the formulas for p^d and p^c from Theorem 2.3.1 and those for π_k^d and π_k^c from Theorem 2.3.2 were used. This concludes the proof. □

The previous result shows that, given a fixed coalition size, in equilibrium a non-coalition firm benefits more from the coalition than a coalition firm does. This replicates the typical result of "the outsider earning higher profits than the insider does" in the theoretical literature on cartels and collusion (Selten 1973; d'Aspremont et al. 1983; Deneckere and Davidson 1985; Bos and Harrington 2010).

2.3.2 Coalition formation

This section endogenizes coalition size. Analyzing such a situation from the equilibrium perspective requires pinning down what makes a coalition a stable set (Neumann and Morgenstern 1944).

Following d'Aspremont et al. (1983), we define a coalition as stable when it satisfies the two conditions of internal and external stability.

Definition 2.3.4. *A coalition Γ is stable if it satisfies the following two conditions*

Internal stability: $\pi_k^d \geq \pi_{k-1}^c$ *(no incentive to leave the collusion)*

External stability: $\pi_k^c \geq \pi_{k+1}^d$ *(no incentive to join the collusion)*

This definition requires that a stable coalition is robust to unilateral deviation by coalition firms (internal stability) and non-coalition firms (external stability). A coalition Γ of size k is internally stable, whenever the profits of a non-coalition firm in the Bertrand equilibrium with a coalition of size $k-1$ do not exceed those of a coalition firm in the Bertrand equilibrium given Γ, i.e., whenever $\pi_k^d \geq \pi_{k-1}^c$. Next, a coalition Γ of size k is externally stable, whenever the profits of a coalition firm in the Bertrand equilibrium with a coalition of size $k+1$ do not exceed those of a non-coalition firm in the Bertrand equilibrium given Γ, i.e., whenever $\pi_k^c \geq \pi_{k+1}^d$.

We are now ready to state our main result, identifying the set of stable coalitions for any finite number of firms in the market. The result shows that a stable coalition always forms in equilibrium and that its size is always unique.

The following is our main result

Theorem 2.3.5. *The size of the stable coalition for the Bertrand model with collusion is*

$$k^* = \begin{cases} n, & n \in \{2,3,4,5\}, \\ 5, & n \in \{6,7,8,9,10,11\}, \\ 6, & n \in \{12,13,...\}. \end{cases}$$

Proof. Please refer to Section 2.4. □

There are three effects here: First, the larger the coalition the higher the price it charges (see Proposition 2.5.2) and, thus, the higher the profits of coalition members (see Proposition 2.5.3). This effect dominates in small markets with five or less firms. Second, also prices and profits of non-colluding firms increase in the size of the coalition (see Proposition 2.5.2 and 2.5.3). This effect

starts to dominate in intermediate markets with six to eleven firms. Finally, there is the effect of competitive pressure decreasing the profits of firms in the coalition. Indeed, profits of these firms decrease in large markets of twelve or more firms with the market size (Proposition 2.5.4).

An immediate implication of Theorem 2.3.5 is that there is a hard upper bound on the size of a stable coalition. Irrespective of the number of firms in the market, a stable coalition never involves more than six firms. When there are five or fewer firms, the grand coalition always forms as the only stable coalition. With more than five, but no more than eleven firms, a coalition is stable precisely when it has five members. In a market with twelve or more firms, a coalition is stable precisely when it is of size six.

2.4 Proof for Theorem 2.3.5

This section proves Theorem 2.3.5. The first part in Subsection 2.4.1 establishes existence. It shows that, for any finite n, there exists a stable coalition of size k where

$$k = \begin{cases} n, & n \in \{3, 4, 5\}, \\ 5, & n \in \{6, 7, 8, 9, 10, 11\}, \\ 6, & n \in \{12, 13, ...\}. \end{cases}$$

The second part in Subsection 2.4.2 then establishes the uniqueness. It proves that no coalition of different size is stable.

2.4.1 Existence

Theorem 2.4.1 establishes the existence for markets with $n \leq 17$ firms and Theorem 2.4.2 does so for markets with $n \geq 18$ firms.

Theorem 2.4.1. *For $n \leq 17$, the Bertrand model with collusion admits a stable coalition of size k, where*

$$k = \begin{cases} n, & n \in \{3,4,5\}, \\ 5, & n \in \{6,7,8,9,10,11\}, \\ 6, & n \in \{12,13,14,15,16,17\}. \end{cases}$$

Proof. To facilitate the analysis, in the following we view the coalition size k as a real number rather than an integer. By Theorem 2.3.2 we have

$$\pi_k^c = \frac{n \cdot (2n-k+2)^2}{Z_n(k)^2} \cdot (A-c)^2 \quad \text{and} \quad \pi_{k+1}^d = \frac{(2n+1)^2 \cdot (n-k)}{Z_n(k+1)^2} \cdot (A-c)^2.$$

So we can write

$$\frac{\pi_{k+1}^d}{\pi_k^c} = \frac{(2n+1)^2 \cdot (n-k) \cdot Z_n(k)^2}{n \cdot (2n-k+2)^2 \cdot Z_n(k+1)^2}.$$

So,

$$\frac{\pi_{k+1}^d}{\pi_k^c} < 1 \quad \Leftrightarrow \quad F_n(k) < \frac{n}{(2n+1)^2} \quad \Leftrightarrow \quad Q_n(k) < 0 \tag{2.3}$$

where $F_n(k) = \frac{(n-k)Z_n(k)^2}{(2n-k+2)^2 Z_n(k+1)^2}$ and $Q_n(k) = (n-k) \cdot (2n+1)^2 \cdot Z_n(k)^2 - n \cdot (2n-k+2)^2 \cdot Z_n(k+1)^2$.

By Footnote 2.3.1, for $k = 1$, all firms are in competition.

Then, by definition, a coalition of size $k > 1$ is stable if it is both externally and internally stable. It is internally stable if $\pi_k^d \geq \pi_{k-1}^c$. Using the relationship in

Equation 2.3, we can relate this internal stability to F_n and Q_n as follows:

$$\begin{aligned}&\text{coalition of size } k > 1 \text{ internally stable}\\ \Leftrightarrow &\frac{\pi_k^d}{\pi_{k-1}^c} \geq 1 \;\Leftrightarrow\; F_n(k-1) \geq \frac{n}{(2n+1)^2} \;\Leftrightarrow\; Q_n(k-1) \geq 0\end{aligned}$$

For external stability, note first that the grand coalition always satisfies it in a trivial way as no coalition of size $k = n+1$ can form. A coalition of size $k < n$ is externally stable if $\pi_{k+1}^d \geq \pi_k^c$. As such, we can relate this external stability via Equation 2.3 to F_n and Q_n as follows

$$\begin{aligned}&\text{coalition of size } k < n \text{ externally stable}\\ \Leftrightarrow &\frac{\pi_{k+1}^d}{\pi_k^c} < 1 \Leftrightarrow F_n(k) < \frac{n}{(2n+1)^2} \Leftrightarrow Q_n(k) < 0\end{aligned}$$

It follows that the grand coalition is a stable coalition if $Q_n(n-1) \geq 0$ holds. Furthermore, a coalition of size $k \in \{2, \ldots, n-1\}$ is stable if both $Q_n(k-1) \geq 0$ and $Q_n(k) < 0$ are satisfied.

Table 2.1: A "+" in cell (k, n) indicates that $Q_n(k) > 0$ and a "−" in cell (k, n) that $Q_n(k) < 0$.

k \ n	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
2		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
3			+	+	+	+	+	+	+	+	+	+	+	+	+	+
4				+	+	+	+	+	+	+	+	+	+	+	+	+
5					−	−	−	−	−	−	+	+	+	+	+	+
6						−	−	−	−	−	−	−	−	−	−	−

Table 2.1 lists the sign of $Q_n(k)$, for $1 \leq n \leq 17$ and $1 \leq k \leq 6$. Since

$Q_n(n-1) \geq 0$ for $2 \leq n \leq 5$, the grand coalition is a stable coalition then. For $6 \leq n \leq 11$, we have $Q_n(4) \geq 0$ and $Q_n(5) < 0$ such that there is a stable coalition of size $k = 5$ for such n. Finally, for $12 \leq n \leq 17$, we have $Q_n(5) \geq 0$ and $Q_n(6) < 0$ such that there is a stable coalition of size $k = 6$ for such n. This concludes the proof. □

Theorem 2.4.2. *For all $n \geq 18$, the Bertrand model with collusion admits a stable coalition of size $k = 6$.*

Proof. By definition, a coalition is stable if it is both externally and internally stable. Equation 2.3 then implies that a coalition of size $k = 6$ in a market with n firms is externally stable if $Q_n(5) \geq 0$ and internally stable if $F_n(6) < \frac{n}{(2n+1)^2}$.

For external stability, observe that

$$\begin{aligned} & Q_n(5) \\ = \; & (n-5) \cdot (2n^2 - n - 23)^2 \cdot (2n+1)^2 - n \cdot (2n-3)^2 \cdot (2n^2 - 2n - 34)^2 \\ = \; & 5 \cdot (4n^5 - 72n^4 + 265n^3 + 753n^2 - 4137n - 529) \\ > \; & 5 \cdot (72n^4 - 72n^4 + 265n^3 + 753n^2 - 4137n - 529) > 0. \end{aligned}$$

where the inequalities follow from $n \geq 18$. Note that this establishes that all markets with $n \geq 18$ firms admit a coalition of size $k = 6$ which is externally stable.

For internal stability, we compute that

$$F_n(6) = \frac{(n-6) \cdot Z_n(6)^2}{(2n-4)^2 \cdot Z_n(7)^2} = \frac{(n-6) \cdot (2n^2 + 4n - 6n - 36 + 2)^2}{(2n-4)^2 \cdot (2n^2 + 4n - 7n - 49 + 2)^2}$$

$$= \frac{(n-6)\cdot(n^2-n-17)^2}{(n-2)^2\cdot(2n^2-3n-47)^2}.$$

As such, we have that

$$\begin{aligned} F_n(6) < \frac{n}{(2n+1)^2} \quad \text{precisely when} \quad & (n-6)\cdot(n^2-n-17)^2\cdot(2n+1)^2 \\ < n\cdot(n-2)^2\cdot(2n^2-3n-47)^2. & \end{aligned}$$

To verify the latter inequality, consider that

$$\begin{aligned} & (n-6)\cdot(n^2-n-17)^2\cdot(2n+1)^2 - n(n-2)^2(2n^2-3n-47)^2 \\ = \; & -85n^5+226n^4+542n^3+1344n^2-15687n-1734 \\ < \; & -510n^4+226n^4+542n^3+1344n^2-15687n-1734 \\ = \; & -284n^4+542n^3+1344n^2-15687n-1734 \\ < \; & -1704n^3+542n^3+1344n^2-15687n-1734 \\ = \; & -1162n^3+1344n^2-15687n-1734 \\ < \; & -6972n^2+1344n^2-15687n-1734 \\ = \; & -5628n-15687n-1734 < 0. \end{aligned}$$

where the inequalities follow from $n \geq 7$. Note that this establishes that all markets with $n \geq 18$ firms admit a coalition of size $k = 6$ which is internally stable.

Hence, a coalition of size $k = 6$ in markets with $n \geq 18$ firms is both externally and internally stable and, thus, represents a stable coalition. This completes the proof. □

2.4.2 Uniqueness

Theorem 2.4.3. *For all finite n, the stable coalition in the Bertrand model with collusion has a unique size.*

Proof. First, consider the case of $n \in \{3, 4, 5\}$. By Table 2.1, $Q_n(k) \geq 0$ holds for each such n and all $k < n$. As such, Equation 2.3 implies that the grand coalition is the only externally stable coalition. Theorem 2.4.1 then implies that it is also the only stable coalition.

Next, consider the case of $n = 6$. By Table 2.1, it holds that $Q_6(k) > 0$, for $k = 2, 3, 4$, such that, by Theorem 2.4.1, any coalition of such size k is not internally stable. Furthermore, by Table 2.1, it holds that $Q_6(5) < 0$ such that, by Theorem 2.4.1, the grand coalition is not internally stable. Therefore, by Theorem 2.4.1, the only stable coalition in markets with 6 firms has size $k = 5$.

What remains to consider is the case of $n \geq 7$. To address this case, we analyze the properties of $F_n(k)$ from Equation 2.3 in more detail. Lemma 2.4.4 relates these properties of $F_n(k)$ to those of the polynomial $M_n(k)$ defined by

$$M_n(k) = -k^5 + Ak^4 + Bk^3 + Ck^2 + Dk + E,$$

where A, B, C, D, and E are constants in k, given by

$$\begin{aligned} A &= -2n - 4 \\ B &= 3n^2 + 3n + 3 \\ C &= 4n^3 + 21n^2 + 23n + 10 \\ D &= -4n^4 - 32n^3 - 58n^2 - 40n - 10 \\ E &= 12n^4 + 28n^3 + 16n^2 - 4n - 4. \end{aligned}$$

Then, Lemma 2.4.4 together with Lemma 2.4.5 show that $F_n'(n-1) < 0$ and $F_n'(k) = 0$ for at most one $k \in (1, n-1)$. Furthermore, Lemma 2.4.9 establishes that $F_n(2) > \frac{n}{(2n+1)^2}$ and Lemma 2.4.10 that $F_n(6) < \frac{n}{(2n+1)^2}$. Hence, it follows that also for markets with 7 or more firms, there exists exactly one k such that a coalition of this size is stable. This concludes the proof. □

Lemma 2.4.4. *For all $n \geq 3$ and all $k \in (0, n-1)$, it holds that*

(i) $F_n'(k) = 0 \quad \Leftrightarrow \quad M_n(k) = 0$

(ii) $F_n'(k) > 0 \quad \Leftrightarrow \quad M_n(k) > 0$

Proof. Since $F_n(k) = \frac{(n-k)\cdot Z_n(k)^2}{(2n-k+2)^2 \cdot Z_n(k+1)^2}$, we can write $F_n(k) = \frac{f(k)}{g(k)}$ where

$$f(k) = (n-k) \cdot Z_n(k)^2 \quad \text{and} \quad g(k) = (2n-k+2)^2 \cdot Z_n(k+1)^2.$$

Since $g(k) > 0$ for every $k \in (0, n-1)$, it suffices to show that, for

$$G(k) = f'(k) \cdot g(k) - f(k) \cdot g'(k)$$

it holds that $G(k) = 0$ precisely when $M_n(k) = 0$, and $G(k) > 0$ precisely when $M_n(k) > 0$. Since $Z_n(k) = 2n^2 + 4n - nk - k^2 + 2$ and, thus, $Z_n'(k) = -n - 2k$, we have

$$f'(k) = -Z_n^2(k) + 2\cdot(n-k)\cdot Z_n(k)\cdot Z_n'(k) = -Z_n(k)\cdot[Z_n(k) + 2\cdot(n-k)\cdot(n+2k)]$$

and

$$g'(k) = -2 \cdot (2n-k+2) \cdot Z_n(k+1) \cdot [Z_n(k+1) + (2n-k+2) \cdot (n+2k+2)].$$

So, factoring out a common factor of $L_k = Z_n(k) \cdot (2n - k + 2) \cdot Z_n(k + 1)$

$$f'(k) \cdot g(k) = -L_k \cdot \Big[(Z_n(k) + 2 \cdot (n - k) \cdot (n + 2k)) \cdot (2n - k + 2) \cdot Z_n(k + 1)\Big]$$

and

$$f(k) \cdot g'(k) = -L_k \cdot \Big[(n-k) \cdot Z_n(k) \cdot 2 \cdot (Z_n(k+1) + (2n-k+2) \cdot (n+2k+2))\Big].$$

Substitution into $G(k)$ yields

$$G(k) = f'(k) \cdot g(k) - f(k) \cdot g'(k) = -L_k \cdot \Big[A(k) - B(k)\Big] = L_k \cdot \Big[B(k) - A(k)\Big]$$

where

$$A(k) = (Z_n(k) + 2 \cdot (n - k) \cdot (n + 2k)) \cdot (2n - k + 2) \cdot Z_n(k + 1)$$

and

$$B(k) = (n - k) \cdot Z_n(k) \cdot 2 \cdot (Z_n(k + 1) + (2n - k + 2) \cdot (n + 2k + 2))$$

Because $L_k = Z_n(k) \cdot (2n - k + 2) \cdot Z_n(k + 1) > 0$ for every k, it suffices to prove that $B(k) - A(k) = 0$ for at most one $k \in (1, \ldots, n)$.

We argue that $M_n(k) = B(k) - A(k)$. We first collect terms of the form $Z_n(k) \cdot Z_n(k + 1)$ in $B(k) - A(k)$. In total we find

$$2 \cdot (n - k) - (2n - k + 2) = -k - 2 = -(k + 2).$$

The remaining part equals

$$2\cdot(n-k)\cdot Z_n(k)\cdot(2n-k+2)\cdot(n+2k+2)-2\cdot(n-k)\cdot(n+2k)\cdot(2n-k+2)\cdot Z_n(k+1).$$

Factoring out a factor of $2\cdot(n-k)\cdot(2n-k+2)$ yields

$$2\cdot(n-k)\cdot(2n-k+2)\cdot\Big[(n+2k+2)\cdot Z_n(k)-(n+2k)\cdot Z_n(k+1)\Big].$$

Note that

$$\begin{aligned}&(n+2k+2)\cdot Z_n(k)-(n+2k)\cdot Z_n(k+1)\\&=(n+2k)\cdot[Z_n(k)-Z_n(k+1)]+2\cdot Z_n(k)\\&=(n+2k)\cdot(n+2k+1)+2\cdot Z_n(k).\end{aligned}$$

So,

$$\begin{aligned}&B(k)-A(k)\\=\;&2\cdot(n-k)\cdot(2n-k+2)\cdot\Big[(n+2k)\cdot(n+2k+1)+2\cdot Z_n(k)\Big]\\&-(k+2)\cdot Z_n(k)\cdot Z_n(k+1)\\=\;&2(n-k)(2n-k+2)(2Z_n(k)+(n+2k)(n+k+1))\\&-(k+2)\cdot Z_n(k)\cdot Z_n(k+1)\\=\;&-k^5-2k^4n-4k^4+3k^3n^2+3k^3n+3k^3+4k^2n^3+21k^2n^2+23k^2n+10k^2\\&-4kn^4-32kn^3-58kn^2-40kn-10k+12n^4+28n^3+16n^2-4n-4\\=\;&M_n(k).\end{aligned}$$

This then implies the following equivalences

$$F_n'(k)=0\quad\Leftrightarrow\quad G(k)=0\quad\Leftrightarrow\quad M_n(k)=0$$

$$F_n'(k) > 0 \quad \Leftrightarrow \quad G(k) > 0 \quad \Leftrightarrow \quad M_n(k) > 0$$

This concludes the proof. □

Lemma 2.4.5. *Let $n \geq 4$. For $M_n(k)$, the following hold*

(i) $M_n(n-1) < 0$, and

(ii) there exists at most one $k \in (1, n-1)$, such that $M_n(k) = 0$.

Proof. Suppose that $n \geq 4$. For Part (i), observe that, for $k = n-1$, we have

$$\begin{aligned} M_n(n-1) &= -16n^3 + 10n^2 + 44n + 10 \\ &< -64n^2 + 10n^2 + 44n + 10 \\ &= -54n^2 + 44n + 10 \\ &\leq -216n + 44n + 10 \\ &= -172n + 10 \ < \ 0 \end{aligned}$$

where the inequalities follow from $n \geq 4$.

Part (ii): By Lemma 2.4.6 we know that $M_n^{(1)}(n-1) > 0$, and that $M_n^{(1)}(k) = 0$ for at most one $k \in (0, n-1)$. This implies that $M_n(k)$ is either a decreasing function, or, first decreasing and then increasing. Since $M_n(n-1) < 0$, this implies that $M_n(k) = 0$, for at most one $k \in (1, n-1)$. □

Lemma 2.4.6. *Let $n \geq 5$. For $M_n^{(1)}(k) = -5k^4 + 4Ak^3 + 3Bk^2 + 2Ck + D$, the following hold*

(i) $M_n^{(1)}(n-1) > 0$, and

(ii) there exist at most one $k \in (1, n-1)$, such that $M_n^{(1)}(k) = 0$.

Proof. Suppose that $n \geq 5$. For Part (i), note that we know that $M_n^{(1)}(k) = -5k^4 + 4Ak^3 + 3Bk^2 + 2Ck + D$. Then for $k = n - 1$, we have

$$\begin{aligned} M_n^{(1)}(n-1) &= 21n^3 - 60n^2 - 95n - 10 \\ &\geq 105n^2 - 60n^2 - 95n - 10 \\ &= 45n^2 - 95n - 10 \\ &> 225n - 95n - 10 \\ &= 130n - 10 \ > \ 0 \end{aligned}$$

where the inequalities follow from $n \geq 5$.

Part (ii): By Lemma 2.4.7 we know that $M_n^{(2)}(k)$ has at most one root and that $M_n^{(2)}(1) > 0$. Then $M_n^{(1)}(k)$ is either an increasing function, or, first increasing and then decreasing. Since $M_n^{(1)}(n-1) > 0$, this implies that $M_n^{(1)}(k) = 0$, for at most one k. □

Lemma 2.4.7. *Let $n \geq 3$. For $M_n^{(2)}(k) = -20k^3 + 12Ak^2 + 6Bk + 2C$, the following hold*

(i) $M_n^{(2)}(1) > 0$, and

(ii) there exist at most one $k \in (1, n-1)$ such that $M_n^{(2)}(k) = 0$.

Proof. Part (i): We know that $M_n^{(2)}(k) = -20k^3 + 12Ak^2 + 6Bk + 2C$. Hence, for $k = 1$, we have

$$\begin{aligned} M_n^{(2)}(1) &= -20 - 24n - 48 + 18n^2 + 18n + 18 + 8n^3 + 42n^2 + 46n + 20 \\ &= 8n^3 + 60n^2 + 40n - 30 \ > \ 0. \end{aligned}$$

where the inequality follows from $n \geq 3$.

Part (ii): By Lemma 2.4.8, $M_n^{(3)}(k)$ has at most one root and $M_n^{(3)}(n-1) < 0$. This implies that $M_n^{(2)}(k)$ is either a decreasing function, or, first increasing and then decreasing. In either case, since $M_n^{(2)}(1) > 0$, it follows that $M_n^{(2)}(k)$ has at most one root. □

Lemma 2.4.8. *Let $n \geq 3$. For $M_n^{(3)}(k) = -60k^2 + 24Ak + 6B$, the following hold*

(i) $M_n^{(3)}(n-1) < 0$, *and*

(ii) *there exists at most one $k \in (1, n-1)$ such that $M_n^{(3)}(k) = 0$.*

Proof. We prove part (i). For $k = n-1$, we have

$$\begin{aligned} M_n^{(3)}(n-1) &= -60 \cdot (n-1)^2 - (48n+96) \cdot (n-1) + 6 \cdot (3n^2+3n+3) \\ &= -90n^2 + 90n + 54 \\ &< -270n + 90n + 54 \\ &= -180n + 54 < 0. \end{aligned}$$

where the inequalities follow from $n \geq 3$.

We prove part (ii). We know that $M_n^{(3)}(k) = -60k^2 + 24Bk + 6C$. Therefore, $M_n^{(4)}(k) = -120k + 24B$. Since $B < 0$, it follows that $M_n^{(4)}(k) < 0$, for all $k \in [1, n-1]$. So, $M_n^{(3)}$ is a strictly decreasing function. Hence, $M_n^{(3)}(k) = 0$, for at most one $k \in (1, n-1)$. □

Lemma 2.4.9. *Let $n \geq 3$. Then $F_n(2) > \frac{n}{(2n+1)^2}$.*

Proof. For $k = 2$, we can compute that

$$F_n(2) = \frac{(n-2)(2n^2+4n-2n-4+2)^2}{(2n-2+2)^2(2n^2+4n-3n-9+2)^2} = \frac{(n-2)(n^2+n-1)^2}{n^2(2n^2+n-7)^2}.$$

So,

$$F_n(2) > \frac{n}{(2n+1)^2} \quad \text{precisely when}$$
$$(n-2)\cdot(n^2+n-1)^2\cdot(2n+1)^2 > n^3\cdot(2n^2+n-7)^2.$$

The latter inequality then follows from

$$\begin{aligned}
& (n-2)(n^2+n-1)^2(2n+1)^2 - n^3(2n^2+n-7)^2 \\
= \;& 8n^5 - 6n^4 - 34n^3 + 12n^2 - 3n - 2 \\
> \;& 24n^4 - 6n^4 - 34n^3 + 12n^2 - 3n - 2 \\
= \;& 18n^4 - 34n^3 + 12n^2 - 3n - 2 \\
> \;& 54n^3 - 34n^3 + 12n^2 - 3n - 2 \\
= \;& 20n^3 + 12n^2 - 3n - 2 \\
> \;& 60n^2 + 12n^2 - 3n - 2 \\
= \;& 72n^2 - 3n - 2 > 0.
\end{aligned}$$

where the inequalities follow from $n \geq 3$. □

Lemma 2.4.10. *Let $n \geq 7$. Then $F_n(6) < \frac{n}{(2n+1)^2}$.*

Proof. Suppose that $n \geq 7$ and $k = 6$. Since $Z_n(k) = 2n^2 + 4n - nk - k^2 + 2$, we have

$$\begin{aligned}
F_n(6) &= \frac{(n-6)\cdot Z_n(6)^2}{(2n-4)^2\cdot Z_n(7)^2} = \frac{(n-6)\cdot(2n^2+4n-6n-36+2)^2}{(2n-4)^2\cdot(2n^2+4n-7n-49+2)^2} \\
&= \frac{(n-6)\cdot(n^2-n-17)^2}{(n-2)^2\cdot(2n^2-3n-47)^2}
\end{aligned}$$

As such, we have that

$$F_n(6) < \frac{n}{(2n+1)^2} \quad \text{precisely when}$$
$$(n-6)\cdot(n^2-n-17)^2\cdot(2n+1)^2 < n\cdot(n-2)^2\cdot(2n^2-3n-47)^2.$$

The latter inequality then follows from

$$\begin{aligned}
& (n-6)\cdot(n^2-n-17)^2\cdot(2n+1)^2 - n(n-2)^2(2n^2-3n-47)^2 \\
= \;& -85n^5+226n^4+542n^3+1344n^2-15687n-1734 \\
< \;& -510n^4+226n^4+542n^3+1344n^2-15687n-1734 \\
= \;& -284n^4+542n^3+1344n^2-15687n-1734 \\
< \;& -1704n^3+542n^3+1344n^2-15687n-1734 \\
= \;& -1162n^3+1344n^2-15687n-1734 \\
< \;& -6972n^2+1344n^2-15687n-1734 \\
= \;& -5628n-15687n-1734 < 0.
\end{aligned}$$

where the inequalities follow from $n \geq 7$. This completes the proof. □

2.5 Discussion

Collusion reduces competition and hampers innovation, lowering welfare and consumer surplus in the short and in the long run.[5] This makes it important to detect and deter illegal acts of collusion, which often take the form of a cartel. Examples of real-world cartels run from the TFT-LCD Panel cartel (European Commission 2010), to the Long Island Asphalt cartel (Porter and Zona 1993), the Vitamin cartels (Igami and Sugaya 2022), the Pharmaceutical

[5] See Bos and Pot (2012) for the exception, i.e., a characterization of cartels which are beneficial for society.

Drug cartels (Starc and Wollmann 2025), and the Libor and the Forex Scandal (U.S. Department of Justice 2015; Federal Reserve 2015).

To identify when and where firms collude, competition authorities screen markets for such practices. But, detecting cartels is no easy task. The challenge lies in identifying which of the many markets and the many firms in these many markets warrant further scrutiny. This section explores where our results can aid this process.

2.5.1 The Number Effect(s)

Selten (1973) provides a theoretical foundation for the widespread belief that the number of firms in a market is a driver of competition. His analysis reveals that the chances of sustaining collusion in markets with many firms are minuscule. Empirical surveys confirm this to a large extent (Margaret C Levenstein and Valerie Y Suslow 2006; Margaret C. Levenstein and Valerie Y. Suslow 2011). Taken at face value, this implies that competition authorities should focus all their attention on markets with few firms. But, the surveys can only contain data on cartels that were detected. So, it is at least conceivable that small cartels in larger markets may exist as well.

Experiments by Huck, Normann, and Oechssler (2004) illustrate that competition drives out collusion when markets involve more than just a few firms. Other experimental studies (Devetag and Ortmann 2007; Fonseca and Normann 2012; Fonseca and Normann 2014) point towards one-off communication providing a means to sustain collusion in markets with more than just a few firms. Viewed from this perspective, Theorem 2.3.5 provides the theoretical argument that no market is too large for a small core to sustain collusion.

collusion / firms in market	this chapter (collusion size)	total probability of collusion in Selten (1973) (collusion size(s))
2	✓ (2)	✓ (2)
3	✓ (3)	✓ (3)
4	✓ (4)	✓ (4)
5	✓ (5)	.2221 (4 or 5)
6	✓ (5)	.2221 (4,5, or 6)
7	✓ (5)	.0103 (5, 6, or 7)
8	✓ (5)	.0006 (5, 6, 7, or 8)
9	✓ (5)	.0009 (6, 7, 8, or 9)
10 or 11	✓ (5)	below .0001
12 or more	✓ (6)	below .0001

Table 2.2: Stable coalitions depending on number of firms in market, ✓ means a stable coalition always forms

Wherein the analysis in this chapter and the one in Selten (1973) agree is that markets with few firms provide an environment conducive to collusion. Since it is always the grand coalition forming as a stable coalition in such markets, screening any firm would reveal this. For markets with more firms, the predictions of this chapter differ from those in Selten (1973) and Deneckere and Davidson (1985). Detecting collusion in such a market is more difficult, either because it is unlikely to occur (Selten 1973) or because it only involves few firms (Theorem 2.3.5). Screening such markets for colluding firms is the harder task. This suggests that implementing leniency programs for these markets may be an especially effective strategy. Table 2.2 summarizes the similarities and differences between the two number effects of firms colluding almost exclusively when there are few of them (Selten 1973) and there always being a few firms who collude (Theorem 2.3.5).

2.5.2 Effectiveness

Selten (1973) distinguishes between economically effective and ineffective collusion.[6] He does so, because his model also allows for ineffective collu-

[6]Selten (1973) refers to economically effective collusion as cartel arrangements and to ineffective ones as cartel agreements.

sion to occur in equilibrium. This is not the case in Deneckere and Davidson (1985), where as in the chapter at hand collusion is always effective. To streamline the discussion of the Bertrand equilibria from Section 2.3 here, we start by pinning down what renders collusion effective.

Definition 2.5.1. *Collusion is said to be effective if the equilibrium payoff of each colluding firm is strictly greater than that from the Bertrand game situation absent collusion.*

As to be expected, if firms collude this leads to all firms increasing their prices and the more so the larger the coalition of the colluding firms is. The next result formalizes this.

Proposition 2.5.2. *The Bertrand equilibrium prices p^d and p^c are both strictly increasing in the number of firms colluding.*

Proof. First, consider that for p^d from Theorem 2.3.1 it holds that

$$\frac{\partial p^d}{\partial k} = \frac{-(2n+1)(-n-2k)}{Z_n(k)^2}(A-c) = \frac{(2n+1)(n+2k)}{Z_n(k)^2}(A-c) > 0$$

Next, consider that for p^c from Theorem 2.3.1 it holds that

$$\begin{aligned}\frac{\partial p^c}{\partial k} &= \frac{-Z_n(k) - (2n-k+2)(-n-2k)}{Z_n(k)^2}(A-c) \\ &= \frac{-2n^2 - 4n + nk + k^2 - 2 + 2n^2 + 4nk - nk - 2k^2 + 2n + 4k}{Z_n(k)^2}(A-c) \\ &= \frac{-2n + 4nk - k^2 - 2 + 4k}{Z_n(k)^2}(A-c)\end{aligned}$$

$$= \frac{(2nk - 2n) + (nk - k^2) + (nk - 2) + 4k}{Z_n(k)^2}(A - c) > 0$$

where the last inequality follows from $n \geq k$, $k \geq 1$, and $n \geq 3$. □

There seems to be no economic incentive to sustain ineffective collusion given that competition is always an outside option. Indeed, our notion of coalition stability with its robustness to unilateral deviation suggests that a stable coalition in our model is effective. The following result confirms this intuition, showing formally that all firms stand to gain from the formation of a stable coalition.

Proposition 2.5.3. *The profit π^d of colluding firms strictly increases in the number of firms colluding. The profit π^c of a non-colluding firm also strictly increases in the number of firms colluding.*

Proof. First, we consider the profits of a coalition member, i.e., π^d. From Theorem 2.3.1 and Theorem 2.3.2, we know that

$$\pi^d = \frac{(n - k + 1)(2n + 1)^2}{Z_n^2(k)}(A - c)^2 = \frac{(n - k + 1)}{Z_n^2(k)}(2n + 1)^2(A - c)^2$$

where $Z_n(k) = 2n^2 + 4n - nk - k^2 + 2$.

Since the only part of π^d which depends on k is $G(k) = \frac{(n-k+1)}{(Z_n(k))^2}$, we next analyze how it changes with k.

$$\begin{aligned}\frac{\mathrm{d}G(k)}{\mathrm{d}k} &= \frac{-(Z_n(k))^2 + 2(Z_n(k))(n + 2k)(n - k + 1)}{(Z_n(k))^4} \\ &= \frac{Z_n(k)(2(n + 2k)(n - k + 1) - Z_n(k))}{(Z_n(k))^4}\end{aligned}$$

$$
\begin{aligned}
&= \frac{2(n+2k)(n-k+1) - Z_n(k)}{(Z_n(k))^3} \\
&= \frac{-4k^2 + 2nk + 4k + 2n^2 + 2n - 2n^2 - 4n + nk + k^2 - 2}{(Z_n(k))^3} \\
&= \frac{-3k^2 + 3nk + 4k - 2n - 2}{(Z_n(k))^3} \; > \; 0
\end{aligned}
$$

where the last inequality holds for all k, n satisfying $n > k > 1$, because $-3k^2 + 3nk + 4k - 2n - 2$ is a downward-opening formula, and its vertex $\frac{n}{2} + \frac{2}{3}$ lies within the interval. Hence, π^d increases over k.

Next, we consider the profits of a firm outside the coalition, i.e., π^c_k. Note that $\pi^c = (p^c - c) \cdot D^c$. By Proposition 2.5.2, p^c is strictly increasing in k. Also D^c is increasing in k. Hence, it follows that also π^c increases in k. □

Recall that a single colluding firm (see Footnote 2.3.1) describes the situation of all firms being in competition. As such, the first implication of Proposition 2.5.3 lies in each firm benefiting when competition turns to collusion. The second implication is that these benefits are increasing in the number of firms colluding. In other words, both coalition and non-coalition firms benefit from collusion, and the more so, the more collusion there is. Driving these increases in firms' surplus are increases in the prices at which each firm supplies its products (see Proposition 2.5.2) and our demand structure. These results contrast with those in Selten (1973), where ineffective coalitions may form, and those in Deneckere and Davidson (1985), where not all coalition firms may benefit from collusion.

On another note, combining Proposition 2.5.3 and the hard upper bound provides another way to address the negative effect collusion has on consumer surplus. This way lies in driving up competition. To mitigate the negative

effects of collusion, governments can exploit the hard upper bound of six by supporting firms' market entry (Starc and Wollmann 2025).

Proposition 2.5.4. *The profits of firms in a stable coalition are decreasing in the number of firms, n, in the market when $n \geq 12$.*

Proof. From Theorem 2.3.1 and Theorem 2.3.2, we know that

$$\pi^d = \frac{(n-k+1)(2n+1)^2}{Z_n^2(k)}(A-c)^2.$$

Note that to analyze how π^d changes with n, we can restrict our attention to $\frac{f(n)}{g(n)}$, where $f(n) = (n-k+1)(2n+1)^2$ and $g(n) = Z_n^2(k)$. Taking the derivative of $\frac{f(n)}{g(n)}$ with respect to n, then yields $\frac{f'(n)\cdot g(n) - f(n)\cdot g'(n)}{g^2(n)}$. Since $g^2(n) > 0$, it holds that

$$\frac{d\pi^d}{dn} < 0 \quad \Leftrightarrow \quad f'(n)\cdot g(n) - f(n)\cdot g'(n) < 0.$$

Since $Z_n(k) = 2n^2 + 4n - nk - k^2 + 2$, the first order derivatives are given by

$$f'(n) = (2n+1)^2 + 4(2n+1)(n-k+1) \quad and \quad g'(n) = 2Z_n(k)(4n+4-k).$$

But, then

$$\begin{aligned}
&f'(n)\cdot g(n) - f(n)\cdot g'(n) \\
=&f'(n)\cdot Z_n^2(k) - 2f(n)\cdot Z_n(k)\cdot(4n+4-k) \\
=&Z_n(k)[(2n+1)(2n+1+4(n-k+1))Z_n(k) - (n-k+1)(2n+1)^2(8n+8-2k)] \\
=&(2n+1)\cdot Z_n(k)[(2n+1+4(n-k+1))Z_n(k) - (n-k+1)(2n+1)(8n+8-2k)] \\
=&(2n+1)\cdot Z_n(k) \\
&* [(6n-4k+5)(2n^2+4n-nk+k^2+2) - (n-k+1)(16n^2+24n-4nk-2k+8)]
\end{aligned}$$

$$=(2n+1)\cdot Z_n(k)\cdot \underbrace{(-4k^3+6k^2n+3k^2+6kn^2+9kn+2k-4n^3-6n^2+2)}_{G(n,k)}$$

Since $(2n+1)\cdot Z_n(k) > 0$, we only need to analyze the term $G(n,k)$. Since, for all $n \geq 12$, the size of the stable coalition is $k^* = 6$, we can restrict attention to

$$G(n,6) = -4n^3 + 30n^2 + 270n - 742.$$

Since $G(n,6) = 0$ only for $n < 12$, it holds that $G(n,6) < 0$, for all $n \geq 12$. Hence, for $n \geq 12$, the profits π^d are a decreasing function. □

2.6 Conclusion

This chapter analyzes the relationship between collusion formation and the number of firms in a market, assuming a Bertrand model with collusion. Similar to Selten (1973), with few firms (less than 5 firms) the grand coalition always forms. However, in contrast to Selten (1973), a stable coalition also forms in markets with more than five firms. In particular, when the number of firms is between five and eleven, the size of the stable coalition equals five. For oligopolies with twelve or more firms, the size of the stable coalition equals six. Contrasting both Selten (1973) and Deneckere and Davidson (1985), our analysis identifies a hard upper bound on the size of a stable coalition, showing that no such coalition ever exceeds a size of six. Regardless of the number of firms in a market, collusion affects prices and profits in this market.

3

Collaboration-Driven Mechanisms

This chapter is based on: Zhilin Xu, Hao Sun, and Weibin Han. "A collaboration-driven mechanism for AI diagnose with multiple requesters under incomplete information". In: *Computer Networks* 220 (2023), p. 109472.

In this chapter, we design a collaboration-driven mechanism with multiple requesters cooperating to collect massive accurate data. Under the process of this incentive mechanism, two issues have been solved, how to guarantee the accuracy level of data and how to encourage requesters to cooperate. A novel payment policy with which the most accurate data receive the highest payment is presented to ensure the accuracy level of the data. A Stackelberg game with requesters as leaders, which describes the interactions between requesters and participants, is presented. To reduce the computational complexity of the Stackelberg equilibrium with multiple leasers, the concept of "team" is introduces, in which each team represents one kind of data. Based on their data requirements, requesters then choose which teams they want to have information from. Requesters within the same team request data jointly. Therefore, we can treat each team separately. Each team now functions as a single leader in the Stackelberg game. A cost allocation method inspired by the Separable Costs Remaining Benefits method (Driessen 2013) is proposed to ensure the increase of requesters' utilities when collecting data as a team. However, since the participant's costs for collecting and uploading data is unknown for requesters, the game mentioned above is incomplete information dynamic MCS game. Hence, Q-learning is used to estimate unknown information from participants to requesters. Simulations are presented to prove that Q-learning can obtain equilibrium in the case of incomplete information.

3.1 Introduction

With the popularity of mobile devices such as smart phones, GPS devices, and computers, Mobile Crowdsensing (MCS), has been able to collect massive and complex data. There are still several issues that should be considered to guarantee successful data collection, such as safety issues, utility issues, and privacy issues. With regard to safety issues, due to the potentially selfish nature of participants, the MCS system is vulnerable to fake sensing attacks by participants who upload fake data to improve utilities and save costs (L. Xiao, J. Liu, et al. 2015). With a view of both requesters' and participants' rationalities, neither of them will join the MCS system at a loss. With regard to privacy issues, the uploaded data may contain sensitive information like locations and personal information. Participants may not want to be part of the system to avoid leaking privacy (L. Xiao, D. Jiang, et al. 2018).

In China alone, there are many requesters who use AI diagnosis. Therefore, some researchers take multiple requesters into account, for example S. Chen, M. Liu, and X. Chen (2016) use a truthful double auction to consider mutual competition where different requesters always compete for potential MCS task performers. Requesters will then compete for data, and this may lead to several problems. One problem is for example that, in competition, requesters with the highest payment scheme get the most accurate data, while other requesters just have to settle for inaccurate data. This then hurts endusers because the services provided by weaker requesters are of inferior quality.

We propose an incentive mechanism where requesters cooperate with each other to obtain accurate data. Requesters can request in the MCS system for different types of data together with a payment scheme to compensate par-

ticipants for the costs of data provision. In general the competition between requesters is difficult to analyze. Instead we analyze teams, where each team represents one type of data. Requesters can decide which teams to join based on their data requirements. Participants can choose which type of data they want to provide. Under the assumption of complete information, we compute the Stackelberg equilibrium in this setting.

In the Stackelberg equilibrium of the complete information game, the strategy of each requester is dependent on the costs of the participants. Since it is not realistic to assume that requesters have this information about participants' costs, we next analyze the corresponding incomplete information game. We use Q-learning to let requesters estimate participants' costs. At each iteration, the Q-learning algorithm estimates participants' costs based on information acquired from previous rounds. Simulation results show that in the incomplete information game, the strategies generated by the Q-learning algorithm converges to the Stackelberg equilibrium of the complete information game.

The contributions of this chapter can be summarized as follows:

- We propose a cooperative incentive based mobile crowdsensing scheme to improve the accuracy of sensing data provided by participants.
- To mitigate computational complexities due to data heterogeneity, we introduce the notion of team. Each team represents a specific type of data. Instead of calculating one Stackelberg equilibrium with multiple leaders, for each team we compute a Stackelberg equilibrium with one leader. These equilibria can be calculated in polynomial time.
- In the Stackelberg game with incomplete information, Q-learning is used to help requesters estimate the costs of the participants. Simula-

tions confirm that the Q-learning process converges to the Stackelberg equilibrium of the complete information game.

The remainder of this chapter is organized as follows. The Stackelberg MCS model is described in Section 3.2. Section 3.3 introduces the sub-Stackelberg MCS model. Section 3.4 presents the simulation results on the incomplete information game. Finally, Section 3.5 summarizes the contributions of this chapter.

3.2 MCS model

An MCS server aims to help the m requesters to recruit some of the n participants located in an area of interest to gather sensing data and thus establish an MCS application, as shown in Fig. 3.1. Requesters choose their payment policies and broadcast a recruiting message that lists the payment of each requester according to sensing accuracy level of data. On the basis of the MCS payment policies, each selfish participant chooses its sensing efforts (time and energy each participant is willing to provide for the task), which in turn determines the sensing accuracy level of the data.

3.2.1 Model definition

Let $M = \{1, 2, ..., m\}$ be the set of requesters and $N = \{1, 2, ..., n\}$ denote the set of participants. $Q = \{1, 2, ..., q\}$ is the set of data types that are required by requesters in the system. The basic model is now defined as follows.

Definition 3.2.1. *An $(m + n)$-person MCS game is a tuple*

$$Z =< M, N, Q, X, Y, T, \bar{u_r}, \bar{u_p} >,$$

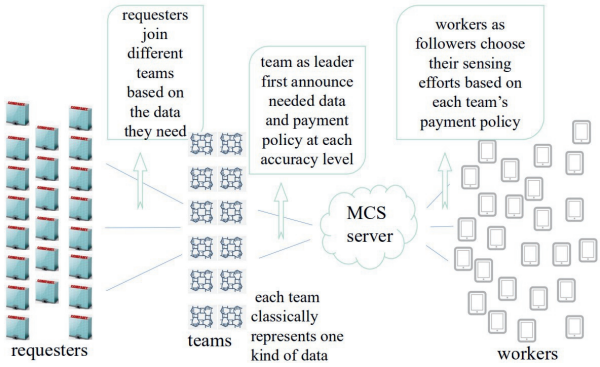


Figure 3.1: Illustration of an MCS system

- *where M denotes the set of leaders which are the requesters, and N the set of followers which are the participants,*
- *Q is the set of data types. Each type has accuracy levels in $[0, 1]$.*
- *$X = \{0, 1\}^{M \times Q}$ is a set of binary vectors. An element $x \in X$ is a vector $x = (x_{ik})_{i \in M, k \in Q}$. If $x_{ik} = 1$ then this indicates that requester i needs data type k.*
- *$Y = [0, +\infty)^{N \times M \times Q}$ is the set of unit prices that the requesters offers to participants. So for $y \in Y$, $i \in M$ $j \in N$, $k \in Q$, y_{ijk} is the price that requester i pays if participant j provides data type k with accuracy level 1.*
- *$T = [0, 1]^{N \times M \times Q}$ is the set of accuracy levels that the participants choose from. To be precise, for each $j \in N$, $i \in M$, and $k \in Q$, $[0, 1]$ is the set of participant j's sensing strategies (the set of possible accuracy levels) for uploading data of type k to requester i.*
- *$\bar{u}_p = (\bar{u}_{pj})_{j \in N}$ is the vector of utility functions for participants. For each*

$j \in N$, $y \in \mathbb{R}$, $t \in [0,1]^Q$,

$$\bar{u}_{pj}(y,t) = y - \sum_{k=1}^{q} c_{jk}(t_{jk}),$$

where y is the amount of money participant j receives in total, and $c_{jk}(t_{jk})$ is the nonnegative costs of participant j for collecting data of type k with accuracy level $t_{jk} \in [0,1]$.

- $\bar{u}_r = (\bar{u}_{ri})_{i \in M}$ *is the vector of utility functions for requesters. For each $i \in M$, $x \in X$, $y \in \mathbb{R}$, $t \in [0,1]^{N \times Q}$, requester i's utilities are defined as*

$$\bar{u}_{ri}(x,y,t) = \sum_{k=1}^{q} \sum_{j=1}^{n} x_{ik} \cdot g_{ik}(1) \cdot t_{jk} - y,$$

where x_{ik} is whether requester i request data type k, y is the total amount of money requester i pays, and $g_{ik}(1)$ means the value for requester i of data type k with accuracy level 1.

Remark: The vector $x \in X$ is exogenously chosen by requesters. The requesters know which types of data they need for their operation. The vector $y \in Y$ is the strategic choice by the requesters, and the vector $t \in T$ is the strategic choice by the participants. The utility functions are also exogenously given.

Given profile $y \in Y$ and $t \in T$, the payment paid by requester i to participant j can be computed as

$$p_{ij}(y,t) = \sum_{k \in Q} y_{ijk} \cdot t_{ijk}.$$

3.3 Stackelberg MCS model

For the given MCS game Z, it is difficult to compute the equilibrium. We introduce the notion of "teams" where a team represents a certain type of data. This way we create a Stackelberg game. In this game, first each requester chooses to join the teams based on data they need. At this point, it is clear for each requester which teams it joins. Note that a single requester can join multiple teams. Then participants get recruited by the teams instead of requesters. In this model we are going to compute the equilibrium which is easier to compute than the equilibrium of the MCS game Z.

3.3.1 Classification of requesters

We introduce a concept of team which uniquely defines a data type. Each team classically represents one kind of data.

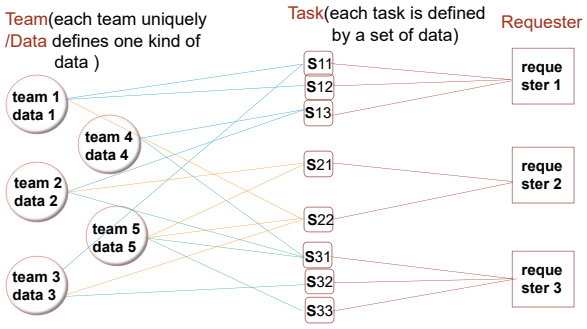


Figure 3.2: An example about the reassignment of requesters

In the example in Fig. 3.2, there are three requesters ($M = \{1, 2, 3\}$) and five teams ($Q = \{1, 2, 3, 4, 5\}$). If there is a line between the team $k \in Q$ and the data required by requester $i \in M$, then it means requester i belongs to task k,

$x_{ik} = 1$. As we can see, requester 1 belongs to team 1, 2, 3, and 4, requester 2 belongs to team 1, 2, 3, and 5, and requester 3 belongs to team 2, 3, 4, and 5.

3.3.2 Model definition

The Stackelberg model is now defined as follows.

Definition 3.3.1. *For the given MCS game*

$$Z = \langle M, N, Q, X, Y, T, \bar{u}_r, \bar{u}_p \rangle,$$

the requester set M is classified into q teams according to the data requirement matrix X. For each $k \in Q$, let $M_k = \{i \in M : x_{ik} = 1\}$ denotes the set of requesters belonging to team k. Then, the corresponding Stackelberg MCS game for data type k is defined as

$$Z_k = \langle \{k\}, N, W, T, u_k, u_p \rangle.$$

- *$\{k\}$ denotes the aggregated leader formed by the requesters in M_k, and N denotes the set of followers which are participants.*
- *$T = [0, 1]^{1 \times N}$ is the strategy space of participants for collecting data type k, where $t_{jk} \in [0, 1]$ represents the sensing accuracy level of data type k from participant j.*
- *$W = [0, +\infty)^{1 \times N}$ is the strategy space of team k. For each $w_{jk} \in [0, +\infty)$, and $t_{jk} \in [0, 1]$ the payment policy is defined as*

$$y_{jk}(w_{jk}, t_{jk}) = \frac{G_{jk}(t_{jk})}{1 + G_{jk}(t_{jk})} \cdot w_{jk} \cdot t_{jk},$$

where $G_{jk}(t_{jk}) = \frac{\sum_{i=1}^{m} g_{ij}^{k}(t_{jk})x_{ik}}{m}$ *represents the average value of sensing data uploaded by participant* j *to team* k *with accuracy level* t_{jk}*, and* $g_{ij}^{k}(t_{jk}) = g_{ik}(1) \cdot t_{jk}$*, the value for requester* i *of data type* k *from participant* j *with accuracy level* t_{jk}

- $u_p = (u_{pj})_{j \in N}$ *is the vector of utility functions for participants for collecting data type* k*. For each* $j \in N$*,* $w_{jk} \in [0, +\infty)$*,*

$$u_{pj}(w_{jk}, t_{jk}) = y_{jk}(w_{jk}, t_{jk}) - c_{jk}(t_{jk}),$$

where $c_{jk}(t_{jk})$ *is the sensing cost of participant* j *for collecting data of type* k *with accuracy level* t_{jk}*.*

- u_k *is the vector of utility function for team* k*. For each* $t_{jk} \in [0, 1]$*,* $w_{jk} \in [0, +\infty)$

$$u_k(w_{jk}, t_{jk}) = G_{jk}(t_{jk}) - y_{jk}(w_{jk}, t_{jk}),$$

where $G_{jk}(t_{jk})$ *is the data value for team* k *from participant* j *with data accuracy level* t_{jk}*.*

We assume in this chapter that the team k cannot precisely validate the accuracy level of the data provided by participants. In this chapter we assume the probability that the server estimate accuracy level t as $\hat{t}$ follows the following conditional probability distribution function.

$$\begin{aligned} Pr(\hat{t}|t) =& 1 - a, && \hat{t} = t \neq -1, \\ Pr(\hat{t}|t) =& 1, && \hat{t} = t = -1, \\ Pr(\hat{t}|t) =& a\mathrm{e}^{-ax}, && \hat{t} \neq t, 1 \geq x \geq 0, \end{aligned} \tag{3.1}$$

where $a \in (0,1)$ means the error rate of MCS server, and $x \in [0,1]$ is an exogenous parameter.

Remark: Due to the incomplete information, the MCS server has to apply an advanced and efficient sensing quality evaluation algorithm, such as the classification algorithm in (D. Peng, Wu, and G. Chen 2015), and the majority voting algorithm in (Huang, Kanhere, and Hu 2014), to estimate the accuracy level, where $\hat{t}_{jk}$ is the estimated when the actual accuracy level is t_{jk}. However, we abstract away from such estimation algorithm and simply take the probability distribution function as given.

Taking the possibility that MCS server evaluates actual sensing accuracy level t as $\hat{t}$ into account, team k's and participant j's utility function can be transformed as

$$v_{pj}(w,t) = \mathbb{E}_{Pr(\hat{t}|t)}[u_{pj}(t)] = \alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x - c_{jk}(t),$$

and

$$v_k(w,t) = \mathbb{E}_{Pr(\hat{t}|t)}[u_k(t)] = \sum_{i=1}^{m}\sum_{j=1}^{n}(\sigma g_{ij}^k(t) x_{ik}) - \sum_{j=1}^{n} \alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x,$$

where $Pr(\hat{t}|t)$ is the probability mass function that MCS server predicates actual sensing accuracy level t as $\hat{t}$ as described by eq. 3.1, $\alpha \in (0,1)$ describes the decrease

In game $Z_k, k \in Q$, team k will first request data with type k from the MCS server and determine the associated payment policy as leader. Next the participants as followers will provide data to the MCS server based on the payment policies.

3.3.3 Strategies of teams and participants

Definition 3.3.2. *In the given sub-Stackelberg MCS game*

$$Z_k = \langle \{k\}, N, W, T, u_k, u_p \rangle,$$

the set $T(w_{jk}) \subset [0,1]$ *defined for each* w_{jk} *by*

$$T(w_{jk}) = \{t_{jk} \in [0,1] : v_{pj}(t_{jk}) \geq v_{pj}(t'_{jk}), \forall t'_{jk} \in [0,1]\} \tag{3.2}$$

is the optimal response (rational reaction) set of participant j *to the strategy* w_{jk} *of leader (team* k*).*

A strategy w^*_{jk} *is called a sub-Stackelberg MCS equilibrium for the leader (team* k*) , if*

$$\min_{t^*_{jk} \in T(w^*_{jk})} v_k = \max_{w_{jk}} \min_{t_{jk} \in T(w_{jk})} v_k. \tag{3.3}$$

The quantity $v_k(t^*_{jk})$ is the sub-Stackelberg MCS utility of the leader (team k). Any element $t^*_{jk} \in T(w^*_{jk})$ is an optimal strategy for the follower (participant j) that is in equilibrium with y^*_{jk}. The quantity $v_{pj}(t^*_{jk})$ is the sub-Stackelberg MCS utility of the follower (participant j). The pair (w^*_{jk}, t^*_{jk}) is a sub-Stackelberg MCS solution for the game $Z_k, k \in Q$ with team k as the leader.

3.3.4 Payment policy

In the sub-Stackelberg MCS game $Z_k, k \in Q$, if the payment policy $y_{jk}(t)$ meets the following three standards,

1. when uploading faked sensing reports, participant's utility is non-positive $v_{pj}(0) < 0$,

2. with the growth of the accuracy level of reports, participant's utility will also increase $\frac{\mathrm{d}v_{pj}(t)}{\mathrm{d}t} > 0$,

3. the more accuracy the reports are, the more payments participants can gain $\frac{\mathrm{d}y_{jk}(t)}{\mathrm{d}t} > 0$.

The payment policy $y_{jk}(t) = \frac{G_{jk}(t)}{1+G_{jk}(t)}wt$ is obvious valid.

As to standard 1, we get $v_{pj}(0) = \alpha \int_0^1 y_{jk}(x) Pr(\hat{t} \mid 0)\mathrm{d}x - c_{jk}(0)$.

It is obvious that $y_{jk}(0) = 0$, hence

$$
\begin{aligned}
&v_{pj}(0) = \alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|0)\mathrm{d}x - c_{jk}(0) \\
&= (1-a)\alpha y_{jk}(0) + \lim_{b\to 0} \int_0^b a\mathrm{e}^{-ax}\alpha y_{jk}(x)\mathrm{d}x - c_{jk}(0) = 0 - c_{jk}(0) < 0.
\end{aligned}
$$

Therefore $v_{pj}(0) < 0$.

With regard to standard 2, we could refer to the proof of Theorem 3.3.3.

$$
\begin{aligned}
v_{pj}(t1) - v_{pj}(t2) = \alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|t1)\mathrm{d}x - c_{jk}(t1) \\
- (\alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|t3)\mathrm{d}x - c_{jk}(t3)),
\end{aligned}
$$

hence,

$$
\begin{aligned}
&v_{pj}(t1) - v_{pj}(t2) \\
&= (1-a)\alpha y_{jk}(t1) + \lim_{h\to t1} \int_0^h a\mathrm{e}^{-ax}\alpha y_{jk}(x)\mathrm{d}x
\end{aligned}
$$

$$
\begin{aligned}
&+ \lim_{d\to t1}\int_d^1 ae^{-ax}\alpha y_{jk}(x)\mathrm{d}x - c_{jk}(t1)\\
&- [(1-a)\alpha y_{jk}(t2) + \lim_{h\to t2}\int_0^h a\mathrm{e}^{-ax}\alpha y_{jk}(x)\mathrm{d}x\\
&+ \lim_{d\to t2}\int_d^1 ae^{-ax}\alpha y_{jk}(x)\mathrm{d}x - c_{jk}(t2)]\\
&= (1-a)\alpha t1\frac{Kt1}{1+Kt1}w + a\alpha\lim_{h\to t1}\int_0^h wx\frac{Kx}{1+Kx}\mathrm{e}^{-ax}\mathrm{d}x\\
&+ a\alpha\lim_{d\to t1}\int_d^1 wx\frac{Kx}{1+Kx}\mathrm{e}^{-ax}\mathrm{d}x - c_{jk}(t1) - [(1-a)\alpha t\frac{Kt2}{1+Kt2}w\\
&+ a\alpha\lim_{h\to t2}\int_0^h wx\frac{Kx}{1+Kx}\mathrm{e}^{-ax}\mathrm{d}x\\
&+ a\alpha\lim_{d\to t2}\int_d^1 wx\frac{Kx}{1+Kx}\mathrm{e}^{-ax}\mathrm{d}x - c_{jk}(t2)].
\end{aligned}
$$

If we want to make sure $v_{pj}(t1) - v_{pj}(t2) > 0, t1 > t2$, then

$$w > \frac{c_{jk}(t1) - c_{jk}(t2)}{(1-a)\alpha\frac{K}{1+K}}.$$

Obviously, the payment policy in Theorem 3.3.3 can make the above formula valid.

No doubt that the payment policy increases as t increases. In summary, the payment policy in this chapter is valid.

We first consider the optimal response (rational reaction) set of follower (participant $j \in N$) in Theorem 3.3.3.

Theorem 3.3.3. *In the sub-Stackelberg MCS game $Z_k, k \in Q$, the rational partici-*

pant j's optimal response set is $T(y_{jk}) = \{1\}$, if

$$w > \frac{c_{jk}(1) - c_{jk}(0)}{(1-a)\alpha\frac{K}{1+K}},$$

where a means estimated error rate of MCS server based on eq.3.1 and α explains the decrease of each requester's costs due to cooperation.

Proof. For the MCS systems, the data accuracy level must be 1. Data with accuracy level $t \in [0, 1)$ may lead to severe damages like misdiagnosis for AI diagnose MCS applications. The problem is, selfish participants may upload inaccurate data for more utilities. Therefore, we have to adjust payment policy to make sure participants' utilities meet maximum when $y_{jk} = 1$.

$$T(w_{jk}) = \{t_{jk} \in [0, 1] : v_{pj}(t_{jk}) \geq v_{pj}(t'_{jk}), \forall t'_{jk} \in [0, 1]\} = \{1\},$$

that is to say

$$v_{pj}(1) > v_{pj}(t), t \in [0, 1),$$

thus,

$$\begin{aligned}
&v_{pj}(1) - v_{pj}(t) \\
&= \alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|1)\mathrm{d}x - c_{jk}(1) - (\alpha \int_0^1 y_{jk}(x) Pr(\hat{t}|t)\mathrm{d}x - c_{jk}(t)) \\
&= (1-a)\alpha y_{jk}(1) + \lim_{b\to 1} \int_0^b a\mathrm{e}^{-ax}\alpha y_{jk}(x)\mathrm{d}x - c_{jk}(1) - [(1-a)\alpha y_{jk}(t) \\
&+ \lim_{h\to t} \int_0^h a\mathrm{e}^{-ax}\alpha y_{jk}(x)\mathrm{d}x + \lim_{d\to t} \int_d^1 ae^{-ax}\alpha y_{jk}(x)\mathrm{d}x - c_{jk}(t)] \\
&= (1-a)\alpha\frac{K}{1+K}w + a\alpha \lim_{b\to 1} \int_0^b w\frac{Kx}{1+Kx}\mathrm{e}^{-ax}x\mathrm{d}x - c_{jk}(1) - [(1-a)\alpha t\frac{Kt}{1+Kt}w
\end{aligned}$$

$$+ a\alpha \lim_{h\to t} \int_0^h wx \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x + a\alpha \lim_{d\to t} \int_d^1 wx \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x - c_{jk}(t)].$$

As to $\lim_{b\to 1} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x$, we have

$$\begin{aligned}
&\lim_{b\to 1} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x = \lim_{b\to 1} \int_0^b x(1 - \frac{1}{1+Kx}) \mathrm{e}^{-ax} \mathrm{d}x \\
=& \lim_{b\to 1} (\int_0^b x\mathrm{e}^{-ax} \mathrm{d}x - \int_0^b \frac{x}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x) \\
=& \lim_{b\to 1} (\int_0^b x\mathrm{e}^{-ax} \mathrm{d}x - \frac{1}{K} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x),
\end{aligned}$$

hence

$$\lim_{b\to 1} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x = \lim_{b\to 1} (\int_0^b x\mathrm{e}^{-ax} \mathrm{d}x - \frac{1}{K} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x),$$

it is obvious that

$$\frac{1+K}{K} \lim_{b\to 1} \int_0^b x \frac{Kx}{1+Kx} \mathrm{e}^{-ax} \mathrm{d}x = \lim_{b\to 1} \int_0^b x\mathrm{e}^{-ax} \mathrm{d}x,$$

where

$$\int \mathrm{e}^{-ax} x dx = -\frac{1}{a} \int x d\mathrm{e}^{-ax} = +\{-\frac{1}{a}(\mathrm{e}^{-ax} x + \frac{1}{a} \mathrm{e}^{-ax})\},$$

Therefore,

$$\begin{aligned}
&v_{pj}(1) - v_{pj}(t) \\
&= (1-a)w\alpha \frac{K}{1+K} - w\alpha \frac{K}{1+K} (\mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a} + \frac{1}{a}) - c_{jk}(1) - [(1-a)tw\alpha \frac{Kt}{1+Kt} \\
&- aw\alpha (\frac{K}{a(1+K)} * [t\mathrm{e}^{-at} - \frac{1}{a}\mathrm{e}^{-at} + \frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a} - t\mathrm{e}^{-at} + \frac{1}{a}\mathrm{e}^{-at}]) - c_{jk}(t)] \\
&= (1-a)w\alpha \frac{K}{1+K} - w\alpha \frac{K}{1+K} (\mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a} + \frac{1}{a}) - c_{jk}(1) - [(1-a)tw\alpha \frac{Kt}{1+Kt}
\end{aligned}$$

$$
\begin{aligned}
&- aw\alpha(\frac{K}{a(1+K)}(\frac{1}{a}+\mathrm{e}^{-a}-\frac{1}{a}\mathrm{e}^{-a}))-c_{jk}(t)] \\
&=(1-a)w\alpha\frac{K}{1+K}-w\alpha\frac{K}{1+K}(\frac{1}{a}+\mathrm{e}^{-a}-\frac{1}{a}\mathrm{e}^{-a})-c_{jk}(1)-(1-a)wt\alpha\frac{Kt}{1+Kt} \\
&+w\alpha\frac{K}{1+K}(\frac{1}{a}+\mathrm{e}^{-a}-\frac{1}{a}\mathrm{e}^{-a})+c_{jk}(t) \\
&=(1-a)w\alpha(\frac{K}{1+K}-t\frac{Kt}{1+Kt})-c_{jk}(1)+c_{jk}(t) \\
&>0.
\end{aligned}
$$

As a result,

$$
w>\frac{c_{jk}(1)-c_{jk}(t)}{(1-a)\alpha(\frac{K}{1+K}-t\frac{Kt}{1+Kt})}.
$$

In addition, for whatever $t\in[0,1)$, the above equation must hold,

$$
w>\frac{c_{jk}(1)-c_{jk}(0)}{(1-a)\alpha\frac{K}{1+K}}.
$$

□

Since $T(w_{jk})=\{1\}$ is follower's (participant j's) only rational strategy, it is safe to say that $t_{jk}=1$ is a sub-Stackelberg MCS equilibrium for the follower (participant j).

Through Theorem 3.3.3, the utility of every team $k\in Q$ will decrease with the growth of the error rate a. The growth of the error rate a indicates the increasing likelihood that an MCS server will mistake accurate data as inaccurate. As shown in Fig. 3.3, participants may feel reluctant to upload accurate data which may hurt teams' utilities.

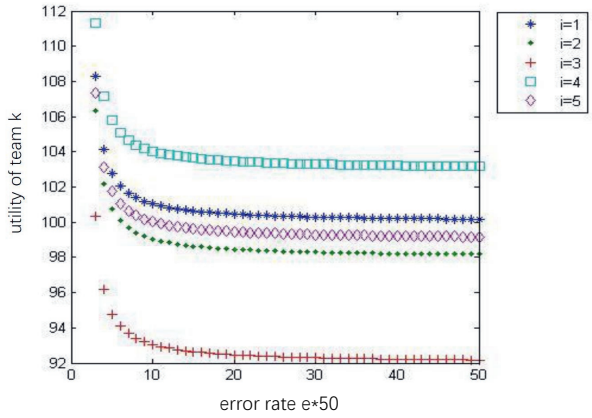


Figure 3.3: Utility of team k as given by Theorem 3.3.3, with $m = 5$, $n = 10$, $q = 5$, and $G_{jk}(1) \sim U(0, 2, 10, 5)$ ($U(0, 2, 10, 5)$ is a 10*5 random matrix whose elements are between 0 and 2) vs. error rate

On account of Theorem 3.3.3, only the willingness of participants is guaranteed. If requesters do not get the maximum utilities from the most accurate data, some selfish requesters may settle for less accurate data in exchange for more utilities, ignoring that this behaviour could lead to servere misdiagnosis. To solve this phenomenon, sub-Stackelberg MCS equilibrium of the leader is used to promise leader's satisfaction with the most accurate data.

Theorem 3.3.4. *For the sub-Stackelberg MCS game $Z_k, k \in Q$ if*

$$w^* = \min w,$$

then,

$$y_{jk}^* = \frac{G_{jk}(1)}{1 + G_{jk}(1)} w^*,$$

is the sub-Stackelberg MCS equilibrium for the leader (team k).

Proof. From Theorem 3.3.3, no doubt that for any y_{jk}, we have $T(y_{jk}) = \{1\}$. Therefore

$$v_k^*(1) = \max_{y_{jk}} v_k(1).$$

Combined with Theorem 3.3.3

$$\begin{aligned} v_k(1) &= \sum_{i=1}^{m}\sum_{j=1}^{n}(\sigma g_{ij}^k(1)x_{ik}) - \sum_{j=1}^{n}\alpha\int_0^1 y_{jk}(\hat{t})Pr(\hat{t}|1)\mathrm{d}\hat{t} \\ &= \sum_{j=1}^{n}(\sigma m K) - \sum_{j=1}^{n}[(1-a)w\alpha\frac{K}{1+K} + w\alpha\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a})]. \end{aligned}$$

As discussed before, the error rate a from eq. 3.1 belongs to $(0,1)$, thus $\mathrm{e}^{-a} \in (\frac{1}{\mathrm{e}}, 1)$. Obviously, $1 - \mathrm{e}^{-a} \in (0, 1 - \frac{1}{\mathrm{e}})$. It is safe to conclude that $(1 - a + \frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a}) > 0$. Undoubtedly, $-w\sum\limits_{j=1}^{n}\alpha\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a})) \leq 0$, which indicates that the smaller w is, the larger $-w\sum\limits_{j=1}^{n}\alpha\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a}))$ is. Thus, it is safe to say that $w^* = \min w$. □

Therefore, the quantity

$$v_k(t_{jk}^*) = \sum_{i=1}^{m}\sum_{j=1}^{n}(\sigma g_{ij}^k(t)x_{ik}) - \sum_{j=1}^{n}\alpha\int_0^1 y_{jk}^*(x)Pr(\hat{t}|t)\mathrm{d}x$$

is the sub-Stackelberg MCS utility of the leader (team k), and the quantity $v_{pj}(t_{jk}^*) = \alpha\int_0^1 y_{jk}^*(x)Pr(\hat{t}|t)\mathrm{d}x - c_{jk}(t)$ is the sub-Stackelberg MCS utility of the follower (participant j).

Through Theorem 3.3.3, we already have one lower bound for the variable w. However, we not only need to consider followers' (participants') rational responses, but also have to make sure requesters' willingness to cooperate rather than compete. The other lower bound of the variable w will be presented in Theorem 3.3.5.

Under this circumstance, both requesters and participants are encouraged to join this system.

3.3.5 Cost allocation

Payment policies above alone is not enough to encourage requesters to cooperate. Sometimes requesters will gamble on competition to gain more utilities even competition means a possibility of failure. Therefore, the MCS server must promise that requesters gain more by cooperation than competition, so how to allocate costs is very important.

In this section, we use Separable Costs Remaining Benefits method to allocate the nonseparable costs in proportion to each requester's willingness to pay minus the separable costs which was already allocated. In this way, requester i will not pay more than it's utility or the costs when it is working alone.

We use $c_M(t)$ to describe the total costs of all requesters, which is the total amount of payments requesters pay to participants.

$$c_M(t) = \alpha \sum_{j=1}^{n} \sum_{k=1}^{q} \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x,$$

where, α explains the decrease of each requester's costs due to cooperation.

Let $c_{ri}(t)$ represent requester i's costs,

$$c_{ri}(t) = c_M(t) - c_{M-\{i\}}(t) + \beta_i(t)(c_M(t) - \sum_{s=1}^{m}(c_M(t) - c_{M-\{s\}}(t))),$$

and

$$\beta_i(t) = \gamma_i(t)(\sum_{s=1}^{m}\gamma_s(t))^{-1},$$
$$\gamma_i(t) = c_{\{i\}}(t) + c_{M-\{i\}}(t) - c_M(t),$$

where $c_{\{i\}}(t)$ means requester i's costs when it is working alone and $c_{M-\{i\}}(t)$ describes costs when other requesters except i cooperate together.

$$c_{\{i\}}(t) = \sum_{k=1}^{q} x_{ik} \sum_{j=1}^{n} \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x,$$
$$c_{M-\{i\}}(t) = \alpha' \sum_{j=1}^{n} \sum_{k=1}^{q} \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x,$$

where α' is cost factor without requester i, $Pr(\hat{t}|t)$ is the possibility mass function that MCS server predicates actual sensing accuracy level t as $\hat{t}$ as described by eq. 3.1, and

$$c_{M-\{i\}}(t) = \frac{\alpha'}{\alpha} c_M(t).$$

Subsequently, we can deduce that requester i's utility when cooperating,

$$v_{ri}(t) = \sum_{j=1}^{n} \sum_{k=1}^{q} (\sigma G_{jk}(t) x_{ik}) - c_{ri}(t),$$

where $c_{ri}(t)$ represents requester i's costs, $G_{jk}(t)$ means the value of sensing report participant j uploaded to team k with accuracy level t, and x_{ik} describes whether requester i belongs to team k or not.

When competing, requester i will not have to share the collected data with other requesters, hence the value of sensing report participant j uploaded to the requester i with accuracy level t will be $g_{ij}(t)$. More than that, without other requesters sharing costs, requester i's costs will be $c_{\{i\}}(t)$.

Requester i's utility when competing under the exact same situation as cooperating is

$$v'_{ri}(t) = \sum_{j=1}^{n} g_{ij}^{k}(t) - \sum_{k=1}^{q} x_{ik} \sum_{j=1}^{n} \int_{0}^{1} y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x,$$

where $g_{ij}(t)$ is the value of sensing report participant j uploaded to the requester i with accuracy level t.

In this section, we assume that the purpose of all requesters is to maximize their utilities. As a result, we have to assure that requesters will gain more by cooperation than competition if we want to motivate requesters to cooperate instead of competing.

Theorem 3.3.5. *For the Stackelberg MCS game $Z_k = \langle \{k\}, N, W, T, u_k, u_p \rangle$, if inequality holds,*

$$w \geq \frac{\sigma \sum_{j=1}^{n} \sum_{k=1}^{q} Ktx_{ik} - Ktx_{ik}}{\sum_{j=1}^{n} \sum_{k=1}^{q} B - M},$$

where α is cost factor, α' is cost factor without requester i,

$$B = (\alpha - \alpha') \sum_{k=1}^{q} M + \frac{(\sum_{k=1}^{q} x_{ik}(\alpha' - \alpha)M)(\sum_{k=1}^{q}(\alpha - m(\alpha - \alpha'))M)}{\sum_{k=1}^{q}(\sum_{s=1}^{m} x_{sk} + m(\alpha' - \alpha)M)},$$

and

$$M = ((1-a)t\frac{Kt}{1+Kt} + \alpha(\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a}))),$$

then, requester i ($\forall i \in M$) will choose cooperation.

Proof. Requester i's cost can be formulated as

$$\begin{aligned} c_{ri}(t) &= c_M(t) - c_{M-\{i\}}(t) \\ &+ \frac{c_{\{i\}}(t) + c_{M-\{i\}}(t) - c_M(t)}{\sum_{s=1}^{m}(c_{\{s\}}(t) + c_{M-\{s\}}(t) - c_M(t))} * (c_M(t) - \sum_{s=1}^{m}(c_M(t) - c_{M-\{s\}}(t))) \end{aligned}$$

where

$$c_{M-\{i\}}(t) = \frac{\alpha'}{\alpha} c_M(t),$$

then

$$c_{ri}(t) = c_M(t) - \frac{\alpha'}{\alpha} c_M(t) + \frac{(c_{\{i\}}(t) + (\frac{\alpha'}{\alpha} - 1)c_M(t))((1 - m(1 - \frac{\alpha'}{\alpha}))c_M(t))}{\sum_{s=1}^{m} c_{\{s\}}(t) + m(\frac{\alpha'}{\alpha} - 1)c_M}(t)$$

$$= (\alpha - \alpha') \sum_{k=1}^{q} D + \frac{(\sum_{k=1}^{q} x_{ik} D + (\alpha' - \alpha) \sum_{k=1}^{q} D)(\alpha - m(\alpha - \alpha')) \sum_{k=1}^{q} D}{\sum_{s=1}^{m} \sum_{k=1}^{q} x_{sk} D + m(\alpha' - \alpha) \sum_{k=1}^{q} D},$$

where

$$\begin{aligned} D &= \int_0^1 y_{jk}(x) Pr(\hat{t}|t) \mathrm{d}x \\ &= c((1-a)t\frac{Kt}{1+Kt} + \alpha(\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a}))) = wM, \end{aligned}$$

$$M = ((1-a)t\frac{Kt}{1+Kt} + \alpha(\frac{K}{1+K}(\frac{1}{a} + \mathrm{e}^{-a} - \frac{1}{a}\mathrm{e}^{-a}))),$$

and a is the error rate from eq. 3.1.

As a result,

$$\begin{aligned} c_{ri}(t) &= (\alpha - \alpha') \sum_{k=1}^{q} wM + \frac{w(\sum_{k=1}^{q} x_{ik}(\alpha' - \alpha)M)(\sum_{k=1}^{q}(\alpha - m(\alpha - \alpha'))M)}{\sum_{k=1}^{q}(\sum_{s=1}^{m} x_{sk} + m(\alpha' - \alpha))M} \\ &= Bw, \end{aligned}$$

where

$$B = (\alpha - \alpha') \sum_{k=1}^{q} M + \frac{(\sum_{k=1}^{q} x_{ik}(\alpha' - \alpha)M)(\sum_{k=1}^{q}(\alpha - m(\alpha - \alpha'))M)}{\sum_{k=1}^{q}(\sum_{s=1}^{m} x_{sk} + m(\alpha' - \alpha)M)}.$$

Therefore,

$$
\begin{aligned}
&u_{ri}(t) - u'_{ri}(t) \\
&= \sum_{j=1}^{n}\sum_{k=1}^{q}(\sigma G_{jk}(t)x_{ik}) - (c_{ri}(t)) - (\sum_{j=1}^{n} g_{ij}(t) - \sum_{k=1}^{q} x_{ik}\sum_{j=1}^{n}\int_{0}^{1} y_{jk}(x)Pr(x|t)\mathrm{d}x) \\
&= \sigma\sum_{j=1}^{n}\sum_{k=1}^{q} Ktx_{ik} - \sum_{j=1}^{n}\sum_{k=1}^{q} Bw - Ktx_{ik} + Mw \\
&= \sigma\sum_{j=1}^{n}\sum_{k=1}^{q} Ktx_{ik} - Ktx_{ik} - (\sum_{j=1}^{n}\sum_{k=1}^{q} B - M)w \geq 0,
\end{aligned}
$$

afterwards

$$
w \geq \frac{\sigma\sum_{j=1}^{n}\sum_{k=1}^{q} Ktx_{ik} - Ktx_{ik}}{\sum_{j=1}^{n}\sum_{k=1}^{q} B - M}.
$$

□

Combining Theorem 3.3.3 with Theorem 3.3.5, we have

$$
w^* = \max(\frac{\sigma\sum_{j=1}^{n}\sum_{k=1}^{q} Ktx_{ik} - Ktx_{ik}}{\sum_{j=1}^{n}\sum_{k=1}^{q} B - M}, \frac{c_{jk}(1) - c_{jk}(0)}{(1-a)\alpha\frac{K}{1+K}}).
$$

The pair $(w^*, t^*_{jk} = 1)$ is a sub-Stackelberg MCS solution for the game $Z_k = \langle\{k\}, N, W, T, u_k, u_p\rangle$ with team k as the leader.

3.3.6 Model of the incomplete information dynamic MCS game

The goal of Q-learning is to learn a strategy that tells the agent (team k) the accurate data $c_{jk}(1)$ of participant j and thereby inform the agent (team k) of

what action (payment policy) to take under the circumstance. Q-learning is a model-free reinforcement learning technique to achieve an optimal payment policy without knowing the participant's perception model. Given an infinite exploration time and a partially random strategy, Q-learning can determine the optimal action selection strategy for any given finite Markov decision process. By Q-learning, the equilibrium of the leader (team $k \in Q$) can be achieved in real time.

Definition 3.3.6. *For the given incomplete information dynamic sub-Stackelberg MCS game at time r,*

$$Z_k^r = \langle \{k\}, N, W^r, T^r, U_k^r, U_p^r \rangle,$$

where $k \in Q$.

- *where $\{k\}$ is the team formed by requesters with the same data requirement, and N is the set of participants.*

- *$T^r = [0,1]^{1\times N}$ is the strategy space of participants for collecting data type k at time r, where $t_{jk}^r \in [0,1]$ denotes the sensing accuracy level of data type k from participant j at time r.*

- *$W^r = [0,+\infty)^{1\times N}$ is the strategy space of team k at time r. For each $w_{jk}^r \in W^r$, and $t_{jk}^r \in [0,1]$ the payment policy is defined as*

$$y_{jk}^r(t_{jk}^r) = \frac{G_{jk}(t_{jk}^r)}{1 + G_{jk}(t_{jk}^r)} w_{jk}^r \cdot t_{jk}^r$$

.

- *The state space of Q-learning is defined as*

$$s^r = (\hat{N}_k^{r-1}(t), w_{jk}^{r-1}),$$

where $\hat{N}_k^r(t) = \sum_{j=1}^n I(\hat{t}_{jk}^r = t), \quad t \in [0,1]$ *is the numbers of reports with sensing accuracy level* t*, and*

$$I(\hat{t}_{jk}^r = t) = \begin{cases} 1, & \hat{t}_{jk}^r = t, \\ 0, & \hat{t}_{jk}^r \neq t. \end{cases}$$

- $U_p^r = (U_{pj}^r)_{j \in N}$ *is the vector of utility functions for participants. For each* $j \in N$*,*

$$U_{pj}^r = \alpha \int_0^1 y_{jk}^r(x) I(\hat{x}_{jk}^r = x) \mathrm{d}x - c_{jk}^r(t'),$$

where $t' = t \cdot I(\hat{x}_{jk}^r = x)$*.*

- U_k^r *is the utility function of team* k*,*

$$U_k^r = \sum_{i=1}^m \sum_{j=1}^n \int_0^1 \sigma G_{jk}(x) I(\hat{x}_{jk}^r = x) x_{ik} \mathrm{d}x - \alpha \int_0^1 \hat{N}_k^r(x) y_{jk}^r(x) \mathrm{d}x.$$

The variable in this state is data $c_{jk}(1)$, while data $c_{jk}(1)$ is not up to the leader to decide. Hence, the action of the Q-learning should be payment policy y_{jk}^r at time r which is not only up to the leader, but also related to data $c_{jk}(1)$. For simplicity, at each precision level, the feasible costs of participant j for team k at each accuracy level are quantized into the C range (C is where all the costs $c_{jk}(1)$ are).

Payment policies achieved by Q-learning are based on system state s^r at time

r. System state s^r not only contains numbers of previous sensing reports but also payment policies, which is why $s^r = (\hat{N}_k^{r-1}(t), y_{jk}^{r-1})$ (y_{jk}^{r-1} is the payment policy at time $r-1$). Team $k \in Q$ will pay participant j based on current sensing estimated efforts $\hat{t}_{jk}^r$. In consequence, the total payment team $k \in Q$ pays participants at time r is $\alpha \int_0^1 \hat{N}_k^r(x) y_{jk}^r(x) \mathrm{d}x$. For any $k \in Q$, team k's utility at time r is

$$U_k^r = \sum_{i=1}^{m} \sum_{j=1}^{n} \int_0^1 \sigma G_{jk}(x) I(\hat{x}_{jk}^r = x) x_{ik} \mathrm{d}x - \alpha \int_0^1 \hat{N}_k^r(x) y_{jk}^r(x) \mathrm{d}x.$$

where $G_{jk}(t)$ is the value of sensing data participant j uploaded to team k with accuracy level t, α explains the decrease of each requester's costs due to cooperation, and σ the influence factor and it will increase with the scarcity of the data.

Correspondingly, participant j's utility at time r is

$$U_{pj}^r = \alpha \int_0^1 y_{jk}(x) I(\hat{x}_{jk}^r = x) \mathrm{d}x - c_{jk}^r(t'),$$

where $t' = t * I(\hat{x}_{jk}^r = x)$, and $c_{jk}^r(t')$ is the costs of participant j with accuracy level t' for collecting data to team k.

Definition 6. In the given incomplete information dynamic sub-Stackelberg MCS game

$$Z_k^r = \langle \{k\}, N, W^r, T^r, U_k^r, U_p^r \rangle$$

at time r, the set $T(w_{jk}^r) \subset [0,1]$ defined for each w_{jk}^r at time r by

$$T(w_{jk}^r) = \{t_{jk}^r \in [0,1] : U_{pj}^r(t_{jk}^r) \geq U_{pj}^r(t_{jk}^{'r}), \forall t_{jk}^{'r} \in [0,1]\}$$

is the optimal response (rational reaction) set of participant j to the strategy w_{jk}^r of leader (team k) at time r.

A strategy w_{jk}^{r*} is called a sub-Stackelberg MCS equilibrium at time r for the leader (team k), if

$$\min_{t_{jk}^{r*} \in T(w_{jk}^{r*})} U_k^r = \max_{w_{jk}^r} \min_{t_{jk}^r \in T(w_{jk}^r)} U_k^r.$$

The quantity $U_k^r(t_{jk}^*)$ is the sub-Stackelberg MCS utility of the leader (team k) at time r.

Any element $t_{jk}^{r*} \in T(w_{jk}^{r*})$ is an optimal strategy for the follower (participant j) at time r that is in equilibrium with w_{jk}^{r*}. The quantity $U_{pj}^r(t_{jk}^{r*})$ is the sub-Stackelberg MCS utility of the follower (participant j) at time r. The pair $(w_{jk}^{r*}, t_{jk}^{r*})$ is a sub-Stackelberg MCS solution at time r for the game $Z_k^r = \langle \{k\}, N, W^r, T^r, U_k^r, U_p^r \rangle$ with team k, ($k \in Q$) as the leader at time r.

3.3.7 The algorithm description

Based on the value of Q-function, teams use the greedy algorithm (GA) to choose payment policy. GA can make sure that teams always choose globally optimal solution. As discussed in Theorem 3.3.3, as long as $w \geq \frac{c_{jk}(1)-c_{jk}(0)}{(1-a)\alpha\frac{K}{1+K}}$, the only rational reaction set of participant j is $\{1\}$. Under that circumstance, the sub-Stackelberg MCS equilibrium at time r for the leader (team k) will be $y_{jk}^{r*} = \arg\max_w Q(s^r, y_{jk}^r)$.

Table 3.1: Simulation Payment Policy Based On Q-learning

1. **Initialize** τ, w, $\hat{N}_k^r(t) = 0, t \in [0,1]$, $Q(s,y) = 0$, and $V(s) = 0, \forall s$
2. **for** $r = 1, 2, 3, ...$ **do**
3.select $\forall t \in (0,1]$ 4. $s^r = (\hat{N}_k^{r-1}(t), y_{jk}^{r-1})$
5. Select $c_{jk}(1)$ within range C, then calculate y_{jk}^r with $w \geq \frac{c_{jk}(1)-c_{jk}(0)}{(1-a)\alpha\frac{K}{1+K}}$
6. Perform y_{jk}^r through greedy algorithm and broadcast the recruiting message
7. **for** $j = 1, 2, ..., n$ **do**
8. Use the sensing evaluation algorithm to get $\hat{t}_{jk}^r$
9. Pay $y_{jk}^r(\hat{t}_{jk}^r)$ to participant j
10. **end for**
11. Obtain U_k^r
12. Calculate $\hat{N}_k^r(t)$
13. Update $Q(s^r, y^r)$
14. Update $V(s^r)$
15. **end for**

As shown in Table 3.1, the algorithm used to get the sub-Stackelberg MCS equilibrium for the leader at time r is based on Q-function ($Q(s,y)$). $Q(s,y)$ is long-term discounted utility about action-move (s,y). Action-state will be updated by Bellman equation. Bellman equation is shown as below:

$$Q(s^r, y^r) \leftarrow (1-\tau)Q(s^r, y^r) + \partial(U_k^r + \gamma V(s^{r+1})),$$

and

$$V(s^{r+1}) \leftarrow \max_{y^{r+1}} Q(s^{r+1}, y^{r+1}),$$

where, s^{r+1} is the new state from state s^r by taking action y^r into account and y^{r+1} is the new payment. Value function $V(\bullet)$ means the highest value of Q-function. γ is discount factor, which means MCS system is short sighted by neglecting future utilities. $\tau \in (0, 1]$ means learning rate based on current experience $s^r - y^r - s^{r+1}$.

3.4 Simulation results on the incomplete information

In this section, we conduct simulations to validate the performance of our proposed incentive mechanism, and to prove that the theory equilibrium in this chapter can be achieved in real time with incomplete information. The proposed algorithms are implemented by Matlab2018.

3.4.1 Simulation results on the incomplete information about requesters

We consider a crowdsensing simulation setting with 1000 requesters and 10000 participants (i.e., $m = 1000$ and $n = 10000$). We focus on the crowdsensing interactions with 1000 episodes (i.e., $r = 1000$). At

each episode, there are at most 10000 teams randomly constituted (i.e., $q <= 10000$). Simulations have been performed to evaluate the system performance of the incomplete information dynamic sub-Stackelberg MCS game, with $\partial = 0.2, \gamma = 0.8$. For each requester i and participant j at episode t, we set the estimated error rate of MCS server a at 0.1. As to the value of sensing report participant j uploaded to team k with accuracy level t ($G_{jk}(t)$) follows $G_{jk}(t) \sim unifrnd(50, 100)$ and the costs of participant j with accuracy level t for collecting data to team k ($c_{jk}(t)$) follows $c_{jk}(t) \sim unifrnd(15, 20)$ (i.e. range C equals $[15, 20]$).

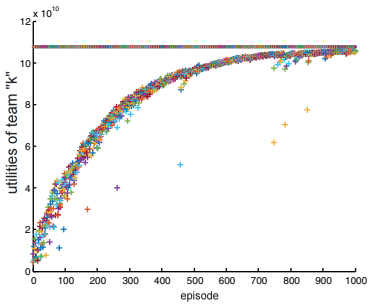
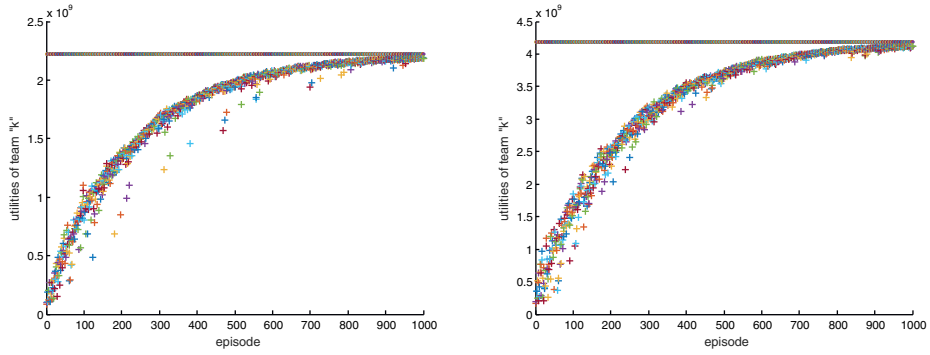


Figure 3.4: Performance of the incomplete information dynamic MCS game, average utility of the team vs. episode

In Fig. 3.4, the top line is the team's value corresponding to the sub-equilibrium of the leader (team $k \in Q$), which is a fixed value and will not change with the number of episodes. With regard to the Q-learning based value achieves an optimal value in this dynamic MCS game after convergence even with incomplete information, which matches the theoretical results of the sub-Stackelberg equilibrium given by Theorem 3.3.4. After about 950 iterations, the value of the team $k \in Q$ converges to stability which equals to the sub-equilibrium solution of the team $k \in Q$. Judging from Fig. 3.4, beyond doubt that in the case of incomplete

information with Q-learning, the sub-equilibrium of the leader (team $k \in Q$) can be achievable in practice. To rule out the influence of parameters, we change the parameter value range one by one in the following text.



(a) Average team utility vs. episode ($p = 0.2$)

(b) Average team utility vs. episode ($p = 0.5$)

Figure 3.5: Performance of the incomplete-information dynamic MCS game.

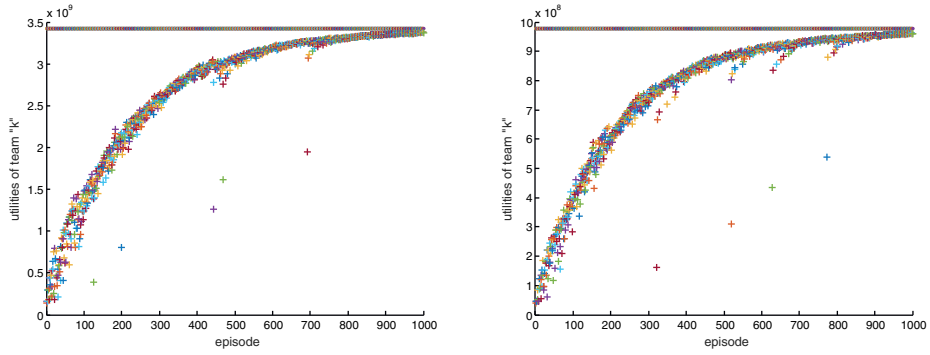
Firstly, we compare the effects of different estimated error rate of MCS server a on the utility of the team. We set the estimated error rate of MCS server a at 0.2, and 0.5. Other parameters remain the same as Fig. 3.4.

According to Fig. 3.5, no matter what the estimated error rate of MCS server a is, the average utility of the team $k \in Q$ always converges to stability after around 950 iterations with incomplete information which is in line with the outcome of Fig. 3.4. Apart from that, the value of the team $k \in Q$ still converges to stability which equals to the theoretical results of sub-equilibrium given by Theorem 3.3.4. In a word, the change of the estimated error rate of MCS server a does not affect getting our theoretical solution in real time.

The only difference between Fig. 3.4, and Fig. 3.5 is the sub-equilibrium utility of the team $k \in Q$ which decreases as the estimated error rate of MCS

server a increases. The difference can also be seen in Fig. 3.3. Due to the increase of the estimated error rate of MCS server a, participants' motivation has been damaged and requesters have to improve payments to make up the loss to participants caused by the increased the estimated error rate.

Whereafter, we change the costs of participant j with accuracy level t for collecting data to team k, (i.e. range C) to [10,15] and [15,20]. The rest parameters remain unchanged as in Fig. 3.4.



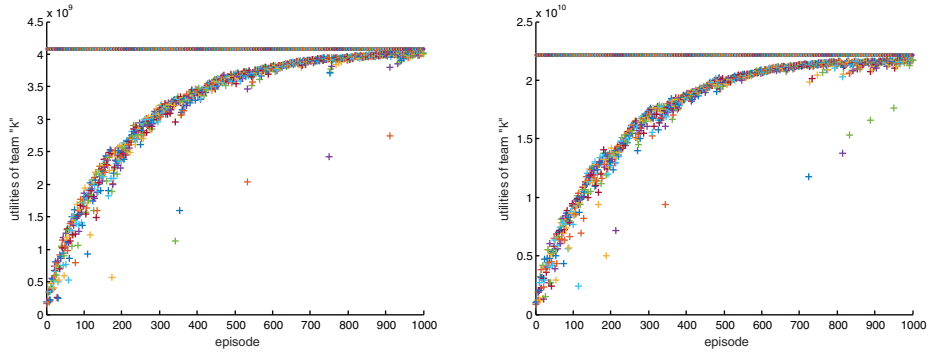
(a) Average team utility vs. episode ($n \in [10, 15]$

(b) Average team utility vs. episode ($n \in [15, 20]$

Figure 3.6: Performance of the incomplete-information dynamic MCS game.

Same as Fig. 3.4, in Fig. 3.6, the value of the team $k \in Q$ also converges to stability about 950 iterations which equals to the sub-equilibrium solution of the team $k \in Q$ despite the impact of incomplete information. This means that the change of range C has no influence on getting the theoretical sub-equilibrium in Theorem 3.3.4 in practice. Because payment policies will increase with the increase of participants costs $c_{jk}(t)$ to stimulate participants to upload more sensing data. That is why the sub-equilibrium solution of the team $k \in Q$ decreases in Fig. 3.6 and Fig. 3.4.

Finally, we change the final parameter that affects the team $k \in Q$, the value of sensing report participant j uploaded to team k with accuracy level t ($G_{jk}(t)$) to [25,50] and [100,150]. The remaining parameters are the same as Fig. 3.4.



(a) (b)

Figure 3.7: Performance of the incomplete-information dynamic MCS game.

On the basis of Fig. 3.7, the value of the team $k \in Q$, regardless of the change of the value of sensing report participant j uploaded to team k with accuracy level t ($G_{jk}(t)$), converges to stability about 950 iterations which equals to the sub-equilibrium solution of the team $k \in Q$ as ever regardless of the incomplete information.

So far, it is safe to say that the theoretical sub-equilibrium in Theorem 3.3.4 can be achieved in regard of parameters in practice and the change of parameters will not influence the number of iterations of the solution which means the theoretical sub-equilibrium in Theorem 3.3.4 is stable.

3.4.2 Simulation results on the incomplete information about participants

As to participants, there are only two parameters, the estimated error rate of MCS server a and the costs of participant j with accuracy level t for collecting data to team k ($c_{jk}(t)$). By setting the parameters as shown in Fig. 3.4, utility of participant j always converges to stability in spite of the incomplete information after around 900 to 1000 iterations. The top line of Fig. 3.8 is the sub-equilibrium utility of participants j which can be achieved by Q-learning after converging. Fig. 3.8 shows that with Q-learning, the sub-equilibrium value of the follower (participant j) can be achieved in real time under the circumstance of incomplete information. To rule out the influence of parameters, we change the parameter value range one by one in the following text.

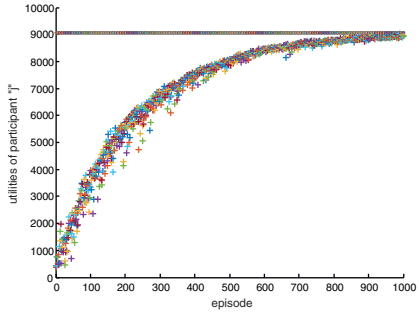
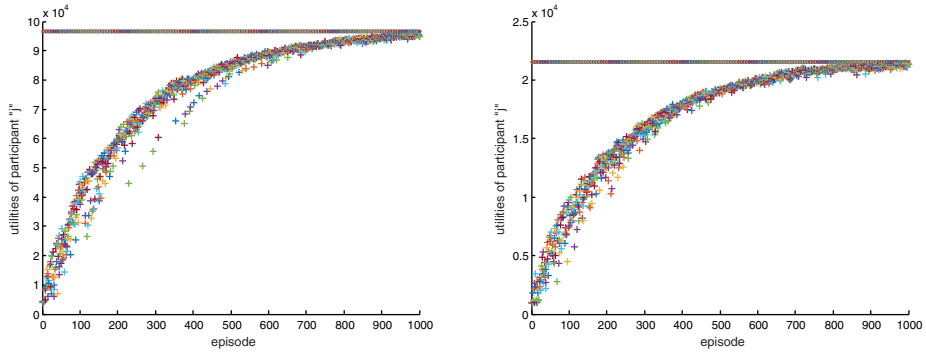


Figure 3.8: Performance of the incomplete information dynamic MCS game, utility of participant j vs. episode.

To begin with, we take the estimated error rate of MCS server a on the utility of the team into consideration. We set the estimated error rate of MCS server a at 0.2, and 0.5. Other parameters remain the same as Fig. 3.4



(a) (b)

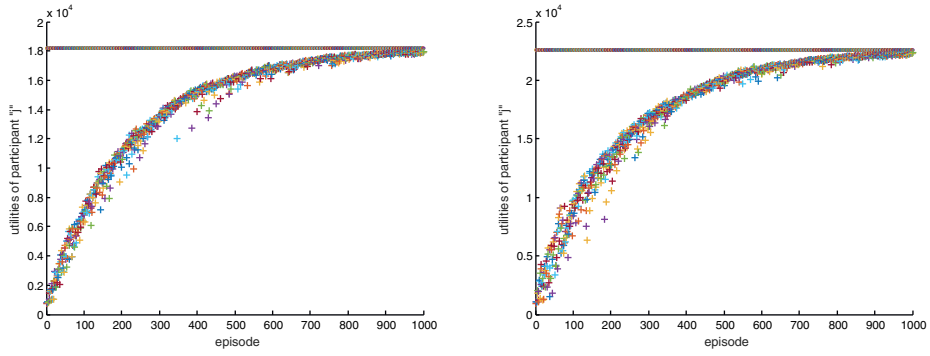
Figure 3.9: Performance of the incomplete information dynamic MCS game, utility of participant j vs. episode.

According to Fig. 3.9, in spite of the changes of the estimated error rate of MCS server a is, the utility of the participant j always converges to stability after around 900 to 1000 iterations along with incomplete information which is identify with the outcome of Fig. 3.8. Besides, the value of the participant j still converges to stability which matches the theoretical results of sub-equilibrium value given by Theorem 3.3.4. Generally speaking, the change of the estimated error rate of MCS server a does not affect getting our theoretical solution in practice.

The only difference between Fig. 3.9, and Fig. 3.8 is the sub-equilibrium utility of the participant j which decreases as the estimated error rate of MCS server a increases. Even though, the leader team k raised the payment, participant j's value still decreased because higher estimated error rate means a higher possibility that participants' accuracy level will be underestimated which leads to a lower payment.

Then, we change the costs of participant j with accuracy level t for collecting

data to team k, (i.e. range C) to [10,15] and [15,20]. The rest parameters remain unchanged as in Fig. 3.4.



(a) (b)

Figure 3.10: Performance of the incomplete information dynamic MCS game, utility of participant j vs. episode.

Same as Fig. 3.8, in Fig. 3.10, the value of the participant j also converges to stability around 900 to 1000 iterations in the presence of incomplete information which equals to the sub-equilibrium solution of the participant j. This indicates that the change of range C has no influence on achieving the theoretical sub-equilibrium in Theorem 3.3.4 in practice.

To sum up, like the sub-equilibrium of the leader, the theoretical sub-equilibrium in Theorem 3.3.4 for participants can be achieved despite different parameters in practice and the change of parameters will not influence the number of iterations of the solution which means the theoretical sub-equilibrium in Theorem 3.3.4 for participants is also stable in the case of incomplete information. Hence, the theoretical sub-equilibrium in Theorem 3.3.4 also exists in real time.

Considering all simulations, our incentive mechanism is capable of providing all requesters in demand with massive accurate data and every participants are willing to upload data as accurate as possible.

3.5 Conclusion

In summary, we proposed a mechanism based on cooperation where all requesters are in collaboration with each other for the sake of getting all kinds of accurate data with minimum costs. In this collaboration-driven mechanism, we solved two issues: the accuracy level of data and how to stimulate requesters to cooperate.

We have provided the sub-equilibrium of the static game, showing that all reasonable requesters and participants will join our MCS system according to the optimal strategy that we've given. In dynamic MCS game, a Q-learning based MCS system has been proposed to achieve the optimal value without knowing the sensing model of the participants. In incomplete information games, simulation results show that with Q-learning the theoretical sub-equilibrium value in Theorem 3.3.4 can be achieved even in the presence of sensing accuracy evaluation error and not knowing participants' private information. The combination of a Game theory with the MCS paradigm and the cooperation between requesters represent a promising approach for increasing the effectiveness of the data collection campaign. Note that the findings in this chapter can be applied to any MCS system with more than one requesters other than the ones about AI diagnose.

However, we assumed that all requesters are willing to join together to collect data which is inconsistent with reality, and this may be a little too absolute. Besides, in this chapter, we assume that the more accurate the data, the better.

Hence, selfish participants may upload a bunch of irrelevant data to cheat on requesters to gain more payments. In our future work, we plan to focus on the coopetition (competition and cooperation) between requesters and implement a more realistic MCS platform.

4

Time Variability

This chapter is based on: Zhilin Xu, Hao Sun, Panfei Sun, and Qianqian Kong. "Requester mobility for mobile crowdsensing system: A dynamic alliance-based incentive mechanism". In: *Ad Hoc Networks* 166 (2025), p. 103680.

In the Mobile Crowdsensing (MCS) system, due to the heterogeneity of requesters, they will chang over time which means requesters will join or leave the MCS system at different times. Time variability makes requesters uncertain. This uncertainty leads to instability in the match between requesters and participants. As a result, changes in competition intensity cause dissatisfaction among requesters and participants, which prevents the established matching algorithm from being completed. For time variability, we design a dynamic coalition-based incentive mechanism where requesters can leave, join the MCS system separately. To the instability, we divide the dynamic mechanism into different stages and will update the matching rules in each stage. A unique algorithm based on the dynamic Stackelberg game and the corresponding updated algorithm is proposed to analyze the matching strategies of requesters and participants to make an optimal match. By proving the stability of the updated rules, we guarantee the stability of the match with time variability. Through numerical analysis, coalition formation can significantly reduce weak requesters' costs by at most 90%. Besides, the mechanism guarantees that each requester receives data within the first two stages in which they participate.

4.1 Introduction

In recent years, the popularity of mobile devices, such as mobile phones, smart watches, and laptops, has provided feasible conditions for MCS systems. Recently, the scale of smartphone users has been rapidly growing, and the number of Chinese telephone users has reached 1.9 billion in 2023, an increase of 43 million in 2022. Not only that, now most mobile devices have a lot of residual computing resources, which would provide the basis for the hardware environment of MCS (Jin, Su, and Nahrstedt 2017). Today, MCS is already on a rapid and profound development path, which has been deeply integrated into paradigms like *TrackYourTinnitus* in which Pryss et al. (2015) implemented a mobile crowd sensing platform to reveal new medical aspects of tinnitus and its treatment, the air quality detection application *Common Sense* developed by the University of California, Berkeley (UCB) (Dutta et al. 2009), the project *Creek Watch* studied by IBM to research the pollution status of waters (Kim et al. 2011), etc.

It is important for the MCS system to consider time variability. First, requesters join the system at different times. Second, requesters collect data at different speeds. Some requesters collect too much data, while others collect none at all. Without considering time variability, existing incentive mechanisms cannot handle changes in requesters. With time variability, requesters have difficulty in making an optimal payment scheme due to their lack of information about the intensity of competition, and without suitable payment policies, requesters cannot be optimally matched with participants. This leads to inefficient assignment of participants to requesters.

We propose a discretized dynamic incentive mechanism. Building on the model in Chapter 3, we now assume that teams, which are called data types

in Chapter 4, can be issued in different quality levels. Each data type can different levels of quality, where a higher level indicates a better quality. For example, a better resolution, or better indication of position in GPS systems. Requesters can decide which data type to request based on their data requirements. Participants can choose which type of data they want to provide. Because data types are independent of each other, we only consider one data type as a typical example in the remainder of this chapter.

For each data type, a requester can ask for an amount of data of a certain minimum level of quality at a per-unit price. The per-unit price is also chosen by the requester in each time period. In the first time period, requesters request data on their own while in the remaining time periods, requesters can choose to request data together. Participants can choose to offer data of certain quality level to a certain requester for the price issued by the requester.

We model the relationship between requesters and participants at the first stage as a Stackelberg game, where requesters as leaders first announce their data requests and payment schemes, and then participants as followers choose their degree of participation, based on requesters' data requests and payment schemes. We then solve for the equilibrium of the Stackelberg model. Since the participants only have one optimal strategy, the requesters can maximize their utility. Based on requesters' equilibrium strategies, each participant will choose one requester to match with to maximize its own utility.

However, the matching results of the former stage may not be efficient in the next stage due to the changes in the set of requesters. Based on the matching results in the former stage, requesters who were not matched with any participant are called weak requesters in this stage. The requesters with a data

value smaller than the maximum data value of weak requesters now form a coalition and request data together.

Requesters within the coalition will keep their required total data amount, but update the minimum data quality requirements and the payment schemes. The minimum data quality requirement of coalition guarantees that all coalition members can get data satisfying their quality requirement. Further the coalition's equilibrium payment is the minimum payment that ensures the coalition can always match with participants.

We prove that the coalition formed in that way is stable even in the presence of time variability of requesters. Our simulations show that the coalition formed by the centralized platform reduces the costs of requesters within the coalition when compared with other methods in Song et al. (2020); Y. Li et al. (2019). Simulations show that with very high probability requesters receive data within the first two stages in which they participate. While the algorithms in Nie et al. (2020); Shinkuma et al. (2020); Yang et al. (2018); Song et al. (2020), have requesters who are not matched with any participants no matter how many time periods have passed, our algorithms ensure that each requesters can will be matched with at least one participant. Moreover, both algorithms proposed in this chapter are polynomial time solvable.

The main contributions of this chapter are summarized below:

- Taking time variability into account, we propose a dynamic coalition-based incentive mechanism where requesters are able to leave or join the MCS system individually.
- Compared with other existing mechanisms, our method is capable of significantly reducing requesters' costs by forming a coalition to compete against other requesters.

- Not only theoretically, but also by simulations, we argue that each requester will be matched with participants within the first two stages in which that requester participates.

The remainder part of this chapter is organized as follows. The Stackelberg MCS model along with the matching algorithm at stage 0 is described in Section 4.2. Section 4.3 describes the updating matching algorithm. Section 4.4 presents the simulation results of the two proposed algorithms. Finally, Section 4.5 summarizes the contributions of this chapter.

4.2 Stage 0: matching participants to requesters

As we have discussed, with time variability, we propose a dynamic coalition-based incentive mechanism where all requesters are free to leave, join, or change data demands at the beginning of each stage. To reduce the impact of the instability brought by time variability, we divide the entire dynamic incentive mechanism into T stages, and in each stage we will update the matching rules of requesters and participants. In this section, we first introduce the matching algorithm between requesters and participants in its initial state.

4.2.1 The establishment of Stackelberg MCS model

Before we devise an algorithm that yields the matching results of requesters and participants, we apply a multi-leaders Stackelberg game to analyze the participation level of participants and the optimal incentive strategies of requesters and then Algorithm 4.1 will be used to match requesters and participants.

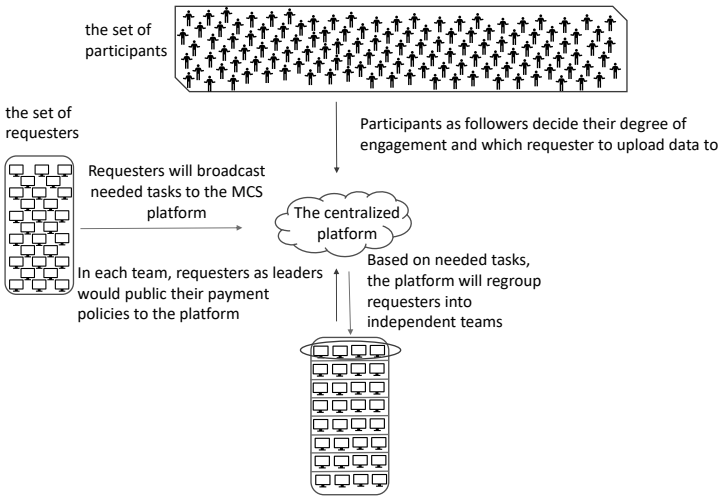


Figure 4.1: The illustration of the Stackelberg MCS system

Symbol description of Stackelberg MCS model

We model the relationship between requesters and participants in stage 0 as a Stackelberg MCS game. With the absence of historical information, we have no way of knowing which requesters will be weak requesters to form the coalition and which requesters will go it alone. Thus, in stage 0, we assume that all requesters will work on their own. Without loss of generality, we use $M^0 = \{1, 2, ..., m\}$ to denote the set of requesters at stage 0 and $N = \{1, 2, ..., n\}$ to represent the set of mobile device users who are willing to participate in the MCS system. At first, requesters as leaders announce tasks $S = \{S_1, S_2, ..., S_i, ..., S_m\}$ to the MCS platform, where $S_i = \{S_{i1}, S_{i2}, ..., S_{il}\}$ is the set of tasks broadcast by requester i. Due to the heterogeneity of the tasks issued by requesters, we use the concept of "data type" which uniquely defines jointly owned data of a class of tasks as in Section 3.2 in Chapter 3.

Requesters request dufferent data types based on their data requirements. All data types are independent of each other. Sequentially, let the set of data types be $Q = \{1, 2, ..., q\}$. Due to the independence of different data types, in the remainder of this chapter, we only consider the data type $k \in Q$ as a typical example. M^{k0} is the set of requesters who request data type $k \in Q$ at stage 0.

For the data type $k \in Q$, requesters public payment policies that are their incentive strategies p_i^{k0} and their data quality requirements α_i^k, which we assume will not change over time to the centralized platform. p_i^{k0} is the unit payment requesters pay to any participants who provide for data type $k \in Q$ at stage 0 and $\alpha_i^k (\alpha_i^k > 0)$ means requester i's requests for the quality of the data participants provide for data type $k \in Q$. It is worth noting that each task is preset with acceptable data quality requirements to avoid failure caused by poor-quality sensing data. Afterwards, participants as followers will decide which requester they upload data to with what kind of participation degree as shown in Fig. 4.1. Spontaneously, participants as followers have two related strategies $\bar{I}_{ij}^{k0}$ and x_{ij}^{k0}, where $\bar{I}_{ij}^{k0}$ is indicative function represents whether participant j will match requester i or not for data type $k \in Q$ at stage 0, and $x_{ij}^{k0} \geq 0$ is participant j's participation degree ($x_{ij}^{k0} = 0$ equals participant j not participate). Moreover, participant j can only upload data to one requester for one data type, and limited by participants' own energy and resources, participants can only collect data for one data type.

$$\bar{I}_{ij}^{k0} = 0, \text{participant } j \text{ does not upload data to requester } i,$$
$$\bar{I}_{ij}^{k0} = 1, \text{participant } j \text{ uploads data to requester } i.$$

Model definition

Both requesters and participants are rational, which means they will only decide on strategies with regard to their own best interests. When it comes to participants, obviously, payments should be the more the merrier. Yet for requesters, higher payments mean higher costs. To figure out what kind of strategies rational requesters and rational participants will choose, we must first calculate their incomes and costs.

Apparently, requesters' incomes will increase as participants' participation degree go up. Thus, by defining requester $i \in M^{k0}$'s income as $v_i^k x_{ij}^{k0}$. Requester i has per-unit data value $v_i^k > 0$ for data type k provided that the data quality is at least of quality α_i^k, and the amount of data at least of quality α_i^k does not exceed the total required data amount W_i^k. The utility to requester i of data type $k \in Q$ in stage 0 can be defined as

$$u_{ri}^{k0}(p_i^{k0}) = \sum_{j \in N} \bar{I}_{ij}^{k0}(v_i^k x_{ij}^{k0} - p_i^{k0} x_{ij}^{k0}), \forall i \in M^{k0}.$$

Participants' incomes are the payments they get and their costs are equal to their unit costs c_j^k which is irrelevant with stage multiplied by their participation degree. Participants' costs will increase with the participation degree and requesters' data demands which is added to participants' costs to guarantee the data collected by the participant must meet the data requirements of the requester. Furthermore, with the growth of participation, the cost of participants would also increase while the growth of payment may slow down. Therefore, the utility function of participants should be decreased after the increase. More than that, the rate of change in participants' utility also increases and then decreases. Because in practice, the utility function of partic-

ipants should first accelerate and then slow down as the participation degree increases. The initial acceleration of participants' utilities is due to to the MCS system wanting to incentivize participants to upload very precise data, thus the higher the participation of the participants, the more comprehensive and accurate the data collected, the faster the utility growth, and the more can stimulate the participants to increase the input and collect more accurate and comprehensive data. However, with the increase of participant's participation degree, there may be a phenomenon of over-sensing, that is, participants collect some seemingly relevant but actually irrelevant data, so the change of participants' utility function will slow down. Sequentially, the utilities of participants are defined as

$$u_{pj}^{k0}(x_{ij}^{k0}) = \sum_{i \in M^{k0}} \bar{I}_{ij}^{k0}(p_i^{k0} x_{ij}^{k0} - c_j^k (x_{ij}^{k0})^3 \alpha_i^k), \forall j \in N.$$

Requesters have the same unit payment scheme for different data quality as long as the data meets the minimum quality requirement. Therefore, the rational participants will only collect and upload data as the data quality requirements required. More than that, as long as the participant upload data satisfying data quality requirements, then the requester must purchase all amount not part of it.

None rational requesters and participants will join the MCS system at a loss, hence both $u_{ri}^{k0}(p_i^{k0})$ and $u_{pj}^{k0}(x_{ij}^{k0})$ should be positive.

Definition 4.2.1. *The Stackelberg Mobile Crowdsensing model at stage* 0 *is defined as*

$$Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q,$$

where

- *M^{k0} denotes the set of requesters requiring data of type k at stage 0, and N denotes the set of participants,*
- *$P^{k0} = [0, +\infty)^{M^{k0} \times 1}$ is the set of unit prices that requesters offer to participants who provide data type k. So for $p \in P^{k0}$, $i \in M^{k0}$, p_i^{k0} is the price that requester i pays for any participants who provide data type k with participation level 1.*
- *$X^{k0} = \{x_j^{k0} : [0, +\infty) \to [0, +\infty) | j \in N\}$ is the set of participation response strategies of participants. Hence for each $j \in N$, $i \in M^{k0}$, $p_i^{k0} \in [0, +\infty)$, $x_{ij}^{k0}(p_i^{k0})$ is the participation level of participant j in providing data type k for requester i given the unit price p_i^{k0} at stage 0.*
- *$U_r^{k0} = \{u_{ri}^{k0} \mid i \in M^{k0}\}$ denotes the set of requester utility functions, where*

$$u_{ri}^{k0} : P^{k0} \times X^{k0} \to \mathbb{R}^+,$$

- *$U_p^{k0} = \{u_{pj}^{k0} \mid j \in N\}$ denotes the set of participant utility functions, where*

$$u_{pj}^{k0} : P^{k0} \times X^{k0} \to \mathbb{R}^+.$$

Strategies of requesters and participants

After introducing the game, we are able to define both requesters' and participants' strategies in the Stackelberg MCS game.

Definition 4.2.2. *In the Stackelberg MCS game*

$$Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q,$$

the set $T(p_i^{k0}) \subset [0, +\infty)$ *defined for each* p_i^{k0} *by*

$$T(p_i^{k0}) = \{x_{ij}^{k0} \in [0, +\infty) : u_{pj}^{k0}(x_{ij}^{k0}) \geq u_{pj}^{k0}(x'^{k0}_{ij}), \forall x'^{k0}_{ij} \in [0, +\infty), i \in M^{k0}\}. \tag{4.1}$$

is the optimal response (rational reaction) set of participant j *to the strategy* p_i^{k0} *of requester* i*. Whatever requesters' payment policies are, participants would always choose the participation degree to maximize their utilities. Then, in view of the strategy* x_{ij}^{k0} *from eq. 4.1, participants will choose one requester to maximize* u_{pj}^{k0}*.*

A strategy p_i^{k0*} *is called a Stackelberg MCS equilibrium for the leader (requester* i*) , if*

$$\min_{x_{ij}^{k0*} \in T(p_i^{k0*})} u_{ri}^{k0} = \max_{p_i^{k0}} \min_{x_{ij}^{k0} \in T(p_i^{k0})} u_{ri}^{k0}. \tag{4.2}$$

As far as risk-avoid, rational requesters are concerned, despite of participants' participation levels, they will always decide on strategies that can maximize their minimum utilities.

Any element $x_{ij}^{k0*} \in T(p_i^{k0*})$ is an optimal strategy for the follower (participant j) that is in equilibrium with p_i^{k0*}. The pair $(p_i^{k0*}, x_{ij}^{k0*})$ is a Stackelberg MCS solution for the game $Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q$, with requester i as the leader.

According to eq. 4.1, for each requester's payment policy, rational participants will always choose an optimal participation degree to maximize their utilities. Afterwards, based on eq. 4.2 the risk-averse requester will choose an optimal payment policy so that no matter how the participant chooses the participation degree within the optimal response set, the minimum utility of the requester can be maximized.

Retrograde method is used to figure out the optimal payment policies of requesters and the optimal participation degree of participants with regard to Definition 4.2.2. First of all, we figure out participant j's optimal response (rational reaction) set as described in Theorem 4.2.3.

Theorem 4.2.3. *In the Stackelberg MCS game*

$$Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q,$$

the optimal response set of rational participant j *is* $T(p_i^{k0}) = \{\sqrt{p_i^{k0}/(3c_j^k\alpha_i^k)}\}$ *for any* $i \in M^{k0}$.

Proof. Despite of which requester, no rational participants will join the Stackelberg MCS system with a degree of participation that harms their own utilities.

To put eq. 4.1 simply, when the payment policy p_i^{k0} is known, we only need to calculate the x_{ij}^{k0} which can maximize participant j's utility. In this part, we do not think about which specific requester the participant will join, hence, participant j's utility can be translated as $u_{pj}^{k0}(x_{ij}^{k0}) = p_i^{k0}x_{ij}^{k0} - c_j^k(x_{ij}^{k0})^3\alpha_i^k$ and take the derivation, we have

$$\frac{\mathrm{d}u_{pj}^{k0}}{\mathrm{d}x_{ij}^{k0}} = -3c_j^k\alpha_i^k(x_{ij}^{k0})^2 + p_i^{k0}.$$

Judging from above, it is apparent that participant j's utility function will first increase and then decrease.

Therefore, we have

$$x_{ij}^{k0} = \sqrt{p_i^{k0}/3c_j^k\alpha_i^k}.$$

Since $T(p_i^{k0}) = \{\sqrt{p_i^{k0}/3c_j^k\alpha_i^k}\}$ is follower's (participant j's) only rational strategy in spite of which requester participant j match, it is safe to say that $x_{ij}^{k0} = \sqrt{p_i^{k0}/3c_j^k\alpha_i^k}$ is a Stackelberg MCS equilibrium for the follower. □

Through Theorem 4.2.3, it is obvious that no rational participants will abandon the participation degree $x_{ij}^{k0} = \sqrt{p_i^{k0}/3c_j^k\alpha_i^k}$ in favor of a strategy with lower utility. Because there is only one participation degree of participants in their optimal respond set, hence, as far as requester i is concerned, participants' participation degree is known information. Deducing from participants rational strategy, requesters can set down the payment policies to maximize their own utilities.

Theorem 4.2.4. *For the Stackelberg MCS game*

$$Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q,$$

$p_i^{k0*} = \frac{v_i^k}{3}$ *is the Stackelberg MCS equilibrium for the leader (requester i).*

Proof. Taking participants equilibrium strategy into requesters' utility function, we can get that

$$u_{ri}^{k0}(p_i^{k0}) = \sum_{j\in N} \bar{I}_{ij}^{k0}(v_i^k x_{ij}^{k0} - p_i^{k0} x_{ij}^{k0}) = \sum_{j\in N} \bar{I}_{ij}^{k0}(v_i^k \sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}} - p_i^{k0}\sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}).$$

After that, we seek derivatives for p_i^{k0} and get

$$\frac{\mathrm{d}u_{ri}^{k0}}{\mathrm{d}p_i^{k0}} = \sum_{j\in N} \bar{I}_{ij}^{k0}\left(\frac{v_i^k - p_i^{k0}}{6c_j^k\alpha_i^k\sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}} - \sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}\right) = \sum_{j\in N} \bar{I}_{ij}^{k0}\left(\frac{v_i^k - p_i^{k0} - 6c_j^k\alpha_i^k\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}{6c_j^k\alpha_i^k\sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}}\right).$$

Hence, requesters' utility function would first increase and then decrease, that means

$$\frac{\mathrm{d}u_{ri}^{k0}}{\mathrm{d}p_i^{k0}} = \sum_{j\in N} \bar{I}_{ij}^{k0}\left(\frac{v_i^k - 3p_i^{k0}}{6c_j^k\alpha_i^k\sqrt{\frac{p_i^k}{3c_j^k\alpha_i^k}}}\right) = 0 \iff p_i^{k0} = \frac{v_i^k}{3}.$$

In summary, the $p_i^{k0} = \frac{v_i^k}{3}$ is requester i's best strategy. □

On account of Theorem 4.2.3 and Theorem 4.2.4, there is no way that requesters could pay less without discouraging participants and participants could ask for more payments without hurting requesters' utilities. Not only participants are encouraged, all rational requesters will choose the payment strategy in Theorem 4.2.4.

After getting both requesters' and participants' strategies, it is easy for rational participants to choose a requester to maximize its own utility.

$$i = \arg\max_{i\in M^{k0}} u_{pj}^{k0}(x_{ij}^{k0}) = \arg\max_{i\in M^{k0}} \frac{2}{81}p_i^{k0}\sqrt{\frac{p_i^{k0}}{3c_j^k\alpha_i^k}}.$$

4.2.2 Pseudo-code of matching algorithm at stage 0

After obtaining both requesters' and participants' Stackelberg equilibriums, we use an easily implementable algorithm for the match between requesters and participants that is to find an optimal i for each rational participant $j \in N$. The pseudo-code of the matching algorithm in stage 0 is summarized in Algorithm 4.1.

Table 4.1: Matching Algorithm Between Requesters and Participants at Stage 0

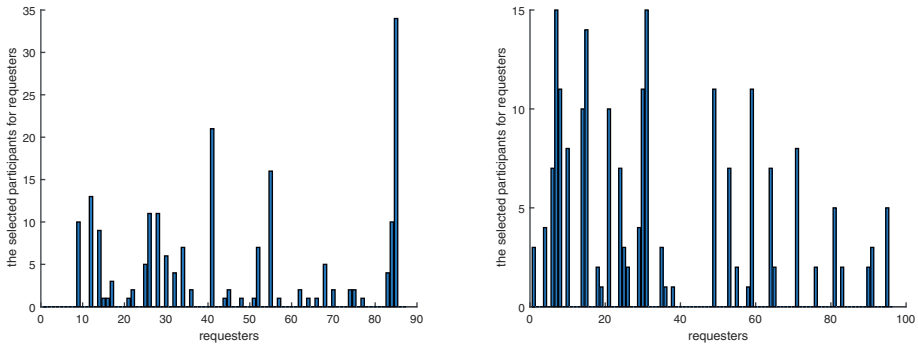
Input: requester set $M = \{1, 2, \ldots, m\}$, participant set $N = \{1, 2, \ldots, n\}$, and task set $S = \{S_1, S_2, \ldots, S_m\}$.

Output: payment policy p_i^{k0}, participation degree x_{ij}^{k0}, and matching results.

Algorithm:

1. Requesters request data types $Q = \{1, 2, \ldots, q\}$ based on their needed data.
2. Update requester set M^{k0} for all $k \in Q$.
3. For each $i \in M^{k0}$, calculate the optimal payment policy p_i^{k0} according to Theorem 4.2.4.
4. Update participation degree $x_{ij}^{k0} = \sqrt{\frac{p_i^{k0}}{3c_j^k \alpha_i^k}}$.
5. Initialize $I = 0$ and $U = 0$.
6. For each $j \in N$:
   6.1 For each $i \in M^{k0}$, if $u_{pj}^{k0}(x_{ij}^{k0}) > U$, set $I = i$ and $U = u_{pj}^{k0}(x_{ij}^{k0})$.

Fig. 4.2 are examples of the match between requesters and participants after one and 500 episodes on account of Algorithm 4.1. There are 100 initial requesters, and 200 participants and at the beginning of every game. The value



(a) 1 episode (b) 500 episodes

Figure 4.2: Match results of requesters

of data represented by data type $k \in Q$ to requester v_i^k is randomly generated from 300 to 400, data quality requirement α_i^k is a random number between 10 and 25 and the unit costs of each participant is between 50 and 100. As shown in Fig. 4.2, each vertical line separately represents the number of participants each requester is matched to. Apparently, if we do not update Algorithm 4.1, then no matter how many games have been played, there will still be requesters that do not pair successfully. To make more requesters selected, we will formate a coalition among weak requesters as in Section 4.3.

4.3 Stage t: updated match between requesters and participants

Due to time variability, the matching results of Algorithm 4.1 are unstable. To deal with time variability and the instability it brings, we will update the set of requesters and the matching rules in stage t. The total sensing period

is split into several equal-interval time slices in terms of the uncertainty of requesters brought by time variability. For all t in this chapter we have $t \in \{1, 2, ..., T\}$.

4.3.1 Formation of the coalition

The matching algorithm in Section 4.2 has the drawback that there is no guarantee that all requesters can get their needed data as shown in Fig. 4.2. To avoid some requesters not being able to match participants, some requesters will update their strategies into forming a coalition through the centralized platform to request data together.

Weak requesters and strong requesters

With regard to requesters who are already capable of getting enough needed data on their own, they have no interest in forming coalitions, only those who are incapable of getting valid data by themselves need to update their strategies into forming a coalition. Thus, in this section, we introduce the concept of "weak requesters" to form allies.

To better present "weak requesters", let us introduce a new parameter "degree of completion" $w_i^{k(t-1)}$ which equals to $\frac{\sum_{j \in N} \bar{I}_{ij}^{k(t-1)} x_{ij}^{k(t-1)}}{W_i^k}$. $\bar{I}_{ij}^{k(t-1)}$ is an indicative function that indicates whether participant j will gather data for requester i with data type k in the last stage $t-1$ or not, and W_i^k is the total amount of requirements requester i has for data type $k \in Q$. W_i^k is preset by the requester itself and related to the number of reports, the quality of sensing reports, the time range of reports and so on. When the amount of data actually collected by the requester is greater than or equal to the amount of

data required by the requester given in advance, the requester has completed the data collection and will leave the system.

As for "strong requesters", they must obey the two following standards,

· in the previous stage (stage $t-1$), the data completion degree of requester i cannot be 0,

· the data value of strong requesters must be greater than the data value of weak requesters, this is to endure the stability of the formed coalition.

Weak requesters are defined as the set of requesters whose data complete degree from stage $t-1$ is smaller than ϵ, $w_i^{k(t-1)} < \epsilon$. We note the set of weak requesters for data type k at stage t as M_a^{kt}. Write v^{kt} as the criteria for judgment for strong requesters of data type k at stage t, which is the maximum data value of weak requesters. $v^{kt} = \max_{i \in M_a^{kt}} v_i^k$.

For the remaining requesters $i \in\in M^{kt} \setminus M_a^{kt}$, if its data value is smaller than the critera for judgment $v_i^{kt} < v^{kt}$, then this requester will join the coalition formed by weak requesters.

The strong requeters are defined as

Definition 4.3.1. *Requester i is defined as a strong requester if and only if for any $\varepsilon > 0$ given by the system, requester i's degree of data completion in stage $t-1$ is greater than ε and conform eq. 4.3*

$$v^{kt} = \max_{i \in M_a^{kt}} v_i^k \tag{4.3}$$

where M_a^{kt} is the set of weak requesters at stage t.

Immediately following the definition of "strong requesters", through the centralized platform a coalition is formed by the rest requesters apart from strong requesters and new coming requesters to reduce costs and increase the number of requesters chosen by participants.

4.3.2 The update of requesters at stage t

Apart from strong requesters and new coming requesters, the rest requesters spontaneously form a coalition through the centralized MCS platform to compete against strong requesters because weak requesters are those who fail to meet expectations. At the start of a new phase, requesters whose data collection is complete may leave the system and new requesters may join the system.

At the beginning of stage t, for any $i \in M^{kt}$, the total degree of completion $\sum_{t'=0}^{t-1} w_i^{kt'}$ and 1 must be compared, where $\sum_{t'=0}^{t-1} w_i^{kt'}$ is requester i's total degree of completion in previous stage. If $\sum_{t'=0}^{t-1} w_i^{kt'} \geq 1$, then requester i's data collection is completed, otherwise requester i will be categorized into weak requester or strong requester to continue. Besides, new requesters will join the system. However, without data completion degree for new requesters in the last stage $t-1$, new requesters will not be part of the coalition. Consequently, both strong requesters and new requesters will participate in the dynamic Stackelberg MCS game at stage t according to the strategy in stage 0.

Without causing any ambiguity, let $M^{kt} = M_c^{kt} \bigcup M_a^{kt} \bigcup M_g^{kt}$ denote the set of requesters of data type $k \in Q$ at stage t, where M_c^{kt} is the set of strong requesters, M_a^{kt} the set of weak requesters, and M_g^{kt} the set of new requesters.

$\alpha_a^{kt} = \max_{i \in M_a^{kt}} \alpha_i^k$ is the coalition's requirements for data quality. While for requesters, whose data quality requirements are lower than the coalition's requirement, their data value will increase as the data quality improves. Hence, $v_i^{kt} = \frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k, i \in M_a^{kt}$. Different requesters have different requirements for data quality. If the coalition cannot meet the data requirements of the coalition members, the coalition members may not be willing to join the coalition. Therefore, for the coalition, the data demand of the coalition is equal to the maximum data demand of the coalition members. For the rest of the consortium, as the quality of the data improves, so does the value of the data. The parameters of strong requesters remain the same as in stage 0.

Therefore, the utilities of requesters and participants are updated as

$$\bar{u}_{ri}^{kt}(p_i^{kt}) = \sum_{j \in N} \bar{I}_{ij}^{kt}(v_i^k x_{ij}^{kt} - p_i^{kt} x_{ij}^{kt}), \quad \forall i \in M_c^{kt} \bigcup M_g^{kt},$$

$$\bar{u}_{ri}^{kt}(p_i^{kt}) = \sum_{j \in N} \bar{I}_{ij}^{kt}(\frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k x_{ij}^{kt} - p_i^{kt} x_{ij}^{kt}), \quad \forall i \in M_a^{kt},$$

where $\frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k$ is the updated value of data for requesters in the coalition a,

$x_{ij}^{kt} = \sqrt{\frac{p_i^{kt}}{3c_j^k \alpha_i^k}}, \forall i \in M_c^{kt} \bigcup M_g^{kt},$

$x_{aj}^{kt} = \sqrt{\frac{p_a^{kt}}{3c_j^k \alpha_a^k}}, a$ represents the coalition formed by weak requesters,

and

$$p_i^{kt} = \frac{v_i^k}{3}, \quad \forall i \in M_c^{kt} \bigcup M_g^{kt},$$
$$p_i^{kt} = p_{ai}^{kt} = \frac{v_i^{kt}}{\sum\limits_{i \in M_a^{kt}} v_i^{kt}} p_a^{kt}, \quad \forall i \in M_a^{kt},$$

p_{ai}^{kt} is the payment required by each requester in the coalition. In this chapter, we use a method of proportional distribution of data value to allocate costs between requesters.

$$\bar{u}_{pj}^{kt}(x_{ij}^{kt}) = \sum_{i \in M_c^{kt} \bigcup M_g^{kt}} \bar{I}_{ij}^{kt}(p_i^{kt} x_{ij}^{kt} - c_j^k (x_{ij}^{kt})^3 \alpha_i^k),$$
$$\forall i \in M_c^{kt} \bigcup M_g^{kt},$$
$$\bar{u}_{pj}^{kt}(x_{ij}^{kt}) = \sum_{i \in M_a^{kt}} \bar{I}_{aj}^{kt}(p_a^{kt} x_{aj}^{kt} - c_j^k (x_{aj}^{kt})^3 \alpha_a^{kt}), \quad \forall i \in M_a^{kt}.$$

4.3.3 The updated Stackelberg MCS game

The same as in Section 4.2, we use an updated Stackelberg game to analyze the updated optimal incentive strategies of requesters and then participants will update their selected requesters according to the updated algorithm to maximize their utilities based on the updated optimal incentive strategies of requesters.

Definition 4.3.2. *For a given Stackelberg mobile crowdsensing game*

$$Z_k^0 = \left\langle M^{k0}, N, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q,$$

its dynamic evolution at stage t is defined as

$$Z_k^t = \left\langle M^{kt}, N, P^{kt}, X^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q.$$

where

- M^{kk} *denotes the set of requesters requiring data of type* k *at stage* k*, and* N *denotes the set of participants,*
- $P^{kt} = [0, +\infty)^{M^{kt} \times 1}$ *denotes the set of unit prices that requesters offer to participants who provide data type* k*. To be precise* $p \in P^{kt}$*,* $i \in M^{kt}$*,* p_i^{kt} *is the price that requester* i *pays for participants who provide data type* k *with participation level* 1*.*
- $X^{kt} = \{x_j^{kt} : [0, +\infty) \to [0, +\infty), j \in N\}$ *denotes the set of participation response strategies of participants. So for each* $j \in N$*,* $i \in M^{kt}$*,* $p_i^{kt} \in [0, +\infty)$*,* $x_{ij}^{kt}(p_i^{kt})$ *is the participation level of participant* j *in providing data type* k *for requester* i *given the unit price* p_i^{kt} *at stage* t*.*
- $\bar{U}_r^{kt} = \{\bar{u}_{ri}^{kt} \mid i \in M^{kt}\}$ *denotes the set of requester utility functions, where*

$$\bar{u}_{ri}^{kt} : P^{kt} \times X^{kt} \to \mathbb{R}^+,$$

- $\bar{U}_p^{kt} = \{\bar{u}_{pj}^{kt} \mid j \in N\}$ *denotes the set of user utility functions, where*

$$\bar{u}_{pj}^{kt} : P^{kt} \times X^{kt} \to \mathbb{R}^+.$$

Furthermore, the decision state of requester i *at stage* t *is defined as*

$$s_i^{kt} = (p_i^{kt}, \chi_i^{kt}),$$

where χ_i^{kt} represents the coalition status of requester i: $\chi_i^{kt} \in \{0, 1\}$, where $\chi_i^{kt} = 0$ indicates that requester i participates in the game independently, whereas $\chi_i^{kt} = 1$ indicates that rquester i joins a coalition and participates in subsequent decisions.

Accordingly, the strategy space at stage t is extended from the independent competition strategy space in the initial stage to

$$P^{kt} \times \mathcal{C}^{kt},$$

where $\mathcal{C}^{kt}$ denotes the coalition selection space. Therefore, the dynamic evolution process can be represented as

$$Z_k^0 \to Z_k^1 \to \cdots \to Z_k^t,$$

forming an evolutionary Stackelberg mobile crowdsensing game that accounts for changes in requester states and coalition selection.

In game Z_k^t, M^{kt} is the new requesters' set after primary requesters leaving and new requesters joining. The strategies of both participants and strong requesters remain the same as in stage 0. To the coalition, we need to update its payment policy to ensure that the coalition has a certain degree of competitiveness which means the coalition would be able to match some but not all participants. Because there exists more than one data type in the system, weak requesters may form a coalition of one data type while becoming strong requesters and competing against each other of other data types. If for this data type, the coalition formed by weak requesters through the centralized platform forms monopoly which may be considered a vicious competition by strong requesters, so for other data types requesters that failed to request data of the original data type could retaliate against the coalition's monopoly

(Yu and Cannella Jr 2013).

Strategy updates of the formed coalition

Proposition 4.3.3. *The payment policy of the coalition should conform the following inequality to be able to compete against strong requesters without forming a monopoly.*

$$p_a^{kt} > \frac{(3\alpha^{kt}-1)^{\frac{2}{3}}\alpha_a^{kt}}{(3\alpha_a^{kt}-1)^{\frac{2}{3}}\alpha^{kt}}\frac{v^{kt}}{3}, \tag{4.4}$$

where v^{kt} is the criteria for judgment and α^{kt} is the data requirements of the strong requester corresponding to the judgment criterion v^{kt} in eq. 4.3.

Proof. To make the coalition competitive against strong requesters, we need to make sure that participants prefer collecting data for coalition to gathering data for strong requesters. Other than that, in order to make the coalition not form a monopoly, not all participants would rather obtain data for the coalition than strong requesters. Thereby, we should assure that

$$\bar{u}_{pj}^{kt}(x_{aj}^{kt}) > \bar{u}_{pj}^{kt}(x_{ij}^{kt}), (\exists i \in M^{kt} - M_a^{kt}),$$

where a represents the coalition.

Combined with the utility function

$$\begin{aligned}&\bar{u}_{pj}^{kt}(x_{aj}^{kt}) - \bar{u}_{pj}^{k1}(x_{ij}^{kt})\\ &= p_a^{kt}\cdot x_{aj}^{kt} - c_j^k(x_{aj}^{kt})^3\alpha_a^{kt} - p_i^{kt}\cdot x_{ij}^{kt} + c_j^k(x_{ij}^{kt})^3\alpha_i^{kt}\end{aligned}$$

$$= p_a^{kt}\sqrt{\frac{p_a^{kt}}{3c_j^k \cdot \alpha_a^{kt}}} - c_j^k \frac{p_a^{kt}}{3c_j^k \cdot \alpha_a^{kt}}\sqrt{\frac{p_a^{kt}}{3c_j^k \cdot \alpha_a^{kt}}} - p_i^{kt}\sqrt{\frac{p_i^k}{3c_j^k \cdot \alpha_i^k}} + c_j^k \frac{p_i^k}{3c_j^k \cdot \alpha_i^k}\sqrt{\frac{p_i^k}{3c_j^k \cdot \alpha_i^k}}$$

$$\iff (p_a^{kt} - \frac{p_a^{kt}}{3\alpha_a^{kt}})\sqrt{\frac{p_a^{kt}}{\alpha_a^{kt}}} > (p_i^{kt} - \frac{p_i^{kt}}{3\alpha_i^k})\sqrt{\frac{p_i^{kt}}{\alpha_i^k}}$$

$$\iff (3\alpha_a^{kt} - 1)(\alpha_i^k \cdot p_a^{kt})^{\frac{3}{2}} > (3\alpha_i^k - 1)(\alpha_a^{kt} \cdot p_i^{kt})^{\frac{3}{2}}$$

$$\iff p_a^{kt} > \frac{(3\alpha_i^k - 1)^{\frac{2}{3}}\alpha_a^{kt}}{(3\alpha_a^{kt} - 1)^{\frac{2}{3}}\alpha_i^k} p_i^{kt}, \text{ to avoid monopoly} p_i^{kt} = v_i^k/3 = v^{kt}/3.$$

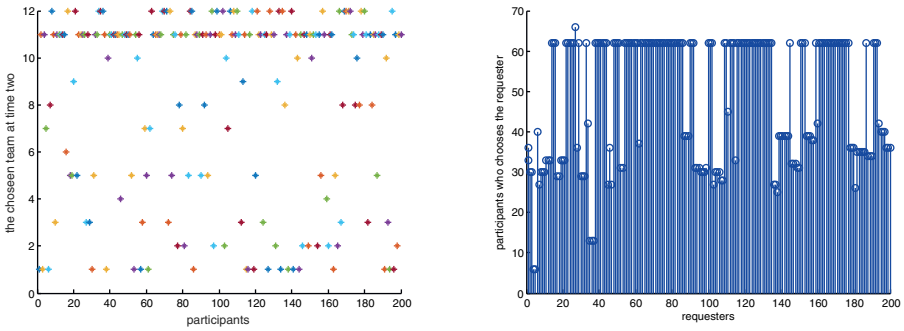
Therefore, when

$$p_a^{kt} = \frac{(3\alpha_i^k - 1)^{\frac{2}{3}}\alpha_a^{kt}}{(3\alpha_a^{kt} - 1)^{\frac{2}{3}}\alpha_i^k} p_i^{kt} + \delta, \quad \delta \to 0,$$

it is possible to ensure that the coalition maintains a certain level of competitiveness while avoiding monopoly. □

Through the process of proof, no doubt that participants who matched the requester corresponding to the "criteria for judgment (v^{kt})" will match the coalition cause from the proof of Proposition 4.3.3, and their utilities will go up by matching the coalition. Because strong requesters could already match participants in the last stage, and weak requesters who failed to match participants in the former stage will form a coalition and can certainly match participants in this stage. In short, all requesters can match some participants after at most two games.

From Fig. 4.3, it is obvious that the coalition can always get some but not all data which is in line with Proposition 4.3.3. In Fig. 4.3(a) where the ordinate 11 is the coalition, the rest ordinate represents strong requesters or new requesters, if there is a point on the horizontal line corresponding to the



(a) 1 episode (b) 200 episodes

Figure 4.3: Updated match results of requesters

ordinate, the requester corresponding to the ordinate is selected by the participant. It is obvious that the coalition has been chosen by some but not all participants. After 200 episodes, judging from Fig. 4.3(b) that each time the coalition has been picked by some but not all 200 participants.

4.3.4 The stability of the match

Since in this chapter requesters are in mobile, the matching results of Algorithm 4.1 is unstable. To handle the instability, we propose the above update strategies. The stability of the match means that both requesters and participants are happy with the matching results of the matching algorithm. To guarantee the happiness of requesters and participants, we only need to promise that both requesters and participants are satisfied with the matching strategies which means their utilities will be damaged if they change strategies. From Theorem 4.2.3 and Theorem 4.2.4, both requesters and participants are satisfied with the original strategies, so we just have to prove that

the updated matching strategies are stable. We will prove the stability of the updated matching strategy by proving the stability of the coalition.

Definition 4.3.4. *A coalition is stable if and only if the following conditions are met.*

- *Coalition membership stability*

  · Internal stability: when strong requesters do not change their strategies, weak requesters will not increase their utilities from leaving the coalition.

  · External stability: when weak requesters do not change their strategies, strong requesters will not increase their utilities from joining the coalition.

- *Coalition form stability when weak requesters do not change their strategies, strong requesters will not increase their utilities by forming any kind of new coalitions.*

To ensure the stability of coalition, how to allocate costs among weak requesters in the coalition is very important. In this chapter, we use a method of proportional distribution of data value to allocate costs between requesters. The payment policy p_a^{kt} in Proposition 4.3.3 is the total costs of the coalition. Thus, let p_{ai}^{kt} represent requester i's costs ($i \in M_a^{kt}$),

$$p_{ai}^{kt} = \frac{v_i^{kt}}{\sum\limits_{i \in M_a^{kt}} v_i^{kt}} p_a^{kt},$$

where the v_i^{kt} means value that data type $k \in Q$ brings to requester $i \in M_a^{kt}$

Now we have to prove that, with this costs allocation method, coalition stability is guaranteed.

Proposition 4.3.5. *The coalition a has coalition membership stability.*

Proof. According to Definition 4.3.4, coalition membership stability refers to weak requesters do not want to leave the coalition which means there's more to be gained from joining a coalition than going it alone, and strong requesters prefer not joining the coalition which indicates as to strong requesters joining the coalition will reduce utilities.

$$\bar{u}_{ri}^{kt}(i \in M_a^{kt}) - \bar{u}_{ri}^{kt}(i \in M^{kt}/M_a^{kt}) = \frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k x_{aj}^{kt} - p_{ai}^{kt} x_{aj}^{kt} - v_i^k x_{ij}^{kt} + p_i^{kt} x_{ij}^{kt}$$

$$= \frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k \sqrt{\frac{p_a^{kt}}{3c_j^k \alpha_a^{kt}}} - \frac{\frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k}{\sum_{i \in M_a^{kt}} \frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k} p_a^{kt} \sqrt{\frac{p_a^{kt}}{3c_j^k \alpha_a^{kt}}} - \frac{v_i^k}{3} \sqrt{\frac{v_i^k}{c_j^k \alpha_i^k}} + \frac{v_i^k}{9} \sqrt{\frac{v_i^k}{c_j^k \alpha_i^k}} > 0$$

$$\iff v_i^k \sqrt{3p_a^{kt} \alpha_a^{kt}} - \frac{v_i^k}{\sum_{i \in M_a^{kt}} \frac{\alpha_a^{kt}}{\alpha_i^k} v_i^k} p_a^{kt} \sqrt{3p_a^{kt} \alpha_a^{k1}} \frac{2}{3} v_i^k \sqrt{v_i^k} > 0$$

$$\iff v_i^k \sqrt{3p_a^{kt} \alpha_i^k} - \frac{v_i^k}{\sum_{i \in M_a^{kt}} \frac{\alpha_i^k}{\alpha_i^k} v_i^k} p_a^{kt} \sqrt{3p_a^{kt} \alpha_i^k} - \frac{2}{3} v_i^k \sqrt{v_i^k \alpha_i^k} > 0$$

$$\iff \sqrt{3p_a^{kt}} > \sqrt{v_i^k}.$$

Taking the fact that $p_a^{kt} \leq \frac{\sum_{i \in M_a^{kt}} v_i^k}{3}$, we can get $v_i^k < 3p_a^{kt}$.

Combining with Proposition 4.3.3, we have as long as $v_i^k < v^{kt}$, requesters prefer to joining the coalition. Conversely, when $v_i^k \geq v^{kt}$, none rational strong requesters would rather join the coalition than go solo. In general, coalition membership stability is guaranteed. □

After demonstrating the membership stability, we move on to the coalition form stability. This chapter discusses a one-to-many form of coalition, which

means we need to make sure that requesters outside the coalition (strong requesters) will not form any kind of new coalition.

Proposition 4.3.6. *The coalition a has coalition form stability, that is to say no rational strong requesters will form any kind of coalitions.*

Proof. We prove by contradiction. Let us assume that $M\bar{M}$ ($M\bar{M} \leq M_c^{kt}$) strong requesters will form a coalition, just prove that the utilities of strong requesters will be reduced then all rational strong requesters will choose not to form any coalitions. If coalition is formed, the new coalition only needs to ensure that compared with the original coalition, it can bring higher benefits to participants. Refer to the proof of Proposition 4.3.3, the payment policy of the new coalition should be

$$p_{na}^{kt} > \frac{(3\alpha_a^{kt} - 1)^{\frac{2}{3}} \alpha_{na}^{kt}}{(3\alpha_{na}^{kt} - 1)^{\frac{2}{3}} \alpha_a^{kt}} p_a^{kt}, \tag{4.5}$$

where p_{na}^{kt} is the new coalition's payment policy and α_{na}^{kt} denotes new coalition's data quality requirements at stage t.

The cost allocated for each strong requester in the new coalition is $p_{nai}^{kt} = \frac{v_i^k}{\sum_{i \in M\bar{M}} v_i^k} p_{na}^{kt}$. Referring to the proof of Proposition 4.3.5, strong requesters' utilities will be damaged if any of them form any kind of new coalitions. Thereby, none rational strong requesters will form any kind of coalitions. The coalition form stability has been proved. □

Theorem 4.3.7. *The coalition a formed in the dynamic Stackelberg MCS game $Z_k^t =< M^{kt}, N, (p_i^{kt}), (x_{ij}^{kt}), (\bar{u}_{ri}^{kt}), (\bar{u}_{pj}^{kt}) >, k \in Q$ is stable.*

Proof. Combining Proposition 4.3.5 and Proposition 4.3.6, obviously the coalition a is stable. □

Because the stability of the coalition is guaranteed, then both requesters' and participants' strategies are stable, that is to say the match between requesters and participants are stable. Thus we can eliminate the instability of the system brought by the requesters' time variability.

4.3.5 Pseudo-code of the updated matching algorithm at stage t

At any stage t, the strong requesters and new requesters will play the game according to the strategy of the competitive game in stage 0, and the new weak requesters will join the coalition and play the game according to the updated strategy. We will repeat this process until no new requesters are added and all requesters have completed data collection. Therefore, after obtaining both requesters' and participants' updated strategies, we use an updated algorithm of Algorithm 4.1 for the match between requesters and participants which is finding the optimal i for each participant with updated strategies. The pseudo-code of the matching algorithm in stage t is summarized in Algorithm 4.2.

Table 4.2: Updated Matching Algorithm Between Requesters and Participants at Stage t

Input: new requesters M_n^{kt}.
Output: matching results for requesters and participants.
Algorithm Steps:
1. Determine whether there is a requester that completes data collection in stage $t-1$.

2. For each $i \in M^{k(t-1)} \cup M^{kt}$:
   a) If $w_i^{k(t-1)} < \varepsilon$, requester i is classified as weak and joins M_o^{kt}.
3. For each $i \in (M^{k(t-1)} \cup M^{kt}) - M_o^{kt}$:
   a) If $v_i^k < \max_{s \in M_o^{kt}}$, requester i is classified as weak and joins M_o^{kt}.
   b) Update weak requester set M_o^{kt}.
4. Update the coalition's payment policy p_a^{kt} according to Proposition 4.3.3.
5. Return to Algorithm 4.1.
6. Repeat Steps 1–5 until no new requesters are joining and all requesters have completed data collection.

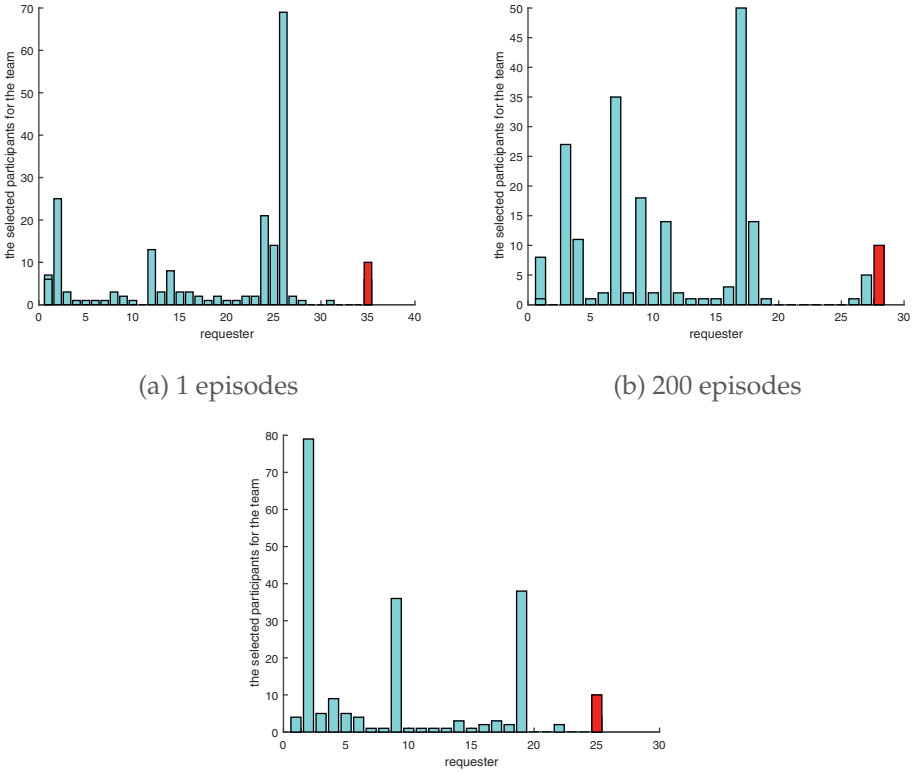
4.4 Simulations

In this section, we conduct simulations to validate the performance of our proposed two algorithms with time variability, and to prove that the simulation results are in agreement with the theoretical results. There are 100 initial requesters, 200 participants and at the beginning of every game, there will be 10 new requesters randomly added to the game. The value of data $k \in Q$ to requester v_i^k is randomly generated from 300 to 400, data quality requirement α_i^k is a random number between 10 and 25 and the unit costs of each participant is between 50 and 100. Unless otherwise specified, all simulations are the above parameters.

4.4.1 Simulation results on the matching algorithm at stage t

In this part, we conduct simulations about the first algorithm, the matching algorithm between requesters and participants, to prove that our proposed

algorithm can be achieved in polynomial time and our theory can be achieved in practice.



(a) 1 episodes

(b) 200 episodes

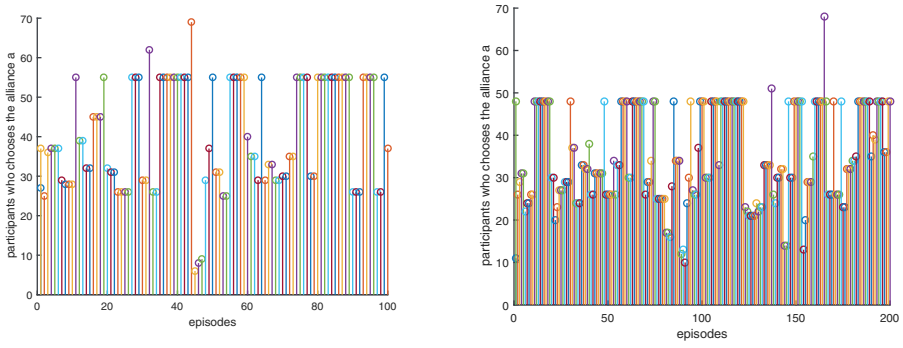
(c) 500 episodes

Figure 4.4: Updated match results of requesters in Algorithm 4.2

With regard to Fig. 4.4, the horizontal coordinate of the red column represents the coalition formed by weak requesters, the remaining horizontal axis means strong requesters and new requesters, and the length of the column in the histogram represents the number of participants matched to this re-

quester. Vividly the coalition formed by weak requesters can collect data and there will be no monopoly which equals to Proposition 4.3.3. Fig. 4.4(b) and Fig. 4.4(c) are the matching results of requesters and participants after 200 and 500 episodes respectively. It is safe to say that the coalition by weak requesters can always collect data and there will be no monopoly which equals to Proposition 4.3.3 no matter how many iterations.

4.4.2 Simulations that the coalition Will definitely match participants



(a) 100 episodes

(b) 200 episodes

Figure 4.5: Coalition match results by Algorithm 4.2

As we can see from Fig. 4.5, after several games, the coalition will always be able to match some participants which is in line with Proposition 4.3.3. Taking Definition 4.3.1 into consideration, any requester who failed to match participants in the last stage will join coalition a in this stage and based on Fig. 4.5 can certainly match some participants. In summary, any requesters after two rounds of the game will match some participants which is one of the most important conclusions of this chapter. Compared with other methods

Table 4.3: Requesters Chosen Rate of Different Algorithms

algorithm	OFF in (Nie et al. 2020)	ON-DYN in (Nie et al. 2020)
requesters chosen rate	0.4	0.3
algorithm	ON-SEG in (Nie et al. 2020)	TDMC in (Yang et al. 2018)
requesters chosen rate	0.24	0.85
algorithm	RA in (Song et al. 2020)	Pre-only in (Song et al. 2020)
requesters chosen rate	0.2	0.6
algorithm	Cost only in (Song et al. 2020)	cTaskMat in (Song et al. 2020)
requesters chosen rate	0.3	0.6
algorithm	MPRA in (Shinkuma et al. 2020)	GBA in (Shinkuma et al. 2020)
requesters chosen rate	0.6	0.6

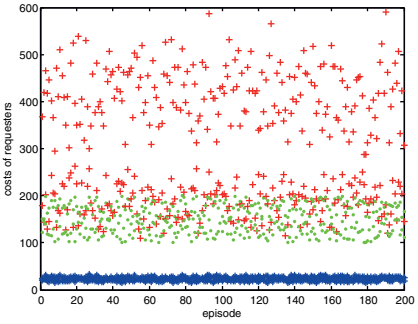
in (Nie et al. 2020; Shinkuma et al. 2020; Yang et al. 2018; Song et al. 2020), our method obviously improved requesters chosen rate since even without time variability, in (Nie et al. 2020; Shinkuma et al. 2020; Yang et al. 2018; Song et al. 2020) there will always be requesters failing to match participants after several episodes let alone taking the time variability of requesters into consideration.

To further show the significant improvement in the requesters chosen rate of the algorithm in this chapter, the requesters chosen rate in Table 4.3 are based on their optimal results in their paper (Nie et al. 2020; Shinkuma et al. 2020; Yang et al. 2018; Song et al. 2020).

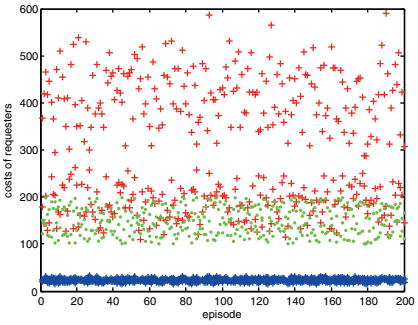
4.4.3 Simulations on cost reduction

In this part, we conduct simulations about Algorithm 4.2, to prove that our proposed algorithm can be achieved in polynomial time, and requesters costs can be greatly reduced.

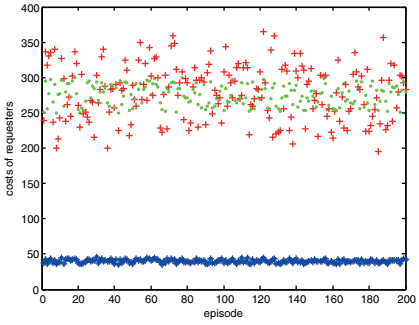
As to Fig. 4.6 in which the green dots represent costs in (Y. Li et al. 2019), the red dots represent costs in (Song et al. 2020) and the blue dots represent costs



(a) costs of requester i



(b) costs of requester i with parameters set one



(c) costs of requester i with parameters set two

Figure 4.6: After forming the coalition, costs of requester i

in this chapter, the reduction in requesters' costs is already very obvious. To rule out the particularity brought about by the parameters, we change the value of data $k \in Q$ to requester v_i^k into randomly generated from 100 to 200, data quality requirement α_i^k into a random number between 5 and 10 and the unit costs of each participant is between 25 and 50 in Fig. 4.6(b) With regard

Table 4.4: Weak requesters' costs of Algorithm 4.1 and Algorithm 4.2

parameters	$M = 100, LM = 63$	$M = 200, LM = 125$	$M = 300, LM = 189$	$M = 500, LM = 354$
cost(Algorithm 1,Algorithm 2)	(39.8052,1.2571)	(40.4346,0.6014)	(39.7719,0.3593)	(40.2817,0.2062)
parameters	v_i^{kt} ∼(100,150),$LM = 63$	v_i^{kt} ∼(150,200),$LM = 125$	v_i^{kt} ∼(200,250),$LM = 118$	v_i^{kt} ∼(250,300),$LM = 121$
cost(Algorithm 1,Algorithm 2)	(39.8052,1.2571)	(73.1077,0.9687)	(89.1153,1.3091)	(106.5237,1.6227)
parameters	c_j ∼(5,10),$LM = 63$	c_j ∼(10,15),$LM = 131$	c_j ∼(15,20),$LM = 120$	c_j ∼(20,25),$LM = 132$
cost(Algorithm 1,Algorithm 2)	(39.8052,1.2571)	(40.113,0.5217)	(39.7802,0.5382)	(40.3646,0.5126)

to Fig. 4.6(c), the value of data $k \in Q$ to requester v_i^k is randomly generated from 200 to 300, data quality requirement α_i^k into a random number between 25 and 35 and the unit costs of each participant is between 10 and 25. Apparently, the costs in this chapter are still greatly smaller than the costs in other paper (Song et al. 2020; Y. Li et al. 2019). It is safe to conclude that, by weak requesters forming a coalition, the costs of requesters are remarkably reduced.

4.4.4 Numerical analysis on cost reduction

In this part, we conduct simulations about Algorithm 4.1 and Algorithm 4.2, to prove that weak requesters' costs can be greatly reduced through forming a coalition. To rule out chance, under each set of parameter settings we repeat the game 200 times and take the average of the average costs of all weak requesters.

It is safe to conclude that, by weak requesters forming a coalition, the costs of requesters are remarkably reduced by at least 96%.

4.5 Conclusion

In summary, we proposed a dynamic coalition-based incentive mechanism with time variability. During the dynamic incentive mechanism, update rules

were used to guarantee the stability of the match between requesters and participants.

From Fig. 4.2 and Fig. 4.3, without updated rules, if we repeat Algorithm 4.1 in different stages, there will always be some weak requesters who cannot match participants. However, with the updated algorithm as Algorithm 4.2, we can prove that whatever requester will be reach a 100% chosen rate after at most two games is a significant improvement in the requester chosen rate compared with other methods in (Nie et al. 2020; Shinkuma et al. 2020; Yang et al. 2018; Song et al. 2020). More than that, with the updated strategies, the costs of requesters have been greatly reduced compared with other methods (Song et al. 2020; Y. Li et al. 2019).

For example, in health-data crowdsensing applications or environmental monitoring applications and so on, with our model and algorithms, applications are free to start or stop data collection with their own needs. That is to say, if they feel they have collected enough data in that time frame, they are free to leave the system and are welcome to rejoin if they ever need more data. With our model and algorithms, requesters have more freedom and choices. By proving the stability of our system, requesters do not have to worry about defectors because as long as the rules are followed, every requester can collect the data with the highest utilities.

However, in this chapter, we failed to consider the time limit of requesters. Just because any requester can match participants after at most two games does not mean any requester can match enough participants within a given amount of time. In our future work, we plan to focus on the time restrictions of requesters and implement a more realistic MCS platform.

5

Irregular Time Variability

This chapter is based on: Zhilin Xu, Hao Sun, and Panfei Sun. "Irregular mobility: A dynamic alliance formation incentive mechanism under incomplete information". In: *Information Sciences* 712 (2025), p. 122155.

As the name of Mobile Crowdsensing suggests, participants will change over time, which means different participants will arrive and leave the system separately and their costs also change over time. The uncertainty of participants caused by participants irregular time variability will generate incomplete information, and requesters will be unaware of participants' arrival times, departure times, and cost information. Consequently, the match between requesters and participants by requesters-centered matching algorithm is infeasible because requesters cannot decide on matching strategies because of the lack of knowledge about which participants can be matched with. Besides, on account of the irregular time variability of participants, the match is also unstable due to the irregular changes in participants, requesters may become dissatisfied with the original matching strategies. For incomplete information, a participants-centered matching algorithm where participants have the dominant power is proposed to eliminate the impact of participants' uncertainty. As to the instability of the match, a requesters classification algorithm that could reduce the computational complexity to polynomials is used to update the matching rules. We also promise that with irregular time variability in our algorithm, each requester receives data within the first two stages in which they participate.

5.1 Introduction

With the rapid development of smart mobile devices, Mobile Crowdsensing (MCS) has attracted increasing research attention (Jin, Su, and Nahrstedt 2017; Yu and Cannella Jr 2013; Shinkuma et al. 2020). Pioneering MCS systems include *TrackYourTinnitus* for prescription of ear ringing (Pryss et al. 2015), *Creek Watch* for water pollution (Kim et al. 2011), *NoiseTube* for noise monitoring (Maisonneuve et al. 2009), *Signal Guru* for traffic monitoring (Koukoumidis, Peh, and Martonosi 2011) and so on.

To make the mechanism from Chapter 4 more realistic and have a wider range of applications we assume that participants feature non-periodic time variability. This means that presence of participants in the sensing activity is not predetermined, and requesters have incomplete information on the location and sensing ability of participants. Participants can join the sensing activity in real-time or withdraw from the sensing activity.

This causes two problems. First, since requesters do not have information about the location of participants, requesters do not know which participants they can be matched with. Second, requesters have incomplete information about participants' costs. This lack of information creates inefficiencies in the requesters-centered matching approach.

Therefore, in Chapter 5, we adopt an alternative perspective that effectively addresses these difficulties. We consider a variation of the model in Chapter 4, where both the set of requesters and the set of participants can vary. Teams are called data types and each data type has multiple quantity levels. This way requesters can prevent that participants provide many small or scattered contributions. Requesters request data types based on their data requirements. Participants can choose which type of data they want to provide.

Because data types are independent of each other, we only need to consider one data type as a typical example in the remainder of this chapter.

For each data type, the relationship between requesters and participants at the first stage is modeled as an incomplete-information Stackelberg game. The incomplete information concerns participants costs and the set of available participants. Requesters do not know participants' costs for collecting and uploading data. Requesters do know how many participants are potentially available in the system, but they do not know the time periods in which a particular participant will join. Otherwise the game is identical to that in Chapter 3 and 4.

We first solve the equilibrium of the complete information variation of the game. As it turns out the requester's equilibrium strategies are related with participants' costs, which is unknown information for the requesters. Thus this is not the equilibrium of the incomplete information game. Therefore, we propose a parameter estimation algorithm to learn participants' cost parameters. Furthermore, since requesters have incomplete information on availability of participants in each given time period, we let participants choose one requester to be matched with. Requesters must first announce data requirements and payment schemes, so that participants know which requesters are available in each given time period. This way, the participants-centered matching algorithm is able to deal with issues caused by the incomplete information.

The matching results of the former time period may not be effective in this time period due to changes in requesters and participants. Based on the matching results in former time period, requesters who were not matched with any participants are called weak requesters in this time period. A classification algorithm is used to differentiate between different types of re-

questers. Weaker types of requesters will form a coalition to request data together and compete against stronger types of requesters. Weaker types of requesters will keep their total required data amount, but update the minimum data quantity requirements and payments. The coalition's data quantity requirement is the maximum minimum data quantity requirement of its members to guarantee all coalition members can get data satisfying quantity requirement. The coalition's equilibrium payment is the minimum payment that ensures the coalition can always match with participants.

Through simulations, we demonstrate that the requester classification algorithm still ensures the coalition is always matched with participants. Generally speaking, the participants-centered matching algorithm based on requesters classification is polynomial time solvable and stable with time variability. Through simulations, we show that with high probability each requester receives data within the first two stages in which they participate.

The main contributions of this chapter are summarized as follows:

- Taking the non-periodic time variability of participants into account, we propose a dynamic coalition-based participants-centered matching algorithm.
- We construct a participants-centered matching algorithm to deal with the inefficiencies of the matching algorithms in previous chapters caused by time variability. In the participant-centered matching algorithm, participants choose requesters rather than vice versa.
- To reduce the computational complexity of updating rules, we propose a new requester classification algorithm. The algorithm guarantees the stability of the match even under time variability. Moreover, it is computable in polynomial time.

- By simulations, we show that with high probability each requester will receive data within the first two stages in which they participate. Compared with other algorithms, the matching rate between requesters and participants is significantly increased.

The remainder of this chapter is organized as follows. The participants-centered matching algorithm in the first time period is introduced in Section 5.2. Section 5.3 describes the updated matching algorithm. Section 5.4 presents the simulation results. Finally, Section 5.5 summarizes the contributions of this chapter.

5.2 Participants-centered matching algorithm in the initial state

To eliminate the impact of participants' uncertainty due to irregular time variability, we discretize the dynamic mechanism into T stages, where requesters and participants can only leave, join, or modify their parameters at the beginning of each stage. In this section, we consider the initial state of the participants-centered matching algorithm, where without historical data requesters are all in competition.

5.2.1 Problem initialization

Before we derive the specific algorithm, a Stackelberg MCS model, which effectively describe the precedence relationship between requesters and participants and can calculate theoretical solutions, is used to model the relationship between requesters and participants. In the MCS system, requesters will

broadcast tasks with payment policies and demands to the platform. Participants then collect data based on these task demands and payment policies. Coincidentally, in the Stackelberg model, leaders first announce their strategies, and then followers will decide on their strategies based on the leaders' strategies. As a consequence, in this section, we model the relationship between requesters and participants as a Stackelberg MCS game. In which, requesters as leaders first announce tasks, demands, and payment policies, and participants, as followers, decide which requester to match with and at what level of participation, based on the requesters' strategies.

Symbol description

Let $M^0 = \{1, 2, ..., m\}$ denote the set of requesters at stage 0 and $N^0 = \{1, 2, ..., n\}$ the set of participants. Requesters first announce their tasks $S^0 = \{S_1, S_2, ..., S_m\}$ to the platform, where $S_i = \{S_{i1}, S_{i2}, ..., S_{il}\}, i \in M^0$ is the set of tasks needed by requester i. To better calculate the equilibrium, the concept of "data type", which uniquely defines jointly owned tasks, is proposed to regroup requesters as in Section 3.2 of Chapter 3. Each data type classically represents one kind of task and every requester could request one or more data types. All data types are independent of each other, thus we only need to consider data type $k \in Q^0$ as a typical example. We use $Q^0 = \{1, 2, ..., q\}$ to determine the set of data types at stage 0. We use M^{k0} to define the set of requesters who request data type $k \in Q^0$ at stage 0.

For data type $k \in Q^0$, requesters as leaders first announce their prices p_{ij}^{k0} and their participation level demand X_i^{k0} to the platform. Participants' participation levels must exceed the requesters' requirements to be able to match. p_{ij}^{k0} is the unit price requester i pays for participant j for data type $k \in Q^0$ at stage

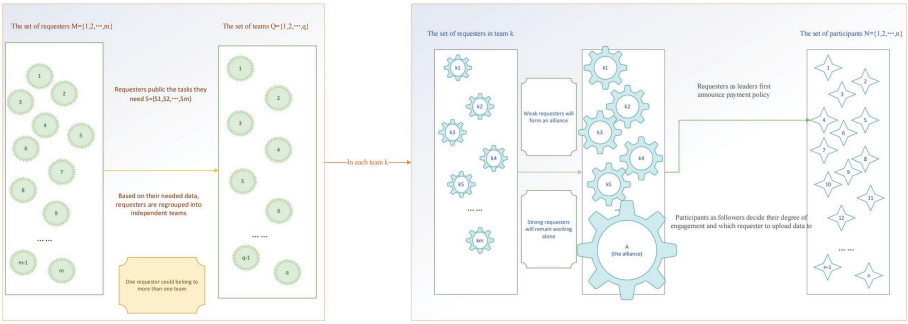


Figure 5.1: The illustration of the Stackelberg MCS system in this chapter

0. Based on each requester's payment policy and participation level requirement, participants as followers will decide to match which requester in what kind of participation degree as shown in Fig. 5.1. Thereby, each participant has two relevant strategies $\bar{I}_{ij}^{k0}$ and X_{ij}^{k0}, where $\bar{I}_{ij}^{k0}$ is an indicative function represents whether participant j will be matched with requester i or not for data type $k \in Q^0$ at stage 0, and $X_{ij}^{k0}, (X_{ij}^{k0} \geq 0)$ is participant j's degree of participation ($X_{ij}^{k0} = 0$ equals participant j not participate). X_{ij}^{k0} is determined by participants' working efficiency and devoting time. Limited by participants' time, energy and sensing equipment, participants' participation level is not infinite. $\bar{X}_j^{k0}$ is each participant's maximum participation level, in any case, the actual participation level of participants will not exceed their maximum participation level and must be larger than the requesters' participation degree demand.

$$\bar{I}_{ij}^{k0} = 0, \quad \text{participant } j \text{ not upload data to requester } i \in M^{k0},$$
$$\bar{I}_{ij}^{k0} = 1, \quad \text{participant } j \text{ uploads data to requester } i \in M^{k0}.$$

Model definition

There are three main factors affecting participants' costs, hardware costs, fixed costs, and their costs. Hardware costs typically include the consumption, computing, network, etc. Fixed costs often refer to the loss of devices. While participants' costs mean the time, effort and so on. We assume that C_{ij}^{k0} is a strongly convex quadratic function as in many other literature proposals (Capponi et al. 2019; Sedghani et al. 2021),

$$C_{ij}^{k0} = \frac{1}{2} a_{ij}^{k0} {X_{ij}^{k0}}^2 + b_{ij}^{k0} X_{ij}^{k0} + c_j^{k0},$$

where $a_{ij}^{k0}, b_{ij}^{k0} > 0$ are parameters depending on both participants' availability level and task difficulty level and $c_j^{k0} \geq 0$ is participants' own costs which is irrelevant with requesters.

Sequentially, utilities of participants can be calculated by

$$u_{pj}^{k0}(X_{ij}^{k0}) = \sum_{i \in M^{k0}} \bar{I}_{ij}^{k0} (p_{ij}^{k0} X_{ij}^{k0} - C_{ij}^{k0}).$$

We define requester i's income as $v_i^k X_{ij}^{k0}$, where v_i^k is the unit value of data type $k \in Q^0$ for requester i which will not change over time, thus v_i^k is irrelevant with stage. What is more, v_i^k is related only to the data type, not to which participant collected it.

Therefore, the utility to requester i for data type $k \in Q^0$ at stage 0 is

$$u_{ri}^{k0}(X_{ij}^{k0}) = \sum_{j \in N^{k0}} \bar{I}_{ij}^{k0} (v_i^k X_{ij}^{k0} - p_{ij}^{k0} X_{ij}^{k0}).$$

All requesters and participants are rational which is the common knowledge among requesters and participants. Thanks to rationality, both $u_{ri}^{k0}(X_{ij}^{k0})$ and $u_{pj}^{k0}(X_{ij}^{k0})$ should be positive. Without prior knowledge, it is difficult for requesters to decide whether they should compete or cooperate for the lack of accurate perception of the intensity of competition. Therefore, in stage 0 all requesters are in competition with each other.

Definition 5.2.1. *For any data type $k \in Q^0$, the Stage-0 Stackelberg game is defined as*

$$Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0,$$

where

- M^{k0} *denotes the set of requesters requiring data type k in Stage 0,*
- N^0 *denotes the set of participants participating in the mobile crowdsensing system in Stage 0,*
- $P^{k0} = \prod_{i \in M^{k0}} P_i^{k0}$ *denotes the strategy space of the applications, where* $P_i^{k0} = \left\{ p_{ij}^{k0} \mid \varphi_{ij}^{k0} \geq 0, j \in N^0 \right\}$ *represents the set of payment strategies that requester i can adopt,*
- $X^{k0} = \prod_{j \in N^0} X_j^{k0}$ *denotes the strategy space of the participants, where* $X_j^{k0} = \left\{ x_{ij}^{k0} \mid \alpha_i^{k0} \leq x_{ij}^{k0} \leq \bar{x}_{ij}^{k0}, i \in M^{k0} \right\}$ *represents the set of participation strategies that participant j can adopt,*
- $U_r^{k0} : Y^{k0} \times X^{k0} \to \mathbb{R}^{|M^{k0}|}$ *denotes the utility function of the requesters,*
- $U_p^{k0} : P^{k0} \times X^{k0} \to \mathbb{R}^{|N^0|}$ *denotes the utility function of the participants.*

Definition 5.2.2. *In the Stackelberg MCS game*

$$Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0,$$

the set $T(p_{ij}^{k0}) \subset [0, +\infty)$ *defined for each* p_{ij}^{k0} *by*

$$T(p_{ij}^{k0}) = \{X_{ij}^{k0} \in [0, +\infty) : u_{pj}^{k0}(X_{ij}^{k0}) \geq u_{pj}^{k0}(X'^{k0}_{ij}), \forall X'^{k0}_{ij} \in [0, +\infty), i \in M^{k0}\}. \tag{5.1}$$

is the optimal response set of participant j *to the strategy* p_{ij}^{k0} *of requester* i*. Whatever requesters' payment policies are, participants would always choose the participation degree to maximize their utilities.*

A strategy p_{ij}^{k0*} *is called a Stackelberg MCS competition equilibrium for the leader (requester* i*) , if*

$$\min_{X_{ij}^{k0*} \in T(p_{ij}^{k0*})} u_{ri}^{k0} = \max_{p_{ij}^{k0}} \min_{X_{ij}^{k0} \in T(p_{ij}^{k0})} u_{ri}^{k0}. \tag{5.2}$$

As far as risk-avoid, rational requesters are concerned, despite of participants' participation levels, they will always decide on strategies that can maximize their minimum utilities.

Any element $X_{ij}^{k0*} \in T(p_{ij}^{k0*})$ is an optimal strategy for the follower (participant j). The pair $(p_{ij}^{k0*}, X_{ij}^{k0*})$ is a Stackelberg MCS competition solution for the game $Z_k^0 =< M^{k0}, N^{k0}, P_{ij}^{k0}, \bar{X}_{ij}^{k0}, (u_{ri}^{k0}), (u_{pj}^{k0}) >, k \in Q^0$ with requester i as the leader.

We solve for equilibriums by working backwards. First we figure out each participants' optimal participation degree, no matter which requester participant will match.

Theorem 5.2.3. *In the Stackelberg MCS game*

$$Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0,$$

for any requester, the rational participant j's optimal response set is $T(p_{ij}^{k0}) = \{\frac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}\}$.

Proof. For now participants do not need to consider which requester to match, so eq. 5.1 can be translated as $u_{pj}^{k0}(X_{ij}^{k0}) = p_{ij}^{k0}(X_{ij}^{k0}) - \frac{1}{2}a_{ij}^{k0}(X_{ij}^{k0})^2 - b_{ij}^{k0}(X_{ij}^{k0}) - c_j^{k0}$ and take the derivation, we have

$$\frac{\mathrm{d}u_{pj}^{k0}}{\mathrm{d}X_{ij}^{k0}} = p_{ij}^{k0} - a_{ij}^{k0}X_{ij}^{k0} - b_{ij}^{k0}.$$

Judging from above, eq. 5.1 is an increment and then subtract equation, hence

$$X_{ij}^{k0} = \frac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}.$$

This completes the proof. □

In turn, every participant's optimal response set for each requester is $T(p_{ij}^{k0}) = \frac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}$. Participants' rationality is common knowledge among requesters in the incomplete information game

$Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0$, so Theorem 5.2.3 is regarded as known information during the process of solving equilibrium by requesters. After working out participants' optimal participation degree, we move on to calculate each requester's optimal price for all participants.

Theorem 5.2.4. *For the Stackelberg MCS competition game* $Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0$, $p_i^{k0*} = \frac{v_i^k + b_{ij}^{k0}}{2}$ *is the Stackelberg MCS competition equilibrium for requester* i.

Proof. Because participants' rationality is common knowledge in the game, we can take Theorem 5.2.3 into eq. 5.2 and get

$$\begin{aligned} u_{ri}^{k0}(X_{ij}^{k0}) &= \sum_{j \in N^{k0}} \bar{I}_{ij}^{k0}(v_i^k X_{ij}^{k0} - p_{ij}^{k0} X_{ij}^{k0}) = \sum_{j \in N^{k0}} \bar{I}_{ij}^{k0}((v_i^k - p_{ij}^{k0}) \frac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}) \\ &= \sum_{j \in N^{k0}} \bar{I}_{ij}^{k0}(\frac{(v_i^k + b_{ij}^{k0}) p_{ij}^{k0} - {p_{ij}^{k0}}^2 - v_i^k b_{ij}^{k0}}{a_{ij}^{k0}}). \end{aligned}$$

Seeking derivatives for p_{ij}^{k0},

$$\frac{\mathrm{d}u_{ri}^{k0}}{\mathrm{d}p_{ij}^{k0}} = \sum_{j \in N^{k0}} \bar{I}_{ij}^{k0}(\frac{v_i^k + b_{ij}^{k0} - 2p_{ij}^{k0}}{a_{ij}^{k0}}).$$

The equation above is an increment and then subtract equation, hence

$$p_{ij}^{k0} = \frac{v_i^k + b_{ij}^{k0}}{2}.$$

All in all, requester i's best strategy is $p_{ij}^{k0} = \frac{v_i^k + b_{ij}^{k0}}{2}$. □

Judging from requesters' equilibriums, requesters' payment policies are related with participants' costs parameters a_{ij}^{k0} and b_{ij}^{k0} which are unknown to requesters. Therefore, the incomplete information for requesters are the decision set and players set (participants set) of the Stackelberg MCS game $Z_k^0 = \left\langle M^{k0}, N^0, P^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0$.

After determining the optimal matching strategies, rational participants can choose one requester to match to maximize their utilities.

$$i = \arg\max_{i \in M^{k0}} u_{pj}^{k0}(i) = \arg\max_{i \in M^{k0}} (\frac{(p_{ij}^{k0} - b_{ij}^{k0})^2}{a_{ij}^{k0}} - c_j^{k0}).$$

5.2.2 Algorithm construction

From Theorem 5.2.3 and Theorem 5.2.4, the optimal matching strategies of requesters are connected with participants' parameters a_{ij}^{k0}, b_{ij}^{k0} which in reality are unknown information to requesters, hence requesters need to estimate participants' unknown parameters a_{ij}^{k0}, b_{ij}^{k0}.

Not all potential participants are suitable participants for requesters because all requesters have unique task requirements. When estimating a_{ij}^{k0}, b_{ij}^{k0}, we must also identify applicable participants for each requester. Based on the estimation algorithm in (Sedghani et al. 2021), we develop an estimation algorithm that not only estimates the parameters but also filters the appropriate participants for each requester individually.

At the start, the system studies the maximum participation degree $\bar{X}_j^{k0}$ of each participant by choosing one requester $i \in M^{k0}$ and setting a very large number for its payment policy. For any requester $s \neq i$, the system will set $\varepsilon \to 0$ for their payment policies. Therefore, any rational participants will

match the chosen requester and because of the very large price, the participation degree that each participant uploads is their maximum participation degree. A requester can only be matched if the participant's maximum participation degree exceeds the degree required by the requester.

After that, the system estimates the parameters a_{ij}^{k0}, b_{ij}^{k0} for each requester. Intuitively, the system will generate a price $p_{ij}^{k0} = \frac{v_i^k}{2}$ for the chosen requester $i \in M^{k0}$, and a price $p_{ij}^{k0} = \varepsilon \to 0$ for any requester $s \neq i$. Thus, any rational participant would match the chosen requester with a participation degree $X_{ij}^{k0} = \frac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}$. If the actual participation degree is the same as participant's maximum participation degree, then we would reduce the price to $p_{ij}^{k0} = p_{ij}^{k0} - \varepsilon$. Repeating this process until the actual participation degree is less than the maximum participation degree. With p_{ij}^{k0} as a known information, we could get an equation of a_{ij}^{k0} and b_{ij}^{k0}. Then the system will alter the price $p_{ij}^{k0} = \frac{v_i^k}{2}$ to $p_{ij}^{k0} = \frac{v_i^k}{2} - \varepsilon$. Thus, we could get another equation of a_{ij}^{k0} and b_{ij}^{k0}. By solving the equations simultaneously, we can get the specific values of a_{ij}^{k0} and b_{ij}^{k0}. Repeat for each $i \in M^{k0}$, we can get all a_{ij}^{k0} and b_{ij}^{k0}.

Pseudo-code of the estimation algorithm

The pseudo-code of the estimation algorithm is summarized in Table 5.1.

Table 5.1: Estimation algorithm of participants' parameters

Input: requester set $M^{k0} = \{1, 2, \ldots, m\}$; participant set $N^{k0} = \{1, 2, \ldots, n\}$.
Output: maximum participation degree $\bar{X}_j^{k0}$; parameters a_{ij}^{k0} and b_{ij}^{k0}.
Choose a requester $l \in M^{k0}$.
1. For each participant $j \in N^{k0}$ do:

2. For each requester $i \in M^{k0}$ do:
   If $i = l$ then:
      Set $p_{ij}^{k0} = P$ $(P \to \infty)$.
      Participants upload participation degree X_{ij}^{k0}.
   End if.
End for.
Set $\bar{X}_j^{k0} = X_{ij}^{k0}$.
End for.

3. For each requester $i \in M^{k0}$ do:
   Set $p_1 = \dfrac{v_i^k}{2}$.
   4. For each participant $j \in N^{k0}$ do:
      If $\bar{X}_j^{k0} > X_i^{k0}$ then:
         Participants upload participation degree x_{ij}^1.
         Obtain one equation of a_{ij}^{k0} and b_{ij}^{k0}.
         Set $p_2 = p_1 + \varepsilon$.
         Participants upload participation degree x_{ij}^2.
         Obtain another equation of a_{ij}^{k0} and b_{ij}^{k0}.
         If $x_{ij}^2 = \bar{X}_j^{k0} = X_{ij}^{k0}$ then:
            Set $p_2 = p_2 - 2\varepsilon$.
            Participants upload participation degree x_{ij}^3.
            Set $x_{ij}^2 = x_{ij}^3$.
         End if.
      End if.
      5. Compute $a_{ij}^{k0} = \dfrac{p_1 - p_2}{x_{ij}^1 - x_{ij}^2}$.
      6. Compute $b_{ij}^{k0} = p_1 - x_{ij}^1 \dfrac{p_1 - p_2}{x_{ij}^1 - x_{ij}^2}$.
   End for.
End for.

With Algorithm 5.1, we are able to solve the incomplete information about participants' costs to get the optimal payment policies of requesters.

5.2.3 Pseudo-code of participants-centered matching algorithm

After determining the optimal strategies for both requesters and participants, we employ an easily implementable algorithm to facilitate the matching process between requesters and participants. The main idea of the matching algorithm is to find an optimal requester i for each participant. The pseudo-code of the participants-centered matching algorithm in stage 0 is summarized in Table 5.2.

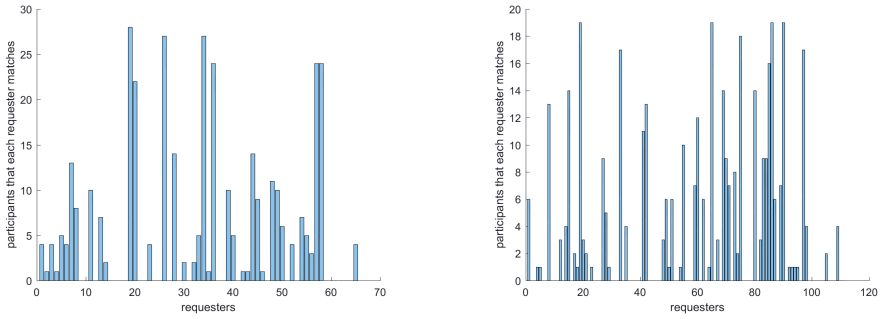
Table 5.2: Participants-centered matching algorithm between requesters and participants at stage 0

Input: requester set $M^0 = \{1, 2, \ldots, m\}$, participant set $N^{k0} = \{1, 2, \ldots, n\}$, and task set $S = \{S_1, S_2, \ldots, S_m\}$.

Output: payment policy p_{ij}^{k0}, participation degree X_{ij}^{k0}, and matching results for requesters and participants.

1. Reclassify requesters based on data types $Q^0 = \{1, 2, \ldots, q\}$.
2. Call Algorithm 5.1.
3. For each requester $i \in M^{k0}$ do:
   Calculate the requester's optimal payment policy p_{ij}^{k0} based on Theorem 5.2.4.
   End for.
4. With regard to equation 5.2.3, update the participation degree as $X_{ij}^{k0} = \dfrac{p_{ij}^{k0} - b_{ij}^{k0}}{a_{ij}^{k0}}$.
5. Initialize $I = 0$ and $U = 0$.
6. For each participant $j \in N^{k0}$ do:

7. For each requester $i \in M^{k0}$ do:
   If $\bar{X}_j^{k0} > X_i^{k0}$ then:
      If $u_{pj}^{k0}(X_{ij}^{k0}) > U$ then:
         Set $I = i$ and $U = u_{pj}^{k0}(X_{ij}^{k0})$.
      End if.
   End if.
End for.
End for.



(a) 1 episode (b) 500 episodes

Figure 5.2: Matching results of requesters by Algorithm 5.2

Fig. 5.2 are examples about the match between requesters and participants in one and 500 games based on Algorithm 5.2. There are 100 initial requesters, 100 participants, each stage randomly generates 1 to 5 new requesters and participants, the maximum participation degree of participants is randomly generated from 50 to 250, the parameter a_{ij}^{k0} is randomly generated from 1 to 16, the parameter b_{ij}^{k0} is between 1 and 11, and the parameter c_j^{k0} is between 1 and 6. The task value of requester i for data type $k \in Q$ v_i^k is randomly

generated from 100 to 150, data quality requirement α_i^k is a random number between 10 and 25.

In Fig. 5.2, the x-coordinate represents requesters and the blue pillar of each x-axis coordinate points is the number of participants matched by the requester. The label of the newly added requester is appended to the original requester.

From Fig. 5.2, if we simply repeat the Stackelberg MCS competition game between requesters and participants, then no matter how many iterations, there will always be requesters that cannot be matched to participants. Therefore, in Section 5.3, we will update the matching strategy to increase the requesters' chosen rate and eliminate the instability caused by the irregular time variability.

So far, we have proved that our participants-centered match is feasible with irregular time variability but it may not be stable due to the constant move of both requesters and participants. In the following chapter, to eliminate the instability, updated matching strategies are introduced.

5.3 Updated participants-centered matching algorithm

To avoid the instability brought by irregular time variability and enable more requesters to match participants, in this section we would update Algorithm 5.2.

5.3.1 Requesters classification algorithm

Rather than simply describing the relationship among requesters as competitive, in this section requesters are divided into three categories, weak

requesters, swing requesters and strong requesters to avoid the instability brought by irregular time variability and increase requesters' chosen rate. Weak requesters and swing requesters will form a coalition through the centralized MCS platform to compete against strong requesters and newly arrived requesters.

We divide the whole requesters into three types, weak requesters who failed to collect data in the previous stage $t-1$, swing requesters who collected data in the last stage but have an incentivization to join the coalition in this stage, and strong requesters who gathered data in the last stage and have an incentivization to join the coalition in this stage.

Symbols of the updated requesters

We first recommend the concept "data completion degree" $w_i^{k(t-1)}$ which equals to $\frac{\sum\limits_{j \in N^{k(t-1)}} \bar{I}_{ij}^{k(t-1)} X_{ij}^{k(t-1)}}{W_i^k}$ where W_i^k is the total amount of data with data type $k \in Q^{(t-1)}$ required by requester i. W_i^k, which is given by the requester in advance, is related to the number of reports, the quality of sensing reports, the time range of reports and so on.

At the end of stage $t-1$, for any $i \in M^{k(t-1)}$, we need to verdict whether it completed data collection or not. Therefore, the total degree of completion $\sum\limits_{t'=0}^{t-1} w_i^{kt'}$ and 1 must be compared, in which $\sum\limits_{t'=0}^{t-1} w_i^{kt'}$ is requester i's total data completion degrees in all previous stages. If $\sum\limits_{t'=0}^{t-1} w_i^{kt'} \geq 1$, then requester i's data collection is completed and will leave the system, otherwise requester i will stay in the system. More than that, there will be new requesters joining the system. Without prior knowledge of the new requesters, they will not

decide whether or not to join the coalition. Hence, both strong requesters and new requesters will participate in the dynamic Stackelberg MCS game at stage $t \in \{1, 2, ..., T\}$ according to the strategy in stage 0.

Definition 5.3.1. ***Weak requesters***
Requester $i \in M^{k(t-1)} \cap M^{kt}$ is defined as a weak requester if and only if its data completion degree in the previous stage is less than ε ($0 \leq \varepsilon < 1$ is given by the system).

It is easy to differentiate weak requesters, any requester with a data completion degree $w_i^{k(t-1)}$ less than ε ($\varepsilon \in [0, 1)$ is given by the system) in the last stage $t-1$ is a weak requester. The trouble is differentiating between swing requesters and strong requesters from th remaining requesters. With irregular time variability, it is unreliable to separate swing requesters from strong requesters depending on history data. Because as the stage changes, different parameters will also change. Thus, before we deliver an algorithm to differentiate swing requesters from strong requesters, we first calculate the discriminant condition of the current stage.

Let $M^{kt} = M_s^{kt} \bigcup M_c^{kt} \bigcup M_o^{kt} \bigcup M_g^{kt}$ denote the set of requesters who request data type $k \in Q^t$ at stage t, where M_s^{kt} is the set of swing requesters, M_c^{kt} the set of strong requesters, M_o^{kt} the set of weak requesters, and M_g^{kt} the set of new requesters. $a_{oj}^{kt} = \max\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} a_{ij}^{kt}$ and $b_{oj}^{kt} = \max\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} b_{ij}^{kt}$ are parameters of participants' cost function regarding the . $X_o^{kt} = \max\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} X_i^{kt}$ is the coalition's participation level demand.

Therefore, the utilities of requesters and participants are updated as

$$\bar{u}_{ri}^{kt}(X_{ij}^{kt}) = \sum_{j \in N^{kt}} \bar{I}_{ij}^{kt}(v_i^k X_{ij}^{kt} - p_{ij}^{kt} X_{ij}^{kt}), \quad \forall i \in M_c^{kt} \bigcup M_g^{kt},$$
$$\bar{u}_{ri}^{kt}(X_{oj}^{kt}) = \sum_{j \in N^{kt}} \bar{I}_{ij}^{kt}(v_i^k X_{oj}^{kt} - p_{ij}^{kt} X_{oj}^{kt}), \quad \forall i \in M_s^{kt} \bigcup M_o^{kt},$$

and

$$\begin{aligned} p_{ij}^{kt} &= \frac{v_i^k + b_{ij}^{kt}}{2}, \quad \forall i \in M_c^{kt} \bigcup M_g^{kt}, \\ p_{ij}^{kt} &= \bar{p}_{ij}^{kt} = \frac{v_i^k}{\sum\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} v_i^k} p_{oj}^{kt}, \quad \forall i \in M_s^{kt} \bigcup M_o^{kt}, \end{aligned} \tag{5.3}$$

$\bar{p_{ij}^{kt}}$ is the payment required by each requester in the coalition. In this chapter, we use a method of proportional distribution of data value to allocate costs between requesters.

$$\bar{u}_{pj}^{kt}(X_{ij}^{kt}) = \sum_{i \in M_c^{kt} \bigcup M_g^{kt}} \bar{I}_{ij}^{kt}(p_{ij}^{kt} X_{ij}^{kt} - \frac{1}{2} a_{ij}^{kt} (X_{ij}^{kt})^2 - b_{ij}^{kt} X_{ij}^{kt} - c_j^{kt}), \forall i \in M_c^{kt} \bigcup M_g^{kt},$$
$$\bar{u}_{pj}^{kt}(X_{oj}^{kt}) = \sum_{i \in M_s^{kt} \bigcup M_o^{kt}} \bar{I}_{oj}^{kt}(p_{oj}^{kt} X_{oj}^{kt} - \frac{1}{2} a_{oj}^{kt} (X_{oj}^{kt})^2 - b_{oj}^{kt} X_{oj}^{kt} - c_j^{kt}), \forall i \in M_s^{kt} \bigcup M_o^{kt}.$$

Model update

In this section, we will update the Stackelberg MCS competition model into the coalition-driven Stackelberg MCS model to make sure more requesters can be matched to participants and promise the stability of the match.

Definition 5.3.2. *For a given Stackelberg mobile crowdsensing competition game*

$$Z_k^0 = \left\langle M^{k0}, N^0, Y^{k0}, X^{k0}, U_r^{k0}, U_p^{k0} \right\rangle, k \in Q^0,$$

after taking into account the irregular dynamic changes in requesters and participants, the Stackelberg mobile crowdsensing game at stage t is updated as

$$Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t,$$

where:

- *M^{kt} denotes the set of requesters requesting data type $k \in Q^t$ at Stage t, and N^{kt} denotes the set of participants available to participate in sensing tasks at stage t, where $M^{kt} \subseteq \mathcal{M}$ and $N^{kt} \subseteq \mathcal{N}$;*
- *$Y_{ij}^{kt} = \{y_{ij}^{kt}\}$ denotes the set of payment strategies offered by requester $i \in M^{kt}$ to participant $j \in N^{kt}$, where $y_{ij}^{kt} \in \mathbb{R}_+$;*
- *$X_{ij}^{kt} = \{x_{ij}^{kt}\}$ denotes the set of participation strategies of participant j for requester i, where $x_{ij}^{kt} \in [0, \bar{\eta}_j^{kt}]$;*
- *$\bar{U}_r^{kt} : Y^{kt} \to \mathbb{R}$ and $\bar{U}_p^{kt} : X^{kt} \to \mathbb{R}$ denote the utility functions of requester i and participant j at Stage t, respectively.*

In game Z_k^t, weak requesters would form a coalition to increase their chances of matching participants, swing requesters might also join the coalition to gain more utilities, while strong requesters with newly arrived requesters still choose to compete. In consequence, we only need to update the coalition's strategy which can neither be too small to avoid failing matches nor too large to avert "mutual forbearance" (Yu and Cannella Jr 2013).

Strategy update

Proposition 5.3.3. *In game* $Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t$, *the updated payment policy of the coalition should meet*

$$p_{oj}^{kt} > \sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}}(\frac{v - b_j^{kt}}{2}) + b_{oj}^{kt},$$

where v *is discriminant requester's data value,* a_j^{kt} *and* b_j^{kt} *are parameters of participant* j *for discriminant requester at stage* t.

Proof. The coalition's strategy must meet two conditions, one is the coalition can definitely match some participants, and another is to avoid monopolies. To sum up, participants' utilities of matching the coalition should be higher than matching to some of the strong requesters.

Thus, we should assure that

$$\bar{u}_{pj}^{kt}(o) > \bar{u}_{pj}^{kt}(i), (o \in M_s^{kt} \bigcup M_o^{kt}, \exists i \in M_c^{kt} \bigcup M_g^{kt}, \forall j \in N^{kt}).$$

Combined with utility function, we have

$$\begin{aligned}&\bar{u}_{pj}^{kt}(o) - \bar{u}_{pj}^{kt}(i)\\ &= p_{oj}^{kt} X_{oj}^{kt} - \frac{1}{2} a_{oj}^{kt} (X_{oj}^{kt})^2 - b_{oj}^{kt} X_{oj}^{kt} - c_j^{kt} - p_{ij}^{kt} X_{ij}^{kt} + \frac{1}{2} a_{ij}^{kt} (X_{ij}^{kt})^2 + b_{ij}^{kt} X_{ij}^{kt} + c_j^{kt}.\end{aligned}$$

Referring to participants' optimal participation level in Theorem 5.2.3, we can

get

$$\begin{aligned}\bar{u}_{pj}^{kt}(o) - \bar{u}_{pj}^{kt}(i) =& p_{oj}^{kt}\frac{p_{oj}^{kt} - b_{oj}^{kt}}{a_{oj}^{kt}} - \frac{1}{2}a_{oj}^{kt}(\frac{p_{oj}^{kt} - b_{oj}^{kt}}{a_{oj}^{kt}})^2 - b_{oj}^{kt}(\frac{p_{oj}^{kt} - b_{oj}^{kt}}{a_{oj}^{kt}}) - c_j^{kt} \\ & - p_{ij}^{kt}\frac{p_{ij}^{kt} - b_{ij}^{kt}}{a_{ij}^{kt}} + \frac{1}{2}a_{ij}^{kt}(\frac{p_{ij}^{kt} - b_{ij}^{kt}}{a_{ij}^{kt}})^2 + b_{ij}^{kt}(\frac{p_{ij}^{kt} - b_{ij}^{kt}}{a_{ij}^{kt}}) + c_j^{kt} \\ =& \frac{(p_{oj}^{kt} - b_{oj}^{kt})^2}{2a_{oj}^{kt}} - \frac{(p_{ij}^{kt} - b_{ij}^{kt})^2}{2a_{ij}^{kt}} > 0,\end{aligned}$$

which equals to

$$a_{ij}^{kt}(p_{oj}^{kt} - b_{oj}^{kt})^2 > a_{oj}^{kt}(p_{ij}^{kt} - b_{ij}^{kt})^2.$$

Because participation degree of participants should always be positive, then $p_{ij}^{kt} > b_{ij}^{kt}, \forall i \in M^{kt}, \forall j \in N^{kt}$.

Thus we can get

$$p_{oj}^{kt} > \sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}}(\frac{v - b_j^{kt}}{2}) + b_{oj}^{kt}.$$

This completes the proof. □

Classification method

Before we introduce the categorization algorithm, we need to calculate the classification conditions for data type $k \in Q^t$ at stage t. The only difference between swing requesters and strong requesters is that swing requesters'

utilities will increase by joining the coalition. Accordingly, we need to decide what are the conditions that the requester meets when joining the coalition is going to be beneficial which means $u_{ri}^{kt}(i \in M_s^{kt} \bigcup M_o^{kt}) - u_{ri}^{kt}(i \in M_c^{kt} \bigcup M_g^{kt}) > 0$.

Taking requesters' utility function into consideration, for all $j \in N^{kt}$ we have

$$\begin{aligned}
&u_{ri}^{kt}(i \in M_s^{kt} \bigcup M_o^{kt}) - u_{ri}^{kt}(i \in M_c^{kt} \bigcup M_g^{kt}) \\
=&v_i^k X_{oj}^{kt} - \frac{v_i^k}{\sum\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} v_i^k} p_{oj}^{kt} X_{oj}^{kt} - (v_i^k X_{ij}^{kt} - p_{ij}^{kt} X_{ij}^{kt}) \\
=&v_i^k \frac{p_{oj}^{kt} - b_{oj}^{kt}}{a_{oj}^{kt}} - \frac{v_i^k}{\sum\limits_{i \in M_s^{kh} \bigcup M_o^{kt}} p_{oj}^{kt}} \frac{p_{oj}^{kt} - b_{oj}^{kt}}{a_{oj}^{kt}} - (v_i^k \frac{p_{ij}^{kt} - b_{ij}^{kt}}{a_{ij}^{kt}} - p_{ij}^{kt} \frac{p_{ij}^{kt} - b_{ij}^{kt}}{a_{ij}^{kt}}) > 0,
\end{aligned}$$

in which $a_{oj}^{kt} > a_{ij}^{kt}$ due to $a_{oj}^{kt} = \max\limits_{i \in M_c^{kt} \bigcup M_o^{kt}} a_{ij}^{kt}$ and $p_{ij}^{kt} = \frac{v_i^k + b_{ij}^{kt}}{2}$.

Therefore, we have

$$v_i^k (p_{oj}^{kt} - b_{oj}^{kt}) - \frac{v_i^k}{\sum\limits_{i \in M_s^{kh} \bigcup M_o^{kt}} p_{oj}^{kt}} p_{oj}^{kt} (p_{oj}^{kt} - b_{oj}^{kt}) - (v_i^k \frac{v_i^k - b_{ij}^{kt}}{2} - \frac{v_i^k + b_{ij}^{kt}}{2} \frac{v_i^k - b_{ij}^{kt}}{2}) > 0,$$

which equals to

$$v_i^k (p_{oj}^{kt} - b_{oj}^{kt}) - \frac{v_i^k}{\sum\limits_{i \in M_s^{kt} \bigcup M_o^{kt}} p_{oj}^{kt}} p_{oj}^{kt} (p_{oj}^{kt} - b_{oj}^{kt}) - v_i^k \frac{v_i^k - b_{ij}^{kt}}{2} > 0.$$

On account of requesters' rationality, no requester will pay participants more

than what their data is worth, which means

$$p_{oj}^{kt} < \alpha \sum_{i \in M_s^{kt} \bigcup M_o^{kt}} v_i^k,$$

in which $\alpha \in (0, 1)$.

Converting to

$$(p_{oj}^{kt} - b_{oj}^{kt}) - \alpha(p_{oj}^{kt} - b_{oj}^{kt}) > \frac{v_i^k - b_{ij}^{kt}}{2},$$

which equals to

$$v_i^k < 2(1-\alpha)(p_{oj}^{kt} - b_{oj}^{kt}) + b_{ij}^{kt}.$$

Therefore,

$$v_i^k < 2(1-\alpha)(\sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}} \frac{v - b_j^{kt}}{2}) + b_{ij}^{kt},$$

which is the same as

$$(v_i^k - b_{ij}^{kt}) < (1-\alpha)\sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}}(v - b_j^{kt}). \tag{5.4}$$

From above, we are able to derive an algorithm to figure out v, a_j^{kt}, and b_j^{kt} to further differ swing requesters from strong requesters.

5.3.2 Algorithm construction

Both swing requesters' and weak requesters' utilities should increase by joining the coalition while strong requesters' utilities would decrease by joining the coalition. In other words only parameters v_i^k and b_{ij}^{kt} of swing requesters and weak requesters should satisfy eq. 5.4.

To ensure the accuracy of classification we use the greedy algorithm to classify. Each requester, except for the weak ones, has two possible classifications–swing or strong. With the greedy algorithm, to go through every possible scenario, the algorithm time complexity is going to be up to $O(2^{\bar{m}})$ where $\bar{m}$ is the number of all requesters other than the weak ones. Since the time complexity of the algorithm is exponential, we need to build an appropriate algorithm to simplify it.

We first find the maximum value of α by minimizing the denominator and maximizing the numerator at the same time. Due to Definition 5.3.1, it is very easy to distinguish weak requesters which means $\sum_{i \in M_s^{kt} \bigcup M_o^{kt}} v_i^k \geq \sum_{i \in M_o^{kt}} v_i^k$. Meanwhile, $p_{oj}^{kt} > \frac{a_{oj}^{kt}}{a_{ij}^{kt}} p_{ij}^{kt} - b_{ij}^{kt} + b_{oj}^{kt}$, the maximum value of p_{oj}^{kt} should be $p_{oj}^{kt} = \frac{\max\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}{\min\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}} + \max\limits_{i \in M^{k(t-1)} \cap M^{kt}} b_{ij}^{kt}$.

After getting the maximum α and the minimum $\sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}} = \frac{\min\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}{\max\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}$, we calculate $v_i^k - b_{ij}^{kt}$ for each $i \in M^{k(t-1)} \cap M^{kt}$. Then we will figure out $vb = \max\limits_{i \in M_o^{kt}} (v_i^k - b_{ij}^{kt})$ and rank the $v_i^k - b_{ij}^{kt}$ values of the remaining requesters from lowest to highest. Comparing vb and $v_i^k - b_{ij}^{kt}$ in turn, if $v_i^k - b_{ij}^{kt}$ is smaller than vb then this requester will be categorized into swing requesters and we will update vb to the $v_i^k - b_{ij}^{kt}$. When comparing to a $v_i^k - b_{ij}^{kt}$ greater than

vb, the comparison will end. At this point, the classification result is the final result.

Pseudo-code of coalition formation algorithm

We organize the process in the above section into Algorithm 5.3.

Table 5.3: Coalition formation algorithm at stage t

Input: requesters' data completion degree $w_i^{k(t-1)}$.

Output: classification results of requesters.

1. Call Algorithm 5.1.
2. For each requester $i \in M^{k(t-1)} \cup M^{kt}$ do:
3. If $w_i^{k(t-1)} < \varepsilon$ then:
4. Classify i as a weak requester and add it to the set M_o^{kt}.
5. End if.
6. End for.
7. For each participant $j \in N^{kt}$ do:
8. Calculate the maximum value $\alpha = \bar{\alpha}$ and the minimum $\sqrt{\dfrac{a_{oj}^{kt}}{a_j^{kt}}} = \dfrac{\min\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}{\max\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}$.
9. Initialize $vb = 0$.
10. For each requester $i \in M^{k(t-1)} \cup M^{kt}$ do:
11. Calculate $v_i^k - b_{ij}^{kt}$.
12. quad If $i \in M_o^{kt}$ then:
13. If $v_i^k - b_{ij}^{kt} > vb$ then:
14. Update $vb = v_i^k - b_{ij}^{kt}$.
15. End if.
16. End if.

17. End for.
18. For each requester $i \in M^{k(t-1)} \cup M^{kt} \setminus M_o^{kt}$ do:
19. Sort $v_i^k - b_{ij}^{kt}$ in ascending order.
20. If $v_i^k - b_{ij}^{kt} > vb$ then:
21. Classify i as a swing requester and add it to the set M_o^{kt}.
22. Update $vb = v_i^k - b_{ij}^{kt}$.
23. Update requester i as the discriminant requester.
24. End if.
25. End for.
26. End for.

Algorithm 5.3 categorizes requesters into weak and swing requesters based on their data completion degree and value comparison. It adapts dynamically, ensuring that weak requesters are separated, while swing requesters are identified based on their ranking.

The loop from line 2 to 6 iterates over all the remaining requesters from the previous stage to check whether they are weak requesters based on Definition 5.3.1. Line 10 to 17 calculates the value $v_i^k - b_{ij}^{kt}$ to distinguish swing requesters from strong requesters. Line 18 to 25 process requesters who are not in the weak requester set to identify the swing requesters. However, from line 8, the judgement condition between swing requesters and strong requesters $\sqrt{\frac{a_{oj}^{kt}}{a_j^{kt}}} = \frac{\min\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}{\max\limits_{i \in M^{k(t-1)} \cap M^{kt}} a_{ij}^{kt}}$ is not and exact equation which means some swing requesters may be misclassified as strong requesters. In Section 5.4, we will prove that although in Algorithm 5.3 we have reduced the criterion, the coalition will still be able to match some but not all participants.

Judging from Algorithm 5.3, we reduce the computing time complexity of classification algorithm from exponential to first order polynomial.

5.3.3 The stability of algorithm 5.3

After we introduce Algorithm 5.3 to categorize requesters into different classes, we can ally by weak requesters and swing requesters. As we have discussed in Introduction, the match between participants and requesters by Algorithm 5.2 is unstable due to the time variability of both requesters and participants, that is why we propose updated matching rules. Hence, we must first guarantee the stability of the updated matching rules which is the stability of the coalition to avoid the instability of the match led by time variability.

Definition 5.3.4. ***Coalition stability***
A coalition is stable if and only if the following conditions are met.

- *Coalition membership stability*

  · Internal stability: when strong requesters do not change their strategies, weak requesters and swing requesters will not increase their utilities from leaving the coalition.

  · External stability: when weak requesters and swing requesters do not change their strategies, strong requesters will not increase their utilities from joining the coalition.

- *Coalition form stability when weak requesters and swing requesters do not change their strategies, strong requesters will not increase their utilities by forming any kind of new coalitions.*

Now we have to prove that, with this costs allocation method in eq. 5.3, coalition stability is guaranteed.

Proposition 5.3.5. *In the coalition-driven Stackelberg MCS game* $Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t$*, the coalition has coalition membership stability.*

Proof. On the basis of Definition 5.3.4, if we could assure that weak requesters and swing requesters will gain more by joining the coalition while strong requesters' utilities would drop by joining the coalition, then we are able to promise coalition's membership stability.

According to the calculation in sec 5.3.1, as long as the parameters of requesters satisfy eq. 5.4, utilities of requesters will increase by joining the coalition. With regard to Definition 5.3.1, both weak requesters and swing requesters satisfy eq. 5.4 while strong requesters not. As a result, with costs allocation method in eq. 5.3, Definition 5.3.1, and Definition 5.3.4 the coalition membership stability is guaranteed. □

After demonstrating the membership stability, we move on to the coalition form stability, which means we need to make sure that requesters outside the coalition (strong requesters) will not form any kind of new coalitions.

Proposition 5.3.6. *The coalition has coalition form stability, that is to say no rational strong requesters will form any kind of coalitions.*

Proof. It is proved by contradiction. Let us assume that $M\bar{M}$ ($M\bar{M} \leq M_c^{kt}$) strong requesters will form a coalition, just prove that the utilities of strong requesters will be reduced then all rational strong requesters will choose not

to form any coalitions. If coalition is formed, the new coalition only needs to ensure that compared with the original coalition, it can bring higher benefits to participants. Refer to the proof of Proposition 1, the payment policy of the new coalition should be

$$p_{na}^{kt} > \sqrt{\frac{a_{nj}^{kt}}{a_j^{kt}}(\frac{v - b_j^{kt}}{2}) + b_{nj}^{kt}}, \forall j \in N^{kt},$$

where p_{na}^{kt} is the new coalition's payment policy and a_{nj}^{kt}, b_{nj}^{kt} denotes participants' parameters for the new coalition at stage t.

The cost allocated for each strong requester in the new coalition is $p_{nai}^{kt} = \frac{v_i^k}{\sum_{i \in M\bar{M}} v_i^k} p_{na}^{kt}$. Referring to the proof of Proposition 5.3.5, strong requesters' utilities will be damaged if any of them form any kind of new coalitions. Thereby, none rational strong requesters will form any kind of coalitions. The coalition form stability has been proved. □

Theorem 5.3.7. *The coalition formed by the centralized MCS platform in the coalition-driven Stackelberg MCS game* $Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t$ *is stable.*

Proof. Combining Proposition 5.3.5 and Proposition 5.3.6, obviously the coalition is stable. □

Because the stability of the coalition is guaranteed, then in each stage $t \in \{1, 2, ..., T\}$ requesters' strategies are stable, that is to say the match between requesters and participants are stable. Thus we can eliminate the instability of the system brought by irregular time variability.

5.3.4 Pseudo-code of updated matching algorithm

At any stage t, in accordance with the updated matching rules of the coalition, we could update the participants-centered matching algorithm into Algorithm 5.4.

Table 5.4: Updated participants-centered matching algorithm between requesters and participants at stage t

Input: new requesters M_n^{kt}.
Output: matching results for requesters and participants.
Determine the remaining requesters from stage $t-1$.
Call Algorithm 5.1 to estimate the parameters of participants.
Call Algorithm 5.3 to classify requesters.
Update the coalition payment policy p_{oj}^{kt} based on eq. 5.3.3.
Call Algorithm 5.2.
Repeat Steps 1 to 5 until no new requesters join and all requesters have completed data collection.

Hereto, our participants-centered matching algorithm is not only feasible but also stable even with irregular time variability.

5.4 Simulation results on the matching algorithm

In this section, we conduct simulations to validate the performance of our proposed two algorithms with irregular time variability, and to prove that the simulation results are in agreement with the theoretical results.

5.4.1 Simulation settings

All the simulations were implemented using Matlab and running on a laptop with 16 GB memory and Intel Core is i5-12500H CPU.

Parameter settings

Initially, there are a total 200 requesters and 200 participants to begin with, at each stage, 5 to 10 new requesters and participants will join the system, the maximum participation degree of participants is randomly generated from 50 to 250, the parameter a_{ij}^{k0} is randomly generated from 1 to 16, the parameter b_{ij}^{k0} is between 1 and 11, and the parameter c_j^{k0} is between 1 and 6. The data value of data type $k \in Q$ for requester i v_i^k is randomly generated from 100 to 150, data quality requirement α_i^k is a random number between 10 and 25.

Baseline

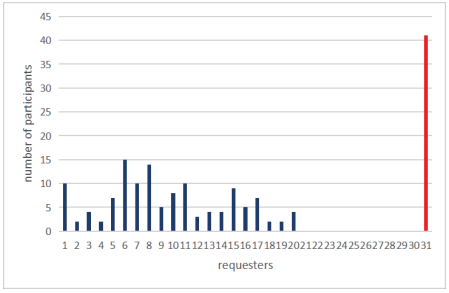
The baseline of the algorithm in this chapter is the requesters chosen rate. For any MCS requester, the most basic thing is to collect enough data. If the requester cannot collect enough data, then even if some high-quality data is collected at a lower cost, the task will fail because the amount of data is too small to complete the data analysis. Therefore, this chapter chooses the requesters chosen rate as the standard to measure whether the algorithm is effective.

5.4.2 Requesters matching results of Algorithm 5.4

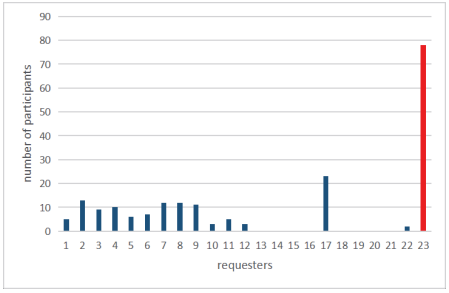
In this part, we conduct simulations about Algorithm 5.4, to prove that our proposed algorithm can be achieved in polynomial time and our theorems can be achieved in practice.

Because there exists more than one data type in the system, weak requesters may form a coalition of one data type while becoming strong requesters and competing against each other of another data type. If of this data type, the coalition formed by weak requesters and swing requesters forms a monopoly which may be considered a vicious competition by strong requesters, so of other data type requesters that failed to collect data of the original data type could retaliate against the coalition's monopoly. In economics, this phenomenon is known as "mutual forbearance", that is in multi-market competition, competitive contact extending across multiple markets may lead to lower rivalry – because the rivals realize that their interdependence increases the risk of broad retaliation (Yu and Cannella Jr 2013). Therefore, in this section, we will prove that our coalition will not form a monopoly.

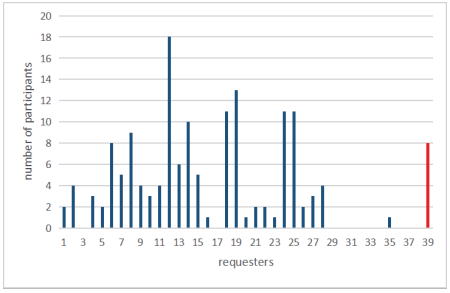
The X-axis represents requesters, the last column corresponds to the coalition. The X-axis in front of the X-axis coordinate corresponding to the coalition represents strong requesters and new requesters. Since this simulation emphasizes that the coalition must match the user, if the other requesters fail to match the participants in this round, according to the division of requesters, the requester will join the coalition in the next stage. Therefore, whether members other than the coalition can match participants is not the focus of this simulation. For requesters other than the coalition, for the sake of clarity of the figure, this chapter does not save their numbers but renum-



(a) 1 iteration



(b) 200 iterations



(c) 500 iterations

Figure 5.3: Requesters' matching results with coalition

bers them after recording data. The length of the column in the histogram represents the number of participants matched to this requester. Vividly the coalition formed by weak requesters and swing requesters can collect data and there will be no monopoly which equals Theorem 5.3.3.

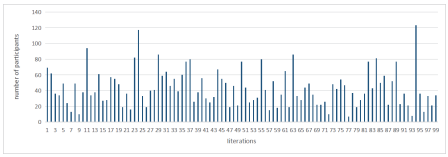
Fig. 5.3(b) and 5.3(c) are the matching results of requesters in 200 and 500 iterations. It is safe to say that the coalition by weak requesters and swing requesters can always collect data and there will be no monopoly which equals to Proposition 1 no matter how many iterations.

Judging from Fig. 5.3, the length of red column is always shorter than 200,

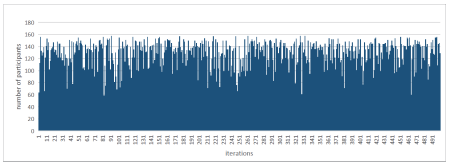
which means that the coalition formed by weak requesters and swing requesters from Algorithm 5.3 do not match all available participants. There are always some strong requesters and new requesters are able to gather data.

Remark 1: The coalition formed by weak requesters and swing requesters by Algorithm 5.3 would not form a monopoly which is in line with Theorem 5.3.3.

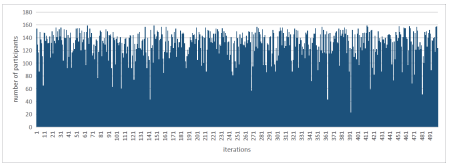
5.4.3 Matching results of the coalition



(a) 1 iteration



(b) 200 iterations



(c) 500 iterations

Figure 5.4: Matching results of the coalition

As we can see from Fig. 5.4(a) and 5.4(b) after several games, the coalition will always be able to match some participants which is in line with Theorem 5.3.3 even with the conditional reduction of Algorithm 5.3 which indicates that Algorithm 5.3 not only reduces the algorithm time but also ensures that the coalition must be selected by participants. Taking Definition 5.3.1 into consideration, it is clear that any requester after at most two games will be able to match some participants.

For the algorithm to be polynomial-time solvable, Algorithm 5.3 does not directly use the greedy algorithm but instead builds an optimization algorithm that yields an approximate reclassification of requesters, according to lines 2 to 6 of the Pseudo-code about Algorithm 5.3, weak requesters can be classified into the coalition. In other words, some swing requesters would be misjudged as strong requesters in Algorithm 5.3. However, based on Definition 5.3.1, swing requesters have already been matched to participants in the previous stage. Moreover, even if swing requesters failed to match participants due to algorithmic neglect, in the next stage, based on Algorithm 5.3, they would be regrouped into weak requesters and could match participants. Therefore, even if Algorithm 5.3 misjudged some swing requesters as strong requesters, the misjudged requesters can still match participants after at most two games.

To ensure polynomial-time solvability, Algorithm 5.3 does not rely purely on a traditional greedy approach but instead incorporates an optimization-based mechanism that reclassifies requesters approximately. According to lines 2 to 6 of the Algorithm 5.3 pseudo-code, weak requesters are effectively classified into the coalition. However, due to the approximation, some swing requesters may be initially misclassified as strong requesters. Nevertheless, based on Definition 5.3.1, these swing requesters would have already been matched to participants in earlier stages. Even if some swing requesters fail to match due to algorithmic misclassification, they will be reassigned as weak requesters in subsequent iterations and will eventually be able to form matches.

Fig. 5.4(c) is the matching results of requesters and participants after 500 iterations and to rule out the particularity brought about by the parameters, in Fig. 5.4(c), v_i^k is randomly generated from 400 to 500, the maximum par-

Table 5.5: Requesters chosen rate comparison

algorithm	M/N / number of games	$M = 200, N = 200$	$M = 200, N = 300$	$M = 200, N = 500$
greedyBTAMC (Amiri 2024)	1	0.26	0.30	0.37
	2	0.24	0.34	0.38
CA-MAB-SFS (Simon et al. 2024)]	1	0.34	0.45	0.62
	2	0.32	0.46	0.60
simpleTSGA (Abdeddine, Iraqi, and Mekouar 2025)	1	0.66	0.85	0.94
	2	0.64	0.83	0.93
WFIV (Xu, H. Sun, and Han 2024)	1	0.45	0.55	0.48
	2	0.51	0.56	0.50
ODRA (Xu, H. Sun, and Han 2024)	1	0.83	0.87	0.92
	2	0.84	0.90	0.91

ticipation degree of participants is randomly generated from 150 to 300, the parameter a_{ij}^{kt} is randomly generated from 10 to 26, the parameter b_{ij}^{kt} is between 10 and 21, and the parameter c_j^{kt} is between 10 and 16. In general, even with the conditional reduction of Algorithm 5.3, the coalition can always be matched with some but not all participants regardless of parameters and iterations.

Remark 2: The coalition formed by weak requesters and swing requesters by Algorithm 5.3 could definitely match some but not all participants despite of parameters which is in line with Theorem 5.3.3.

5.4.4 Requesters chosen rate comparison

In this section, we will compare the requesters' chosen rate of the several algorithms in (Amiri 2024; Simon et al. 2024; Abdeddine, Iraqi, and Mekouar 2025; Xu, H. Sun, and Han 2024).

To eliminate coincidence, we run each algorithm 10 times and take the average requesters chosen rate as shown in the table. The parameter settings of all algorithms in the table are the same as those in Section 5.4.1. If some parameters do not appear in this chapter, the design of that parameter is based

on the original paper.

All mentioned algorithms in Table 5.5, with Algorithm 5.4 aim to enhance MCS task allocation under constraints. While different algorithms focused on different constrains, budget constrains, decentralized decision-making, or ireregular time variability.

Compared to state-of-the-art algorithms such as (Amiri 2024; Simon et al. 2024; Abdeddine, Iraqi, and Mekouar 2025; Xu, H. Sun, and Han 2024), which employ dynamic pricing models and game-theoretic approaches, Algorithm 5.4 offers a more computationally efficient and structurally stable mechanism. Judging from Table 5.5, the requesters' chosen rate of our algorithm is higher than the other algorithms despite of parameter sets which indicates that, our algorithm outperformed the other algorithm in requesters chosen rate. While some existing models require extensive iterations and complex optimizations, our approach ensures convergence within at most two iterations, significantly reducing computational overhead while maintaining high matching success rates. Furthermore, unlike traditional greedy algorithms that may fail to reallocate misclassified requesters efficiently, Algorithm 5.3 dynamically adjusts classifications, ensuring long-term participation stability. These advantages make our method highly relevant to modern mobile crowdsensing applications.

5.4.5 Discussion and implications

In this section, theoretical contributions are presented, followed by practical implications of the results for future task allocation in MCS system developments.

Theoretical implications

This research extends the application and understanding of Mobile Crowdsensing (MCS) by addressing the challenges associated with the irregular time variability of participants.

Traditional MCS task allocation is mainly based on requester-centered matching methods, which either assume complete information about participants or assume that the model can predict available participants. However, this assumption is unrealistic in the real world because participants exhibit irregular time variability that can neither be known in advance nor predicted with 100% accuracy. By introducing a participant-centered matching algorithm based on the Stabckelberg game, this study transfers the decision-making power to the participants. Algorithm 5.4 is consistent with the game theory approach, especially the Stackelberg model, which guarantees the infeasibility of matching and utility.

Our research addresses the instability of traditional matching mechanisms caused by irregular time variability of participants. By combining an updated matching algorithm with a greedy-based requester classification algorithm, this research provides a systematic approach to dynamically adjust the matching strategy. This provides a solution for real-time applications.

More than that, the theoretical findings are reinforced through simulations, showing the feasibility and stability of the proposed model. The validation of the participant-centered matching algorithm through simulation results ensures that the theoretical advancements can be translated into practical implementations, contributing to both academic research and real-world deployment.

The findings of this study have significant practical implications for real-world applications, particularly with irregular time variability.

Traditional MCS systems often suffer from unpredictable participant availability (e.g., offline systems) and inaccurate predictions or unrealistic assumptions (e.g., online systems). Our algorithm uses a dynamic matching mechanism that is updated based on real-time participant behavior, ensuring that requesters can be matched with participants after playing at most two games, even with irregular time variability patterns. This adaptability is critical for real-time applications such as disaster response, which require immediate and flexible task allocation.

The approach in this study can also be applied to other industries besides MCS applications. For example, in the sharing economy, similar matching challenges exist. The proposed framework can serve as a basis for improving the efficiency of various allocation models.

By addressing the challenges of incomplete information and instability in task allocation, this study provides a practical framework for enhancing the efficiency, adaptability, and scalability of MCS systems. The participant-centered approach, combined with real-time adaptation, ensures that MCS can be successfully deployed in various real-world scenarios, improving both requester and participant experiences while maximizing the system's overall effectiveness.

5.5 Conclusion

For the irregular time variability of participants, a participants-centered match based on coalition formation is proposed to deal with infeasibility and instability. By theorems and simulations, we proved that all requesters will match some participants after at most two games and requesters' costs have been greatly reduced through forming a coalition.

Compared with a requesters-centered match which by predicting participants moving trails inevitably has errors and only applies for regular time variability, a participants-centered match with participants having dominant power could improve the accuracy of the system and broaden the system's scope of application. With the participants-centered match, some requesters may feel reluctant because of their lack of control over which participants are matched to guarantee their utilities. To avoid requesters' dissatisfaction, a Stackelberg MCS model where requesters as leaders first decide on matching strategies to maximize their utilities is proposed. Therefore, our algorithm can satisfy requesters, improve accuracy, and have a wider range of applications.

In this chapter, the only concern is the requesters chosen rate, which cannot represent the success of data collection. On the one side, matching with participants does not mean that requesters will be able to collect enough data. On the other side, limited by requesters' budgets and clouds, the amount of data is not always the more the better, for it may cause the waste of resources, and the increase of data processing. Therefore, in the future, we would like to focus on the task completion rate of the time-sensitive MCS system to further advance the data utilization rate.

6

Time-sensitive MCS systems

This chapter is based on: Zhilin Xu, Hao Sun, and Weibin Han. "Boosting task completion rate for time-sensitive MCS system". In: *Computer Networks* 251 (2024), p. 110636.

In Mobile Crowdsensing system, many sensing tasks are time-sensitive, hence, time validity is essential cause data outside the time frame is useless which will not only increase requesters' costs but also damage requesters' efficiency and utilities. Therefore, it is crucial to design an incentive mechanism to increase system's task completion rate within time limitations. A tension arises between requesters and the MCS system due to the trad-off between requester's goal to maximize its own utilities and the MCS system's purpose to increase the whole system's task completion rate. To solve the dilemma, we come up with a dynamic coopetition (competition and cooperation) incentive mechanism where in every stage there are competition to assure requesters' utilities and cooperation to reallocate data to advance the system's task completion rate. During competition, a matching algorithm with budget constraints is presented to derive the optimal matching strategies for requesters and participants without exceeding requesters' budgets while maximizing utilities. After competing, the centralized MCS platform would reallocate requesters' excessive data to those who failed to get enough data on time through a data reallocation algorithm. By simulations, we compare the performance of the coopetition incentive mechanism with different orders and parameters. Regardless of parameters or orders, our algorithm can promise a minimum task completion rate of 94% compared with less than 32% task completion rate with only competition. Among all three orders, emergency degree has the best performance with task completion rate up to 99%.

6.1 Introduction

The rapid development of the 5th generation mobile networks (5G) as well as the popularity of smart mobile devices, allow Mobile Crowdsensing (MCS) system to grow rapidly. MCS enables and inspires a vast number of people to contribute data. Thus it become a convenient method for many Internet of Things applications, such as smart cities (Z. Peng et al. 2019; Wang et al. 2020), environmental monitoring (Tseng et al. 2020; Dutta et al. 2009; Forestiero and Papuzzo 2022), smart transportation (Dutta et al. 2009), and intelligent medicine (Mankodiya et al. 2022; Trilles et al. 2020).

The above MCS applications are time-sensitive, which means requesters need to collect enough data within a pre-specified time window. Ignoring such time limitations can lead to collection of invalid data (Ipaye et al. 2022; Kraft et al. 2022; Yucel and Bulut 2022; Cicek and Kantarci 2023). This increases the cost of requesters and may lead to inefficient outcomes.

With the time limitations, the maximization of task completion rate becomes more complicated. Therefore, in this chapter we propose a multi-requesters dynamic coopetition (competition and cooperation) incentive mechanism. Similar to our approach in Chapter 4, we discretize the data collection process into multiple stages. As in Chapter 3, in each stage, requesters will request different data types based on their data requests. Because each data type represents one unique type of data, all data types are independent of each other. Therefore, we only consider one data type as an example.

During the competition phase, we focus on maximizing requesters' utilities. The competition model in Chapter 6 is the same model as the model in Chapter 4 in this first phase. In this first phase, the platform does not participate, it just passes data through.

Although the model is the same, our solution method is slightly different. In Chapter 6, we consider requesters' budgets. In the equilibrium of the Stackelberg model, requesters select participants in a particular order. Requesters can be ordered based on budgets, payments or emergency degree. Requesters with more budgets, higher payments or less time left will select participants first. If a requester has enough budget to recruit a participant, the match is made and the remaining budget is updated. Otherwise this requester will move on to the next participant.

From the simulation results of the matching algorithm, we observe that some requesters may obtain too much data, while some requesters may not get enough data. Without proper parameter settings, the task completion rate is only 8%, and the percentage of requesters who collected too much data can reach 100%.

In the cooperation phase, the centralized MCS platform classifies the requesters into different categories based on the matching results in the competition phase. Requesters who get more data than needed are classified as strong requesters with redundant data. Requesters who fail to get enough data in the competition phase are classified as weak requesters with insufficient data. Middle requesters are those who get exactly the amount of data they required. To increase the task completion rate and reduce spurious data collection, the centralized MCS platform reallocates surplus data from strong requesters to weak requesters. We formulate the data reallocation process as a cooperation model. The characteristic function is the task completion margin, which is the difference between the amount of data actually obtained by the requester and the requested amount.

A data reallocation algorithm among requesters is proposed to find an optimal solution to the cooperation game. The algorithm maximizes the system's

task completion rate.

Using simulations, we compare the task completion rate of the time-sensitive MCS system under different requester orders (budgets, emergency degree, and payments) and different parameter sets. We find that the task completion rate was at least 94%. If the requesters are ordered in the same order in both the competition phase and the cooperation phase, the task completion rate is even higher. Moreover, when all requesters are sorted by emergency degree (prioritized by the remaining time), the task completion rate can reach 99% through our dynamic coopetition incentive mechanism.

The main contributions of this chapter are summarized below:

- In this chapter, we take the time-sensitive nature of an MCS system into account, and assign a time limitation to each task. We propose a dynamic coopetition (competition and cooperation) incentive mechanism to reduce spurious data collection and improve task completion rate for the time-sensitive MCS system.

- We run simulations and find that, in our dynamic coopetition incentive mechanism the percentage of requesters with surplus data can be reduced up to 68.75% and the task completion rate can be improved from 32% to at least 94%.

The remainder of this chapter is organized as follows. The competition phase is described in Section 6.2. Section 6.3 describes the cooperation phase. The simulation results are presented in Section 6.4. Finally, Section 6.5 is the conclusion of this chapter.

6.2 Competition for time-sensitive MCS system stability

The precondition of the time-sensitive MCS system is the system stability which refers to the happiness of requesters. Due to the rationality of requesters, without the guarantee of their utilities maximization, some selfish requesters may refuse to follow prescriptive strategies which would eventually lead to the instability of the time-sensitive MCS system and damage the task completion rate.

In this section, for the dilemma between requesters and the time-sensitive MCS system, we first guarantee requesters' utilities maximization which is the only way to get requesters obey the MCS system's strategies. Requesters are more likely to buy higher-quality data at a lower price, while participants are more likely to sell lower-quality data at a higher price. In that case, we model the relationship between requesters and participants as a multi-requesters Stackelberg MCS competition model where requesters as leaders first announce strategies (payment policies) to maximize utilities, then participants as followers choose their strategies (the amount of workload) based on requesters' payment policies.

6.2.1 Multi-requesters Stackelberg MCS competition model

First of all, we model the relationship between requesters and participants during competition as a Stackelberg game with requesters as leaders and participants as followers. For definiteness , we use $M = \{1, 2, ..., m\}$ to denote the set of requesters and $N = \{1, 2, ..., n\}$ to represent the set of mobile device users that are willing to participate. To begin with, requesters as leaders first announce tasks $S = \{S_1, S_2, ..., S_i, ..., S_m\}$, where $S_i = \{S_{i1}, S_{i2}, ..., S_{il}\}$ is the set of tasks issued by requester i. Due to the heterogeneity of the tasks issued

by requesters, we use the concept of "data type" which uniquely defines a jointly owned data of a class of tasks as in Chapter 3. Requesters request different data types based on the data demanded by requesters. Therefore, each requester is particularly defined by a set of data types. Considering the inequable data of the dissimilar data types, all data types are independent of each other. Due to the independence of different data types, in the remainder of this chapter we only consider data type $k \in Q$ as a typical example.

For data type $k \in Q$, along with tasks, requesters will public payment policies which are their matching strategies p_i^{kt} and their data quality requirements α_i^k. p_i^{kt} is the unit payment requesters pay to participants who collect data with data type $k \in Q$ for them at stage t and $\alpha_i^k (\alpha_i^k > 0)$ means requester i's requests for the quality of the data participants provide of data type $k \in Q$. It is worth noting that each task is preset with the acceptable data quality requirements, so as to avoid its failure caused by poor-quality sensing data. Afterwards, participants as followers will decide which requester they upload data to with what kind of participation degree based on requesters' strategies. Participants as followers have two related strategies $\bar{I}_{ij}^{kt}$ and X_{ij}^{kt}, where $\bar{I}_{ij}^{kt}$ is indicative function represents whether participant j will upload data to requester i or not with data type $k \in Q$ at stage t, and $X_{ij}^{kt}, (X_{ij}^{kt} \geq 0)$ is participant j's degree of participation ($X_{ij}^{kt} = 0$ equals participant j not participate). Moreover, participant j can only upload data to one requester with one data type (if $\bar{I}_{ij}^{kt} = 1$, than $\bar{I}_{sj}^{kt} = 0$ for any $s \in M^{kt}, s \neq i$).

Due to the rationality of both requesters and participants, they will only decide strategies with regard to their own best interests. When it comes to participants, obviously, the payment should be the more the merrier. Yet for requesters, higher payments means higher costs. To figure out what kind of strategies rational requesters and rational participants will choose, their

incomes and costs are calculated as in Chapter 4.

Apparently requesters' incomes will increase as participants degrees of participation go up. Besides, the costs of requester i is its payment to participants j which is also irrelevant with stage. Therefore, the utility to requester i of data type $k \in Q$ at stage t can be defined as

$$u_{ri}^{kt}(p_i^{kt}, X_{ij}^{kt}) = \sum_{j \in N} \bar{I}_{ij}^{kt} * (V_i^k * X_{ij}^{kt} - p_i^{kt} * X_{ij}^{kt}),$$

where V_i^k is the value of data with data type $k \in Q$ to requester i, α_i^k is data quality requirements and $\bar{I}_{ij}^{kt}$ is the indicative function about whether requester i would match participant j at stage t or not.

However, when it comes to requesters, more engagement of participants isn't always better because participants may spend time and energy gathering large amount of seemly related but really out of context data. Participants' actual costs will increase with the participation degree and requesters' data demands which is added in participants' costs to guarantee the data collected by the participant must meet the data requirements of the requester. Furthermore, the higher the participation, the more the costs and the gain of payments will be offset by the increase in costs gradually. Hence, the derivative function of participants' utilities are convex as in Chapter 4.

$$u_{pj}^{kt}(p_i^{kt}, X_{ij}^{kt}) = \sum_{i \in M} \bar{I}_{ij}^{kt} * (p_i^{kt} * X_{ij}^{kt} - c_j^k * {X_{ij}^{kt}}^3 * \alpha_i^k),$$

in which c_j^k is participant j's costs for collecting data of data type $k \in Q$.

Definition of the Stackelberg MCS competition model

Due to the time-sensitive nature of the MCS system, we discretized the dynamic tasks completing progress into T stages. $T = \max_{i \in M^k} T_i$ where M^k is the set of requesters for data type $k \in Q$ and T_i is the number of games that requester i can join. In the Stackelberg MCS competition model at stage t, every requester must first decide whether the t-round game exceed their time limitation.

Definition 6.2.1. *The model of a multi-requesters Stackelberg MCS competition game at stage t is denoted by*

$$Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t,$$

where:

- *M^{kt} denotes the set of requesters requesting data type $k \in Q^t$ at Stage t, and N^{kt} denotes the set of participants available to participate in sensing tasks at stage t, where $M^{kt} \subseteq \mathcal{M}$ and $N^{kt} \subseteq \mathcal{N}$;*
- *$Y_{ij}^{kt} = \{y_{ij}^{kt}\}$ denotes the set of payment strategies offered by requester $i \in M^{kt}$ to participant $j \in N^{kt}$, where $y_{ij}^{kt} \in \mathbb{R}_+$;*
- *$X_{ij}^{kt} = \{x_{ij}^{kt}\}$ denotes the set of participation strategies of participant j for requester i, where $x_{ij}^{kt} \in [0, \bar{\eta}_j^{kt}]$;*
- *$\bar{U}_r^{kt} : Y^{kt} \to \mathbb{R}$ and $\bar{U}_p^{kt} : X^{kt} \to \mathbb{R}$ denote the utility functions of requester i and participant j at Stage t, respectively.*

Notice, each requester has a budget constrain D_i^{kt} at stage t. The actual amount of money $p_i^{kt} X_{ij}^{kt}$ requester i pays for participant j cannot exceed its budget. In this

chapter, we assume that each requester's budget divide equally by the number of data types it requested and the number of games it can participate.

Definition 6.2.2. *In the Stackelberg MCS competition game* $Z_k^t = \left\langle M^{kt}, N^{kt}, Y_{ij}^{kt}, X_{ij}^{kt}, \bar{U}_r^{kt}, \bar{U}_p^{kt} \right\rangle, k \in Q^t$*, the set* $T(p_i^{kt}) \subset [0, +\infty)$ *defined for each* p_i^{kt} *by*

$$\begin{aligned} T(p_i^{kt}) = \{X_{ij}^{kt} \in [0, +\infty) : u_{pj}^{kt}(X_{ij}^{kt}) \geq u_{pj}^{kt}(X'^{kt}_{ij}), \\ \forall X'^{kt}_{ij} \in [0, +\infty),\ i \in M^{kt}\}, \end{aligned} \tag{6.1}$$

is the optimal response (rational reaction) set of participant j *to the strategy* p_i^{kt} *of requester* i*. A strategy* p_i^{kt*} *is called a Stackelberg MCS competition equilibrium for the leader (requester* i*) , if*

$$\min_{X_{ij}^{kt} \in T(p_i^{kt*})} u_{ri}^{kt} = \max_{p_i^{kt}} \min_{X_{ij}^{kt} \in T(p_i^{kt})} u_{ri}^{kt}, \tag{6.2}$$

which means regardless of how participant chooses the strategy within the optimal response set, will the requester chooses the strategy that maximizes its minimum utility.

Judging from Theorem 4.2.3 and Theorem 4.2.4 in Chapter 4, the only Stackelberg MCS competition equilibrium of participant j and requester i is

$$X_{ij}^{kt*} = \sqrt{p_i^{kt}/3c_j^k\alpha_i^k}, \tag{6.3}$$

$$p_i^{kt*} = \frac{V_i^k}{3}. \tag{6.4}$$

6.2.2 Matching algorithm between requesters and participants under budget constraints

With regard to time-sensitive tasks, requesters would like to match as many participants as possible to fulfill their desires to complete tasks in time, but limited by their budgets, they could only match some participants. Restricted by their energy, one participant could only gather data for one requester. To avoid different requesters matching the same participant, we would sort requesters based on three different orders in simulation phase. The first one is the budget which is sorted from high to low, the more the budget, the more likely the requester is to match enough participants. The second order is emergency degree which is ordered from small to big, the more urgent the requester's remaining time, the more requesters can be preferred. And the last order is the unit amount of money requesters pay for participants which is sorted the same as budget, the higher the payment, the higher the preference of participants.

After getting both requesters' and participants' optimal matching strategies (payment policies in eq. 6.4 and workload in eq. 6.3, requesters would select participants in order. If and only if the requester has enough budget to recruit this participant, it will match and update its remaining budget. The purpose of this chapter is task completion rate, hence, during the matching preference, participants with lower costs would be matched preferentially based on eq. 6.3, cause in the same case, the lower the participant cost, the higher the amount of data that can be collected. The pseudo-code of the matching algorithm between requesters and participants under budget constraints with different orders is summarized in Algorithm 6.1.

Table 6.1: Matching algorithm between requesters and participants at stage t

Input: requester set $M = \{1, 2, \ldots, m\}$, participant set $N = \{1, 2, \ldots, n\}$, time limits T_i, and budget constraints D_i.
Output: payment policy p_i^{kt}, degree of engagement X_{ij}^{kt}, and matching results Y for requesters and participants.

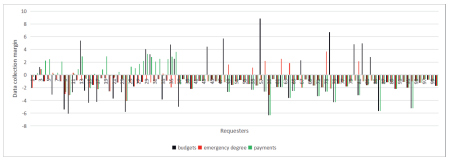
1. Requesters join data types $Q = \{1, 2, \ldots, q\}$ based on published tasks and obtain the set M^k. Owing to the independence of different data types, only data type $k \in Q$ is considered as a representative example in the remainder of this pseudo-code.
2. For each requester $i \in M^k$ do:
3. Calculate the requester's optimal payment policy p_i^{kt} according to eq. 6.2.
4. End for.
5. Sort requesters in ascending order of budgets, emergency degree, and payments.
6. With regard to eq. 6.1, update the degree of engagement as $X_{ij}^{kt} = \sqrt{\frac{p_i^{kt}}{3c_j^k \alpha_i^k}}$.
7. For each requester $i \in M^k$ do:
8. If $t \leq T_i$ then:
9. Add requester i to the set M^{kt}.
10. End if.
11. End for.
12. Initialize the matching result matrix $Y = \mathrm{zeros}(M^{kt}, N)$.
13. For each requester $i \in M^{kt}$ do:
14. Set $d = D_i$.
15. For each participant $j \in N$ do:
16. Determine whether participant j has already been matched:
17. $yy = \mathrm{ismember}(1, Y(1{:}M^{kt}, j{:}j))$.

18. If participant j is not matched to any requester ($yy = 0$) then:
19. If $d > p_i^{kt} X_{ij}^{kt}$ then:
20. Set $Y(i{:}i, j{:}j) = 1$.
21. Update $d = d - p_i^{kt} X_{ij}^{kt}$.
22. End if.
23. End if.
24. End for.
25. End for.

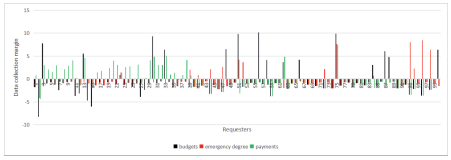
In the pseudo-code of Algorithm 1, the data requirement is corresponding to line 6, line 7 to line 10 are time constraints and line 14 is budget constraints.



(a) $N = 150, D_i \sim unifrnd(150, 200)$



(b) $N = 150, D_i \sim unifrnd(150, 200)$



(c) $N = 200, D_i \sim unifrnd(200, 250)$

Figure 6.1: Data collecting results of requesters by Algorithm 6.1

Fig. 6.1 is about the differentials between the amount of data collected by requesters with three different orders and the schedule amount of data requesters should gathered at stage $t = 1$ with only competition. There are

total 100 requesters in Fig. 6.1, the task value of requester i (V_i^k) is randomly generated from 100 to 150, data quality requirement α_i^k is a random number between 10 and 25, the unit costs of each participant is between 50 and 100, and T_i between 10 and 110 is the number of games that requester i can join. The black, red and green column represent the differentials between the actual collected data and the demand data when ordered in budgets, emergency degree and payments. If the column is above the zero axis, then this requester has collected too much data than needed and is one of the requesters who wasted data as shown in Table 6.2. If the column is bellow the zero axis, then this requester failed to get enough data. Requesters without any column means they got just enough data without any waste, along with requesters with column above the zero axis are those who completed data collection at stage t and is shown in Table 6.3. As we can see from Fig. 6.1, despite from the change of parameters, there are always requesters who collect too much data and those who fail to collect sufficient data. To avoid the waste of data and improve the whole MCS system's task completion rate, the centralized MCS platform would reallocate data to different requesters in cooperation phase. In short words, from Fig. 6.1, when some requesters collected too much data, there are also requesters failed to collect enough. Therefore, for the sake of maximizing task completion rate of the whole time-sensitive MCS system, requesters with column above the zero axis would share their extra data to requesters with column below the zero axis in cooperation.

Table 6.2: Data waste rate at stage 1 of Algorithm 6.1

(parameters,orders)	budgets	emergency degree	payments
$N = 100$,			
$D_i \sim unifrnd(100, 150)$	94.11%	42.10%	72.72%
$N = 150$,			
$D_i \sim unifrnd(150, 200)$	100%	42.30%	67.56%
$N = 200$,			
$D_i \sim unifrnd(200, 250)$	100%	69.56%	78.04%

Table 6.2 is about the percentage of requesters with too much data among requesters who completed data collection. From Table 6.2 about 42.10% to 100% requesters have collected too much data which is a huge waste of resources.

Table 6.3: Data completion rate at stage 1 of Algorithm 6.1

(parameters,orders)	budgets	emergency degree	payments
$N = 100$,			
$D_i \sim unifrnd(100, 150)$	16%	8%	24%
$N = 150$,			
$D_i \sim unifrnd(150, 200)$	14%	11%	25%
$N = 200$,			
$D_i \sim unifrnd(200, 250)$	16%	16%	32%

Table 6.3 is about the percentage of requesters who finished data collection among all requesters. Judging from Table 6.3, the data completion rate of

Algorithm 6.1 is about 8% to 32% which means only a small amount of requesters got enough data as scheduled. To avoid the waste of data and improve the MCS system's task completion rate, in the following section the centralized MCS platform would reallocate data to different requesters.

6.3 Cooperation to improve time-sensitive MCS system task completion rate

After satisfying one of the conflicting party (utilities maximization of requesters), we move on to satisfy the other conflicting party (maximum task completion rate of the time-sensitive MCS system). Obviously from Table 6.2, the data waste rate could be up to 100% which is a huge waste of resources and a great damage on the task completion rate. Besides, from Table 6.3, without proper parameters setting, the task merely has a 8% completion rate in stage t cause some requesters could have gathered data if strong requesters did not forcibly occupy more data than needed. To avoid waste of resources, and improve system's task completion rate, we build a dynamic cooperative MCS game where cooperative means the time-sensitive MCS system would reallocate exceed data among requesters.

6.3.1 Dynamic cooperative MCS game

In the dynamic cooperative MCS game, the centralized MCS platform would reallocate data by sharing strong requesters' redundant data to those who failed their data collection on schedule as shown in Fig. 6.2. Because only requesters who failed to gather enough data in time during competition of

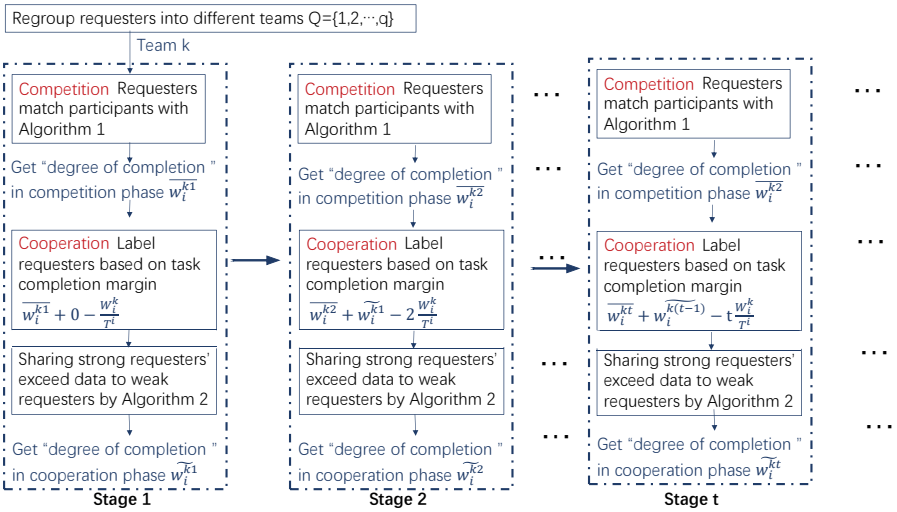


Figure 6.2: Illustration of the dynamic coopetition model. Firstly, requesters collect data in the competition phase with the goal of maximizing utility. Then, requesters share data in the cooperation phase with remaining budgets from competition to maximize task completion rate. Afterwards, requesters join the competition phase at next stage and repeat until out of time.

coopetition need data in cooperation, those requesters may be willing to settle for less valid data in cooperation to complete data collection on time. Therefore, in cooperation, requesters will neglect α_i^k (data quality requirement) due to time constrains.

Reclassification of requesters

In the dynamic cooperative MCS game, there are three kinds of players– strong requesters who has redundant data to share (the corresponding column in Fig. 6.1 is above the zero axis), middle requesters whose data collection is just as planned and weak requesters who failed to gather enough data

on time (the corresponding column in Fig. 6.1 is beyond the zero axis) . Before we introduce the dynamic cooperative MCS game, we must first determine each requester's kind.

To better classify requesters, we need to compare requesters actual data completion degree with their required amount of data. Hence, let us introduce a new parameter "degree of completion" in competition phase and cooperation phase separately which uniquely defines requesters' current data collection status. During competition phase, requesters' data completion is determined by their matched participants and the amount of data that can be supplied. $\bar{w}_i^{kt} = \sum_{j \in N} \bar{I}_{ij}^{kt} X_{ij}^{kt}$ is requester i's "degree of completion" in competition at stage t. Besides requesters' "degree of completion" in cooperation phase $\tilde{w}_i^{kt}$ is decided by the amount of data they shared or received during cooperation. After that, we need to compare requesters' "degree of completion" with its total amount of requirements W_i^k which is given in advance by each requester. If requester i's actual "degree of completion" is higher than its total amount of requirements, then requester i had collected too much data and would be classified as strong requesters. Otherwise, requester i failed to collect data as schedule and would be regrouped as weak requesters. The difference value of requesters' "degree of completion" and its total amount of requirements W_i^k is known as task completion margin ${w^*}_i^{kt}$.

$$ {w^*}_i^{kt} = \sum_{t'=1}^{t} (\bar{w}_i^{kt'}) + \sum_{t'=1}^{t-1} (\tilde{w}_i^{kt'}) - (\frac{W_i^k}{T_i} * t). \tag{6.5} $$

Therefore, the strong requesters, middle requesters and weak requesters can be defined as in Definition 6.3.1.

Definition 6.3.1. *For every requester i, if its task completion margin w^{*kt}_i is positive, then it is classified as* **Strong Requesters**, *if its task completion margin w^{*kt}_i is negative, then it is classified as* **Weak Requesters**, *if its task completion margin w^{*kt}_i is equal to 0, then it is classified as* **Middle Requesters**.

Formation of the dynamic cooperative MCS game

The purpose of the dynamic cooperative MCS game is to figure out the proper amount of data that is going to share between requesters. Therefore, the goal of the time-sensitive MCS system differs significantly from the traditional cooperative game. Instead of benefits, the characteristic function of the dynamic cooperative MCS game $v^{kt}(S)$ represents the greatest amount of data that member in the coalition S can get through adequate consultation.

$$v^{kt}(S) = \sum_{i \in S \bigcap M^{kt}_a} w^{*kt}_i, \ \forall S \subset M^{kt}, \tag{6.6}$$

where $M^{kt}_a = \{i \in M^{kt} : \ w^{*kt}_i > 0\}$ is the set of strong requesters of data type $k \in Q$ at stage t.

$\Gamma = (M^{kt}, v^{kt})$ is said to be an M^{kt}- player dynamic cooperative MCS game cause judging from eq.6.6, the characteristic function at stage t is related with the "degree of completion" at stage $t-1$, thus the cooperative MCS game is dynamic.

Definition 6.3.2. *The dynamic cooperative MCS game at stage t is defined as $\Gamma = (M^{kt}, v^{kt})$ where*

- $M^{kt} = \{i \in M^k : \ T_i \geq t\} = M^{kt}_a \bigcup M^{kt}_b \bigcup M^{kt}_c$ *is the set of requesters as the same in Definition 1 where M^{kt}_a is the set of strong requesters, M^{kt}_b the set*

of middle requesters, M_c^{kt} the set of weak requesters and T_i is the number of games that requester i can join.

- *The characteristic function in eq. 6.6 is a real-valued set-function on the set $2^{M^{kt}}$ of all subsets of M^{kt} ($v^{kt}: 2^{M^{kt}} \to R$) such that $v(\Phi) = 0$. Because the characteristics function in 6.6, the characteristic function at stage t is related with the "degree of completion" at stage $t-1$.*

To ensure that all requesters are sufficiently willing to share or receive data, in Proposition 6.3.3 we prove some notation of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$.

Proposition 6.3.3. *With the characteristic function v^{kt} in eq. 6.6, the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ is monotonic, additive and superadditive, where **additive** is*

$$v^{kt}(S) = \sum_{i \in S} v^{kt}(\{i\}),\ \forall S \in M^{kt},$$

***monotonic** means*

$$v^{kt}(S) \leq v^{kt}(T),\ \forall S \subset T \subset M^{kt},$$

*and **superadditive** stands for*

$$v^{kt}(S) + v^{kt}(T) \leq v^{kt}(S \cup T),\ \forall S, T \subset M^{kt}, S \cup T = \phi.$$

Proof. Consider any $S \in M^{kt}$.

We first prove that the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ is additive. It is quiet obvious cause the characteristic function in our chapter

stands for the amount of data that can be shared and on the basis of eq. 6.6, we have

$$v^{kt}(\{i\}) = \sum_{i \in \{i\} \cap M_a^{kt}} w^{*kt}_i,$$

which means

$$v^{kt}(\{i\}) = w^{*kt}_i, \text{if } i \in M_a^{kt},$$
$$v^{kt}(\{i\}) = 0, \text{if } i \in M_b^{kt} \cup M_c^{kt}.$$

Hence, combining together we have,

$$\begin{aligned}\sum_{i \in S} v^{kt}(\{i\}) &= \sum_{i \in S \cap M_a^{kt}} w^{*kt}_i + \sum_{i \in (S - S \cap M_a^{kt})} w^{*kt}_i \\ &= \sum_{i \in S \cap M_a^{kt}} w^{*kt}_i = v^{kt}(S),\end{aligned}$$

then the game $\Gamma = (M^{kt}, v^{kt})$ is additive.

From $S \subset T$, we can get that $S \cap M_a^{kt} \subset T \cap M_a^{kt}$ which means for any $i \in S \cap M_a^{kt}$ we also have $i \in T \cap M_a^{kt}$, thus

$$v^{kt}(T) = \sum_{i \in S \cap M_a^{kt}} w^{*kt}_i + \sum_{i \in (T \cap M_a^{kt} - S \cap M_a^{kt})} w^{*kt}_i.$$

With strong requesters' definition in Definition 6.3.1, $w^{*kt}_i > 0$ for all $i \in (T \cap M_a^{kt} - S \cap M_a^{kt})$. As a result, the above equation stands which proves that the game $\Gamma = (M^{kt}, v^{kt})$ is monotonic.

Judging from eq. 6.6, we can easily split $v^{kt}(S \cup T)$ into

$$v^{kt}(S \cup T) = \sum_{i \in (S \cup T) \cap M_a^{kt}} w^{*kt}_i = \sum_{i \in (S \cap M_a^{kt}) \cup (T \cap M_a^{kt})} w^{*kt}_i,$$
$$= \sum_{i \in S \cap M_a^{kt}} w^{*kt}_i + \sum_{i \in T \cap M_a^{kt}} w^{*kt}_i = v^{kt}(S) + v^{kt}(T).$$

All in all, the game $\Gamma = (M^{kt}, v^{kt})$ is superadditive. □

The superadditivity notion states that it is advantageous (in terms of the amount of data to share) for disjoint coalitions to form their union which means it is beneficial for the time-sensitive MCS system that all strong requesters form a coalition together, while the monotonicity notion expresses that the amount of data available for sharing does not decrease whenever the coalition is enlarged which indicates the joining of weak requesters will not weaken the time-sensitive MCS system. Generally speaking, by proving Proposition 1 we can be sure that in cooperation phase, all requesters form a grand coalition is in favour of the time-sensitive MCS system which is used in the establishment of Algorithm 6.4.

Data reallocation of the dynamic cooperative MCS game

The allocation of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ should be around the amount of data which is available to share. Assume the data allocation is $\beta^{kt} = \{\beta^{kt}(1), \beta^{kt}(2), ..., \beta^{kt}(m^{kt})\}$, where $\beta^{kt}(i)$ is the amount of data requester i can get in cooperation phase at stage t based on the data reallocation function β^{kt}.

Not any data reallocation function β is proper in the time-sensitive MCS system, for the necessity and construction of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$, the data reallocation function must follow the two standards.

I. Strong requesters will only share their excess data, and weak requesters will only buy what they lack.

II. Weak requesters should not spend more money than budgeted to buy data.

Therefore, based on standards I and II, all possible data reallocation function β^{kt} of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ is defined in Definition 6.3.2.

Definition 6.3.4. *In the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$, $I(v)$ is said to be the set of all possible data reallocation function.*

$$I(v) = \{\beta^{kt} \in R^{M^{kt}} : \sum_{j \in M^{kt}} (I_{ij}^{kt} \beta_{ij}^{kt}(i)) \leq |w^{*kt}_i|, \ \forall i \in M^{kt}$$
$$\text{and} \sum_{j \in M_a^{kt}} (I_{ij}^{kt} \beta_{ij}^{kt}(i)(\frac{V_i^k}{3})) \leq \tilde{D}_i^{kt}, \ \forall i \in M_c^{kt}\},$$

$$I_{ij}^{kt} = 0, \textit{strong requester } j \textit{ not share data with weak requester } i,$$
$$I_{ij}^{kt} = 1, \textit{strong requester } j \textit{ shares data with weak requester } i.$$

where $\beta_{ij}^{kt}(i)$ is the amount of data requester $j \in M_a^{kt}$ shares to requester $i \in M_c^{kt}$, $\tilde{D}_i^{kt}$ is requesters remaining budget from the competition phase, $\tilde{D}_i^{kt} \in [0, D_i^{kt}]$, and

I_{is}^{kt} is an indicative function regarding to whether the extra data collected by strong requester s would be reallocated to weak requester i or not.

The first condition in $I(v)$ is correlated with the standard I and the second condition of $I(v)$ is in coincide with the standard II.

There are more than one data reallocation function β^{kt} in the set $I(v)$, while we need to find an optimal data reallocation function $\beta^{kt*} \in I(v)$ that is able to maximize the task completion rate of the time-sensitive MCS system to fulfill the needs of the system.

6.3.2 Optimal data reallocation mechanism design

The optimal data reallocation mechanism design is about finding the optimal data reallocation function $\beta^{kt*} \in I(v)$ of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ which can optimize the allocation of exceed data with standards I and II. The data reallocation mechanism is optimized for one single purpose–the maximization of the time-sensitive MCS system's task completion rate. Thus, through the data reallocation mechanism, we are able to transform the optimal solution $\beta^{kt*} \in I(v)$ of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ solving process into an optimization problem with constraint conditions. Generally speaking, the optimal data reallocation mechanism is mathematically formulated as follow:

$$\max \quad \frac{\sum_{i=1}^{m} \tilde{I}_i^{kT^i}}{m}, \ (T^i = \min\{T_i, t\}) \tag{6.7}$$

$$\text{s.t.} \quad \tilde{I}_i^{kT^i} = 1, \text{if} \sum_{t'=1}^{T^i} (\sum_{j=1}^{n} (\bar{I}_{ij}^{kt'} X_{ij}^{kt'}) + \tilde{w}_i^{kt'}) \geq \frac{W_i^k}{T_i} T^i, \tag{6.8}$$

$$\tilde{I}_i^{kT^i} = 0, \text{if } \sum_{t'=1}^{T^i}(\sum_{j=1}^{n}(\bar{I}_{ij}^{kt'} X_{ij}^{kt'}) + \tilde{w}_i^{kt'}) < \frac{W_i^k}{T_i} T^i, \tag{6.9}$$

$$\tilde{w}_i^{kt'} = \sum_{j \in M_a^{kt'}} (I_{ij}^{kt'} \beta_{ij}^{kt'}(j)), (\text{if } i \in M_c^{kt'}), \tag{6.10}$$

$$\tilde{w}_j^{kt'} = - \sum_{i \in M_c^{kt'}} (I_{ij}^{kt'} \beta_{ij}^{kt'}(i)), (\text{if } j \in M_a^{kt'}), \tag{6.11}$$

$$\sum_{j \in M_a^{kt}} (I_{ij}^{kt} \beta_{ij}^{kt}(i)) \leq |w^{*kt}_i|, \tag{6.12}$$

$$\sum_{j \in M_a^{kt'}} (I_{ij}^{kt'} \beta_{ij}^{kt'}(i) \frac{V_j^k}{3}) \leq \tilde{D}_i^{kt'}, \tag{6.13}$$

$$\forall t' \in \{1, 2, ..., T^i\},\ i \in M^{kt'}. \tag{6.14}$$

The objective function 6.7 is to maximize the percentage of requesters that complete tasks regarding to their schedule time which is the target of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ and $T^i = \min\{T_i, t\}$ is to guarantee that requesters do not exceed the specified time which is the requirement of the time-sensitive MCS system. Equation 6.8 and 6.9 are indicative function represents whether requesters have completed the data collection plan at stage t . The actual amount of data requester $i \in M^{kt}$ collected is indicated by eq. 6.10 and eq. 6.11 which is the optimal solution of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$. The standards I, and II of game $\Gamma = (M^{kt}, v^{kt})$ are represented by 6.12, and 6.13. Due to the failure of collecting enough data during competition, weak requesters would reduce their requirements for data quality in the cooperation to complete data collection on time along with the proven notion of Proposition 6.3.3, the satisfactory of all requesters in eq. 6.10 to eq. 6.13 is promised.

Up till now, the data reallocation mechanism has successfully transformed the optimal solution solving process of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ into optimization problem 6.7-6.13 solving. A data reallocation algorithm is introduced in Algorithm 6.4 where weak requesters will decide whether to get data from the strong requester or not in order (budgets, emergency degree and payments). If the weak requester has enough budgets, then data would be shared. For the sake of not wasting budgets, if the weak requester do not have enough budgets for the strong requester, then it would get as much data as its budgets allows.

6.3.3 Data reallocation algorithm of cooperative game

By solving the optimization problem with constraint conditions, the data reallocation algorithm in Algorithm 6.4 is able to get the optimal $\beta^{kt*} \in I(v)$ of the dynamic cooperative MCS game $\Gamma = (M^{kt}, v^{kt})$ which can maximize the task completion rate of the time-sensitive MCS system. The pseudo-code of the data reallocation algorithm is summarized in Algorithm 6.4.

Table 6.4: Data reallocation algorithm between requesters at stage t

Input: requester set $M = \{1, 2, \ldots, m\}$, total amount of requirements W_i^k, number of games that requester i can join T_i, and remaining budget constraints $\tilde{D}_i^{kt}$.
Output: solution concept $\beta_{is}^{kt}(i)$ and requesters' task completion situation.

1. Set $T^i = \min\{T_i, t\}$.
2. For each requester $i \in M^{kt}$ do:
3. Calculate the task completion margin according to eq. 6.5.
4. Calculate $\tilde{I}_i^{kT^i}$ based on eq. 6.12 and eq. 6.13.

5. If ${w^*}_i^{kt} > 0$ then:
6. Classify requester i as a strong requester and share its extra amount of data ${w^*}_i^{kt}$ with weak requesters.
7. Else if ${w^*}_i^{kt} = 0$ then:
8. Classify requester i as a middle requester.
9. Else:
10. Classify requester i as a weak requester and require a total amount of $|{w^*}_i^{kt}|$ data from strong requesters.
11. End if.
12. End for.
13. Calculate each requester's remaining budget from the competition phase as $\tilde{D}_i^{kt} = D_i^{kt} - \sum_{j \in N}(\bar{I}_{ij}^{kt} X_{ij}^{kt} \frac{V_i^k}{3})$.
14. Sort requesters in ascending order of payments, budgets, and emergency degree.
15. For each requester $i \in M_c^{kt}$ do:
16. If $T_i > t$ then:
17. For each requester $j \in M_a^{kt}$ do:
18. Calculate the amount of data to share as $w = \min\{|{w^*}_i^{kt}|, {w^*}_j^{kt}\}$.
19. If $\tilde{D}_i^{kt} > w\frac{V_j^k}{3}$ then:
20. Requester j shares its extra data with requester i and update $I_{ij}^{kt} = 1$.
21. Output $\beta_{ij}^{kt}(i) = w$.
22. Update $\tilde{D}_i^{kt} = \tilde{D}_i^{kt} - w\frac{V_j^k}{3}$.
23. Update $\tilde{D}_j^{kt} = \tilde{D}_j^{kt} + w\frac{V_j^k}{3}$.
24. Update ${w^*}_i^{kt} = {w^*}_i^{kt} + w$.
25. Update ${w^*}_j^{kt} = {w^*}_j^{kt} - w$.
26. If ${w^*}_i^{kt} = 0$ then:

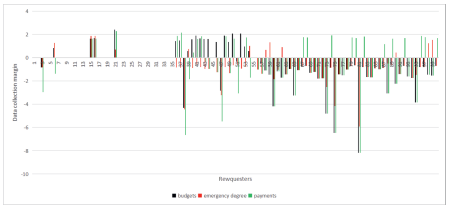
27. Update requester i as a middle requester and set $\tilde{I}_i^{kT^i} = 1$.
28. End if.
29. If $w^{*kt}_j = 0$ then:
30. Update requester j as a middle requester.
31. End if.
32. Else:
33. Set $w = \frac{3\tilde{D}_i^{kt}}{V_j^k}$.
34. Requester j shares its extra data with requester i and update $I_{ij}^{kt} = 1$.
35. Output $\beta_{ij}^{kt}(i) = w$.
36. Update $\tilde{D}_i^{kt} = 0$.
37. Update $\tilde{D}_j^{kt} = \tilde{D}_j^{kt} + \tilde{D}_i^{kt}$.
38. Update $w^{*kt}_i = w^{*kt}_i + w$.
39. Update $w^{*kt}_j = w^{*kt}_j - w$.
40. If $w^{*kt}_i = 0$ then:
41. Update requester i as a middle requester and set $\tilde{I}_i^{kT^i} = 1$.
42. End if.
43. If $w^{*kt}_j = 0$ then:
44. Update requester j as a middle requester.
45. End if.
46. End if.
47. End for.
48. End if.
49. End for.

Line 1 and 16 can guarantee requesters are within their time limitations. From

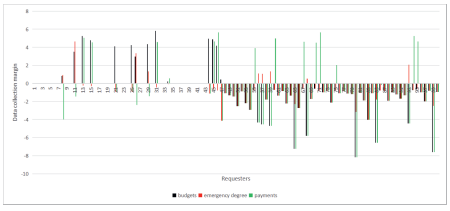
line 5 to line 11, requesters are regrouped into strong requesters, weak requesters or middle requesters with regard to their task completion margin w^{*kt}_i. Line 15 to line 51 is the specific data redistribution process, if requester $i \in M^{kt}_c$ has enough budget to gather data from requester $j \in M^{kt}_a$, then $\beta^{kt}_{ij}(i)$ is the smaller of the task completion margin between requester j and i according to standards I. Line 33 to line 48 is to make sure that weak requester do not waste their budgets which means if it does not have enough budget to collect enough data or all the amount of data strong requester can share, then it will purchase as much data as its budgets allows. The feasibility of line 20 and line 35 is guaranteed by Proposition 6.3.3 which is as long as requesters have enough budgets, they are willing to receive or share data.



(a) $N = 150, D_i \sim unifrnd(150, 200)$



(b) $N = 150, D_i \sim unifrnd(150, 200)$



(c) $N = 200, D_i \sim unifrnd(200, 250)$

Figure 6.3: Task completion margin by Algorithm 6.4

The parameters set in Fig. 6.3 is the same as in Fig. 6.1. The black, red and green column is the task completion margin of requesters when ordered in budgets, emergency degree and payments by Algorithm 6.4. Requesters with

column above the zero axis are those who collected too much data as in Table 6.5 and requesters with column bellow the zero axis are those who failed to get enough data. Requesters without any column means they got just enough data without any waste, along with requesters with column above the zero axis are those who completed data collection at stage $t = 1$ as shown in Table 6.6. In short words, requesters with column above the zero axis are strong requesters, bellow the zero axis are weak requesters and requesters with no column are middle requesters as defined in Definition 6.3.1.

Apparently, compared with Fig. 6.1, by cooperation some requesters failed to meet standards in competition can successfully gather enough data as scheduled and requesters who collect too much data in competition are able to share their exceed data. Later, we will compare the outcome in different orders and parameters in detail.

Table 6.5: Data waste rate in stage 1 with cooperation

(parameters,orders)	budgets	emergency degree	payments
$N = 100$,			
$D_i \sim unifrnd(100, 150)$	$94.11\% \to 44.44\%$	$42.10\% \to 9.52\%$	$72.72\% \to 31.94\%$
$N = 150$,			
$D_i \sim unifrnd(150, 200)$	$100\% \to 33.33\%$	$42.30\% \to 20.86\%$	$67.56\% \to 38.33\%$
$N = 200$,			
$D_i \sim unifrnd(200, 250)$	$100\% \to 31.25\%$	$69.56\% \to 8.86\%$	$78.04\% \to 27.27\%$

From Table 6.5, there is a significant decrease in data waste rate with cooperation compared with none cooperation. The waste rate of resources has been reduced at most 68.75% which proved the remarkable results of cooperation.

Table 6.6: Data completion rate in stage 1 with cooperation

(parameters,orders)	budgets	emergency degree	payments
$N = 100$,			
$D_i \sim unifrnd(100, 150)$	$16\% \rightarrow 34\%$	$8\% \rightarrow 27\%$	$24\% \rightarrow 37\%$
$N = 150$,			
$D_i \sim unifrnd(150, 200)$	$14\% \rightarrow 53\%$	$11\% \rightarrow 58\%$	$25\% \rightarrow 54\%$
$N = 200$,			
$D_i \sim unifrnd(200, 250)$	$16\% \rightarrow 47\%$	$16\% \rightarrow 41\%$	$32\% \rightarrow 48\%$

Compared with Table 6.3, the data completion rate in stage 1 with cooperation has been increased at most 47% with the same parameters set, which proved the significant performance of cooperation. Different requesters have various task end time, that is, requesters would join distinguish stages. However, Table 6.6 only concludes the task completion rate in one stage which cannot be used to measure the performance of the coopetition game. Therefore, later, we would evaluate performance of cooperation in whole stages.

6.4 Simulation results on the performance of time-sensitive MCS system

In this section, there are total 1000 requesters and 1000 participants to begin with, the data value of requester i with data type $k \in Q$ v_i^k is randomly generated from 100 to 150, data quality requirement α_i^k is a random number between 10 and 25, the unit costs of each participant is between 50 and 100, and the number of games that requester i can join T_i is between 10 and 110.

First, we will evaluate the performance of the dynamic coopetition MCS game with different orders to figure out the optimal order of requesters with the highest task completion rate, and then we will figure out the influence various parameters sets have on the task completion rate to prove that our performance is stable and not by chance.

6.4.1 Performance comparison of time-sensitive MCS system with requesters in different orders

To avoid contingency, we would take all kinds of orders into consideration during simulation to get the optimal order of requesters with the best performance (task completion rate).

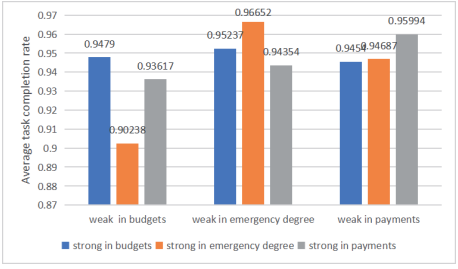


Figure 6.4: Task completion rate of time-sensitive MCS system, $D_i \sim unifrnd(100, 150)$

In Fig. 6.4, "weak in budgets" is short for "weak requesters are ordered by budgets" and the rest is in a similar way. The column in Fig. 6.4 represents the average task completion rate of the dynamic coopetition incentive mechanism in 100 episodes. Judging from columns of the same colour, it is obvious that when weak requesters are in the same order with strong requesters, the performance of the dynamic coopetition incentive mechanism is definitely

better than strong requesters and weak requesters are in distinct order. Compare among columns with different colours, when requesters are all in order of emergence degree or payments, the system performs better than when requesters are all in order of budgets. According to standards I, and II of game $\Gamma = (M^{kt}, v^{kt})$, weak requesters will not update their budgets in different stages which means requesters with the first option will always choose first, hence the requester who fails to collect the data on time will not be able to collect the data in later stages too. Even though from the perspective of a single requester, the more budget, the more likely it is to complete data collection. But from the long-term, system-wide perspective, for weak requesters, because the budget is the same for each round, those do not collect data in this round will never collect data, thus the overall system will perform poorly. As for sorting by emergency degree, the requester with the less time left gets the priority to collect data and is more likely to gather enough data in time. With time goes by, each requester is likely to get the first option sometimes, thus as a whole the system performs best when sorted by emergency degree.

Remark 1. In the coopetition MCS game, when strong requesters and weak requesters are in the same order, the time-sensitive MCS system has the highest task completion rate.

6.4.2 Performance comparison of time-sensitive MCS system with different parameters sets

As stated before, in this section, we will rule out the influence of parameters settings. Judging from standards I and II, the two parameters have significant affect on the performance (the task completion rate) of time-sensitive MCS system is the number of participants and requesters' budgets. Hence, to

analysis the impact of parameters, requesters are all in the same order according to Remark 1 and change the parameter settings gradually. To begin with, we fix the number of participants as 1000 which the same as requesters to avoid the effect of the number of participants on the results, and then change the budgets separately.

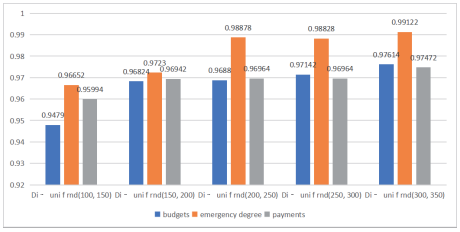


Figure 6.5: Task completion rate of time-sensitive MCS system with different budgets

Fig. 6.5 is the average task completion rate of the dynamic coopetition incentive mechanism in 100 episodes under different budgets settings when strong requesters and weak requesters are in the same order based on Remark 1. The blue column means all requesters are ordered in budgets, the orange one is the emergency degree, the grey one is payments and the number on top of each column is the average task completion rate. With the increase of budgets, the task completion rate of the time-sensitive MCS system generously grows up from the minimum 94% to the maximum 99%.

Remark 2. In the coopetition MCS game, despite the change of budgets, all requesters should be ordered by emergency degree which has an optimal 99.12% task completion rate.

After budgets, in Fig. 6.6, we exclude the effect of the number of participants on the performance of the dynamic coopetition incentive mechanism. By fixing the budgets the same as data values $D_i \sim unifrnd(100, 150)$ to avoid the

affect of budgets on the performance, and then alter the number of participants separately.

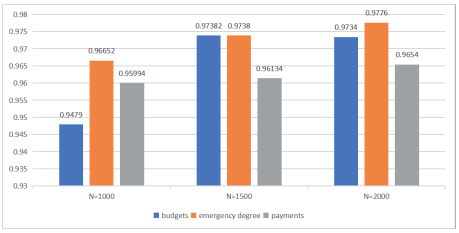


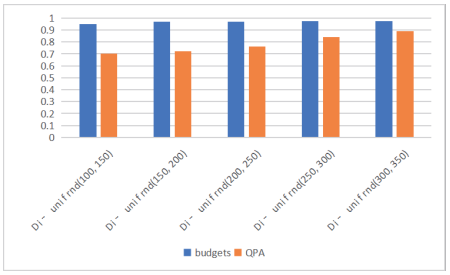
Figure 6.6: Task completion rate of time-sensitive MCS system with different participants' numbers

The change of number of participants only have a small effect on the best performance of the time-sensitive MCS system which is less than 1%, but improves the worst performance of the dynamic coopetition incentive mechanism. When sorted by emergency degree and payments, no matter how the number of participants changes, the task completion rate is always smaller than when budgets changes. Combined with Fig. 6.5, we can conclude that the number of participants determines the lower limit of algorithm performance, while the upper limit of algorithm performance is determined by budget which is in line with reality. The number of participants ensures the full amount of data that can be collected by requesters regardless of how requesters cooperate or compete with each other, thus, it determines the lower limit of time-sensitive MCS system's performance. The budget determines the maximum amount of data that can be collected by each requestor, hence it promises the upper limit of time-sensitive MCS system's performance.

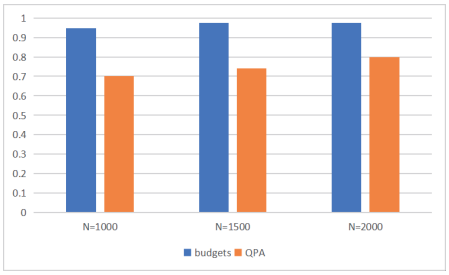
Remark 3. In the coopetition MCS game, the lower limit of the system's task completion rate is affected by participants' numbers and the budgets determine the upper limits of the system's performance.

6.4.3 Performance comparison between time-sensitive MCS system with classic quality/progress based algorithm

In this section, we will compare the task completion rate of our algorithm and the classic quality/progress based algorithm (QPA) (Gong, Zhang, and C. Li 2018; M. Xiao et al. 2016; L.-Y. Jiang et al. 2017). When a new participant arrives, QPA will add a new task for that participant based on the maximum quality cost ratio. In the implementation of QPA, the quality used is the task quality increment, and the cost used is the participant data collection cost.



(a) $N = 150, D_i \sim unifrnd(150, 200)$



(b) $N = 150, D_i \sim unifrnd(150, 200)$

Figure 6.7: Task completion rate comparison

Judging from Fig. 6.7, when requesters are ordered by budgets, the task completion rate of our algorithm is higher than the QPA algorithm despite of parameter seets which indicates that, our algorithm outperformed the QPA algorithm.

6.5 Conclusion

For time-sensitive MCS tasks with budget constraints, a multi-requesters competition and cooperation dynamic incentive mechanism is proposed

to increase tasks completion rate while maximize utilities. By theorems, utilities are guaranteed and by simulations, task completion rate can be up to 99% based on our dynamic incentive mechanism.

To balance the dilemma between requesters and the time-sensitive MCS system, rather than settling the two goals (utilities maximization and task completion rate maximization) simultaneously, a dynamic coopetition incentive mechanism is proposed to deal with them one by one. Therefore, our dynamic incentive mechanism can guarantee both aims in polynomial time, reduce the data waste rate from 100% to a minimum 8.86%, and improve the task completion rate to 99%.

Future works will extend our method to consider the effect of different budget allocation methods on the outcome of the budget in the competition and cooperation phases and the update of budgets at different stages.

Bibliography

[1] Abderrafi Abdeddine, Youssef Iraqi, and Loubna Mekouar. "An efficient task allocation in mobile crowdsensing environments". In: *IEEE Transactions on Network and Service Management* (2025).

[2] Ali Amiri. "The bundled task assignment problem in mobile crowdsensing: a lagrangean relaxation-based solution approach". In: *Information Technology and Management* (2024), pp. 1–12.

[3] Jose Apesteguia, Martin Dufwenberg, and Reinhard Selten. "Blowing the whistle". In: *Economic Theory* 31.1 (2007), pp. 143–166.

[4] Yu Awaya and Vijay Krishna. "On communication and collusion". In: *American Economic Review* 106.2 (2016), pp. 285–315.

[5] Iwan Bos, Berardino Cesi, and Marco A. Marini. "Cartel stability with quality-anchored buyers". In: *Economics Letters* 254 (2025), p. 112449.

[6] Iwan Bos and Joseph E. Harrington Jr. "Endogenous cartel formation with heterogeneous firms". In: *The RAND Journal of Economics* 41.1 (2010), pp. 92–117.

[7] Iwan Bos and Joseph E. Harrington Jr. "Competition policy and cartel size". In: *International Economic Review* 56.1 (2015), pp. 133–153.

[8] Iwan Bos, Wilko Letterie, and Dries Vermeulen. "Antitrust as facilitating factor for collusion". In: *The BE Journal of Economic Analysis & Policy* 15.2 (2015), pp. 797–814.

[9] Iwan Bos and Marco A. Marini. "Collusion in quality-segmented markets". In: *Journal of Public Economic Theory* 24.2 (2022), pp. 293–323.

[10] Iwan Bos and Erik Pot. "On the possibility of welfare-enhancing hard core cartels". In: *Journal of Economics* 107.3 (2012), pp. 199–216.

[11] Andrea Capponi et al. "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities". In: *IEEE communications surveys & tutorials* 21.3 (2019), pp. 2419–2465.

[12] H Chamberlin Edward. "The Theory of Monopolistic Competition, Cambridge, Mass". In: *Traduction française (1954) La théorie de la concurrence monopolistique, PUF, Paris* (1933).

[13] Shuang Chen, Min Liu, and Xiao Chen. "A truthful double auction for two-sided heterogeneous mobile crowdsensing markets". In: *Computer Communications* 81 (2016), pp. 31–42.

[14] Didem Cicek and Burak Kantarci. "Use of mobile crowdsensing in disaster management: A systematic review, challenges, and open issues". In: *Sensors* 23.3 (2023), p. 1699.

[15] Olivier Compte, Frédéric Jenny, and Patrick Rey. "Capacity constraints, mergers and collusion". In: *European Economic Review* 46.1 (2002), pp. 1–29.

[16] Claude d'Aspremont et al. "On the stability of collusive price leadership". In: *The Canadian Journal of Economics* (1983), pp. 17–25.

[17] Charles Darwin. "On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life". In: *London: Murray* 18 (1859), p. 59.

[18] Raymond Deneckere and Carl Davidson. "Incentives to Form Coalitions with Bertrand Competition". In: *The RAND Journal of Economics* 16.4 (1985), pp. 473–486.

[19] Giovanna Devetag and Andreas Ortmann. “When and why? A critical survey on coordination failure in the laboratory”. In: *Experimental Economics* 10.3 (2007), pp. 331–344.

[20] Effrosyni Diamantoudi. “Stable cartels revisited”. In: *Economic Theory* 26.4 (2005), pp. 907–921.

[21] M.-P. Donsimoni, N. S. Economides, and H. M. Polemarchakis. “Stable Cartels”. In: *International Economic Review* 27.2 (1986), pp. 317–327.

[22] Marie-Paule Donsimoni. “Stable heterogeneous cartels”. In: *International Journal of Industrial Organization* 3.4 (1985), pp. 451–467.

[23] Theo Driessen. *Cooperative games, solutions and applications*. Springer Science & Business Media, 2013.

[24] Prabal Dutta et al. “Common sense: participatory urban sensing using a network of handheld air quality monitors”. In: *Proceedings of the 7th ACM conference on embedded networked sensor systems*. 2009, pp. 349–350.

[25] European Commission. *Antitrust: Commission fines six LCD panel producers €648 million for price fixing cartel*. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_10_1685. Dec. 2010.

[26] Federal Reserve. *Federal Reserve announces fines totaling more than $1.8 billion against six major banking organizations for their unsafe and unsound practices in the foreign exchange (FX) markets*. URL: https://www.federalreserve.gov/newsevents/pressreleases/enforcement20150520a.htm. May 2015.

[27] Miguel A. Fonseca and Hans-Theo Normann. “Explicit vs. tacit collusion–The impact of communication in oligopoly experiments”. In: *European Economic Review* 56.8 (2012), pp. 1759–1772.

[28] Miguel A. Fonseca and Hans-Theo Normann. "Endogenous cartel formation: Experimental evidence". In: *Economics Letters* 125.2 (2014), pp. 223–225.

[29] Agostino Forestiero and Giuseppe Papuzzo. "Recommendation platform in Internet of Things leveraging on a self-organizing multiagent approach". In: *Neural Computing and Applications* 34.18 (2022), pp. 16049–16060.

[30] David Genesove and Wallace P. Mullin. "Rules, communication, and collusion: Narrative evidence from the sugar institute case". In: *American Economic Review* 91.3 (2001), pp. 379–398.

[31] Wei Gong, Baoxian Zhang, and Cheng Li. "Location-based online task assignment and path planning for mobile crowdsensing". In: *IEEE Transactions on Vehicular Technology* 68.2 (2018), pp. 1772–1783.

[32] Joseph E. Harrington and Andrzej Skrzypacz. "Private monitoring and communication in cartels: Explaining recent collusive practices". In: *American Economic Review* 101.6 (2011), pp. 2425–2449.

[33] Kuan Lun Huang, Salil S Kanhere, and Wen Hu. "On the need for a reputation system in mobile phone based sensing". In: *Ad Hoc Networks* 12 (2014), pp. 130–149.

[34] Steffen Huck, Hans-Theo Normann, and Jörg Oechssler. "Two are few and four are many: Number effects in experimental oligopolies". In: *Journal of Economic Behavior & Organization* 53.4 (2004), pp. 435–446.

[35] Mitsuru Igami and Takuo Sugaya. "Measuring the incentive to collude: The vitamin cartels, 1990–99". In: *The Review of Economic Studies* 89.3 (2022), pp. 1460–1494.

[36] Aridegbe A Ipaye et al. “Location and time aware multitask allocation in mobile crowd-sensing based on genetic algorithm”. In: *Sensors* 22.8 (2022), p. 3013.

[37] Ling-Yun Jiang et al. “Quality-Aware Incentive Mechanism for Mobile Crowd Sensing”. In: *Journal of Sensors* 2017.1 (2017), p. 5757125.

[38] Haiming Jin, Lu Su, and Klara Nahrstedt. “CENTURION: Incentivizing multi-requester mobile crowd sensing”. In: *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE. 2017, pp. 1–9.

[39] Abhimanyu Khan and Ronald Peeters. “Stable cartel configurations and product differentiation: The case of multiple cartels”. In: *Mathematical social sciences* 132 (2024), pp. 57–68.

[40] Abhimanyu Khan and Ronald Peeters. *Stability of multiple cartels in differentiated markets*. `http://dx.doi.org/10.2139/ssrn.5338629`. Available at SSRN 5338629. 2025.

[41] Sunyoung Kim et al. “Creek watch: pairing usefulness and usability for successful citizen science”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2011, pp. 2125–2134.

[42] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. “Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory”. In: *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 2011, pp. 127–140.

[43] Robin Kraft et al. “Dealing with inaccurate sensor data in the context of mobile crowdsensing and mhealth”. In: *IEEE Journal of Biomedical and Health Informatics* 26.11 (2022), pp. 5439–5449.

[44] Margaret C Levenstein and Valerie Y Suslow. “What determines cartel success?” In: *Journal of economic literature* 44.1 (2006), pp. 43–95.

[45] Margaret C. Levenstein and Valerie Y. Suslow. "Breaking up is hard to do: Determinants of cartel duration". In: *The Journal of Law and Economics* 54.2 (2011), pp. 455–492.

[46] Youqi Li et al. "MP-coopetition: Competitive and cooperative mechanism for multiple platforms in mobile crowd sensing". In: *IEEE Transactions on Services Computing* 14.6 (2019), pp. 1864–1876.

[47] Nicolas Maisonneuve et al. "NoiseTube: Measuring and mapping noise pollution with mobile phones". In: *Information Technologies in Environmental Engineering: Proceedings of the 4th International ICSC Symposium Thessaloniki, Greece, May 28-29, 2009*. Springer. 2009, pp. 215–228.

[48] Harsh Mankodiya et al. "A real-time crowdsensing framework for potential COVID-19 carrier detection using wearable sensors". In: *Mathematics* 10.16 (2022), p. 2927.

[49] Jesse W Markham. "The nature and significance of price leadership". In: *The American Economic Review* 41.5 (1951), pp. 891–905.

[50] David A. Miller. "Robust collusion with private information". In: *The Review of Economic Studies* 79.2 (2011), pp. 778–811.

[51] Nathan H. Miller. "Strategic leniency and cartel enforcement". In: *American Economic Review* 99.3 (2009), pp. 750–768.

[52] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.

[53] Jiangtian Nie et al. "A multi-leader multi-follower game-based analysis for incentive mechanisms in socially-aware mobile crowdsensing". In: *IEEE Transactions on Wireless Communications* 20.3 (2020), pp. 1457–1471.

[54] Dan Peng, Fan Wu, and Guihai Chen. “Pay as how well you do: A quality based incentive mechanism for crowdsensing”. In: *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. 2015, pp. 177–186.

[55] Zhenlong Peng et al. “Multi-task oriented data diffusion and transmission paradigm in crowdsensing based on city public traffic”. In: *Computer Networks* 156 (2019), pp. 41–51.

[56] Robert H. Porter and J. Douglas Zona. “Detection of Bid Rigging in Procurement Auctions”. In: *Journal of Political Economy* 101.3 (1993), pp. 518–538.

[57] Rüdiger Pryss et al. “Mobile crowd sensing services for tinnitus assessment, therapy, and research”. In: *2015 IEEE international conference on mobile services*. IEEE. 2015, pp. 352–359.

[58] Hamta Sedghani et al. “An incentive mechanism based on a Stackelberg game for mobile crowdsensing systems with budget constraint”. In: *Ad Hoc Networks* 123 (2021), p. 102626.

[59] Reinhard Selten. “A simple model of imperfect competition, where 4 are few and 6 are many”. In: *International Journal of Game Theory* 2.1 (1973), pp. 141–201.

[60] Ryoichi Shinkuma et al. “Incentive mechanism for mobile crowdsensing in spatial information prediction using machine learning”. In: *International Conference on Advanced Information Networking and Applications*. Springer. 2020, pp. 792–803.

[61] Martin Shubik and Richard Levitan. *Market Structure and Behavior*. Cambridge, MA and London, England: Harvard University Press, 1980.

[62] Bernd Simon et al. “Decentralized online learning in task assignment games for mobile crowdsensing”. In: *IEEE Transactions on Communications* 72.8 (2024), pp. 4945–4960.

[63] Shiwei Song et al. “Coverage-oriented task assignment for mobile crowdsensing”. In: *IEEE Internet of Things journal* 7.8 (2020), pp. 7407–7418.

[64] Amanda Starc and Thomas G Wollmann. “Does entry remedy collusion? Evidence from the generic prescription drug cartel”. In: *American Economic Review* 115.5 (2025), pp. 1400–1438.

[65] Sergio Trilles et al. “Development of an open sensorized platform in a smart agriculture context: A vineyard support system for monitoring mildew disease”. In: *Sustainable Computing: Informatics and Systems* 28 (2020), p. 100309.

[66] Lewis Tseng et al. “Blockchain-based database in an IoT environment: challenges, opportunities, and analysis”. In: *Cluster Computing* 23.3 (2020), pp. 2151–2165.

[67] U.S. Department of Justice. *Five Major Banks Agree to Parent-Level Guilty Pleas*. URL: `https://www.justice.gov/archives/opa/pr/five-major-banks-agree-parent-level-guilty-pleas`. May 2015.

[68] Helder Vasconcelos. “Tacit Collusion, Cost Asymmetries, and Mergers”. In: *The RAND Journal of Economics* 36.1 (2005), pp. 39–62.

[69] Danxin Wang et al. “A general location-authentication based secure participant recruitment scheme for vehicular crowdsensing”. In: *Computer Networks* 171 (2020), p. 107152.

[70] Liang Xiao, Donghua Jiang, et al. “Secure mobile crowdsensing based on deep learning”. In: *China Communications* 15.10 (2018), pp. 1–11.

[71] Liang Xiao, Jinliang Liu, et al. “Secure mobile crowdsensing game”. In: *2015 IEEE International Conference on Communications (ICC)*. IEEE. 2015, pp. 7157–7162.

[72] Mingjun Xiao et al. “Online task assignment for crowdsensing in predictable mobile social networks”. In: *IEEE Transactions on Mobile Computing* 16.8 (2016), pp. 2306–2320.

[73] Zhilin Xu, Hao Sun, and Weibin Han. “A collaboration-driven mechanism for AI diagnose with multiple requesters under incomplete information”. In: *Computer Networks* 220 (2023), p. 109472.

[74] Zhilin Xu, Hao Sun, and Weibin Han. “Boosting task completion rate for time-sensitive MCS system”. In: *Computer Networks* 251 (2024), p. 110636.

[75] Zhilin Xu, Hao Sun, and Panfei Sun. “Irregular mobility: A dynamic alliance formation incentive mechanism under incomplete information”. In: *Information Sciences* 712 (2025), p. 122155.

[76] Zhilin Xu, Hao Sun, Panfei Sun, and Qianqian Kong. “Requester mobility for mobile crowdsensing system: A dynamic alliance-based incentive mechanism”. In: *Ad Hoc Networks* 166 (2025), p. 103680.

[77] Shuo Yang et al. “Fine-grained user profiling for personalized task matching in mobile crowdsensing”. In: *arXiv preprint arXiv:1812.02074* (2018).

[78] Tieying Yu and Albert A Cannella Jr. “A comprehensive review of multimarket competition research”. In: *Journal of Management* 39.1 (2013), pp. 76–109.

[79] Fatih Yucel and Eyuphan Bulut. "Three-dimensional stable task assignment in semi-opportunistic mobile crowdsensing". In: *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE. 2022, pp. 386–395.

Summary

This thesis examines strategic decision-making in settings where individual incentives may conflict with collective efficiency. Motivated by this tension, it studies the emergence and stability of cooperation under competition, incomplete information and changing participation. The analysis considers coalition formation in Bertrand model and strategic interaction in Mobile Crowdsensing systems with heterogeneous requesters and participants. Despite the differences between these settings, the thesis addresses a common economic question: under what conditions can cooperation be sustained?

Chapter 2 studies cooperation in competitive markets. Using a Bertrand model with collusion, it examines the effects of the number of firms on coalition formation and characterizes the size of stable coalitions under different market sizes. The results show that collusion can persist in equilibrium as the number of firms increases. The analysis also shows how the formation of coalitions affects prices and profits in the market.

Chapter 3 develops a cooperation-driven model for the MCS system with multiple heterogeneous requesters. By modeling the interactions between requesters and participants as multiple sub-games, the proposed model enables collaborative data collection while maintaining individual rationality of the requesters. Based on this model, a payment method and a cost-sharing method are designed to improve data collection quality and ensure the benefits of cooperation. To address the case of incomplete information, a parameter learning method based on reinforcement learning (Q-learning) is further

introduced to estimate participants' private information from historical data, thereby supporting the design of dynamic payment strategies.

Chapter 4 develops a dynamic model to characterize the MCS system where requesters can join or leave the system while also demands can change over time. In this setting, a staged matching algorithm and corresponding state-update rules are designed to dynamically coordinate requester demands with participant resources. The proposed approach reduces the system cost while improving frequency of requester participation and maintaining the stability of the matching outcome.

Chapter 5 further addresses the challenges arising from incomplete information and the irregular presence of participants by proposing a participant-centric matching mechanism. The mechanism performs participant-centric task allocation to adapt to the dynamic changes in participants' presence. Furthermore, a requester classification algorithm is designed to reduce the computational complexity of the matching algorithm. Theoretical analysis demonstrates that the proposed mechanism achieves stable matching performance across all stages of the decision-making process.

Chapter 6 addresses the time-sensitive nature of sensing tasks by proposing a dynamic coopetition mechanism that integrates competition and cooperation. In the competition stage, a budget-constrained matching algorithm is designed to improve the utility of requesters. In the cooperation stage, a data redistribution strategy is introduced to enhance the overall task completion rate of the system. Theoretical analysis and simulation results demonstrate that, compared with a purely competitive mechanism, the proposed dynamic coopetition mechanism can significantly improve the overall task completion rate while maintaining an acceptable utility level for individual requesters.

Taken together, this thesis examines how cooperation can arise and persist under different forms of strategic interaction. The first part focuses on coalition formation in Bertrand competition, while the second part studies incentive mechanisms for cooperation in Mobile Crowdsensing systems. Despite their different settings, both parts examine how individual incentives and collective outcomes are connected.

Samenvatting

Deze dissertatie onderzoekt strategische besluitvorming in situaties waarin individuele belangen kunnen botsen met collectieve efficiëntie. Vanuit deze tegenstelling onderzoekt de dissertatie het ontstaan en de stabiliteit van samenwerking onder concurrentie, onvolledige informatie en veranderende deelname. De analyse behandelt coalitievorming in het Bertrand-model en strategische interactie in mobiele crowdsensing-systemen met heterogene opdrachtgevers en deelnemers. Ondanks de verschillen tussen deze situaties behandelt de dissertatie een gemeenschappelijke economische vraag: onder welke voorwaarden kan samenwerking worden behouden?

Hoofdstuk 2 onderzoekt samenwerking tussen agenten in competitieve markten. Met behulp van een Bertrand-model met collusie worden de effecten van het aantal ondernemingen op coalitievorming onderzocht. Ook wordt de omvang van stabiele coalities bij verschillende marktomvang bepaald. De resultaten laten zien dat collusie in evenwicht mogelijk blijft wanneer het aantal ondernemingen toeneemt. De analyse laat ook zien hoe coalitievorming de prijzen en winsten in de markt beïnvloedt.

Hoofdstuk 3 ontwikkelt een samenwerkingsgericht model voor een mobiel crowdsensing-systeem met meerdere heterogene opdrachtgevers. De interacties tussen opdrachtgevers en deelnemers worden gemodelleerd als meerdere deelspellen. Het model maakt gezamenlijke gegevensverzameling mogelijk en behoudt tegelijkertijd de individuele rationaliteit van de opdrachtgevers. Op basis van dit model worden een betalingsmethode en een methode voor kostendeling ontworpen. Deze methoden verbeteren

de kwaliteit van de gegevensverzameling en waarborgen de voordelen van samenwerking. Voor situaties met onvolledige informatie wordt daarnaast een methode voor het leren van parameters op basis van reinforcement learning (Q-learning) geïntroduceerd. Deze methode schat de privé-informatie van deelnemers op basis van historische gegevens. Dit ondersteunt het ontwerp van dynamische betalingsstrategieën.

Hoofdstuk 4 ontwikkelt een dynamisch model voor een mobiel crowdsensing-systeem. In dit systeem kunnen opdrachtgevers toetreden tot of vertrekken uit het systeem. Ook kunnen hun behoeften in de tijd veranderen. In deze situatie worden een gefaseerd matchingalgoritme en bijbehorende regels voor het bijwerken van de toestand ontworpen. Deze methode stemt de behoeften van opdrachtgevers dynamisch af op de beschikbare middelen van deelnemers. De voorgestelde methode verlaagt de systeemkosten. Tegelijkertijd verhoogt zij de deelnamefrequentie van opdrachtgevers en behoudt zij de stabiliteit van het matchingresultaat.

Hoofdstuk 5 behandelt de uitdagingen die ontstaan door onvolledige informatie en veranderende aanwezigheid van deelnemers. Hiervoor wordt een deelnemergerichte matchingmethode voorgesteld. Deze methode verdeelt taken vanuit het perspectief van de deelnemers. Zo kan de methode zich aanpassen aan dynamische veranderingen in hun aanwezigheid. Daarnaast wordt een classificatiealgoritme voor opdrachtgevers ontworpen. Dit algoritme vermindert de computationele complexiteit van het matchingalgoritme. De theoretische analyse laat zien dat de voorgestelde methode in alle fasen van het besluitvormingsproces stabiele matchingresultaten oplevert.

Hoofdstuk 6 behandelt het tijdgevoelige karakter van sensing-taken. Hiervoor wordt een dynamisch coöpetitiemechanisme voorgesteld dat

concurrentie en samenwerking combineert. In de concurrentiefase wordt een budgetbeperkt matchingalgoritme ontworpen. Dit algoritme verbetert het nut van de opdrachtgevers. In de samenwerkingsfase wordt een strategie voor gegevensherverdeling geïntroduceerd. Deze strategie verbetert het totale voltooiingspercentage van de taken in het systeem. De theoretische analyse en de simulatieresultaten laten zien dat het voorgestelde dynamische coöpetitiemechanisme, vergeleken met een mechanisme met alleen concurrentie, het totale voltooiingspercentage van de taken aanzienlijk kan verbeteren. Tegelijkertijd blijft het individuele nut van de opdrachtgevers op een acceptabel niveau.

Samengevat onderzoekt deze dissertatie hoe samenwerking kan ontstaan en behouden kan blijven onder verschillende vormen van strategische interactie. Het eerste deel richt zich op coalitievorming in Bertrand-concurrentie. Het tweede deel onderzoekt stimulansmechanismen voor samenwerking in mobiele crowdsensing-systemen. Ondanks de verschillen tussen beide delen onderzoeken zij beide de relatie tussen individuele belangen en collectieve resultaten.

Impact of the thesis

In this thesis, we study the societal impact of cooperation between strategic agents. We show that cooperation between agents can both harm and improve the welfare of society. On one hand, cooperation can occur naturally due to agents' preferences, while it is not necessarily socially desirable. On the other hand, cooperation may be socially desirable but may not align with personal incentives of agents.

Understanding this tension is therefore important for both explaining strategic behavior and designing mechanisms that can improve collective performance without violating individual rationality.

In the first part of the thesis, we focus on the situation in which cooperation is endogenous but potentially harmful to society. An important insight in the first part concerns the stability of a cartel in an oligopolistic setting with price competition. In this setting, cartel formation is a natural and often occurring phenomenon. The thesis identifies the conditions under which a stable cartel can emerge. The size of such stable cartel is uniquely determined, with an upper bound of six firms. This shows that a stable cartel can exist even in a market with a very large number of firms. Consequently, a cartel may consist of only a small fraction of the firms in a large market while still being strategically stable. Such small cartels may receive less attention precisely because their size is relatively small compared with the market as a whole. This finding has important implications for cartel detection and antitrust enforcement. It suggests that the absence of a large-scale cartel structure should not necessarily be interpreted as evidence that stable collusion is unlikely to exist

in a large market. Instead, competition authorities may also need to pay attention to relatively small but stable groups of firms when assessing potential collusive behavior.

The second part of this thesis provides insights into the emergence and stability of cooperation under competition in different settings. We study stability of cooperation under incomplete information, heterogeneous interests, and changing participation. Under such conditions, cooperation cannot simply be assumed to occur. It must be supported by mechanisms that make cooperative behavior individually acceptable while also generating benefits at the collective level. The analysis in this thesis helps clarify how the underlying incentive structures shape the formation, stability, and effectiveness of cooperative arrangements.

The second part of the thesis specifically concerns Mobile Crowdsensing (MCS) systems. MCS systems rely on a large number of distributed participants who use mobile devices or sensing equipment. Such devices can be used to collect data for requesters who use the data to develop applications for mobile devices and electronic equipment. For example, such applications are used in smart cities, transportation, environmental monitoring, healthcare, and other data-intensive services. However, requesters typically face limited budgets and may have varying data requirements and priorities. These characteristics naturally create conflicts between individual incentives and system-level objectives.

The incentive mechanisms and cooperation-based approaches studied in this thesis provide guidance for addressing these conflicts. By taking individual rationality and resource constraints into account, the proposed mechanisms seek to encourage participation while improving the efficiency of collective resource allocation. Cooperation among requesters can also provide a means

of reducing inefficiencies that arise when requesters independently compete for the same sensing resources. In particular, alliance-based mechanisms illustrate how requesters with different capabilities or requirements can coordinate their actions while maintaining incentives for individual participation. Such mechanisms can be valuable in practical platforms where limited budgets, heterogeneous requirements, and competition for participants make purely independent decision-making inefficient.

Another important practical implication is the need to account for heterogeneity and dynamics when designing incentive mechanisms. Participants in MCS systems may differ in their locations, sensing capabilities, costs, data quality, and willingness to participate. Requesters may also differ in their budgets, data requirements, priorities, and task urgency. Furthermore, participants and requesters may enter or leave the system over time. This calls for the design of a mechanism which takes such types of heterogeneity among agents in the system into account. Considering these characteristics make the proposed framework more closely connected to real-world decentralized platforms, where decision makers and available resources cannot be assumed to be or remain homogeneous.

The relevance of these findings extend beyond Mobile Crowdsensing systems. Similar problems occur in sharing services, collaborative sensing networks, and decentralized data markets. These examples share the same underlying tension between individual rationality and collective efficiency. The theoretical and methodological perspectives developed in this thesis can therefore provide broader guidance for designing incentive mechanisms in such cases.

The thesis also contributes to the societal value of economic research by showing how theoretical concepts concerning strategic behavior and

cooperation can be used to address practical resource-allocation and coordination problems. As digital platforms become increasingly important in economic and social activities, more services depend on interactions among distributed individuals and organizations. In these environments, centralized control may be difficult, costly, or inappropriate. Mechanisms that can align individual incentives with collective objectives may therefore support more efficient, reliable, and sustainable forms of decentralized cooperation.

Overall, the significance of this thesis lies in its ability to connect economic questions about cooperation with practical challenges. The first part shows that cooperation may occur endogenous and may be difficult to detect. The second part shows that cooperation does not necessarily require complete coordination or the elimination of competition. Instead, under appropriate conditions, cooperation can emerge from strategic interaction when mechanisms are designed so that individual rationality and collective efficiency are sufficiently aligned. By providing theoretical insights into the conditions supporting cooperation and translating these insights into incentive and alliance-based mechanisms, the thesis contributes to both economic understanding and practical system design. These findings can ultimately support the development of more efficient and sustainable interactive mechanisms in an increasingly interconnected world.

About the author

Zhilin Xu was born in Yancheng, P.R. of China. In 2015, she became a student at Northwestern Polytechnical University and received a Bachelor of Science degree in 2019. After graduation, she continued her studies for a master's degree at Northwestern Polytechnical University, supervised by Prof. Dr. Hao Sun. One year later, she decided to join the Master-PhD project and subsequently became a PhD student under Prof. Dr. Hao Sun in 2021. From September 2024, she was awarded a scholarship under the State Scholarship Fund to pursue study in the School of Business and Economics, Maastricht University, as a joint PhD student, under the supervision of Prof. Dr. Dries Vermeulen and Dr. Christopher Kops. Her research focuses on game theory and its applications in multi-agent systems, with particular emphasis on cooperation mechanisms and incentive design under dynamic and incomplete information environments. Her work integrates theoretical modeling with algorithmic design to analyze strategic interactions and improve system performance in complex environments.



Small steps
lead to
big dreams



Keep going



Finally here