

Mechanism Design for Strategic Crowdsourcing

A Thesis

Submitted for the Degree of
Doctor of Philosophy
in the **Faculty of Engineering**

by

Swaprava Nath



Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

DECEMBER 2013

© Swaprava Nath
December 2013
All rights reserved

DEDICATED TO

My parents

who introduced me to the beautiful game called life

Signature of the Author:

.....

Swaprava Nath
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Signature of the Thesis Supervisor:

.....

Y. Narahari
Professor
Dept. of Computer Science and Automation
Indian Institute of Science, Bangalore

Acknowledgements

I thank my advisor, Prof. Y. Narahari, not only for giving me the opportunity to work on several interesting problems, but also for the freedom he gave to choose the research topics. It is because of him, I got the opportunity of working with the world class researchers in India and abroad. Beyond academics, he had been a great motivator that helped a graduate student to evolve through several stages of the research life. If I become a teacher someday, I wish I could be as helpful to my students as he had been to me.

Onno and Chris, it had been wonderful working with you. David, I must say that the research atmosphere in Harvard was awesome and despite a short stint, I could gather some experience which will be helpful in my entire research career. Dinesh and Pankaj, you folks gave me some indigenous research flavor that is of the highest class. Panos and James, thanks for your care during my stay at Harvard, and engaging in some thrilling discussions on research, life, and philosophies. Murali, you have been great, both in math and interpreting them to life; I wish I could stay in Bangalore longer to have some more such discussions. Kundan and Bhushan, with the hard work and determination, I am sure you will go a long way.

I express my gratitude to Prof. Arunava Sen, for patiently listening to all the presentations I made and the discussions we had on different occasions, and giving his well-thought comments and thereby motivating young researchers. I thank Prof. Anurag Kumar, who taught me how to *write in mathematics* rather than writing in English. I enjoyed several discussions with Prof. N. Viswanadham on the various stages of research life.

Thanks to the Department of Computer Science and Automation, along with all the faculty members, students, and non-teaching staff, who made my stay in the department a charm. I would express my gratitude to the department of Electrical Communication Engineering, where I was first introduced to *real* research, and got baptized to the research philosophy of the Indian Institute of Science. The courses offered by CSA, ECE, and Math departments of IISc had been an excellent learning experience. The institute offers an environment, arguably the best in India, to work, play, explore, and learn. Starting from the on-campus hostel facilities to the food in the mess, from huge academic resources to the recreational facilities of the gymkhana, it had

Acknowledgements

provided me a lot. Gymkhana football ground needs a special mention, since it had been a stress reliever on numerous occasions.

My parents have a special place in my heart as they were the people who could motivate my research from a distance of 2000 kms. I apologize to them for the fact that I had not been very accessible during the past few years, with the promise that it is going to end soon.

To me, a PhD is more than just the research. The social life is a significant component of it, which IISc provides in abundance. I am lucky to have some great friends in my life who made this significant component a pleasure. Among others, Achintya needs a special mention, who showed me the ease at which the pressure of a research life can be handled. Apart from being a multi-talented genius, he had been another great motivator for me in the past years. Rohith, I am glad that I had such a nice labmate like you. I really enjoyed our discussions, travels, lab treats, going to the festivals, and so on. I am sure I am missing many on this part, but do not feel disheartened, I do remember all of you even though it is not written here.

The Game Theory Lab (formerly the Electronic Commerce Lab, and even formerly, Electronic Enterprises Lab), with the tradition of its changing names, keeps on changing the members as well. The former and current lab members, Sujit, Suri, Sunil, Udaya, Rohith, Satya, Swapnil, Shweta, Pankaj, Praphul, Murali, Chetan, Moon, Mayur, Premm, Kiran, Alfia, Ratul, Sourav, Ashutosh, Palash, Dilpreet, Debmalaya, Gaurav, Arup, Arpita, Priyanka, Akanksha, Afsal have made the lab a lively place to work and study. The IISc B Mess brings a fresh breeze to the summer of work pressure. Instead of naming individually, I would like to thank all the present and past boarders of B Mess and the staff for helping us with the ambience, chit-chats, fun, frolic, and of course hygienic and delicious food. My life-long loyalty with football brings me closer to the gymkhana grounds and the people whom I played with. In particular, I must mention the team with which we won the intra-IISc football tournament earlier this year.

Though hardly anybody acknowledges this, but I feel Google products including the search, scholar, citation etc. play a crucial role in the research today. Finding the right resources is as important as the research itself. And when you feel tired, a nice video or song on YouTube is at your rescue. So, no way I can deny the contribution of Google in my research.

Last but not the least, I thank the Ministry of Human Resource Development, Government of India, for supporting my research for the first year and Tata Consultancy Services (TCS) for supporting me with the Doctoral Fellowship for the past 3 years of my PhD. The contribution of Xerox Research in offering me an internship and funding travels is gratefully acknowledged. I also thank Yahoo! Research for recognizing my research with the Yahoo! Key Scientific Challenges Honorable Mention Award in 2012.

Abstract

This thesis looks into the economics of *crowdsourcing* using game theoretic modeling. The art of aggregating information and expertise from a diverse population has been in practice since a long time. The Internet and the revolution in communication and computational technologies have made this task easier and given birth to a new era of online resource aggregation, which is now popularly referred to as crowdsourcing. Two important features of this aggregation technique are: (a) crowdsourcing is always human driven, hence the participants are rational and intelligent, and they have a payoff function that they aim to maximize, and (b) the participants are connected over a social network which helps to reach out to a large set of individuals. To understand the behavior and the outcome of such a strategic crowd, we need to understand the economics of a crowdsourcing network. In this thesis, we have considered the following three major facets of the strategic crowdsourcing problem.

- (i) *Elicitation of the true qualities of the crowd workers*: As the crowd is often unstructured and unknown to the designer, it is important to ensure if the crowdsourced job is indeed performed at the highest quality, and this requires elicitation of the true qualities which are typically the participants' private information.
- (ii) *Resource critical task execution ensuring the authenticity of both the information and the identity of the participants*: Due to the diverse geographical, cultural, socio-economic reasons, crowdsourcing entails certain manipulations that are unusual in the classical theory. The design has to be robust enough to handle fake identities or incorrect information provided by the crowd while performing crowdsourcing contests.
- (iii) *Improving the productive output of the crowdsourcing network*: As the designer's goal is to maximize a certain measurable output of the crowdsourcing system, an interesting question is how one can design the incentive scheme and/or the network so that the system performs at an optimal level taking into account the strategic nature of the individuals.

In the thesis, we design novel mechanisms to solve the problems above using game theoretic modeling. Our investigation helps in understanding certain limits of achievability, and provides design protocols in order to make crowdsourcing more reliable, effective, and productive.

Research Papers from the Thesis

1. **Productive Output in Hierarchical Crowdsourcing**, Swaprava Nath and Balakrishnan Narayanaswamy, to appear, in Proceedings of the Conference on *Autonomous Agents and Multi Agent Systems (AAMAS)*, 2014.
2. **A Strict Ex-post Incentive Compatible Mechanism for Interdependent Valuations**, Swaprava Nath, Onno Zoeter, *Economics Letters*, 121(2):321–325, 2013.
3. **Mechanism Design for Time Critical and Cost Critical Task Execution via Crowdsourcing**, Swaprava Nath, Pankaj Dayama, Dinesh Garg, Y. Narahari, James Zou, In *Proceedings of Web and Internet Economics (WINE 2012)*, pages 212–226, 2012.
4. **Threats and Trade-offs in Resource Critical Crowdsourcing Tasks over Networks**, Swaprava Nath, Pankaj Dayama, Dinesh Garg, Y. Narahari, James Zou, In *Proceedings of the Association for Advancement Artificial Intelligence (AAAI 2012)*, pages 2447–2448, 2012.
5. **Dynamic Mechanism Design for Markets with Strategic Resources**, Swaprava Nath, Onno Zoeter, Y. Narahari, Chris Dance, In *Proceedings of Uncertainty in Artificial Intelligence (UAI 2011)*, pages 539–546. AUAI Press, 2011.
6. **Dynamic Learning-based Mechanism Design for Dependent Valued Exchange Economies**, Swaprava Nath, In *Proceedings of the International World Wide Web Conference (WWW 2011)*, pages 397–402, 2011.
7. **Dynamic Mechanism Design with Interdependent Valuations**, Swaprava Nath, Onno Zoeter, Y. Narahari, Chris Dance, submitted to *Review of Economic Design*.
8. **Improving Productive Output in Influencer-Influencee Networks**, Swaprava Nath, Balakrishnan Narayanaswamy, working paper, arXiv preprint arXiv:1302.3045, 2013.

Contents

Acknowledgements	i
Abstract	iii
Research Papers from the Thesis	v
Contents	vii
List of Figures	xi
List of Tables	xiii
List of Acronyms	xv
1 Introduction	1
1.1 Crowdsourcing: An Introduction	1
1.1.1 Early Day Example: The Oxford English Dictionary	2
1.1.2 Early Day Example: The Smithsonian Experience	3
1.1.3 Modern Day Example: Wikipedia	3
1.1.4 Modern Day Example: ESP Game	4
1.1.5 Modern Day Example: Linux and Opensource	4
1.1.6 Modern Day Example: InnoCentive	5
1.1.7 Modern Day Example: DARPA Red Balloon Challenge	5
1.2 The Economics of Crowdsourcing	8
1.3 Contributions of the Thesis	9
1.3.1 Eliciting Skills of the Strategic Workers	9
1.3.2 Resource Critical Task Execution via Crowdsourcing	10
1.3.3 Efficient Team Formation from a Crowd	11

1.4	Outline of the Thesis	11
2	Game Theory and Mechanism Design: A Quick Review	17
2.1	Game Theory	17
2.2	Mechanism Design	23
2.2.1	Mechanism Design Environment	24
2.2.2	Direct and Indirect Mechanisms	26
2.2.3	Implementation of Social Choice Functions	27
2.2.4	Bayesian Game Induced by a Mechanism	28
2.2.5	Implementation of a Social Choice Function by a Mechanism	28
2.2.6	A Limit to the Desire: The Gibbard-Satterthwaite Impossibility Theorem	30
2.2.7	Domain Restriction: Quasi-Linear Utilities	32
2.2.8	Mechanism Design with Interdependent Valuations	34
2.2.9	Dynamic Mechanism Design	35
3	Skill Elicitation in Crowdsourcing	37
3.1	Static Mechanism Design with Interdependent Valuations	39
3.1.1	Introduction	39
3.1.2	Model and Definitions	40
3.1.3	Main Results	43
3.2	Dynamic Mechanism Design with Interdependent Valuations	49
3.2.1	Introduction	49
3.2.2	Background and Model	52
3.2.3	The MATRIX Mechanism	56
3.2.4	Simulations	65
3.2.5	A Discussion on the Characterization of Dynamic Mechanisms	68
3.3	Conclusions	69
4	Resource Critical Task Execution via Crowdsourcing	71
4.1	Introduction	72
4.2	Prior Work	75
4.3	Contributions and Outline of the Chapter	76
4.4	The Model	78
4.4.1	Desirable Properties	79
4.5	Impossibility and Possibility Results	80
4.6	Approximate Versions of Desirable Properties	82

4.6.1	Motivation for δ -SCR and γ -SEC	83
4.7	Cost Critical Tasks	84
4.7.1	Characterization Theorem for MINCOST	85
4.8	Time Critical Tasks	87
4.8.1	Characterization Theorem for MAXLEAF	88
4.9	Partial Information Aggregation: Synergy among the Agents	90
4.10	Conclusions	91
4.A	Proof of Two Lemmata	91
5	Efficient Team Formation using a Strategic Crowd	93
5.1	Incentive Design for Improving Productive Output	94
5.1.1	Prior Work	96
5.1.2	Overview and Main Results	97
5.1.3	A Hierarchical Model of Influencer and Influencee	98
5.1.4	A General Network Model of Influencer and Influencee	111
5.2	Network Design for Improving Productive Output	115
5.2.1	The Model	116
5.2.2	Drawback: Free Riding	122
5.2.3	Benefit: Information Sharing	126
5.2.4	Trade-off between Free-riding and Information Sharing	128
5.2.5	An Algorithm for Finding a Near Optimal Hierarchy	132
5.3	Conclusions	133
5.A	Proofs for the Exponential Productivity Model	135
5.B	Proofs of the Price of Anarchy Results in Balanced Hierarchies	137
5.C	Proofs for General Networks	140
6	Conclusions and Future Research	143
6.1	Conclusions	143
6.2	Scope of Future Research	145
	Bibliography	147

List of Figures

1.1	Commercial crowdsourcing platforms. Image courtesy: crowdsourcing.org	7
1.2	Key research components of the crowdsourcing problem.	9
2.1	The fair division problem.	24
2.2	The general mechanism design environment.	27
2.3	The indirect mechanism design environment and its implications.	29
3.1	Online image annotation via crowdsourcing.	38
3.2	Graphical illustration of VCPM.	41
3.3	Graphical illustration of a candidate dynamic mechanism in an interdependent value setting.	54
3.4	Utility of task owner and production team 1 under MATRIX as a function of true type profiles. The ordering of the $3^3 = 27$ type profiles is represented in the bottom-most plot.	67
3.5	Utility of task owner and team 1 under CONST as function of true type profiles. The x-axis follows same true profile order as in Fig. 3.4.	68
3.6	Payment consistency and budget properties of MATRIX. The x-axis follows same true profile order as in Fig. 3.4.	69
4.1	The DARPA red balloon challenge location map.	74
4.2	MIT team’s solution to DARPA red balloon challenge.	74
4.3	Sybil attack in the MIT scheme.	75
4.4	Collapse attack.	75
4.5	The space of $(\delta, \epsilon, \gamma)$ tuples characterized by Theorem 4.4.	87
5.1	A typical corporate hierarchy.	96
5.2	A typical hierarchical model.	99
5.3	Tightness of the sufficiency (Theorem 5.2).	104

LIST OF FIGURES

5.4	FLAT hierarchy.	106
5.5	Bounds on PoA for FLAT, $d = 6, D = 1$	110
5.6	Impact on the production effort of a node having higher EFOR from the influencees.	112
5.7	Geometric interpretation of the condition for uniqueness of the Nash equilibrium.	115
5.8	Server queue model for the task arrival to the network.	117
5.9	Node 1 is rewarded indirectly by 2 and 3.	119
5.10	Function μ , with $a = 1, s = 3, b = 4$	121
5.11	Plot of $f(T_i, \Delta)$ (Δ averaged) versus $ T_i $	123
5.12	Plot of $\sum_{i \in N} \lambda_i^*$ versus the geometric reward share factor, δ	126
5.13	Hierarchy formed when free riding is not a concern.	128
5.14	Plot of $\sum_{i \in N} f(T_i, \Delta)$ (Δ averaged) vs n	129
5.15	Hierarchy formed considering free riding.	132
5.16	Absolute difference in the fractional decay rate of risk bound in OPT-CFR with OPT-IFR.	133
5.17	Percentage difference in the fractional decay rate of risk bound in OPT-CFR with OPT-IFR.	133
5.18	Fractional decay rate of the risk bound $\rho_{\text{CFR}}(t, T)$ comparison between the Opti- mal and Sequential Greedy algorithms (Algorithm 3). The x-axis shows different average variances of the population. Number of nodes = 6.	133
6.1	Summary of the findings of the thesis.	143

List of Tables

2.1	The prisoner's dilemma game.	18
2.2	The modified prisoner's dilemma game.	20
2.3	The battle of sexes game.	21
2.4	The matching pennies game.	22
3.1	The different paradigms of mechanism design problems with their solutions. . .	51
3.2	Simulation summary	68

List of Acronyms

BB	Budget Balance
BIC	Bayesian Incentive Compatible
BNI	Bayesian Nash Implementable
CFR	Consider Free-Riding
CP	Collapse-Proofness
CR	Contribution Rationality
DARPA	Defense Advanced Research Projects Agency
DPM	Dynamic Pivot Mechanism
DSIC	Dominant Strategy Incentive Compatible
DSI	Dominant strategy implementable
DSP	Downstream Sybil-Proofness
EFF	Efficient
EP	Exponential Productivity
EPIC	Ex-Post Incentive Compatible
EPIR	Ex-Post Individually Rational
EUf	Effort Update Function
IFR	Ignore Free-Riding
MATRIX	MDP-based Allocation and TRansfer in Interdepedent valued eXchange economies
MBR	MLE Best Response
MDP	Markov Decision Process
MLE	Maximum Likelihood Estimate
MSNE	Mixed Strategy Nash Equilibrium
OPT-CFR	Optimization Considering Free-Riding
OPT-IFR	Optimization Ignoring Free-Riding
PoA	Price of Anarchy

PIV	Peer-Influenced Valuations
PSNE	Pure Strategy Nash Equilibrium
SA	Subset Allocation
SCF	Social Choice Function
SCR	Strict Contribution Rationality
SDSE	Strongly Dominant Strategy Equilibrium
SMTT	Stationary Markov Type Transitions
SO	Social Output
VCG	Vickrey Clarke Groves mechanism
VCPM	Value Consistent Pivotal Mechanism
WCR	Weak Contribution Rationality
WDSE	Weakly Dominant Strategy Equilibrium
WTA	Winner Takes All

Chapter 1

Introduction

Individuals and organizations often face the challenge of executing tasks for which they do not have enough resources or expertise. Outsourcing tasks to experts at a cost helps them execute the tasks without procuring extra resources. The idea of outsourcing for commercial purposes started in the late twentieth century and is actively in practice for businesses even today. However, with the advent of the Internet, outsourcing has become even more convenient. Online social networks have given access to a huge *crowd* with plenty of diverse expertise. Today it is easy to find a group of people to collectively solve a problem or generate a web content by aggregating collective knowledge, which is known as *crowdsourcing* in literature. Examples of crowdsourcing applications are many, and we will discuss a few in this chapter. Even though the concept looks appealing and innovative, there are certain challenges in designing the crowdsourcing protocols that ensure authenticity of the solutions, individuals, and their strategies.

This thesis addresses the question of designing efficient crowdsourcing mechanisms when the crowd consists of rational and intelligent participants. Since the majority of crowdsourcing applications involve tasks and payments to be routed online, these applications use techniques from *computer science*. On the other hand, to analyze the behavior of the strategic participants and to design efficient crowdsourcing protocols, one needs to leverage the concepts of *game theory* and *mechanism design*, two classic tools from the *microeconomic theory*. This thesis considers certain interesting theoretical questions in strategic crowdsourcing problem, and provides answers that help explain the limits of achievability and provide design prescriptions for efficient crowdsourcing.

1.1 Crowdsourcing: An Introduction

According to the Merriam-Webster Dictionary, *crowdsourcing* is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, and especially

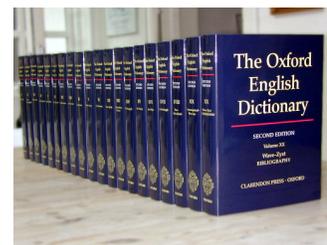
from an online community, rather than from traditional employees or suppliers. Howe [36] coined the term *crowdsourcing* in an article written in the Wired magazine in 2006. It is a portmanteau of ‘crowd’ and ‘outsourcing’. Though initially the term was used to refer to a generalized version of outsourcing, a lot of effort has gone in designing a more efficient crowdsourcing mechanism, and at the same time, to find what is its most exhaustive definition. Recently, Estellés-Arolas and González-Ladrón-de Guevara [27] provided a definition which is quite detailed in terms of the functioning of crowdsourcing.

“Crowdsourcing is a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage that what the user has brought to the venture, whose form will depend on the type of activity undertaken.”

Crowdsourcing came to much limelight after it was used extensively in the online social media. However, there have been many examples of historical usage of crowdsourcing. We mention a few of them here.

1.1.1 Early Day Example: The Oxford English Dictionary

The Oxford English Dictionary (OED) came into existence through crowdsourcing, which is arguably one of the earliest examples of this methodology. In the late nineteenth century, Professor James Murray was leading a literary project that draws from the knowledge, expertise and time of tens of thousands of volunteers. An open call was made to the community for contributions by volunteers to index all words in the English language and example quotations of their usages for each one. Operating in Oxford, England, he received hundreds of thousands of slips of paper over the course of several decades, each containing the definition of a particular English word. To Murray, these contributors were strangers - unpaid, but working together as one to collate the definitions and origins of every word in the English language [45].



The Oxford English Dictionary.

They received over 6 million submissions over a period of 70 years. The making of the OED is detailed in the book by Winchester [88].¹

1.1.2 Early Day Example: The Smithsonian Experience



“Secretary Henry Posts Daily Weather Map in Smithsonian Building, 1858,” by Louise Rochon Hoover, Commissioned for the Smithsonian exhibition at the Chicago Century of Progress Exhibition in 1933

The Smithsonian Institution, established in 1846 “for the increase and diffusion of knowledge”, is a group of museums and research centers administered by the United States government. One of the first crowdsourcing projects at the Institution was the Meteorological Project started by the Smithsonian’s first Secretary, Joseph Henry. In 1849, he set up a network of some 150 voluntary weather observers all over the country. Within a decade, the project had more than 600 voluntary observers and had spread to Canada, Mexico, Latin America, and the Caribbean. The amateur weather enthusiasts submitted monthly reports that were then analyzed by James H. Coffin, professor of mathematics and natural philosophy at Lafayette College in Easton, Pennsylvania, and finally published in

1861 in the first of a two volume compilation of climatic data and storm observations based on the volunteers’ reports [15].²

The era of computer networks and the Internet resulting in a communication revolution has made the job of crowdsourcing easier. A myriad of crowdsourcing applications and platforms throng the online social media ranging from blogs, opinion polls, user review forums, opensource softwares, and many more. Here are some examples.

1.1.3 Modern Day Example: Wikipedia

Wikipedia is a classic example of harnessing the knowledge of a crowd. Conceptualized by opensource stalwart Richard Stallman in December 2000, Wikipedia was officially launched on January 15, 2001 by Jimmy Wales and Larry Sanger, using the concept and technology of a ‘wiki’ pioneered in 1995 by Ward Cunningham. As opposed to the



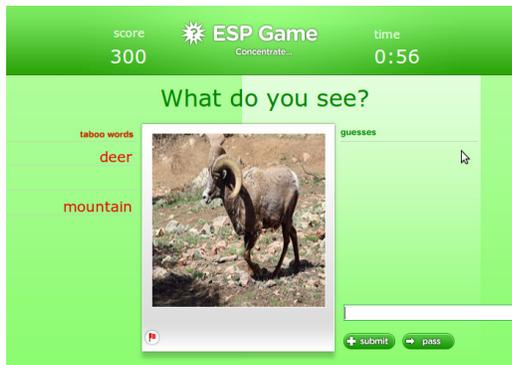
¹Image courtesy: [http:// allthingsd.com](http://allthingsd.com)

²Image courtesy: Smithsonian Institution Archives Record Unit 7005, Box 186, Folder: 4; Record Unit 95, Box 11, Folder: 12, Negative Number: 84-207.

expert based system, the idea of letting the crowd edit and control the quality of the content revolutionized the content generation process for Wikipedia.

As of May 2013, Wikipedia includes over 26 million freely usable articles in 285 languages, written by over 39 million registered users and numerous anonymous contributors worldwide [85]. According to Alexa-Internet [1], Wikipedia is the world's sixth-most-popular website, visited monthly by around 11% of all internet users. ¹

1.1.4 Modern Day Example: ESP Game



Human image recognition capabilities are much better than that of a computer. For example, a human observer can look at certain images and clearly figure out the details of an image, which a machine, equipped with the most sophisticated image processing algorithm often is not able to discern. This particular capability is the key of the game called ESP (extra sensory perception). In the disguise of a game, the designer can build a database of images by exploiting certain intrinsic human skills.

The rules of the game is the following. Once logged in, a user is automatically matched with a random partner from some other part of the Internet. The partners do not know each other's identity and they cannot communicate. A same image is shown to both of them. Their task is to agree on a word that would be an appropriate label for the image. They both enter possible words, and once the words entered by both partners (not necessarily at the same time) match, that word becomes a label for the image. In the game, they both get some points, and move to the next round where they are shown some different image. One can hope that if two random human labelers label the same image with the same tag, it must be a right tag for the image. The game was originally conceived by Luis von Ahn of Carnegie Mellon University [8]. ²

1.1.5 Modern Day Example: Linux and Opensource

Linux is by far the most popular opensource operating system. However, the majority of this operating system is developed and maintained by the Linux community, which is a set of individuals interested in developing softwares and disseminating them free of cost along with their source codes. This is part of the opensource software movement initiated in 1983

¹Image courtesy: <http://www.wikipedia.org/>

²Image courtesy: <http://www.espgame.org/>

by Richard Stallman. The defining component of Linux is the Linux kernel, which was first released on October 5, 1991, by Linus Torvalds.

Since it originated from the GNU project, the Free Software Foundation prefers the name GNU/Linux. Even today, the philosophy of the operating system remains to cater people with software that is free and opensource. The idea is to contribute voluntarily and get the benefits of the contributions from other people. In other words, this is a collaborative crowdsourcing project that has revolutionized the operating systems [86].¹



1.1.6 Modern Day Example: InnoCentive

A screenshot of the InnoCentive website. The header includes the InnoCentive logo and navigation links like 'My IC', 'Products/Services', 'For Solvers', 'Challenge Center', and 'Resources'. The main content area features a challenge card with a red handprint icon, the title 'USAID & Humanity United: How to Identify and Spotlight Intentional and Unintentional Enablers of Mass Atrocities?', and details such as 'AWARD: \$10,000 USD', 'DEADLINE: 11/28/12', and 'ACTIVE SOLVERS: 241'. A red circle highlights the award amount.

InnoCentive Inc. is an open innovation company based in US which acts as a platform between task posters and task executers, used usually for large projects with a significant amount of prize money. The research and development problems associated with these projects lie in a broad range of domains such as engineering, computer science, mathematics, chemistry, life sciences, physical sciences,

and business. The problems are phrased as “challenge problems” for anyone to solve. In 2006, Prize4Life partnered with InnoCentive to launch the \$1 million ALS Biomarker Prize, which was a Grand Challenge designed to find a biomarker (a biological indicator) to measure the progression of ALS - also known as Lou Gehrigs disease - in patients. In February 2011, the \$1 million prize was awarded to Dr. Seward Rutkove for his creation and validation of a clinically viable biomarker. In early 2011, InnoCentive launched four more Grand Challenges on behalf of Life Technologies. In the space of crowdsourcing platforms, InnoCentive offers one of the largest monetary rewards for executing challenging tasks [44].²

1.1.7 Modern Day Example: DARPA Red Balloon Challenge

In 2009, the defense research organization of the United States, DARPA, introduced a network challenge which is popularly known as the DARPA *Red Balloon Challenge* [21]. The challenge was to identify the locations of 10 red weather balloons in the shortest possible time. In the press release, DARPA said: any individual or organization who reports the correct location

¹Image courtesy: The original Tux, the official Linux mascot created by Larry Ewing, <http://en.wikipedia.org/wiki/File:Tux.png>

²Image courtesy: <https://www.innocentive.com/>



of all of the balloons in the shortest time will get a reward of \$40,000. Balloons were spread across the continental US, and it is impossible for any individual to travel to all the places and locate them. Crowdsourcing emerged as a natural approach in this setting. The champion was the team from MIT [65] who implemented a crowdsourcing based incentive scheme for people to participate and earn rewards for their contributions, and this scheme found the accurate locations of all the balloons in less than 9 hours. ¹

There are many other crowdsourcing applications in the web space. *Amazon Mechanical Turk* is one of the early examples of online crowdsourcing platform. The other example of such online crowdsourcing platforms include *oDesk*, *Rent-A-Coder*, *kaggle*, *Galaxy Zoo*, and *Stardust@home*. In recent times, an explosive growth in online social media has given a novel twist to crowdsourcing applications where participants can exploit the *underlying social network* for inviting their friends to help executing the task. In such a scenario, the task owner initially recruits individuals from her immediate network to participate in executing the task. These individuals, apart from attempting to execute the task by themselves, recruit other individuals in their respective social networks to also attempt the task and further grow the network. Apart from the DARPA *Red Balloon Challenge* [21], *DARPA CLIQR quest* [22], *query incentive networks* [41], and *multi-level marketing* [26] fill this space for crowdsourcing over social networks.

A larger space of crowdsourcing applications and platforms is shown in Figure 1.1. Many of them are commercially active, that is, the participants are paid monetary compensations for the task they do on those platforms. In particular, the last two examples of modern day crowdsourcing also fall into the monetary compensation category. Since the crowd is unstructured and unknown, the monetary reward can lead them towards behaving strategically to maximize their individual payoffs. It is, therefore, worthwhile and timely to conduct a thorough economic analysis of the crowdsourcing problem. In the following section, we discuss three major paradigms of crowdsourcing that require a detailed economic analysis.

¹Image courtesy: <http://archive.darpa.mil/networkchallenge/>



Figure 1.1: Commercial crowdsourcing platforms. Image courtesy: crowdsourcing.org

1.2 The Economics of Crowdsourcing

The need to study the economics of crowdsourcing arises from the fact that the crowd could potentially be strategic. Crowdsourcing has proved to be a splendid tool to aggregate the knowledge from a pool of individuals in order to perform a task efficiently, and it also leverages the social connectivity of the individuals in a crowd. However, we need to remember that the crowd consists of human participants, who are rational and intelligent (either fully or partially). They have payoffs for participating in the crowdsourcing applications and pick their actions so that it maximizes their payoffs. The payoffs could be monetary or non-monetary, but the goal is to maximize it given the fact that there are multiple other participants who also aim for a similar goal. This induces a strategic interaction or *game* between the crowd workers, and that is why we need to do a game theoretic modeling of crowdsourcing. The design of a crowdsourcing mechanism and an incentive scheme need to be carefully chosen in order to mitigate the untoward consequences of the strategic behavior. Addressing different paradigms of this strategic crowdsourcing setting is the theme of this thesis as we provide solutions to them. We consider three important aspects of the crowdsourcing problem in this thesis.

- (a) *Elicitation of the true qualities of the crowd workers.* This is important since the crowd is often unstructured and unknown to the designer, and therefore, it is difficult to determine if the job that is crowdsourced is indeed being performed at the highest quality.
- (b) *Resource critical task execution ensuring the authenticity of both the information and the identity of the participants.* Due to the diverse geographical, cultural, socio-economic reasons, crowdsourcing entails certain manipulations that are unusual in the classic mechanism design theory. For example, individuals can cheat by creating fake identities, or virtually reincarnating as a different individual, thus cheating the system, or by providing false information. The design has to consider all these aspects.
- (c) *Improving the productive output of the crowdsourcing network.* As a designer, the goal is to maximize some measurable output of the crowdsourcing system as a whole. An interesting question is that how one can design the incentive scheme and/or the network so that the system performs at an optimal level considering the strategic nature of the individuals.

In this thesis, we address the three above mentioned aspects of the crowdsourcing problem, shown graphically in Figure 1.2.

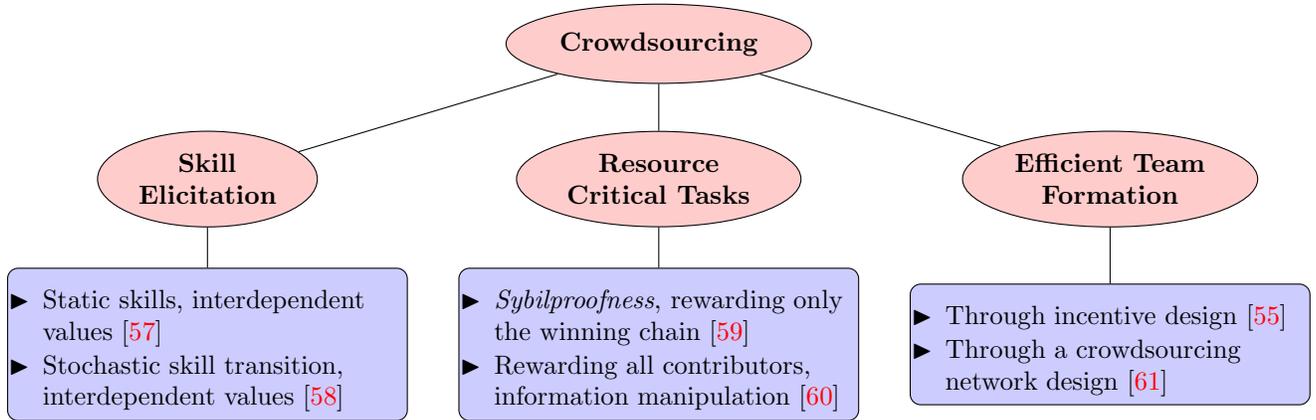


Figure 1.2: Key research components of the crowdsourcing problem.

1.3 Contributions of the Thesis

This thesis considers three related problem domains in crowdsourcing as given in Figure 1.2, and provides solutions to each of them. Each domain is interesting on its own right since the three domains handle three major aspects that a crowdsourcer (an individual or an organization) would be concerned about. We model certain crowdsourcing applications to be interactions between strategic agents and provide mechanism design solutions to them. In the following, we discuss our major findings and summarize our contributions to the space of crowdsourcing mechanisms.

1.3.1 Eliciting Skills of the Strategic Workers

In this part, we focus on an optimal crowd worker selection problem. The goal of the crowdsourcer is to select an optimal subset from a known set of experts. The experts privately observe their skills (hidden from the crowdsourcer) and the goal of the incentivizing mechanism is to ensure that they report their skills truthfully. Task execution by a selected subset of the experts fall into the category of *interdependent values* since the benefit experienced by a participant depends on the outcome and the skills of *all* the other participants. We address this question in this thesis and show the following.

- First, we address the mechanism design problem in a static setting, that is, the private skills are invariant over time. It has been shown that for interdependent valuations, it is impossible to design a single stage mechanism that can ensure efficiency and truthful reports of skill by the participants [39]. However, there is a two stage mechanism [48]

that can mitigate this problem, but not completely. In the second stage of this mechanism, players are indifferent between truthful reporting and misreporting. We provide an improved mechanism that makes truthful reporting a strictly better strategy for all agents.

- Next, we consider the setting where the experts' qualities vary stochastically over time according to a Markov process. For the independent values, Bergemann and Välimäki [9] have proposed an efficient mechanism called the *dynamic pivot mechanism*, which is a generalization of the classic Vickrey-Clarke-Groves (VCG) mechanism. However, the problem of dynamic mechanism design with interdependent valuations has received very little attention in the literature [16]. We look into this problem and propose a dynamic mechanism that is efficient and truthful. Under a special setting, this mechanism incentivizes the agents to voluntarily participate in the game.

1.3.2 Resource Critical Task Execution via Crowdsourcing

Crowdsourcing is unique in the potential strategic manipulation that it offers to its participants. One such problem in resource critical crowdsourcing contests is that it can have *false-name attacks* or *sybil attacks*, in which the players can increase their payoffs by creating multiple false identities of themselves. This is undesirable since it denies the right payment to the actual contributors. At the same time it is necessary to ensure that the information provided by the participants are true. We address these questions in the following way.

- We show that in crowdsourcing contests like the DARPA red balloon challenge [21], which falls into the class of atomic tasks, it is impossible to satisfy sybilproofness, collapse-proofness, and a property that rewards all contributors with positive compensation. We introduce approximate versions of these properties and show that the mechanism space is non-empty. To the best of our knowledge, this is a first attempt to design *approximately sybilproof crowdsourcing mechanisms*. Under certain resource critical paradigms, some of those properties are more preferable than the others and we characterize the space of mechanisms that satisfy those properties under *cost-critical* and *time-critical* paradigms.
- We notice that in resource critical crowdsourcing contests, e.g., the red balloon challenge, a great deal of human effort can get wasted, as people can potentially explore the same region already explored by someone else. A more time efficient and fair scheme could be to distribute the reward in proportion to the information contributed by each agent. For example, if an agent searches for an object in a certain part of a city and reports

that the object is not present in that region, it saves a lot of time and effort for the other participants. We discuss briefly about this point and provide a solution proposal using tools from information theory and prediction markets to aggregate information in a truthful, time efficient manner.

1.3.3 Efficient Team Formation from a Crowd

The next interesting question in crowdsourcing is to understand how highly productive teams emerge from a loosely structured population. A crowdsourcing network, once formed, is very similar to an organizational network, where the reason for success (failure) is often attributed to the (im)proper *management* of human resources. The goal of the designer in this context is to maximize the net productive output of the networked crowdsourcing system. We build on the papers of Bramoullé and Kranton [14], Ballester et al. [7] and develop an understanding of the effort levels in influencer-influencee networks. In particular, we show the following.

- First, we analyze how individuals, connected in a network, trade off between their production and communication efforts given the network positions and the reward sharing scheme. We show that under certain sufficient conditions, there exists a unique Nash equilibrium of this effort trade-off.
- We then provide a condition of achievability of optimal social output for certain representative networks. Our results show that the equilibrium output may not always achieve the globally optimal. However, by choosing the right reward sharing scheme we can maximize the output and we provide a recipe of such reward sharing schemes.
- Next, we analyze how the network design can lead to interesting structures for maximizing the net productive output. We show that a network design that considers the strategic behavior of the agents performs better than the one that does not consider it.

1.4 Outline of the Thesis

In this section, we provide a brief outline of the problems addressed and the organization of each chapter of this thesis.

Chapter 2: Game Theory and Mechanism Design: A Quick Review

The goal of this chapter is to introduce the reader to the basic concepts and definitions of game theory and mechanism design. We discuss only the part that is required to present the concepts of this thesis satisfactorily. The chapter is organized as follows.

- We begin with a basic example of a game to introduce the concept of game theory, and using the example introduce the notation for a *strategic form game*.
- Using multiple other examples, we introduce the concepts of *strongly* and *weakly dominant strategies* and the corresponding equilibria.
- Next we introduce the concept of the *pure strategy* and *mixed strategy* Nash equilibrium, which concludes the game theory review section.
- The section on mechanism design also starts with popular examples to introduce the setup and notation.
- We define the *social choice function* (SCF), *direct* and *indirect* mechanisms, and the implementation of SCFs using direct and indirect mechanisms.
- We discuss how the indirect mechanism induces a *Bayesian game* among the players and discuss about *dominant strategy* and *Bayesian incentive compatibility* of a SCF. The section concludes with the *revelation principle* which essentially says that the direct revelation mechanisms are enough to consider.
- Next we state a very fundamental theorem named after Gibbard and Satterthwaite (GS) which shows the limits of achievability in mechanism design. This is a negative result that shows that three very desirable properties are not simultaneously satisfiable.
- One way to circumvent the roadblock caused by the GS theorem is to consider a restricted domain. For the purpose of this thesis, we focus on the quasi-linear domain, and exhibit a mechanism that satisfies certain good properties.
- We present a very general characterization of the quasi-linear domain with *private valuations* given by Roberts, which basically shows that if the type set is rich enough, the implementable SCFs in quasi-linear domain has to be affine maximizers.
- In the next two sections, we briefly review *mechanisms with interdependent valuations* and *dynamic mechanisms* which sets the stage for the next chapter where we address the crowdsourcing problem from these paradigms and provide solutions.

Chapter 3: Skill Elicitation in Crowdsourcing

This chapter deals with the first major contribution of this thesis. It models the skill elicitation problem in crowdsourcing as a direct revelation mechanism with interdependent valuations. The sequence of the presentation is as follows.

- In the first part of this chapter, we focus on the classical problem of mechanism design with *interdependent valuation*.
- We refer to an impossibility result in this setting that refrains us from designing a single stage mechanism. The current state-of-the-art mechanism given by Mezzetti yields a two-stage solution, but suffers from the *weak indifference problem* in the second stage.
- We setup the model and definitions of *efficiency*, *ex-post incentive compatibility*, *ex-post individual rationality*, and a domain called *subset allocation*.
- We propose a mechanism called Value Consistent Pivotal Mechanism (VCPM) which satisfies the first two properties mentioned above in the interdependent value setting. It is a two-stage mechanism but it solves the problem of the weak indifference in the second stage.
- In addition, VCPM is also ex-post individually rational in the subset allocation domain. However, the improvement in the weak indifference of the classic two-stage mechanism of Mezzetti comes at the cost of subgame perfection. We discuss this with an example. At the end of this part, we discuss how the designer's observation of a variable introduced in the classic mechanism can affect the VCPM.
- In the second part of this chapter, we look into the *dynamic mechanism design* problem with interdependent valuation. The dynamicity comes from the fact that the types of the agents vary according to a stationary Markov process.
- We propose a mechanism called MDP-based Allocation and TRansfer in Interdependent-valued eXchange economies (MATRIX) that satisfies three very desirable properties as mentioned in the previous part, suitably modified for the dynamic setting. We show that certain naïve mechanism do not work in this setting, which makes both the problem and its solution non-trivial.
- We compute the complexity of MATRIX and show it to be equal to that of the mechanism which is applicable to independent valuation setting.

Chapter 4: Resource Critical Task Execution via Crowdsourcing

The second major contribution of this thesis is on the aggregation of information in resource critical crowdsourcing contests. We consider one potential manipulation that is unique to crowdsourcing setup and provide solution and discuss about how the flow of information from

the crowd can be efficiently harnessed by a proper incentive scheme. The details of the chapter is as follows.

- In the first and the major part of this chapter, we consider crowdsourcing contests, where several individuals are either trying to locate an object or finding an answer to a query or solving a computationally hard problem, and the fastest correct response gives them a reward. We use the DARPA red balloon challenge as one of the motivations for this work.
- We define the *viral* and *atomic* tasks, which clearly distinguishes the domain for our work. We consider crowdsourcing tasks that fall into the atomic task category.
- For the atomic crowdsourcing tasks, we consider three important design criteria: *collusion-proofness*, *dominant strategy implementability*, and *resource criticality*. To the best of our knowledge, these design criteria has not been addressed in the context of crowdsourcing in the literature.
- Next we introduce the model and notation to analyze the crowdsourcing contest. We also define a set of desirable properties in this setting. The first property is the *downstream sybilproofness*, which ensures that the agents connected over a network cannot gain more by creating fake identities. The second property ensures that the *budget* is balanced. *Contribution rationality* ensures that every positive contributor in the contest gets positive reward. The last property, *collapse-proofness* ensures that the agents do not collude and report as a single node.
- Even though the properties are plausible, we show an impossibility result that claims that a subset of these properties are not simultaneously satisfiable.
- However, if we consider the approximate versions of certain properties, we show that the space of mechanisms satisfying other pure properties and one approximate property is non-empty. This is a positive result and we characterize the space of those mechanisms.
- Given that it is possible to design mechanisms, the next question is which mechanism to pick. The answer depends on which kind of applications we are looking at. We consider two kinds of tasks in this chapter: *cost critical* and *time critical* tasks.
- In both the classes of tasks, we characterize the mechanisms that satisfy certain set of properties, one of them is approximate and others pure.

- We show that the winning solution of the DARPA red balloon challenge becomes a special case of the cost critical class of this characterization result.
- We initiate a brief discussion about how the crowdsourcing competitions can be made more efficient by giving partial rewards for partial information given by the agents. This is a work in progress, and we discuss the plans how we can exploit a synergy between the participants with the proper incenting mechanism.

Chapter 5: Efficient Team Formation using a Strategic Crowd

This chapter deals with the question of maximizing the productive output of the crowdsourcing network. We model the crowd workers who are connected over a social network as an organization, and the designer of the crowdsourcing system aims to maximize the net productive output of the system. Therefore the design of the incentive scheme and the network is of utmost importance for crowdsourcing as it is important for any connected organization. The approach we follow has two complementary facets, described as follows.

- In the first part of the chapter, we assume that the network is given to the designer, and the goal is to design proper incentive shares to maximize output.
- To understand how a rational and intelligent inter-networked crowd can behave, we conduct a game theoretic analysis of the network. We assume that each agent can spend his/her bounded effort either in *production* or in *communication*. The production effort gives direct benefit to the organization, while the communication effort indirectly influences the output of the system by making other individuals more productive. If Alice spends her time communicating with Bob, then we call Alice an *influencer* and Bob an *influencee*.
- We begin with a hierarchical model of influencer and influencee, and propose a utility model for the individuals capturing the fact of production and communication.
- We show the *existence* and *uniqueness* of Nash equilibrium effort levels. For hierarchical networks, the computational complexity of computing the Nash equilibrium is not hard.
- Next we define the notion of *social output* appropriately to capture the net outcome of the system. The natural question that follows is how the equilibrium efforts affect the social output.

- We show that the *reward sharing* (a part of the utility model) influences the equilibrium effort and thereby the social outcome. We provide the recipe of finding the right reward share that maximizes the social output, and then compare that with an optimal social output of the network. We use the concept of *price of anarchy* (PoA) in order to capture this and provide bounds on the PoA for stylized networks.
- The results of the existence and uniqueness of the Nash equilibrium generalizes to a general influencer-infleece network. We show that there is a connection of the uniqueness result to the matrix stability criterion, which connects the equilibrium efforts with the structure of the network.
- In the second part of the chapter, we consider the complementary problem of designing the network for a given reward sharing scheme.
- We introduce a somewhat different model in order to analyze this problem. The arrival and execution of the tasks are assumed to be Poisson. We assume the agent model and their utility model in this modified setting. The design goal in this setting is to minimize the *risk* of not being able to consume the incoming rate of tasks by the entire network.
- For ease of exposition, we again consider hierarchies. We show that the hierarchy gives rise to a drawback, namely *free-riding* of well positioned nodes, and a benefit of *information sharing*.
- We show that there exists a trade-off between the two, and a network design that takes care of both effects yields better performance by reducing the overall *risk* of the system. We also provide an approximation algorithm to find an optimal hierarchy in this setting.

Chapter 6: Conclusions

The contributions of the thesis is summarized and the conclusions that can be drawn from this work is presented in this chapter. We also provide some interesting future directions of this work.

Chapter 2

Game Theory and Mechanism Design: A Quick Review

Crowdsourcing requires a designer and a (possibly large) number of participants who can execute tasks, provide content, generate ideas, or deliver services, etc. In this thesis, we address the crowdsourcing problem with *rational* and *intelligent* participants. A rational agent always aims to maximize a well defined payoff function. Intelligence implies that the agent understands the rules of a game well enough and is capable of taking an optimal decision given the game. Microeconomics offers two appropriate tools under this scenario, namely *game theory* and *mechanism design*, in order to analyze, predict, and design outcomes of strategic interaction between decision making agents. In this chapter, we will take a quick look at some basic concepts of game theory and mechanism design that are relevant for presenting the material in this thesis.

Here we consolidate the concepts and definitions from a number of books [46, 76, 51, 64, 53, 47]. However, we primarily follow the notation of the book by Narahari [52]. In the places where we use new notations and definitions, we explain them in the respective chapters. We first look into the concepts of game theory followed by that of mechanism design. For brevity, we restrict the discussions to the extent that is necessary to present the material of the subsequent chapters of this thesis.

2.1 Game Theory

Game theory, often referred to as the science of strategic interaction, can be defined as the formal study of the interaction of decision making agents who are *rational* and *intelligent*. As defined earlier in this chapter, rationality means that an agent always aims to maximize her own payoff, and intelligence ensures that the agent has enough knowledge of the game to pick

the right action. The theory of games came into public attention after the monumental work of von Neumann and Morgenstern [83]. Following the contributions of this book, there had been many pioneering works that have contributed to developing game theory as the science of economics.

For the discussions of this chapter and the rest of the thesis, we will restrict our attention to non-cooperative games and their strategic form representation. An interested reader can look up standard references [52, 51, 76, 47] for the other types of games and their representations.

Let us start with an example game called the *Prisoner’s Dilemma* to introduce the concepts. We will set up the notation to make the description more formal afterwards.

Suppose two prisoners, Anthony and Monica, are caught under the suspicion of doing a common mischief and now they are locked up in two separate cells in the police station and are interrogated. In order to get the true information from them, let us suppose that the police officer tells Anthony: “*You can either confess your crime or not confess. If both of you confess, then the crime is proved and both of you are going to be charged with 5 years of prison term. If both of you do not confess, then I can still get you charged for 2 years jail term each. But if your partner confesses and you do not, due to her honesty, I will let her go free, but charge you with 10 years in jail.*”

Now Anthony knows that the same thing has been told to Monica as well, and they cannot communicate with each other. This is a classic example of the non-cooperative game. Let us consider the payoff of the prisoners as being the negative of the number of years in jail, so that their goal is to maximize this number which in turn minimizes their stay in prison. A compact way of representing this is by writing down the payoff matrix given by Table 2.1, where the numbers at the bottom left of each cell denote the payoffs of the row player (Anthony) and the ones at the top right of each cell denote the payoffs to the column player (Monica).

		<i>Monica</i>	
		Confess	Not Confess
<i>Anthony</i>	Confess	-5 -5	-10 0
	Not Confess	0 -10	-2 -2

Table 2.1: The prisoner’s dilemma game.

Now from Anthony’s perspective, it is easy to see that *confessing* gives strictly higher payoff for both the actions (confess or not confess) taken by Monica. Hence, any rational and intelligent agent would choose confess as his strategy when he is not able to communicate with the other

player. Similar is the case for Monica as this is a symmetric game. So, a rational play by both players lead to the outcome (confess, confess), which leads to a 5 years of prison term for both of them. However, they could have done much better by not confessing together and saving 3 years each in terms of staying in prison. But that is the dilemma, since in a non-cooperative setting it is hard to reason for the action of the other player, personal greed can lead to a situation which is collectively worse. We are now in a position to develop the notation to explain this game and the games to follow.

Strategic Form Game. A *strategic* or *normal* form game is a representation of a strategic interaction that gives us a mathematical language to express the game precisely. Let us denote the set of the players by $N = \{1, 2, \dots, n\}$, the strategy set of agent i by S_i , and the utility function by u_i , which are used in the formal definition of the strategic form games given below.

Definition 2.1 A *strategic form game* Γ is a tuple $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$, where,

- ▶ $N = \{1, 2, \dots, n\}$ is the set of players;
- ▶ S_1, S_2, \dots, S_n are the strategy sets of the players $1, 2, \dots, n$ respectively; and
- ▶ $u_i : S_1 \times S_2 \times \dots \times S_n \rightarrow \mathbb{R}$, for $i = 1, 2, \dots, n$ are mappings called the *payoff functions*.

In the prisoner's dilemma example, there are two players, i.e., $N = \{1, 2\}$, the strategy set of each agent is given by $S_1 = S_2 = \{\text{confess}, \text{not confess}\}$, and the utilities, given by Table 2.1, are as follows.

$$\begin{array}{llll}
 u_1(\text{confess}, \text{confess}) & = & -5; & u_2(\text{confess}, \text{confess}) & = & -5 \\
 u_1(\text{confess}, \text{not confess}) & = & 0; & u_2(\text{confess}, \text{not confess}) & = & -10 \\
 u_1(\text{not confess}, \text{confess}) & = & -10; & u_2(\text{not confess}, \text{confess}) & = & 0 \\
 u_1(\text{not confess}, \text{not confess}) & = & -2; & u_2(\text{not confess}, \text{not confess}) & = & -2
 \end{array}$$

We call the strategy *confess* to be a “strongly dominant” strategy since the payoff at this strategy is strictly more than that of all other strategies of the players for any chosen strategies of the other players. Let us define it formally.

Definition 2.2 (Strongly Dominant Strategy) A strategy $s_i^* \in S_i$ is said to be a *strongly dominant strategy* for player i if the payoff is strictly greater than all other strategy choices of the player for every strategy profile $s_{-i} \in S_{-i}$ of the other agents. That is, $\forall s_i \neq s_i^*$,

$$u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}), \forall s_{-i} \in S_{-i}.$$

Definition 2.3 (Strongly Dominant Strategy Equilibrium) A strategy profile given by $s^* = (s_1^*, \dots, s_n^*)$ is said to be a strongly dominant strategy equilibrium of the game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ if for every player i , s_i^* be the strongly dominant strategy.

It is clear from these definitions that *(confess, confess)* is a strongly dominant strategy equilibrium (SDSE) of the prisoner’s dilemma game. However, this equilibrium is quite restrictive, and may not exist in all games. For example, let us consider the same prisoner’s dilemma game with the payoff matrix given by Table 2.2. This game no longer has a SDSE, however, it has a weaker type of equilibrium called the weakly dominant strategy equilibrium (WDSE), defined as follows.

		<i>Monica</i>	
		Confess	Not Confess
<i>Anthony</i>	Confess	-5	-10
	Not Confess	-5	0
		0	0
		-10	0

Table 2.2: The modified prisoner’s dilemma game.

Definition 2.4 (Weakly Dominant Strategy) A strategy $s_i^* \in S_i$ is said to be a weakly dominant strategy for player i if the payoff is no less than all other strategy choices of the player for every strategy profile $s_{-i} \in S_{-i}$ of the other agents, and is strictly better for at least one strategy profile $\tilde{s}_{-i} \in S_{-i}$ of the other agents. That is, $\forall s_i \neq s_i^*$,

$$u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i}), \quad \forall s_{-i} \in S_{-i},$$

$$u_i(s_i^*, \tilde{s}_{-i}) > u_i(s_i, \tilde{s}_{-i}), \quad \text{for some } \tilde{s}_{-i} \in S_{-i}.$$

Definition 2.5 (Weakly Dominant Strategy Equilibrium) A strategy profile given by $s^* = (s_1^*, \dots, s_n^*)$ is said to be a weakly dominant strategy equilibrium of the game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ if for every player i , s_i^* be the weakly dominant strategy.

Even though the strategy profile *(confess, confess)* turns out to be the WDSE of the modified prisoner’s dilemma game given by Table 2.2, WDSE is also not guaranteed to exist in all games. For example, let us consider the Battle of Sexes game. Suppose a couple wants to go for a movie, where the husband likes action movies more than romantic movies, and the wife likes them the other way. To put the preferences into numbers, let us assume the payoff for the husband is

double than that of the wife if both decide to go for an action movie. The opposite happens when they both go for a romantic movie. But if they decide to go to different movies, both get a zero payoff as they value their ‘togetherness’. The situation is captured by Table 2.3.

		<i>Wife</i>	
	<i>Husband</i>	Action	Romantic
Action		1	0
	2	0	0
Romantic		0	2
	0	1	1

Table 2.3: The battle of sexes game.

It is clear that this game does not have either a SDSE or a WDSE. However, if we consider the strategy profile $(action, action)$, each of the agents get smaller payoff if they unilaterally deviate from this profile when the other agent sticks to it. The complementary situation happens for the strategy profile $(romantic, romantic)$. These strategy profiles are called *pure strategy Nash equilibria* (PSNE), named after one of the founding figures of game theory, John F. Nash. This is defined formally as follows.

Definition 2.6 (Pure Strategy Nash Equilibrium) *A strategy profile $s^* = (s_1^*, \dots, s_n^*)$ is said to be a pure strategy Nash equilibrium of the game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ if,*

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \quad \forall s_i \in S_i, i \in N.$$

Even though PSNE looks a nicer concept for a strategic form game, this is also not guaranteed to exist. To illustrate this point, we refer to the Matching Pennies game (see Table 2.4) where two players can simultaneously choose either *heads* or *tails*. This game does not have a PSNE. But if the agents are allowed to make a probabilistic mix of their pure strategies, e.g., say the row player decides to play H or T with probability 0.5 each, we are led to a scenario of mixed strategies, which we define formally as follows.

Definition 2.7 (Mixed Strategy) *Given a player i with S_i as the set of pure strategies, a mixed strategy σ_i of player i is a probability distribution over S_i . That is, $\sigma_i : S_i \rightarrow [0, 1]$ assigns to each pure strategy $s_i \in S_i$, a probability $\sigma_i(s_i)$ such that,*

$$\sum_{s_i \in S_i} \sigma_i(s_i) = 1.$$

		<i>Mismatcher</i>	
		Heads	Tails
	<i>Matcher</i>		
Heads		-1	1
		1	-1
Tails		-1	1

Table 2.4: The matching pennies game.

A pure strategy of a player, say $s_i \in S_i$, can be considered as a mixed strategy that assigns probability 1 to s_i and probability 0 to all other strategies of player i . Such a mixed strategy is called a *degenerate mixed strategy*. The mixed strategy σ_i for agent i is picked from the *mixed extension* of the pure strategy space S_i . We assume S_i to be finite for the discussions in this chapter. The *mixed extension* of S_i is denoted by $\Delta(S_i)$ and defined as follows.

$$\Delta(S_i) = \left\{ \sigma \in \mathbb{R}_+^{|S_i|} : \mathbf{1}^\top \sigma = 1 \right\}.$$

Following the same spirit of the PSNE (Definition 2.6), we are now in a position to define a *mixed strategy Nash equilibrium* (MSNE).

Definition 2.8 (Mixed Strategy Nash Equilibrium) *A mixed strategy profile given by $\sigma^* = (\sigma_1^*, \dots, \sigma_n^*)$ is said to be a mixed strategy Nash equilibrium of the game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ if,*

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*), \quad \forall \sigma_i \in \Delta(S_i), i \in N.$$

It can be shown that for the matching pennies game (Table 2.4), the MSNE is given by (0.5, 0.5) for each player. The question of existence of a MSNE for a finite game (both the number of players and their strategy sets are finite) was answered by Nash [54], which we reproduce here.

Theorem 2.1 (Nash [54]) *Every finite strategic form game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ has at least one mixed strategy Nash equilibrium.*

The review of the game theory concepts presented in this section would enable the reader to follow the discussions and the definitions in the chapters to come. For more detailed description of the concepts and definitions, one can refer to standard texts [52, 51, 76, 47, e.g.]. Let us now move on to mechanism design theory and review the basic concepts.

2.2 Mechanism Design

While game theory allows us to predict the outcome of a given game, mechanism design equips us with techniques to design institutions or protocols that satisfy certain desired objectives. Due to this reason, mechanism design is often called the *reverse engineering* of game theory. The assumption of rationality and intelligence of the agents still holds, and the question that mechanism design addresses is: “*how to design the rules of a game so that in the equilibrium of the game the desired properties are satisfied?*” The strategic agents are expected to play in an equilibrium and that helps the designer to satisfy society’s goals through mechanism design. However, there are certain inherent difficulties in the theory of mechanism design, given by certain impossibility theorems, which says that not all desirable properties can be simultaneously satisfied. The goal, therefore, is to understand the limits of satisfiability and design the best mechanism given a specific setting.

Let us introduce mechanism design using the well known story of a mother of two kids who has to design a mechanism to make her two kids share a cake equally (see Figure 2.1). The mother is the mechanism designer in this case. If the mother slices the cake into equal pieces and distributes one piece to each of the kids, the solution is not necessarily acceptable to the kids because each kid may be left with the perception that he/she got the smaller of the two pieces. On the other hand, consider the following mechanism: (1) One of the kids would slice the cake into two pieces and (2) the other kid would pick up one of the pieces, leaving the remaining piece to the kid who sliced the cake into two pieces. This mechanism implements the desirable outcome of the kids sharing the cake equally and further each kid has no reason to complain about this mechanism. The first kid would divide the cake exactly in half according to his notion of equality, since he knows if he divides unequally, the larger piece is going to go to the other kid. Hence he is indifferent between the pieces. The other kid would also be happy since she gets the opportunity to choose. Employing this simple mechanism, the mother is able to achieve fair division, even though she did not know the preferences of the kids initially.

There are two things to notice in this example. First, even though the mother wants to achieve a fair division, she does not know what the ‘fair’ division is, because the cake is observed differently by the two kids. Secondly, the true ‘fair’ piece is private observation of the kids (who are the players in this context). As we introduce the general setup of mechanism design in the following section, we carry in mind that all mechanism design problem has these two fundamental traits, namely the private information held by the agents and that the designer’s objective is to achieve a goal under that partial knowledge. In other words, mechanism design is an optimization problem where not all the parameters of the problem are known.

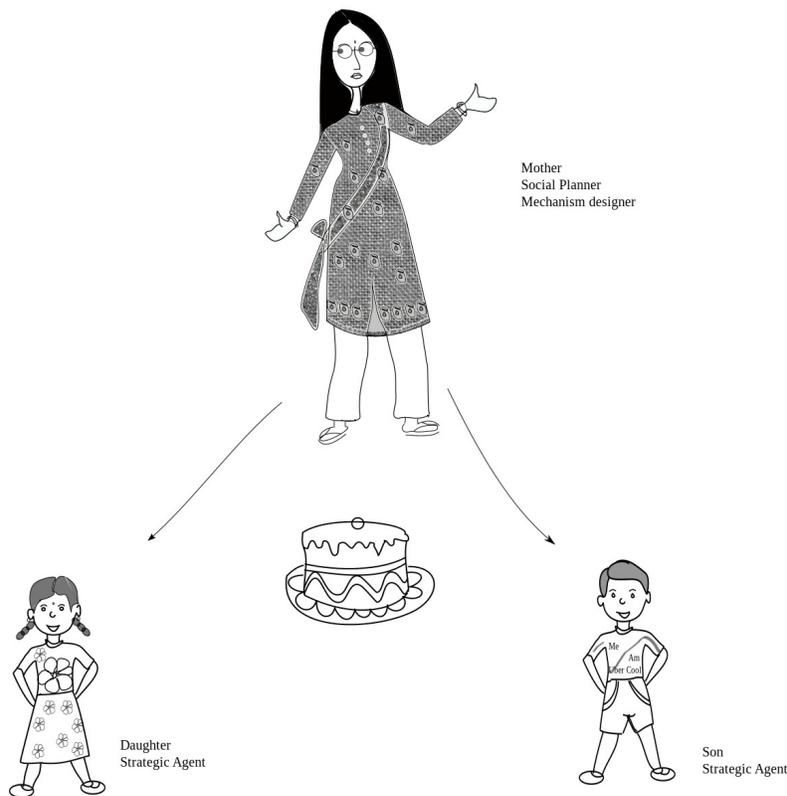


Figure 2.1: The fair division problem.

2.2.1 Mechanism Design Environment

The following provides a general setting for formulating, analyzing, and solving mechanism design problems.

- There are n agents, $1, 2, \dots, n$, with $N = \{1, 2, \dots, n\}$. The agents are rational and intelligent, and interact strategically among themselves towards making a collective decision.
- X is a set of *alternatives* or *outcomes*. The agents are required to make a collective choice from the set X . The outcome set is determined by the particular choice of model. For instance, a critical choice is whether or not monetary compensation is allowed (voting vs. exchange).

- Prior to making the collective choice, each agent privately observes his preferences over the alternatives in X . This is modeled by supposing that agent i privately observes a parameter or signal θ_i that determines his preferences. Agent i knows the value of θ_i perfectly and the other agents may not know it. θ_i is called a private value or type of agent i .
- The preferences could be either *cardinal* (it is possible to quantitatively say how much one alternative is preferred over another) or *ordinal* (only the relative preference is available, not the quantitative measure). Clearly, the cardinal preference is a smaller subset of the ordinal preferences. We will consider ordinal preferences for the voting examples, but cardinal preferences are discussed in the major part of this chapter.
- We denote by Θ_i the set of private values of agent i , $i = 1, 2, \dots, n$. The set of all type profiles is given by $\Theta = \Theta_1 \times \dots \times \Theta_n$. A typical type profile is represented as $\theta = (\theta_1, \dots, \theta_n)$.
- It is assumed that there is a common prior distribution $P \in \Delta(\Theta)$. To maintain consistency of beliefs, individual belief functions $p_i : \Theta \rightarrow \Delta(\Theta_i)$ that describe the beliefs that player i has about the type profiles of the rest of the players can all be derived from the common prior.
- Individual agents have preferences over outcomes that are represented by a utility function $u_i : X \times \Theta_i \rightarrow \mathbb{R}$. Given $x \in X$ and $\theta_i \in \Theta_i$, the value $u_i(x, \theta_i)$ denotes the payoff that agent i having type $\theta_i \in \Theta_i$ receives from a decision $x \in X$. In the more general case, u_i depends not only on the outcome and the type of player i , but could depend on the types of the other players as well, and so $u_i : X \times \Theta \rightarrow \mathbb{R}$. We restrict our attention to the former in this chapter for simplicity.
- The set of outcomes X , the set of players N , the type sets Θ_i ($i = 1, \dots, n$), the common prior distribution $P \in \Delta(\Theta)$, and the payoff functions u_i ($i = 1, \dots, n$) are assumed to be *common knowledge* among all the players. The specific value θ_i observed by agent i is private information of agent i .

Social Choice Functions

Since the preferences of the agents depend on the realization of their types $\theta = (\theta_1, \dots, \theta_n)$, it is logical and natural to make the collective decision to depend on θ . This leads to the definition of a social choice function.

Definition 2.9 (Social Choice Function) Suppose $N = \{1, 2, \dots, n\}$ is a set of agents with the type sets $\Theta_1, \Theta_2, \dots, \Theta_n$ respectively. Given a set of outcomes X , a social choice function is a mapping

$$f : \Theta_1 \times \dots \times \Theta_n \rightarrow X$$

that assigns to each possible type profile $(\theta_1, \theta_2, \dots, \theta_n)$, a collective choice from the set of alternatives.

Preference Elicitation Problem

Consider a social choice function $f : \Theta_1 \times \dots \times \Theta_n \rightarrow X$. The types $\theta_1, \dots, \theta_n$ of the individual agents are private information of the agents. Hence for the social choice $f(\theta_1, \dots, \theta_n)$ to be chosen when the individual types are $\theta_1, \dots, \theta_n$, each agent must disclose its true type to the social planner. However, given a social choice function f , a given agent may not find it in its best interest to reveal this information truthfully. This is called the *preference elicitation* problem or the *information revelation* problem.

Preference Aggregation Problem

Once all the agents report their types, the profile of reported types has to be transformed to an outcome, based on the social choice function. Let θ_i be the true type and $\hat{\theta}_i$ the reported type of agent i ($i = 1, \dots, n$). The process of computing $f(\hat{\theta}_1, \dots, \hat{\theta}_n)$ is called the *preference aggregation* problem.

Figure 2.2 provides a pictorial representation of all the elements making up a mechanism design environment.

2.2.2 Direct and Indirect Mechanisms

One can view mechanism design as the process of solving an incompletely specified optimization problem where the specification is first elicited and then the underlying optimization problem is solved. Specification elicitation is basically the preference elicitation or type elicitation problem. To elicit the type information from the agents in a truthful way, there are broadly two kinds of approaches, which are aptly called *direct mechanisms* and *indirect mechanisms*. We define these below. In these definitions, we assume that the set of agents N , the set of outcomes X , the sets of types $\Theta_1, \dots, \Theta_n$, a common prior $P \in \Delta(\Theta)$, and the utility functions $u_i : X \times \Theta_i \rightarrow \mathbb{R}$ are given and are common knowledge.

Definition 2.10 (Direct Mechanism) Given a social choice function $f : \Theta_1 \times \Theta_2 \times \dots \times \Theta_n \rightarrow X$, a direct mechanism (also called a direct revelation mechanism) consists of the tuple $(\Theta_1, \Theta_2, \dots, \Theta_n, f(\cdot))$.

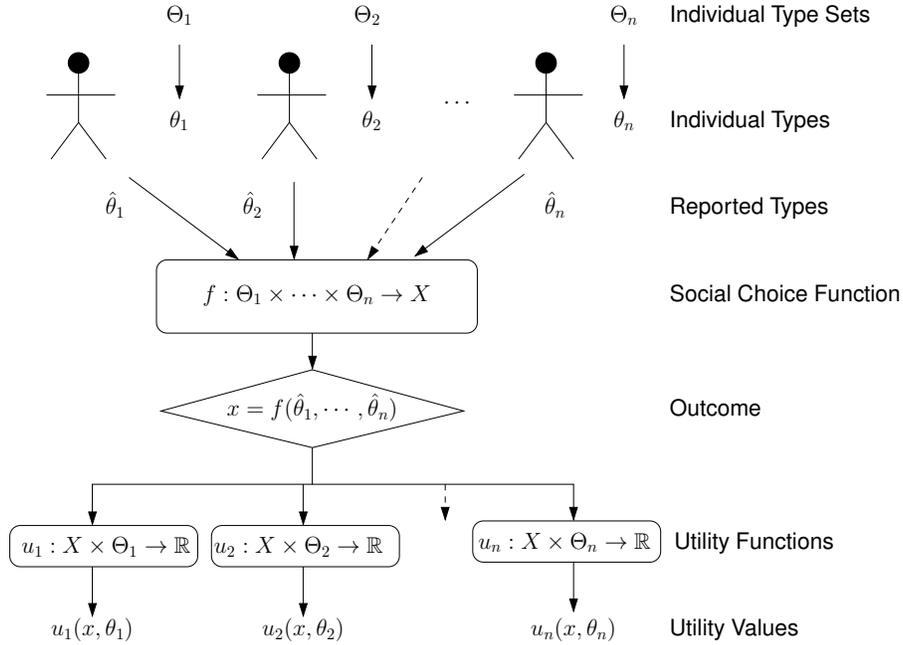


Figure 2.2: The general mechanism design environment.

The idea of a direct mechanism is to *directly* seek the type information from the agents by asking them to reveal their true types.

Definition 2.11 (Indirect Mechanism) *An indirect mechanism (also called an indirect revelation mechanism) consists of a tuple $(S_1, S_2, \dots, S_n, g(\cdot))$ where S_i is a set of possible actions for agent i ($i = 1, 2, \dots, n$) and $g : S_1 \times S_2 \times \dots \times S_n \rightarrow X$ is a function that maps each action profile to an outcome.*

The idea of an indirect mechanism is to provide a choice of actions to each agent and specify an outcome for each action profile. This induces a game among the players and the strategies played by the agents in an equilibrium of this game will indirectly reflect their original types. The idea behind and the difference between direct mechanisms and indirect mechanisms will become clear in the next section when we discuss implementation of social choice functions by mechanisms.

Note that a direct mechanism corresponding to a social choice function $f : \Theta \rightarrow X$ is a special case of an indirect mechanism $(S_1, S_2, \dots, S_n, g(\cdot))$ with $S_i = \Theta_i, \forall i \in N$ and $g = f$.

2.2.3 Implementation of Social Choice Functions

The social choice function (SCF) captures the goal of the mechanism designer. However, to make the mechanism work in practice, we need to implement the SCF either directly or in-

directly. Since, direct implementation is a special case of the indirect implementation, in this section, we focus on indirect implementation. We will see that the indirect mechanism induces a Bayesian game among the players, and implementation is concerned with the outcome at the equilibria of that Bayesian game. Later in this section, we discuss about the direct and indirect implementability and show that they are equivalent.

2.2.4 Bayesian Game Induced by a Mechanism

From Section 2.2.1, we see that an indirect mechanism $M = (S_1, S_2, \dots, S_n, g(\cdot))$ endows the players with the strategy spaces $S_i, i \in N$, and an aggregation function $g(\cdot)$, that maps the strategies chosen by the players into alternatives. Together with the common prior P that gives rise to the individual beliefs p_i , this indirect mechanism framework induces the Bayesian game $\Gamma := \langle N, (\Theta_i)_{i \in N}, (S_i)_{i \in N}, (p_i)_{i \in N}, (U_i)_{i \in N} \rangle$. The strategies of individual agents here are the mappings, $s_i : \Theta_i \rightarrow S_i, i \in N$. Thus, given a type $\theta_i \in \Theta_i$, $s_i(\theta_i)$ will give the action of player i corresponding to θ_i . The strategy $s_i(\cdot)$ will specify actions corresponding to types. In the auction scenario, the bid b_i of player i is a function of the valuation θ_i . For example, $b_i(\theta_i) = \alpha_i \theta_i$ is a particular strategy for player i . The utility function U_i in the induced Bayesian game is related to the original utility function u_i as follows.

$$U_i(s_1(\theta_1), \dots, s_n(\theta_n)) = u_i(g(s_1(\theta_1), \dots, s_n(\theta_n)), \theta_i), \forall i \in N.$$

Figure 2.3 captures the idea behind an indirect mechanism and the Bayesian game that is induced by an indirect mechanism.

Note that a direct mechanism corresponding to a social choice function $f(\cdot)$ is a special case of an indirect mechanism with the strategy sets same as the type sets and the outcome rule $g(\cdot)$ same as the social choice function $f(\cdot)$. It goes without saying that a Bayesian game is induced by a direct mechanism as well.

2.2.5 Implementation of a Social Choice Function by a Mechanism

Let us now define the implementation of an SCF by a mechanism.

Definition 2.12 (Implementation of an SCF) *We say a mechanism $M = \langle (S_i)_{i \in N}, g(\cdot) \rangle$ where $g : S_1 \times \dots \times S_n \rightarrow X$, implements the social choice function $f(\cdot)$ if there is a pure strategy equilibrium $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ of the Bayesian game Γ induced by M such that,*

$$g(s_1^*(\theta_1), \dots, s_n^*(\theta_n)) = f(\theta_1, \dots, \theta_n), \forall (\theta_1, \dots, \theta_n) \in \Theta.$$

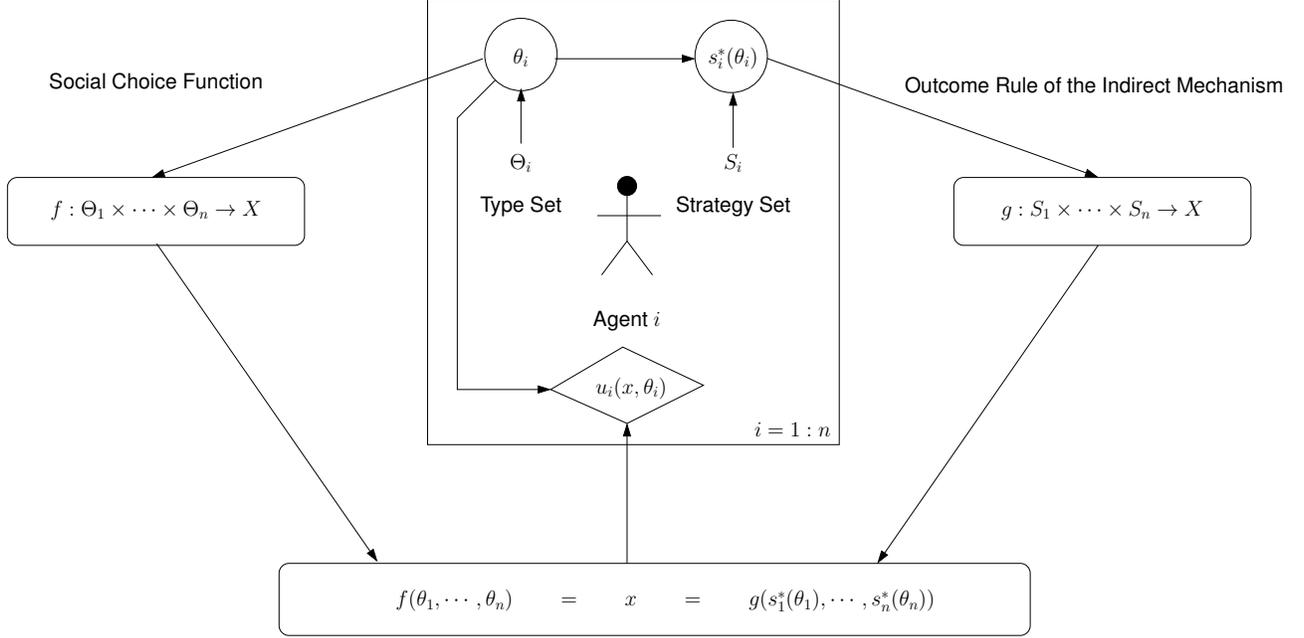


Figure 2.3: The indirect mechanism design environment and its implications.

The implementation is called dominant strategy or Bayesian Nash implementation depending on whether the pure strategy equilibrium mentioned in the definition above is a dominant strategy one or a Bayesian Nash one. In a Bayesian game, the dominant strategy or Bayesian Nash equilibria are defined as follows.

Definition 2.13 (Dominant Strategy Equilibrium in a Bayesian Game) *Given a Bayesian game Γ , a strategy profile $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ is called a dominant strategy equilibrium if $\forall a_i \in S_i, \forall s_{-i} : \Theta_{-i} \rightarrow S_{-i}, \forall \theta_i \in \Theta_i, \forall i \in N$,*

$$\mathbb{E}_{\theta_{-i}} U_i(s_i^*(\theta_i), s_{-i}(\theta_{-i})) \geq \mathbb{E}_{\theta_{-i}} U_i(a_i, s_{-i}(\theta_{-i})).$$

Definition 2.14 (Bayesian Nash Equilibrium in a Bayesian Game) *Given a Bayesian game Γ , a strategy profile $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ is called a Bayesian Nash equilibrium if $\forall a_i \in S_i, \forall \theta_i \in \Theta_i, \forall i \in N$,*

$$\mathbb{E}_{\theta_{-i}} U_i(s_i^*(\theta_i), s_{-i}^*(\theta_{-i})) \geq \mathbb{E}_{\theta_{-i}} U_i(a_i, s_{-i}^*(\theta_{-i})).$$

Of particular interest is the direct mechanism, $D = \langle (\Theta_i)_{i \in N}, f(\cdot) \rangle$. The following two incentive compatibility properties are defined on the direct mechanism.

Definition 2.15 (Dominant Strategy Incentive Compatibility (DSIC)) *A social choice function $f : \Theta_1 \times \Theta_2 \times \dots \times \Theta_n \rightarrow X$ is said to be dominant strategy incentive compatible (or truthfully implementable in dominant strategies) if the direct revelation mechanism $D = \langle (\Theta_i)_{i \in N}, f(\cdot) \rangle$ has a dominant strategy equilibrium $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ in which $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, i \in N$.*

Similarly an SCF is Bayesian incentive compatible when the equilibrium is a Bayesian Nash equilibrium.

Definition 2.16 (Bayesian Incentive Compatibility (BIC)) *A social choice function $f : \Theta_1 \times \Theta_2 \times \dots \times \Theta_n \rightarrow X$ is said to be dominant strategy incentive compatible (or truthfully implementable in dominant strategies) if the direct revelation mechanism $D = \langle (\Theta_i)_{i \in N}, f(\cdot) \rangle$ has a Bayesian Nash equilibrium $s^*(\cdot) = (s_1^*(\cdot), \dots, s_n^*(\cdot))$ in which $s_i^*(\theta_i) = \theta_i, \forall \theta_i \in \Theta_i, i \in N$.*

A natural question here is why we are interested in the direct revelation mechanism D . This is because of the revelation principle given by the following two theorems, which we mention without proof. This principle essentially says that for each social choice function that is implementable in indirect mechanism is also implementable by a direct mechanism. Therefore, it is enough to consider only direct mechanisms and properties like incentive compatibility.

Theorem 2.2 (Revelation Principle for Dominant Strategy Implementation) *Suppose that there exists a mechanism $M = \langle (S_i)_{i \in N}, g(\cdot) \rangle$ that implements the social choice function $f(\cdot)$ in dominant strategy equilibrium. Then $f(\cdot)$ is dominant strategy incentive compatible.*

Theorem 2.3 (Revelation Principle for Bayesian Nash Implementation) *Suppose that there exists a mechanism $M = \langle (S_i)_{i \in N}, g(\cdot) \rangle$ that implements the social choice function $f(\cdot)$ in Bayesian Nash equilibrium. Then $f(\cdot)$ is Bayesian incentive compatible.*

An inquisitive reader is referred to [52] for a proof of these theorems.

2.2.6 A Limit to the Desire: The Gibbard-Satterthwaite Impossibility Theorem

Ideally, a mechanism designer would like the SCF to satisfy a number of “good” properties. However, mechanism design theory deals with a lot of results that put limits on those desirable properties. One such result, named after Gibbard and Satterthwaite, says us that the following set of desirable properties cannot be satisfied together. The Gibbard-Satterthwaite (G-S) Impossibility Theorem is more general and works with even ordinal preferences on the alternative

set X . What we consider in this chapter and in the rest of the thesis are cardinal preferences, i.e., an agent i prefers alternative a over b , $a, b \in X$, in accordance with a utility function u_i , that is, $u_i(a, \theta_i) \geq u_i(b, \theta_i)$. It may not always be the case that there exists a well defined utility function that leads to the preferences. Those kind of preferences are called ordinal preferences, where the agents can have any preference ordering P_i which is a *strict preference ordering* over the alternative set X . Since the G-S theorem applies to the more general ordinal preferences, it will continue to hold even for the cardinal preferences. However, for a more general exposition, we present the general version of the theorem. Before going into the theorem, some definitions are as follows. For this section, we follow the notation and definitions of [74].

Strict preference ordering A preference ordering P is called *strict* if it is a binary relation on X satisfying,

- ▶ **Completeness:** for all $a, b \in X$, either aPb or bPa or both.
- ▶ **Reflexivity:** for all $a \in X$, aPa .
- ▶ **Transitivity:** if aPb and bPc , then aPc .
- ▶ **Anti-symmetry:** for all $a, b \in X$, if aPb and bPa , then $a = b$.

Hence, for all $a, b \in X$, and a strict preference order P_i of agent i , $aP_i b$ is interpreted as “ a is strictly preferred to b under P_i ”.

Let \mathbb{P} be the set of all linear orderings over X (there are $|X|!$ such orders assuming $|X| < \infty$). A preference profile $P = (P_1, \dots, P_n) \in \mathbb{P}^n$ is an n -list of orderings, one for each voter. An SCF f is a mapping $f : \mathbb{P}^n \rightarrow X$. The strategy-proofness (incentive compatibility) property introduced in the previous section is restated below for this environment.

Definition 2.17 *The SCF f is manipulable if there exists a voter i , a profile $P \in \mathbb{P}^N$ and an ordering P'_i such that,*

$$f(P'_i, P_{-i}) P_i f(P_i, P_{-i})$$

Definition 2.18 *The SCF f is strategy-proof if it is not manipulable.*

One class of SCFs which is always strategy-proof is the constant SCF which selects the same alternative at all profiles. In order to rule out this possibility, it will be assumed that SCFs under consideration satisfy the property of *unanimity*.

For all voters i and $P_i \in \mathbb{P}$, let $\tau(P_i)$ denote the maximal element in X according to P_i .

Definition 2.19 *The SCF f satisfies unanimity if $f(P) = a$ whenever $\tau(P_i) = a$ for all $i \in N$.*

Definition 2.20 *The voting rule f is dictatorial if there exists $d \in N$ such that for all $P \in \mathbb{P}^n$, $f(P) = \tau(P_d)$.*

Theorem 2.4 (Gibbard [31], Satterthwaite [73]) *Assume $|X| \geq 3$. If f satisfies unanimity then it is strategy-proof if and only if it is dictatorial.*

This is one of the classic results in mechanism design theory. For a direct proof of this theorem, we refer the reader to [75].

A natural way of evading the negative conclusions of the G-S theorem is to assume that admissible preferences are subject to certain restrictions. In the following section, we focus on the *Quasi-linear Domain*, which is quite popular and appropriate for the discussions of this thesis.

2.2.7 Domain Restriction: Quasi-Linear Utilities

These are models where monetary compensation is feasible. Moreover money enters the utility function in an additively separable way.

Let A be the set of alternatives. Agent i 's type is $\theta_i \in \Theta_i$ determines her valuation for every $a \in A$ according to the function $v_i : A \times \Theta_i \rightarrow \mathbb{R}$, i.e. $v_i(a, \theta_i)$ is the valuation of alternative a when her type is θ_i . The agent may also receives a monetary payment $t_i \in \mathbb{R}$. The outcome x is therefore given by the tuple $(a, \mathbf{t}) \in X = A \times \mathbb{R}^n$. The overall utility of the agent is given by,

$$u_i((a, \mathbf{t}), \theta_i) = v_i(a, \theta_i) + t_i.$$

We re-define the earlier notions in this environment.

Definition 2.21 *An SCF is a mapping $f : \Theta_1 \times \dots \times \Theta_n \rightarrow A$.*

Definition 2.22 *A transfer scheme is a collection of mappings $t \equiv (t_1, \dots, t_n)$ where $t_i : \Theta_1 \times \dots \times \Theta_n \rightarrow \mathbb{R}$ for all $i \in N$.*

Definition 2.23 *A pair (f, \mathbf{t}) where f is an SCF and t is a transfer scheme, is dominant strategy incentive compatible if,*

$$v_i(f(\theta_i, \theta_{-i}), \theta_i) + t_i(\theta_i, \theta_{-i}) \geq v_i(f(\theta'_i, \theta_{-i}), \theta_i) + t_i(\theta'_i, \theta_{-i}),$$

for all $\theta_i, \theta'_i \in \Theta_i$, for all $\theta_{-i} \in \Theta_{-i}$ and for all $i \in N$.

The SCF f is *implementable in dominant strategies* if there exists a transfer scheme t such that the pair (f, \mathbf{t}) is dominant strategy incentive compatible. Let us use the following shorthands, $\theta \equiv (\theta_i, \theta_{-i})$ and $\Theta \equiv \Theta_1 \times \dots \times \Theta_n$.

Question: What are the SCFs that are implementable?

Below we provide an example of an important implementable SCF.

Example 2.1 The following is the *efficient* SCF f^e . For all $\theta \in \Theta$

$$f^e(\theta) = \operatorname{argmax}_{a \in A} \sum_{i \in N} v_i(a, \theta_i). \quad (2.1)$$

We claim that f^e is implementable. Let $t_i(\theta) = \sum_{j \in N \setminus \{i\}} v_j(f^e(\theta), \theta_j) + h_i(\theta_{-i})$ where h_i is an arbitrary function $h_i : \Theta_{-i} \rightarrow \mathbb{R}$. We show that (f^e, \mathbf{t}) is strategy-proof.

Observe that,

$$\begin{aligned} & v_i(f^e(\theta_i, \theta_{-i}), \theta_i) + t_i(\theta_i, \theta_{-i}) \\ &= v_i(f^e(\theta_i, \theta_{-i}), \theta_i) + \sum_{j \in N \setminus \{i\}} v_j(f^e(\theta_i, \theta_{-i}), \theta_j) + h_i(\theta_{-i}) \\ &= \sum_{i \in N} v_i(f^e(\theta_i, \theta_{-i}), \theta_i) + h_i(\theta_{-i}) \\ &\geq \sum_{i \in N} v_i(f^e(\theta'_i, \theta_{-i}), \theta_i) + h_i(\theta_{-i}) \\ &= v_i(f^e(\theta'_i, \theta_{-i}), \theta_i) + \sum_{j \in N \setminus \{i\}} v_j(f^e(\theta'_i, \theta_{-i}), \theta_j) + h_i(\theta_{-i}) \\ &= v_i(f^e(\theta'_i, \theta_{-i}), \theta_i) + t_i(\theta'_i, \theta_{-i}) \end{aligned}$$

The inequality holds by definition of f^e (Equation (2.1)). Therefore (f^e, x) is strategy-proof. The transfer scheme is known as the Vickrey-Clarke-Groves (VCG) scheme. If Θ is “rich enough”, this scheme is the unique scheme with the property that (f^e, x) is strategy-proof. This is a special case of a class of results called *Revenue Equivalence Theorems*. \square

Very general characterizations of implementability in general domains exist in terms of “monotonicity properties”. Below are explicit characterizations in special domains.

The Complete Domain

Definition 2.24 (Unrestricted Domain) *The domain Θ is unrestricted if, for all $\alpha \in \mathbb{R}^n$, there exists $\theta_i \in \Theta_i, \forall i \in N$, such that for all $a, b \in A$,*

$$v_i(a, \theta_i) - v_i(b, \theta_i) = \alpha_i, \quad \forall i \in N.$$

Theorem 2.5 (Roberts [69]) *Assume A to be finite and $|A| \geq 3$. Let Θ be an unrestricted domain and preferences be quasi-linear. The SCF $f : \Theta \rightarrow A$ is implementable in dominant strategies if and only if there exist non-negative real numbers k_1, \dots, k_N and real numbers $\gamma(a)$*

for all $a \in A$ such that for all $\theta \in \Theta$,

$$f(\theta) = \operatorname{argmax}_{a \in A} \sum_{i \in N} \{k_i v_i(a, \theta_i) + \gamma(a)\}.$$

Moreover the associated transfers are of the form,

$$t_i(\theta) = \frac{1}{k_i} \sum_{j \in N \setminus \{i\}} \{k_j v_j(a, \theta_j) + \gamma(a)\} + h_i(\theta_{-i}).$$

This result characterizes the space of strategy-proof mechanisms in an unrestricted quasi-linear domain. In particular, the VCG scheme of mechanisms is a special case of the above characterization. However, all these results hold assuming that the valuation is independent, i.e., given the alternative a , the value function v_i depends only on the type of agent i , θ_i . However, in many real world scenarios where agents collaboratively solve a problem, e.g., joint projects in multinational companies, or large distributed online projects, the valuation an agent experiences depends on the private types of other agents as well. The model where the valuation of an agent for an allocation depends not just only on her private type but on the types of other agents, is the *interdependent value* model and has also received a great deal of attention in the literature [43, Chap. 6]. The valuation function v_i becomes a function of the alternative a and the entire type vector θ . In the following section, we briefly discuss about this setting of the mechanism design problem.

2.2.8 Mechanism Design with Interdependent Valuations

The interdependent value model poses a more difficult challenge for mechanism design. In increasing generality, Jehiel and Moldovanu [39] and Jehiel et al. [40] have shown that the efficient social choice function cannot *generically* be ex-post implemented. Ex-post implementation requires agents to be truthful about their own type reports when all others are reporting their types truthfully. This is a strong negative result - it rules out the existence of a mechanism that takes type reports from the agents and yields an allocation and a payment rule which satisfies ex-post incentive compatibility and efficiency. However, Mezzetti [48] has shown that these goals can be achieved if the mechanism designer can split the allocation and payment decision into two stages. The idea is that agents report their types in the first stage on the basis of which the designer implements an allocation. Each agent then observes her own valuation and reports this to the designer in the second stage. The designer then proposes a payment based on these valuation announcements. This mechanism is called *generalized Groves mechanism* and in the Nash equilibrium the allocation is efficient. However, a drawback of the mechanism

pointed out by Mezzetti [48] himself is that the agents are indifferent between truth-telling and lying in the second stage, i.e., agents have weak incentives for truth-telling in the second stage.

In Chapter 3, we consider the interdependent valuation setting in greater detail and show how mechanisms can be designed in order to satisfy certain design goals, both in static and dynamic setting. In the following section, we provide a brief introduction to the dynamic mechanism design problem which is a recent development in the mechanism design literature.

2.2.9 Dynamic Mechanism Design

Under this setting, the types of the agents vary over time. This is a relatively new and active research area of mechanism design. Therefore, solutions are available only for special domains and with stylized models of type evolution. For example, for the classic quasi-linear domain with Markov type transitions and independent valuations, some early results are shown by Bergemann and Välimäki [9]. They have proposed an efficient mechanism called the *dynamic pivot mechanism*, which is a generalization of the Vickrey-Clarke-Groves (VCG) mechanism [82, 19, 33] in a dynamic setting, which also serves to be truthful and efficient. Athey and Segal [5] consider a similar setting with an aim to find an efficient mechanism that is budget balanced. Cavallo et al. [17] develop a mechanism similar to the dynamic pivot mechanism in a setting with agents whose type evolution follows a Markov process. In a later work, Cavallo et al. [18] consider *periodically inaccessible* agents and dynamic private information jointly. Even though these mechanisms work for an exchange economy, they have the underlying assumption of *private values*, i.e., the reward experienced by an agent is a function of the allocation and her own private observations.

In the second part of Chapter 3, we consider the dynamic mechanism design problem in interdependent valuation setting. We draw inspiration from the papers of Bergemann and Välimäki [9] and Mezzetti [48], and design a mechanism in this setting, that satisfies a set of very desirable properties.

With this introduction to the game theory and mechanism design, starting from the classic results to the current state-of-the-art, we move to the main contributions of the thesis, which are spread across the following three chapters.

Chapter 3

Skill Elicitation in Crowdsourcing

A particular problem in crowdsourcing is to allocate tasks to a set of experts. The experts can be geographically diverse, but with the help of the Internet, they can be connected to perform tasks efficiently. Because of the diversity, they may have very different levels of expertise for a given task. Even though they know their expertise to a reasonable accuracy, it is often unknown to the designer. If the experts are asked about their expertise, they can potentially misreport if that maximize their payoffs. Therefore, one needs to design a mechanism so that the agents' (experts') expertise can be truthfully elicited, that enables to select an optimal set of experts. In this chapter, we provide a solution to this problem in two situations of increasing generality. In the first part, we model the static version of this problem, while in the second, we extend our model to the *dynamic* setting. We show that the skill elicitation problem falls into the category of *interdependent valuations* in the classical mechanism design theory. We solve a long standing problem in this class of mechanism design problems by providing a mechanism that is strictly truthful in the second round, thereby improving the Mezzetti mechanism in static setting. Then we generalize this result for the dynamic setting.

Let us start with an example of a crowdsourcing experiment. Suppose an amateur botanist wants to know all biological details of a flower that he has found. In order to know how much he can use the expertise of the crowd, he takes a photo of the flower, and puts it on a crowdsourcing platform (e.g., Amazon Mechanical Turk). There are multiple annotators, who can now view the image of the flower and gives their inputs on the different details of the flower. Some annotator can provide the scientific name, someone else can provide the natural habitat, and so on. The situation is schematically shown in Figure 3.1. Depending on their inputs, the botanist can decide which of the annotators' contributions were worth and pay them. However, the payment has to be carefully designed so that the crowd has incentive to participate in this

experiment. Also, the botanist wants that the team of selected experts are chosen in such a way that the task is done by the people with the best available skill-set. Hence, both the allocation of the tasks to agents and the payment are important mechanism design problems. In the following, we are going to present a formal mechanism design framework to model the problems of this kind.

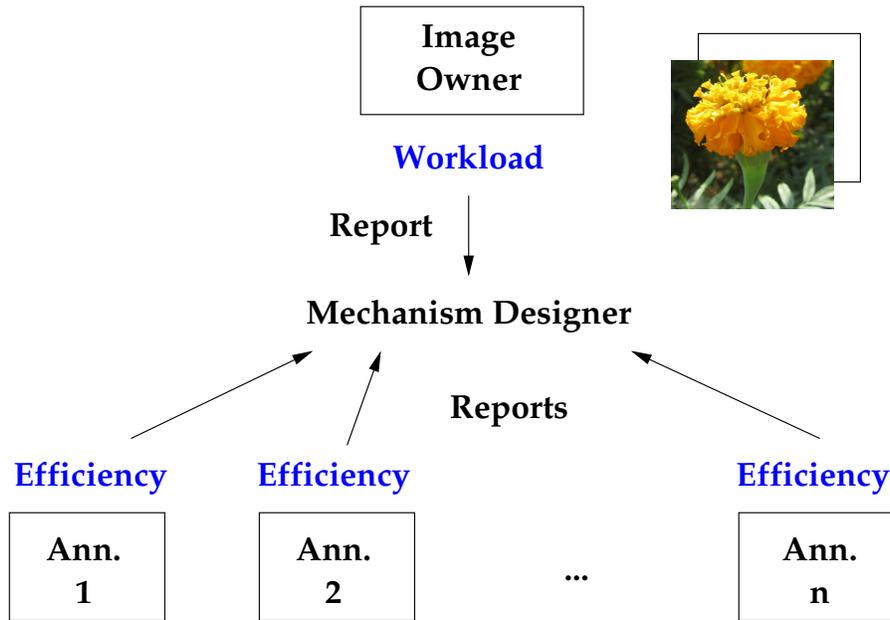


Figure 3.1: Online image annotation via crowdsourcing.

We model the skills of the experts (agents) as their private *types*, and the valuations as the benefit they get (for the task owner) or the cost they incur (for the workers). The setting is a classical quasi-linear utility setting of mechanism design. The difference, however, is that in the crowdsourcing setting, even after the set of agents are selected for a given task, the valuations that the agents get depends not only on their individual types, but also on the types of all selected agents (e.g., the relative success of a project depends on the skills of the team members). This makes the problem to fall into the *interdependent valuations* class. In this chapter, we study the mechanism design problem of this class. In the next section, we are going to address the static mechanism design problem with interdependent valuations. We develop the notation according to the context. In the latter part of this chapter, we address value interdependency when the types vary stochastically.

3.1 Static Mechanism Design with Interdependent Valuations ¹

The impossibility result by Jehiel and Moldovanu [39] says that in a setting with interdependent valuations, any efficient and ex-post incentive compatible mechanism must be a constant mechanism. Mezzetti [48] circumvents this problem by designing a two stage mechanism where the decision of allocation and payment are split over the two stages. This mechanism is elegant, however it faces a limitation. In the second stage, agents are weakly indifferent about reporting their valuations truthfully: an agent’s payment is independent of her reported valuation and truth-telling for this stage is by assumption. We propose an improvement to this mechanism which makes the second stage strictly truthful.

3.1.1 Introduction

In the classical independent private values model [46], each agent observes her valuation which depends on the allocation and her own private type. One can design efficient, dominant strategy incentive compatible mechanisms in this setting, e.g., the VCG mechanism achieves these properties. However, in many real world scenarios where agents collaboratively solve a problem, e.g., joint projects in multinational companies, or large distributed online projects, such as *crowdsourcing* experiments like the DARPA red balloon challenge [65], the valuation of an agent depends on the private types of other agents as well. The model where the valuation of an agent for an allocation depends not just only on her private type but on the types of other agents, is the *interdependent value* model and has also received a great deal of attention in the literature [43].

The interdependent value model poses a more difficult challenge for mechanism design. In increasing generality, Jehiel and Moldovanu [39] and Jehiel et al. [40] have shown that the efficient social choice function cannot *generically* be ex-post implemented. Ex-post implementation requires agents to be truthful about their own type reports when all others are reporting their types truthfully. This is a strong negative result - it rules out the existence of a mechanism that takes type reports from the agents and yields an allocation and a payment rule which satisfies ex-post incentive compatibility and efficiency. However, Mezzetti [48] has shown that these goals can be achieved if the mechanism designer can split the allocation and payment decisions into two stages. The agents report their types in Stage 1, and the designer implements an allocation based on that. Then, each agent observes her own valuation and reports the values to the designer in Stage 2. The designer then proposes a payment based on the two-stage

¹This work has appeared in Economics Letters [57].

reports. This mechanism is called *generalized Groves mechanism* and in the Nash equilibrium the allocation is efficient. However, a drawback of the mechanism pointed out by Mezzetti [48] is that the agents are indifferent between truth-telling and lying in Stage 2, i.e., agents have weak incentives for truth-telling in Stage 2. Hereafter, we will refer to this mechanism as the *classic mechanism*.

In this section, we propose a mechanism called Value Consistent Pivotal Mechanism (VCPM) that overcomes this difficulty. In particular it proposes a different set of payments from the classic mechanism in Stage 2 which makes it a strict ex-post Nash equilibrium for each agent to reveal her valuation truthfully at this stage.

The question may arise whether this mechanism hurts some other properties of Mezzetti's original mechanism. Since VCPM yields the same payoff to the agents as that of the classic in equilibrium, it continues to satisfy all the properties that the classic mechanism satisfies in equilibrium. For example, we show that in a restricted problem domain, a refinement of our mechanism satisfies *individual rationality* (IR) as it is satisfied by the classic mechanism. However, truth-telling in the classic mechanism is also a subgame perfect equilibrium, which is compromised by VCPM. We illustrate this with an example.

3.1.2 Model and Definitions

Let the set of agents be denoted by $N = \{1, \dots, n\}$. Each agent observes her private type $\theta_i \in \Theta_i$. Let $\Theta = \times_{i \in N} \Theta_i$ denote the type profile space where $\theta \equiv (\theta_1, \dots, \theta_n)$ be an element of Θ . We will denote the type profile of all agents except agent i by $\theta_{-i} \equiv (\theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_n) \in \Theta_{-i}$. Types are drawn independently across agents and each agent can only observe her own type and does not know the types of other agents. We consider a standard quasi-linear model. Therefore, the payoff u_i of agent i is the sum of her value v_i and transfer p_i .

We will consider a two stage mechanism similar to that of Mezzetti [48], because of the impossibility result by Jehiel and Moldovanu [39]. We call this mechanism Value Consistent Pivotal Mechanism (VCPM).

In Stage 1, agents are asked to report their types, and after that the mechanism designer chooses an alternative from the set A via the allocation function $a : \Theta \rightarrow A$. We denote the reported types by $\hat{\theta}$, hence, the allocation for such a report is given by $a(\hat{\theta})$.

All agents then experience the consequence of the allocation via the valuation function which is defined by $v_i : A \times \Theta \rightarrow \mathbb{R}$, for all $i \in N$. Note: the value function is different from the independent private value setting, where it is a mapping $v_i : A \times \Theta_i \rightarrow \mathbb{R}$. This difference makes mechanism design in interdependent value settings difficult as discussed in Section 4.1.

In Stage 2, agents report their experienced valuations; transfers given by $p_i : \Theta \times \mathbb{R}^n \rightarrow$

\mathbb{R} , $\forall i \in N$ are then decided by the designer. If the reported valuations are $\hat{v} \in \mathbb{R}^n$, the transfer to agent i is given by $p_i(\hat{\theta}, \hat{v})$.

The two stage mechanism VCPM is graphically illustrated in Figure 3.2.

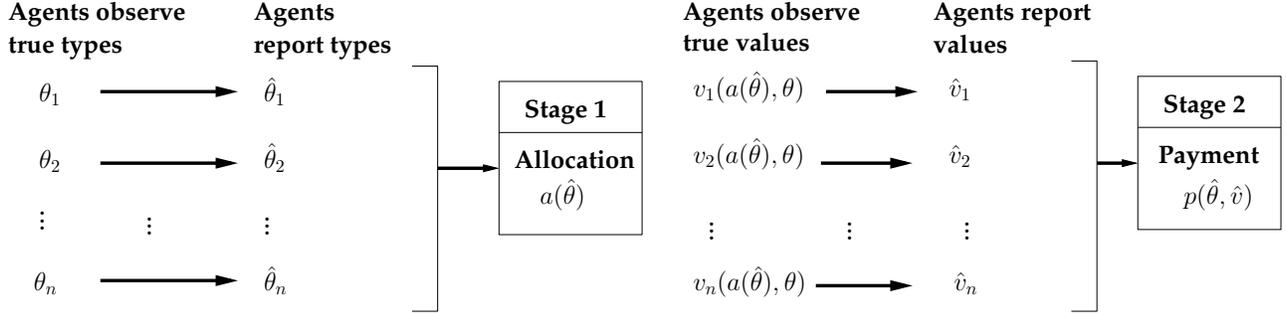


Figure 3.2: Graphical illustration of VCPM.

Definitions

As discussed earlier, in the section we will consider only two stage mechanisms, where in Stage 1, agents report their types and in Stage 2, their experienced valuations. The allocation decision is made after Stage 1 and the payment after Stage 2. It is, therefore, necessary to define the notions of efficiency, truthfulness, and voluntary participation, in this setting.

We consider only quasi-linear domains where the payoff is the sum of the valuation and transfer. A mechanism M in this domain is fully characterized by a tuple of allocation and payment $\langle a, p \rangle$. For a truthful mechanism in this setting, we need to ensure that it is truthful in both stages. In Stage 1, truthfulness implies that the agents report their true types. In the second round, the valuation is a function of the allocation chosen in Stage 1. Here truthfulness would mean that they report their observed valuations due to that allocation.

The true type profile is given by θ . With a slight abuse of notation, we represent the true valuation vector by $v = (v_1(a(\theta), \theta), \dots, v_n(a(\theta), \theta))$ under mechanism $M = \langle a, p \rangle$. Let us denote the payoff of agent i by $u_i^M(\hat{\theta}, \hat{v} | \theta, v)$ under mechanism M when the reported type and value vectors are $\hat{\theta}$ and \hat{v} respectively, while the true type and value vectors are given by θ and v . Therefore, due to the quasi-linear assumption, the payoff is given by,

$$u_i^M(\hat{\theta}, \hat{v} | \theta, v) = v_i(a(\hat{\theta}), \theta) + p_i(\hat{\theta}, \hat{v}).$$

Definition 3.1 (Efficiency (EFF)) *A mechanism $M = \langle a, p \rangle$ is efficient if the allocation rule*

maximizes the sum valuation of the agents. That is, for all θ ,

$$a(\theta) \in \operatorname{argmax}_{a \in A} \sum_{j \in N} v_j(a, \theta).$$

Definition 3.2 (Ex-post Incentive Compatibility (EPIC)) A mechanism $M = \langle a, p \rangle$ is ex-post incentive compatible if reporting the true type and valuation is an ex-post Nash equilibrium of the induced game. That is, for all true type profiles $\theta = (\theta_i, \theta_{-i})$ and true valuation profiles $v = (v_i, v_{-i}) = (v_i(a(\theta), \theta), v_{-i}(a(\theta), \theta))$, and for all $i \in N$,

$$\begin{aligned} & u_i^M((\theta_i, \theta_{-i}), (v_i(a(\theta), \theta), v_{-i}(a(\theta), \theta)) | \theta, v) \\ & \geq u_i^M((\hat{\theta}_i, \theta_{-i}), (\hat{v}_i, v_{-i}(a(\hat{\theta}_i, \theta_{-i}), \theta)) | \theta, v), \quad \forall \hat{\theta}_i, \hat{v}_i. \end{aligned}$$

Definition 3.3 (Ex-post Individual Rationality (EPIR)) A mechanism $M = \langle a, p \rangle$ is ex-post individually rational if the payoff of each agent in the true type and valuation profile is non-negative. That is, for all $i \in N$, $\theta = (\theta_i, \theta_{-i})$, and $v = (v_i, v_{-i}) = (v_i(a(\theta), \theta), v_{-i}(a(\theta), \theta))$,

$$u_i^M(\theta, v | \theta, v) \geq 0.$$

Subset Allocation (SA) Later in this section, we will focus on a problem domain named *subset allocation*, where the allocation set is the set of all subsets of the agents, i.e., $A = 2^N$. In such a setting, we assume that the valuation of agent i is given by,

$$v_i(a, \theta) = \begin{cases} v_i(a, \theta_a) & \text{if } i \in a, \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

We use θ_a to denote the type vector of the allocated agents, i.e., $\theta_a = (\theta_j)_{j \in a}, \forall a \in A$. This means that when agent i is not selected, her valuation is zero, and when she is selected, the valuation depends only on the types of the selected agents. This restricted domain is relevant for distributed projects in organizations, where the skill level of only the allocated agents matter in the value achieved by the other allocated agents. The skill levels of all the workers/employees participating in the project impact the success or failure of the project, and the reward or loss is shared by the participants of the project. This and several other examples of collaborative task execution falls under the SA domain, which makes it interesting to study.

3.1.3 Main Results

With the dynamics of the mechanisms as in Figure 3.2, the mechanism design problem is to design the allocation and the transfer rules. In VCPM, we adopt the following allocation and transfer rules.

Stage 1: Agents report types $\hat{\theta} = (\hat{\theta}_i, \hat{\theta}_{-i})$. The allocation is chosen as,

$$a^*(\hat{\theta}) \in \operatorname{argmax}_{a \in A} \sum_{j \in N} v_j(a, \hat{\theta}). \quad (3.2)$$

Stage 2: Agents report valuations \hat{v} . The transfer to agent i , $i \in N$ is,

$$p_i^*(\hat{\theta}, \hat{v}) = \sum_{j \neq i} \hat{v}_j - g(\hat{v}_i, v_i(a^*(\hat{\theta}), \hat{\theta})) - h_i(\hat{\theta}_{-i}). \quad (3.3)$$

Where h_i is any arbitrary function of $\hat{\theta}_{-i}$, and $g(x, \ell)$ is a non-negative function of x with a unique zero at ℓ .

An example of the function $g(x, \ell)$ would be $(x - \ell)^2$.

The stages of the mechanism are shown in algorithmic form in Algorithm 1. The difference between this mechanism with that of Mezzetti's is that we charge a tax to the agent i for not being consistent with Stage 1 of type reports. Note that the value function is a common knowledge. Together with the reported type vector $\hat{\theta}$, the designer can compute the value $v_i(a^*(\hat{\theta}), \hat{\theta})$ in Equation (3.3). The amount of tax is positive whenever the agents valuation announcement are inconsistent with the value computed according

Algorithm 1 VCPM

Stage 1:

for agents $i = 1, \dots, n$ **do**

agent i observes θ_i ;

agent i reports $\hat{\theta}_i$;

end for

compute allocation $a^*(\hat{\theta})$ according to Equation (3.2);

Stage 2:

for agents $i = 1, \dots, n$ **do**

agent i observes $v_i(a^*(\hat{\theta}), \theta)$;

agent i reports \hat{v}_i ;

end for

compute payment to agent i , $p_i^*(\hat{\theta}, \hat{v})$, Equation (3.3);

to their reported types in Stage 1. We will show in the following theorem that this modification in the transfer makes VCPM truth-telling a strict best-response in Stage 2.

Theorem 3.1 *VCPM is EFF and EPIC. In particular, reporting the valuations truthfully in Stage 2 of this mechanism is a strict best-response for each agent.*

Proof: The allocation rule of VCPM given by Equation (3.2) ensures efficiency by construction. Therefore, we are only required to show that the mechanism is ex-post incentive compatible.

To show that VCPM is ex-post incentive compatible, without loss of generality, let us assume that all agents except agent i are reporting their types and valuations truthfully in the two stages. Let us assume that the true types are given by $\theta = (\theta_i, \theta_{-i})$. Hence, under these assumptions, $\hat{\theta} = (\hat{\theta}_i, \theta_{-i})$. The value reports in Stage 2 is dependent on the allocation in the first. Hence, for the agents $j \neq i$, who are truthful, the value reports are given by, $\hat{v}_j = v_j(a^*(\hat{\theta}), \theta)$. As defined earlier, we denote the payoff of agent i by $u_i((\hat{\theta}_i, \theta_{-i}), (\hat{v}_i, v_{-i}(a^*(\hat{\theta}), \theta)) | \theta, v)$ when all agents except i report their types and values truthfully, and the true type and value profiles are θ and v . The payoff of agent i is given by,

$$\begin{aligned}
& u_i^{\text{VCPM}}((\hat{\theta}_i, \theta_{-i}), (\hat{v}_i, v_{-i}(a^*(\hat{\theta}), \theta)) | \theta, v) \\
&= v_i(a^*(\hat{\theta}_i, \theta_{-i}), \theta) + p_i((\hat{\theta}_i, \theta_{-i}), (\hat{v}_i, v_{-i})) \\
&= v_i(a^*(\hat{\theta}_i, \theta_{-i}), \theta) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_i, \theta_{-i}), \theta) - g(\hat{v}_i, v_i(a^*(\hat{\theta}), \hat{\theta})) - h_i(\theta_{-i}) \\
&\leq v_i(a^*(\hat{\theta}_i, \theta_{-i}), \theta) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_i, \theta_{-i}), \theta) - h_i(\theta_{-i}) \\
&= \sum_{j \in N} v_j(a^*(\hat{\theta}_i, \theta_{-i}), \theta) - h_i(\theta_{-i}) \\
&\leq \sum_{j \in N} v_j(a^*(\theta_i, \theta_{-i}), \theta) - h_i(\theta_{-i}) \\
&= v_i(a^*(\theta), \theta) + \sum_{j \neq i} v_j(a^*(\theta), \theta) - h_i(\theta_{-i}) \\
&= u_i^{\text{VCPM}}(\theta, v | \theta, v)
\end{aligned}$$

The first and second equalities are by definition and by substituting the expression of transfer (Eq. (3.3)). The first inequality comes since we are ignoring a non-positive term. The third equality is via simple reorganization the terms. The second inequality comes by definition of the allocation rule (Eq. (3.2)). The rest of the steps are simple reorganization of the expressions. The last equality is due to the fact that the function g is zero when $\hat{v}_i = v_i(a^*(\theta), \theta)$. Hence, we prove that VCPM is ex-post incentive compatible.

Let us explain that in this ex-post Nash equilibrium of this game, reporting values truthfully in Stage 2 is a strict best-response. This is because, when types are reported truthfully in Stage 1, the second term in the expression of the transfer for agent i (Eq. (3.3)) is reduced to $g(\hat{v}_i, v_i(a^*(\theta), \theta))$. This term is minimized (thereby the payoff to agent i is maximized) when $\hat{v}_i = v_i(a^*(\theta), \theta)$, which is the true report. Hence, truthful report in stage two of VCPM is a strict best-response. ■

Comparison with the Classic Mechanism

It is reasonable to ask whether the proposed mechanism VCPM that achieves the strict Nash equilibrium in Stage 2, continues to satisfy all other desirable properties that the original mechanism given by Mezzetti [48] used to satisfy. Since we prove that VCPM is EPIC, truthful reporting is an ex-post Nash equilibrium, and in that equilibrium, the penalty term $g(\hat{v}_i, v_i(a^*(\hat{\theta}), \hat{\theta}))$ in the expression of the transfer (Eq. (3.3)) is zero. Therefore, the allocation and the transfer in this truthful equilibrium are exactly the same as in the classic mechanism, and so is the payoff of each agent. So, any property that the classic mechanism used to satisfy in the ex-post Nash equilibrium will continue to hold even for VCPM. In addition, the truthful reporting in Stage 2 a strict best-response.

In the next section, we illustrate one such desirable property, namely the ex-post individual rationality, and show that under a restricted domain, a refined VCPM satisfies this property as does the classic mechanism.

It is important to note that the weak indifference in Stage 2 of the classic mechanism enables truth reporting also a subgame perfect equilibrium. The strict EPIC of VCPM comes at the expense of the subgame perfection. We illustrate this in the section following the next with an example.

VCPM and Ex-post Individual Rationality

In this section, we investigate the incentives for individuals to participate in this game. We consider the subset allocation (SA) domain. Hence, $A = 2^N$. Let us define the social welfare as,

$$W(\theta) = \max_{a \in A} \sum_{j \in a} v_j(a, \theta_a). \quad (3.4)$$

Similarly, the social welfare excluding agent i is given by,

$$W_{-i}(\theta_{-i}) = \max_{a_{-i} \in A_{-i}} \sum_{j \in a_{-i}} v_j(a_{-i}, \theta_{a_{-i}}), \quad (3.5)$$

where $A_{-i} = 2^{N \setminus \{i\}}$. Notice that in the SA domain $A_{-i} \subseteq A$, and therefore we make the following observation,

Observation 3.1 *In SA problem domain, with W and W_{-i} defined as in Eqs. (3.11) and (3.5), $W(\theta) \geq W_{-i}(\theta_{-i})$.*

This is because, $a_{-i} \in A_{-i} \subseteq A$. Therefore, while choosing allocation a that yields the social welfare including agent i , the designer has the choice of choosing all a_{-i} 's as well. Therefore, with the form of valuation functions in SA, the social welfare including agent i will always dominate that excluding her.

Let us refine the VCPM to yield RVCPM by redefining the allocation (Equation (3.2)) in stage one as follows.

$$a^{\text{RVCPM}}(\hat{\theta}) \in \operatorname{argmax}_{a \in A} \sum_{j \in a} v_j(a, \hat{\theta}_a). \quad (3.6)$$

We also redefine the payment (Equation (3.3)) in stage two as follows.

$$p_i^{\text{RVCPM}}(\hat{\theta}, \hat{v}) = \sum_{j \in a^{\text{RVCPM}}(\hat{\theta}) \setminus \{i\}} \hat{v}_j - g(\hat{v}_i, v_i(a^{\text{RVCPM}}(\hat{\theta}), \hat{\theta})) - W_{-i}(\hat{\theta}_{-i}). \quad (3.7)$$

Note that the h_i function in the VCPM is replaced by W_{-i} in RVCPM. The following Corollary is now immediate from Theorem 3.2 and Observation 3.1.

Corollary 3.1 *In the SA problem domain, RVCPM is EFF, EPIC, and EPIR.*

Proof: Since RVCPM is a special case of VCPM and SA is a restricted domain, the results of VCPM holds in this setting too. Therefore from Theorem 3.2, we conclude that RVCPM is EFF and EPIC. Now, in the ex-post Nash equilibrium, the payoff of agent i is given by,

$$\begin{aligned} & u_i^{\text{RVCPM}}(\theta, v | \theta, v) \\ &= v_i(a^{\text{RVCPM}}(\theta), \theta_{a^{\text{RVCPM}}(\theta)}) + p_i^{\text{RVCPM}}(\theta, v) \\ &= v_i(a^{\text{RVCPM}}(\theta), \theta_{a^{\text{RVCPM}}(\theta)}) + \sum_{j \in a^{\text{RVCPM}}(\theta) \setminus \{i\}} v_j(a^{\text{RVCPM}}(\theta), \theta_{a^{\text{RVCPM}}(\theta)}) - W_{-i}(\theta_{-i}) \\ &= \sum_{j \in a^{\text{RVCPM}}(\theta)} v_j(a^{\text{RVCPM}}(\theta), \theta_{a^{\text{RVCPM}}(\theta)}) - W_{-i}(\theta_{-i}) \\ &= W(\theta) - W_{-i}(\theta_{-i}) \\ &\geq 0. \end{aligned}$$

The first three equalities are by definition and simple reorganization of the terms. The fourth equality is by the definition of $W(\theta)$ (Equation (3.11)) and $a^{\text{rVCPM}}(\theta)$ (Equation (3.7)). The inequality is due to Observation 3.1. Hence, rVCPM is ex-post individually rational. ■

VCPM and Subgame Perfect Equilibrium

In this section, we show with an example that truth reporting in VCPM is not a subgame perfect equilibrium (SPE).

Example 3.1 Let us consider two alternatives a_1 and a_2 and three agents, $i = 1, 2, 3$. Let $\Theta_1 = \Theta_2 = \{0, 2\}$, and $\Theta_3 = \phi$ (i.e., agents 1 and 2 have two types each, agent 3 has no private information). Assume, $v_i(a_1, \theta) = 0$, for all i and θ , $v_1(a_2, \theta) = \theta_1 + 2\theta_2$, $v_2(a_2, \theta) = 2\theta_1 + \theta_2$, $v_3(a_2, \theta) = -3$, for all θ . The efficient allocation under VCPM is, $a^*(\theta) = a_1$, if $\theta_1 = \theta_2 = 0$, and a_2 otherwise. Let us assume, $h_i \equiv 0$ for simplicity. Under VCPM, no matter what was reported in Stage 1, the unique best reply for each agent is to report a value, $\hat{v}_i = v_i(a^*(\hat{\theta}), \hat{\theta})$. Now suppose that the true types are $\theta_1 = \theta_2 = 0$ and that agent 2 truthfully reports in Stage 1, i.e., $\hat{\theta}_2 = 0$ and then reports $\hat{v}_2 = v_2(a^*(\hat{\theta}), \hat{\theta})$ (as it must in a SPE).

If agent 1 reports truth, i.e., $\hat{\theta}_1 = 0$, then the implemented allocation is $a^*(0, 0) = a_1$. The allocation gives him zero value, and the second round transfer to agent 1 is $\sum_{j \neq 1} v_j(a_1, (0, 0)) = 0$. In short, agent 1 obtains a total utility of zero if he reports his true type.

If he instead lies and reports $\hat{\theta}_1 = 2$, then the implemented allocation is $a^*(2, 0) = a_2$. The allocation still gives him zero value (as the true types are $\theta_1 = \theta_2 = 0$), but the second round transfer to agent 1 now is $\sum_{j \neq 1} v_j(a_2, (2, 0)) = 4 - 3 = 1$. In short, agent 1 obtains a total utility of one if he misreports. Hence, truth-telling in Stage 1 is not a SPE for agent 1. □

The classic mechanism is SPE because the payoffs of the agents in Stage 2 are independent of their value reports, thereby making the EPIC weak. Hence, we can see that there is a trade-off between the strict EPIC in Stage 2 and the subgame perfection.

We observe from the example that even though agents can misreport in Stage 1, in order to report optimally according to that misreported $\hat{\theta}$, the agents need to know the reported $\hat{\theta}$ in Stage 2. This is often impractical in actual implementation, and the mechanism designer can stop the reported $\hat{\theta}$ information from disseminating before the agents take their actions in Stage 2. So, the inability to satisfy subgame perfection poses a much smaller threat than the weak indifference on reporting the values truthfully in Stage 2. Hence, VCPM solves an important limitation in the mechanism design with interdependent valuations.

Reduced Form vs State-of-the-World Formulation

In the original paper by Mezzetti [48], a *state-of-the-world* variable ω was introduced, which is realized after the allocation and before the agents observe their valuations. The valuations are

functions of this variable, and the payment is decided after they report their observed valuations that depend on the realization of ω . The mechanism proposed in that paper is weak EPIC in the second stage since the payment does not depend on the agent concerned's report. The EPIC in the first round, however, is with the expectation over the ω , since the allocation decision is done in the first stage and before ω realizes. In contrast, the *reduced form* refers to the setting where all the analysis is done taking expectation over ω . In this section, we have discussed the reduced form analysis so far, and now we make a few observations on the state-of-the-world formulation.

The state-of-the-world variable ω affects any strict EPIC mechanism including VCPM in the following way. It depends on the following cases regarding the observability of the designer.

Case 1: *The designer can observe the state-of-the-world ω .* In such a case, by redefining the penalty term to be $g(\hat{v}_i, v_i(a^*(\hat{\theta}), \hat{\theta}, \omega))$, where $v_i(a^*(\hat{\theta}), \hat{\theta}, \omega)$ is now computable by the designer, we can satisfy the strict EPIC in the second round for each realization of ω , in addition to satisfying EPIC in the first round with expectation over ω .

Case 2: *The designer cannot observe the state-of-the-world ω , but is able to decipher it given the agents reports.* This is going to hold for a restricted class of problems. For example, if for each allocation a , type profile θ , and $x \in \text{Range}(v)$, there exists at least two agents $i_1(x)$ and $i_2(x)$ such that both $v_{i_1(x)}^{-1}(x)$ and $v_{i_2(x)}^{-1}(x)$ gives back a , θ , and a unique ω . This implies that, each state of the world uniquely affects at least two agents in the population. In such a setting, it is possible for the designer to retrieve the true state-of-the-world ω from the other agents' report and use the g function as used in case 1 above, and make VCPM a strict EPIC in the second stage as well.

Case 3: *The designer cannot observe the state-of-the-world ω , and is not able to decipher it given the agents reports.* This scenario is difficult for designing any strict EPIC mechanism. We provide an example where a naive expectation of g given by, $\int_{\Omega} g(\hat{v}_i, v_i(a^*(\hat{\theta}), \hat{\theta}, \omega)) d\omega$ does not work, and leave a detailed investigation as an interesting future work.

Suppose, ω can take only two possible states, 1 and 0. The priors are $\mathbb{P}(\omega = 1) = 0.99$, and $\mathbb{P}(\omega = 0) = 0.01$. Let us fix an allocation a and a type profile θ , and let $v_i(a, \theta, \omega = 1) = 1$, and $v_i(a, \theta, \omega = 0) = 0$, then if i observes $v_i = 0$, she might still report $v_i = 1$ because the ω corresponding to that outcome is a lot more likely. Say $g(x, \ell) = (x - \ell)^2$, the agent would look at minimizing $(\hat{v}_i - 1)^2 \times 0.99 + (\hat{v}_i - 0)^2 \times 0.01$ which is minimized by $\hat{v}_i = 1$ even if $v_i = 0$.

The example above clearly shows that there is no easy way to convert a strict EPIC mechanism in the reduced form into a strict EPIC mechanism with state-of-the-world formulation. It appears to us that in an unrestricted interdependent value domain, it could be impossible to design any strict EPIC mechanism, an investigation of which we leave as a future work.

3.2 Dynamic Mechanism Design with Interdependent Valuations ¹

It is often the case that the types of the agents vary over time. In this section, we extend the results on mechanism design discussed in the last section into a dynamic setting, where the horizon is infinite, and valuation is interdependent. We assume that the types of the agents are evolving according to a first order Markov process and are independent across agents in every round and across rounds. However, the valuations of the agents are functions of the types of all the agents, which makes the problem fall into an interdependent value model. Designing mechanisms in this setting is non-trivial not only because of the impossibility result of Jehiel and Moldovanu [39], but also because of the dynamic evolution of the types. In this section, we provide a first attempt at designing a dynamic mechanism which is *strict* ex-post incentive compatible and efficient in interdependent value setting with Markovian type evolution. In a restricted domain, which appears often in real-world scenarios, we show that our mechanism is ex-post individually rational as well.

3.2.1 Introduction

Organizations often face the problem of executing a task for which they do not have enough resources or expertise. It may also be difficult, both logistically and economically, to acquire those resources. For example, in the area of healthcare, it has been observed that there are very few occupational health professionals and doctors and nurses in all specialities at the hospitals in the UK [63]. With the advances in computing and communication technologies, a natural solution to this problem is to outsource the tasks to experts outside the organization. Hiring experts beyond an organization was already in practice. However, with the advent of the Internet, this practice has extended even beyond the international boundaries, e.g., some U.S. hospitals are outsourcing the tasks of reading and analyzing scan reports to companies in Bangalore, India [4]. Gupta et al. [34] give a detailed description of how the healthcare industry uses the outsourcing tool.

The organizations where the tasks are outsourced (let us call them vendors) have quite varied efficiency levels. For tasks like healthcare, it is extremely important to hire the right set of experts. If the efficiency levels of the vendors and the medical task difficulties of the hospitals are observable by a central management (controller), and these levels vary over time according to a Markov process, the problem of selecting the right set of experts reduces to a

¹Part of this work has been published in the Conference on Uncertainty in Artificial Intelligence, UAI 2011 [58], and the complete work has been submitted to the Review of Economic Design [62].

Markov Decision Problem (MDP), which has been well studied in the literature [10, 66]. Let us call the efficiency levels and task difficulties together as *types* of the tasks and resources.

However, the types are usually observed privately by the vendors and hospitals (agents), who are rational and intelligent. The efficiencies of the vendors are private information of the vendors (depending on what sort of doctors they hire, or machinery they use), and they might misreport this information in order to win the contract and to increase their net returns. At the same time the difficulty of the medical task is private to the hospital, and is unknown to the experts. A strategic hospital, therefore, can misreport this information to the hired experts as well. Hence, the asymmetry of information at different agents' end transforms the problem from a *completely* or *partially* observable MDP into a *dynamic game* among the agents.

Motivated by examples of this kind, in this section, we analyze them using a formal mechanism design framework. We consider only cases where the solution of the problem involves monetary transfer, which makes the payoffs quasi-linear. The reporting strategy of the agents and the decision problem of the controller is dynamic since we assume that the types of the tasks and resources are varying with time. In addition, the above problem has two characteristics, namely, *interdependent values*: the task execution generates values to the task owners that depend on the efficiencies of the assigned resources, and *exchange economy*: a trade environment where both buyers (task owners) and sellers (resources) are present. In this section, the theme of modeling and analysis would be centered around the settings of task outsourcing to strategic experts. We aim to have a socially efficient mechanism, and at the same time, that would demand truthfulness and voluntary participation of the agents in this setting.

Prior Work

The above properties have been investigated separately in literature on *dynamic mechanism design*. Bergemann and Välimäki [9] have proposed an efficient mechanism called the *dynamic pivot mechanism*, which is a generalization of the Vickrey-Clarke-Groves (VCG) mechanism [82, 19, 33] in a dynamic setting, which also serves to be truthful and efficient. Athey and Segal [5] consider a similar setting with an aim to find an efficient mechanism that is budget balanced. Cavallo et al. [17] develop a mechanism similar to the dynamic pivot mechanism in a setting with agents whose type evolution follows a Markov process. In a later work, Cavallo et al. [18] consider *periodically inaccessible* agents and dynamic private information jointly. Even though these mechanisms work for an exchange economy, they have the underlying assumption of *private values*, i.e., the reward experienced by an agent is a function of the allocation and her own private observations. Mezzetti [48, 49], on the other hand, explored the other facet, namely, interdependent values, but in a static setting, and proposed a truthful mechanism.

The mechanism proposed in these papers use a two-stage mechanism, since it is impossible to design a single-stage mechanism satisfying both truthfulness and efficiency even for a static setting [39].

Contributions

In this section, we propose a dynamic mechanism named **MDP-based Allocation and TRansfer in Interdependent-valued eXchange economies** (abbreviated **MATRIX**), which is designed to address the class of *interdependent values*. It extends the results of Mezzetti [48] to a dynamic setting, and serves as an efficient, truthful mechanism. Under a certain realistic domain restriction, agents receive non-negative payoffs by participating in it. The key feature that distinguishes our model and results from that of the existing dynamic mechanism literature is that we address the interdependent values and dynamically varying types (in an exchange economy) jointly. In Table 3.1, we have summarized the different paradigms of the mechanism design problem, and their corresponding solutions in the literature.

Valuations	STATIC	DYNAMIC
Independent	VCG Mechanism [82, 19, 33]	Dynamic Pivot Mechanism [9, 17]
Interdependent	Generalized VCG [48]	Mechanism MATRIX (this section)

Table 3.1: The different paradigms of mechanism design problems with their solutions.

Our main contributions in this section can be summarized as follows.

- We propose a dynamic mechanism **MATRIX**, that is *efficient*, and *truthful* for the participants in an *interdependent-valued exchange economy* (Theorem 3.2).
 - ▶ This extends the classic mechanism proposed by Mezzetti [48] to a dynamic setting.
 - ▶ It solves the issue of weak indifference by the agents in the second stage of the classic mechanism.
- Under a restricted domain, we show that this mechanism is also *individually rational* for the agents (Theorem 3.3).
- We discuss why known mechanisms like a fixed payment mechanism, VCG, or the dynamic pivot mechanism [9] do not satisfy all the properties that **MATRIX** satisfies (beginning of Section 5.1.3 and Section 3.2.3).

We discuss that **MATRIX** comes at a computational cost which is the same as that of its independent value counterpart (Section 3.2.3). This work provides a first attempt of designing a

dynamic mechanism which is *strict ex-post* incentive compatible and efficient in interdependent value setting with Markovian type evolution. In a restricted domain, which appears often in real-world scenarios, we show that our mechanism is ex-post individually rational as well.

3.2.2 Background and Model

Let the set of agents be given by $N = \{1, \dots, n\}$, who interact with each other for a countably infinite time horizon indexed by time steps $t = 0, 1, 2, \dots$. The time-dependent type of each agent is denoted by $\theta_{i,t} \in \Theta_i$ for $i \in N$. We will use the shorthands $\theta_t \equiv (\theta_{1,t}, \dots, \theta_{n,t}) \equiv (\theta_{i,t}, \theta_{-i,t})$, where $\theta_{-i,t}$ denotes the type vector of all agents excluding agent i . We will refer to θ_t as the type profile, $\theta_t \in \Theta \equiv \times_{i \in N} \Theta_i$.

Stationary Markov Type Transitions, SMTT. The combined type θ_t follows a first order Markov process which is governed by the transition probability function $F(\theta_{t+1}|a_t, \theta_t)$, which is independent across agents, defined formally below.

Definition 3.4 (Stationary Markov Type Transitions, SMTT) *We call the type transitions to follow stationary Markov type transitions if the joint distribution F of the types of the agents $\theta_t \equiv (\theta_{1,t}, \dots, \theta_{n,t})$, and the marginals F_i 's exhibit the following for all t .*

$$\begin{aligned} F(\theta_{t+1}|a_t, \theta_t, \theta_{t-1}, \dots, \theta_0) &= F(\theta_{t+1}|a_t, \theta_t), \text{ and} \\ F(\theta_{t+1}|a_t, \theta_t) &= \prod_{i \in N} F_i(\theta_{i,t+1}|a_t, \theta_{i,t}). \end{aligned} \tag{3.8}$$

We will assume the types to follow SMTT throughout the rest of the chapter.

The allocation set is denoted by A . In each round t , the mechanism designer chooses an allocation a_t from this set and decides a payment $p_{i,t}$ to agent i . The allocation leads to a valuation to agent i , $v_i : A \times \Theta \rightarrow \mathbb{R}$. This is in contrast to the classical independent valuations (also called *private values*) case where valuations are assumed to depend only on i 's own type; $v_i : A \times \Theta_i \rightarrow \mathbb{R}$. However, we assume for all i , $|v_i(a, \theta)| < \infty$, for all a and θ .

In the later part of this chapter, we will restrict our attention to a restricted space of allocations and valuations as discussed below.

Subset Allocation, SA. Let us motivate this restriction with the medical task assignment example given in the previous section. The organizations outsource tasks to experts for a payment, where the expert may have different and often time-varying capabilities of executing the task. The task owners come with a specific task difficulty (type of the task owner), which is usually privately known to them, while the workers' capabilities (types of the workers) are their private information. A central planner's job in this setting is to efficiently assign the tasks

to the workers. Clearly, in this setting, the set of possible allocations is the set of the subsets of agents, i.e., $A = 2^N$. Note that, for a finite set of players, the allocation set is always finite. So, we can formally define this setting as follows.

Definition 3.5 (Subset Allocation, SA) *When the set of allocations is the set of all subsets of the agent set, i.e., $A = 2^N$, we call the domain a subset allocation domain.*

PEER INFLUENCED VALUATIONS, PIV Even though the valuation of agent i is affected by not only her private type but also by the types of others, it is often the case that the valuation is affected by the types (e.g. the efficiencies of the workers in a joint project) of only the *selected* agents. The valuation therefore is a function of the types of the allocated agents and not the whole type vector, $v_i : A \times \Theta_A \rightarrow \mathbb{R}$. We also assume that the value of a non-selected agent is zero. Formally, we define this setting as *peer influenced valuations* (PIV).

Definition 3.6 (Peer Influenced Valuations, PIV) *This is a special case of interdependent valuations in the SA domain, where the valuation of agent i is a function of the types of other selected agents, $v_i : A \times \Theta_A \rightarrow \mathbb{R}$. In particular, the value function is given by,*

$$v_i(a, \theta) = \begin{cases} v_i(a, \theta_a) & \text{if } i \in a \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

Efficient Allocation, EFF. The mechanism designer aims to maximize the sum of the valuations of task owners and workers, summed over an infinite horizon, geometrically discounted with factor $\delta \in (0, 1)$. The discount factor accounts for the fact that a future payoff is less valued by an agent than a current stage payoff. We assume δ to be common knowledge. If the designer would have perfect information about the θ_t 's, his objective would be to find a *policy* π_t , which is a sequence of allocation functions from time t , that yields the following for all t and for all type profiles θ_t ,

$$\pi_t \in \underset{\gamma}{\operatorname{argmax}} \mathbb{E}_{\gamma, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{i \in N} v_i(a_s(\theta_s), \theta_s) \right], \quad (3.10)$$

where $\gamma = (a_t(\cdot), a_{t+1}(\cdot), \dots)$ is any arbitrary sequence of allocation functions. Here we use $\mathbb{E}_{\gamma, \theta_t}[\cdot] = \mathbb{E}[\cdot | \theta_t; \gamma]$ for brevity of notation. We point to the fact that the allocation policy γ is not a random variable in this expectation computation. The policy is a functional that specifies what action to take in each time instant for a given type profile. Different policies will lead to different sequences of allocation functions over the infinite horizon, and the efficient allocation is the one that maximizes the expected discounted sum of the valuations of all the agents.

In general, the allocation policy π_t depends on the time instant t . However, for the special kind of stochastic behavior of the type vectors, namely SMTT, and due to the infinite horizon discounted utility, this policy becomes stationary, i.e., independent of t . We will denote such a stationary policy by $\pi = (a(\cdot), a(\cdot), \dots)$. Thus, the efficient allocation under SMTT reduces to solving for the optimal action in the following stationary Markov Decision Problem (MDP).

$$\begin{aligned} W(\theta_t) &= \max_{\pi} \mathbb{E}_{\pi, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N} v_j(a(\theta_s), \theta_s) \right] \\ &= \max_{a \in A} \mathbb{E}_{a, \theta_t} \left[\sum_{j \in N} v_j(a, \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a, \theta_t} W(\theta_{t+1}) \right]. \end{aligned} \quad (3.11)$$

Here, with a slight abuse of notation, we have used a to denote the actual action taken in t rather than the allocation function. The second equality comes from a standard recursive argument for stationary infinite horizon MDPs. We refer an interested reader to standard text [66, e.g.] for this reduction. Above we have used the following shorthand, $\mathbb{E}_{\theta_{t+1} | a, \theta_t}[\cdot] = \sum_{\theta_{t+1}} p(\theta_{t+1} | \theta_t; a)[\cdot]$. We will refer to W as the social welfare. The efficient allocation under SMTT is defined as follows.

Definition 3.7 (Efficient Allocation, EFF) *An allocation policy $a(\cdot)$ is **efficient** under SMTT if for all type profiles θ_t ,*

$$a(\theta_t) \in \operatorname{argmax}_{a \in A} \mathbb{E}_{a, \theta_t} \left[\sum_{j \in N} v_j(a, \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a, \theta_t} W(\theta_{t+1}) \right]. \quad (3.12)$$

Challenges in Mechanism Design with Interdependent Valuations

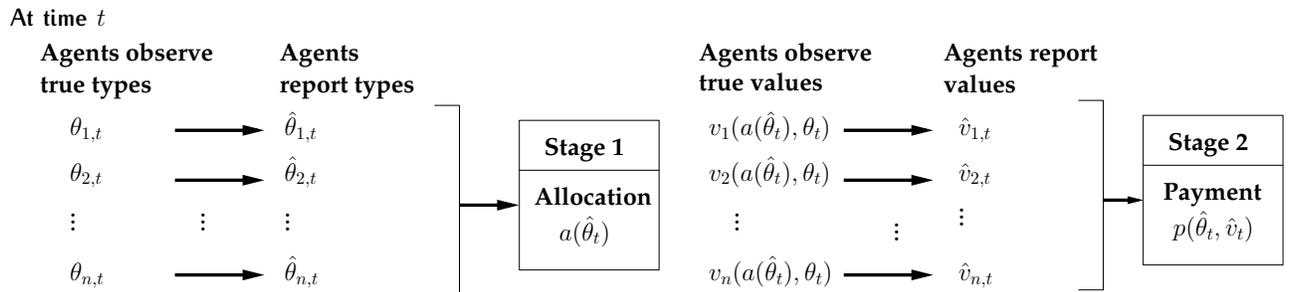


Figure 3.3: Graphical illustration of a candidate dynamic mechanism in an interdependent value setting.

The value interdependency among the agents poses a challenge for designing mechanisms. Even in a static setting, if the allocation and payment are decided simultaneously under the interdependent valuation setting, efficiency and incentive compatibility together can only be satisfied by a constant mechanism [39]. This strong negative result compels us to split the decisions of allocation and payment into two separate stages. We would mimic the two-stage mechanism of [48] for each time instant of the dynamic setting (see Figure 3.3). The designer decides the allocation $a(\hat{\theta}_t)$ after the agents report their types $\hat{\theta}_t$ in first stage. After allocation, the agents observe their valuations $v_i(a(\hat{\theta}_t), \theta_t)$'s, and report $\hat{v}_{i,t}$'s to the designer. The payment decision is made after this second round of reporting. Our definition of incentive compatibility is accordingly modified for a two stage mechanism.

Due to SMTT and the infinite horizon of the MDP, we will focus only on stationary mechanisms, that give a stationary allocation and payment to the agents in each round of the dynamic game. Let us denote a typical two-stage dynamic mechanism by $M = \langle a, p \rangle$. The function $a : \Theta \rightarrow A$ yields an allocation for a reported type profile $\hat{\theta}_t$ in round t . Depending on the reported types in the first stage, the mechanism designer decides the allocation $a(\hat{\theta}_t)$, and due to which agent i experiences a valuation of $v_i(a(\hat{\theta}_t), \theta_t)$ in round t . Let us suppose that in the second stage, the reported value vector is given by \hat{v}_t . The payment function p is a vector where $p_i(\hat{\theta}_t, \hat{v}_t)$ is the payment received by agent i at instant t . Combining the value and payment in each round we can write the expected discounted utility of agent i in the *quasi-linear* setting, denoted by $u_i^M(\hat{\theta}_t, \hat{v}_t | \theta_t)$, when the true type vector is θ_t and the reported type and value vectors are $\hat{\theta}_t$ and \hat{v}_t respectively. This utility has two parts: (a) the current round utility, and (b) expected utilities over the future rounds. The expectation over the future rounds is taken on the true types. Thus the effect of manipulation is limited only to the current round in this utility expression. This is enough to consider due to the *single deviation principle* of Blackwell [13].

$$\begin{aligned}
& u_i^M(\hat{\theta}_t, \hat{v}_t | \theta_t) \\
&= \underbrace{v_i(a(\hat{\theta}_t), \theta_t) + p_i(\hat{\theta}_t, \hat{v}_t)}_{\text{current round utility}} + \underbrace{\mathbb{E}_{\pi, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a(\theta_s), \theta_s) + p_i(\theta_s, v_s)) \right]}_{\text{expected discounted future utility}} \quad (3.13)
\end{aligned}$$

Here π denotes the stationary policy of actions, $(a(\cdot), a(\cdot), \dots)$. For the SMTT, the type evolution is dependent on only the current type profile and action. To avoid confusion, we will use π , $a(\hat{\theta}_t)$, or $a(\theta_s)$, $s \geq t + 1$, according to the context.

Equipped with these notations, we can now define *incentive compatibility*.

Definition 3.8 (w.p. EPIC) *A mechanism $M = \langle a, p \rangle$ is within period Ex-post Incentive Compatible (w.p. EPIC) if for all agents $i \in N$, for all possible true types θ_t , for all reported types $\hat{\theta}_{i,t}$, for all reported values $\hat{v}_{i,t}$, and for all t ,*

$$\begin{aligned} & u_i^M(\theta_t, (v_i(a(\theta_t), \theta_t), v_{-i}(a(\theta_t), \theta_t)) | \theta_t) \\ & \geq u_i^M((\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{v}_{i,t}, v_{-i}(a(\hat{\theta}_{i,t}, \theta_{-i,t}), \theta_t)) | \theta_t) \end{aligned}$$

That is, reporting the types and valuations in the two stages truthfully is an ex-post Nash equilibrium. In this context, *individual rationality* is defined as follows.

Definition 3.9 (w.p. EPIR) *A mechanism $M = \langle a, p \rangle$ is within period Ex-post Individually Rational (w.p. EPIR) if for all agents $i \in N$, for all possible true types θ_t and for all t ,*

$$u_i^M(\theta_t, (v_i(a(\theta_t), \theta_t), v_{-i}(a(\theta_t), \theta_t)) | \theta_t) \geq 0.$$

That is, reporting the types and valuations in the two stages truthfully yields non-negative expected utility.

3.2.3 The MATRIX Mechanism

In the setting mentioned above, our goal is to design a mechanism which is efficient (Definition 3.7), w.p. EPIC (Definition 3.8), and w.p. EPIR (Definition 3.9). Before we present the mechanism, let us discuss why it is non-trivial to design such a mechanism in this setting. We start with discussing a couple of naïve attempts to decide the allocation and the payment.

Fixed payment mechanism: A candidate mechanism that is often applied in organizations is a fixed payment mechanism. The allocation is done using the performance history of the agents. That is, select the agent(s) who has(have) been proved to be the most capable of doing the task in the past. But one can immediately notice that this fixed payment mechanism would not be efficient since the capabilities (types) of the agents vary over time. To make the correct decision on the allocation, it is important to know the realized type of the agent. Since this is private, the history will give only a (possibly incorrect) estimate of the type. Hence, this is not efficient.

Repeated static VCG mechanism: We know that the VCG mechanism is truthful in dominant strategies for static settings. The mechanism works on an efficient allocation for a single stage game, and pays each agent her marginal contribution. One can think of applying

the static VCG in each stage of the dynamic setting. However, now, social welfare is no longer the sum of the values at the current stage, rather is the expected discounted sum of the values over the horizon of the dynamic game. Hence the allocation given by the static VCG mechanism would not be efficient in a dynamic setting.

Discussion. The above two candidate mechanisms are designed to be truthful in each round, however, they fail to be efficient. It suggests that, to achieve efficient allocation in a dynamic setting, one needs to consider the expected future evolution of the types of the agents, which would reflect in the allocation and payment decisions. The value interdependency among the agents plays a crucial role here. The reason the above approach does not work is not an accident. We have already mentioned that even in a static interdependent value setting, if the allocation and payment are decided simultaneously, one cannot guarantee efficiency and incentive compatibility together [39]. One way out is to split the decision of allocation and payment into two stages [48].

Following this observation, we propose **MDP-based Allocation and TRansfer** in **Interdependent-valued eXchange economies (MATRIX)**, which we prove to satisfy EFF and w.p. EPIC for general interdependent valuations, and w.p. EPIR under the restricted setting of SA and PIV.

Given the above dynamics of the game as illustrated in Figure 3.3, the agents report their types in the first stage, and then the allocation is decided. In the second stage, they report their experienced values and the payment is decided. The task of the mechanism designer, therefore, is to design the allocation and payment rules $\langle a, p \rangle$.

We have already defined the social welfare given by Equation (3.11). Let us also define the maximum social welfare excluding agent i to be $W_{-i}(\theta_{-i,t})$, which is the same as Equation (3.11) except now the sum of the valuations and the allocations are over all agents $j \neq i$.

$$W_{-i}(\theta_{-i,t}) = \max_{a_{-i} \in A_{-i}} \mathbb{E}_{a_{-i}, \theta_{-i,t}} \left[\sum_{j \in N \setminus \{i\}} v_j(a_{-i}, \theta_{-i,t}) + \delta \mathbb{E}_{\theta_{-i,t+1} | a_{-i}, \theta_{-i,t}} W_{-i}(\theta_{-i,t+1}) \right] \quad (3.14)$$

Notice that, when i is absent, the following two notations are equivalent: $\mathbb{E}_{\theta_{t+1} | a_{-i,t}, \theta_t}[\cdot] = \mathbb{E}_{\theta_{-i,t+1} | a_{-i,t}, \theta_{-i,t}}[\cdot]$, since the type of i will be unchanged when she is not in the game. However, we adopt the former for the ease of notation.

Using the definitions above and in the previous section, now we formally present MATRIX.

Mechanism 3.1 (MATRIX) *Given the reported type profile $\hat{\theta}_t$ in stage 1, choose the agents*

$a^*(\hat{\theta}_t)$ as follows.

$$a^*(\hat{\theta}_t) \in \operatorname{argmax}_a \mathbb{E}_{a, \hat{\theta}_t} \left[\sum_{j \in N} v_j(a, \hat{\theta}_t) + \delta \mathbb{E}_{\theta_{t+1} | a, \hat{\theta}_t} W(\theta_{t+1}), \right] \quad (3.15)$$

and transfer to agent i after agents report \hat{v}_t in stage 2, a payment of,

$$p_i^*(\hat{\theta}_t, \hat{v}_t) = \left(\sum_{j \neq i} \hat{v}_{j,t} \right) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) - W_{-i}(\hat{\theta}_{-i,t}) - \left(\hat{v}_{i,t} - v_i(a^*(\hat{\theta}_t), \hat{\theta}_t) \right)^2. \quad (3.16)$$

The last quadratic term in the above equation is agent i 's penalty of not being consistent with the first stage report. The intuition of charging a penalty is to make sure that agent i be consistent with her reported type $\hat{\theta}_{i,t}$ in the first stage and her value report $\hat{v}_{i,t}$ in the second stage, given that others are reporting their types and values truthfully. We will argue that when all agents other than agent i reports their types and values truthfully in those stages, it is the best response for agent i to do so as well. This term distinguishes our mechanism from that given by Mezzetti [48], where the agents are weakly indifferent between reporting true and false values in the second round.

It is worth mentioning that we have used this quadratic term for the ease of exposition. However, it is easy to show that any non-negative function $g(x, \ell)$ having the property that $g(x, \ell) = 0 \Leftrightarrow x = \ell$ would still satisfy the claims made in this section.

Efficiency and Incentive Compatibility

We summarize the dynamics of MATRIX using an algorithmic flowchart in Algorithm 2. The following theorem shows that MATRIX satisfies two desirable properties in the unrestricted setting.

Theorem 3.2 *Under SMTT, MATRIX is EFF and w.p. EPIC. In addition, the second stage of MATRIX is strictly EPIC.*

Note that the above theorem does not put any restriction on the allocation space and the valuation functions. MATRIX is a two stage mechanism, and we need to ensure that truth-telling is a best response in both these stages when other agents also do the same. Also, the strict EPIC in the second stage of this mechanism improves upon the mechanism given by Mezzetti [48]. Let us prove the above theorem.

Algorithm 2 MATRIX

for all time instants t do

Stage 1:

for agents $i = 0, 1, \dots, n$ do

agent i observes $\theta_{i,t}$;

agent i reports $\hat{\theta}_{i,t}$;

end for

compute allocation $a^*(\hat{\theta}_t)$ according to Equation (3.15);

Stage 2:

for agents $i = 0, 1, \dots, n$ do

agent i observes $v_i(a^*(\hat{\theta}_t), \theta_t)$;

agent i reports $\hat{v}_{i,t}$;

end for

compute payment to agent i , $p_i^*(\hat{\theta}_t, \hat{v}_t)$, Equation (3.16);

types evolve $\theta_t \rightarrow \theta_{t+1}$ according to SMTT;

end for

Proof: Clearly, given true reported types, the allocation of MATRIX is efficient by Definition 3.7. Hence, we need to show only that MATRIX is w.p. EPIC.

To show that MATRIX is w.p. EPIC, let us assume that the true type profile at time t is θ_t , and all agents $j \neq i$ report their true types and values in each round $s = t, t + 1, \dots$ etc. Only agent i reports $\hat{\theta}_{i,t}$ and $\hat{v}_{i,t}$ in the two stages. Therefore, $\hat{\theta}_t = (\hat{\theta}_{i,t}, \theta_{-i,t})$ and $\hat{v}_{j,t} = v_j(a^*(\hat{\theta}_t), \theta_t)$, for all $j \neq i$. Using the *single deviation principle* [13], we conclude that it is enough to consider only a single shot deviation from the true report of the type. Hence, without loss of generality, let us assume that agent i deviates only in round t of this game.

Let us write down the discounted utility to agent i at time t .

$$\begin{aligned}
 & u_i^{\text{MATRIX}}((\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{v}_{i,t}, v_{-i}(a(\hat{\theta}_{i,t}, \theta_{-i,t}), \theta_t)) | \theta_t) \\
 &= \underbrace{v_i(a^*(\hat{\theta}_t), \theta_t) + p_i^*(\hat{\theta}_t, \hat{v}_t)}_{\text{current round utility}} + \underbrace{\mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right]}_{\text{expected discounted future utility}} \\
 &= v_i(a^*(\hat{\theta}_t), \theta_t) + \sum_{j \neq i} \hat{v}_{j,t} + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) - W_{-i}(\hat{\theta}_{-i,t}) \\
 &\quad - \left(\hat{v}_{i,t} - v_i(a^*(\hat{\theta}_t), \hat{\theta}_t) \right)^2 + \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right]
 \end{aligned}$$

We use the shorthand π^* to denote the allocation policy under MATRIX. This gives rise to the allocations $a(\cdot)$ in each round given the type profiles (either reported or true). The first

equality is from Equation (3.13). The second equality comes by substituting the expression of payment from Equation (3.16).

Now, from the previous discussion on the $\hat{v}_{j,t}$'s and $\hat{\theta}_{j,t}$'s, $j \neq i$, we get,

$$\begin{aligned}
& u_i^{\text{MATRIX}}((\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{v}_{i,t}, v_{-i}(a(\hat{\theta}_{i,t}, \theta_{-i,t}), \theta_t)) | \theta_t) \\
&= v_i(a^*(\hat{\theta}_t), \theta_t) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) - W_{-i}(\theta_{-i,t}) \\
&\quad - \left(\hat{v}_{i,t} - v_i(a^*(\hat{\theta}_t), \hat{\theta}_t) \right)^2 + \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right] \\
&\leq v_i(a^*(\hat{\theta}_t), \theta_t) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) - W_{-i}(\theta_{-i,t}) \\
&\quad + \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right] \tag{3.17}
\end{aligned}$$

The equality comes because of the assumption that all agents $j \neq i$ report their types and values truthfully. The inequality is because we are ignoring a non-positive term. Now, let us consider the last term of the above equation.

$$\begin{aligned}
& \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right] \\
&= \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) \right. \\
&\quad \left. + \sum_{j \neq i} v_j(a^*(\theta_s), \theta_s) + \delta \mathbb{E}_{\theta_{s+1} | a^*(\theta_s), \theta_s} W_{-i}(\theta_{-i,s+1}) - W_{-i}(\theta_{-i,s})) \right] \\
&= \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} \left(\sum_{j \in N} v_j(a^*(\theta_s), \theta_s) + \delta \mathbb{E}_{\theta_{s+1} | a^*(\theta_s), \theta_s} W_{-i}(\theta_{-i,s+1}) - W_{-i}(\theta_{-i,s}) \right) \right]
\end{aligned}$$

The first equality comes from Equation (3.16). We can now rearrange the expectation for the first term above using the Markov property of θ_t that gives, $\mathbb{E}_{\pi^*, \theta_t}[\cdot] = \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \theta_t}[\mathbb{E}_{\pi^*, \theta_{t+1}}[\cdot]]$. Therefore,

$$\mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} (v_i(a^*(\theta_s), \theta_s) + p_i^*(\theta_s, v_s)) \right]$$

$$\begin{aligned}
&= \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} \left[\mathbb{E}_{\pi^*,\theta_{t+1}} \left(\sum_{s=t+1}^{\infty} \delta^{s-t} \sum_{j \in N} v_j(a^*(\theta_s), \theta_s) \right) \right] \\
&\quad + \mathbb{E}_{\pi^*,\theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} \left(\delta \mathbb{E}_{\theta_{s+1}|a^*(\theta_s),\theta_s} W_{-i}(\theta_{-i,s+1}) - W_{-i}(\theta_{-i,s}) \right) \right] \\
&= \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} (\delta W(\theta_{t+1})) \\
&\quad + \mathbb{E}_{\pi^*,\theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} \left(\delta \mathbb{E}_{\theta_{s+1}|a^*(\theta_s),\theta_s} W_{-i}(\theta_{-i,s+1}) - W_{-i}(\theta_{-i,s}) \right) \right] \tag{3.18}
\end{aligned}$$

The last equality comes from the definition of $W(\theta_{t+1})$. Let us now focus on the last term of the above equation.

$$\begin{aligned}
&\mathbb{E}_{\pi^*,\theta_t} \left[\sum_{s=t+1}^{\infty} \delta^{s-t} \left(\delta \mathbb{E}_{\theta_{s+1}|a^*(\theta_s),\theta_s} W_{-i}(\theta_{-i,s+1}) - W_{-i}(\theta_{-i,s}) \right) \right] \\
&= \cancel{\delta^2 \mathbb{E}_{\pi^*,\theta_t} W_{-i}(\theta_{-i,t+2})} - \delta \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} W_{-i}(\theta_{-i,t+1}) \\
&\quad + \cancel{\delta^3 \mathbb{E}_{\pi^*,\theta_t} W_{-i}(\theta_{-i,t+3})} - \cancel{\delta^2 \mathbb{E}_{\pi^*,\theta_t} W_{-i}(\theta_{-i,t+2})} \\
&\quad \quad + \dots - \cancel{\delta^3 \mathbb{E}_{\pi^*,\theta_t} W_{-i}(\theta_{-i,t+3})} \\
&\quad \quad \quad + \dots - \dots \\
&= -\delta \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} W_{-i}(\theta_{-i,t+1}) \tag{3.19}
\end{aligned}$$

Let us show the reduction from the first term on the LHS to the first term on the RHS above. The reduction of the other terms comes from similar exercises which is straightforward and not shown here.

$$\begin{aligned}
&\mathbb{E}_{\pi^*,\theta_t} \mathbb{E}_{\theta_{t+2}|a^*(\theta_{t+1}),\theta_{t+1}} W_{-i}(\theta_{-i,t+2}) \\
&= \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} [\mathbb{E}_{\theta_{t+2}|a^*(\theta_{t+1}),\theta_{t+1}} W_{-i}(\theta_{-i,t+2})] \\
&= \mathbb{E}_{\theta_{t+1}|a^*(\hat{\theta}_t),\theta_t} [\mathbb{E}_{\theta_{t+2}|a^*(\theta_{t+1}),\theta_{t+1},\theta_t} W_{-i}(\theta_{-i,t+2})] \\
&= \mathbb{E}_{\theta_{t+2}|\pi^*,\theta_t} W_{-i}(\theta_{-i,t+2}) \\
&= \mathbb{E}_{\pi^*,\theta_t} W_{-i}(\theta_{-i,t+2})
\end{aligned}$$

The first equality above comes from the fact that the function inside bracket is only a function of θ_{t+1} , and the second equality is due to the Markov property.

Hence, combining Equations 3.17, 3.18, and 3.19, we get,

$$\begin{aligned}
& u_i^{\text{MATRIX}}((\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{v}_{i,t}, v_{-i}(a(\hat{\theta}_{i,t}, \theta_{-i,t}), \theta_t)) | \theta_t) \\
& \leq v_i(a^*(\hat{\theta}_t), \theta_t) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) \\
& \quad - W_{-i}(\theta_{-i,t}) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} [W(\theta_{t+1}) - W_{-i}(\theta_{-i,t+1})]
\end{aligned} \tag{3.20}$$

We also note that,

$$\mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} W_{-i}(\theta_{-i,t+1}) = \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \theta_t} W_{-i}(\theta_{-i,t+1}) \tag{3.21}$$

This is because when i is removed from the system (while computing $W_{-i}(\theta_{-i,t+1})$), the values of none of the other agents will depend on the type $\theta_{i,t+1}$. And due to the independence of type transitions, i 's reported type $\hat{\theta}_{i,t}$ can only influence $\theta_{i,t+1}$. Hence, the reported value of agent i at t , i.e., $\hat{\theta}_{i,t}$ cannot affect $W_{-i}(\theta_{-i,t+1})$.

Hence, Equation 3.20 can be rewritten and we can show the following inequality.

$$\begin{aligned}
& u_i^{\text{MATRIX}}((\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{v}_{i,t}, v_{-i}(a(\hat{\theta}_{i,t}, \theta_{-i,t}), \theta_t)) | \theta_t) \\
& \leq v_i(a^*(\hat{\theta}_t), \theta_t) + \sum_{j \neq i} v_j(a^*(\hat{\theta}_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \hat{\theta}_t} \cancel{W_{-i}(\theta_{-i,t+1})} \\
& \quad - W_{-i}(\theta_{-i,t}) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \theta_t} [W(\theta_{t+1}) - \cancel{W_{-i}(\theta_{-i,t+1})}] \text{ (from Equation (3.21))} \\
& = \sum_{j \in N} v_j(a^*(\hat{\theta}_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\hat{\theta}_t), \theta_t} W(\theta_{t+1}) - W_{-i}(\theta_{-i,t}) \\
& \leq \sum_{j \in N} v_j(a^*(\theta_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\theta_t), \theta_t} W(\theta_{t+1}) - W_{-i}(\theta_{-i,t}) \\
& \quad \text{(by definition of } a^*(\theta_t), \text{ Equation (3.15))} \\
& = u_i^{\text{MATRIX}}(\theta_t, (v_i(a(\theta_t), \theta_t), v_{-i}(a(\theta_t), \theta_t))) | \theta_t).
\end{aligned} \tag{3.22}$$

This shows that utility of agent i is maximized when $\hat{\theta}_{i,t} = \theta_{i,t}$ and $\hat{v}_{i,t} = v_i(a^*(\theta_t), \theta_t)$. This proves that MATRIX is within period ex-post incentive compatible.

We now argue that the second stage is strictly EPIC. This happens because of the quadratic penalty term $(\hat{v}_{i,t} - v_i(a^*(\hat{\theta}_t), \hat{\theta}_t))^2$ in the payment p_i^* (Equation (3.16)). If all the agents except i report the types and values truthfully, and agent i also reports her type truthfully in the first stage, then the penalty term will always penalize her if $\hat{v}_{i,t}$ is different from $v_i(a^*(\theta_t), \theta_t)$, which is her true valuation. Hence, the best response of agent i would be to report the true values in the second stage, which makes MATRIX strictly EPIC in this stage. \blacksquare

Why the Dynamic Pivot Mechanism would not work in this Setting

It is interesting to note that, if we tried to use the dynamic pivot mechanism (DPM) [9], unmodified in this setting, the true type profile θ_t in the first summation of Equation (3.20) would have been replaced by $\hat{\theta}_t$, since this comes from the payment term (Equation (3.16)). The proof for the DPM relies on the *private value* assumption (see the beginning of Section 3.2.2 for a definition) such that, when reasoning about the valuations for the other agents $j \neq i$, we have $v_j(a^*(\hat{\theta}_{i,t}, \theta_{-i,t}), (\hat{\theta}_{i,t}, \theta_{-i,t})) = v_j(a^*(\hat{\theta}_t), \theta_{j,t})$, with which the EPIC claim of DPM can be shown. But in the interdependent value setting, we cannot do such a substitution, and hence the proof of EPIC in DPM does not work. We have to invoke the second stage of value reporting in order to satisfy the EPIC.

Ex-post Individual Rationality for a Restricted Domain

In this section, we consider subset allocation (SA) and the values to be peer influenced (PIV). Note that now the valuation of agent i is given by $v_i(a, \theta_a)$, and the maximum social welfare would be given by,

$$\begin{aligned} W(\theta_t) &= \max_{\pi} \mathbb{E}_{\pi, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N} v_j(a(\theta_s), \theta_{a(\theta_s)}) \right] \\ &= \max_{a \in A} \mathbb{E}_{a, \theta_t} \left[\sum_{j \in N} v_j(a, \theta_a) + \delta \mathbb{E}_{\theta_{t+1} | a, \theta_t} W(\theta_{t+1}) \right] \end{aligned} \quad (3.23)$$

Similarly the maximum social welfare excluding agent i is given by,

$$\begin{aligned} W_{-i}(\theta_{-i,t}) &= \max_{a_{-i}} \mathbb{E}_{a_{-i}, \theta_t} \left[\sum_{j \in N \setminus \{i\}} v_j(a_{-i}, \theta_{a_{-i}}) + \delta \mathbb{E}_{\theta_{t+1} | a_{-i}, \theta_t} W_{-i}(\theta_{-i,t+1}) \right] \end{aligned} \quad (3.24)$$

Both these definitions are the same as Definitions 3.11 and 3.14, but redefined in this restricted domain. We now state the following theorem on individual rationality.

Theorem 3.3 (Individual Rationality) *When the allocations are chosen from class SA, values are in PIV, and types evolve in SMTT, MATRIX is w.p. EPIR.*

Proof: We observe that the allocation set is the set of subsets of N , the player set. Therefore, the set of allocations excluding agent i , denoted by $A_{-i} = 2^{N \setminus \{i\}}$, is already contained in the

set of allocations including i , denoted by $A = 2^N$. Formally, this means $a_{-i} \in A_{-i} \subseteq A \ni a$. Therefore, the policies $\pi_{-i} \in A_{-i}^\infty \subseteq A^\infty \ni \pi$. From Equation 3.11, we can write the optimal social welfare in terms of the optimal policy π^* as follows.

$$\begin{aligned}
W(\theta_t) &= \sum_{j \in N} v_j(a^*(\theta_t), \theta_{a^*(\theta_t)}) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\theta_t), \theta_t} W(\theta_{t+1}) \\
&= \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N} v_j(a^*(\theta_s), \theta_{a^*(\theta_s)}) \right]
\end{aligned} \tag{3.25}$$

Hence in the ex-post Nash equilibrium, the utility of agent i is given by,

$$\begin{aligned}
&u_i^{\text{MATRIX}}(\theta_t, (v_i(a(\theta_t), \theta_t), v_{-i}(a(\theta_t), \theta_t)) | \theta_t) \\
&= \sum_{j \in N} v_j(a^*(\theta_t), \theta_t) + \delta \mathbb{E}_{\theta_{t+1} | a^*(\theta_t), \theta_t} W(\theta_{t+1}) - W_{-i}(\theta_{-i,t}) \\
&= W(\theta_t) - W_{-i}(\theta_{-i,t}).
\end{aligned}$$

The first equality comes from the last equality in Equation 3.22. The last expression in the equation above can be written as,

$$\begin{aligned}
&W(\theta_t) - W_{-i}(\theta_{-i,t}) \\
&= \mathbb{E}_{\pi^*, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N} v_j(a^*(\theta_s), \theta_{a^*(\theta_s)}) \right] \\
&\quad - \mathbb{E}_{\pi_{-i}^*, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N \setminus \{i\}} v_j(a_{-i}^*(\theta_{-i,s}), \theta_{a_{-i}^*(\theta_{-i,s})}) \right] \\
&\geq \mathbb{E}_{\pi_{-i}^*, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N \setminus \{i\}} v_j(a_{-i}^*(\theta_{-i,s}), \theta_{a_{-i}^*(\theta_{-i,s})}) \right] \\
&\quad - \mathbb{E}_{\pi_{-i}^*, \theta_t} \left[\sum_{s=t}^{\infty} \delta^{s-t} \sum_{j \in N \setminus \{i\}} v_j(a_{-i}^*(\theta_{-i,s}), \theta_{a_{-i}^*(\theta_{-i,s})}) \right] \\
&= 0
\end{aligned} \tag{3.26}$$

The inequality holds since while choosing the optimal policy including agent i , i.e., π^* , one has the option of choosing π_{-i}^* as well, as we are in the SA domain, and the fact that the valuations

of the unallocated agents are zero, a consequence of the PIV domain. If this inequality was not true, then there would exist some $\pi_{-i}^* \in A_{-i}^\infty$ which would have achieved a social welfare more than the maximum, which is a contradiction. This proves that MATRIX is within period ex-post individually rational. ■

Complexity of Computing the Allocation and Payment

The non-strategic version of the resource to task assignment problem was that of solving an MDP, whose complexity was polynomial in the size of state-space [90]. Interestingly, for the proposed mechanism, the allocation and payment decisions are also solutions of MDPs (Equations 3.15, 3.16). Hence the proposed mechanism MATRIX has polynomial time complexity in the number of agents and size of the state-space, which is the same as that of the dynamic pivot mechanism [9].

In order to get a feel for the theory behind the mechanism MATRIX, let us illustrate the mechanism through an example in the following section.

3.2.4 Simulations

In this section, we demonstrate the properties satisfied by MATRIX through simple illustrative experiments, and compare the results with a naïve fixed payment mechanism (CONST). In addition to the already proven properties of MATRIX, we also explore two more properties here, namely payment consistency and budget balance (defined later). For brevity, we choose a relatively small example, but analyze it in detail. For the simplicity of the illustration, we simulate only the stage 1 reports, i.e., only the type reports, since in the second round MATRIX has been proved to be strictly truthful.

Experimental Setup

Let us consider an example of a task execution in a small organization, which has three agents: a task owner (center) and two production teams (agents). We index them by 0, 1, and 2. At time t , let the difficulty of the task be denoted by $\theta_{0,t}$, which is a private information of the task owner, i.e., agent 0, and the efficiencies of agents 1 and 2 be denoted by $\theta_{1,t}$ and $\theta_{2,t}$ respectively, which are private to the respective agents. We assume that each of these types can take three possible values: high (H), medium (M), and low (L). To define value functions, we associate a real number to each of these types, given by 1 (high), 0.75 (medium), and 0.5 (low). We consider the following value structure.

$$v_0(a_t, \theta_t) = \left(\frac{k_1}{\theta_{0,t}} \sum_{i \in a_t, i \neq 0} \theta_{i,t} - k_2 \right) \mathbf{1}_{0 \in a_t};$$

$$v_j(a_t, \theta_t) = -k_3 \theta_{j,t}^2 \mathbf{1}_{j \in a_t}, j = 1, 2; k_i > 0, i = 1, 2, 3.$$

The intuition behind choosing such a value function is the following. The value of the center is directly proportional to the sum total efficiency of the selected employees and inversely proportional to the difficulty of the task. For each production team, the value is negative (representing cost). It is proportional to the square of the efficiency level, representing a law of diminishing returns. Though the results in this section do not place any restriction on the value structure, we have chosen a form that is reasonable in practice. Note that the value of the center depends on the types of all the selected agents, giving rise to PIV setting. Also, because of the presence of both buyers and sellers, the market setting here is that of an exchange economy.

Type transitions are independent and follow a first order Markov chain. We choose a transition probability matrix that reflects that efficiency is likely to be reduced after a high workload round, improved after a low workload round (e.g. when a production team is not assigned).

A Naïve Mechanism (CONST)

We consider another mechanism, where the allocation decision is the same as that of MATRIX, that is, given by Equation 3.15 but the payment is a fixed constant p if the team is selected, and the task owner is charged an amount p times the number of agents selected. This mechanism satisfies, by construction, PC and BB properties. We call this mechanism CONST.

The experiment is run for an infinite horizon with discount factor $\delta = 0.7$ and the results are summarized in Figures 3.4, 3.5, and 3.6. There are 3 agents each having 3 possible types. Therefore the $3^3 = 27$ possible type profiles are represented along the x -axis of all the plots, however, it is explicitly shown only in the bottom-most plot of Figure 3.4 (the true types are denoted by the letters ‘H’, ‘M’, and ‘L’). This bottom-most plot also shows the actual (stationary) allocation for all the 27 type profiles (if the agents would have reported their types truthfully) when the allocation rule of MATRIX is applied. A ‘o’ denotes the respective agent is selected, a ‘x’ it is not.

The y -axis of the top plot in Figure 3.4 shows the utility under the payment rule of MATRIX (defined in Eq. 3.22) to agent 0 (the task owner). The y -axis of the plot in the middle shows the utility to agent 1 under the same mechanism (note that the production teams are symmetric, so it suffices to study only one). Since we are interested in ex-post equilibria, we show utilities in the setting where all other agents report truthfully, and consider the impact of misreporting by the agent under study. In these two figures, a ‘o’ represents true report, a ‘+’ denotes the utilities from a misreport. We see in Figure 3.4 that truthful reporting is a best response for

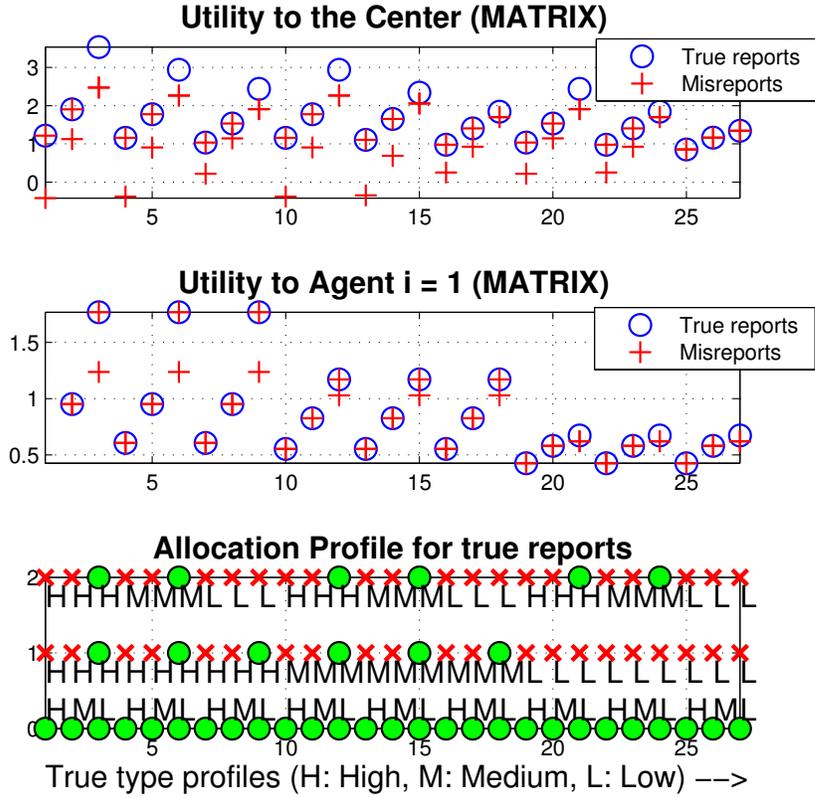


Figure 3.4: Utility of task owner and production team 1 under MATRIX as a function of true type profiles. The ordering of the $3^3 = 27$ type profiles is represented in the bottom-most plot.

both the agents (\circ 's dominates $+$'s), which illustrates the EPIC result (Theorem 3.2). Also, all the utilities under truthful reports lie above the zero (Theorem 3.3).

Figure 3.5 shows the plots similar to Figure 3.4 under the mechanism CONST. Since the allocation rule for both MATRIX and CONST are same, the bottom-most plot of Figure 3.4 would reappear as the x -axis of the plots, which we have suppressed for brevity. The plots show that the naïve method is not EPIC (for both task owner and production teams there are \circ 's lying below $+$'s).

Figure 3.6 investigates the two other properties of MATRIX: *payment consistency* (PC) and *budget balance* (BB). We call a mechanism *payment consistent* (PC) if the task owner pays and the production teams receive payment in each round. We call a mechanism *budget balanced* (BB) if the sum of the monetary transfers to all the agents is non-positive (no deficit), failing which the mechanism runs into a deficit. We observe that neither of these properties are

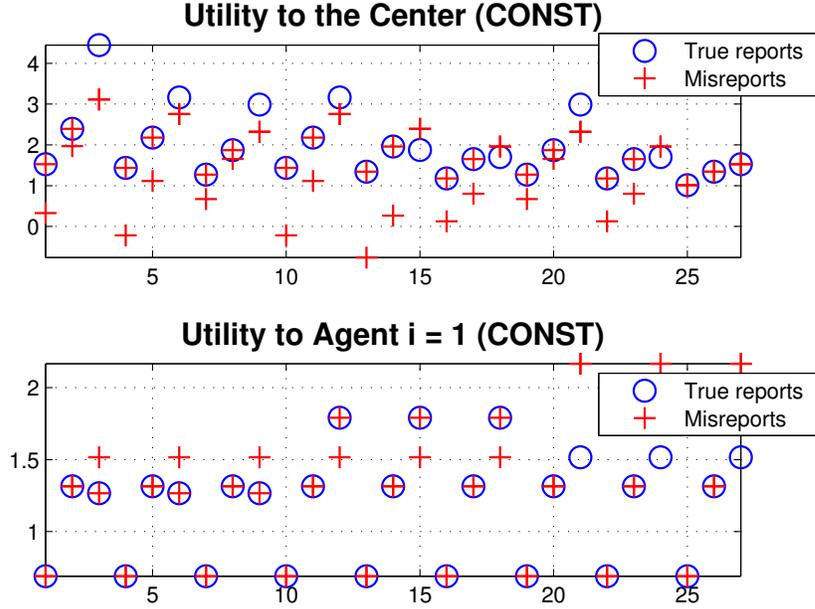


Figure 3.5: Utility of task owner and team 1 under CONST as function of true type profiles. The x-axis follows same true profile order as in Fig. 3.4.

	EFF	EPIC	EPIR	PC	BB
MATRIX	✓	✓	✓	×	×
CONST	×	×	×	✓	✓

Table 3.2: Simulation summary

satisfied for MATRIX. We summarize the results in Table 3.2.4. Not surprisingly, MATRIX satisfies three very desirable properties: EFF, EPIC, and EPIR. However, there are instances where it does not satisfy PC and BB. On the other hand, CONST satisfies PC and BB by construction, but fails to satisfy the others EFF, EPIC, and EPIR.

It seems that all these properties may not be simultaneously satisfiable in this restricted domain of dependent valued exchange economy. However, it is promising to derive bounds on payment inconsistency and budget deficit for a truthful mechanism such as MATRIX. We leave both proving the impossibility result and deriving the bounds as interesting open problems.

3.2.5 A Discussion on the Characterization of Dynamic Mechanisms

Let us consolidate our findings in this section. We have discussed the interesting and challenging domain of mechanism design with dynamically varying types and interdependent valuations.

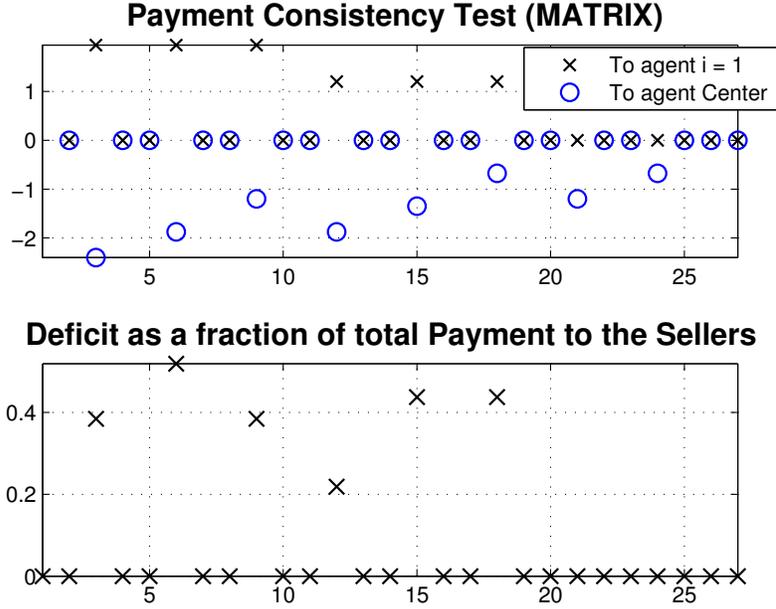


Figure 3.6: Payment consistency and budget properties of MATRIX. The x-axis follows same true profile order as in Fig. 3.4.

There is very little work where dynamic types and interdependent values have been addressed together. Hence, there is very little known on the limits of achievable properties in this domain. We have provided one mechanism, namely MATRIX, that is w.p. EPIC, strict in the second stage, and under a restricted domain, even w.p. EPIR. However, we do not know what mechanism characterizes those properties in this domain. For example, a question that may arise is “Is this the only dynamic mechanism that satisfies strict w.p. EPIC in an interdependent value setting?”. For the static setting with independent values we have the Green-Laffont characterization result that answers this question for efficiency and DSIC. However, such a characterization result is absent for interdependent valuations for both static and dynamic mechanisms. Developing such a full characterization would be worthwhile.

3.3 Conclusions

In this chapter we designed mechanisms in the interdependent valuation setting. The setting is very relevant for a collaborative task execution in crowdsourcing. In the process of designing mechanism for the crowdsourcing problem, we have also improved the solutions for general static and dynamic interdependent value mechanism design problem. In the first part, we

have looked into the static mechanism design with interdependent valuations, and improved the classic mechanism of Mezzetti [48] by making it strictly incentive compatible in the second round.

In the second part of the chapter, we designed a dynamic mechanism that is *strict* ex-post incentive compatible and efficient in an interdependent value setting with Markovian type evolution. In a restricted domain, which appears often in real-world scenarios, we showed that our mechanism is ex-post individually rational as well. This mechanism, VCPM, extends the mechanism proposed by Mezzetti [48] to a dynamic setting and connects it to the mechanism proposed by Bergemann and Välimäki [9].

Both parts open up certain interesting questions: in the first part, we improve the classic mechanism of Mezzetti by making truthful reporting in the second stage a strict Nash equilibrium, though it comes at the cost of subgame perfection of the classic mechanism. It will be interesting to investigate if these two properties are possible to satisfy together. In the second part, we have given one mechanism that satisfy a set of properties, but do not know if this is the maximal set of properties that can be satisfied by any mechanism in dynamic setting. The full set of mechanisms that satisfy the properties studied in this section is also not characterized. These questions can form interesting directions for future research.

Chapter 4

Resource Critical Task Execution via Crowdsourcing ¹

In this chapter, we consider resource critical crowdsourcing tasks and show the limits of achievability of certain desirable properties in that context. Examples include search and rescue operations like the DARPA red balloon challenge, DARPA cliqr quest, finding answers in query incentive networks, or multi-level marketing. The common thread in all these problems is that we need to incentivize the crowd to find the object/answer as well as to grow the network. Hence, the mechanism designer needs to design a mechanism that splits the reward into direct and indirect parts to incentivize both task execution and task forwarding. However, when the crowd is strategic, certain naïve mechanisms may run into problems such as fake node attacks (known as *sybil attacks* in literature), as we show through examples. In this chapter, we axiomatize some key desired properties in this kind of crowdsourcing settings, and show that certain properties are impossible to satisfy simultaneously. Therefore, we focus our attention on approximation schemes and our exploration leads to a parametrized family of payment mechanisms that characterize a certain set of desirable properties to achieve certain resource critical objectives.

An exciting application of crowdsourcing is to use social networks in complex task execution. In this chapter, we address the problem of a planner who needs to incentivize agents within a network in order to seek their help in executing an *atomic task* as well as in recruiting other agents to execute the task. We study this mechanism design problem under two natural resource optimization settings: (1) cost critical tasks, where the planner’s goal is to minimize the total cost, and (2) time critical tasks, where the goal is to minimize the total time elapsed

¹A major part of this chapter has appeared in the International Conference on Web and Internet Economics, WINE 2012 [59].

before the task is executed. We identify a set of desirable properties that should ideally be satisfied by a crowdsourcing mechanism. In particular, *sybil-proofness* and *collapse-proofness* are two complementary properties in our desiderata. We prove that no mechanism can satisfy all the desirable properties simultaneously. This leads us naturally to explore approximate versions of the critical properties. We focus our attention on approximate sybil-proofness and our exploration leads to a parametrized family of payment mechanisms which satisfy collapse-proofness. We characterize the approximate versions of the desirable properties in cost critical and time critical domain.

4.1 Introduction

Advances in the Internet and communication technologies have made it possible to harness the wisdom and efforts from a sizable portion of the society towards accomplishing tasks which are otherwise herculean. Examples include labeling millions of images, prediction of stock markets, seeking answers to specific queries, searching for objects across a wide geographical area, etc. This phenomenon is popularly known as *crowdsourcing* (for details, see Surowiecki [79] and Howe [37]). *Amazon Mechanical Turk* is one of the early examples of online crowdsourcing platform. The other example of such online crowdsourcing platforms include *oDesk*, *Rent-A-Coder*, *kaggle*, *Galaxy Zoo*, and *Stardust@home*.

In recent times, an explosive growth in online social media has given a novel twist to crowdsourcing applications where participants can exploit the *underlying social network* for inviting their friends to help executing the task. In such a scenario, the task owner initially recruits individuals from her immediate network to participate in executing the task. These individuals, apart from attempting to execute the task by themselves, recruit other individuals in their respective social networks to also attempt the task and further grow the network. An example of such applications include the DARPA *Red Balloon Challenge* [21], *DARPA CLIQR quest* [22], *query incentive networks* [41], and *multi-level marketing* [26]. The success of such crowdsourcing applications depends on providing appropriate incentives to individuals for both (1) executing the task by themselves and/or (2) recruiting other individuals. Designing a proper incentive scheme (*crowdsourcing mechanism*) is crucial to the success of any such crowdsourcing based application. In the red balloon challenge, the winning team from MIT successfully demonstrated that a crowdsourcing mechanism can be used to accomplish such a challenging task (see [65]).

A major challenge in deploying such crowdsourcing mechanisms in realistic settings is their vulnerability to different kinds of manipulations (e.g. false name attacks, also known as *sybil attacks* in the literature) that rational and intelligent participants would invariably attempt.

This challenge needs to be addressed in a specific manner for a specific application setting at the time of designing the mechanism. The application setting is characterized, primarily, by the nature of the underlying task and secondly, by the high level objectives of the designer. Depending on the nature of the underlying task, we can classify them as follows.

Viral Task. A viral task is the one where the designer’s goal is to involve as many members as possible in the social network. This kind of tasks do not have a well defined stopping criterion. Examples of such a task include viral marketing, multi-level marketing, users of a social network participating in an election, etc.

Atomic Task. An atomic task is one in which occurrence of a particular event (typically carried out by a single individual) signifies the end of the task. By definition, it comes with a well defined measure of success or accomplishment. Examples of an atomic task include the DARPA Red Balloon Challenge, DARPA CLIQR quest, query incentive networks, and transaction authentication in Bitcoin system [6].

In this chapter, we focus on the problem of designing crowdsourcing mechanisms for atomic tasks such that the mechanisms are robust to any kind of manipulations and additionally achieve the stated objectives of the designer.

DARPA Red Balloon Challenge: An Example

We provide an example to show that how resource critical crowdsourcing can lead to structural manipulation by the participants. In 2009, the defense research organization of the US, DARPA, introduced a network challenge, which is popularly known as the DARPA red balloon challenge [21]. The challenge was to find the location of 10 red weather balloons which were dispersed across the continental US, and whoever finds the correct location of all the balloons will get \$40,000 as the prize money. A snapshot of the locations of the balloons post-competition is shown in Figure 4.1.

From the geographical diversity, it is clear that the task is impossible for a single individual or even an organization. However, the winning solution came from the MIT media labs [65], who found the locations of all the balloons within 9 hours. They employed an incentive scheme, which split the total reward into 10 parts, one for each balloon, i.e., \$4000 per balloon. Now, an open call was made through the social media so that people can join MIT team’s website and through the website, they can report their information about the location of the balloon. For example, if Alice joins the network first, and she knows the location of one or more balloons, she can report it directly. If she does not, she can invite her friends whom she thinks could have the information, and the process continues through multiple levels. A typical network formed in this way is shown in Figure 4.2. The reward is shared in a bottom-up fashion. The locater

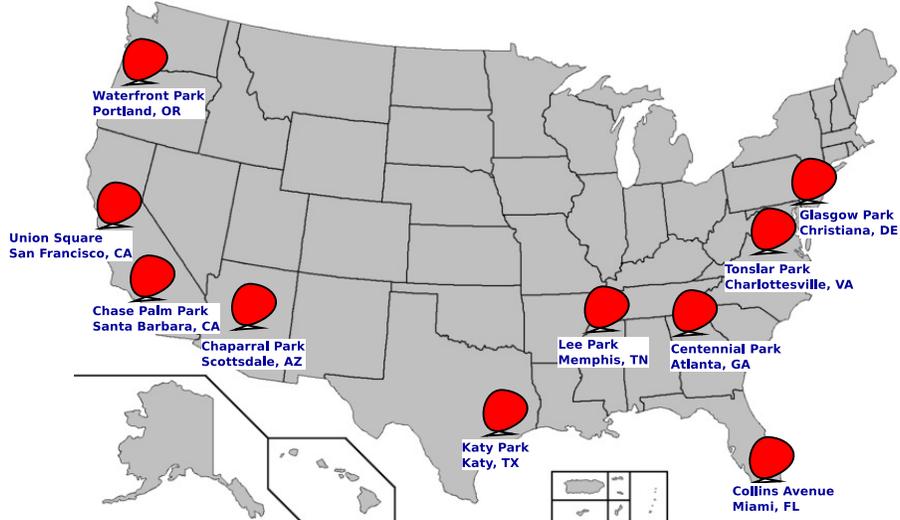


Figure 4.1: The DARPA red balloon challenge location map.

of the balloon, Dave in this case, gets \$2000 and the ancestors in the referral tree gets 0.5 times the reward of their immediate child. The scheme ensured that the referral structure becomes a tree. We notice that, in this scheme each agent has *incentive to search* for the balloons, since it gives them the maximum reward, and also *incentive to spread* the information, since it gives them some passive reward.

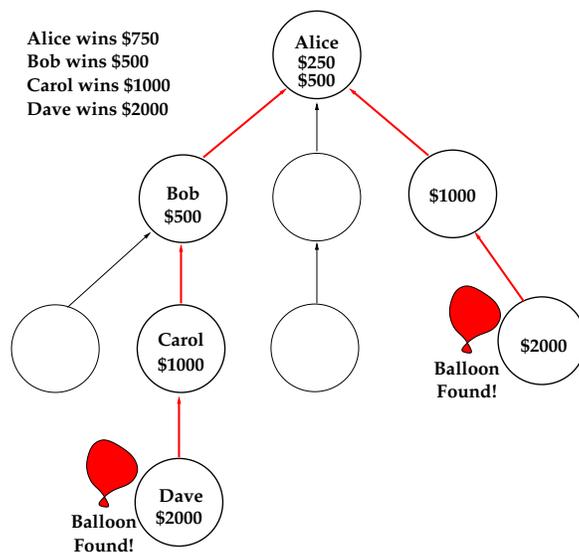


Figure 4.2: MIT team’s solution to DARPA red balloon challenge.

However, the mechanism is not totally strategyproof. An agent, e.g., Carol can manipulate the structure as shown in Figure 4.3 and gain. This kind of attacks are called *sybil attacks* in

literature. We want our mechanism to be robust against this.

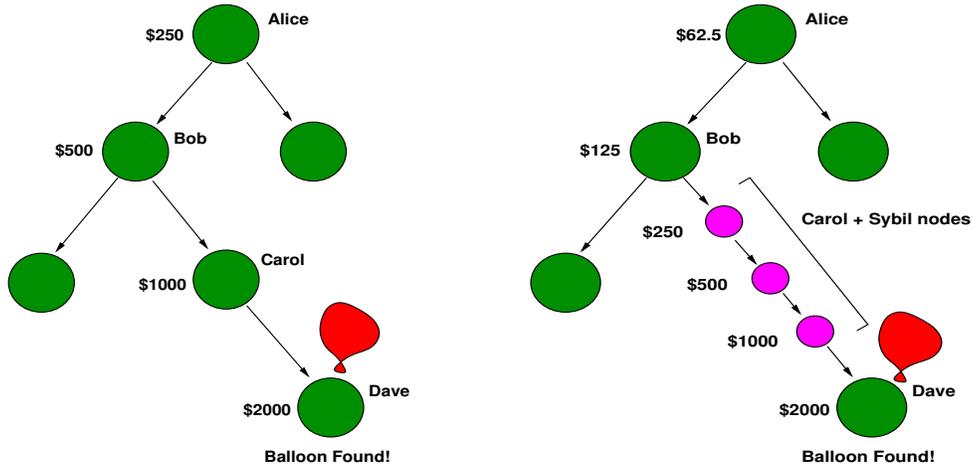


Figure 4.3: Sybil attack in the MIT scheme.

On the other hand, if we devise a naïve top-down scheme as shown in Figure 4.4, where the node at depth d from the root gets a reward of $\frac{1}{2^{d+t}}$ \$4000 if the length of the winning chain is t . It can be shown that this incentive scheme stops sybil attack, however, it introduces a new problem called the *node collapse* attack, where agents are better off by colluding with each other and reporting the location of the balloon as a consolidated single node (see Figure 4.4).

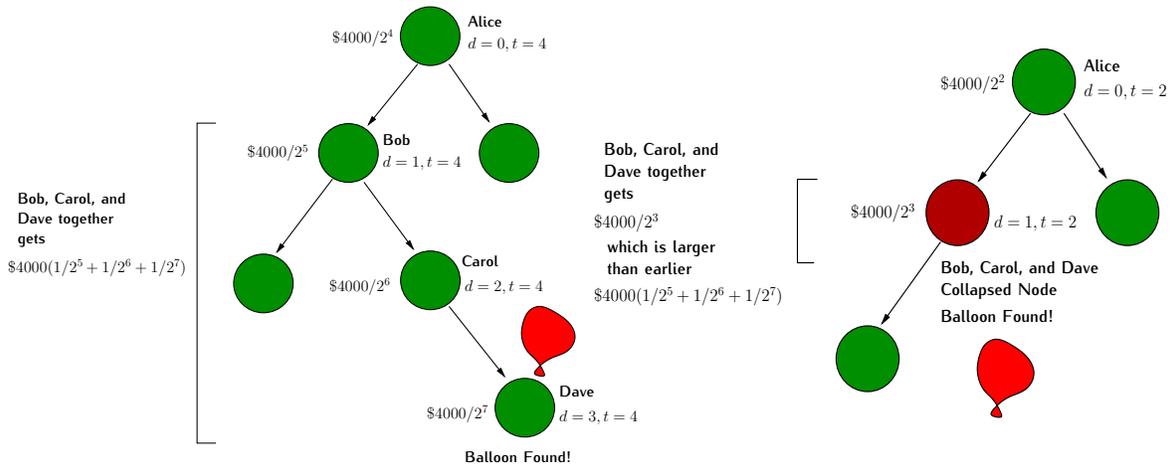


Figure 4.4: Collapse attack.

4.2 Prior Work

Prior work can be broadly classified into two categories based on the nature of the underlying task - viral or atomic.

Viral Task: The literature in this category focuses, predominantly, on the problem of multi-level marketing. Emek et al. [26] and Drucker and Fleischer [24] have analyzed somewhat similar models for multi-level marketing over a social network. In their model, the planner incentivizes agents to promote a product among their friends in order to increase the sales revenue. While Emek et al. [26] show that the geometric reward mechanism uniquely satisfies many desirable properties except false-name-proofness, Drucker and Fleischer [24] present a *capping reward mechanism* that is locally sybil-proof and collusion-proof. The collusion here only considers creating fake nodes in a collaborative way. In all multi-level marketing mechanisms, the revenue is generated *endogenously* by the participating nodes, and a fraction of the revenue is redistributed over the referrers. On slightly different kind of tasks, Conitzer et al. [20] propose mechanisms that are robust to false-name manipulation for applications such as facebook inviting its users to vote on its future terms of use. Further, Yu et al. [91] proposes a protocol to limit corruptive influence of sybil attacks in P2P networks by exploiting insights from social networks.

Atomic Task: The red-balloon challenge [21], query incentive networks [41], and transaction authentication in Bitcoin system [6] are examples of *atomic tasks*. The reward in such settings is *exogenous*, and hence the strategic problems are different from the viral tasks such as multi-level marketing. Sybil attacks still pose a problem here. Pickard et al. [65] proposed a novel solution method for Red Balloon challenge and can be considered as an early work that motivated the study of strategic aspects in crowdsourcing applications. [6] provides an *almost uniform* mechanism where sybil-proofness is guaranteed via iterated elimination of weakly dominated strategies. The work by Kleinberg and Raghavan [41] deals with a branching process based model for query incentive networks and proposes a decentralized reward mechanism for the nodes along the path from the root to the node who answers the query.

4.3 Contributions and Outline of the Chapter

In this chapter, we propose design of crowdsourcing mechanisms for atomic tasks such that the mechanisms are robust to any kind of manipulations and additionally achieve the stated objectives of the designer. Our work is distinct from the existing body of related literature in the following aspects.

(1) Collapse-Proofness: We discover that agents can exhibit an important strategic behavior, namely *node collapse attack*, which has not been explored in literature. Though the sybil attack has been studied quite well, a sybil-proof mechanism cannot by itself prevent multiple nodes colluding and reporting as a single node in order to increase their collective reward. Node collapse behavior of the agents is undesirable because, (i) it increases cost to the designer, (ii)

the distribution of this additional payment creates a situation of bargaining among the agents, hence is not suitable for risk averse agents, and (iii) it hides the structure of the actual network, which could be useful for other future purposes. A node collapse is a form of collusion, and it can be shown that the sybil-proof mechanisms presented in both [6] and [24] are vulnerable to collapse attack. In this chapter, in addition to sybil attacks, we also address the problem of *collapse attacks* and present mechanisms that are collapse-proof.

(2) Dominant Strategy Implementation: In practical crowdsourcing scenarios, we cannot expect all the agents to be fully *rational and intelligent*. We, therefore, take a complementary design approach, where instead of satisfying various desirable properties (e.g. sybil-proofness, collapse-proofness) in the Nash equilibrium sense,¹ we prefer to address a *approximate versions* of the same properties, and design *dominant strategy* mechanisms. If a mechanism satisfies an approximate version of a cheat-proof property then it means the loss in an agents' utility due to him following a non-cheating behavior is bounded (irrespective of what others are doing).

(3) Resource Optimization Criterion: The present literature mostly focuses on the design of a crowdsourcing mechanism satisfying a set of desirable cheat-proof properties. The feasible set could be quite large in many scenarios and hence a further level of optimization of the resources would be a natural extension. In this chapter, we demonstrate how to fill this gap by analyzing two scenarios - (1) cost critical tasks, and (2) time critical tasks.

A summary of our specific contributions in this chapter is as follows.

1. We identify a set of desirable properties, namely (1) *Downstream Sybil-proofness (DSP)*, (2) *Collapse-proofness (CP)*, (3) *Strict Contribution Rationality (SCR)*, (4) *Budget Balance (BB)*, and (5) *Security to Winner (SEC)*.
2. We first prove that not all properties above (not even a subset of these properties) are simultaneously satisfiable (Theorem 4.1).
3. We next prove a possibility result which shows that DSP, SCR, CP, and BB can be simultaneously satisfied but under a very restrictive mechanism (Theorem 4.2).
4. Next, we propose dominant strategy mechanisms for *approximate versions* of these properties, which is complementary to the solution provided by Babaioff et al. [6] that guarantees sybil-proofness in Nash equilibrium. In particular, we define the notion of ϵ -DSP, δ -SCR, and γ -SEC. The need for defining an approximate version of the CP property does not arise since all the proposed mechanisms satisfy exact collapse-proofness.

¹For example, the solution provided by Babaioff et al. [6] guarantees sybil-proofness only in Nash equilibrium and not in dominant strategies.

5. The approximate versions help expand the space of feasible mechanisms, leading us naturally to the following question: *How should the mechanism designer (task owner or planner) choose a particular mechanism from a bunch of possibilities?* We ask this question in two natural settings: (a) *cost critical tasks*, where the goal is to minimize the total cost, (b) *time critical tasks*, where the goal is to minimize the total time for executing the task¹. We provide characterization theorems (Theorems 4.4 and 4.5) in both the settings for the mechanisms satisfying approximate properties (ϵ -DSP, δ -SCR, and γ -SEC) in conjunction with the CP property.

To the best of our knowledge, this is the first attempt at providing approximate sybil-proofness and exact collapse-proofness in dominant strategies with certain additional fairness guarantees (δ -SCR and γ -SEC).

4.4 The Model

Consider a planner (such as DARPA) who needs to get an atomic task executed. The planner recruits a set of agents and asks them to execute the task. The recruited agents can try executing the task themselves or in turn forward the task to their friends and acquaintances who have not been offered this deal so far, thereby recruiting them into the system. If an agent receives separate invitations from multiple nodes to join their network, she can accept exactly one invitation. Thus, at any point of time, the recruited agents network is a tree. The planner stops the process as soon as the atomic task gets executed by one of the agents and offers rewards to the agents as per a centralized monetary reward scheme, say R . Let $T = (V_T, E_T)$ denote the final recruitment tree when the atomic task gets executed by one of the recruited agents. In T , the agent who executes the atomic task first is referred to as the *winner*. Let us denote the winner as $w \in V_T$. The unique path from the winner to the root is referred to as the *winning chain*. We consider the mechanisms where only winning chain receives positive payments.

For our setting, we assume that the planner designs the centralized reward mechanism R , which assigns a non-negative reward to every node in the winning chain and zero to all other nodes. Hence, we can denote the reward mechanism as a mapping $R : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}^+$ where \mathbb{N} is the set of natural numbers and \mathbb{R}^+ is the set of nonnegative reals. In such a mechanism, $R(k, t)$, $k \leq t$ denotes the reward of a node which is at depth k in the winning chain, where length of the winning chain is t . The payment is made only after completion of the task. Note, this reward mechanism is anonymous to node identities and the payment is solely dependent on

¹Note, *query incentive networks* [41] and *multi-level marketing* [26] fall under the category of cost critical tasks, while search-and-rescue operations such as red balloon challenge [21] fall under that of time critical tasks.

their position in T . Throughout this chapter, we would assume that the payment to all nodes of any *non-winning chain* is zero. Hence, all definitions of the desirable properties apply only to the winning chain.

An example of such a reward mechanism is the geometric payment used by Emek et al. [26] and Pickard et al. [65]. These mechanisms pay the largest amount to the winner node and geometrically decrease the payment over the path to the root. This class of mechanisms are susceptible to sybil attacks. For example, the winning node can create a long chain of artificial nodes, $\{x_1, \dots, x_m\}$, and report that x_i recruits x_{i+1} and x_m is the winner. Then each fake x_i would extract payment from the mechanism.

4.4.1 Desirable Properties

An ideal reward mechanism of our model should satisfy several desirable properties. In what follows, we have listed down a set of very natural properties that must be satisfied by an ideal mechanism under dominant strategy equilibrium.

Definition 4.1 (Downstream Sybilproofness, DSP) *Given the position of a node in a recruitment tree, a reward mechanism R is called downstream sybil-proof, if the node cannot gain by adding fake nodes below itself in the current subtree (irrespective of what others are doing). Mathematically,*

$$R(k, t) \geq \sum_{i=0}^n R(k+i, t+n) \quad \forall k \leq t, \forall t, n. \quad (4.1)$$

Definition 4.2 (Budget Balance, BB) *Let us assume the maximum budget allocated by the planner for executing an atomic task is R_{\max} . Then, a mechanism R is budget balanced if,*

$$\sum_{k=1}^t R(k, t) \leq R_{\max}, \quad \forall t. \quad (4.2)$$

Definition 4.3 (Contribution Rationality, CR) *This property ensures that a node gets non-negative payoff whenever she belongs to the winning chain. We distinguish between strict and weak versions of this property as defined below. For all $t \geq 1$,*

Strict Contribution Rationality (SCR):

$$R(k, t) > 0, \quad \forall k \leq t, \text{ if } t \text{ is the length of the winning chain.} \quad (4.3)$$

Weak Contribution Rationality (WCR):

$$R(k, t) \geq 0, \quad \forall k \leq t-1, \text{ if } t \text{ is the length of the winning chain.}$$

$$R(t, t) > 0, \quad \text{winner gets positive reward.} \quad (4.4)$$

DSP ensures that an agent in the network cannot gain additional payment by creating fake identities and pretending to have recruited these nodes. SCR ensures that nodes have incentive to recruit, since all members of the winning chain are rewarded.

There are many reward mechanisms that satisfy these three properties. For example, let us consider a mechanism that diminishes the rewards geometrically in both k and t , i.e. $R(k, t) = \frac{1}{2^{k+t}} \cdot R_{\max}$. This mechanism pays heavy to the nodes near the root and less near the leaf. We call this class of mechanisms as *top-down* mechanisms. This mechanism satisfies DSP, BB, and SCR properties for any finite t . However, the best response strategy of the agents in this type of mechanisms could introduce other kinds of undesirable behavior. For example, the agents of any chain would be better off by colluding among themselves and representing themselves as a single node in front of the the designer, since if the winner emerges from that particular chain, they would gain more collective reward than they could get individually. We call this *node collapse problem*. This introduces a two-fold difficulty. First, the designer cannot learn the structure of the network that executed the task, and hence cannot use the network structure for future applications. Second, she ends up paying more than what she should have paid for a true network. Hence, in the scenario where designer is also willing to minimize the expenditure, she would like to have *collapse-proofness*.

Definition 4.4 (Collapse-Proofness, CP) *Given a depth k in a winning chain, a reward mechanism R is called collapse-proof, if the subchain of length p down below k collectively cannot gain by collapsing to depth k (irrespective of what others are doing). Mathematically,*

$$\sum_{i=0}^p R(k+i, t) \geq R(k, t-p) \quad \forall k+p \leq t, \forall t. \quad (4.5)$$

In the following section, we will show that some of these properties are impossible to satisfy together. To this end, we need to define a class of mechanisms, called *Winner Takes All (WTA)*, where the winning node receives a positive reward and all other nodes get zero reward.

Definition 4.5 (WTA Mechanism) *A reward mechanism R is called WTA mechanism if $R_{\max} \geq R(t, t) > 0$, and $R(k, t) = 0, \forall k < t$.*

4.5 Impossibility and Possibility Results

Theorem 4.1 (Impossibility Result) *No reward mechanism can satisfy DSP, SCR, and CP together.*

Proof: Suppose the reward mechanism R satisfies DSP, SCR, and CP. Then by CP, let us put $t \leftarrow t + n$ and $p \leftarrow n$ in Equation 4.5, and we get, $\sum_{i=0}^n R(k + i, t + n) \geq R(k, t + n - n) = R(k, t)$, $\forall k \leq t, \forall t, n$. This is same as Equation 4.1 with the inequality reversed. So, to satisfy DSP and CP together, the inequalities reduce to the following equality.

$$R(k, t) = \sum_{i=0}^n R(k + i, t + n), \forall k \leq t, \forall t, n. \quad (4.6)$$

Now we use the following substitutions, leading to the corresponding equalities.

put $k \leftarrow t - 2, t \leftarrow t - 2, n \leftarrow 2$, to get,

$$R(t - 2, t - 2) = R(t - 2, t) + R(t - 1, t) + R(t, t) \quad (4.7)$$

put $k \leftarrow t - 1, t \leftarrow t - 1, n \leftarrow 1$, to get,

$$R(t - 1, t - 1) = R(t - 1, t) + R(t, t) \quad (4.8)$$

put $k \leftarrow t - 2, t \leftarrow t - 2, n \leftarrow 1$, to get,

$$R(t - 2, t - 2) = R(t - 2, t - 1) + R(t - 1, t - 1) \quad (4.9)$$

put $k \leftarrow t - 2, t \leftarrow t - 1, n \leftarrow 1$, to get,

$$R(t - 2, t - 1) = R(t - 2, t) + R(t - 1, t) \quad (4.10)$$

Substituting the value of Eq. 4.8 on the RHS of Eq. 4.9,

$$R(t - 2, t - 2) = R(t - 2, t - 1) + R(t - 1, t) + R(t, t) \quad (4.11)$$

Substituting Eq. 4.11 on the LHS of Eq. 4.7 yields

$$R(t - 2, t) = R(t - 2, t - 1) \quad (4.12)$$

From Eq. 4.12 and Eq. 4.10, we see that,

$$R(t - 1, t) = 0. \quad (4.13)$$

which contradicts SCR. ■

From the above theorem and the fact that additional properties reduce the space of feasible mechanisms, we obtain the following corollary.

Corollary 4.1 *It is impossible to satisfy DSP, SCR, CP, and BB together.*

Theorem 4.2 (Possibility Result) *A mechanism satisfies DSP, WCR, CP and BB iff it is a WTA mechanism.*

Proof: (\Leftarrow) It is easy to see that WTA mechanism satisfies DSP, WCR, CP and BB. Hence, it suffices to investigate the other direction.

(\Rightarrow) From Equations 4.8 and 4.13, we see that, $R(t-1, t-1) = R(t, t)$, which is true for any t . By induction on the analysis of Theorem 4.1 for length $t-1$ in place of t , we can show that $R(t-2, t-1) = 0$. But, by Eq. 4.12, $R(t-2, t-1) = R(t-2, t)$. Hence, $R(t-2, t) = 0$. Inductively, for all t and for all $k < t$, $R(k, t) = 0$. It shows that for all non-winner nodes, the reward would be zero. So, we can assign any positive reward to the winner node and zero to all others, which is precisely the WTA mechanism. This proves that for WCR, the reward mechanism that satisfies DSP, CP and BB must be a WTA mechanism. ■

4.6 Approximate Versions of Desirable Properties

The results in the previous section are disappointing in the sense that the space of mechanisms satisfying the desirable properties is extremely restricted (WTA being the only one). This suggests two possible ways out of this situation. The first route is to compromise on stronger equilibrium notion of dominant strategy and settle for a slightly weaker notion such as Nash equilibrium. The other route could be to weaken these stringent properties related to cheat-proofness and still look for a dominant strategy equilibrium. We choose to go by the latter way because Nash equilibrium makes the assumption that all the players can efficiently compute and find the equilibrium. This assumption may not be true in practical crowdsourcing applications. Therefore, we relax some of the desirable properties to derive their approximate versions. We begin with approximation of the DSP property.

Definition 4.6 (ϵ - Downstream Sybilproofness, ϵ -DSP) *Given the position of the node in a tree, a payment mechanism R is called ϵ - DSP, if the node cannot gain by more than a factor of $(1 + \epsilon)$ by adding fake nodes below herself in the current subtree (irrespective of what others are doing). Mathematically,*

$$(1 + \epsilon) \cdot R(k, t) \geq \sum_{i=0}^n R(k+i, t+n), \forall k \leq t, \forall t, n. \quad (4.14)$$

Theorem 4.3 *For all $\epsilon > 0$, there exists a mechanism that is ϵ -DSP, CP, BB, and SCR.*

Proof: The proof is constructive. Let us consider the following mechanism: set $R(t, t) = (1 - \delta) \cdot R_{\max}$, $\forall t$, the reward to the winner, where $\delta \leq \frac{\epsilon}{1+\epsilon}$. Also, let $R(k, t) = \delta \cdot R(k+1, t) =$

$\delta^{t-k} \cdot R(t, t) = \delta^{t-k}(1 - \delta)R_{\max}, k \leq t - 1$. By construction, this mechanism satisfies BB. It is also SCR, since $\delta \in (0, 1)$. It remains to show that this satisfies ϵ -DSP and CP. Let us consider,

$$\begin{aligned} \sum_{i=0}^n R(k+i, t+n) &= \sum_{i=0}^n \delta^{t+n-k-i} \cdot R(t+n, t+n) \\ &= \delta^{t-k} \cdot (1 + \delta + \dots + \delta^n) \cdot (1 - \delta)R_{\max} \\ &= R(k, t) \cdot (1 + \delta + \dots + \delta^n) \\ &\leq R(k, t) \cdot \frac{1}{1 - \delta} \leq (1 + \epsilon) \cdot R(k, t), \quad \text{since } \delta \leq \frac{\epsilon}{1 + \epsilon}. \end{aligned}$$

This shows that this mechanism is ϵ -DSP. Also,

$$\begin{aligned} \sum_{i=0}^p R(k+i, t) &= \sum_{i=0}^p \delta^{t-k-i} \cdot R(t, t) \\ &= \sum_{i=0}^{p-1} \delta^{t-k-i} \cdot R(t, t) + \underbrace{\delta^{t-k-p} \cdot R(t, t)}_{=R(k, t-p)} \geq R(k, t-p) \end{aligned}$$

This shows that this mechanism is CP as well. ■

Discussions:

- Above theorem suggests that merely weakening the DSP property allows a way out of the impossibility result given in Theorem 4.1. One can try weakening the CP property analogously (instead of DSP) and check for the possibility/impossibility results. This we leave as an interesting future work.
- One may argue that no matter how small is ϵ , as long as we satisfy ϵ -DSP property, an agent would always find it beneficial to add as many sybil nodes as possible. However, in real crowdsourcing networks, there would be a non-zero cost involved in creating fake nodes and hence there must be a tipping point so that agent's net gain would increase till he creates that many sybil nodes but starts declining after that. Note, it is impossible for an agent to compute the tipping point a priori as his own reward is uncertain at the time of him getting freshly recruited by someone and he trying to create sybil nodes. Therefore, in the face of this uncertainty, the agent can assure himself of a bounded regret if he decides not to create any sybil nodes.

4.6.1 Motivation for δ -SCR and γ -SEC

As per previous theorem, the class of mechanisms that satisfy ϵ -DSP, CP, BB, and SCR is quite rich. However, the exemplar mechanism of this class, which was used in the proof of

this theorem, prompts us to think of the following undesirable consequence - the planner can assign arbitrarily low reward to the winner node and still manage to satisfy all these properties. This could discourage the agents from putting in effort by themselves for executing the task. Motivated by this considerations, we further extend the SCR property by relaxing it to δ -SCR and also introduce an additional property, namely *Winner's γ Security (γ -SEC)*.

Definition 4.7 (δ - Strict Contribution Rationality, δ -SCR) *This ensures that a node in the winning chain gets at least $\delta \in (0, 1)$ fraction of her successor. Also the the winner gets a positive reward. For all $t \geq 1$,*

$$\begin{aligned} R(k, t) &\geq \delta R(k + 1, t), \forall k \leq t - 1, \quad t: \text{winning chain.} \\ R(t, t) &> 0, \quad \text{winner gets positive reward.} \end{aligned} \tag{4.15}$$

Definition 4.8 (Winner's γ Security, γ -SEC) *This ensures that payoff to the winning node is at least γ fraction of the total available budget.*

$$R(t, t) \geq \gamma \cdot R_{\max}, \quad t \text{ is the length of the winning chain} \tag{4.16}$$

Discussions:

- The δ -SCR property guarantees that recruiter of each agent on the winning chain gets a certain fraction of the agent's reward. This property will encourage an agent to propagate the message to her acquaintances even though she may not execute the task by herself. This would result in rapid growth of the network which is desirable in many settings.
- On the other hand, γ -SEC ensures that the reward to the winner remains larger than a fraction of the total reward. This works as a motivation for any agent to spend effort on executing the task by herself.

In what follows, we characterize the space of mechanisms satisfying these properties.

4.7 Cost Critical Tasks

In this section, we design crowdsourcing mechanisms for the atomic tasks where the planner's objective is to minimize total cost of executing the task.

Definition 4.9 (MINCOST over \mathcal{C}) *A reward mechanism R is called MINCOST over a class of mechanisms \mathcal{C} , if it minimizes the total reward distributed to the participants in the winning*

chain. That is, R is MINCOST over \mathcal{C} , if

$$R \in \operatorname{argmin}_{R' \in \mathcal{C}} \sum_{k=1}^t R'(k, t), \quad \forall t. \quad (4.17)$$

We will show that the MINCOST mechanism over the space of ϵ -DSP, δ -SCR, and BB properties is completely characterized by a simple geometric mechanism, defined below.

Definition 4.10 ((γ, δ)-Geometric Mechanism, (γ, δ)-GEOM) *This mechanism gives γ fraction of the total reward to the winner and geometrically decreases the rewards towards root with the factor δ . For all t , $R(t, t) = \gamma \cdot R_{\max}$; $R(k, t) = \delta \cdot R(k+1, t) = \delta^{t-k} \cdot R(t, t) = \delta^{t-k} \cdot \gamma R_{\max}$, $k \leq t-1$.*

4.7.1 Characterization Theorem for MINCOST

Now, we will show that (γ, δ)-Geometric mechanism characterizes the space of MINCOST mechanisms satisfying ϵ -DSP, δ -SCR, γ -SEC, and BB. We start with an intermediate result.

Lemma 4.1 *A mechanism is δ -SCR, γ -SEC and BB only if $\gamma \leq 1 - \delta$.*

Proof: Suppose $\gamma > 1 - \delta$. Then by δ -SCR, we have,

$$\sum_{k=1}^t R(k, t) \geq (1 + \delta + \dots + \delta^{t-1}) \cdot R(t, t) \quad (4.18)$$

$$\geq (1 + \delta + \dots + \delta^{t-1}) \cdot \gamma R_{\max} \quad (4.19)$$

$$> (1 + \delta + \dots + \delta^{t-1})(1 - \delta)R_{\max}$$

This holds for all $t \geq 1$. It must hold for $t \rightarrow \infty$. Hence, $\lim_{t \rightarrow \infty} \sum_{k=1}^t R(k, t) > \frac{1}{1-\delta} \cdot (1 - \delta)R_{\max} = R_{\max}$. Which is a contradiction to BB. \blacksquare

Theorem 4.4 *If $\delta \leq \min\{1 - \gamma, \frac{\epsilon}{1+\epsilon}\}$, a mechanism is MINCOST over the class of mechanisms satisfying ϵ -DSP, δ -SCR, γ -SEC, and BB iff it is (γ, δ)-GEOM mechanism.*

Proof: (\Leftarrow) It is easy to see that (γ, δ)-GEOM is δ -SCR and γ -SEC by construction. It is also BB since $\delta \leq 1 - \gamma$ or $\gamma \leq 1 - \delta$. For the ϵ -DSP property, we consider the following expression,

$$\sum_{i=0}^n R(k+i, t+n) = \sum_{i=0}^n \delta^{t+n-k-i} \cdot R(t+n, t+n),$$

which is equal to,

$$\begin{aligned} \delta^{t-k} \cdot (1 + \delta + \dots + \delta^n) \cdot \gamma R_{\max} &= R(k, t) \cdot (1 + \delta + \dots + \delta^n) \\ &\leq R(k, t) \cdot \frac{1}{1 - \delta} \leq (1 + \epsilon) R(k, t), \text{ as } \delta \leq \frac{\epsilon}{1 + \epsilon}. \end{aligned}$$

Also for a given δ and γ , this mechanism minimizes the total cost as it pays each node the minimum possible reward. Thus, δ -GEOM mechanism is MINCOST over ϵ -DSP, δ -SCR, γ -SEC, and BB.

(\Rightarrow) Since $\delta \leq 1 - \gamma$, from Lemma 4.1, we see that δ -SCR, γ -SEC, and BB are satisfiable. In addition the objective of the mechanism designer is to minimize the total reward (R_{total}) given to the winning chain.

$$R_{total} = \sum_{k=1}^t R(k, t) \stackrel{\text{Eq. 4.19}}{\geq} (1 + \delta + \dots + \delta^{t-1}) \cdot \gamma R_{\max}$$

We require a mechanism that is also ϵ -DSP and minimizes the above quantity. Let us consider a mechanism R_1 that pays the leaf an amount of γR_{\max} and any other node at depth k , an amount $\delta^{t-k} \gamma R_{\max}$. We ask the question if this mechanism is ϵ -DSP. This is because if this is true, then there cannot be any other mechanism that minimizes the cost, as this achieves the lower bound of R_{total} . To check for ϵ -DSP of this mechanism, we consider the following expression.

$$\begin{aligned} \sum_{i=0}^n R_1(k+i, t+n) &= \sum_{i=0}^n \delta^{t+n-k-i} \cdot R_1(t+n, t+n) \\ &= \delta^{t-k} \cdot (1 + \delta + \dots + \delta^n) \cdot \gamma R_{\max} \\ &= R_1(k, t) \cdot (1 + \delta + \dots + \delta^n) \\ &\leq R_1(k, t) \cdot \frac{1}{1 - \delta} \leq (1 + \epsilon) R_1(k, t) \quad \text{since } \delta \leq \frac{\epsilon}{1 + \epsilon} \end{aligned}$$

implying R_1 is also ϵ -DSP. Hence, R_1 is *the* MINCOST mechanism over ϵ -DSP, δ -SCR, γ -SEC, and BB. Note, R_1 is precisely the (γ, δ) -GEOM mechanism. ■

Discussions:

- Note, (γ, δ) -GEOM mechanism additionally satisfies CP. The proof for this is given in the Appendix.
- Theorem 4.4 imposes a constraint on the values of the parameters δ , ϵ , and γ , for which

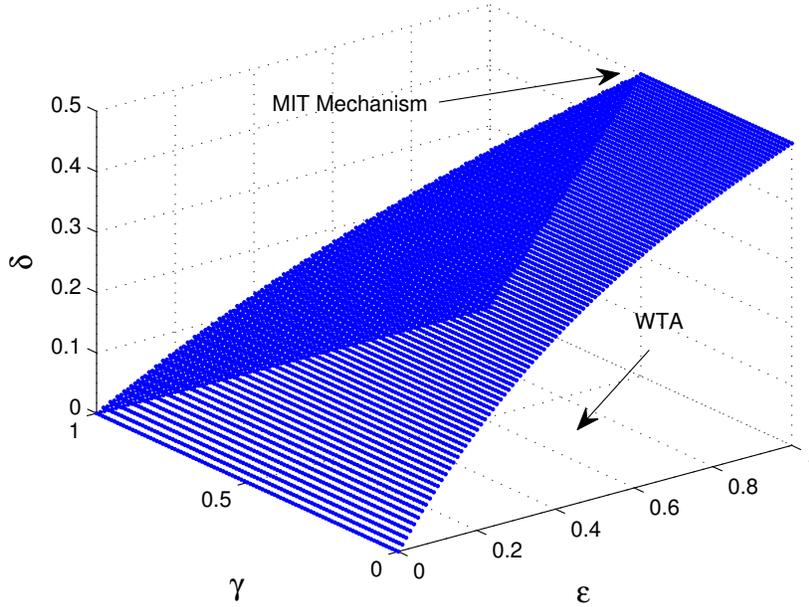


Figure 4.5: The space of $(\delta, \epsilon, \gamma)$ tuples characterized by Theorem 4.4.

the characterization result holds. Let us define,

$$\mathcal{E} = \{(\delta, \epsilon, \gamma) : \delta \leq \min\{1 - \gamma, \frac{\epsilon}{1 + \epsilon}\}\}.$$

The space \mathcal{E} is graphically illustrated in the Figure 4.5.

- The set \mathcal{E} of $(\delta, \epsilon, \gamma)$ tuples that are characterizable via Theorem 4.4 is the space below the shaded region in Figure 4.5. MIT mechanism ($\epsilon = 1, \delta = 0.5, \gamma = 0.5$) and the WTA mechanism ($\delta = 0$, the floor of the space in the figure above) are special cases. Theorem 4.4 says that if $(\delta, \epsilon, \gamma) \in \mathcal{E}$, the characterization result in that theorem holds.

4.8 Time Critical Tasks

In applications where the faster growth of network is more important than maximizing the surplus, the designer can spend the whole budget in order to incentivize participants to either search for the answer or forward the information quickly among their acquaintances. In such settings, we can design mechanisms which aim to maximize reward of the leaf node of the winning chain. In this section, we show that such kind of mechanisms with the same fairness guarantees can also be characterized by a similar mechanism that exhausts the budget even for

a finite length of the winning chain. In what follows, we define the design goal and a specific geometric mechanism.

Definition 4.11 (MAXLEAF over \mathcal{C}) A reward mechanism R is called **MAXLEAF** over a class of mechanisms \mathcal{C} , if it maximizes the reward of the leaf node in the winning chain. That is, R is **MAXLEAF** over \mathcal{C} , if

$$R \in \operatorname{argmax}_{R' \in \mathcal{C}} R'(t, t), \quad \forall t. \quad (4.20)$$

Definition 4.12 (δ -Geometric mechanism, δ -GEOM) This mechanism gives $\frac{1-\delta}{1-\delta^t}$ fraction of the total reward to the winner and geometrically decreases the rewards towards root with the factor δ , where t is the length of the winning chain. For all t , $R(t, t) = \frac{1-\delta}{1-\delta^t} \cdot R_{\max}$; $R(k, t) = \delta \cdot R(k+1, t) = \delta^{t-k} \cdot R(t, t)$, $k \leq t-1$.

4.8.1 Characterization Theorem for MAXLEAF

Theorem 4.5 If $\delta \leq \frac{\epsilon}{1+\epsilon}$, a mechanism is **MAXLEAF** over the class of mechanisms satisfying ϵ -DSP, δ -SCR, and BB iff it is δ -GEOM mechanism.

Proof: (\Leftarrow) By construction, the δ -GEOM mechanism is δ -SCR and BB for all t . It is also ϵ -DSP, as,

$$\begin{aligned} \sum_{i=0}^n R(k+i, t+n) &= \sum_{i=0}^n \delta^{t+n-k-i} \cdot R(t+n, t+n) \\ &= \delta^{t-k} \cdot (1 + \delta + \dots + \delta^n) \cdot R(t+n, t+n) \\ &= \delta^{t-k} R(t, t) \cdot \frac{R(t+n, t+n)}{R(t, t)} \cdot \frac{1 - \delta^{n+1}}{1 - \delta} \\ &= R(k, t) \cdot \frac{R(t+n, t+n)}{R(t, t)} \cdot \frac{1 - \delta^{n+1}}{1 - \delta} \\ &= R(k, t) \cdot \frac{\frac{1-\delta}{1-\delta^{t+n}} \cdot R_{\max}}{\frac{1-\delta}{1-\delta^t} \cdot R_{\max}} \cdot \frac{1 - \delta^{n+1}}{1 - \delta} \\ &= R(k, t) \cdot \frac{1 - \delta^{n+1}}{1 - \delta^{t+n}} \cdot \frac{1 - \delta^t}{1 - \delta}. \end{aligned}$$

Since $\frac{1-\delta^{n+1}}{1-\delta^{t+n}} \uparrow n$ and $\frac{1-\delta^t}{1-\delta} \uparrow t$, we can take limits as $n \rightarrow \infty$ and $t \rightarrow \infty$ respectively to get an upper bound on the quantity of the RHS, which gives,

$$\sum_{i=0}^n R(k+i, t+n) = R(k, t) \cdot \frac{1}{1-\delta} \leq (1+\epsilon) \cdot R(k, t),$$

since $\delta \leq \frac{\epsilon}{1+\epsilon}$. Hence this is ϵ -DSP. Suppose this is not **MAXLEAF**. Then \exists some other mechanism R' in the same class that pays $R'(t, t) > \frac{1-\delta}{1-\delta^t} \cdot R_{\max}$. Since, R' is also δ -SCR,

$$\begin{aligned} \sum_{k=1}^t R'(k, t) &\geq (1 + \delta + \dots + \delta^{t-1}) \cdot R'(t, t) \\ &= \frac{1 - \delta^t}{1 - \delta} \cdot R'(t, t) > \frac{1 - \delta^t}{1 - \delta} \cdot \frac{1 - \delta}{1 - \delta^t} \cdot R_{\max} = R_{\max}, \end{aligned}$$

which is a contradiction to BB. Hence proved.

(\Rightarrow) Let R be a mechanism that is **MAXLEAF** over the class of mechanisms satisfying ϵ -DSP, δ -SCR, and BB. Hence,

$$\begin{aligned} R_{\max} &\geq \sum_{k=1}^t R(k, t) \stackrel{\text{Eq. 4.18}}{\geq} \frac{1 - \delta^t}{1 - \delta} \cdot R(t, t) \\ \Rightarrow R(t, t) &\leq \frac{1 - \delta}{1 - \delta^t} \cdot R_{\max}, \quad \text{for all } t. \end{aligned} \tag{4.21}$$

The first and second inequalities arise from BB and δ -SCR respectively. Now, from the ϵ -DSP condition of R , we get, for all $n, t, k \leq t$,

$$\begin{aligned} (1 + \epsilon)R(k, t) &\geq \sum_{i=0}^n R(k + i, t + n) \\ &\geq \sum_{i=0}^n \delta^{t+n-k-i} \cdot R(t + n, t + n) \\ &= \delta^{t-k} \cdot (1 + \delta + \dots + \delta^n) \cdot R(t + n, t + n), \end{aligned}$$

where the second inequality comes from δ -SCR of R . Rearranging, we obtain,

$$1 + \epsilon \geq \delta^{t-k} \cdot \frac{1 - \delta^{n+1}}{1 - \delta} \cdot \frac{R(t + n, t + n)}{R(k, t)} \tag{4.22}$$

Since this is a necessary condition for any $k \leq t$, it should hold for $k = t$ in particular. Using this in Equation 4.22 the necessary condition becomes,

$$1 + \epsilon \geq \frac{1 - \delta^{n+1}}{1 - \delta} \cdot \frac{R(t + n, t + n)}{R(t, t)} \tag{4.23}$$

Now, we have two conditions on $R(t + n, t + n)$ as follows.

$$R(t + n, t + n) \leq (1 + \epsilon) \cdot \frac{1 - \delta}{1 - \delta^{n+1}} \cdot R(t, t)$$

$$\stackrel{\text{Eq. 4.21}}{\leq} \underbrace{(1 + \epsilon) \cdot \frac{1 - \delta}{1 - \delta^{n+1}} \cdot \frac{1 - \delta}{1 - \delta^t} \cdot R_{\max}}_{=:A(n,t)} \quad (4.24)$$

and using Eq. 4.21 directly on $R(t + n, t + n)$, we get,

$$R(t + n, t + n) \leq \underbrace{\frac{1 - \delta}{1 - \delta^{t+n}} \cdot R_{\max}}_{=:B(n,t)} \quad (4.25)$$

It is clear that to satisfy δ -SCR, ϵ -DSP and BB, it is necessary for R to satisfy,

$$R(t + n, t + n) \leq \min_{n,t} \{A(n, t), B(n, t)\}.$$

We can show the following bounds for the quantity $\frac{B(n,t)}{A(n,t)}$, which we skip due to space constraints.

$$\frac{1}{1 + \epsilon} \leq \frac{B(n, t)}{A(n, t)} \leq \frac{1}{(1 + \epsilon)(1 - \delta)}. \quad (4.26)$$

Since $\delta \leq \frac{\epsilon}{1 + \epsilon}$, we see that the upper bound $\frac{1}{(1 + \epsilon)(1 - \delta)} \leq 1$. Hence, $A(n, t)$ uniformly dominates $B(n, t)$, $\forall n, t$. Hence, $R(t + n, t + n) \leq B(n, t)$. Since R is also MAXLEAF, equality must hold and it must be true that,

$$R(t, t) = \frac{1 - \delta}{1 - \delta^t} \cdot R_{\max}, \forall t. \quad (4.27)$$

Also, since R is BB, it is necessary that,

$$R(k, t) = \delta^{t-k} \cdot R(t, t), \quad k \leq t - 1. \quad (4.28)$$

This shows that R has to be δ -GEOM. ■

Discussion: A δ -GEOM mechanism also satisfies CP property. The proof for this is given in the Appendix.

4.9 Partial Information Aggregation: Synergy among the Agents

So far in this chapter we have discussed only atomic tasks where the reward is distributed only on the winning chain, i.e., the chain that found the object. A more time efficient and fair scheme could be to distribute the reward in proportion to the information contributed by each agent. For example, if an agent searches for an object in a certain part of a city and

reports that the object is not present in that region, it saves a lot of time and effort for the other participants assuming that the original reporter was truthful. In our current work [60], we are exploring how to incentivize and exploit the synergy between participants satisfying certain desirable properties of crowdsourcing. We employ the machinery of prediction market in order to achieve two objectives: (a) agents can express their confidence in reporting certain information by trading in the information market, and (b) the reward can be shared by all participants who gave *useful* information towards the goal. In general, prediction market allows for certain properties like truthfulness and arbitrage-free information aggregation which is very desirable in the crowdsourcing application as well.

4.10 Conclusions

In this chapter, we have studied the problem of designing manipulation free crowdsourcing mechanisms for atomic tasks under the cost critical and time critical scenarios. We have motivated the need for having CP as an additional property of the mechanism beyond what already exists in the literature. Starting with an impossibility result, we have developed mechanisms for both cost and time critical scenarios which satisfy CP property along with weaker versions of other desirable properties (all under dominant strategy equilibrium) that characterize the space of feasible mechanisms in this setting. We have also made some progress on how the partial information can be processed to make an aggregation of information with some design desiderata. This opens up several bigger questions both in crowdsourcing and in prediction market on how raw information can be converted into distributions over outcomes.

APPENDIX

4.A Proof of Two Lemmata

Lemma 4.2 *The (γ, δ) -GEOM Mechanism is Collapse-Proof.*

Proof: For the mechanism (γ, δ) -GEOM, the reward of the leaf node is independent of the length of the winning chain, i.e., t . Hence, we can use the same proof technique used in Theorem 4.3 to show that (γ, δ) -GEOM is CP. The steps are as follows.

$$\begin{aligned} \sum_{i=0}^p R(k+i, t) &= \sum_{i=0}^p \delta^{t-k-i} \cdot R(t, t) \\ &= \sum_{i=0}^{p-1} \delta^{t-k-i} \cdot R(t, t) + \underbrace{\delta^{t-k-p} \cdot R(t, t)}_{=R(k, t-p)} \geq R(k, t-p) \end{aligned}$$

This shows that this mechanism is CP. ■

Lemma 4.3 *The δ -GEOM Mechanism is Collapse-Proof.*

Proof: In order to prove the claim, we need to show that,

$$\begin{aligned}
& \sum_{i=0}^p R(k+i, t) \geq R(k, t-p) \quad \forall k+p \leq t, \forall t \\
\Rightarrow & \sum_{i=0}^p \delta^{t-k-i} \cdot R(t, t) \geq \delta^{t-k-p} \cdot R(t-p, t-p) \\
\Rightarrow & \sum_{i=0}^p \delta^{t-k-i} \cdot \frac{1-\delta}{1-\delta^t} \cdot R_{\max} \geq \delta^{t-k-p} \cdot \frac{1-\delta}{1-\delta^{t-p}} \cdot R_{\max} \\
\Rightarrow & (1-\delta^{t-p}) \sum_{i=0}^p \delta^{t-k-i} \geq \delta^{t-k-p} (1-\delta^t) \\
\Rightarrow & (1-\delta^{t-p})(\delta^{t-k-p} + \dots + \delta^{t-k}) \geq \delta^{t-k-p} - \delta^{2t-k-p} \\
\Rightarrow & \cancel{\delta^{t-k-p}} + \dots + \delta^{t-k} - (\delta^{2t-2p-k} + \dots \\
& \quad \quad \quad + \cancel{\delta^{2t-p-k}}) \geq \cancel{\delta^{t-k-p}} - \cancel{\delta^{2t-k-p}} \\
\Rightarrow & (\delta^{t-k} + \dots + \delta^{t-k-p+1}) - (\delta^{2t-2p-k} + \dots \\
& \quad \quad \quad \delta^{2t-p-k-1}) \geq 0. \tag{4.A.1}
\end{aligned}$$

Where both the terms within the parentheses on the LHS are in ascending order. We can rewrite inequality (4.A.1) as,

$$\delta^{t-k-p} [(\delta^p - \delta^{t-p}) + (\delta^{p-1} - \delta^{t-p+1}) + \dots + (\delta^1 - \delta^{t-1})] \geq 0.$$

Let us define, $a_k^t := \delta^k - \delta^{t-k}$. Therefore, from the above inequality, we need to show that,

$$\sum_{k=1}^p a_k^t \geq 0. \tag{4.A.2}$$

This is a partial sum of the complete sum, $\sum_{k=1}^{t-1} a_k^t$ which equals zero. Since, $\delta < 1$, we also see that,

$$\begin{aligned}
a_k^t & \geq 0 & \forall k = 1, \dots, \lfloor t/2 \rfloor, \\
a_k^t & \leq 0 & \forall k = \lfloor t/2 \rfloor + 1, \dots, t-1.
\end{aligned}$$

For any p , the partial sum would be non-negative. So, inequality (4.A.2) holds for any p . Therefore, we have shown that δ -GEOM is CP. ■

Chapter 5

Efficient Team Formation using a Strategic Crowd

Many crowdsourcing applications leverage the social connectivity of the individuals by allowing them to invite their acquaintances and by offering them a reward share from their invitees. The result is the formation of a network of referrals, which can be either hierarchical or non-hierarchical. If we consider the whole crowdsourcing system as a consolidated organization, a primary goal of the crowdsourcer is to maximize the net productive output of this network. Every individual in these networks has a limited amount of effort that they can split between productive and communicating components, and they trade off these two components depending on their positions and share of rewards in the network. Since the efficiency of the crowdsourcing network depends on both these components, it is important to understand how the agents trade off between them considering that their goal is to maximize their own individual payoffs.

In this chapter, we capture this phenomenon using a game theoretic modeling. We model each agent's payoff as a sum of the direct reward obtained from her own production effort and the indirect reward obtained from the efforts of communicating the task to others. We identify two factors for an optimal trade-off of efforts: (a) the reward sharing policy and (b) the network structure. We investigate how these two factors influence the trade-off and how these factors can be better designed to maximize the net productive output of the system. Our results show that under the strategic behavior of the agents, it may not always be possible to achieve an optimal output and we provide bounds on the achievability in such scenarios.

We consider two approaches towards efficient team formation in crowdsourcing. The first, discussed next, considers the reward sharing between the *influencers* and *influencees*. The

second considers designing the network and is discussed later in this chapter. Since we are modeling the crowdsourcing network as a single team, it is similar to a networked organization both in operation and the choice of rational efforts. We, therefore, would be using the term organization in the rest of this chapter to imply the networked crowdsourcing system.

5.1 Incentive Design for Improving Productive Output ¹

The organization of economic activity as a means for the efficient co-ordination of effort is a cornerstone of economic theory. In networked organizations, agents are responsible for two processes: information flow and productive effort. A major objective of the organization is to maximize the net productive output of the networked system. However, in real organizations the individuals are rational and intelligent. They select their degree of effort which maximizes their payoff. Hence, to understand how organizations can boost their productive output, we need to understand how the individuals connected over a network split their efforts between work *vs.* investing effort in explaining tasks to others depending on the amount of direct and indirect rewards. We model the agents having dual responsibilities of executing the task (production effort) and communicating the information (communication effort) to other agents. When an agent communicates with another, we call the former an *influencer* and the latter an *influencee*. Influencers improve the productivity of the influencees. Influencees, in turn, share a part of their rewards with the influencers, and this interaction induces a game between the agents connected over the network. In our model, working on a task is costly and brings a direct payoff, whereas investing effort in explaining a task can improve the productivity of others (depending on the quality of communication in the network). This can in turn generate additional indirect reward for an agent through reward sharing incentives.

We model the network as a directed graph, where the direction represents the direction of information flow or communication between nodes and the rewards are shared in the reverse direction. Of particular interest to us are directed trees, which represent a hierarchy, and are the most prevalent in organizations and firms. In the first part of this section, our analysis is focused on hierarchies, and in the second, we generalize our results to arbitrary directed graphs. Our goal is to understand how the reward sharing scheme affects the choice of efforts of the agents and thereby influences the total output of the network.

The focus of this section is to maximize the productive effort of the organization, and in the process, understand the dynamics of the influence phenomenon and find the equilibrium efforts chosen by the rational and intelligent agents. Within firms, organizational networks are often hierarchical and there is a long history on the role of organizational structure on

¹Part of this work is going to appear in AAMAS [56], the rest is taken from the full version of the paper [55].

economic efficiency going back to Tichy et al. [80] (on social network analysis within organizations). More recently, Radner [67], Ravasz and Barabási [68], Mookherjee [50] study the role of hierarchies; see Van Alstyne [81] for a survey of different perspectives. There is also a growing interest in crowdsourcing, and relevant here, the ability to generate effective networks for solving challenging problems. Our model also captures some aspects of so called ‘diffusion-based task environments’ where agents become aware of tasks through recruitment [65, 84]. For example, the winner of the 2009 DARPA Red Balloon Challenge adopted an indirect reward scheme where the reward associated with successful completion of subtasks was shared with other agents in the network [65]. At the same time modern massive online social networks and online gaming networks¹ require information and incentive propagation to organize activity. In this section, we draw attention to the *interaction* between various aspects of network influence, such as profit sharing [30], information exchange [12], and influence in networks.

Motivated by the possibility that this phenomenon of splitting effort into production and communication can be understood as a consequence of the strategic behavior of the participants, we adopt a game theoretic perspective where individual members in a networked organization decide on effort levels motivated by their self interest. Agents are coordinated by incentives, including both direct wages and indirect profit sharing. We construct quantitative models of organizations, that are general enough to capture social and economic networks, but specific enough for us to obtain insightful results. We quantify the effects of reward sharing and communication quality on the performance of work organizations in equilibrium. We then turn to the question of designing proper reward shares that can motivate people to work hard and maximize the social output of the system. We show that for stylized networks, under certain conditions, a proper incentive design can lead to an optimal social output. But when the condition is not satisfied, we capture the loss in optimality using the Price of Anarchy (PoA) [42] framework. In particular, we provide the worst case bound on the sub-optimality.

Efforts in Networks: An Example

In this section, we illustrate the effort trade-off through an example of the corporate hierarchies. As we model the crowdsourcing system as an organization in this chapter, the operation and the goal of the design of a networked crowdsourcing system that we discuss next can be visualized using this example. A typical (and restricted) organizational hierarchy is shown in Figure 5.1. A usual observation in hierarchies is that the employees who are in the leaf-level of the tree spend more time in production and as people are promoted, e.g., to manager or senior manager etc. they spend a significant amount of time in communicating with the people they manage.

¹<http://www.eveonline.com/>

In this chapter, we are interested in the question: ‘How much time should a manager spend in *working* on a problem versus *managing* his employees?’

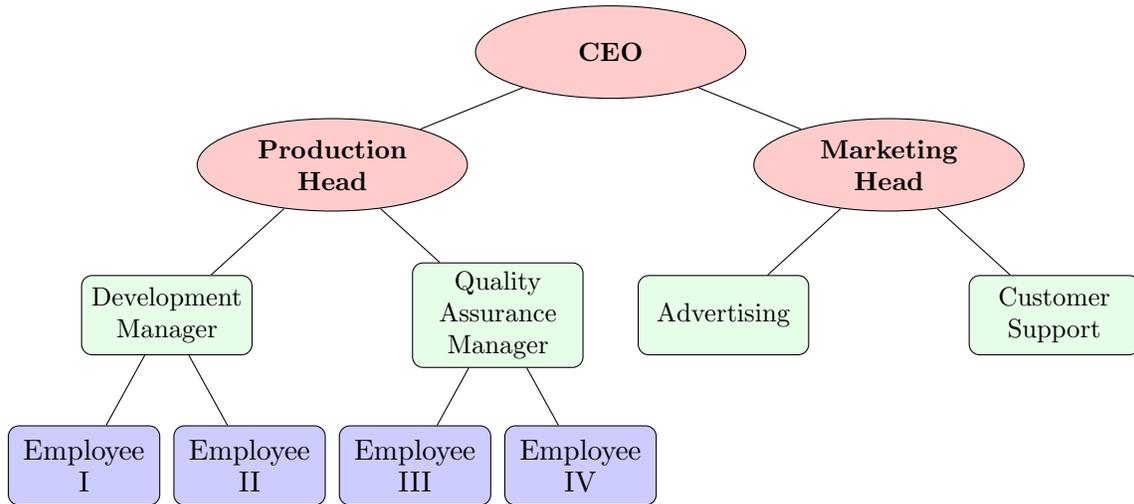


Figure 5.1: A typical corporate hierarchy.

5.1.1 Prior Work

The study of effort levels in network games, where an agent’s utility depends on actions of neighboring agents has recently received much attention [28]. For example, Ballester et al. [7] show how the level of activity of a given agent depends on the Bonacich centrality of the agent in the network, for a specific utility structure that results in a concave game. Our model differs in two aspects: (a) we have multiple types of efforts (namely production and communication) which has different nonlinear correlation among the agents, and (b) utilities are non-concave. In addition, our results give a design recipe for the reward sharing scheme. Rogers [70] analyzes the efficiency of equilibria in two specific types of games (i) ‘giving’ and (ii) ‘taking’, where an edge means utility is sent on an edge. A strategic model of effort is discussed in the public goods model of Bramoullé and Kranton [14], where utility is concave in individual agents’ efforts, and the structures of the Nash and stable equilibria are shown. Their model applies to a very specific utility structure where the same benefit of the ‘public good’ is experienced by all the first level neighbors on a graph. In our model, the individual utilities can be asymmetric, and depend on the efforts and reward shares in multiple levels on the graph. Building on these efforts our utility model cleanly separate the effects of two types of influence, that we termed information and incentives.

The recent DARPA Red Balloon Challenge, and particularly the hierarchical network and specific reward structure used by the winning MIT team [65], has led to a renewed interest

in the analysis of effort exerted by agents in networks. The winning team’s strategy, utilized a recursive incentive mechanism. Our results show that, in this case for example, too much reward sharing encourages managers to spend more time recruiting or managing and not enough time searching or working, though we do not study network formation games here.

The literature on strategic social network formation games and organizational design is vast [38, 35]. We use the Price of Anarchy (PoA) introduced by Koutsoupias and Papadimitriou [42] to measure the sub-optimality in outcome efforts, as a function of network structure and incentives, due to the self interested nature of agents. In the network contribution games literature, the PoA has been investigated in different contexts. Anshelevich and Hoefer [3] consider a model where an agent’s contribution locally benefit the nodes who share an edge with him, and give existence and PoA results for pairwise equilibrium for different contribution functions. The PoA in cooperative network formation is considered by Demaine et al. [23], while Roughgarden [72], Garg and Narahari [29] have considered the question in a selfish network routing context. Our setting is different from all of these since in our model the strategies are the efforts of the agents, which distinguishes it from the network formation and selfish routing literature, and we use multiple levels of information and reward sharing and study utilities that are asymmetric even for the neighboring nodes in the network, which distinguishes itself from the network contribution games.

5.1.2 Overview and Main Results

For the ease of exposition, in the first and major part of this work, we study hierarchies where the network is a directed tree. Each agent decides how to split its effort between (i) production effort, which results in direct payoff for the agent and indirect reward to other agents on the path from the root to the agent, and (ii) communication effort, which serves to improve the productivity of his descendants on the tree (e.g., explaining the problem to others, conveying insights and the goals of the organization). A natural constraint is imposed on the complementary tasks of production and communication, such that the more effort an agent invests in production the less he can communicate. Investing production effort incurs a cost to an agent, in return for some direct payoff. But committing effort to communication that can improve productivity of descendants, which in turn improves their output, should they decide to invest effort in direct work, and thus give an agent a return on investment through an indirect payoff.

Each agent decides, *based on his position in the hierarchy*, how to split his effort between production and communication, in order to maximize the sum of direct payoff and indirect reward, accounting for the cost of effort. For most of our results we adopt an *exponential*

productivity (EP) model, where the quality of communication falls exponentially with effort spent in production with a parameter β . The model has the useful property that a pure-strategy Nash equilibrium always exists (Theorem 5.1) even though the game is non-concave. In a concave game, the agents’ payoffs are concave in their choices (production efforts), and a pure-strategy Nash equilibrium is guaranteed to exist [71]. We develop tight conditions for the uniqueness of the equilibrium (Theorem 5.2). In addition, for the EP model of communication, the Nash equilibrium can be computed in time that is quadratic in the number of agents, despite the non-concave nature of the problem, by exploiting the hierarchical structure.

We then ask the question what effect this equilibrium effort level has on the total output of the hierarchical organization. We define the *social output* to be the sum of the individual outputs which are products of *productivity* and *production effort*. Our next result is that for *balanced hierarchies* and in the EP model, there exists a threshold β^* on communication quality parameter β such that if communication performance is below the threshold (communication is ‘good enough’) then the *equilibrium social output* can be made equal to the *optimal social output* by choosing *an optimal reward sharing scheme*. The phenomenon is captured by the fraction called PoA, which is the ratio of an optimal and the equilibrium social output. If the reward share is not chosen appropriately, PoA can be large (Theorem 5.4). For β above this threshold (low quality communication), we give closed-form bounds on the PoA (Theorem 5.5), which we show are tight in special networks, e.g., single-level hierarchies. This highlights the importance of the design of reward sharing in organizations accounting for both network structure and communication process in order to achieve a higher network output.

In the second part, we consider general directed network graphs and establish the existence of a pure-strategy Nash equilibrium and a characterization for when this equilibrium is unique (Theorems 5.6 and 5.7). We also provide a geometric interpretation of these conditions in terms of the stability properties of a suitably defined Jacobian matrix (Figure 5.7). This connection between control-theoretic stability and uniqueness of Nash equilibrium in network games is an interesting property of the model.

For ease of reading, some proofs are deferred to the Appendix.

5.1.3 A Hierarchical Model of Influencer and Influencee

In this section, we formalize a specific version of the hierarchical network model. Let $N = \{1, 2, \dots, n\}$ denote a set of agents who are connected over a hierarchy T . Each node i has a set of *influencers*, whose communication efforts influence his own direct payoff, and a set of *influencees*, whose direct payoffs are influenced by node i . In turn the production efforts of these influencees endow agent i with indirect payoffs. The origin (denoted by node θ) is a node

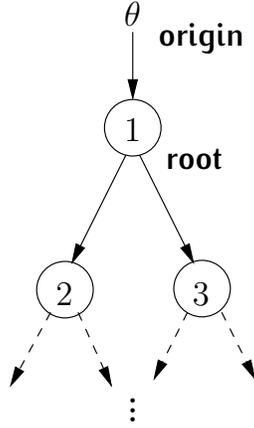


Figure 5.2: A typical hierarchical model.

assumed to be outside the network, and communicates perfectly with the first (root) node, denoted by 1.

We number nodes lexicographically, so that each child has a higher index than his parent, thus the adjacency matrix is an upper triangular matrix with zeros on the diagonal. Figure 5.2 illustrates the model for an example hierarchical network.

The set of influencers of node i consists of the nodes (excluding node i) on the unique path from the origin to the node, and is denoted by $P_{\theta \rightarrow i}$. The set of influencees of node i consists of the nodes (again, excluding node i) in the subtree T_i below her.

The production effort, denoted by $x_i \in [0, 1]$, of node i yields a direct payoff to the node, and the particular way in which this occurs depends on its *productivity*. The remaining effort, $1 - x_i$, goes to communication effort, and improves the productivity of the influencees of the node. The constant sum of production effort and communication effort models the constraint on an agent's time, and it is enough to write both the direct and indirect payoff of a node as a function of the production effort x_i . In particular, the productivity of a node, denoted by $p_i(\mathbf{x}_{P_{\theta \rightarrow i}})$, depends on the communication effort (and thus the production effort) of the influencers on path $P_{\theta \rightarrow i}$ to the node. The production effort profile of these influences is denoted by $\mathbf{x}_{P_{\theta \rightarrow i}}$.

It is useful to associate $x_i p_i(\mathbf{x}_{P_{\theta \rightarrow i}})$ with the value from the *direct output* of node i . The payoff to node i comprises two additive terms that capture:

(1) the direct payoff, which depends on the value generated by the direct output of a node and the cost of production and communication effort, and is modulated by the productivity of the node, and

(2) the indirect payoff, which is a fraction of the value associated with the direct output of any influencee j of the node.

Taken together, the payoff to a single node i is:

$$u_i(x_i, x_{-i}) = p_i(\mathbf{x}_{P_{\theta \rightarrow i}})f(x_i) + \sum_{j \in T_i \setminus \{i\}} h_{ij} p_j(\mathbf{x}_{P_{\theta \rightarrow j}}) x_j. \quad (5.1.1)$$

The first term is the product of the direct payoff and a function $f(x_i)$ (which models production output and cost) and captures the trade-off between direct output and cost of production and communication effort. The second term is the total indirect payoff received by node i due to the output $p_j(\mathbf{x}_{P_{\theta \rightarrow j}})x_j$ of its influences. We insist that the productivity $p_j(\cdot)$ of any node j is non-decreasing in the communication effort of each influencer, and thus non-increasing in the production effort of each influencer, and we require $\frac{\partial}{\partial x_i} p_j(\mathbf{x}_{P_{\theta \rightarrow j}}) \leq 0$ for all nodes j , where i is an influencer of j .

Each node i receives a share h_{ij} of the value of the direct output of influencee j . The model can also capture a setting where an agent can only share output he creates, i.e., the total fraction of the output an agent retains and shares with the influencers is bounded at 1. Let us assume that agent j retains a share s_{jj} and shares s_{ij} with influencers $i \in P_{\theta \rightarrow j}$. A budget-balance constraint on the amount of direct value that can be shared requires $\sum_{i \in P_{\theta \rightarrow j} \cup \{j\}} s_{ij} \leq 1$. Assume that $s_{jj} = \gamma > 0$, for all j , so that each node retains the same fraction γ of its direct output value. Then, the earlier inequality can be written as, $\sum_{i \in P_{\theta \rightarrow j}} \frac{s_{ij}}{\gamma} \leq \frac{1}{\gamma} - 1$. By now defining $h_{ij} = \frac{s_{ij}}{\gamma}$, the whole system is scaled by a factor γ . In addition to notational cleanliness, this transformation gives the advantage of not having any upper bound on the $\sum_{i \in P_{\theta \rightarrow j}} h_{ij}$, since any finite sum can always be accommodated with a proper choice of γ . Let us call the matrix $H = [h_{ij}]$ containing all the reward shares as the *reward sharing scheme*.

To highlight our results, we focus on a specific form of the payoff model, namely the *Exponential Productivity* (EP) model. A model is an instantiation of the direct-payoff function $f(x_i)$ and the productivity function $p_i(\cdot)$. In particular, in the EP model:

$$f(x_i) = x_i - \frac{x_i^2}{2} - b \frac{(1 - x_i)^2}{2}, \quad (5.1.2)$$

$$p_i(\mathbf{x}_{P_{\theta \rightarrow i}}) = \prod_{k \in P_{\theta \rightarrow i}} \mu(C_k) e^{-\beta x_k}, \quad (5.1.3)$$

where $b \geq 0$ is the cost of communication, C_k is the number of children of node k , function $\mu(C_k) \in [0, 1]$ required to be non-increasing, and $\beta \geq 0$ denotes the quality of communication, with higher β corresponding to a lower quality of communication. We assume $p_1 = 1$ for the root node. This models the root having perfect productivity. We interpret the term $\mu(C_k) e^{-\beta x_k} \in [0, 1]$ as the communication influence of node k on the agents in his subtree.

The direct payoff of an agent i is quadratic in production effort x_i , and reflects a linear benefit x_i from direct production effort but a quadratic cost $x_i^2/2$ for effort. The utility model given by Equation (5.1.1) resembles the utility model given by Ballester et al. [7]. However, there are a few subtle differences in our model than that in this paper: (a) the utility of agent i is not concave in her production effort x_i (caused by the exponential term in the productivity); thus the existence of Nash equilibrium is nontrivial (for concave games Nash equilibrium is guaranteed to exist [71]); (b) each agent has two types of effort, namely production and communication, and the communication effort of an agent is *complementary* to the production efforts of her influencees, while the production efforts are *substitutable* to each other. Also, the complementarity is nonlinear. In Section 5.1.4, we address quite general nonlinear complementarity. This is a step forward to the multidimensional effort distribution with nonlinear correlation between the efforts among agents. We chose this particular form to capture a realistic organizational hierarchy; (c) In addition, we also consider the cost due to communication, captured by $b(1 - x_i)^2/2$.

The productivity of node j , given by $p_j(\mathbf{x}_{P_{\theta \rightarrow j}})$, where $j \in T_i \setminus \{i\}$ warrants a careful observation. Here we explain the components of this function and the reasons for choosing them. Consider $\mu(C_k)$, which is non-increasing in the number of children. C_k captures the idea that the effect of the communication effort is reduced if the node has more children to communicate with. An increase in production effort x_k reduces the productivity of influencees of node k . In particular, the exponential term in the productivity captures two effects: (a) a linear decrease in production effort gives exponential gain in the productivity of influencee, which captures the importance of communication and management in organizations [2]. Smaller values of β model better communication and a stronger positive effect on an influencee. (b) We can approximate other models by choosing β appropriately. Linear productivity corresponds to small values of β . This property is useful when the effects of production and communication on the payoff are equally important. For large β , communication quality between agents is poor and the value of communication effort is low.

The successive product of these exponential terms in the path from root to a node reflects the fact that a change in the production effort of an agent affects the productivity of the entire subtree below her. We note that the productivity of node j , where $j \in T_i \setminus \{i\}$, is not a concave function of x_i , leading to the payoff function u_i to be non-concave in x_i . Hence the existence of a Nash equilibrium is not guaranteed a priori through known results on concave games [71]. In the next section we will demonstrate the required conditions on existence and uniqueness of a Nash equilibrium. For brevity of notation, we will drop the arguments of productivity p_i at certain places where it is understood.

Our results on existence, uniqueness and their interpretations generalize to other network structures beyond hierarchies, which we show in the later part of this section. However, despite the mathematical simplicity of the EP model, it allows us to obtain interesting results on the importance of influence, both communication and incentives, and gives insight on outcome efforts in a networked organization.

Main Results

The effect of communication efforts between nodes i and j , where $i \in P_{\theta \rightarrow j}$ is captured by the fractional productivity $\frac{p_j}{p_i}$ defined as, $p_{ij}(\mathbf{x}_{P_{i_- \rightarrow j}}) = \prod_{k \in P_{i_- \rightarrow j}} \mu(C_k) e^{-\beta x_k}$, (the node i_- is the parent of i in the hierarchy). This term is dependent only on the production efforts in the path segment between i and j and accounts for ‘local’ effects. We show in the following theorem that the Nash equilibrium production effort of node i depends on this local information from all its descendants.

Theorem 5.1 (Existence of a Nash Equilibrium) *A Nash equilibrium always exists in the effort game in the EP model, and is given by the production effort profile (x_i^*, x_{-i}^*) that satisfies,*

$$x_i^* = \left[1 - \frac{\beta}{1+b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij}(\mathbf{x}_{P_{i_- \rightarrow j}}^*) x_j^* \right]^+ \quad (5.1.4)$$

Proof: The proof of this theorem uses the hierarchical structure of the network and the fact that the productivity functions (p_i ’s) are bounded. We present the proof in Appendix 5.A. ■

This theorem shows that the EP model allows us to guarantee the existence of (at least one) Nash equilibrium. In particular, we can make certain observations on the equilibrium production effort, some of which are intuitive.

- If communication improves, i.e., β becomes small, the production effort of each node increases.
- If the cost of management b increases, the production effort of each node increases.
- When reward sharing (h_{ij}) is large, agents reduce production effort and focus more on communication effort, which is more productive in terms of payoffs.
- The computation of a Nash equilibrium at any node depends only on the production efforts of the nodes in its subtree. Thus, we can employ a backward induction algorithm which exploits this property that helps in an efficient computation of the equilibrium (this will be shown formally in the corollaries later in this section).

We turn now to establishing conditions for the uniqueness of this Nash equilibrium. Let us define the maximum amount of reward share that any node i can accumulate from a hierarchy T given a reward sharing scheme H as, $h_{\max}(T) = \sup_i \sum_{j \in T_i \setminus \{i\}} h_{ij}$. We also define the *effort update function* as follows.

Definition 5.1 (Effort Update Function (EUF)) *Let the function $F : [0, 1]^n \rightarrow [0, 1]^n$ be defined as,*

$$F_i(\mathbf{x}) = \left[1 - \frac{\beta}{1+b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij}(\mathbf{x}_{P_{i \rightarrow j}}) x_j \right]^+.$$

Note that the RHS of the above expression contains the production efforts of all the agents in the subtree of agent i . This function is a prescription of the choice of the production effort of agent i , if the agents below the hierarchy choose a certain effort profile. Hence the name ‘effort update’.

Theorem 5.2 (Sufficiency for Uniqueness) *If $\beta < \sqrt{\frac{1+b}{h_{\max}(T)}}$, the Nash equilibrium effort profile (x_i^*, x_{-i}^*) is unique and is given by Equation (5.1.4).*

Proof: The proof of this theorem shows that F is a contraction, and is given in Appendix 5.A. ■

Theorem 5.3 (Tightness) *The sufficient condition of Theorem 5.2 is tight.*

Proof: Consider a 3 node hierarchy with nodes 2 and 3 being the children of node 1 (Figure 5.3). We show that if the sufficient condition is violated, we get multiple equilibria. Let $b = 0$, and $h_{12} = h_{13} = 0.25$, therefore $h_{\max}(T) = 0.25$. Theorem 5.2 requires that $\beta < 1/\sqrt{0.25} = 2$. We choose $\beta = 2$. The equilibrium efforts for node 2 and 3 are 1. Node 1 solves the following equation to find the equilibria.

$$1 - x_1 = e^{-2x_1}.$$

This equation has multiple solutions, $x_1 = 0, 0.797$, showing non-uniqueness. ■

The uniqueness condition indicates that the communication quality needs to be ‘good enough’ (small β) to ensure uniqueness of an equilibrium. It is worth noting that the uniqueness condition ensures the convergence of the best response dynamics, in which all the players start from any arbitrary effort profile \mathbf{x}_{init} , and sequentially update their efforts via the function F , to the unique equilibrium. This is a consequence of the fact that F is a contraction.

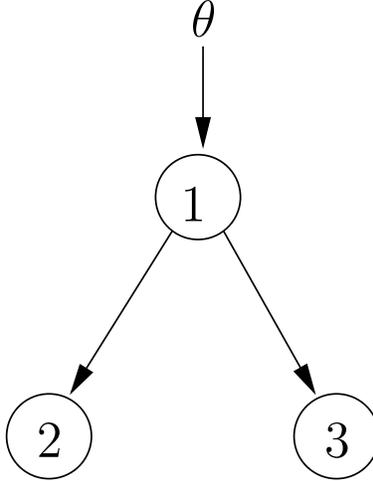


Figure 5.3: Tightness of the sufficiency (Theorem 5.2).

We now turn to the computational complexity of a Nash equilibrium. If there are multiple NE, these complexity results hold for computing *a* NE. Recall that the equilibrium computation of an agent requires only the production efforts and the reward structure of its subtree, and we can take advantage of the backward induction. This observation leads to the following corollaries.

Corollary 5.1 *The worst-case complexity of computing the equilibrium effort for node i is $O(|T_i|^2)$. As a result, The worst-case complexity of computing the equilibrium efforts of the whole network is $O(n^2)$.*

Proof: To compute the equilibrium production effort x_i^* , node i needs to compute Equation (5.1.4). This requires to compute the equilibrium efforts for each node in his subtree T_i . Because of the fact that x_i^* depends only on the equilibrium efforts of the subtree below i , we can apply the backward induction method starting from the leaves towards the root of this sub-hierarchy T_i . The worst-case complexity of such a backward induction occurs when the sub-hierarchy is a line. In such a case the complexity would be $|T_i|(|T_i| - 1)/2 = O(|T_i|^2)$. In order to compute the equilibrium efforts of the whole network, it is enough to determine the equilibrium effort at the root because this would, in the process, determine the equilibrium efforts of each node in the hierarchy. This is also a consequence of the backward induction method of computing the equilibrium. The worst-case complexity of finding the equilibrium effort at the root is $O(n^2)$ and therefore the worst-case complexity of computing the equilibrium efforts of the whole network is also $O(n^2)$. ■

Given the characterization of the Nash equilibrium above, we now move on to questions of characterizing the amount of direct output value generated in equilibrium.

Maximizing the Productive Output of the Network

In our model, the equilibrium behavior of the agents are tightly coupled with the network structure and the reward sharing scheme as seen from Equation (5.1.4). In this section, we look at how the equilibrium behavior affects the *social output* of the hierarchy T for a given effort vector $\mathbf{x} \in [0, 1]^n$, defined as follows.

$$SO(\mathbf{x}, T) = \sum_{i \in N} p_i(\mathbf{x}_{P_{\theta \rightarrow i}}) x_i \quad (5.1.5)$$

This quantity captures the sum of the output of each individual agents in the network, where the output of each agent is the product of their productivity and production effort. For a given hierarchy T , let us define an optimal effort vector as $\mathbf{x}^{\text{OPT}} \in \operatorname{argmax}_{\mathbf{x}} SO(\mathbf{x}, T)$. This is the production effort profile across the network that maximizes the total direct output value, considering also the effect of communication effort (induced by lower production effort) on the productivity of other nodes. Ideally the designer would like to achieve this maximal social output for the given hierarchy. However, the strategic choice of the individuals might not always lead to this performance of the system as a whole. The question we address in this section is how the Nash equilibrium effort level \mathbf{x}^* performs in comparison to the socially optimal outcome \mathbf{x}^{OPT} .

We will consider cases where the equilibrium is unique, hence, the *price of anarchy* [42] is given by:

$$\text{PoA} = \frac{SO(\mathbf{x}^{\text{OPT}}, T)}{SO(\mathbf{x}^*, T)}. \quad (5.1.6)$$

This quantity measures the degree of efficiency of the network. Making PoA equal to unity would be the ideal achievement for the designer. However, that may not always be possible given the parameters of the model. In such a case, we provide a design procedure of the reward sharing scheme that yields the maximum social output.

We note that the equilibrium effort profile \mathbf{x}^* depends on the reward sharing scheme H , while \mathbf{x}^{OPT} does not. The goal of this section is to understand how one can engineer the H to reduce the PoA (thereby making the social output closer to an optimal). The following theorem shows that if the reward sharing is not properly designed, the PoA can be arbitrarily large. We first consider a single-level hierarchy (see Figure 5.4). To simplify the analysis, we also assume

that the function $\mu(C_1) = 1$, irrespective of the number of children of node 1. By symmetry, we consider a single value h , such that $h_{12} = h_{13} = \dots = h_{1n} = h$. We refer to this model as FLAT. We will return to this model later as well, after presenting our results for more general balanced hierarchies. We first consider what happens when there is bad communication (β large) and no profit sharing ($h = 0$), between node 1 and its children.

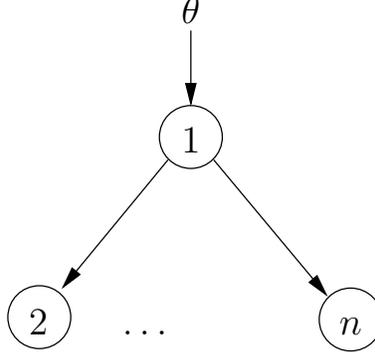


Figure 5.4: FLAT hierarchy.

Theorem 5.4 (Large PoA) For $n \geq 3$, the PoA is $\frac{n-1}{2}$ in the FLAT hierarchy when $\beta = \ln(n-1)$ and $h = 0$.

Proof: For FLAT, the social output is given by, $SO(\mathbf{x}, \text{FLAT}) = \sum_{i=2}^n e^{-\beta x_i} x_i + x_1$. We see that $\beta = \ln(n-1) \geq -\ln(1 - \frac{1}{n-1})$, for all $n \geq 3$. The optimal effort profile $\mathbf{x}^{\text{OPT}} = (0, 1, \dots, 1)$ maximizes the social output (stated in Corollary 5.2, for the proof see Lemma 5.7 in Appendix 5.B). Hence an optimal social output is $n-1$. However, for reward sharing factor $h = 0$, we get the equilibrium effort profile from Equation (5.1.4) to be $\mathbf{x}^* = (1, 1, \dots, 1)$. This yields a social output of $(n-1)e^{-\beta} + 1$. Hence the PoA is $\frac{n-1}{(n-1)e^{-\ln(n-1)} + 1} = \frac{n-1}{2}$. ■

However, if h is chosen appropriately, e.g., if it were chosen to be large positive, the equilibrium effort profile given by Equation (5.1.4) would have been closer to that of an optimal. Hence PoA could have been reduced and made closer to 1.

This raises a natural question: *is it always possible to design a suitable reward sharing scheme that can make PoA = 1 for any given hierarchy?* In order to answer that, we define the *stability* of an effort profile \mathbf{x} .

Definition 5.2 (Stable Effort Vector) An effort profile $\mathbf{x} = (x_1, \dots, x_n)$ is stable, represented by $\mathbf{x} \in S$, if $\mathbf{x} \geq \mathbf{0}$, and there exists a reward sharing matrix $H = [h_{ij}]$, $h_{ij} \geq 0$, such that,

$$\sum_{j \in T_i \setminus \{i\}} a_{ij}(\mathbf{x}) h_{ij} \geq 1 - x_i; \quad \sum_{j \in T_i \setminus \{i\}} h_{ij} \leq \frac{1+b}{\beta^2}, \quad \forall i \in N. \quad (5.1.7)$$

Where, $a_{ij}(\mathbf{x}) = \frac{\beta}{1+b} p_{ij}(\mathbf{x}_{P_{i \rightarrow j}}) x_j$, for all $j \in T_i \setminus \{i\}$, and zero otherwise. If such a solution H exists, we call it a stable reward sharing matrix.

The inequalities capture a required balance between incentives and information flow. In the first inequality, for a fixed communication factor β and cost coefficient b , the term $a_{ij}(\cdot)$ is proportional to the fractional output (fractional productivity \times production effort) of an agent j . After multiplying with h_{ij} , this is the effective indirect output that i receives from j . The RHS of the inequality can be interpreted as the communication effort of agent i . Hence, this inequality says that the total indirect benefit should be at least equal to the effort put in by a node for communicating the information to its subtree. If we consider that the agents share information based on the reward share they receive, the flow of information and reward forms a closed loop. The second inequality says that the closed loop ‘gain’ of the information flow (β^2) and the reward share accumulated by agent i ($\sum_{j \in T_i \setminus \{i\}} h_{ij}$) should be bounded by the cost of sharing the information. The closed loop ‘gain’ is essentially the reward that an agent accumulates due to his communication effort *through* his descendants. We can connect a stable effort vector with the Nash equilibrium of the effort game.

Lemma 5.1 (Stability-Nash Relationship) *If an effort profile $\mathbf{x} = (x_1, \dots, x_n)$ is stable, it is the unique Nash equilibrium of the effort game with the corresponding stable reward sharing matrix.*

Proof: Let \mathbf{x} is a stable effort profile. So, there exists a stable reward sharing matrix corresponding to it. Let $H = [h_{ij}]$, $h_{ij} \geq 0$ be the matrix, s.t. Equation (5.1.7) is satisfied with \mathbf{x} . Also $\mathbf{x} \geq \mathbf{0}$. Therefore, reorganizing the first inequality of Equation (5.1.7) and noting the fact that $x_i \geq 0$, $\forall i \in N$, we get,

$$x_i = \left[1 - \sum_{j \in T_i \setminus \{i\}} a_{ij}(\mathbf{x}) h_{ij} \right]^+, \quad \forall i \in N.$$

Under the condition given by the second inequality of Equation (5.1.7), the Nash equilibrium is unique and is given by the above expression (recall Theorem 5.2). Hence, \mathbf{x} is the unique Nash equilibrium of this game. ■

Now it is straightforward to see that the stability of \mathbf{x}^{OPT} is sufficient for PoA to be 1. This is because now the H that makes the \mathbf{x}^{OPT} vector stable can be used as the reward sharing scheme, and for that H the equilibrium effort profile will coincide with \mathbf{x}^{OPT} . In other words, an optimal effort vector can be supported in equilibrium by a suitable reward sharing scheme. Hence, the following lemma is immediate.

Lemma 5.2 (No Anarchy) *A stable reward sharing scheme corresponding to \mathbf{x}^{OPT} yields a PoA of 1.*

A couple of important questions are then: *how efficiently can we check if a given effort profile \mathbf{x} is stable or not? And how to choose a reward sharing scheme that makes the effort profile stable?* The answer is that we can solve the following feasibility linear program (LP) for a given effort profile:

$$\begin{array}{ll} \min & 1 \\ \text{s.t.} & \left. \begin{array}{l} \sum_{j \in T_i \setminus \{i\}} a_{ij}(\mathbf{x}) h_{ij} \geq 1 - x_i, \\ \sum_{j \in T_i \setminus \{i\}} h_{ij} \leq \frac{1+b}{\beta^2}, \\ h_{ij} \geq 0, \forall j, \end{array} \right\} \forall i \in N. \end{array} \quad (5.1.8)$$

If a solution exists to the above LP, we conclude that \mathbf{x} is stable and declare the corresponding H to be the resulting reward sharing scheme. Linear programs can be efficiently solved and therefore checking an effort profile for stability can be done efficiently.

A Note on the Reward Share Design This condition gives us a recipe of the design of the reward sharing scheme. However, the next question is: *what happens when the \mathbf{x}^{OPT} is unstable?* If the above feasibility LP does not return any solution matrix H , we conclude that $\mathbf{x}^{OPT} \notin S$. In such a scenario, we cannot guarantee PoA to be unity. However, for any given reward sharing matrix H , there is an equilibrium effort profile $\mathbf{x}^*(H)$. We can, therefore, solve for $H_{\max} \in \arg\max_{H: \mathbf{x}^*(H) \in S} SO(\mathbf{x}^*(H))$ which leads to an equilibrium effort profile $\mathbf{x}^*(H_{\max})$ that lies in the stable set and maximize the social output. Therefore, when we cannot find a reward sharing scheme to achieve an optimal social output, H_{\max} is our best bet. Computing H_{\max} for general hierarchies may be a hard problem, and we leave that as an interesting future work. However, for certain special classes of hierarchies, it is possible to derive bounds on the PoA (thereby providing a design recipe for H to achieve a lower bound on the social output). In the following section, we do this for the balanced hierarchies. The price of anarchy analysis, therefore, serves as a means to find an optimal reward sharing scheme that gives a theoretical guarantee on the social output of the system.

Price of Anarchy in Balanced Hierarchies

In this section we consider a simple yet representative class of hierarchies, namely the balanced hierarchies, and analyze the effect of communication on PoA and provide efficient bounds. Hierarchies in organizations are often (nearly) balanced, and the FLAT or linear networks are special cases of the balanced hierarchy (depth = 1 or degree = 1). Hence, the class of balanced hierarchies can generate useful insights. In addition, the symmetry in balanced hierarchies

allows us to obtain interpretable closed-form bounds and understand the relative importance of different parameters.

We consider a balanced d -ary tree of depth D . By symmetry, the efforts of the nodes that are at the same level of the hierarchy are same at both equilibrium and optimality. This happens because of the fact that in the EP model, both the equilibrium and optimal effort profile computation follows a backward induction method starting from the leaves towards the root. Since the nodes in the same level of the hierarchy is symmetric in the backward induction steps, they have identical effort profiles.

With a little abuse of notation, we denote the efforts of each node at level i by x_i . We start numbering the levels from root, hence, there are $D + 1$ levels. Note that there are a few interesting special cases of this model, namely (a) $d = 2$: balanced binary tree, (b) $D = 1$: flat hierarchy, (c) $d = 1$: line. We assume, for notational simplicity only, that the function $\mu(C_k) = 1$, for all C_k , though our results generalize. This function is the coefficient of the productivity function. $\mu(C_k) = 1$ also models organizations where each manager is assigned a small team and there is no attenuation in productivity due to the number of children. In order to present the *price of anarchy* (PoA) results, we define the set ξ :

$$\xi(\beta) = \left\{ x : x = \left[1 - \frac{1}{\beta} e^{-\beta x} \right]^+ \right\}. \quad (5.1.9)$$

This set is the set of possible equilibrium effort levels for agents at the penultimate level of the EP model hierarchy when $\beta > 1$. Note that this set is a singleton, when $\beta > 1$. Depending on β , we define a lower bound $\phi(d, \beta)$ on the contribution of an agent toward the social output, and a sequence of nested functions t_i , where d is the degree of each node.

$$\begin{aligned} \phi(d, \beta) &= \max \left\{ \frac{1}{\beta} (1 + \ln(d\beta)), d\beta + (1 - d\beta)\xi(\beta) \right\}, \\ t_1(d, \beta) &= \phi(d, \beta), t_2(d, \beta) = \phi(d \cdot \phi(d, \beta), \beta), \dots, t_D(d, \beta) = \phi(d \cdot t_{D-1}(d, \beta), \beta). \end{aligned} \quad (5.1.10)$$

Theorem 5.5 (Price of Anarchy) *For a balanced d -ary hierarchy with depth D , as β increases, we can show the following price of anarchy results.*

$$\begin{aligned} \text{When } 0 \leq \beta \leq 1, \quad \text{PoA} &= 1, \\ \text{and when } 1 < \beta < \infty, \quad \text{PoA} &\leq \frac{d^D}{t_D(d, \beta)}. \end{aligned} \quad (5.1.11)$$

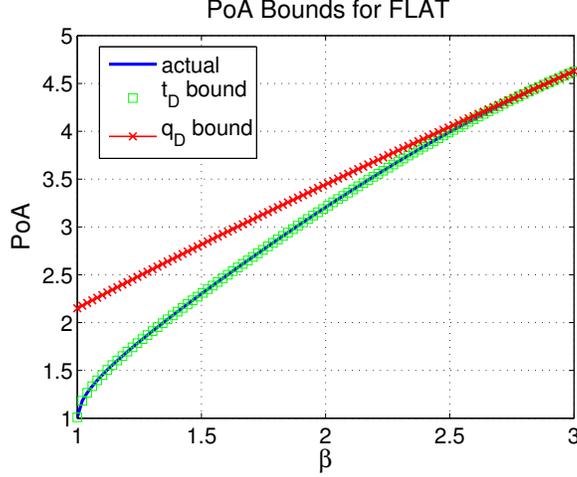


Figure 5.5: Bounds on PoA for FLAT, $d = 6, D = 1$.

Proof: The proof is constructive and sets the H matrix appropriately to achieve the bounds on PoA. The H matrix constructed this way acts as the reward sharing scheme to achieve a reasonable enough social output. For details, see Appendix 5.B. ■

As opposed to our choice of lower bound ϕ , a naïve lower bound of $\frac{1}{\beta}(1 + \ln(d\beta))$ can also be used. We denote the corresponding sequence of nested functions similar to the ones defined in Equation (5.1.10) in that case to be $q_i, i = 1, \dots, D$. However, this gives a weaker bound for any hierarchy. As an example, we compare the bounds with the actual PoA for FLAT (recall Figure 5.4) in Figure 5.5 (the FLAT hierarchy is a balanced tree with $D = 1, d = n - 1$). Figure 5.5 shows that the bound given by our analysis is tight for FLAT, indicating the value of the analysis and also gives intuition to the shape of the effect of β on the PoA. We have the following corollaries of Theorem 5.5,

Corollary 5.2 (Optimal Effort) *For the FLAT hierarchy, if $0 \leq \beta < -\ln(1 - \frac{1}{n})$, an optimal effort profile is where all nodes put unit effort. When $-\ln(1 - \frac{1}{n}) \leq \beta < \infty$, an optimal changes to the profile where the root node puts zero effort and each other node puts unit effort.*

Corollary 5.3 *For the FLAT hierarchy, when $0 \leq \beta \leq 1$, $PoA = 1$, and when $1 < \beta < \infty$, $PoA \leq \frac{n}{\phi(d, \beta)}$.*

The second corollary above makes rigorous the intuition that when β is small enough an optimal \mathbf{x} can be achieved by choosing a small enough reward share h . However, when β grows, in order to ensure uniqueness of the Nash equilibrium, the choice of h becomes limited (as it has to satisfy $\leq (1 + b)/\beta^2$) resulting in a PoA, as captured in Figure 5.5.

5.1.4 A General Network Model of Influencer and Influencee

In this section, we show that the results on existence and uniqueness of a pure strategy Nash equilibrium generalize to a much broader setting of agents as influencer and influencees interacting over an arbitrary network.

Suppose that the agents are connected over a (possibly non-hierarchical) network G . Each node i has a set of influencers, denoted by R_i (generalizing $P_{\theta \rightarrow i}$), and a set of influencees, E_i (generalizing $T_i \setminus \{i\}$). We import the notation from Section 5.1.3 with their exact or analogous meanings for productivity $p_i(\mathbf{x}_{R_i})$ and reward sharing scheme H . Now, the payoff function of agent i is given by,

$$u_i(x_i, x_{-i}) = p_i(\mathbf{x}_{R_i})f(x_i) + \sum_{j \in E_i} h_{ij}p_j(\mathbf{x}_{R_j})x_j. \quad (5.1.12)$$

We assume that f is a strictly concave function, and is continuously differentiable. We will refer to the product of effort x_i and productivity $p_i(\mathbf{x}_{R_i})$ as the output, and denote it by y_i . In this context, we do not impose any condition on the nature of the productivity function $p_i(\cdot)$, and as before, this game is also not necessarily a concave game and the existence of a Nash equilibrium is not always guaranteed.

Results

The payoff function given by Equation (5.1.12) induces a game between the influencers and the influencees. In addition, as before, every agent faces a trade-off when deciding how much production and communication effort to exert. We will need the following facts.

Fact 5.1 *If a function is continuously differentiable and strictly concave, its derivative is continuous and monotone decreasing.*

Fact 5.2 *A continuous and monotone decreasing function is invertible and the inverse is also continuous and monotone decreasing.*

Using the above two facts, we see that the inverse of f' exists and is monotone decreasing. Let us denote f'^{-1} by ℓ . Let us define two functions g and T similar to that defined in Section 5.1.3.

$$g_i(\mathbf{x}) = \sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i(\mathbf{x}_{R_i})} \frac{\partial p_j(\mathbf{x}_{R_j})}{\partial x_i} x_j \right) \Big|_{\mathbf{x}}, \quad (5.1.13)$$

$$T(x) = \min\{\max\{0, x\}, 1\}. \quad (5.1.14)$$

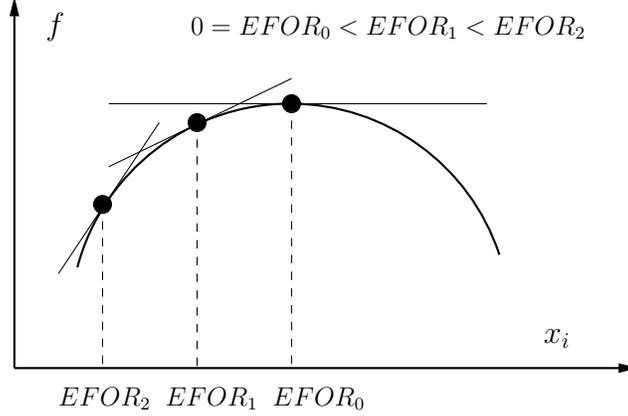


Figure 5.6: Impact on the production effort of a node having higher EFOR from the influencees.

Fact 5.3 *The function T is continuous.*

Lemma 5.3 (Necessary Condition for Nash equilibrium) *If a Nash equilibrium exists for the effort game in a influencer-influencee network, the effort profile (x_i^*, x_{-i}^*) must satisfy,*

$$x_i^* = T \circ \ell \circ g_i(\mathbf{x}^*), \quad \forall i \in N. \quad (5.1.15)$$

To illustrate what this necessary condition means, let us assume, for simplicity, that we do not hit the edges of the truncation function T . Therefore we can rewrite Equation (5.1.15) as,

$$f'(x_i^*) = \sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i} \frac{\partial p_j}{\partial x_i} x_j \right) \Big|_{\mathbf{x}^*} = \sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i} \frac{\partial y_j}{\partial x_i} \right) \Big|_{\mathbf{x}^*} \quad (5.1.16)$$

Where $y_j = p_j x_j$ is the output of node j . We have dropped the arguments of p_i and p_j for brevity of notation. The expression on the LHS is the rate of change of direct benefit for agent i . The RHS is the rate at which the passive output of agent i changes w.r.t. his effort x_i and productivity p_i . If the LHS is larger, the agent would gain more at the margin by increasing x_i . This is because the derivative $(-\partial y_j / \partial x_i)$ is non-negative since $\partial p_j / \partial x_i$ is always non-positive. Similarly, if the RHS was larger, the agent could gain at the margin by decreasing x_i . Hence Equation (5.1.16) resembles a rate balance equation (or demand-supply curve) where the rate of effective direct payoff matches the rate of passive payoffs.

Let us define the *effective fractional output rate* (EFOR) at \mathbf{x} as $\sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i} \frac{\partial y_j}{\partial x_i} \right) \Big|_{\mathbf{x}}$. In some settings, e.g., if $p_i(\mathbf{x}_{R_i}) = \prod_{k \in R_i} (1 - x_k)$, the fractional output rate $\left(-\frac{1}{p_i} \frac{\partial y_j}{\partial x_i} \right)$ can be independent of the production effort of i , i.e., x_i . In such settings, Equation (5.1.16) shows that if the EFOR of node i increases, the equilibrium for node i will move in a direction that decreases production effort x_i . This happens because the slope of f in equilibrium is always non-

negative and its increase leads to a smaller x_i because of the concavity of f . This phenomenon is graphically shown in Figure 5.6. This shows that the nodes having a higher EFOR, which is a function of the network position of an agent, can leverage more on the production efforts of the influencees.

Following Definition 5.1, we define the *effort update function* (EUF) $F : [0, 1]^n \rightarrow [0, 1]^n$ for the general setting as, $F(\mathbf{x}) = T \circ \ell \circ g(\mathbf{x})$, where $T \circ \ell$ operates on the vector function g element-wise. Therefore, the question of existence of a Nash equilibrium of this effort game is the same as asking the question if the following fixed point equation has a solution: $\mathbf{x} = F(\mathbf{x})$.

In the following, we provide a sufficient condition for existence of the Nash equilibrium, and its uniqueness.

Lemma 5.4 *For $p_i > 0$, for all $i \in N$, and continuously differentiable, $F(\mathbf{x})$ is continuous.*

Proof: Given $p_i > 0$, for all $i \in N$, and is continuously differentiable. Therefore, the function g , defined in Equation (5.1.13), is continuous in \mathbf{x} . Using Facts 5.2 and 5.3, we see that the functions ℓ and T are continuous. Hence, $F \equiv T \circ \ell \circ g$ is continuous in \mathbf{x} . ■

Theorem 5.6 (Sufficient Condition for a Nash Equilibrium) *For $p_i > 0$, for all $i \in N$, and continuously differentiable, the effort game has at least one Nash equilibrium.*

Proof: From Lemma 5.3, we see that the Nash equilibrium of the effort game is same as the fixed point of the equation, $\mathbf{x} = F(\mathbf{x})$. Since F is continuous (Lemma 5.4), Brouwer's fixed point theorem immediately ensures a fixed point of the above equation to exist. Hence, the effort game has at least one Nash equilibrium. ■

Let us use the shorthand $G \equiv \ell \circ g$. The following theorem provides a sufficient condition for the uniqueness of the Nash equilibrium.

Theorem 5.7 (Sufficient Condition for unique Nash Equilibrium) *If $\sup_{\mathbf{x}_0} |\nabla G(\mathbf{x}_0)| < 1$, then the Nash equilibrium effort profile (x_i^*, x_{-i}^*) is unique and is given by Equation (5.1.15).*

Proof: The key here is to show that F is a contraction. We follow the steps of Theorem 5.2 as follows:

$$\|F(\mathbf{x}) - F(\mathbf{y})\| \leq \|G(\mathbf{x}) - G(\mathbf{y})\| \leq |\nabla G(\mathbf{x}_0)| \cdot \|\mathbf{x} - \mathbf{y}\|.$$

This is a contraction as $\sup_{\mathbf{x}_0} |\nabla G(\mathbf{x}_0)| < 1$. ■

Interpretation of the Sufficient Condition of the Uniqueness

The sufficient condition given by Theorem 5.7 is a technical one. We now discuss an interesting geometric interpretation of this condition. By the Taylor expansion of G with first order remainder term, we get,

$$G(\mathbf{x}) - G(\mathbf{y}) = \nabla G(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{y}).$$

Where \mathbf{x}_0 lies on the line joining \mathbf{x} and \mathbf{y} . Using singular value decomposition, we get, $\nabla G(\mathbf{x}_0) = U_0 \Sigma_0 V_0^\top$. Therefore, for each pair of points \mathbf{x} and \mathbf{y} , we can transform the space of efforts with a pure rotation as follows.

$$\begin{aligned} G(\mathbf{x}) - G(\mathbf{y}) &= U_0 \Sigma_0 V_0^\top \cdot (\mathbf{x} - \mathbf{y}), \\ \Rightarrow U_0^\top (G(\mathbf{x}) - G(\mathbf{y})) &= \Sigma_0 \cdot (\bar{\mathbf{x}} - \bar{\mathbf{y}}), \text{ where } \bar{\mathbf{x}} = V_0^\top \mathbf{x}, \bar{\mathbf{y}} = V_0^\top \mathbf{y} \\ \Rightarrow R(\bar{\mathbf{x}}) - R(\bar{\mathbf{y}}) &= \Sigma_0 \cdot (\bar{\mathbf{x}} - \bar{\mathbf{y}}), \text{ where } R \equiv U_0^\top G V_0. \end{aligned}$$

Hence, for any pair of points \mathbf{x} and \mathbf{y} , we can rotate the space so that the effect of the deviation to \mathbf{x} from \mathbf{y} can be captured by a weight on each of the coordinates in the rotated space. Here, the diagonal matrix Σ_0 contains the weights along its diagonal.

Theorem 5.7 says that for any point \mathbf{x}_0 , if the absolute value of all the elements of this diagonal matrix is smaller than unity, the uniqueness of the Nash equilibrium is guaranteed. Let us denote the rotated vector of \mathbf{x}_0 by $\mathbf{z}_0 := V_0^\top \mathbf{x}_0$. The diagonal elements can be written w.r.t. the vectors in the rotated space as,

$$(\Sigma_0)_{ii} = \sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i} \frac{\partial^2 p_j}{\partial z_i^2} z_j \right) \Big|_{\mathbf{z}_0} = \frac{\partial \text{EFOR}(\mathbf{z}_0)}{\partial z_{i,0}}.$$

In other words, the diagonal elements are the rate of change of EFOR at \mathbf{z}_0 . Having the rate of change of EFOR bounded by 1 is a sufficient condition for a unique Nash equilibrium. One can think of the EFOR as the product of two effects: (1) the rate of change in productivity, which increases the payoff of the influencees, (2) the reward share h_{ij} 's. The sufficient condition essentially says that the net effect should not be too large in order to guarantee unique equilibrium.

Figure 5.7 shows a graphical illustration of the phenomenon in polar co-ordinates, where the directions represent that of the vectors. The results say that if for any vector, the singular values of the Jacobian matrix of G at that point lies entirely within the unit ball, then there exists an unique Nash equilibrium. This is similar to the feedback loop gain of a feedback controller, where the closed loop gain being smaller than unity ensures stability. We find this natural

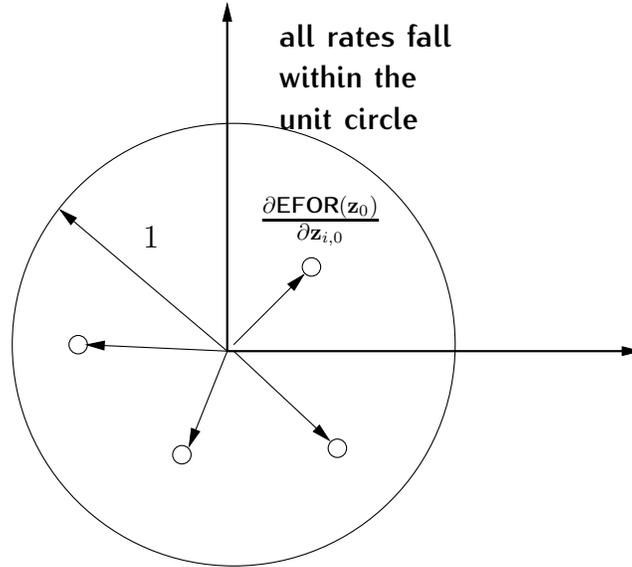


Figure 5.7: Geometric interpretation of the condition for uniqueness of the Nash equilibrium.

parallel between notions of stability (from control theory) and uniqueness of Nash equilibria interesting.

5.2 Network Design for Improving Productive Output ¹

In this section, we focus on the network design aspect of the crowdsourcing system in order to maximize the output. We use a slightly different model here. We assume that the tasks come in the form of a Poisson process and they are consumed by the agents by a similar process. The performance criterion that we focus on is minimization of the *risk* of not meeting the incoming rate of tasks. In this setting, a crowdsourcing network is more productive if it has a lower *risk*, i.e., the probability of the event that the incoming rate of tasks falls short of the sum delivery rate of all the agents in the network. For simplicity, we analyze the hierarchical networks in this section.

We propose a queuing-theoretic model of task arrival and processing, and a natural model of profit sharing that provides a direct reward for completing a task and an indirect reward if subordinates complete a task. We then model employee competition for tasks based on the cost of the effort involved and the possible rewards. In addition to considerations that follow from boundedly rational models of managers [77, 87] with limited information about opportunities, we show how common organizational features, in particular profit sharing and intra-organizational competition, can reduce overall efficiency due to a free-riding phenomenon,

¹This section is based on the paper [61].

which is an interesting feature in this model.

In an environment with profit sharing and without the benefits of information sharing that can accrue within an organization, our first result (Theorem 5.8) quantifies the effect of free riding on reducing the efficiency of an organization. In particular, the total output of an organization decreases when compared to a flat organization. We then introduce a model of information asymmetry and consider the positive effects of information sharing within an organization, where managers are able to improve the estimates subordinates have for the rewards for task completion. That is, managers share a form of ‘business intelligence’ about the value of tasks to the organization and their potential rewards.

Based on this queuing and game theoretic model, we study the trade-offs between free riding and the positive effects of information sharing. In particular we seek to answer quantitative questions of the form ‘how well should management communicate to overcome the free riding effects’? We show that information sharing can provide benefit even in the presence of free riding, and quantify the magnitude of both effects (Theorem 5.9). The simulation results serve to illustrate that the performance of the organization improves when the hierarchy is designed considering free riding, as opposed to constructions that do not model these effects.

Section outline: We first introduce our general model, which includes rewards, information asymmetry and information sharing. In Section 5.2.2, we address the perfect information scenario where agents observe the true reward they obtain from task completion, and describe the free riding that occurs when managers share in the profits of employees. In Section 5.2.3, we study the effect of information asymmetry, and quantify how information sharing and propagation improves the effectiveness of agent effort. In Section 5.2.4, we combine the two, and discuss how the optimization of information propagation in hierarchies with free riding results in interesting network structures with improved efficiency.

5.2.1 The Model

We first develop a model of task (or work) arrival and processing in an organization using a server-queue model. The employees and managers of the organization are modeled as strategic autonomous agents who choose actions to maximize their own payoffs, given the organizational structure and with the possibility of indirect rewards for the tasks completed by others. The agents may have asymmetric information and can communicate when they are connected.

The Task Arrival and Processing Model

We model an organization as a task processing unit where the tasks arrive at a Poisson rate λ , and wait until served. For example, the tasks could be projects that need to be executed, or

even small modules that need to be developed or completed as part of a larger contract. Other situations where such a model would be applicable arise for example in sales organizations for which agents get commissions [25].

Let the set of employees be denoted by $N = \{1, 2, \dots, n\}$, and connected over a hierarchy, represented by a directed tree T . We assume that if an individual performs a task she gets a direct reward and the nodes along the path from the individual to the root receive indirect incentives. These indirect rewards model bonuses or the ‘credit’ a manager receives when a task is completed by her employee.

Each node i attempts to capture a task from the work queue according to a Poisson process with rate λ_i . The rate λ_i corresponds to the effort that the agent expends. The cost of maintaining an attempt rate λ_i is $C\lambda_i$, where $C > 0$ is a known constant. Such a linear increase model of cost has been adopted in literature on public goods network [14]. If the total reward of a task is R , corresponding to the total bonus or credit that the organization allocates to a task, we assume that the node that executes it gets a direct reward of γR ($\gamma \in (0, 1)$), and the rest shared as indirect incentives among its predecessors in the organizational tree. The true reward of a task, R , is stochastic and distributed as F_R .

In our initial analysis, we assume that every agent knows the realized reward of a task and in the subsequent analysis we assume every agent has a noisy estimate of the reward of a task, corresponding to information asymmetry between agents in an organization, where some agents can better estimate the true value of a task and thus expend their effort more effectively. To keep the model simple, every agent is assumed to have the same intrinsic ability to complete a task, all tasks carry the same reward and that the time to complete a task is small compared to the inter-attempt time of an agent. Some of these assumptions can be relaxed with considerable increase in notational complexity though our basic model is sufficient to capture the effects of profit sharing and information asymmetry.

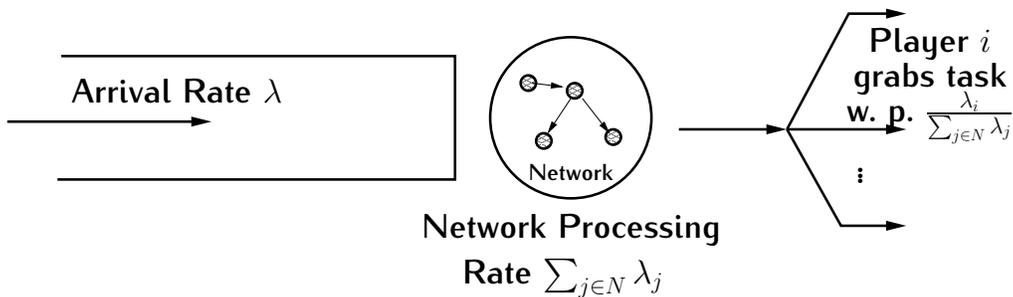


Figure 5.8: Server queue model for the task arrival to the network.

We model the task arrival and departure in a server-queue model, as shown in Figure 5.8, where the arrival rate is $\lambda (> 0)$ and the consolidated service rate of the entire network is

$\sum_{j \in N} \lambda_j$, using the simple fact that the superposition of Poisson processes is Poisson with the rate being the sum of the rates [11]. When agent i tries to capture a job from the task queue, the probability that she succeeds is given by $\lambda_i / \sum_{j \in N} \lambda_j$. This can be shown using the fact that the inter-attempt times are exponentially distributed for a Poisson process, and exponential random variables exhibit memoryless property [89]. This basic model can be extended using other queuing theoretic models of job arrival and task completion, though that is beyond the scope of this section.

Agent Utility and Reward Sharing

Organizations have profit sharing schemes where a manager is rewarded for a successful execution of a project by his team. Hence all agents connected in the hierarchy have two parts to their net payoffs. If node j executes a task, she receives a ‘direct’ reward, and each node i on the directed path from the root to j receives an ‘indirect’ reward. Since efforts are costly ($C\lambda_i$), each agent decides on an effort level to maximize her net payoff, i.e., (direct + indirect) reward - cost. The choice of attempt rate $\lambda_i \in S_i = [0, \infty)$ is a strategy for the agent, and this induces a game between the nodes. If there is a large indirect reward a node can (and will) reduce efforts to reduce costs. In the following, we derive an quantitative expression for the expected utility for an agent i , when there are explicit direct and indirect rewards.

The indirect incentives are shared with other players as specified by a *reward sharing matrix* $\Delta = [\delta_{ij}]$ where δ_{ij} is the fraction of j ’s reward received by i when j executes a task. We assume, $\delta_{ii} = \gamma > 0$, $\forall i \in N$. Moreover, since the network is a directed tree, if j does not appear in subtree of i then $\delta_{ij} = 0$. Our analysis extends to more general organizational structures, such as ‘matrix’ type organizations where employees may report to multiple managers. We call the matrix Δ *monotone non-increasing* if $\delta_{ij} \geq \delta_{ij'}$, whenever the hop-distance $\text{dist}_T(i, j) \leq \text{dist}_T(i, j')$, and *anonymous* if the δ_{ij} depends only on the distance of i and j on T , $\text{dist}_T(i, j)$, and not on the identities of the nodes. To ensure that the total reward is bounded above by R , we assume $\sum_{k \in N} \delta_{kj} \leq 1$, $\forall j \in N$.

An example of a monotone non-increasing and anonymous reward sharing scheme is a *geometric reward sharing* scheme. Node i , a predecessor of j in the tree T , gets a fraction of reward $\delta^{\text{dist}_T(i, j)}$, $\delta \in (0, 1)$, when j grabs the task. Node j gets γ fraction for her own effort, and $\delta + \gamma \leq 1$ is required to balance the budget. For concreteness, we will use this reward sharing scheme for the simulations.

Let us denote with T_i the subtree rooted at i . Combining the direct and indirect rewards and costs, the expected utility of agent $i \in N$ is given by,

$$u_i(\lambda_i, \lambda_{-i}) := \lambda \gamma R \frac{\lambda_i}{\sum_{k \in N} \lambda_k} + \lambda R \sum_{j \in T_i \setminus \{i\}} \delta_{ij} \frac{\lambda_j}{\sum_{k \in N} \lambda_k} - C \lambda_i, \quad (5.2.1)$$

where $\lambda_{-i} = (\lambda_1, \dots, \lambda_{i-1}, \lambda_{i+1}, \dots, \lambda_n)$. In the above equation, the first term on the RHS denotes the expected utility of agent i due to her own effort λ_i , the second term represents the indirect utility coming from the efforts λ_j of all the nodes $j \in T_i \setminus \{i\}$, and the third term captures the cost an agent pays to maintain the effort level. The function u_i can be shown to be concave in λ_i , we skip the proof due to space constraints.

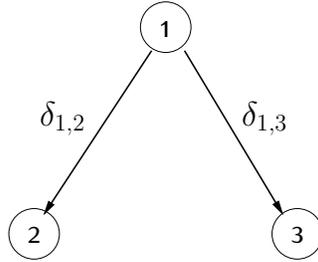


Figure 5.9: Node 1 is rewarded indirectly by 2 and 3.

Depending on the efforts exerted by the agents, the system could operate in one of the following two zones:

Zone 1: When $\sum_{j \in N} \lambda_j > \lambda$, the task queuing process is a positive recurrent Markov chain and all tasks will be served, and hence this is a desirable zone for the organization to operate. Let us denote this zone by $Z_1 = \{\boldsymbol{\lambda} : \sum_{j \in N} \lambda_j > \lambda\}$.

Zone 2: When $\sum_{j \in N} \lambda_j \leq \lambda$, the queuing process is a null or transient Markov chain and with high probability (probability approaching unity) the queue will be non-empty and growing at a steady state [89]. This is an undesirable state for an organization since certain incoming projects will never be served. Let us denote this zone by $Z_2 = \{\boldsymbol{\lambda} : \sum_{j \in N} \lambda_j \leq \lambda\}$.

Simply stated, the desirable state for the organization as a whole is in Zone Z_1 , with all incoming customers served. To illustrate the utilities and the reward sharing among the agents, let us consider a small hierarchy $T^{(3)}$ with nodes 1, 2, and 3, connected as shown in Figure 5.9. Node 2 and 3 gets only direct reward due to their own effort, but 1 gets both direct and indirect rewards.

Information Asymmetry

In the full version of our model, we model agents as having only noisy estimates of the reward associated with a task. Agent i observes the reward as $R_i = R + \eta_i$, which is a noisy version of the actual reward. The noise is independent across agents, and is additive to true reward R , and is distributed according to a Normal distribution: $\eta_i \sim \mathcal{N}(0, \sigma_i^2)$. Agents vary in their intrinsic variance σ_i^2 , reflecting their ability to track the reward accurately. The parameter σ_i is assumed private to agents. For this reason, agent i knows that agent j observes a realization of $R_j = R + \eta_j$, but does not know σ_j or the value of R_j . In this section, we assume that agent i picks its action after observation R_i is realized; i.e., the employees put their efforts after a project arrives.

We allow information to be shared between agents. We model information propagation as the change in the variance of the individuals, corresponding to a change in the uncertainty or business insight that an employee has through communication with her manager. We assume the variances evolve over time, and the dynamics depend on how agents are connected in the hierarchy and how well they communicate. With a little abuse of notation, we use the lowercase t as the index for time, and the uppercase T to denote the hierarchy. Let $\sigma_j^2(t, T)$ be the instantaneous variance of node j on tree T , with $\sigma_j^2(0, T) = \sigma_j^2$, the intrinsic variances, for all T . The rate of change of agent j 's variance is proportional to the instantaneous difference of variance between her parent $p(j)$ and herself.¹ Formally,

$$\frac{d\sigma_j^2(t, T)}{dt} = \mu(C_{p(j)}(T))(\sigma_{p(j)}^2(t, T) - \sigma_j^2(t, T)). \quad (5.2.2)$$

The set of children of node i on tree T is denoted by $C_i(T) := \{k \in N : (i, k) \in T\}$. The communication function μ maps the set of children of the parent $p(j)$, given by $C_{p(j)}(T)$, to \mathbb{R} , and is a monotone decreasing function. Where the tree T is understood, we leave this silent to simplify notation.

To get a feel for this information propagation model, let us consider a case where the parent $p(j)$ has a smaller instantaneous variance than j , i.e., the manager knows the business potential more accurately. Under this model, agent j also understands the business better due to her connection with a better informed manager. Similarly, if the instantaneous variance of the manager was larger, then the employees managed by him would be worse informed about the business, which would reflect in their variance, even though their intrinsic variances were low.

The monotone decreasing function μ models that if a manager manages a large team, his

¹This is similar to the information ‘osmosis’ model [32].

effort per employee would be small. For the simulations, we adopt a sigmoid function:

$$\mu(C_{p(j)}) = a \cdot \frac{1}{1 + e^{s(|C_{p(j)}|-b)}}, \quad (5.2.3)$$

where the parameters $a \in [0, 1]$, $s \in \mathbb{R}_{>0}$, and $b \in \mathbb{N}$ denote the amplitude, steepness, and breadth of communication respectively. A typical plot of μ with $a = 1$, $s = 3$, $b = 4$ is given in Figure 5.10. Parameter a controls the amplitude of the function, s captures how steeply the curve falls, and b captures how many children can the parent support without significant fall in μ .

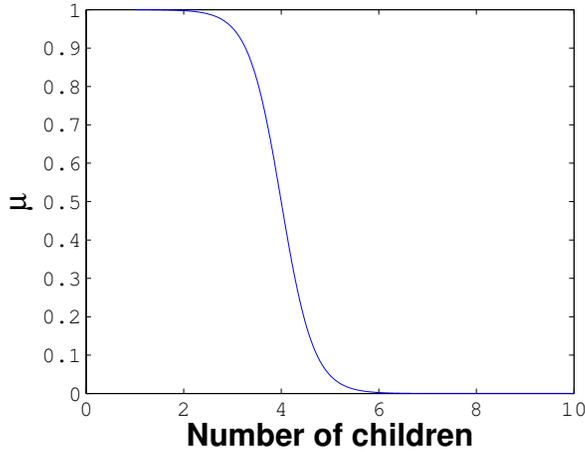


Figure 5.10: Function μ , with $a = 1$, $s = 3$, $b = 4$.

Agent Model

The agents are assumed to be strategic, and they choose their effort λ_i 's to maximize their payoff, given by Equation (5.2.1). If the reward is perfectly known to the agents, they maximize this utility function by choosing appropriate effort in a pure-strategy Nash equilibrium.

When R is observed with noise, according to the information asymmetry model (Section 5.2.1), we assume that each agent i estimates the reward perceived by the other agents. That is, each agent i tries to estimate R_j given their privately observed R_i , $\forall j \neq i$. The two random variables are related as, $R_i = R_j + \eta_i - \eta_j$. Since both η_i and η_j are zero mean Gaussians with variances σ_i^2 and σ_j^2 respectively, R_i given R_j is distributed as $\mathcal{N}(R_j, \sigma_i^2 + \sigma_j^2)$. Hence the *maximum likelihood* (MLE) estimate of R_j after observing R_i is given by R_i itself.

In this setting, we adopt a simple model of the agent i 's behavior, in which the agent acts in a certainty-equivalent way and adopts this estimate R_i in place of R , including for the purpose

of modeling the reward perceived by other agents. In particular, agent i adopts as its strategy the effort level that it would adopt in the pure strategy Nash equilibrium in a subjective model of the game, in which it assumes that every other agent adopts as the reward for the current task the MLE estimate of agent i .¹ In this section, we refer to this behavioral model as the *MLE Best Response (MBR)* model.

Performance Metric

To ensure that all incoming tasks are served in the long run, the sum of the best response efforts should be at least the incoming rate of tasks. Let us denote the best response effort of agent i under the MBR model by λ_i^{MBR} . The sum effort is a random variable under the information asymmetry model of the agents. We define the *risk* of the organization as the probability that the sum effort does not meet the arrival rate:

$$\text{Risk} := P\left(\sum_{i \in N} \lambda_i^{MBR} \leq \lambda\right) \quad (5.2.4)$$

5.2.2 Drawback: Free Riding

In this section, we analyze the effect of reward sharing on the effort level of the agents in a model with no information asymmetry, such that all agents observe R perfectly. The *effort sharing function* f is a recursive function, which is computed at the leaf first and then is computed on the nodes above till the root, for a given tree T and a reward sharing matrix Δ , defined as follows. Let us denote the set of single-rooted directed trees with n nodes by \mathcal{T} .

Definition 5.3 (Effort Sharing Function) *An effort sharing function is a mapping $f : \mathcal{T} \times [0, 1]_{n \times n} \rightarrow [0, 1]$ given by the recursive formula,*

$$f(T_i, \Delta) = \max \left\{ 0, 1 - \frac{1}{\gamma} \sum_{j \in T_i \setminus \{i\}} \delta_{ij} \cdot f(T_j, \Delta) \right\}, \quad (5.2.5)$$

where T_i is the subtree rooted at i , which includes i .

This function is maximum when i is a leaf, and decreases as we move towards the root. In Figure 5.11, we plot $f(T_i, \Delta)$ as a function of the size of the subtree T_i . The function decreases

¹This model of agent behavior is related to formal models in behavioral game theory; e.g., the *Quantal Level k (QL- k) model* [78]. In these models, agents make (different, depending on their level) simplifying assumptions about the reasoning of other agents. In our MLE best response model, agents make simplifying assumptions about the beliefs of other agents about the reward of a task.

as the size of the subtree T_i increases. The following theorem states that the equilibrium effort level of each node in a hierarchy is proportional to this function, leading to a smaller effort levels of nodes near the root.

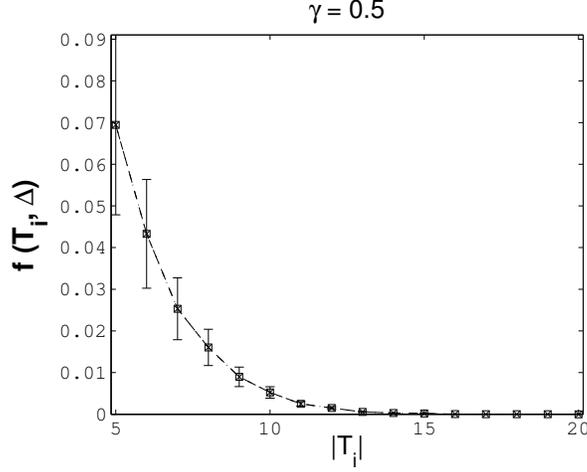


Figure 5.11: Plot of $f(T_i, \Delta)$ (Δ averaged) versus $|T_i|$.

Theorem 5.8 (Nash Equilibrium Characterization) *Under the perfect observation of the reward R , the Nash equilibrium effort profile λ^* , is uniquely given by,*

$$\lambda_i^* = \frac{\lambda \gamma R}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta) \right)^2} \right) f(T_i, \Delta), \forall i \in N, \quad (5.2.6)$$

and lies in $Z_1 = \left\{ \lambda : \sum_{j \in N} \lambda_j > \lambda \right\}$, if and only if,

$$\frac{\gamma \cdot R}{C} > \frac{\sum_{j \in N} f(T_j, \Delta)}{\sum_{j \in N} f(T_j, \Delta) - 1} \quad (5.2.7)$$

Proof: (\Rightarrow) Substituting Equation (5.2.6) in $\sum_{j \in N} \lambda_j > \lambda$ we obtain $\frac{\gamma \cdot R}{C} > \frac{\sum_{j \in N} f(T_j, \Delta)}{\sum_{j \in N} f(T_j, \Delta) - 1}$.

(\Leftarrow) We show this in two steps.

Step 1: Unique PSNE effort profile: If a PSNE effort profile $(\lambda_i^*, \lambda_{-i}^*)$ exists in the given game, then it must satisfy, $u_i(\lambda_i^*, \lambda_{-i}^*) \geq u_i(\lambda_i, \lambda_{-i}^*), \forall \lambda_i \in S_i = [0, \infty), \forall i \in N$. This implies, $\lambda_i^* = \operatorname{argmax}_{\lambda_i \in S_i = [0, \infty)} u_i(\lambda_i, \lambda_{-i}^*), \forall i \in N$.

Thus in order to find the Nash equilibrium we have to solve the following optimization problem for each $i \in N$.

$$\left. \begin{array}{l} \max_{\lambda_i} \quad u_i(\lambda_i, \lambda_{-i}^*) \\ \text{s.t.} \quad \lambda_i \geq 0, \end{array} \right\} \Rightarrow \begin{array}{l} \min_{\lambda_i} \quad -u_i(\lambda_i, \lambda_{-i}^*) \\ \text{s.t.} \quad -\lambda_i \leq 0. \end{array} \quad (5.2.8)$$

Due to concavity of u_i , this is a convex optimization problem with linear constraints, which can be solved using KKT theorem. At the minimizer λ_i^* of problem (5.2.8), $\exists \mu \in \mathbb{R}$ such that, (i) $\mu \geq 0$, (ii) $-\frac{\partial}{\partial \lambda_i} u_i(\lambda_i^*, \lambda_{-i}^*) - \mu = 0$, (iii) $-\mu \lambda_i^* = 0$, (iv) $-\lambda_i^* \leq 0$.

Case 1: $\mu > 0 \Rightarrow \lambda_i^* = 0$ and in this case $\frac{\partial}{\partial \lambda_i} u_i = -\mu \leq 0$.

Case 2: $\mu = 0 \Rightarrow \frac{\partial}{\partial \lambda_i} u_i(\lambda_i^*, \lambda_{-i}^*) = 0$ and in this case $\lambda_i^* \geq 0$. This leads us to,

$$\left(\sum_{j \in N} \lambda_j^* \right)^2 = \frac{\lambda \gamma R}{C} \left(\sum_{j \neq i} \lambda_j^* - \sum_{j \in T_i \setminus \{i\}} \lambda_j^* \delta_{ij} \right). \quad (5.2.9)$$

The above expression is obtained by differentiating u_i .

For a given tree and its equilibrium profile λ^* , let us substitute x for $\sum_{j \in N} \lambda_j^*$, then manipulation of Equation (5.2.9) leads to,

$$\lambda_i^* + \sum_{j \in T_i \setminus \{i\}} \lambda_j^* \delta_{ij} = x - \frac{x^2 C}{\lambda \gamma R}, \quad \forall i \in N. \quad (5.2.10)$$

We do another variable substitution to denote the RHS of Equation (5.2.10) by y (≥ 0 since LHS is ≥ 0). That is,

$$y = x - \frac{x^2 C}{\lambda \gamma R}. \quad (5.2.11)$$

Claim: $\lambda_i^* = y f(T_i, \Delta)$, $\forall i \in N$.

Proof: We prove this claim via induction on the levels of T . Let the depth of T be D . From Equation (5.2.10), $\lambda_i^* + \sum_{j \in T_i \setminus \{i\}} \delta_{ij} \lambda_j^* = y$, $\forall i \in N$. From Cases 1 and 2 above,

$$\lambda_i^* = \max \left(0, y - \sum_{j \in T_i \setminus \{i\}} \delta_{ij} \lambda_j^* \right) \quad \forall i \in N. \quad (5.2.12)$$

Step 1: For an arbitrary node j at level D , from Equation (5.2.12), $\lambda_j^* = y$. Hence, the proposition is true as $f(T_j, \Delta) = 1$ for a leaf. Now, select an arbitrary node i (which is not a leaf) at level $D - 1$. From Equation (5.2.12) we get, $\lambda_i^* = \max(0, y - \sum_{j \in T_i \setminus \{i\}} \delta_{ij} y) = y \max(0, 1 - \sum_{j \in T_i \setminus \{i\}} \delta_{ij} 1) = y f(T_i, \Delta)$.

Step 2: Let $\lambda_j^* = yf(T_j, \Delta)$ be true for all nodes j upto level $D - l$. Consider an arbitrary node i at level $D - l - 1$. From Equation (5.2.12) and Equation (5.2.5),

$$\lambda_i^* = \max \left(0, y - \sum_{j \in T_i \setminus \{i\}} y \cdot f(T_j, \Delta) \delta_{ij} \right) = yf(T_i, \Delta) \quad (5.2.13)$$

which concludes the induction. ■

To find an expression for PSNE we now evaluate y . The sum of efforts of all the players is defined as x . Hence, $x = y \sum_{j \in N} f(T_j, \Delta)$. Substituting for y from Equation (5.2.11) in this expression and solving for x yields,

$$x = \sum_{j \in N} \lambda_j^* = \frac{\lambda \gamma R}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\sum_{j \in N} f(T_j, \Delta)} \right). \quad (5.2.14)$$

Using Equation (5.2.11) we get,

$$y = \frac{\lambda \gamma R}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta) \right)^2} \right). \quad (5.2.15)$$

Combining Equation (5.2.15) and the claim above, the PSNE is given by,

$$\lambda_i^* = \frac{\lambda \gamma R}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta) \right)^2} \right) f(T_i, \Delta), \forall i \in N.$$

KKT equations led to a unique solution of the optimization problem, hence PSNE is unique.

Step 2: $\sum_{j \in N} \lambda_j > \lambda$: We use Equation (5.2.6) to compute the sum $\sum_{j \in N} \lambda_j^*$, and use the fact that $\gamma R/C > \frac{\sum_{j \in N} f(T_j, \Delta)}{\sum_{j \in N} f(T_j, \Delta) - 1}$ to get $\sum_{j \in N} \lambda_j^* > \lambda$. ■

Example 5.1 To get a feel for how free riding can affect the total effort in the system, let us look at the specific example of of Figure 5.12. We consider the geometric reward sharing with factor δ (see Section 5.2.1), and plot the equilibrium sum effort level with increasing δ . If $\delta \approx 0.4$, and no free riding happens (that is, pretend $\delta = 0$ in the figure), in order to reach a sum effort level of 8 the organization would have needed a reward of $R = 10$. However, because of free riding phenomenon, to attain the same sum effort level, one needs to put $R = 20$, see the circled curve on the top at $\delta \approx 0.4$. □

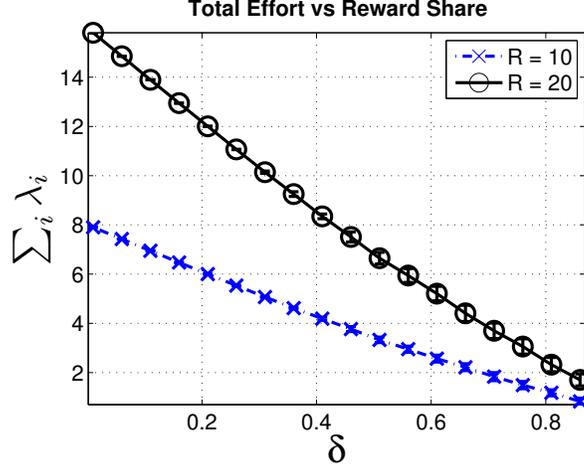


Figure 5.12: Plot of $\sum_{i \in N} \lambda_i^*$ versus the geometric reward share factor, δ .

5.2.3 Benefit: Information Sharing

In this section, we discuss the benefit of placing asymmetrically informed nodes in a hierarchy, for now studying this in the absence of free riding behavior. The goal is to understand one of the ‘pure’ advantages of having a hierarchy. In particular, while agents still aim to maximize their payoffs, we restrict them to ignore the reward share coming from the subtree below them (and thus the indirect reward).

We assume that the reward R is sufficient to guarantee $\gamma R/C > \frac{n}{n-1}$ (n is the number of nodes) with high probability (i.e., approaches 1 in the limit). Since we ignore indirect rewards in this section, agent i perceives the utility:

$$u_i(\lambda_i, \lambda_{-i}) = \lambda \gamma R_i \frac{\lambda_i}{\sum_{k \in N} \lambda_k} - C \lambda_i.$$

The above utility is a modification of Equation (5.2.1), which does not contain passive reward terms, due to our assumption that the nodes do not free ride. Also the reward R is replaced by R_i because of the MBR agent behavior model.

Taking the partial derivative of the above equation w.r.t. λ_i and equating it to zero, and solving the set of equations similar to the ones in Section 5.2.2, we get,

$$\lambda_i^{MBR} = \frac{\lambda \gamma R_i n - 1}{C} \frac{n - 1}{n^2} \Rightarrow \sum_{i \in N} \lambda_i^{MBR} = \frac{\lambda \gamma \sum_i R_i n - 1}{C} \frac{n - 1}{n^2}.$$

Notice that, R_i ’s are conditionally independent given R , and are distributed as $\mathcal{N}(0, \sigma_i^2(t))$.

Hence, the sum effort $\sum_{i \in N} \lambda_i^{MBR}$, is also distributed as a Gaussian random variable given R , with mean m_{total} and variance $\sigma_{\text{total}}^2(t)$, such that,

$$\begin{aligned} m_{\text{total}} &= \frac{\lambda \gamma R n - 1}{C} \frac{1}{n}, \\ \sigma_{\text{total}}^2(t) &= \left(\frac{\lambda \gamma}{C} \right)^2 \left(\frac{n-1}{n^2} \right)^2 \sum_i \sigma_i^2(t). \end{aligned} \quad (5.2.16)$$

Since, $\gamma R/C > \frac{n}{n-1}$, it implies, $m_{\text{total}} > \lambda$. Hence, using the upper bound on the tail distribution of the standard normal distribution, we can show the following bound on the risk:

$$P \left(\sum_{i \in N} \lambda_i^{MBR} \leq \lambda \right) = O \left(\frac{\sigma_{\text{total}}(t)}{\epsilon} e^{-\epsilon^2/2\sigma_{\text{total}}^2(t)} \right), \quad (5.2.17)$$

where, $\epsilon = m_{\text{total}} - \lambda > 0$. To minimize risk, we want the RHS of the above equation to be small. Let us define the *risk bound ignoring free riding (IFR)* as,

$$\rho_{\text{IFR}}(t) := \frac{\sigma_{\text{total}}(t)}{\epsilon} e^{-\epsilon^2/2\sigma_{\text{total}}^2(t)}. \quad (5.2.18)$$

The risk is bounded by a constant factor of the above quantity. This is a function of the total variance $\sigma_{\text{total}}^2(t)$, which has been calculated ignoring the free riding effect in this section, and hence this expression does not consider free riding effect. Since $\rho_{\text{IFR}}(t)$ is a function of time, we should consider its rate of fall. However, if the magnitude of the risk bound remains large, then the rate alone is not a good measure of how the actual risk falls. Therefore, a good metric to quantify both the rate and magnitude of the risk is the fractional rate, $\frac{1}{\rho_{\text{IFR}}(t)} \frac{d\rho_{\text{IFR}}(t)}{dt}$.

For this reason, we formulate a hierarchy design problem as the problem of finding the tree T_{min} that minimizes the fractional rate of fall of the risk bound at time $t = 0$. Formally, denoting the set of all single rooted trees with n nodes by \mathcal{T} , we get the following *Optimization problem Ignoring Free Riding (OPT-IFR)*:

$$T_{\text{min}} \in \operatorname{argmin}_{T \in \mathcal{T}} \frac{1}{\rho_{\text{IFR}}(t)} \frac{d\rho_{\text{IFR}}(t)}{dt} \Big|_{t=0} \quad \text{OPT-IFR} \quad (5.2.19)$$

The fractional rate of fall of $\rho_{\text{IFR}}(t)$ at time $t = 0$ can be shown to be:

$$\frac{1}{\rho_{\text{IFR}}(t)} \frac{d\rho_{\text{IFR}}(t)}{dt} \Big|_{t=0} = \frac{1}{\sigma_{\text{total}}(0)} \left(1 + \frac{\epsilon^2}{\sigma_{\text{total}}^2(0)} \right) \cdot \frac{d\sigma_{\text{total}}^2(t)}{dt} \Big|_{t=0}$$

The rate of fall in the total variance is given by,

$$\begin{aligned}
 \left. \frac{d\sigma_{\text{total}}^2(t)}{dt} \right|_{t=0} &\propto \sum_{i \in N} \left. \frac{d\sigma_i^2(t)}{dt} \right|_{t=0} \\
 &= \sum_{i \in N} \mu(C_{p(i)}) (\sigma_{p(i)}^2(t) - \sigma_i^2(t)) \Big|_{t=0} \\
 &= \sum_{i \in N} \mu(C_{p(i)}) (\sigma_{p(i)}^2 - \sigma_i^2).
 \end{aligned} \tag{5.2.20}$$

The first proportionality comes from the Equation (5.2.16). The second equality comes due to the information propagation model (Equation (5.2.2)). The parent of i is denoted by $p(i)$ (and unique due to the tree structure).

Example 5.2 Under this model, the benefit of the hierarchy comes from the fact that it helps reduce the uncertainty levels of the employees, and thereby reduces the risk. Figure 5.13 shows a typical hierarchy formed using Equation (5.2.19) with 6 nodes with a typical set of intrinsic variances. We will show in the next section that the hierarchy changes when we consider both information propagation and free riding of the nodes jointly into account. \square

Ignoring FR, Rate of Fall = -0.3466, $\delta = 0.25$

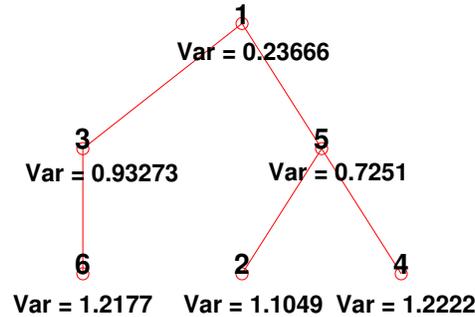


Figure 5.13: Hierarchy formed when free riding is not a concern.

5.2.4 Trade-off between Free-riding and Information Sharing

In the previous two sections, we have addressed separately the issues of free-riding and information sharing. In an actual hierarchical organization, both these effects take place simultaneously, making it necessary to consider both aspects while designing a hierarchy.

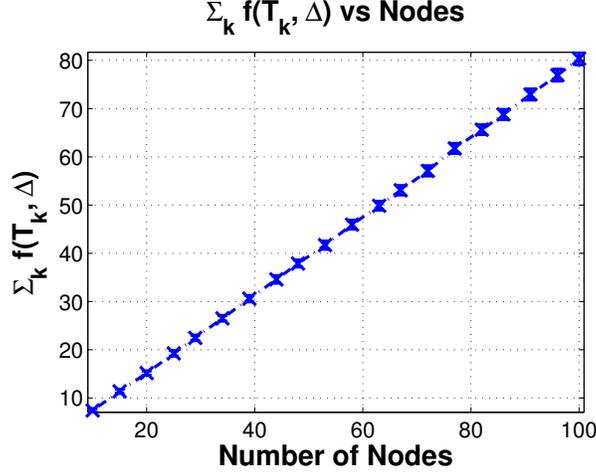


Figure 5.14: Plot of $\sum_{i \in N} f(T_i, \Delta)$ (Δ averaged) vs n .

We consider hierarchies with n nodes. The sum, $\sum_{i \in N} f(T_i, \Delta)$ is increasing in n (see Fig. 5.14). Hence, $\frac{\sum_{i \in N} f(T_i, \Delta)}{\sum_{i \in N} f(T_i, \Delta) - 1} \downarrow n$ and approaches 1 in the limit. If we assume that the potential reward R is distributed such that with high probability (w.h.p.), $\gamma R/C \geq (1 + \epsilon)$, where $\epsilon > 0$ is any fixed number, then for large enough n , the parameters will satisfy Eq. (5.2.7) w.h.p.

Lemma 5.5 *For a population of agents with MBR behavioral model, the MBR effort of agent i is given by,*

$$\lambda_i^{MBR} = \frac{\lambda \gamma R_i}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta) \right)^2} \right) f(T_i, \Delta), \quad \forall i \in N, \forall T. \quad (5.2.21)$$

Proof: By definition of the MBR model (Section 5.2.1), an agent i makes the MLE estimate of the reward observed by agent j given her own observed reward R_i , which is R_i . Hence, the utility of agent i is given by Equation (5.2.1), with R replaced by R_i .

$$u_i(\lambda_i, \lambda_{-i}) = \lambda \gamma R_i \frac{\lambda_i}{\sum_{k \in N} \lambda_k} + \lambda R_i \sum_{j \in T_i \setminus \{i\}} \delta_{ij} \frac{\lambda_j}{\sum_{k \in N} \lambda_k} - C \lambda_i. \quad (5.2.22)$$

To maximize this w.r.t. λ_i , we follow a similar analysis technique as in the proof of Theorem 5.8, by writing down the constrained optimization problem and the KKT conditions, and solving the set of equations we get the expression of Equation (5.2.21). ■

Note: Lemma 5.5 does not claim the equilibrium result given by Theorem 5.8, but uses the same analysis technique to solve the set of equations and find the best response effort of agent i , given by λ_i^{MBR} , that maximizes the utility in Equation (5.2.22).

The following theorem presents the distribution of the total effort of the nodes in a hierarchy, and gives the expressions for mean and variance.

Theorem 5.9 *The sum effort of a population of agents with MBR behavior under the information asymmetry and reward sharing model on a hierarchy T , is distributed as $\mathcal{N}(m_{total}, \sigma_{total}^2(t))$, where for large n , $m_{total} \approx \frac{\lambda\gamma R}{C}$, and $\sigma_{total}^2(t) = \Theta\left(\frac{1}{n^2} \sum_{i \in N} f^2(T_i, \Delta) \sigma_i^2(t)\right)$.*

Proof: From Lemma 5.5 we get,

$$\begin{aligned} \sum_{i \in N} \lambda_i^{MBR} &= \sum_{i \in N} \frac{\lambda\gamma R_i}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta)\right)^2} \right) f(T_i, \Delta) \\ &= \sum_{i \in N} \frac{\lambda\gamma}{C} \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta)\right)^2} \right) f(T_i, \Delta) \cdot (R + \eta_i) \end{aligned} \quad (5.2.23)$$

Conditioned upon the true reward R , this sum is also distributed as Gaussian with mean and variance as follows.

$$\begin{aligned} m_{total} &:= \mathbb{E} \left[\sum_{i \in N} \lambda_i^{MBR}(T, R_i) \mid R \right] \\ &= \frac{\lambda\gamma R \sum_{j \in N} f(T_j, \Delta) - 1}{C \sum_{j \in N} f(T_j, \Delta)} \end{aligned} \quad (5.2.24)$$

$$\begin{aligned} \sigma_{total}^2(t) &:= \text{Var} \left[\sum_{i \in N} \lambda_i^{MBR}(T, R_i) \mid R \right] \\ &= \left(\frac{\lambda\gamma}{C} \right)^2 \left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta)\right)^2} \right)^2 \sum_{i \in N} f^2(T_i, \Delta) \sigma_i^2(t) \end{aligned} \quad (5.2.25)$$

For large enough n , $\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\sum_{j \in N} f(T_j, \Delta)} \approx 1$, and,

$$\left(\frac{\sum_{j \in N} f(T_j, \Delta) - 1}{\left(\sum_{j \in N} f(T_j, \Delta) \right)^2} \right)^2 = \Theta(1/n^2).$$

Hence, we have proved the theorem. ■

To minimize the risk of the organization, the approach of minimizing the risk bound (Equation (5.2.18)) is followed. However, the risk bound considering free riding (CFR) is different from that of Equation (5.2.19) since the total variance $\sigma_{\text{total}}^2(t)$ now considers the free riding effect and is different (given by Equation (5.2.25) as opposed to Equation (5.2.16)). The modified risk bound is:

$$\rho_{\text{CFR}}(t) := \frac{\sigma_{\text{total}}(t)}{\epsilon} e^{-\epsilon^2/2\sigma_{\text{total}}^2(t)}. \quad (5.2.26)$$

The modified design goal is to identify the hierarchy that minimizes the fractional fall rate of the risk bound at time $t = 0$. Hence the *Optimization problem Considering Free Riding (OPT-CFR)* is given by:

$$T_{\min} \in \operatorname{argmin}_{T \in \mathcal{T}} \frac{1}{\rho_{\text{CFR}}(t)} \frac{d\rho_{\text{CFR}}(t)}{dt} \Big|_{t=0} \quad \text{OPT-CFR} \quad (5.2.27)$$

Following similar steps to the analysis following Equation (5.2.19), and using the expression of $\rho_{\text{CFR}}(t)$ from Equation (5.2.26), we can show that the RHS of the above optimization problem is given by,

$$\frac{1}{\rho_{\text{CFR}}(t)} \frac{d\rho_{\text{CFR}}(t)}{dt} \Big|_{t=0} = \frac{1}{\sigma_{\text{total}}(0)} \left(1 + \frac{\epsilon^2}{\sigma_{\text{total}}^2(0)} \right) \cdot \frac{d\sigma_{\text{total}}^2(t)}{dt} \Big|_{t=0}$$

Taking the derivative of the total variance given by Theorem 5.9, and following the steps as in Equation (5.2.25), we have:

$$\frac{d\sigma_{\text{total}}^2(t)}{dt} \Big|_{t=0} \propto \sum_{i \in N} f^2(T_i, \Delta) \mu(C_{p(i)}) (\sigma_{p(i)}^2 - \sigma_i^2). \quad (5.2.28)$$

We now see that the term accounting for the free riding effect, $f^2(T_i, \Delta)$, appears in the sum in the above equation, which makes it different from Equation (5.2.20). The rate of change of the total variance is now a weighted sum of the squared effort sharing function f , and therefore

Considering FR, Rate of Fall = -0.39759 , $\delta = 0.25$

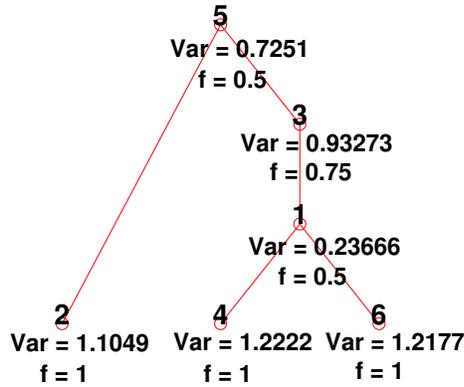


Figure 5.15: Hierarchy formed considering free riding.

the output of OPT-CFR will potentially be a different hierarchy from the OPT-IFR. In the following section we show the difference between the hierarchies found by these two optimization problems.

Example 5.3 To show how the structure of the hierarchy changes when one designs a network considering the free riding effect, we take the same set of nodes as in Figure 5.13, but now solve the optimization problem given by Equation (5.2.27), and find the hierarchy as shown in Figure 5.15. \square

We compare the performances of the OPT-IFR and OPT-CFR. Figure 5.16 plots the fractional fall rate w.r.t. the amplitude factor of the function μ and shows that the fall rate is faster in OPT-CFR. Figure 5.17 shows that the difference in the fall rate between OPT-CFR and OPT-IFR is around 10% for 5 nodes and increases with the number of nodes.

5.2.5 An Algorithm for Finding a Near Optimal Hierarchy

Equation (5.2.27) gives a method of jointly optimizing over the free riding and information propagation phenomena in hierarchies. It is natural to ask how one can solve the optimization problem to find an optimal hierarchy. An exhaustive search algorithm would iterate over the space of all single rooted directed trees with n nodes. To find an optimal hierarchies in the simulations of this section, we have used the exhaustive search. However, a simple *sequential greedy* algorithm, given in tabular form in Algorithm 3, runs in $O(n^3)$ time and gives reasonably good approximation to an optimal hierarchy. Figure 5.18 shows the performance comparison of the rate of fall of the fractional risk bound given by the greedy algorithm with that of an optimal hierarchy for 6 nodes with varying average noise variance of the population.

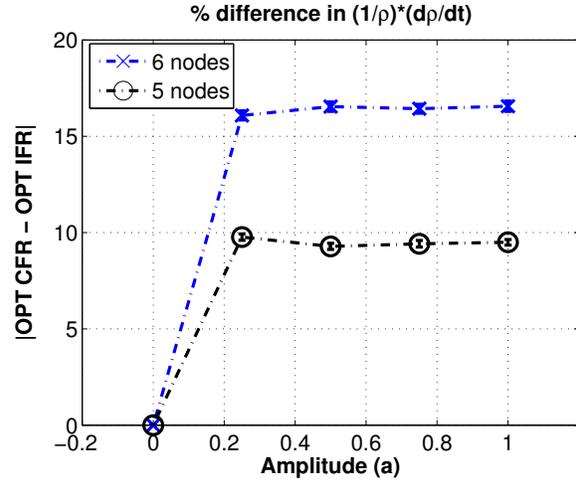
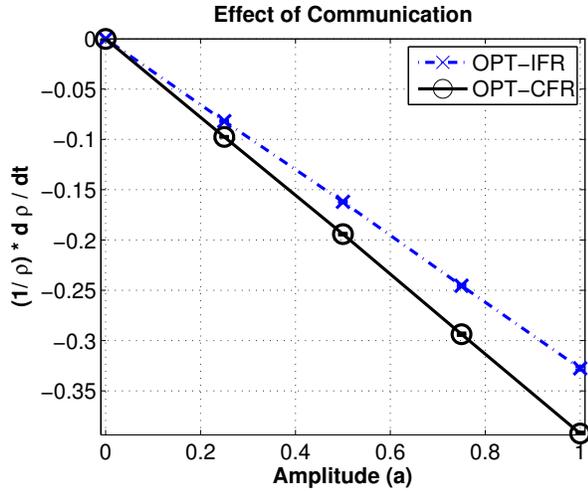


Figure 5.16: Absolute difference in the fractional decay rate of risk bound in OPT-CFR with OPT-IFR.

Figure 5.17: Percentage difference in the fractional decay rate of risk bound in OPT-CFR with OPT-IFR.

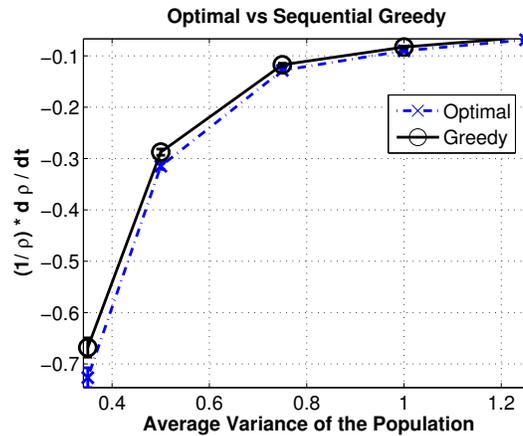


Figure 5.18: Fractional decay rate of the risk bound $\rho_{CFR}(t, T)$ comparison between the Optimal and Sequential Greedy algorithms (Algorithm 3). The x-axis shows different average variances of the population. Number of nodes = 6.

5.3 Conclusions

In the first part of this chapter, we built on the papers by Bramoullé and Kranton [14], Ballester et al. [7] and develop an understanding of the effort levels in influencer-influencee networks. Taking a game theoretic perspective, we introduced a general utility model which results in a non-concave game, but are able to show results on the existence and uniqueness of Nash

Algorithm 3 Sequential Greedy

Input: A set of n nodes with intrinsic variances σ_i^2 's

Output: A tree connecting the nodes

Step 1: Sort nodes in ascending order of variances

Step 2:

for Both configuration of *the first two nodes* **do**

for Nodes $j \leftarrow 3$ to n **do**

 Sequentially add j as a child of one of the previous nodes, such that the total variance of the subnetwork after adding j is minimum

end for

end for

Step 3: Output the tree giving the lowest total variance with n nodes

equilibrium efforts. For the ease of exposition, we focused on hierarchical networks, and with the EP model we found closed form expressions and bounds on the PoA for balanced hierarchies. These results provide us the insight on the importance of communication in hierarchies on the design of efficient networks. At the same time, for a given network structure and communication level, we provide a design recipe for the reward sharing in order to achieve highly productive output, and thereby minimize the PoA.

The connection between matrix stability and uniqueness of Nash equilibria that arose in our work, is of particular interest to us for future research. In particular, for the general networks there was a direct interpretation of the uniqueness condition in terms of a Jacobian matrix stability. This stability property is directly related to the contraction property that shows that agents following local updates on effort levels will converge to the Nash equilibrium another desirable property. Pursuing these connections in the investigation of organizational network formation games is an important direction of future research.

In the second part of this chapter, we have studied organizational hierarchies to understand the effects of competition, information asymmetry, and reward sharing on the performance of organizations, for a simple queuing-theoretic model. Our analysis has highlighted a trade-off between the inefficiencies that come from free riding and the benefits that come from information sharing, and thus improving the accuracy of agent models for the reward of tasks. We have demonstrated the effect of considering both phenomena on the design of organizational hierarchies.

These two complementary approaches provide a theoretical foundation of the crowdsourcing environment looked from an organization theoretic viewpoint. It shows that both the reward sharing and the structure of the network has important implications in the rational choice of efforts of the agents, leading to the net productive output of the crowdsourcing network.

APPENDICES

5.A Proofs for the Exponential Productivity Model

Proof of Theorem 5.1

Proof: The argument for the existence of a Nash equilibrium is straightforward in this particular setting. We see that because of the hierarchical structure of the network, the leaf nodes will always put unit effort, i.e., $\mathbf{x}_{\text{leaves}}^* = 1$. To compute the equilibrium in the level above the leaves one can run a backward induction algorithm to maximize Equation (5.1.1) at each level, where the equilibrium efforts in the levels below is already computed by the algorithm. Since, all p_i 's are bounded and the maximization is over $x_i \in [0, 1]$, a compact space, maxima always exists. Hence, a Nash equilibrium always exists.

Now we show that a Nash equilibrium profile (x_i^*, x_{-i}^*) must satisfy Equation (5.1.4). For notational convenience, we drop the arguments of p_i and p_{ij} , which are functions of $\mathbf{x}_{P_{\theta \rightarrow i}}$ and $\mathbf{x}_{P_{i \rightarrow j}}$ respectively. Each agent $i \in N$ solves the following optimization problem.

$$\begin{aligned} \max_{x_i} \quad & u_i(x_i, x_{-i}) \\ \text{s.t.} \quad & x_i \geq 0 \end{aligned} \tag{5.A.1}$$

Combining Equations (5.1.1), (5.1.2), and (5.1.3), we get,

$$u_i(x_i, x_{-i}) = p_i(\mathbf{x}_{P_{\theta \rightarrow i}}) \left(x_i - \frac{x_i^2}{2} - b \frac{(1 - x_i)^2}{2} \right) + \sum_{j \in T_i \setminus \{i\}} h_{ij} p_j(\mathbf{x}_{P_{\theta \rightarrow j}}) x_j.$$

Note that we have relaxed the constraint from $0 \leq x_i \leq 1$. The first additive term in the utility function has the peak at $x_i = 1$. The second term has $e^{\beta x_i}$ in the p_j , which is decreasing in x_i . Therefore, an optimal x_i that maximizes this utility will be ≤ 1 . Hence, in this problem setting, an optimal solution for both the exact and the relaxed problems is the same. So, it is enough to consider the above problem. For this non-linear optimization problem, we can write down the Lagrangian as follows.

$$\mathcal{L} = u_i(x_i, x_{-i}) + \lambda_i x_i, \quad \lambda_i \geq 0.$$

The KKT conditions for this optimization problem (5.A.1) are:

$$\frac{\partial \mathcal{L}}{\partial x_i} = 0, \Rightarrow \frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) + \lambda_i = 0, \tag{5.A.2}$$

$$\lambda_i x_i = 0, \tag{5.A.3} \quad \text{complementary slackness.}$$

Case 1: $\lambda_i = 0$, then from Equation (5.A.2) we get,

$$\begin{aligned}
& p_i(1 - x_i + b(1 - x_i)) + \sum_{j \in T_i \setminus \{i\}} h_{ij} \frac{\partial p_j}{\partial x_i} x_j = 0 \\
\Rightarrow & p_i(1 + b)(1 - x_i) - \beta \sum_{j \in T_i \setminus \{i\}} h_{ij} p_j x_j = 0 \\
\Rightarrow & 1 - x_i = \frac{\beta}{1 + b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij} x_j, \text{ with } p_{ij} \text{ as defined} \\
\Rightarrow & x_i = 1 - \frac{\beta}{1 + b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij} x_j. \tag{5.A.4}
\end{aligned}$$

Case 2: $\lambda_i > 0$, then from Equation (5.A.3) we get $x_i = 0$, and from Equation (5.A.2), $\frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) < 0$. Carrying out the differentiation as in Equation (5.A.4) we get,

$$0 = x_i > 1 - \frac{\beta}{1 + b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij} x_j. \tag{5.A.5}$$

$$\therefore x_i = \left[1 - \frac{\beta}{1 + b} \sum_{j \in T_i \setminus \{i\}} h_{ij} p_{ij} x_j \right]^+.$$

Since this condition has to hold for all nodes $i \in N$, the equilibrium profile (x_i^*, x_{-i}^*) must satisfy the above equality. ■

Proof of Theorem 5.2

We prove this theorem via the following Lemma.

Lemma 5.6 *If $\beta < \sqrt{\frac{1+b}{h_{\max}(T)}}$, the function F is a contraction.*

Proof: The Taylor series expansion of g with a first order remainder term is as follows. There exists a point \mathbf{x}_0 that lies on the line joining \mathbf{x} and \mathbf{y} , such that,

$$g(\mathbf{x}) = g(\mathbf{y}) + \nabla g(\mathbf{x}_0) \cdot (\mathbf{x} - \mathbf{y}).$$

Where, $\nabla g(\mathbf{x}_0)$ is the Jacobian matrix.

$$\nabla g(\mathbf{x}_0) = \left(\begin{array}{ccc} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n} \end{array} \right) \Big|_{\mathbf{x}_0}$$

In order to show that F is a contraction, we note that F is a truncation of g . Hence, $\|F(\mathbf{x}) - F(\mathbf{y})\| \leq \|g(\mathbf{x}) - g(\mathbf{y})\|$, for all $\mathbf{x}, \mathbf{y} \in [0, 1]^n$. Let us consider the following term,

$$\|F(\mathbf{x}) - F(\mathbf{y})\| \leq \|g(\mathbf{x}) - g(\mathbf{y})\| \leq |\nabla g(\mathbf{x}_0)| \cdot \|\mathbf{x} - \mathbf{y}\| \quad (5.A.6)$$

Where the matrix norm $|\nabla g(\mathbf{x}_0)|$ is the largest singular value of the Jacobian matrix $\nabla g(\mathbf{x}_0)$. We see that in our special structure in the problem, this matrix is upper triangular, hence the diagonal elements are the singular values. Suppose, the k -th diagonal element yields the largest singular value.

$$\begin{aligned} |\nabla g(\mathbf{x}_0)| &= \left. \frac{\partial g_k}{\partial x_k} \right|_{\mathbf{x}_0} = \frac{\beta^2}{1+b} \sum_{j \in T_k \setminus \{k\}} h_{kj} p_{kj} x_j \Big|_{\mathbf{x}_0} \\ \Rightarrow \sup_{x_0} |\nabla g(\mathbf{x}_0)| &\leq \frac{\beta^2}{1+b} \cdot h_{\max}(T) < 1, \quad \text{since } \beta^2 < \frac{1+b}{h_{\max}(T)}. \end{aligned}$$

The first inequality above holds due to the fact that p_{kj} 's and x_j 's are ≤ 1 , and by the definition of $h_{\max}(T)$. Hence, from Equation (5.A.6), we get that F is a contraction. \blacksquare

Proof: [of Theorem 5.2] We know from Theorem 5.1 that a Nash equilibrium exists. Under the sufficient condition given by Lemma 5.6, the fixed point of $\mathbf{x} = F(\mathbf{x})$ is unique. Therefore, the Nash equilibrium is also unique, and is given by Equation (5.1.4). \blacksquare

5.B Proofs of the Price of Anarchy Results in Balanced Hierarchies

Proof of Theorem 5.5

We prove this theorem via the following lemma, which finds out an optimal effort profile for β above a threshold.

Lemma 5.7 (Optimal Efforts) *For a balanced d -ary hierarchy with depth D , any optimal effort profile has $x_{D+1}^{OPT} = 1$. When $-\ln(1 - \frac{1}{d}) \leq \beta < \infty$, an optimal effort profile is $x_i^{OPT} = 0$, $\forall i = 1, \dots, D$, and $x_{D+1}^{OPT} = 1$.*

Proof: The social outcome for a given effort vector \mathbf{x} on the balanced hierarchy is as follows. Since, the hierarchy is understood here, we use $SO(\mathbf{x})$ instead of $SO(\mathbf{x}, \text{BALANCED})$.

$$SO(\mathbf{x}) = x_1 + d e^{-\beta x_1} x_2 + d^2 e^{-\beta(x_1+x_2)} x_3 + \dots + d^D e^{-\beta(\sum_{i=1}^D x_i)} x_{D+1}.$$

It is clear that for any effort profile of the other nodes the effort at the leaves that maximizes the above expression is $x_{D+1} = 1$. This proves the first part of the lemma. Hence we can simplify the above expression by,

$$\begin{aligned}
SO(\mathbf{x}) &= x_1 + de^{-\beta x_1}x_2 + d^2e^{-\beta(x_1+x_2)}x_3 + \dots + d^D e^{-\beta(\sum_{i=1}^D x_i)} \\
&= x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-1}e^{-\beta(\sum_{i=1}^{D-1} x_i)}(x_D + de^{-\beta x_D}) \\
&\leq x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-1}e^{-\beta(\sum_{i=1}^{D-1} x_i)} \cdot d.
\end{aligned} \tag{5.B.1}$$

The last inequality occurs since $\beta \geq -\ln(1 - \frac{1}{d})$, and $x_D = 0$ meets this inequality with an equality. Also since $\beta \geq -\ln(1 - \frac{1}{d})$ implies that $\beta \geq -\ln(1 - \frac{1}{d^k})$, for all $k \geq 2$, the next inequality will also be met by $x_{D-1} = 0$ as shown below.

$$\begin{aligned}
SO(\mathbf{x}) &= x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-1}e^{-\beta(\sum_{i=1}^{D-1} x_i)} \cdot d \\
&= x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-2}e^{-\beta(\sum_{i=1}^{D-2} x_i)}(x_{D-1} + d^2e^{-\beta x_{D-1}}) \\
&\leq x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-2}e^{-\beta(\sum_{i=1}^{D-2} x_i)} \cdot d^2.
\end{aligned}$$

This inequality is also achieved by $x_{D-1} = 0$. We can keep on reducing the terms from the right in the RHS of the above equation, and in all the reduced forms, $x_i = 0, i = D-1, D-2, \dots, 1$ will maximize the social output expression. Hence proved. \blacksquare

Proof: [of Theorem 5.5] *Case 1* ($0 \leq \beta \leq 1$): From Lemma 5.7, $x_{D+1} = 1$ for optimal effort. However, for any equilibrium effort profile $x_{D+1} = 1$ as well. Therefore we consider the equilibrium effort of the nodes at level D .

$$x_D = 1 - \frac{\beta}{1+b}de^{-\beta x_D}h_{D,D+1}. \tag{5.B.2}$$

The constraint for unique equilibrium demands that $dh_{D,D+1} \leq (1+b)/\beta^2$, which makes $\frac{\beta}{1+b}dh_{D,D+1} \leq 1/\beta$, while $1/\beta \geq 1$. So, we have the liberty of choosing the right $h_{D,D+1}$ to achieve any $x_D \in [0, 1]$, and in particular, the x_D^{OPT} . We apply backward induction on the next level above.

$$x_{D-1} = 1 - \frac{\beta}{1+b}[de^{-\beta x_{D-1}}x_D h_{D-1,D} + d^2e^{-\beta(x_{D-1}+x_D)}h_{D-1,D+1}].$$

The constraints are $dh_{D-1,D} + d^2h_{D-1,D+1} \leq (1+b)/\beta^2$. We claim that any $x_{D-1} \in [0, 1]$ is

achievable here as well. To show that, put $h_{D-1,D} = 0$. The above equation becomes then,

$$\begin{aligned}
x_{D-1} &= 1 - \frac{\beta}{1+b} d^2 e^{-\beta(x_{D-1}+x_D)} h_{D-1,D+1} \\
&= 1 - \frac{\beta}{1+b} d^2 e^{-\beta x_{D-1}} \frac{1+b}{d\beta h_{D,D+1}} h_{D-1,D+1}, \text{ from the earlier expression} \\
&= 1 - \frac{dh_{D-1,D+1}}{h_{D,D+1}} e^{-\beta x_{D-1}}
\end{aligned}$$

This again can satisfy any x_{D-1} , since the coefficient of the exponential term can be made anywhere between 0 and 1. It can be made 0 by choosing $h_{D-1,D+1} = 0$, and 1 by choosing $\frac{dh_{D-1,D+1}}{h_{D,D+1}} = 1$ which is feasible, since $d^2 h_{D-1,D+1} = dh_{D,D+1} \leq (1+b)/\beta^2$.

In the similar way we can continue the induction till the root and can make $\mathbf{x}^* = \mathbf{x}^{\text{OPT}}$. Hence, $\text{PoA} = 1$.

Case 2 ($1 < \beta < \infty$): We note that this region of β falls in the region specified by Lemma 5.7. Hence an optimal effort is 1 for all the leaves and 0 for everyone else. Hence, an optimal social output is given by d^D . The equilibrium effort for the leaves, $x_{D+1} = 1$. However, Equation (5.B.2) may not be satisfiable for any x_D since $1/\beta < 1$. In order to push the solution as close to zero as possible, we choose $h_{D,D+1} = (1+b)/\beta^2$, and plug it in Equation (5.B.2), and the solution is given by $\xi(\beta)$ (recall Equation (5.1.9)) and the solution set is singleton under this condition. The social output is d^D , which is the numerator of the PoA expression. The denominator is given by the social output at the Nash equilibrium, which we will try to lower bound. From Equation (5.B.1), for the equilibrium, we know that $x_D = \xi(\beta)$. Therefore,

$$x_D + de^{-\beta x_D} = x_D + d\beta(1 - x_D) = d\beta + (1 - d\beta)\xi(\beta).$$

At the same time, we see that the leftmost expression is convex in x_D , which can be lower bounded by the minima, given by,

$$x_D + de^{-\beta x_D} \geq \frac{1}{\beta}(1 + \ln(d\beta)).$$

Combining the two, a tight lower bound of the expression would be,

$$x_D + de^{-\beta x_D} \geq \max \left\{ \frac{1}{\beta}(1 + \ln(d\beta)), d\beta + (1 - d\beta)\xi(\beta) \right\} = \phi(d, \beta).$$

Plugging this lower bound in Equation (5.B.1), we see that,

$$\begin{aligned}
SO(\mathbf{x}) &\geq x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-1}e^{-\beta(\sum_{i=1}^{D-1}x_i)} \cdot \phi(d, \beta) \\
&= x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-2}e^{-\beta(\sum_{i=1}^{D-2}x_i)} \cdot (x_{D-1} + d\phi(d, \beta)e^{-\beta x_{D-1}})
\end{aligned}$$

Let us consider the last term within parenthesis.

$$\begin{aligned}
&x_{D-1} + d\phi(d, \beta)e^{-\beta x_{D-1}} \\
&= x_{D-1} + d\phi(d, \beta)\frac{\beta}{\xi(\beta)}(1 - x_{D-1}) \\
&\geq x_{D-1} + d\phi(d, \beta)\beta(1 - x_{D-1}), \quad \text{as } \xi(\beta) \leq 1 \\
&= d\phi(d, \beta)\beta + (1 - d\phi(d, \beta)\beta)x_{D-1} \\
&\geq d\phi(d, \beta)\beta + (1 - d\phi(d, \beta)\beta)\xi(\beta)
\end{aligned}$$

The first equality comes since we can make the equilibrium x_{D-1} s.t., $x_{D-1} = 1 - \frac{\xi(\beta)}{\beta}e^{-\beta x_{D-1}}$, by choosing $dh_{D-1,D} = (1 + b)/\beta^2$, $d^2h_{D-1,D+1} = 0$. Also, since $\xi(\beta) \leq 1$, we conclude, $x_{D-1} \geq x_D = \xi(\beta)$, which gives the second inequality above. On the other hand, using the fact that the expression $x_{D-1} + d\phi(d, \beta)e^{-\beta x_{D-1}}$ is convex in x_{D-1} , it can be lower bounded by, $\frac{1}{\beta}(1 + \ln(d\phi(d, \beta)\beta))$. Combining this and the above inequality, we get the following.

$$\begin{aligned}
SO(\mathbf{x}) &\geq x_1 + de^{-\beta x_1}x_2 + \dots + d^{D-2}e^{-\beta(\sum_{i=1}^{D-2}x_i)} \cdot \phi(d \cdot \phi(d, \beta), \beta) \\
&\quad \vdots \quad \text{repeating the steps above} \\
&\geq t_D(d, \beta), \quad \text{as defined in Equation (5.1.10)}.
\end{aligned}$$

Therefore the PoA $\leq \frac{d^D}{t_D(d, \beta)}$. ■

5.C Proofs for General Networks

Proof of Lemma 5.3

Proof: We follow the line of proof of Theorem 5.1. Each agent $i \in N$ is solving the following optimization problem.

$$\begin{aligned}
&\max_{x_i} \quad u_i(x_i, x_{-i}) \\
&\text{s.t.} \quad 0 \leq x_i \leq 1
\end{aligned} \tag{5.C.1}$$

This is a non-linear optimization problem. Hence we can write down the Lagrangian as follows.

$$\mathcal{L} = u_i(x_i, x_{-i}) + \lambda_i x_i + \gamma_i(1 - x_i), \quad \lambda_i, \gamma_i \geq 0.$$

The KKT conditions are necessary for this optimization problem (5.C.1), which are the following.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial x_i} &= 0, \\ \Rightarrow \frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) + \lambda_i - \gamma_i &= 0, \end{aligned} \tag{5.C.2}$$

$$\lambda_i x_i = 0, \quad \gamma_i(1 - x_i) = 0. \tag{5.C.3}$$

Case 1: $\lambda_i = 0, \gamma_i = 0$, then from Equation (5.C.2) we get,

$$\begin{aligned} \frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) &= 0 \\ \Rightarrow p_i f'(x_i) + \sum_{j \in E_i} h_{ij} \frac{\partial p_j}{\partial x_i} x_j &= 0 \\ \Rightarrow f'(x_i) = \sum_{j \in E_i} h_{ij} \left(-\frac{1}{p_i} \frac{\partial p_j}{\partial x_i} x_j \right) &= g_i(\mathbf{x}) \\ \Rightarrow x_i = \ell \circ g_i(\mathbf{x}), \text{ from the definition of } \ell & \end{aligned} \tag{5.C.4}$$

Case 2: $\lambda_i > 0, \gamma_i = 0$, then from Equation (5.C.3) we get $x_i = 0$, and from Equation (5.C.2),

$$\frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) < 0.$$

Carrying out the differentiation as in Equation (5.A.4), we get,

$$\begin{aligned} f'(x_i) < g_i(\mathbf{x}) &\Rightarrow 0 = x_i > \ell \circ g_i(\mathbf{x}), \text{ since } f \text{ is concave} \\ \Rightarrow x_i &= T \circ \ell \circ g_i(\mathbf{x}), \text{ where } T \text{ is the truncation function.} \end{aligned} \tag{5.C.5}$$

Case 3: $\lambda_i = 0, \gamma_i > 0$, then from Equation (5.C.3) we get $x_i = 1$, and from Equation (5.C.2),

$$\frac{\partial}{\partial x_i} u_i(x_i, x_{-i}) > 0.$$

Carrying out similar steps as before, we get,

$$\begin{aligned} f'(x_i) > g_i(\mathbf{x}) &\Rightarrow 1 = x_i < \ell \circ g_i(\mathbf{x}) \\ &\Rightarrow x_i = T \circ \ell \circ g_i(\mathbf{x}), \text{ where } T \text{ is the truncation function.} \end{aligned} \quad (5.C.6)$$

Case 4: $\lambda_i > 0, \gamma_i > 0$, this cannot happen since it will lead to a contradiction $0 = x_i = 1$. Therefore, combining Equations (5.C.4), (5.C.5), and (5.C.6), we get,

$$x_i^* = T \circ \ell \circ g_i(\mathbf{x}^*), \quad \forall i \in N.$$

Hence proved. ■

Chapter 6

Conclusions and Future Research

This thesis has addressed several game theoretic questions in crowdsourcing, and provided mechanism design solutions. In this chapter, we summarize the findings and contributions, and sketch a broad outline of a future research that these results can lead to.

6.1 Conclusions

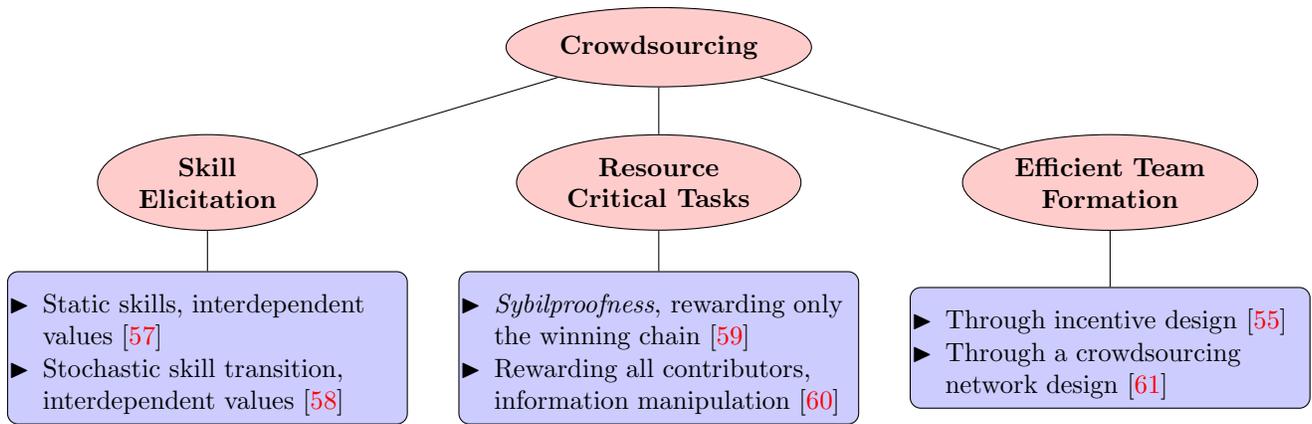


Figure 6.1: Summary of the findings of the thesis.

This thesis addressed three critical problems of strategic crowdsourcing. To visualize the problems addressed and the contributions of the thesis, let us look at Figure 6.1. In each paradigm, the major findings are as follows.

Eliciting Skills of the Strategic Workers

We addressed this problem in Chapter 3. We focused on the optimal crowd worker selection problem, where the goal of the crowdsourcer was to select an optimal subset from a known set of experts. The experts had private skills (hidden from the crowdsourcer) and the mechanism design problem was to ensure that they report their skills truthfully.

Task execution by a selected subset of the experts fall into the category of *interdependent values* since the benefit experienced by a participant depends on the outcome and the skills of *all* the other participants.

First, we addressed the mechanism design problem in a static setting, that is, the private skills are invariant over time. We provided an mechanism that made truthful reporting a strictly better strategy for all agents, and improved on the mechanism provided by Mezzetti [48].

We extended the result to dynamic setting where the experts' qualities vary stochastically over time according to a Markov process. This provided an efficient and truthful mechanism for the Markovian type transitions with interdependent values and improved upon the *dynamic pivot mechanism* [9]. We also showed that under a special setting, this mechanism incentivizes the participants to voluntarily participate in the game.

Resource Critical Task Execution via Crowdsourcing

Crowdsourcing is unique in the potential strategic manipulation that it offers to its participants. One such problem in resource critical crowdsourcing contests is that it can have *false-name attacks* or *sybil attacks*, in which the players can increase their payoffs by creating multiple false identities of themselves. We discuss this in Chapter 4.

We showed that in crowdsourcing contests like the DARPA red balloon challenge [21], which falls into the class of atomic tasks, it is impossible to satisfy sybilproofness, collapse-proofness, and a property that rewards all contributors with positive compensation. We introduced approximate versions of these properties and show that the mechanism space is non-empty. To the best of our knowledge, this is a first attempt to design *approximately sybilproof crowdsourcing mechanisms*. Under certain resource critical paradigms, some of those properties are more preferable than the others and we characterize the space of mechanisms that satisfy those properties under *cost-critical* and *time-critical* paradigms.

We noticed that in resource critical crowdsourcing contests, e.g., the red balloon challenge, a great deal of human effort can gets wasted, as people can potentially explore the same region already explored by someone else. A more time efficient and fair scheme could be to distribute the reward in proportion to the information contributed by each agent. We discussed briefly

about this point and provided a solution proposal using the tools from information theory and prediction market to aggregate information in a truthful, time efficient manner.

Efficient Team Formation from a Crowd

The other interesting question in crowdsourcing is to understand how highly productive teams emerge from a loosely structured population. A crowdsourcing network, once formed, is very similar to an organizational network, where the reason for success (failure) is often attributed to the (im)proper *management* of human resources. The goal of the designer in this context is to maximize the net productive output of the networked crowdsourcing system. We develop an understanding of the effort levels in influencer-influencee networks in Chapter 5.

We analyzed how individuals connected in a network trade-off between their production and communication effort given the network positions and the reward sharing scheme. We showed that under certain sufficient conditions, there exists a unique Nash equilibrium of this effort trade-off. We then provided a condition of achievability of optimal social output for stylized networks. Our results show that the equilibrium output may not always achieve the globally optimal. However, by choosing the right reward sharing scheme we can maximize the output and we provide a recipe of such reward sharing schemes.

Next, we analyzed how the network design can lead to interesting structures for maximizing the net productive output. We show that a network design that considers the strategic behavior of the agents performs better than the one that does not consider it.

6.2 Scope of Future Research

Each of these facets of the crowdsourcing problems gives rise to some more research directions that can be drawn from this thesis. Following are a few of them.

Learning the Skills of the Strategic Experts

So far we have assumed the skills to be perfectly known to the agents. However, in many practical scenarios, even the experts may not exactly *know* their qualities rather *experience* by doing it. It is, therefore, interesting to understand how *learning* the expertise of the participants can help design an efficient outsourcing process. The mechanism design question would be to make the outsourcing truthful for the experts even when the skills of the experts are unknown to both the crowdsourcer and the experts. There is a chance that a ‘learned good’ expert can exploit the learning of the designer by doing a sub-skillful job. An interesting question would be to understand how severe would its impact be, and if there is any approximately truthful mechanism in such settings.

Information Theoretic Crowdsourcing

Information theory provides us the analytical techniques to find the limits of information transmission. In the domain of crowdsourcing, we are actually interested in scavenging the true information from the strategic crowd. It is interesting to design (exact or approximate) sybil-proof mechanisms using the *information theoretic* tools, where the best response of an agent would be to report partial or whole information as soon as she receives it. This approach will lead us to designing mechanisms that blends the flavor of both competition and collaboration and help the designer crowdsource information in a time-efficient manner.

Network Stability Analysis in Crowdsourcing

Though we have answered the question of how efficient teams can be built by a central planner, either by the reward sharing design or the network design, an important question is whether these networks are structurally stable. This means that in the resulting network with the rewarding scheme, no agent should find deviating from the current configuration to a different one beneficial in terms of the expected payoff. If the three aspects: (a) reward share, (b) network structure, and (c) network stability, are jointly optimized, then it would serve to perform as the ideal solution for the efficient team formation problem in crowdsourcing.

In a nutshell

We have looked into an interesting set of mechanism design problems in the context of strategic crowdsourcing, which elicits the true skills of the crowd workers, limits the false identity problem, and yields a scheme of sharing the rewards for efficient output of the crowdsourcing network. The future of these problems are interesting both from the application and analytical points of view, and opens the door to a plethora of new research directions.

Bibliography

- [1] Alexa-Internet. Wikipedia.org site info, May 2013. URL <http://www.alexa.com/siteinfo/wikipedia.org>. 4
- [2] W. David Allen and Thomas W. Hall. Innovation, Managerial Effort, and Start-up Performance. *Journal of Entrepreneurial Finance, JEF*, 12(2):87–118, 2007. ISSN 1551-9570. URL <http://hdl.handle.net/10419/55930>. 101
- [3] Elliot Anshelevich and Martin Hoefer. Contribution Games in Networks. *Algorithmica*, 63:51–90, 2012. ISSN 0178-4617. 97
- [4] Associated-Press. Some U.S. hospitals outsourcing work, June 2004. URL <http://www.msnbc.msn.com/id/6621014>. 49
- [5] Susan Athey and Ilya Segal. An efficient dynamic mechanism. Working Paper, Stanford University, Earlier version circulated in 2004, 2007. 35, 50
- [6] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. On Bitcoin and Red Balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC 2012)*, pages 56–73. ACM, 2012. 73, 76, 77
- [7] Coralio Ballester, Antoni Calvó-Armengol, and Yves Zenou. Who’s Who in Networks. Wanted: The Key Player. *Econometrica*, 74(5):1403–1417, September 2006. 11, 96, 101, 133
- [8] BBC. Solving the Web’s Image Problem, May 2008. URL <http://news.bbc.co.uk/2/hi/technology/7395751.stm>. 4
- [9] Dirk Bergemann and Juuso Välimäki. The Dynamic Pivot Mechanism. *Econometrica*, 78: 771–789, 2010. 10, 35, 50, 51, 63, 65, 70, 144

- [10] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*, volume I. Athena Scientific, 1995. [50](#)
- [11] Dimitri P. Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, 1992. [118](#)
- [12] Ganesh D. Bhatt. Knowledge Management in Organizations: Examining the Interaction between Technologies, Techniques, and People. *Journal of Knowledge Management*, 5(1): 68–75, 2001. [95](#)
- [13] David Blackwell. Discounted Dynamic Programming. *The Annals of Mathematical Statistics*, 36(1):226–235, 1965. [55](#), [59](#)
- [14] Yann Bramoullé and Rachel Kranton. Public Goods in Networks. *Journal of Economic Theory*, 135(1):478 – 494, 2007. ISSN 0022-0531. [11](#), [96](#), [117](#), [133](#)
- [15] Elena Bruno. Smithsonian crowdsourcing since 1849!, April 2011. URL <http://siarchives.si.edu/blog/smithsonian-crowdsourcing-1849>. [3](#)
- [16] Ruggiero Cavallo. *Social Welfare Maximization in Dynamic Strategic Decision Problems*. PhD thesis, School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts, May 2008. [10](#)
- [17] Ruggiero Cavallo, David C. Parkes, and Satinder Singh. Optimal coordinated planning amongst self-interested agents with private state. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI 2006)*, pages 55–62, 2006. [35](#), [50](#), [51](#)
- [18] Ruggiero Cavallo, David C. Parkes, and Satinder Singh. Efficient Mechanisms with Dynamic Populations and Dynamic Types. Technical report, Harvard University, 2009. [35](#), [50](#)
- [19] Edward Clarke. Multipart Pricing of Public Goods. *Public Choice*, (8):19–33, 1971. [35](#), [50](#), [51](#)
- [20] Vincent Conitzer, Nicole Immorlica, Joshua Letchford, Kamesh Munagala, and Liad Wagman. False-name-proofness in Social Networks. In *Proceedings of Internet and Network Economics (WINE 2010)*, pages 209–221, 2010. [76](#)

- [21] DARPA. DARPA Network Challenge Project Report, 2010. URL <http://archive.darpa.mil/networkchallenge/>. 5, 6, 10, 72, 73, 76, 78, 144
- [22] DARPA. DARPA CLIQR Quest Challenge, 2012. URL <https://www.cliqrquest.com>. 6, 72
- [23] Erik D. Demaine, Mohammadtaghi Hajiaghayi, Hamid Mahini, and Morteza Zadimoghaddam. The Price of Anarchy in Cooperative Network Creation Games. *ACM SIGecom Exchanges*, 8(2):2, 2009. 97
- [24] Fabio Drucker and Lisa Fleischer. Simpler Sybil-Proof Mechanisms for Multi-Level Marketing. In *Proceedings of ACM Electronic Commerce (EC 2012)*, pages 441–458. ACM, 2012. 76, 77
- [25] Kathleen M. Eisenhardt. Agency-and institutional-theory explanations: the case of retail sales compensation. *Academy of Management Journal*, 31(3):488–511, 1988. 117
- [26] Yuval Emek, Ron Karidi, Moshe Tennenholtz, and Aviv Zohar. Mechanisms for Multi-Level Marketing. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC 2011)*, pages 209–218. ACM, 2011. 6, 72, 76, 78, 79
- [27] Enrique Estellés-Arolas and Fernando González-Ladrón-de Guevara. Towards an integrated crowdsourcing definition. *Journal of Information Science*, 38(2):189–200, 2012. 2
- [28] Andrea Galeotti, Sanjeev Goyal, Matthew O. Jackson, Fernando Vega-Redondo, and Leeat Yariv. Network Games. *The Review of Economic Studies*, 77(1):218–244, 2010. 96
- [29] Dinesh Garg and Y. Narahari. Price of anarchy of network routing games with incomplete information. *Internet and Network Economics (WINE 2005)*, pages 1066–1075, 2005. 97
- [30] Barry A. Gerhart, Harvey B. Minkoff, and Ray N. Olsen. Employee Compensation: Theory, Practice, and Evidence. *CAHRS Working Paper Series*, page 194, 1995. 95
- [31] Allan Gibbard. The Manipulation of Voting Schemes: A General Result. *Econometrica*, 41:587–601, 1973. 32
- [32] Melissa K. Gibson and Michael J. Papa. The Mud, the Blood, and the Beer Guys: Organizational Osmosis in Blue-Color Work Groups. *Journal of Applied Communication Research*, 28:68–88, 2000. 120
- [33] Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–31, July 1973. 35, 50, 51

- [34] Amar Gupta, Raj K Goyal, Keith A Joiner, and Sanjay Saini. Outsourcing in the healthcare industry: Information technology, intellectual property, and allied aspects. *Information Resources Management Journal (IRMJ)*, 21(1):1–26, 2008. 49
- [35] M. Harris and A. Raviv. Organization design. *Management Science*, 48(7):852–865, 2002. 97
- [36] Jeff Howe. The rise of crowdsourcing, June 2006. URL <http://www.wired.com/wired/archive/14.06/crowds.html>. 2
- [37] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Three Rivers Press, 2009. 72
- [38] Matthew O. Jackson. Networks and economic behavior. *Annual Review of Economics*, 1(1):489–511, 2009. 97
- [39] Philippe Jehiel and Benny Moldovanu. Efficient Design with Interdependent Valuations. *Econometrica*, (69):1237–1259, 2001. 9, 34, 39, 40, 49, 51, 55, 57
- [40] Philippe Jehiel, Moritz Meyer-ter Vehn, Benny Moldovanu, and William R Zame. The Limits of Ex-post Implementation. *Econometrica*, 74(3):585–610, 2006. 34, 39
- [41] Jon Kleinberg and Prabhakar Raghavan. Query Incentive Networks. In *Proceedings 46th IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, pages 132–141, 2005. 6, 72, 76, 78
- [42] Elias Koutsoupias and Christos H. Papadimitriou. Worst-case Equilibria. In *Proceedings of Symposium on Theoretical Aspects of Computer Science (STACS 99)*, pages 404–413. Springer, 1999. 95, 97, 105
- [43] Vijay Krishna. *Auction Theory*. Academic press, 2009. 34, 39
- [44] Karim R Lakhani, Lars Bo Jeppesen, Peter Andreas Lohse, and Jill A Panetta. *The Value of Openness in Scientific Problem Solving*. Division of Research, Harvard Business School, 2007. 5
- [45] Nate Lanxon. How the Oxford English Dictionary started out like Wikipedia, January 2011. URL <http://www.wired.co.uk/news/archive/2011-01/13/the-oxford-english-wiktionary>. 2

- [46] Andreu Mas-Colell, Michael D. Whinston, and Jerry R. Green. *Micreconomic Theory*. Oxford University Press, 1995. 17, 39
- [47] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013. 17, 18, 22
- [48] Claudio Mezzetti. Mechanism Design with Interdependent Valuations: Efficiency. *Econometrica*, 72(5):1617–1626, 2004. 9, 34, 35, 39, 40, 45, 47, 50, 51, 55, 57, 58, 70, 144
- [49] Claudio Mezzetti. Mechanism Design with Interdependent Valuations: Surplus Extraction. *Economic Theory*, 31:473–488, 2007. 50
- [50] Dilip Mookherjee. Incentives in Hierarchies. preliminary version, prepared for Handbook of Organizational Economics, 2010. URL <http://people.bu.edu/dilipm/wkpap/hndbkchap0810.pdf>. 95
- [51] Roger B. Myerson. *Game theory: analysis of conflict*. Harvard University Press, 1997. 17, 18, 22
- [52] Y. Narahari. *Game Theory and Mechanism Design*. IISc Press and World Scientific, forthcoming, 2013. 17, 18, 22, 30
- [53] Y. Narahari, Dinesh Garg, Ramasuri Narayanam, and Hastagiri Prakash. *Game Theoretic Problems in Network Economics and Mechanism Design Solutions*. Springer-Verlag, London, 2009. 17
- [54] John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951. 22
- [55] Swaprava Nath and Balakrishnan Narayanaswamy. Improving Productive Output in Influencer-Influencee Networks. *arXiv preprint arXiv:1302.3045*, 2013. 9, 94, 143
- [56] Swaprava Nath and Balakrishnan Narayanaswamy. Productive Output in Hierarchical Crowdsourcing. In *Proceedings of the Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, 2014. to appear. 94
- [57] Swaprava Nath and Onno Zoeter. A strict ex-post incentive compatible mechanism for interdependent valuations. *Economics Letters*, 121(2):321–325, 2013. 9, 39, 143
- [58] Swaprava Nath, Onno Zoeter, Y. Narahari, and Chris Dance. Dynamic Mechanism Design for Markets with Strategic Resources. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI 2011)*, pages 539–546. AUAI Press, 2011. 9, 49, 143

- [59] Swaprava Nath, Pankaj Dayama, Dinesh Garg, Y. Narahari, and James Zou. Mechanism Design for Time Critical and Cost Critical Task Execution via Crowdsourcing. In *Proceedings of Internet and Network Economics (WINE 2012)*, pages 212–226, 2012. [9](#), [71](#), [143](#)
- [60] Swaprava Nath, Pankaj Dayama, Rohith D. Vallam, Dinesh Garg, and Y. Narahari. Synergistic Crowdsourcing. Working Paper, 2012. [9](#), [91](#), [143](#)
- [61] Swaprava Nath, Balakrishnan Narayanaswamy, Kundan Kandhway, Bhushan Kotnis, and David C. Parkes. On Profit Sharing and Hierarchies in Organizations. Presented in Asian Meeting of the Econometric Society (AMES), December 2012. [9](#), [115](#), [143](#)
- [62] Swaprava Nath, Onno Zoeter, Y. Narahari, and Christopher R. Dance. Dynamic Mechanism Design with Interdependent Valuations. submitted to Review of Economic Design, June 2013. [49](#)
- [63] P. J. Nicholson. Occupational health services in the UK - challenges and opportunities. *Occupational Medicine*, 54:147152, 2004. doi: 10.1093/occmed/kqg125. [49](#)
- [64] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, 2007. [17](#)
- [65] Galen Pickard, Wei Pan, Iyad Rahwan, Manuel Cebrian, Riley Crane, Anmol Madan, and Alex Pentland. Time-Critical Social Mobilization. *Science*, 334(6055):509–512, October 2011. [6](#), [39](#), [72](#), [73](#), [76](#), [79](#), [95](#), [96](#)
- [66] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Interscience, 2005. [50](#), [54](#)
- [67] Roy Radner. Hierarchy: The Economics of Managing. *Journal of Economic Literature*, pages 1382–1415, 1992. [95](#)
- [68] Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, 2003. [95](#)
- [69] Kevin Roberts. *The Characterization of Implementable Choice Rules*, chapter Aggregation and Revelation of Preferences, pages 321–348. North Holland Publishing, 1979. [33](#)
- [70] Brian Rogers. A Strategic Theory of Network Status. Technical report, Working paper, Northwestern University, 2008. [96](#)

- [71] J. B. Rosen. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pages 520–534, 1965. [98](#), [101](#)
- [72] Tim Roughgarden. *Selfish routing and the price of anarchy*. MIT press, 2005. [97](#)
- [73] Mark Satterthwaite. Strategy-proofness and Arrow’s Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. *Journal of Economic Theory*, 10:187–217, 1975. [32](#)
- [74] Arunava Sen. Lecture Notes in Mechanism Design Theory. URL <http://www.isid.ac.in/~asen/scdoc/iisclectures.pdf>. [31](#)
- [75] Arunava Sen. Another Direct Proof of the Gibbard-Satterthwaite Theorem. *Economics Letters*, 70:381–385, 2001. [32](#)
- [76] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2010. [17](#), [18](#), [22](#)
- [77] Herbert Simon. Bounded Rationality and Organizational Learning. *Organization Science*, 2:125 – 134, 1991. [115](#)
- [78] Dale O. Stahl and Paul W. Wilson. Experimental Evidence on Players’ Models of Other Players. *Journal of Economic Behavior & Organization*, 25:309 – 327, 1994. [122](#)
- [79] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005. [72](#)
- [80] Noel M. Tichy, Michael L. Tushman, and Charles Fombrun. Social Network Analysis for Organizations. *Academy of Management Review*, pages 507–519, 1979. [95](#)
- [81] Marshall Van Alstyne. The state of network organization: a survey in three frameworks. *Journal of Organizational Computing and Electronic Commerce*, 7(2-3):83–151, 1997. [95](#)
- [82] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961. [35](#), [50](#), [51](#)
- [83] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. [18](#)
- [84] Duncan J. Watts and Jonah Peretti. *Viral marketing for the real world*. Harvard Business School Publishing, 2007. [95](#)

- [85] Wikimedia.org. Wikipedia statistics, article count (official), February 2013. URL <http://stats.wikimedia.org/EN/TablesArticlesTotal.htm>. 4
- [86] Wikipedia. Linux, June 2013. URL <http://en.wikipedia.org/wiki/Linux>. 5
- [87] Oliver E. Williamson. The economics of organization: The transaction cost approach. *American Journal of Sociology*, 87:548 – 577, 1981. 115
- [88] Simon Winchester. *The Surgeon of Crowthorne: A tale of murder, madness and the Oxford English Dictionary*. Penguin, 2002. 3
- [89] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues*. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1989. 118, 119
- [90] Yinyu Ye. A New Complexity Result on Solving the Markov Decision Problem. *Mathematics of Operations Research*, 30:733–749, August 2005. 65
- [91] Haifeng Yu, Michael Kaminsky, Phillip B. Gibbons, and Abraham Flaxman. SybilGuard: Defending against Sybil attacks via Social Networks. *SIGCOMM Computer Communication Review*, 36:267–278, August 2006. 76