The Dissertation Committee for Avhishek Chatterjee certifies that this is the approved version of the following dissertation:

Understanding Dynamics and Resource Allocation in Social Networks

Committee:

Sriram Vishwanath, Supervisor

François Baccelli

Sujay Sanghavi

Sanjay Shakkottai

Mihai Sirbu

Lav Varshney
Understanding Dynamics and Resource Allocation in Social Networks

by

Avhishek Chatterjee, B.E.; M.TECH.

DISSERTATION

Presented to the Faculty of the Graduate School of The University of Texas at Austin in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2015
Dedicated to my parents.
Acknowledgments

There are so many human beings who directly or indirectly, knowingly or unknowingly, with or without my knowledge have touched my life. It is impossible to acknowledge all of them by names. So, I start by saluting the benevolence and the auspiciousness pervading through humanity.

I am grateful to my Ph.D. advisor Prof. Sriram Vishwanath for being very understanding, cooperative and a big source of encouragement. He has been very kind and accommodative, always made me feel comfortable, and allowed me freedom at my work. He helped me to become an independent researcher with my own thoughts. I am thankful to him for creating the opportunities for me to have useful collaborations. I am also thankful to him for his patience, generosity and positive attitude during the difficult phases.

I was very fortunate to collaborate with Prof. François Baccelli, who has always been extremely kind and supportive. His immense technical insights, his humility in spite of his stature, his dedication, his research philosophy and his flexibility in coming down from his level to my level are great lessons for me. I am grateful to him for all his support, for sharing some of his technical treasures, and for the wise advices about research in general.

I thank Prof. Sujay Sanghavi for his fine lectures on high-dimensional learning and for his mentoring during our research collaboration at the begin-
ning of my time at the university. His uncanny ability to always ask the right questions and to hit the crux of every problem inspires awe in me, and at the same time it motivates me to do my best within my very limited ability.

Prof. Sanjay Shakkottai has always showed kindness and eagerness whenever I approached him with any technical or general question. I am grateful to him for that and all the valuable advices coming from a researcher of his ability, for his excellent lectures on concentration inequalities, and for the great learning experience that I had during our research collaboration.

I thoroughly enjoyed the lectures on the theory of probability by Prof. Mihai Sirbu. I thank him for the beautiful lectures, for always being very nice to me and for agreeing to be on my dissertation committee.

Prof. Lav Varshney has always been extremely kind, patient, and enthusiastic with his help whenever I approached him with any academic or non-academic issue. His versatility, his keen eye for interesting problems, and his passion for research offer a great opportunity to learn. I am fortunate to have such a researcher as a collaborator, and I am grateful to him for all the support and for introducing me to the field of crowdsourcing.

I thank Prof. Gustavo de Veciana for his excellent lectures on queueing networks. Prof. Lei Ying has been very helpful during our research collaboration towards the beginning of my Ph.D, for which I am thankful to him. I am grateful to my collaborators Siddhartha Banerjee, Michael Borokhovich, Ankit Singh Rawat, and Prof. Anand Sarwate for all that I have learnt from
them.

I owe to my excellent lab-mates for the great experience that I had during my time at WNCG. I have enjoyed all our interactions and I would like to thank all of them for that. I would like to extend my special thanks to Kumar Appaiah, Megasthenis Asteris, Siddhartha Banerjee, Soumya Basu, Srinadh Bhojanapalli, Abhik Kumar Das, Deepjyoti Deka, Harpreet Singh Dhillon Ethan Elenberg, Javad Ghaderi, Joyce Ho, Hardik Jain, Ali Jalali, Jubin Jose, Vinay Joseph, Muryong Kim, Murat Kocaoglu, Subhashini Krishnasamy, Mandar Kulkarni, Xingqin Lin, Erik Lindgren, Derya Malak, Ioannis Mitliagkas, Sharayu Moharir, Praneeth Netrapalli, Duhyong Park, Dimitris Papailiopoulos, Zrinka Puljiz, Ankit Singh Rawat, Avik Ray, Aneesh Reddy, Abishek Sankararaman, Rajat Sen, Karthikeyan Shanmugam, Hongbo Si, Virag Shah, Vatsal Shah, Sarabjot Singh, Rajiv Soundararajan, and Louie Wu.

I sincerely thank Lauren Bringle, Melanie Gullick, Alfred Koh, Karen Little, Lisa McDow, Melody Singleton, and Diana Vega for being very friendly and always helping me out with administrative work.

My days at Austin could never be so enjoyable without the presence of my old and sincere friend Subhendu Roy, and his wife and also an old friend of mine, Urmimala Roy. I very much enjoyed the frequent companies of Abhik Bhattacharya, Proma Bhattacharya, Rudra Chatterjee, and Arindam Sanyal. Ayan Acharya, Somsubahra Barik, Debarati Kundu, and Avik Ray provided some enriching experiences.
I am grateful to my family members: maternal uncle, paternal uncles, aunts, cousins, nephews and nieces for their constant support. I would like to pay my gratitude to Dr. Subhankar Ghosh and Dr. Asit Chakraborty for always giving me the right advice and encouragement. I thank my friend Debarghya Chakraborty for always being a well wisher. I am also thankful to many of my family friends for their constant presence by my side: Dr. and Mrs. A. Chakraborty, Mr. and Mrs. S. Chakrabarti, Dr. and Mrs. S. Ghosh, Dr. and Mrs. R. K. Mondal, Dr. and Mrs. S. K. Roy, and Drs. D. Panigrahi and S. Roy.

Whatever good I have in me I owe that completely to my parents. Any attempt to express my gratitude to them will surely fall far shorter of what is intended. So, I pray that, like the only possible expression of gratitude by a cloud to the ocean is the life of the cloud itself, may my life be a fitting carrier of the goodness that my parents instill in me.
Widespread popularity of various online social networks has attracted significant attention of the research community. Research interest in social networks are broadly divided into two categories: understanding the social or human network dynamics and harnessing the social network dynamics to gain economic, business or political advantage using minimal resource. These two research directions fuel each other. Better understanding offers better resource utilization/allocation in harnessing the network and the need for better resource utilization/allocation drives the fundamental research in understanding human networks. This thesis considers important problems in both directions as well as at their intersection.

We first study opinion dynamics in social networks. We propose a new stochastic dynamics which generalizes two widely used and complementary models of opinion dynamics, graph-based linear dynamics and bounded confidence dynamics into a single stochastic dynamics. We analytically study the
conditions under which such dynamics result in reconciliation or some sort of consensus. Our findings relate well to observed behaviors of societies.

The next problem that we consider is related to designing personalized/targeted advertisements or campaigns for social network users. Currently viral marketing or campaigning rely only on the structure of the friendship graph. In reality friends may have different opinions on different topics or issues. It is understood that if opinions regarding a topic were known one could design better targeted campaigns. We propose algorithms which can infer opinions of people by observing their interactions regarding a topic or an issue. As data gathering and computation requires resources, our algorithm is designed to work with fewer such resources for a broad class of social networks and interaction patterns.

A recent trend among different businesses is to work with social software providers (e.g., Lithium, Salesforce.com) to engage consumers online and often involve the online crowd directly in developing and running business ideas. This trend, popularly known as crowdsourcing uses human cloud to do jobs that cannot be done by machines. Crowdsourcing has been successfully used to do simple human tasks (Amazon Mechanical Turk), scientific research (fold.it), freelance software development(oDesk) as well as in impacting the lives of people in poverty (Samasource). Many big business houses use crowdsourcing, e.g., Microsoft, Samsung, Intel etc., IBM harness its employee pool using internal crowdsourcing. As employing humans (a.k.a. agents) for jobs, and especially for skilled jobs (like software development, scientific studies) is
costly, an efficient job to agent allocation is key to the success of crowdsourcing. Motivated by this, in the last part of the thesis we study efficient resource allocation in skill-based crowdsourcing platforms.
# Table of Contents

Acknowledgments v

Abstract ix

Chapter 1. Introduction 1

Chapter 2. Stochastic Bounded Confidence Opinion Dynamics 6
  2.1 Introduction .................................................. 6
  2.1.1 Background ................................................. 8
  2.1.2 Motivation .................................................. 10
  2.1.3 Our Contribution ......................................... 12
  2.2 Dynamics ..................................................... 13
  2.3 Dynamics Stability .......................................... 17
    2.3.1 Structure of the section ............................... 19
    2.3.2 The Two Body Problem ................................. 20
    2.3.3 Complete Interaction Case ............................ 21
    2.3.4 Interaction Graph ...................................... 23
    2.3.5 The Case with Bias .................................... 25
    2.3.6 On Tail Behavior ...................................... 27
  2.4 Generalization of the Results ............................ 28
    2.4.1 Endogenous process ................................... 28
    2.4.2 Influence function .................................... 28
    2.4.3 Interaction process ................................... 29
  2.5 Conclusion .................................................. 29
5.3.2 Centralized Allocation ........................................ 98
5.3.3 Decentralized Allocation ..................................... 102
5.4 Conclusion .......................................................... 104

Chapter 6. Conclusion ................................................. 105

Appendices ............................................................... 107

Appendix A. .............................................................. 108
A.1 Proofs ................................................................. 108
  A.1.1 Proof of Theorem 2 ........................................... 109
  A.1.2 Proof of Theorem 5 ........................................... 117
  A.1.3 Proof of Theorem 6 ........................................... 124
  A.1.4 Proof of Theorem 8 ........................................... 127
  A.1.5 Proof of Theorem 7 ........................................... 129
  A.1.6 Proof of Theorem 9 ........................................... 132
  A.1.7 Proof of Theorem 10 ......................................... 133
  A.1.8 Proof of Theorem 11 ......................................... 135
  A.1.9 Proof of Theorem 12 ......................................... 136
  A.1.10 Proof of Theorem 13 ........................................ 137
  A.1.11 Proof of Theorem 14 ........................................ 142

Appendix B. .............................................................. 148
B.1 Proofs ................................................................. 148
  B.1.1 Proof of Theorem 15 ......................................... 148
  B.1.2 Proof of Theorem 16 ......................................... 150
  B.1.3 Proof of Theorem 18 ......................................... 163
  B.1.4 Proof of Theorem 19 ......................................... 163
  B.1.5 Proof of Theorem 21 ......................................... 165
  B.1.6 Proof of Theorem 22 ......................................... 166
Appendix C. 167
C.1 Proofs .................................................. 167
   C.1.1 Proof of Theorem 27 ............................... 167
   C.1.2 Proof of Theorem 28 ............................... 169
   C.1.3 Proof of Proposition 29 ......................... 170
   C.1.4 Proof of Theorem 30 ............................... 171
   C.1.5 Proof of Theorem 33 ............................... 174
   C.1.6 Proof of Proposition 34 ......................... 178
   C.1.7 Proof of Theorem 35 ............................... 180
   C.1.8 Proof of Theorem 36 ............................... 187
   C.1.9 Proof of Theorem 37 ............................... 188
   C.1.10 Proof of Theorem 38 ............................ 190
   C.1.11 Proof of Theorem 39 ............................ 190
   C.1.12 Proof of Theorem 40 ............................ 193
C.2 Computation ........................................... 197
   C.2.1 Computation for Centralized Allocation ....... 197

Appendix D. 203
D.1 Proofs .................................................. 203
   D.1.1 Proof of Theorem 42 ............................... 203
   D.1.2 Proof of Theorem 43 ............................... 204
   D.1.3 Proof of Theorem 44 ............................... 204
   D.1.4 Proof of Theorem 45 ............................... 214
   D.1.5 Proof of Theorem 46 ............................... 215

Bibliography 224

Vita 238
Chapter 1

Introduction

In recent years, with the advent of online social networks, the Internet has become an extremely popular medium for social interactions. Online social networks like Facebook, Twitter, Quora, Reddit, etc. are among today’s most popular forums. Undoubtedly, these platforms have shaped the mode of modern social interactions and have greatly enhanced the connectivity and the strength of human network. Beyond just facilitating social interactions, online social networks have significantly impacted socio-economics. Online social networks have revolutionized the way of product advertising by following a more organized and targeted approach that harness the social connections and the observed online behavior of the individuals. Politics have also experienced a similar impact, today online social networks are important platforms for political campaigning. These new developments have necessitated thorough study of social networks.

Traditionally, human social interactions have been studied by social scientists. Research in this field have mostly been of qualitative nature, primarily because of the difficulty of obtaining real life experimental data of sufficient scale for quantitative studies. With the advent of online social networks we
have moved from a data-scarce regime to the regime of data-deluge. Working with these huge data sets of social interactions, designing reasonable and tractable analytical models, understanding the nature of social interactions based on them, and designing algorithms/strategy to harness the social networks require a wide range of expertise. As a natural consequence, social networks have drawn significant research attention from various disciplines, to name a few: computer science, economics, electrical engineering, mathematics, and physics.

Modern day research interest in social networks can be divided into two broad categories: understanding the social dynamics and harnessing the social network to attain social, political, economic and business goals using minimal resources.

Understanding social dynamics involves proposing reasonable and analytically tractable models for various phenomena in social networks. This line of research is often driven by the quest for developing the science of networks in general and human networks in particular. The end results of this line of research are good models and quantitative insights drawn from the analytical study of the models. The quantitative findings of these analytical studies are in turn used for designing strategies to harness social networks.

Towards harnessing social networks, based on the quantitative results and the models, strategies are proposed for maximizing certain quantitative goals, e.g., popularizing a product/business, increasing social awareness about important issues, political campaigning, crowdfunding etc. Various business
entities, governments bodies, political parties, charitable organizing rely heavily on online social networks like Facebook and Twitter to meet their goals. Also, new kinds of businesses have emerged: today we have specialized technical teams and companies to help various entities to utilize social networks for their purpose, e.g., Lithium, Salesforce.com etc. As every business operates on a budget, there is a need to judiciously use its resource, money as well as human resource. So, in harnessing the social networks, optimal resource allocation is extremely important.

Apart from spreading popularity and increasing customer/follower base, recently social networks have been used to revolutionize business operations by involving online crowd actively. For example, Lithium facilitates business to engage with customers as well as to use the online crowd to develop and run new ideas. Use of online crowd for doing general and specialized work, which is popularly known as crowdsourcing is an increasing trend among various enterprises. There are many crowdsourcing platforms to facilitate this, for example Amazon Mechanical Turk for simple short-term jobs and oDesk for more specialized and longer term projects. Many business organizations have their own platforms for engaging crowd, and some big enterprises even harness their huge internal worker pool for crowdsourcing specialized jobs. Employing crowd workers (a.k.a agents) costs money, and hence, it is important to ensure that the jobs go to respective suitable agents. Thus, optimal resource allocation is a necessity for efficient crowdsourcing.

This thesis is concerned with understanding social network dynamics
and efficient resource allocation in social systems [1–3]. In Chapter 2 of the thesis, we study opinion dynamics in social networks to understand how opinions evolve in a social system. Graph-based linear dynamics [4,5] and bounded confidence dynamics [6–8] are the two most studied classes of models of opinion dynamics and they are complementary in nature. Graph-based models capture the effect of the underlying social friendship structure (or the social graph or the friendship graph, e.g., Facebook graph) on opinion evolution, but fails to capture a well known social phenomenon, ‘like minds share more’. On the other hand, bounded confidence models capture this phenomenon by introducing ‘opinion dependent friendship’, but largely ignores the social graph. There have been some work towards this [8], but the models are partial and lack analytical insights. We propose a stochastic generalization of the bounded confidence dynamics that captures real-life interactions more closely and combines bounded confidence and graph-based dynamics. From the models we derive analytical insights that pertain to observed behaviors of real societies.

In Chapter 3 we consider the problem of harnessing social networks. Advertisements and political campaigns on social networks mainly harness the structure of social network (or friendship graph) to spread word-of-mouth. It is understandable that these campaigns would be more effective if they are made more targeted, i.e., they are shaped for a person as per his/her mindset. As mindset or opinions of people are often private and hence unknown, we consider the problem of estimating the opinions from social interactions. Towards this we exploit the fact that ‘like minds share more’ and propose efficient
(requiring fewer resource in terms of compute cycles and data) algorithms to infer opinions from social interactions.

Chapters 4 and 5 are about efficient resource allocation in crowdsourcing systems. On crowdsourcing platforms that involve specialized work and relatively longer time projects (like oDesk, and unlike microtasks in Amazon Mechanical Turk), sometimes referred to as freelance markets, jobs often require multiple skills and collaboration among many agents that have varied skill sets. On these platforms it is important to allocate appropriate agents (or resources) to the jobs to ensure best utilization of their working hours and skill sets. We characterize the limits of job-to-agent allocations in such systems and propose algorithms to achieve that.
Chapter 2

Stochastic Bounded Confidence Opinion Dynamics

2.1 Introduction

Understanding the theoretical underpinnings of opinion dynamics is an important component of developing a complete picture of the manners in which social systems evolve and behave. Although an extremely interdisciplinary and complex problem, the modeling of opinion dynamics can be broken down to some essential components. Indeed, a variety of tools, ranging from combinatorics to game theory have already been applied to this domain in order to distill and model the main components and interactions within social systems, as detailed below.

A majority of existing work in this domain is initialized with a setup where a (static) social graph determines the set of all allowable forms of interactions between agents in the system. Subsequently, the dynamics proceeds based on interactions among neighbors in this graph. Such models are a great starting point for multiple real-world social interactions both online and offline, and this body of work forms one of the directions that we build on in this chapter.
In addition to these, there is a growing body of work on settings where no such graph dictates interactions among agents, and/or where the graph is not static. Examples of interactions to be modeled include online interactions in forums such as Reddit, Quora where groups are self-selective in involvement, bringing like-minded people together to interact with one another. In the offline world, conferences, meetups serve a similar purpose of bringing people together, and the relationship structure among agents evolves due to this self-selecting interactions among agents. Along a similar vein, Twitter represents a dynamic graph where agents “follow” or “unfollow” one another based on proximity of opinions, much more in line with forums and gatherings than static graph models.

Bounded confidence, the other family of well known models for opinion dynamics, models interactions in systems where agents interact with one another based on the proximity of their opinions. In the bounded confidence opinion dynamics, agents interact over time in a pairwise manner. Each interaction triggers an opinion update which is again a (possibly linear) function of the opinions of the agents involved; however the interaction is only effective if the opinion difference is within a threshold “distance”. Thus, this model allows for opinion-dependent social exchanges and the incorporation of each agent’s internal views.

In this work, we present a stochastic model featuring pairwise interactions among agents, generalizing the mechanism of interactions currently studied in the context of bounded confidence opinion dynamics models and
bringing it closer, in spirit, to existing graph-based interaction models. In addition to opinion-dependent social exchanges, our model incorporates the inherent stochasticity in interactions, imperfect exchange of opinions as well as self-beliefs, which capture the endogenous evolution of opinions innate to each agent. In addition, this model can be combined as we shall see to graph-based and bounded confidence dynamics.

We characterize the conditions under which these dynamics are stable, in a mathematical sense, and discuss the implications of this result from a sociological perspective. Overall, our work aims to build a stronger connection between the two bodies of work on graph-based and bounded confidence based dynamics, in addition to providing a stochastic generalization of both.

2.1.1 Background

Social networks and systems have drawn interest from various fields that are not traditionally related to social or political sciences, particularly so after the advent of web and online social networks. Beyond social and political sciences; economics, physics, computer science and electrical engineering have seen significant research drives for understanding opinion dynamics [6–15].

Research efforts in this multidisciplinary domain aim at (1) understanding the structure of social connections [13, 14], (2) studying the spread and evolution of thoughts or opinions [11, 16], as well as (3) designing strategies to maximize (or minimize) spread [12, 17], among other goals.

Typically, across multiple domains of research in opinion dynamics,
the opinions of agents are considered to be real numbers that evolve over time based on the interactions among these agents. As mentioned before, research in opinion dynamics can be broadly classified into two basic themes: linear dynamics on a graph, and bounded confidence opinion dynamics. Dynamics on graphs represent the evolution of opinions as a discrete-time linear system whose properties, like equilibrium, convergence etc., are studied under a variety of conditions, e.g., [18–20]. This body of literature forms an important stepping stone to building our understanding of interactions in social systems; however, it typically ignores the fact that the probability of interaction among agents is often a decreasing function of the difference of their opinions.

The main feature of bounded confidence opinion dynamics [6,7] is their opinion-dependent social interactions. In this model, two agents interact and influence one another only when the difference in their opinions is below a threshold. The bounded confidence model proposed by Deffuant et al. [6] is a discrete-time dynamics with pairwise interactions. At each time $t \in \mathbb{Z}_+$, two agents $i$ and $j$ interact and update their opinions based on each other’s opinions only if their opinion difference is less than a threshold $\eta$. Formally, for opinions $x_i(t), x_j(t) \in \mathbb{R}$ at time $t$, the agents update their opinions to $x_i(t+1), x_j(t+1) \in \mathbb{R}$ as follows:

$$x_i(t+1) = x_i(t) + \gamma(x_j(t) - x_i(t))\mathbf{1}(|x_i(t) - x_j(t)| \leq \eta)$$
$$x_j(t+1) = x_j(t) + \gamma(x_i(t) - x_j(t))\mathbf{1}(|x_i(t) - x_j(t)| \leq \eta),$$

where $\gamma \in (0, \frac{1}{2}]$. Such opinion dynamics do not necessarily result in consensus,
with agents potentially forming multiple isolated clusters [15, 21]. There has been work focusing on analyzing the convergence time of these dynamics [15, 22] as well as attempts to numerically study variations, including different initial conditions, multi-dimensional opinions, impact of noise, heterogeneous thresholds and underlying network structure [21, 23–26]. Reference [8] presents a survey of this line of research.

2.1.2 Motivation

By incorporating opinion-dependent interactions or exchanges, bounded confidence dynamics take a good first step towards modeling opinion evolution in social systems, but this class of models and their existing variants do not yet capture some of the inherent characteristics of opinion dynamics in social systems. We discuss the components we find to be missing in existing models, and subsequently describe ways in which our model incorporates them.

First, existing models assume a deterministic and thresholded behavior of agents in considering opinions of other agents. On contrary, in real life, social interactions possess a fair degree of inherent randomness, and lack sharp thresholds in terms of interactions and overall behavior.

Second, in most bodies of existing work, it is often assumed that each agent has full knowledge of the opinions of the agents it interacts with. However, in practice, opinions may not be known exactly, and there may be an associated error in estimation. This estimation error can substantially impact the process of incorporation of other agents’ opinions in both space and time.
Third, error/noise in estimating the opinion of an agent can also directly impact the actual opinion update process.

Fourth, each agent may possess its own innate self-beliefs that influence its opinion, in addition to external interactions with other agents within the social system.

Fifth, not all agents that share similar opinions may interact with one other, as they may not gain the opportunity to do so. In addition, the strengths of friendships, and therefore, the extent of interaction between all agents may not be the same.

Some of these issues have been studied in part, e.g., work on noisy bounded confidence with heterogeneous thresholds and bounded confidence on social graph [21, 24–26]. However, existing results are largely limited to numerical studies of variations of the bounded confidence dynamics; and the few theoretically structured approaches that exist along this line do not encompass all the aspects discussed above. The stationary distribution of a dynamics with stochastic interactions and error-free updates has been studied in [27] when the number of agents tends to infinity. [28] studied consensus with noisy communication as a control problem. [29] studied the limits of a bounded confidence dynamics (time-invariant $\eta$) with stochastic approximation like discounted updates in the presence of noise. Our model aims to capture the above mentioned five missing elements into a stochastic framework generalizing bounded-confidence opinion dynamics, as detailed next.
2.1.3 Our Contribution

In this work, we present a model for opinion dynamics that addresses the characteristics of social systems listed in the previous section. Towards this goal, by allowing opinion-dependent probabilistic exchange of opinions among agents, it incorporates the inherent stochasticity as desired by the first and second characteristics above. In addition, the impact of erroneous opinion estimates on updates and the impact of self-belief in shaping an agent’s opinion, as discussed in the third and fourth characteristics, are captured by the introduction of terms that we call endogenous processes. Finally, an interaction process and pairwise strengths of influence cover the notion of social proximities and their impact on opinion dependent social exchanges.

In this chapter, we present an analytic understanding of our model. In this process, we aim to translate concepts of importance in socio-political literature to relate them with mathematical conditions and constraints on our dynamics.

The rest of the chapter is organized as follows. In Section 2.2, we present a mathematical description of the model. In Section 2.3 we discuss the main theoretical results (proofs are presented in Appendix A). We conclude with Section 5.4.
2.2 Dynamics

We consider a discrete time opinion dynamics of \( n \) agents, where agents update their opinions based on pairwise interactions. Time is denoted by \( t \in \mathbb{Z}_+ \). The opinion of agent \( i \) at time \( t \) is denoted by \( x_i(t) \in \mathbb{R} \). Interactions among agents form a random process \( I(t) \) that takes values in the set of subsets of cardinality 2 of the set of agents: \( \{\{i, j\} : 1 \leq i \neq j \leq n\} \). \( I(t) = \{i, j\} \) indicates that at time \( t \), agents \( i \) and \( j \) interact. At any given time, there is hence a single pairwise interaction.

Given \( I(t) = \{i, j\} \), agents \( i \) and \( j \) update their opinions according to a rule that depends on the nature of their interaction, and on their respective opinions. The nature of the interaction is characterized by the \( \{0, 1\} \) random variables \( U_{i,j}^t \) and \( U_{j,i}^t \): \( U_{i,j}^t = 1 \) means that \( i \) considers the opinion of \( j \) to update its opinion at time \( t \), with a symmetrical interpretation for \( U_{j,i}^t \). If \( U_{i,j}^t = 1 \), then \( i \) averages its opinion with that of \( j \) with a weight \( \gamma_{i,j} \in (0, \frac{1}{2}] \).

In addition, at any time \( t \), irrespective of the nature of the interaction, the opinion of agent \( i \) is perturbed by \( n_i(t) \), due to some endogenous belief or bias, or due to some error in estimating others’ opinions.

Thus given \( I(t) = \{i, j\} \), the opinions \( x(t) = \{x_i(t)\}_i \) evolve as follows:

\[
\begin{align*}
x_i(t + 1) &= x_i(t) + \gamma_{i,j} (x_j(t) - x_i(t)) + n_i(t) \text{ if } U_{i,j}^t = 1 \\
&= x_i(t) + n_i(t) \text{ else} \\
x_j(t + 1) &= x_j(t) - \gamma_{j,i} (x_j(t) - x_i(t)) + n_i(t) \text{ if } U_{j,i}^t = 1 \\
&= x_j(t) + n_j(t) \text{ else}
\end{align*}
\]
\[ x_k(t + 1) = x_k(t) + n_k(t) \text{ for } k \notin \{i, j\}. \]

In this work, for analytical purposes, we restrict ourselves to the following stochastic assumptions.

- The *endogenous* process \( n(t) = (n_1(t), n_2(t), \ldots) \) is i.i.d. across time. This process may be centered (this will be referred to as an *endogenous noise*, modelling centered self beliefs and estimation error) or not (referred to as *endogenous bias*, modelling self beliefs with drifts); the second moment of \( n(t) \) is finite. \( n_i(t) \) are independent across agents.

- The interaction process \( I(t) \) is i.i.d. across time; the interaction process \( I(t) \) may come from the social structure; for example \( P(I(t) = \{i, j\}) \) is non-zero only if \( i \) and \( j \) have a connection in the social structure (an edge in some social graph); there is a chance of no interaction, i.e., \( P(I(t) = \emptyset) > 0 \).

- The endogenous processes and interaction processes are independent.

- The processes \( \{U_{i,j}^t\} \) are independent of the past given \( x(t) \). We furthermore assume that the probability that \( U_{i,j}^t \) takes value 1 given \( x(t) \) is a function \( f_{i,j}(|x_i(t) - x_j(t)|) \) which is non-increasing in its argument. This means that given \( i \) and \( j \) interact at time \( t \), whether \( i \) accepts \( j \)'s opinion or in other words, \( j \) influences \( i \)'s opinion depends only on the two agents and more precisely on their opinion difference. This allows us to account for the fact that, given a potential pairwise interaction, the opinions have to
be close enough to be accepted for averaging to take place. For $t$ fixed, the correlation between the $U_{i,j}^t$ variables is arbitrary.

In general, the above dynamics that will be referred to as the \textit{stochastic bounded confidence opinion dynamics}, is a model of dynamics with opinions in $\mathbb{R}$. In this work, for simplicity, we restrict ourselves to dynamics with opinions in $\mathbb{Z}$. The opinion dynamics in $\mathbb{Z}$ is similar to that above with slight modifications:

\begin{align*}
x_i(t+1) &= x_i(t) + \lfloor \gamma_{i,j} (x_j(t) - x_i(t)) \rfloor + n_i(t) \text{ if } U_{i,j}^t = 1 \\
&= x_i(t) + n_i(t) \text{ else} \tag{2.1} \\
x_j(t+1) &= x_j(t) - \lceil \gamma_{j,i} (x_j(t) - x_i(t)) \rceil + n_i(t) \text{ if } U_{j,i}^t = 1 \\
&= x_j(t) + n_j(t) \text{ else} \tag{2.2} \\
x_k(t+1) &= x_k(t) + n_k(t) \text{ for } k \notin \{i,j\}.
\end{align*}

Here $\lfloor \cdot \rfloor$ is a rounding mechanism e.g., flooring, randomized rounding etc. The endogenous processes $\{n_i(t)\}$ are integers as well. $x(t)$ is a sample path realization of the opinion dynamics process. The process of opinions is denoted by $X(t) = \{X_i(t) : t \in \mathbb{Z}_+, 1 \leq i \leq n\}$. We use the convention of denoting random variables or processes by capital letters and their realizations by small letters.

We now introduce some terminology which is used later.

\textbf{Influence} When $U_{i,j}^t = 1$ and $I(t) = \{i,j\}$, we say $i$ is \textit{influenced} by $j$ or $i$ \textit{listens} to $j$. 

\addcontentsline{toc}{section}{} 
\addcontentsline{toc}{subsection}{}
**Symmetry** If \( f_{i,j} = f_{j,i} \), we say that \( i \) and \( j \) have *symmetric* influence. In general, influence is *asymmetric*, i.e. \( f_{i,j} \) and \( f_{j,i} \) may be different. If \( f_{i,j} = f_{k,j} \) for all \( i, k \), i.e., whether \( j \) influences \( i \) or not depends only on \( j \), we say \( j \) is an *isotropic* influencer.

Another special case is that where for all pairs \( i \neq j \), \( U_{i,j}^t = U_{j,i}^t \) a.s. This will be referred to as *mutual influence*. Mutual influence is stronger than symmetric influence.

**Tails** Similarly one can classify interactions based on how the function \( f_{i,j}(r) \) depends on \( r \). If \( f_{i,j} \) is such that for all sufficiently large \( r \), \( f_{i,j}(r) \geq \frac{c}{r^\alpha} \) for some \( \alpha < 2 \) and \( c > 0 \) and if in addition \( P(I(t) = \{i,j\}) > 0 \), then we say \( j \) strongly influences \( i \). Further, if \( f_{i,j} \) is such that \( f_{i,j}(r) \geq \frac{c}{r^\alpha} \) for some \( \alpha < 1 \) and \( c > 0 \) for all sufficiently large \( r \), and also \( P(I(t) = \{i,j\}) > 0 \), then we say \( j \) very strongly influences \( i \). On the other hand, if \( f_{i,j}(r) \leq \frac{c}{r^\alpha} \) for some \( \alpha > 2 \) and \( c > 0 \) then we say \( j \) moderately influences \( i \). We say \( j \) has no influence on \( i \) if \( f_{i,j} \) is identically 0.

Using the classical asymptotic notation\(^1\), given \( P(I(t) = \{i,j\}) > 0 \), the influence of \( j \) on \( i \) is strong if \( f_{i,j}(r) = \Omega \left( \frac{1}{r^\alpha} \right) \) for \( \alpha < 2 \), very strong if \( f_{i,j}(r) = \Omega \left( \frac{1}{r^\alpha} \right) \) for \( \alpha < 1 \) and moderate if \( f_{i,j}(r) = O \left( \frac{1}{r^\alpha} \right) \) for \( \alpha > 2 \). The special cases were \( f_{i,j}(r) \) has a negative exponential tail or a bounded support belong to the last class. This classification is important for the characterization

---

\(^1\)For \( f, g : \mathbb{R}^+ \to \mathbb{R}^+ \) we say \( f(x) = \Omega(g(x)) \) if \( \exists c > 0 \) s.t. \( f(x) \geq cg(x) \forall x \). We use the \( f = O(g) \) notation to mean \( \exists c' > 0 \) s.t. \( f(x) \leq c'g(x) \forall x \).
of the dynamics in Section 2.3.

\section*{2.3 Dynamics Stability}

In a society or a system of interacting (social) agents, convergence of opinions to a consensus point is often desirable, as also sought after in the literature. Unfortunately, in the presence of agents that are selective in incorporating others’ opinions into their own, this is rarely possible \cite{6, 7, 15}. Moreover, when the interactions as well as incorporation of opinions are probabilistic and, additionally, when the agents are driven by their own beliefs, consensus is almost impossible \cite{18, 30}. In case of dynamics without consensus, differences persist which may remain finite under favorable circumstances. If the dynamics is stochastic, as in our case, an equilibrium distribution of differences may be reached if there is a steady state. Note that in case of a stochastic dynamics with agents driven by their own beliefs, it is possible that no steady state be reached, and that agents diverge in opinions.

Note that in this context, even if opinions move to $\infty$ while remaining close to each other, the society is defined to be stable. Thus unlike other dynamics where stability of the process is related to the finiteness of the process variables, here, the finiteness of the opinion differences is more important than that of opinions themselves \cite{31–33}.

The probability laws of the interaction process $I(t)$ together with the $\{f_{ij}(t)\}$ functions define a class of opinion dynamics when varying the laws of the endogeneous process $n(t)$ within framework defined above (finite mean and
variance i.i.d. vectors). Since the stability of this class of opinion dynamics pertains to the finiteness of opinion differences, the following definition is in order.

**Definition 1.** The class of opinion dynamics $X(t)$ defined by the law of $I(t)$ and the $\{f_{i,j}\}$ functions is **stable** if for all finite initial opinions and all i.i.d. endogenous processes with finite mean and second moment, the process of opinion differences $Y(t) = \{Y_{ij}(t) = X_j(t) - X_i(t) : 1 \leq i \neq j \leq n\}$ converges in distribution to a stationary (proper) distribution when $t \to \infty$. On the other hand, this class of opinion dynamics is **not stable** if for any $\epsilon > 0$, there exists a class of endogenous processes with $E(n_i^2) < \epsilon$ for all $i$ and such that $Y(t)$ does not converge to a proper distribution when $t \to \infty$.

Note that a deterministic equilibrium or a consensus is a special (and stronger) case of this notion of stability where the limiting distribution is a degenerate distribution. For the Hegselmann-Krause and Deffuant et al. dynamics, opinion differences always converge, whereas in a stochastic dynamics, opinion differences may not converge (in distribution), as we see next.

In the remaining of this section, we focus on the conditions under which the dynamics is stable and also provide tight converses, i.e., conditions under which the dynamics is not stable. We present the results under the following additional technical assumptions:

1. Each of the $f_{i,j}$ functions is non-increasing.
2. In case of centered endogenous processes, the marginal distribution of \( \{n_i(t)\} \) is symmetric and unimodal, i.e., \( \forall k \geq 0, P(n_i(t) = k) = P(n_i(t) = -k) \) is non-increasing with \( k \) and is with finite support.

These assumptions are not always needed. For simplicity we stick to them and make comments when the results do not require these technical assumptions.

2.3.1 Structure of the section

The section discusses many cases and is structured based on the principles.

- The results for dynamics with endogenous noise are discussed first, and those for endogenous bias last.
- We start with the two agent system as the results for system with an arbitrary number of agents build on this.
- Scenarios with symmetry (either all agent interactions are symmetric, or most) are discussed first and completely asymmetric scenarios last.
- the exposition of the results for the \( n \) agent case is structured in terms of the properties of the interaction graph. The nodes of this graph is the set of agents. The undirected version of this graph has and edge between \( i \) and \( j \) if the probability that \( I \) is \( (i,j) \) is positive and either \( f_{i,j} \) or \( f_{i,j} \) are not everywhere equal to 0. We first consider scenarios where the interaction
graph is complete. We then describe more complex scenarios where this graph is not complete.

2.3.2 The Two Body Problem

**Theorem 2.** A two-agent dynamics with endogenous noise is stable if at least one of the agents has a strong influence on the other. Moreover, if both agents have only moderate influence on the other, then the dynamics is not stable.

**Remark 3.** This theorem does not require the technical assumptions 1–2.

First, recall that agent 1 has a strong influence on agent 2 if 2 accepts the opinion of 1 with a probability that decays with opinion difference $y$ no faster than $\frac{c}{y^\alpha}$ for $\alpha < 2, c > 0$. On the other hand, agent 1 has moderate influence on agent 2 if this probability decays faster than $\frac{c}{y^\alpha}$ for some $\alpha > 2$. Thus the stability result and the converse result are almost tight (up to the critical case with $\alpha = 2$).

Towards obtaining a deeper insight into the result above, the following observation is in order. As $f_{i,j}$ is a non-increasing function, we can interpret the event $U_{i,j}^t \in \{1, 0\}$ in the following way: for all $(i, j)$, sample an i.i.d. sequence of random variable $R_{i,j}^t$ with tail distribution function $f_{i,j}$; we call $R_{i,j}^t$ the influence radius (or reach) of $j$ with respect to $i$ at time $t$. Then $j$ influences $i$, i.e., $U_{i,j}^t = 1$ iff $R_{i,j}^t > |x_i(t) - x_j(t)|$. This leads to a direct connection between the pairwise interactions of our model and the class of Pareto distributions (a.k.a. Zipf distributions) [34], namely the class of distributions on $\mathbb{R}_+$ with
tail distribution function having the following form

\[ P(R > r) = \frac{c}{r^\alpha}, \quad \forall r \geq r_0. \]

Here \( \alpha > 0 \) is the exponent of the Pareto distribution and \( c \) is a normalizing constant. Note that the existence of moments depends on \( \alpha \). For example, if \( \alpha < 2 \) then the second moment is infinite whereas if \( \alpha < 1 \) then even expectation is infinite.

Thus agent \( j \) strongly influencing agent \( i \) can be seen as agent \( j \) having an influencing radius distribution with respect to \( i \) which is Pareto like with an infinite second moment. Similarly in case of very strong influence, expectation is infinite.

**Remark 4.** A dynamics with two agents is stable if and only if (up to the critical case) at least one of the agents has an influencing radius with infinite second moment.

### 2.3.3 Complete Interaction Case

We now focus on the \( n \) agent case. In this subsection, we consider three simple scenarios where the interaction graph is complete and with predominantly symmetric influences. We recall that by symmetric influence between \( i \) and \( j \), we mean \( f_{i,j} = f_{j,i} \), i.e., the marginal distribution of \( i \) successfully influencing \( j \) (given \( Y_1 \)) and vice versa are the same.

In the next theorem we consider the case where influences are mutual (mutual influence means that not only \( U_{i,j}^t \) and \( U_{j,i}^t \) have the same marginals...
(given $Y_1(t)$), but in addition, they are same a.s., i.e., $U^t_{i,j} = U^t_{j,i}$, except for one agent (say agent 1) who influences but is never influenced, i.e. $f_{1,j} \equiv 0$ for all $j$. We start with this case because it turns out to be the simplest in terms of proof.

Below, we use the following notation for integer intervals: \{p, p + 1, \cdots, q - 1, q\} := [p : q] for $p \leq q \in \mathbb{Z}$.

**Theorem 5.** Consider an $n$-agent unbiased system with a complete interaction graph. Assume for all $i \neq j$ in $[2 : n]$, agents $i$ and $j$ have symmetric and mutual influence and that $\gamma_{i,j} = \gamma_{j,i}$. Assume that agent 1 influences but is not influenced. Then the dynamics is stable if $f_{j,1}(x) = \Omega \left( \frac{1}{x^{2-\epsilon}} \right)$ for all $j \in [2 : n]$. It is not stable if $f_{i,j} = O \left( \frac{1}{x^{2+\epsilon}} \right)$ for all $i,j \in [1 : n]$.

In the above result there are two assumptions that are not very natural to societies, (i) influences in certain social relations are symmetric (e.g., friends, colleagues) but they are rarely mutual and (ii) it is not common to have social agents who only influence (like agent 1 in Theorem 5) but are never influenced. In the following we relax these restrictions gradually. The next theorem relaxes the mutual interaction assumption and replace it by symmetric interactions.

**Theorem 6.** Consider an $n$-agent unbiased system with a complete interaction graph. Assume for all $i \neq j$ in $[2 : n]$, agents $i$ and $j$ have symmetric influence and that $\gamma_{i,j} = \gamma_{j,i}$. Assume that agent 1 influences but is not influenced. Then the dynamics is stable if $f_{j,1}(x) = \Omega \left( \frac{1}{x^{2-\epsilon}} \right)$ for all $j \in [2 : n]$. It is not stable if $f_{i,j} = O \left( \frac{1}{x^{2+\epsilon}} \right)$ for all $i,j \in [1 : n]$. 

22
The next result relaxes the assumption that there is a special agent who only influences.

**Theorem 7.** Consider an $n$-agent unbiased system with a complete interaction graph. Assume for all $i \neq j$ in $[1 : n]$, agents $i$ and $j$ have symmetric influence and that $\gamma_{i,j} = \gamma_{j,i}$. Then the dynamics is stable if there exists an agent $k$ s.t. $f_{j,k}(x) = \Omega\left(\frac{1}{x^{2\epsilon}}\right)$ for all $j \in [2 : n]$. It is not stable if $f_{i,j} = O\left(\frac{1}{x^{2\epsilon}}\right)$ for all $i, j \in [1 : n]$.

One may see the convergence of the opinion differences to a proper distribution as a “weak” consensus in comparison to consensus where the limiting difference is 0. Then, in the context of social dynamics, the above results have an interesting implication. It says that in a society where everybody interacts with everybody else, the existence of one agent with strong influence on everybody else is sufficient (and to some extent necessary) for reaching a weak consensus.

2.3.4 Interaction Graph

We now move to the case where the interaction graph $G$ is not the complete graph on $n$ nodes.

We first consider the counterpart of Theorem 5, namely the case where influences are symmetric (e.g. mutual) in $[2 : n]$, and where there agent 1 influences but is not influenced.

We introduce a subgraph $G'$ of $G$ called the *strong influence graph*. 
This is the largest subgraph of $G$ where every edge in the graph corresponds to a strong influence: its nodes are $[2 : n]$ and for $2 \leq i, j \leq n$, $(i, j)$ is an edge of $G'$ if $f_{i,j}(x) = \Omega\left(\frac{1}{x^2 - \epsilon}\right)$ for some $\epsilon$.

Note that the strong influence graph defined above does not impact the dynamics in the same way as the friendship graph considered in the literature on linear opinion dynamics on graphs. Unlike in linear opinion dynamics, the fact that two agents have an edge in the strong influence graph does not imply that they incorporate each other’s opinion irrespective of their difference of opinion. The notion of strong influence graph can hence be seen as some combination of the features of bounded confidence and graph based linear dynamics.

Theorem 8. Consider an $n$ agent unbiased system with an interaction graph $G$. Assume that for all $(i, j) \in G$ influences among $\{2, \ldots, n\}$ are symmetric with in addition $\gamma_{i,j} = \gamma_{j,i}$. The dynamics is stable if $G'$ is connected and agent 1 has a strong influence on some agent $l \in [2 : n]$.

In words, if instead of having one agent with strong influence on every other agent, there is a strong influence of agent 1 on at least another agent, and a “path” of strong symmetric influences from the latter to the whole social network, then the dynamics is stable.

The following theorem is the graph extension of Theorem 7. In this theorem, the strong influence graph $G'$ is on $[1 : n]$ with every edge in the graph corresponding to a strong symmetric influence.
**Theorem 9.** Consider an $n$ agent unbiased system with interaction graph $G$. Assume all agent influences are symmetric and that $\gamma_{i,j} = \gamma_{j,i}$ for all $i,j$ pairs. The dynamics is stable if the strong influence graph $G'$ is connected.

The last theorem implies that the connectedness (or strong connectedness) of the underlying social graph leads to weak consensus. This theorem can be seen as an analogue of the results on classical linear opinion dynamics on graphs in the context of non-linear stochastic opinion dynamics with opinion dependent influence.

### 2.3.5 The Case with Bias

We next discuss the case of dynamics with endogenous bias (non centered noise). Towards this we first make two observations. First, all the results that pertain to dynamics with centered noise are also applicable to dynamics with non-centered noise if the mean values of the noise processes are the same for all agents, as opinion differences are not affected by it. Below, we give a sufficient condition for the stability of the dynamics with general (non-centered) i.i.d. noise.

**Theorem 10.** A two-agent dynamics with endogenous bias is stable if at least one of the agents has very strong influence on the other.

Results similar to those of Theorem 5 and 6 hold true where we need the influence to be $\Omega\left(\frac{1}{x^{1-\epsilon}}\right)$ instead of $\Omega\left(\frac{1}{x^{2-\epsilon}}\right)$. Below we state the result corresponding to the Theorem 7 in the case of endogenous bias.
Theorem 11. Consider an $n$-agent system with a complete interaction graph and endogenous bias. Assume for all $i \neq j$ in $[1 : n]$, agents $i$ and $j$ have symmetric influence and that $\gamma_{i,j} = \gamma_{j,i}$. Then the dynamics is stable if there exists an agent $k$ s.t. $f_{j,k}(x) = \Omega \left( \frac{1}{x_{i}} \right)$ for all $j \in [2 : n]$.

This result can be extended to the case where interactions are restricted to a graph.

Theorem 12. Consider an $n$ agent biased system with an underlying interaction graph $G$. Assume influences between $i, j \in [1 : n]$ are symmetric and that $\gamma_{i,j} = \gamma_{j,i}$ for all $i, j$ pairs. Let $G'$ be the subgraph of $G$ where all edges are such that $f_{i,j}(x) = \Omega \left( \frac{1}{x_{i}} \right)$. The dynamics is stable if $G'$ is connected.

Hence, in case of endogenous bias, an infinite expected spread of influence of some agent on others is sufficient for stability. Comparing this with the case of centered noise where unbounded second moment of spread of influence was sufficient, we note that the influences among agents have to be stronger when the agents are biased in their beliefs.

We now investigate the case when influences are asymmetric. To understand the impact of asymmetry we consider the case of extreme asymmetry, where in an interaction between two agents, only one of them can influence the other. Let $\Lambda$ be the following directed graph: it set of nodes is the set of agents. There is directed edge $j \rightarrow i$ iff the probability that $I(t) = (i, j)$ is positive and $f_{i,j} > 0$.  


Theorem 13. Consider a biased opinion dynamics where $\Lambda$ is a tree. If $f_{i,j}(x) = \Omega \left( \frac{1}{x^\epsilon} \right)$ for all $i \rightarrow j \in \Lambda$, then the dynamics is stable.

The graph $\Lambda$ that we call the directed very strong influence graph can be related to leader-follower based social interactions, as those present in e.g., twitter, public forums, blogs where influences are often non-reciprocative. In that context this theorem states that as long as there exist one or more entities or persons whose views eventually reach to everyone (not necessarily in a single hop), the differences in the society do not grow arbitrarily.

2.3.6 On Tail Behavior

So far we have mainly discussed properties of the opinion dynamics pertaining to stability and existence of stationary regimes. As these dynamics are non-linear and stochastic, an analytical characterization of the distribution of opinion differences is a highly non-trivial problem. An explicit functional equation can be written for the steady state distribution, but solving it with full generality is out of the scope of this work. Here we obtain insight into the steady state behavior by bounding the tail of the opinion difference of a two-agent dynamics.

Theorem 14. For an opinion dynamics of two agents with $\{\pm 1\}$ relative endogenous noise, if at least one agent is a very strong influencer, in steady state, the opinion difference has a distribution not heavier than Weibull, i.e., $P(|Y| \geq y) \leq O \left( e^{-ay^b} \right)$ for some $a > 0$ and $0 < b < 1$. 

27
2.4 Generalization of the Results

We presented our theorems under certain assumptions on the endogenous processes, interaction process and influence function. Some of these assumptions can be relaxed.

2.4.1 Endogenous process

We assumed the endogenous processes to be independent across agents, bounded, symmetric and unimodal. As it would have been apparent from the proofs, that for proving stability of the dynamics we do not need the assumption that the endogenous processes are bounded and independent across agents. If the endogenous processes of all the agents behave like an i.i.d. (across time only) vector valued process with all second moments bounded, the stability results hold. In other words, for the class of dynamics where endogenous processes across agents can be arbitrarily correlated the stability results hold. Moreover, in this general class of endogenous processes one can prove that the sufficient conditions for stability in Theorems 8 and Theorem 9 are also necessary up to the critical case (like Theorem 7).

2.4.2 Influence function

We assumed that the influence functions are non-increasing. The stability results extend to a general case, but the proofs of the non-stability results require this assumption.
2.4.3 Interaction process

As our proof techniques are based on Markov property of the dynamics, i.i.d interaction processes are central to the proofs. But we believe the results would be true for non-Markovian ergodic dynamics as well, but may require a different proof technique.

Also, note that as the stability proofs are based on Lyapunov drift arguments and build on theorems that have counterpart for continuous state space, all the stability results generalize to opinion dynamics on $\mathbb{R}$. But, the non stability results build on a theorem specific to Markov chains on positive integers and hence cannot be directly extended to opinion dynamics on $\mathbb{R}$.

2.5 Conclusion

In this work, we presented a generalization of existing bounded confidence opinion dynamics models which captures stochasticity in interactions, opinion-dependent probabilistic exchanges of opinions, self-beliefs and errors in estimation of opinions. Our model extends both graph based opinion dynamics and bounded confidence dynamics within a common stochastic framework. We characterized the conditions under which the dynamics reached a weak consensus, in the sense that opinion differences converge in distribution to a proper steady state distribution. The stochastic bounded confidence opinion dynamics proposed here covers a rich class of dynamics, that we believe will offer qualitative as well as quantitative insights into important social phenomena in the future.
Chapter 3

Inference in Social Networks

3.1 Introduction

Recently social networks, especially online social networks have drawn significant attention from various research communities, including computer science, economics, electrical engineering, mathematics, physics, and social sciences. The interest in social networks are due to various reasons that can be broadly categorized as scientific and economic/political/business. From the scientific point of view, social networks are different from other networks, as they involve interactions among humans. Online social networks provide a unique platform to observe human interactions and social exchanges and to understand them in quantitative terms, which has its own pure scientific interest. On the other hand, as social networks are extremely popular and used by billions around the world, they offer a unique platform for business or political or other social entities to draw attention of the mass. Online marketing [35], online political campaigning [36], online fund-raising [37] are some of the examples where social networks have been successfully used to achieve tangible goals.

These two main aspects of the study of social networks have an inti-
mate connection between them. Better understanding of the social exchanges helps in better controlling of the online social, business or political campaign, and the need to design better campaigning strategies drive the research of understanding social exchanges. This work lies in the intersection of these two research aspects. We consider a problem founded on the understanding of the social exchanges and impacts economic/political/social campaigning.

In social/economic/political campaign, the main goal is to use the underlying social network to spread word of mouth. There have been significant research (see [38, 39] for literature survey) in this direction since the concept of viral marketing on social networks became a popular campaigning strategy. The idea here is to spread a word of mouth to as many agents as possible by appropriately seeding or giving incentives to influential agents in the graph (in terms of degree and other graph parameters). Here we consider a problem motivated by a more personalized or targeted campaigning strategy on social networks.

Viral marketing is based on the central idea that neighbors in the social graph influence each other in adopting an idea/product. Hence, spreading the word at appropriate nodes would result in greater adoption in society. Here all the nodes are treated as identically behaving agents (i.e., with identical belief and adoption model) with different social connections. In reality, different agents have different behavior and adoption model which is strongly correlated to its opinion about the idea/product/topic. Thus for better campaigning strategy it is important to understand the agent-behaviors and design
campaigns appropriately, instead of using a generic strategy. For designing personalized/targeted campaigning strategy for a particular issue/product/topic, we need to have the corresponding sentiments (or opinions) of social agents. But, often the opinions of social agents regarding different topics are not directly available from their profiles (opinions are private). The opinions are only implicitly present in their interactions. The question that we ask here is the following. Can we use social network interactions to understand sentiments?

Towards this we make an important observation. In social networks we can observe interactions among agents about different topics (for example, mining Twitter or Facebook by a topic gives many posts/tweets on that topic and who liked/shared/re-tweeted those posts/tweets). When two agents interact about a topic, their “likes” (or not-“likes”) of each other depend on their closeness of opinions regarding that topic. In general probability of “like” decreases with increasing opinion difference. Thus, in a sense, these interactions (and corresponding likes) capture stochastic realizations of opinion differences. Then a natural question to ask: can we infer the opinions if we have the data of interactions?

We consider the problem of opinion inference from pairwise interactions where opinions lie in a finite subset of \( \mathbb{Z} \) and probability of “like” is a non-increasing function of opinion difference.

Though we consider recovering values in an ordered space from pairwise data, our problem setting is very different from ranking using pairwise comparisons [40–44].
In [45] the authors consider a problem of recovering potentials of nodes that interact and form edges probabilistically according to an exponential function of their potentials. This work is concerned with a particular (exponential) model of interaction and an acceptance probability that is a function of the sum of node potentials (unlike social interactions where it depends on the difference). Though different from our work, this work philosophically inspires our work, as it deals with recovering node parameters just from pairwise binary interactions. Another direction of work which is complementary to the work in this chapter is to infer underlying unknown interaction pattern from the opinion values [46].

Agents with same opinion in a social network can be considered to be a single group. In that sense, the closest relation that our problem has is with the literature on graph clustering (specially, planted partition model). But, as we discuss later, this analogy is not correct and the two problems have significant differences. Planted partition graph clustering in particular (a.k.a stochastic block model) and graph clustering in general is a very rich field which we do not attempt to survey here. To mention a few: [47] proposed the planted partition model, [48–50] did early work on this, followed by significant improvements by [51–54]. In simple terms, in planted partition model there are $K$ equal sized groups of agents. Agents from the same group form edges with probability $p$ and agents across groups form edges with probability $q < p$. Given the graph, the goal is to recover these groups. There are some notable difference between graph clustering in planted partition model and our problem.
In graph clustering the goal is to discover the groups, here we also want to recover the opinions associated with these groups. Towards this we harness the inherent structure in the social interactions: chance of opinion acceptance decreases as the difference grows. We do not restrict to any particular functional dependence between opinion difference and probability of acceptance.

In a social network, interactions of an agent is limited to a social graph. We have to group agents as per their opinions (and infer the opinions as well) by observing these limited interactions. This is unlike the planted partition model where an agent can potentially interact with every other agent. In planted partition model the probability of an inter-partition edge is strictly less than that of an intra-partition edge. But, in social interactions, for two close opinions, often the probability of an inter-opinion “like” is the same as that of an intra-opinion “like” (e.g., bounded confidence dynamics). In addition, as opinions are different for different topics, and popularity of the topics keep changing, a fast algorithm that can work with few samples of interactions is required.

In this work, we consider the problem of opinion recovery from interactions data without the knowledge of the probabilities of “likes” and the size of the set of opinions. We first consider the case where every agent interacts with every other agent. Later we extend our algorithms to the case where interactions are restricted to a social graph. Our algorithm has linear computational and sample complexity, which are arguably order-wise optimal.

This chapter is organized as follows. In Sec. 3.2 we describe the problem in detail. In Sec. 3.3 we first investigate into the special case of bounded
confidence interaction and propose an inference algorithm with theoretical guarantees. In Sec. 3.4 we propose an algorithm for general interactions and state its theoretical guarantees. Followed by that in Sec. 3.5 and 3.6 we propose modifications of the algorithms to improve computational complexity and sample complexity and in Sec. 3.7 we discuss the case where the interactions may change with the system size. In Sec. 3.8 we consider the case where the interactions are limited to a social graph and discuss how the algorithms and the respective theoretical guarantees extend to a broad class of social graphs. Proofs are in Appendix B.

3.2 Inference Model

There are $n$ agents in the system. Each agent $i \in [n] := \{1, 2, \ldots, n\}$ has a scalar opinion $x_i$ which lies in the set $[L] := \{1, 2, \ldots, L\}$, for $L \in \mathbb{Z}_+$. Every agent interacts with every other agent once\(^1\). On an interaction between agents $i$ and $j$, either they mutually like each other or they do not like each other. This is given by $\{0, 1\}$ random variable $e_{i,j} (= e_{j,i})$, where $e_{i,j} = 1$ means that $i$ and $j$ like each other.

For agents $i$ and $j$ with opinions $x_i = l$ and $x_j = l'$, $l, l' \in [L]$, $e_{i,j}$ is independent of any other variable and is 1 with a probability $P_{l,l'}$. $P_{l,l'}$ is symmetric in the subscripts, i.e., $P_{l,l'} = P_{l',l}$. The probability $P_{l,l'}$ is a function of $l'$ for a given $l$. For any $l$, $P_{l,l'}$ is symmetric around $l$, i.e., $P_{l,l'} = P_{l,l''}$ if

\(^1\)Later we consider the case where interactions are restricted to a social graph.
\(|l - l'| = |l - l''|\). \(P_{l,\cdot}\) is non-increasing in \(| \cdot \cdot - l|\) for any \(l\).

After the interactions we have a realization of \(\{e_{i,j} : 1 \leq i \neq j \leq n\}\). Given any such realization of \(\{e_{i,j} : 1 \leq i \neq j \leq n\}\), our goal is to infer the opinions \(\{x_i : i \in [n]\}\) from this realization without the prior knowledge of the interaction probabilities \(\{P_{l,l'}\}\) and the size of the opinion space \(L\).

The likes \(\{e_{i,j}\}\) to constitute an adjacency matrix \(A\) of a graph \(G_L\) of likes on \(n\) nodes. Here each node is an agent and an edge in \(G_L\), given by \(A_{i,j} = e_{i,j} = 1\) is a like. Thus the problem is: given \(A\), infer \(\{x_i\}\) without the prior knowledge of \(\{P_{l,l'}\}\) and \(L\).

### 3.3 Bounded Confidence Interactions

Before considering the inference problem with a general stochastic interaction, we consider a well-known deterministic model of social interaction called the bounded confidence model [6,7].

In bounded confidence dynamics two agents \(i\) and \(j\) like each other or consider each other to be neighbors (friends in the sense of opinion closeness), i.e., \(e_{i,j} = e_{j,i} = 1\) if and only if the opinions of the two agents \(x_i, x_j \in [L]\) are within a distance \(\eta > 0\) of each other. This model of interaction is a special case of the model described in Sec. 5.2 with

\[
P_{l,x} = \begin{cases} 1 & \text{if } |l - x| \leq \eta \\ 0 & \text{else}. \end{cases}
\]

Every agent interacts with every other agent once. We are given the
realizations of \( \{ e_{i,j} \} \). Our goal is to infer \( \{ x_i \} \) from this. We do not know the model parameter \( \eta \), but know that the interactions happen according to the bounded confidence dynamics.

We start with this problem, because bounded confidence is the first model to reflect the dependence between opinion difference and mutual acceptance. Also, as this is a first-order approximation of the more general stochastic interactions, we hope to build on the insights obtained here.

To infer opinions from interactions we need some representative agents for each opinion. So, we assume that for each \( l \in [L] \) there is at least one agent with opinion \( l \). We also assume that \( 1 \leq \eta \leq \frac{L}{2} \). This assumption on \( \eta \) is arguably necessary. To see that consider the case where \( \eta > \frac{L}{2} \). Then agents with opinion \( \frac{L}{2}, \frac{L}{2} + 1 \) and \( \frac{L}{2} - 1 \) (assuming \( L \) is even) are within \( \eta \) of all opinions. So, these three kinds of agents have no difference between them in terms of pairwise interactions. As the only information we have are that of the interactions, these agents are indistinguishable.

We propose Algorithm 1 for inferring opinions \( \{ x_i \} \) from \( \{ e_{i,j} \} \). The main idea behind the algorithm is as follows.

In bounded confidence model, any two agents \( i \) and \( j \) with \( x_i = x_j \) have \( A_{i, \cdot} = A_{j, \cdot} \), where \( A_{i, \cdot} \) is the \( i \)th row of \( A \). On the other hand, no two agents \( i \) and \( j \) with \( x_i \neq x_j \) can have \( A_{i, \cdot} = A_{j, \cdot} \). Hence, agents can be grouped based on their similarity in \( A_{i, \cdot} \), which in turn gives a grouping based on opinions (but the opinions are not known yet). Steps in Agent Grouping is based on
this intuition.

For two agents $i$ and $j$ $A_{i,j} = 1$ if and only if $|x_i - x_j| \leq \eta$. Hence, the edges between any two groups give information about their absolute opinion difference. Again, the groups with extreme opinions 1 and $L$ have edges to least number of other groups. This can be used to differentiate them from others. Also, the number of other groups that a group has edges to increases as opinions move away from the extremes. This can be used to learn the opinions close to the extremes. Also, note that an agent with opinion $l$ has an edge with $l + \eta$ but not with $l + \eta + 1$. This is used to infer opinions of agents that are not close to the extremes. These observations form the basis of Inferring Opinions.

As the interactions only capture the absolute value of opinion differences (without sign), based on the interactions we can find opinions only up to a reversal of ordering. In Side Information we recover the ordering of opinions with a side information about the relative size of the population.

We prove the following result regarding the performance of the Algorithm 1 in the context of bounded confidence interactions.

**Theorem 15.** For a bounded confidence interaction, given $A$ and a strict ordering between the number of agents having opinion 1 and $L$ (and the side information about that), Algorithm 1 infers the opinions $\{x_i\}$ correctly with $O(n^2)$ computations and without the knowledge of $L$ and $\eta$ (given $\eta < \frac{L}{2}$).

The algorithm can recover the opinions up to a reflection about $L + 1$
Algorithm 1 Inference in Bounded Confidence

Input: $A$
Output: $\{\hat{x}_i\}$

Agent Grouping:
Initialize: $K = 1$,
Create set $S_1$ and put 1 in $S_1$.

for $i = 2 : n$ do
    for $k = 1 : K$ do
        Pick $s_k \in S_k$
        if $A_{i,j} = A_{s_k,j}$ then
            $S_k = S_k \cup \{i\}$
            break
        end if
    end for
    if $i \notin S_k, \forall 1 \leq k \leq K$ then
        $K = K + 1$
        $S_K = \{i\}$
    end if
end for

Inferring Opinions:
Form a graph $G_K$ on $K$ nodes: add an edge between $k$ and $k'$ if $\exists i \in S_k, j \in S_{k'}$ s.t. $A_{i,j} = 1$
Form $G_K$ pick nodes $S$ with smallest degree.
Pick $k_1 \in S$
Assign $\ell(k_1) = 1$
Find the set of neighbors of $k_1$, $\beta$ be the cardinality.
Sort neighbors of $k_1$ in $G_K$, say $\{k_2, k_3, \ldots, k_{\beta+1}\}$ according to increasing degree: assign them $\ell(k_2) = 2, \ell(k_3) = 3, \ldots, \ell(k_{\beta+1}) = \beta + 1$.
Initialize: $R = \{k_1, k_2, \ldots, k_{\beta+1}\}, m = 1$

for $m = 1 : K$ do
    Pick $g \in G_k \setminus R$ s.t. $g$ is a neighbor of $k$ with $\ell(k) = m + 1$, but not a neighbor of $k'$ with $\ell(k') = m$
    Assign $\ell(g) = m + \beta + 1$
    $R = R \cup \{g\}$
end for
Side Information: size of populations with opinions $1$ and $L$

\[
\begin{align*}
\text{if } & |S_{k_1}| \text{ and } |S_{k\ast,l(k\ast)}=L| \text{ ordered as per prior information then } \\
& \forall k, \forall i \in S_k, \hat{x}_i = \ell(k) \\
\text{else } & \forall k, \forall i \in S_k, \hat{x}_i = L + 1 - \ell(k) \\
\end{align*}
\]

without the side information regarding the number of agents with opinions $1$ and $L$. By reflection about $K$, we mean that $x$ is indistinguishable from $K - x$.

### 3.4 Stochastic Interactions

Next, we consider the general problem posed in Sec. 5.2. The scenario changes significantly when there is stochasticity. When likes or agreements are stochastic in nature, two agents with same opinion may like two different sets of agents. Hence, the method proposed in Alg. 1 cannot be directly extended here. We have to design a new inference algorithm for the stochastic scenario based on the insights obtained above.

As we aim to infer opinions from pairwise interactions, for successful inference, the pairwise interactions must encode the opinion differences in some sense. If agents interact in a fashion which is irrespective of their opinions, then it is arguably impossible to recover the opinions.

We consider the scenario where the following assumption is true.

**Assumption:** For any $l$, $p(l, x) := P_{l,x}/P_{l,l}$ is a function of $x - l$ which has the following properties: (i) symmetric around $l$ and non-increasing with $|x - l|$, (ii) $p(l, x) = p(l', x')$ for all $l, l'$ if $|l - x| = |l' - x'|$ (i.e., functions are same
up to a shift), (iii) \( \exists x \neq l \) s.t. \( p(l, x) > 0 \), and (iv) \( \exists z \in [L, \frac{L}{2}] > |l - z| > 0 \) s.t. \( p(l, z) - p(l, z + sgn(z)) > 0 \).

Conditions (i) and (ii) imply symmetry of mutual interactions, whereas (iii) implies that each agent likes some opinion other than that of itself with a positive probability. The last condition implies that as the opinion difference keeps increasing from 1, the probability of like (agreement) must change at some point before the difference is \( \geq \frac{L}{2} \). This assumption is arguably a necessity due to the reasons discussed in Sec. 3.3. Note that this assumption does not require that the probabilities with which an agent likes same and different opinions have to be different. In reality an agent may like another agent with close opinion (but not exactly the same) with the same probability with which it likes an agent with the same opinion as itself. In other words, we do not exclude the case where agents across different opinions may have the same density of likes as the agents with the same opinion.

Though Alg. 1 cannot be used here directly, we can use the main structure or the philosophy of the algorithm. The main approach in the algorithm is to first group agents into appropriate mutually exclusive and exhaustive groups so that the agents in the same group have the same opinion. Then based on the interactions (likes) between agents in different groups we infer the opinions.

We propose an inference algorithm along the same idea. The algorithm has two components. The first component \textit{StochasticGroup}, described in Alg. 2 partitions the agents into sets according to their opinions (without knowing
Algorithm 2 StochasticGroup

Input: $A$
Output: $S_{\lambda}$, $1 \leq \lambda \leq \Lambda$, for some $\Lambda \in \mathbb{Z}_+$
Parameters: $W \in \mathbb{Z}_+^+$, $\xi > 0$, $\gamma(n) = n^{\frac{1}{2} + \xi}$

1: Create a set $T$ by picking $c \log n$ agents uniformly at random
2: $\mathcal{T} = \{\text{all subsets of } T \text{ of cardinality } \leq W\}$
3: for $i \in [n]$ do
4: Initialize $d_i = (0, 0, \cdots, 0) \in \mathbb{R}^{|\mathcal{T}|}$
5: for $W \in \mathcal{T}$ do
6: $d_i(W) = \sum_k A_{i,k} \prod_{r \in W} A_{r,k}$
7: end for
8: end for
9: $\Lambda = 1$
10: Put agent 1 in $S_1$.
11: for $i \in [n]$ do
12: for $\lambda = 1 : \Lambda$ do
13: Pick $a_\lambda \in S_\lambda$
14: if $||d_i - d_{a_\lambda}||_1 \leq \gamma(n)$ then
15: $S_\lambda = S_\lambda \cup i$
16: end if
17: end for
18: if $i \not\in S_\lambda$, $1 \leq \lambda \leq \Lambda$ then
19: $\Lambda = \Lambda + 1$
20: $S_\Lambda = \{i\}$
21: end if
22: end for
Algorithm 3 StochasticInfer

Input: \( A, S_\lambda, 1 \leq \lambda \leq \Lambda \)

Output: inferred opinions \( \{ \hat{x}_i \} \)

Parameter: \( \epsilon > 0 \)

1: Pick \( a_\lambda \in S_\lambda, 1 \leq \lambda \leq \Lambda \)
2: Define \( S \subset [\Lambda] \), initialize \( S = \) \( \{ \lambda \} \)
3: for \( \lambda = 1 : \Lambda \) do
4: for \( \lambda' = 1 : \Lambda \) do
5: \( \hat{P}_{\lambda,\lambda'} = \frac{1}{|S_{\lambda'}|} \langle A_{a_\lambda, \cdot}, 1_{S_{\lambda'}} \rangle \)
6: end for
7: Create sets: \( B_1^\lambda = \{ \lambda' : |\hat{P}_{\lambda,\lambda'} - 1| < \epsilon \} \) and \( B_2^\lambda = \{ \lambda' \not\in B_1^\lambda : \max_{\lambda' \in B_1^\lambda} |\hat{P}_{\lambda,\lambda'} - 1| < \epsilon \} \).
8: end for
9: Create directed graphs \( G_1^\Lambda \) and \( G_2^\Lambda \) with nodes in \( \Lambda \) and child nodes of each \( \lambda \in \Lambda \) being \( B_1^\lambda \) and \( B_1^\lambda \cup B_2^\lambda \) respectively.
10: if Any of \( G_1^\Lambda \) and \( G_2^\Lambda \) has asymmetric adjacency matrix then
11: Declare error; break;
12: else
13: if degree of each node in \( G_1^\Lambda \) \( \geq 1 \) then
14: Run Inferring Opinions of Alg. 1 with \( G_1^\Lambda \) instead of \( G_K \)
15: else
16: Run Inferring Opinions of Alg. 1 with \( G_2^\Lambda \) instead of \( G_K \)
17: end if
18: end if

Side information: ordering between population with opinions 1 and \( L \)

1: if \( |S_{\lambda_1}| > |S_{\lambda^*}: \ell(\lambda^*) = L| \) then
2: \( \forall \lambda, \forall i \in S_\lambda, \hat{x}_i = \ell(\lambda) \)
3: else
4: \( \forall \lambda, \forall i \in S_\lambda, \hat{x}_i = L + 1 - \ell(\lambda) \)
5: end if
or recovering the opinions). Then the second component StochasticInfer, described in Alg. 3 infers the opinion of each group based on the inter group interactions. Alg. 2 and Alg. 3 differ significantly from the Agent Grouping and Inferring Opinions steps (of Alg. 1) respectively, but share the similar flavors.

StochasticGroup partitions agents based on the following idea. If we pick $\Omega(\log n)$ agents at random, with high probability $\left(1 - 1 / \text{poly}(n)\right)$ we have agents of all different opinions. Given this, if we consider every subset of these agents (of maximum size $W$) we have all opinion combinations. Then for every agent $i$ we compute a vector $d_i$ whose each component corresponds to such a subset. A component of the vector $d_i$ is the total number of agents that like $i$ and the agents in the corresponding subset.

The intuition is that for two agents with the same opinion these two vectors differ (in $|| \cdot ||_1$ norm) by $O(\sqrt{n}\text{poly}(\log n))$, whereas for two agents with different opinions these two vectors differ by $\Omega(n)$. Hence, using $\{d_i\}$ we would be able to group the agents correctly for a large system.

Given we obtained the correct grouping of agents from StochasticGroup, StochasticInfer associates the right opinion to each group. Towards this it first computes an estimate of the inter-group probabilities of “likes”. Note that as we do not know the group’s opinion yet, we just estimate a vector $\hat{P}_\lambda$ for each group $\lambda$. Then from the symmetry and non-increasing behavior of $pl, \cdot = P_{i, \cdot} / P_{i,i}$, we find the two extreme opinions and the relative opinion differences. After starting with an extreme opinion we find other opinions by

44
induction (using Inferring Opinions steps of Alg. 1).

We have the following guarantee for the inference algorithm which runs StochasticGroup followed by StochasticInfer.

**Theorem 16.** Let $n\frac{\pi_l}{L}$ be the number of agents with opinion $l$, $\pi_l = \Omega(1)$ for all $l$ and there exists a finite $W \in \mathbb{Z}_+$, $\alpha > 0$ such that for any $l, l' \in [L]$, $\sum_x \prod_{i=1}^{w} p(l_i, x)(p(l, x) - p(l', x))\pi_x \geq \alpha$ for some $w \leq W - 1$ (where $\prod_{i=1}^{0} := 1$), then the inference algorithm which runs StochasticGroup + StochasticInfer recovers all opinions correctly with probability $\geq 1 - \frac{1}{n^2}$ in a time $o(n^{2+\delta})$, for any $\delta > 0$ without the knowledge of $\{P_{l,l'}\}$ and $L$, given the prior knowledge that the number of agents with opinion 1 is strictly more (less) than that for opinion $L$.

This theorem implies that without the knowledge of the opinion space $[L]$ and interaction parameters ($\{P_{l,l'}\}$) all opinions can be recovered with a very high probability in almost quadratic time. The side information that the number of agents with opinion 1 is strictly more (less) than that for opinion $L$ the algorithm is useful for exact recovery. It can be shown that without this side information the opinions can still be recovered with high probability, but up to a reflection around $L + 1$.

The assumption in the theorem on $W$ and $p(l, \cdot)$ is not restrictive as it is true in most models. For example, consider the case similar to the stochastic block model, where we have two equal sized opinion groups say 0 and 1, and intra-opinion edges and inter-opinion edges happen with probabilities...
and $q$ respectively. Then if we pick $w = 1$ and $l_1 = 0$, then we have
\[ \sum_x \prod_{i=1}^w p(l_i, x)(p(l, x) - p(l', x)) \pi_x = \frac{5}{2}(p - q) - \frac{1}{2}(p - q) \neq 0. \]
In cases where \{\pi_i\} are chosen (uniformly at random) from a simplex in $\mathbb{R}^L$, the condition is true most of the time for a small $W$.

**Remark 17.** By simple modifications in the parameters $\gamma(n)$ and $|T|$ it is possible to show that probability of correct recovery is $1 - \frac{1}{n \log n}$ while having $o(n^{2+\delta}), \forall \delta > 0$ computational complexity.

### 3.5 Almost Linear Time Algorithms

In inference over a social network, a desirable property of the inference algorithm is low computational complexity. In Sec. 3.3 and 3.4 we proposed algorithms Alg. 1 and Alg. 2 + 3 with computational complexity $o(n^{2+\delta})$ for all $\delta > 0$. As social networks often have more than million nodes, an algorithm with a quadratic computational complexity is not sufficiently fast. For fast computation in practice we would like a linear time algorithm. Linear time is arguably the best that we can have, because to infer opinions we have to iterate over each agent at least once. In this section we discuss how simple adaptations of Alg. 1 and 2 + 3 result in almost linear time algorithms with strong performance guarantees.

#### 3.5.1 Bounded Confidence

We make the following adaptations to Alg. 1.

Construct a subset $U \subset [n]$ by picking agents uniformly at random.

46
Take $|U| = \Theta(\log n)$. Then define

$$
\hat{A} = \{A_{-j} : j \in U\},
$$

i.e., pick all the rows of $A$ but the columns are restricted to the set $|U|$. Then run an algorithm exactly same as Alg. 1 but using $\hat{A}$ instead of $A$. This means that in Agent Grouping we compare $\hat{A}_i$. and $\hat{A}_{sk}$. instead of comparing $A_i$. and $A_{sk}$..

We have the following guarantee for the modified algorithm.

**Theorem 18.** For a bounded confidence interaction, if there are $n\frac{\pi_l}{L}$ number of agents with opinion $l$, and $\pi_l = \Omega(1)$, the above proposed modification of Algorithm 1 infers all the opinions correctly with probability $\geq 1 - \frac{1}{n^2}$ in $o(n^{1+\delta})$ time, $\forall \delta > 0$, without the knowledge of $L$, $\{\pi_l\}$ and $\eta$, given the side information about the ordering between $\pi_1$ and $\pi_L$.

Thus the computation time becomes almost linear in $n$, but instead of deterministic guarantee we have a very high probability guarantee. The benefit in computation time comes from the fact that the previous algorithm does comparison of $n$-length vectors where as this algorithm compares $\log n$ length vectors.

### 3.5.2 Stochastic Interaction

A similar modification as above reduces the computational complexity of Alg. 2+3 to almost linear time. Like in the case of the Alg. 1 the mod-
ification is only needed in the grouping part. We modify StochasticGroup as follows.

Choose

$$\phi(n) = \text{sub-poly}(n),$$

e.g., $$\phi(n) = (\log n)^{10}.$$

Construct a set $$U$$ of size $$\phi(n)$$ by picking agents uniformly at random. Define $$\hat{A}$$ as in Sec. 3.5.1,

$$\hat{A} = \{A_{-j} : j \in U\},$$

and choose

$$\gamma_{\phi}(n) = \phi(n)^{\frac{1}{2} + \xi}.$$

Run the algorithm StochasticGroup with $$\hat{A}$$ and $$\gamma_{\phi}(n)$$ instead of $$A$$ and $$\gamma(n)$$. Thus the only change in the algorithm is in obtaining $$d_i()$$ and comparing $$d_i()$$ and $$d_{a\lambda}()$$. 

**Theorem 19.** Under the same conditions as in Theorem 16, the above modified algorithm has the same correctness guarantees if we choose $$\phi(n)$$ to be $$\omega((\log n)^5)$$ and $$O(\text{sub-poly}(n))$$, and the computational complexity is $$o(n^{1+\delta})$$, $$\forall \delta > 0$$.

**Remark 20.** By simple modifications in the parameters $$\gamma_{\phi}(n), |T|$$ and $$|U|$$ it is possible to show that the probability of correct recovery (in case of all the stochastic algorithms above) is $$1 - \frac{1}{n^{\log n}}$$ with $$o(n^{1+\delta}), \forall \delta > 0$$ computational complexity.
3.6 Inference with Fewer Samples

There is a cost involved in obtaining the data of interactions in a social network. The entity interested in knowing peoples’ opinions often has to pay the social network operator for the data. In general, this payment increases with the size of the data. In this setting a natural question to ask is that whether inference can be done successfully with lesser data.

In Sec. 3.3 and 3.4 we designed inference algorithms where all interaction data are present. In a system of $n$ agents, the data set of all interactions is of size $O(n^2)$. On the other hand, to infer opinions of $n$ agents at least $\Omega(n)$ samples are needed, if the opinions are not strongly correlated. This is because, even if we know opinions of $n-1$ agents, to know the opinion of $n$th agent, we need at least one sample of interaction between this agent and any other agent.

We consider the case where the entity interested in opinions can specify the interactions whose data it wants. If we have the freedom to choose the interaction data, then a natural question to ask: what is the minimum size of the data from which we can recover the opinions?

Towards this we take the following approach based on the algorithms and their modifications discussed in Sec. 3.4 and 3.5.

We start with the deterministic bounded confidence model and propose the following algorithm that builds on Alg. 1.

*Algorithm*: Let $\psi(n) = \text{sub-poly}(n) = \omega((\log n)^{10})$. Construct disjoint
sets (subsets of \([n]\)) \(\{U_r : 1 \leq r \leq \frac{n}{\text{sub-pol}(n)}\}\) each of size \(\psi(n)\) by picking agents uniformly at random from \([n]\). For each \(U_r\) collect the data of interactions only between agents in \(U_r\), say \(A^r \subset A\). For each \(U_r\) run Alg. 1 (exactly as in Sec. 3.3). Obtain \(\{\hat{x}_i : i \in U_r\}\) from each \(U_r\).

**Theorem 21.** Under the same condition as in Theorem 18 the above algorithm has the same correctness guarantee and the same time complexity with \(o(n^{1+\delta})\), \(\forall \delta > 0\) pairwise interaction samples.

A similar scheme as in the case of bounded confidence model works well in the stochastic interaction. Here instead of running Alg. 1 for each \(U_r\) we have to run StochasticGroup followed by StochasticInfer (as in Sec. 3.4, not the modified one in Sec. 3.5) on \(A^r\) with \(\gamma_\psi(n)\) instead of \(\gamma(n)\), where

\[
\gamma_\psi(n) = (\psi(n))^{\frac{1}{2}+\xi}.
\]

**Theorem 22.** Under the same conditions as in Theorem 19, the above modified algorithm has the same correctness and time complexity guarantee with \(o(n^{1+\delta})\), \(\forall \delta > 0\) pairwise interaction samples.

### 3.7 Scaling of Unobserved Parameters

In previous sections, for simplicity of presentation we stated Theorem 16, 19 and 22 for the case when \(\{P_{l,l'}\}\) are constants and do not scale with the number of agents. Our proofs show that the above results extend almost directly to the case where \(P_{l,l'} = O\left(\frac{1}{u(n)}\right)\), for some \(u(n)\) which is sub-polynomial.
in $n$, i.e., $u(n) = o(n^\delta)$ for any $\delta > 0$. In this case we only have to choose $\psi(n)$ and $\phi(n)$ (discussed before) to be $\omega((u(n))^{2W}(\log n)^{10})$. Note that we do not need to know the scaling of $P_{l,l'}$ exactly, a lower-bound (in order sense) is sufficient.

Also, note that the above results have been stated for $L = O(1)$. The recovery guarantees of the theorems also extend to the case where $L = O((\log n)^k), k \in \mathbb{Z}_+$ through a minor modification of the algorithms. For $StochasticGroup$ we have to choose $T$ s.t. $|T| = \Omega(c\tilde{L}(\log n)^2)$, for some upper-bound $\tilde{L}$ on $L$. As would be apparent from the proofs that this does not change the theoretical guarantees. Note that we do not need to know $L$, an upper-bound (in order sense) is sufficient.

In case of the linear time algorithm for bounded confidence interaction, we need to choose $U$ s.t. $|U| = \Theta(\tilde{L}(\log n))$. Similar guarantees hold here.

### 3.8 Interactions Limited to Social Graph

In the sections above we designed algorithms for inferring opinions of agents based on only pairwise interactions and showed that the algorithms recover all the opinions correctly with probabilities close to 1. One main assumption in the designs of the algorithms as well as in the derived guarantees is that every agent interacts with every other agent in the network. This assumption is true for many not-so-large societies, e.g., social network of small towns, small/medium enterprises, small/medium fraternities etc. But as these societies are of the size $10^2 - 10^3$, these systems are not really in large $n$ regime.
Still, as the guarantees (high probability) are polynomial (in fact, faster than polynomial) in the size of the societies, the algorithms have strong performance guarantees for these systems.

The immediate question that arises then is what can we say about societies with a general social graph (instead of a clique) where an agent does not interact with all other agents. Interestingly, the methodologies developed in the previous sections extend to many such cases, where the interactions are limited to a social graph. Our algorithms and the guarantees extend to a wide class of such social graphs seen in practice.

In this setting the problem changes as follows. We have $n$ nodes, but not all of them interact, rather their interactions are confined to a social graph $G$. We observe results of these interactions (likes or not) only for the edges of this graph (unlike previously where we could potentially have $n^2$ interactions). The goal is to infer opinions of all the agents from these interactions.

It is well known that the social graphs are often composed of multiple cliques or clusters, i.e., a social graph contains many dense subgraphs which are in turn connected by relatively fewer edges between them. For such a social network, under certain conditions on the graph structure, if we are given interaction data (restricted to $G$) we can still use our methodology to infer the opinions of the agents.

Consider the case where the social graph has clusters (or cliques) of poly-logarithmic sizes, i.e., each cluster/clique is $\Omega((\log n)^2)$ and each node is
in (at least) one of these. Such a graph structure is common in many online and off-line social networks. For example, for a social network of million nodes, it means that we have cliques of size $10^1 - 10^2$, which is likely as many of the friends have multiple common friends. Let each cluster has enough opinion diversity, i.e., each cluster has $\Omega\left(\frac{1}{L}\right)$ fractions of agents per opinions. This is in general true if opinion of each agent is independent of others. We assume that $\{P_{l,l'}\}$ and $L$ are constants.$^2$

In this case, we can use the modified algorithm with linear sample complexity, because here we have sub-polynomial (but more than logarithmic) groups. The same performance and computational guarantees extend to this case as well. For this social graph if the clusters are known a priori we can directly run our algorithm for each cluster. On the other hand, if the clusters are not known, we can first run a graph clustering algorithm to obtain the clusters (e.g., [50]).

In social networks interactions happen regarding many topics of common interest. Each agent has an opinion on each (or most of) of such topics. To infer these topic related opinions we have to collect the data of interactions related to each such topic and for each topic we need to run our algorithm. This is where the linear time and sample complexity (which is arguably order optimal) of our algorithm is very useful. On the other hand, as the social graph is the same irrespective of the topic, we need to run the graph cluster-

$^2$Results extend when they scale, if the cliques/clusters are order-wise larger.
ing algorithm only once (which is often computationally more expensive, e.g. spectral clustering is $O(n^3)$).

There is another kind of social network structure that is commonly seen in practice, the core-periphery network [55]. In such networks there is a dense core of size much smaller than the whole network and a large sparse periphery. Though this model looks different from the previous one, our methodology still extends here.

Consider a core-periphery network with $O(n)$ periphery nodes and $\Omega((\log n)^2)$ core nodes. Core nodes form a clique and each periphery node has $\log n$ edges to the core and few edges to other periphery nodes. This means that each periphery is connected to a vanishing fraction ($O(\frac{1}{\log n})$) of the core nodes and the network between periphery nodes is very sparse. Also, let the opinion of each agent is independent of other agents.

First, we can use existing techniques [55] to find the core of the network. Given the knowledge of core, we can run StochasticGroup followed by StochasticInfer for these nodes. As the core is of poly-logarithmic size, from Theorem 22 it follows that this method would recover opinions with high probability. After the opinions of the cores have been recovered, for each periphery node, we collect the interactions of it with the core and using the same techniques as in Sec. 3.4 we can estimate the relative opinion difference of the periphery node from the core nodes. As in the core we have all opinions, these relative differences in turn can be used to infer the opinion of the periphery node.
Thus, we see that though we developed the inference algorithms for clique-like social network, they extend to a broad class of social networks often encountered in practice.

3.9 Conclusion

Motivated by targeted (and/or personalized) campaigning on online social networks we considered the problem of inferring opinions regarding a topic from observed interactions. We considered a generic model of opinion dependent interactions in social networks and proposed inference algorithms with low sample and computational complexity. Our algorithm has provable performance guarantees for a large class of social networks seen practice.
Chapter 4

Skilled Crowdsourcing

4.1 Introduction

Methods and structures for information processing have been changing. Enabled by the proliferation of modern communication technologies, globalization and specialization of workforces has led to the emergence of new decentralized models of informational work. Moreover, the millennial generation now entering the workforce often favors project-based or job-based work, as in crowdsourcing and social production [56, 57], rather than long-term commitments [58]. Indeed over the last decade, more than 100 ‘human clouds’ have launched with a variety of structures. These platforms serve clients by harnessing external crowds, and global enterprises similarly harness their internal crowds [59–61], making use of human cognitive surplus for information processing [62].

Platforms follow different collective intelligence models [63, 64], and require different strategies for allocating informational work to workers. In crowdsourcing contest platforms like InnoCentive and TopCoder, there is self-selection: work is issued as an open call and anyone can participate in any job; the best submission wins the reward [65–68]. In microtask crowdsourcing plat-
forms like Amazon Mechanical Turk, any worker is assumed able to do any job and so first-come-first-serve strategies are often used; level of reliability may be considered in optimal allocation [69]. In freelance markets like oDesk and Elance, however, specialized jobs must be performed by skilled workers: allocation requires careful selection from the large pool of variedly-skilled freelancers.

Freelance market platforms serve as spot markets for labor by matching skills to tasks, often performing on-demand matching at unprecedented scales. For example, oDesk had 2.5 million workers and nearly 0.5 million clients in 2013 [64]. Herein we study allocation and scheduling of informational work within these kinds of platforms, via a queuing framework. We aim to establish fundamental limits through a notion of work capacity, and also develop decentralized algorithms, which are easily-computed, that nearly achieve these performance limits.

Freelancers may have one or more skills (that are known, cf. [70]) and jobs may have multiple parts, called tasks, that require separate skills. Due to job skill requirement variety and limited freelancer ability, it is often not possible to find a freelancer that meets all requirements for a job: a job may have to be divided among freelancers. Moreover, a task in a job may require so much time that even the task may have to be divided among multiple freelancers. There are reputation systems within freelance market platforms, so freelancers have a reputation level as well as minimum acceptable hourly rate and skills, which allow worker categorization. Some freelancers are adaptable
in terms of hours available to spend on a particular type of task, whereas
others pre-specify hours available for each kind of task. Here we consider the
non-adaptable setting where, for example, a freelancer may be available for 20
hours (per week) of any C++ or Java programming, or may be available for 10
hours (per week) of C++ and 10 hours of Java. Studying limits for adaptable
freelancers and designing centralized schemes (and their approximations) are
similar, but the distributed schemes require a different approach.

The objective of the platform is to find a good allocation of jobs (and
tasks) to freelancers. Since working on a task requires synchronization among
freelancers, work can only start when the whole task has been allocated. On
the other hand, for some jobs there are interdependencies between different
tasks [71] and hence, for these jobs all tasks must be allocated before the job
starts. Moreover, some jobs may require all parts to be done by freelancers
with the same level of expertise for uniform quality and money spent. These
considerations lead to concepts of decomposability and flexibility that are cen-
tral to our development.

In the ethos of self-selection, it is desirable for crowd systems to not be
centrally controlled, but rather for jobs and freelancers to choose each other.
Currently, this may happen randomly or greedily. This is clearly not optimal,
as the following example illustrates. Consider two types of jobs (single task)
and two categories of freelancers. A type 1 job can be served by either of the
worker categories (example, lower reputation requirement) whereas type 2 jobs
can only be served by category 2 workers. If freelancers and jobs are allocated
arbitrarily then it may happen that type 1 uses many category 2 freelancers and many type 2 jobs remain unserved.

Optimal centralized allocation of informational tasks under the constraints of crowd systems is related to hard combinatorial problems such as the knapsack problem. Compared to scheduling problems in computer science [72], communication networks [73,74], and operations research [75], crowd systems face challenges of freedom of self-selection, need for decentralized operation, and uncertainty in resource availability.

Prior works in the information theory, networking, and queueing literatures are similar to our work in terms of theoretical framework, performance metrics, and the nature of performance guarantees, but are not directly related. The notion of capacity of a resource-shared system where jobs are queued until they are served and the notion of a capacity-achieving resource allocation scheme for this kind of system came to prominence with the work of Tassiulas and Ephremides [76, 77]. The capacity concept and capacity-achieving schemes were subsequently developed for applications in communication networks [73, 74, 78, 79], cloud computing [80], online advertising [81, 82], and power grids [83], among others. With the advent of cloud services, large-scale systems have attracted significant research interest: resource allocation schemes and their performance (queueing delays, backlogs, etc.) in the large-scale regimes have been studied [?, 84–86].

In this chapter, our goal is to understand the fundamental limits (capacity) of freelance markets and ways to achieve this ultimate capacity. We
first develop a centralized scheme for achieving these maximum allocations
where a central controller makes all job allocation decisions. Given the po-
tential large scale of platforms, we also discuss low-complexity approximations
of the centralized scheme that almost achieve the limit. Finally, with an eye
towards giving flexibility to customers (job requesters) in choosing freelancers,
we propose simple decentralized schemes with minimal central computation
that have provable performance guarantees. Further, since job arrival and
freelancer availability processes are random (and sometimes non-stationary),
we also address ways to adapt when the system is operating outside its capacity
limits.

4.2 System Model

We first provide formal definitions of the nature of informational work
and workers, and establish notation.

Freelancers (or agents) are of \( L \) categories. In each category \( l \in [L] \),
there are \( M^l \) types of agents depending on their skill sets and available hours.
There are \( S \) skills among agents of all categories and types. An agent of
category \( l \) and type \( i \) has a skill-hour vector \( h^i_l \), i.e. \( h^i_{l,s} \) available hours for
work involving skill \( s \in [S] \).

Jobs posted on the platform are of \( N \) types. Each type of job \( j \in [N] \)
needs a skill-hour service \( r_j \), i.e. \( r_{j,s} \) hours of skill \( s \). A part of a job of type \( j \)
involving skill \( s \) is called a \((j, s)\)-task if \( r_{j,s} > 0 \), which is the size of this task.
A job of type $j$ can only be served by agents of categories $l \in N(j) \subset [L]$. This restriction is captured by a bipartite graph $G = ([N], [L], E)$, where a tuple $(j, l) \notin E \subset [N] \times [L]$ implies that category $l$ agents cannot serve jobs of type $j$.

On the platform, jobs are allocated at regular time intervals to available agents, these epochs are denoted by $t \in \{1, 2, \ldots\}$. Jobs that arrive after epoch $t$ has started are considered for allocation in epoch $t + 1$, based on agents available at that epoch. Unallocated jobs (due to insufficient number of skilled agents) are considered again in the next epoch.

Jobs arrive according to a $\mathbb{Z}_+^N$-valued stochastic process $A(t) = (A_1(t), A_2(t), \ldots, A_N(t))$, where $A_j(t)$ is the number of jobs of type $j$ that arrive in scheduling epoch $t$.

The stochastic process of available agents at epoch $t$ is $U(t) = (U^1(t), U^2(t), \ldots, U^L(t))$. For each agent category $l$, $U^l(t) = (U^l_1(t), U^l_2(t), \ldots, U^l_{M_l}(t))$ denotes the number of available agents of different types at epoch $t$.

We assume processes $A(t)$ and $U(t)$ are independent of each other and that each of these processes is independent and identically distributed for each $t$. We also assume that each of these processes has a bounded (Frobenius norm) covariance matrix. Let $\Gamma(\cdot)$ be the distribution of $U(t)$, and let $\lambda = \mathbb{E}[A(t)]$ and $\mu^l = \mathbb{E}[U^l(t)]$ for $l \in [L]$ be the means of the processes.

At any epoch $t$, only an integral allocation of a task (say $(j, s)$) is

\footnote{Most of our results can be extended to stationary ergodic processes.}
possible. A set of tasks $t_1, t_2, \ldots, t_n$ of size $r_1, r_2, \ldots, r_n$ of skill $s$ can be allocated to agents $1, 2, \ldots, m$ only if available skill-hours for skill $s$ of these agents $h_1, h_2, \ldots, h_m$ satisfy

\[
\sum_{p=1}^{n} v_{ip} \leq h_i, \sum_{q=1}^{m} v_{qj} \geq r_j, j \in [n], i \in [m]
\]

for some \( \{v_{pq} \geq 0\} \).

Whether different tasks of a job can be allocated at different epochs and across different categories of agents depend on the type of the job.

**Definition 23.** A type of job $j$ is called non-decomposable (decomposable) if different tasks comprising it are (are not) constrained to be allocated at the same epoch.

**Definition 24.** A type of job $j$ is called inflexible (flexible) if different tasks as well parts of tasks comprising it are (are not) constrained to be allocated to the same category of agents.

In a system with only decomposable jobs, given a set of \( \{u^l = (u^l_1, u^l_2, \ldots, u^l_{M_l}) : l \in [L]\} \) agents (that is, $u^l_i$ agents of category $l$ and of type $i$ within that category), a number $a_{j,s}$ of $(j,s)$-tasks can be allocated only if there exist non-negative \( \{z^l_{j,s} : (l, j, s) \in [L] \times [N] \times [S]\} \) satisfying

\[
\sum_{l} z^l_{j,s} = a_{j,s}, z^l_{j,s} = 0 \text{ if } (j,l) \notin E, \text{ for all } j \in [N], s \in [S],
\]

\[
\sum_{j \in [N]} z^l_{j,s} r_{j,s} \leq \sum_{i \in [M_l]} u^l_i h^l_{i,s}, \text{ for all } l \in [L], s \in [S] . \tag{4.1}
\]
On the other hand, given a set of \( \{u_l = (u^l_1, u^l_2, \ldots, u^l_M) : l \in [L] \} \) agents in a system with only non-decomposable jobs, \( a_j \) jobs of type \( j \) (for each \( j \)) can be allocated only if there exist non-negative \( \{z^l_{j,s} : (l, j) \in [L] \times [N] \} \) satisfying

\[
(4.1) \text{ and } a_{j,s} = a_j \text{ for all } j, s.
\]  

Intuitively, the conditions imply that required skill-hours for the set of jobs is less than the available skill-hours of agents. The \( \{z^l_{j,s}\} \) capture a possible way of dividing tasks across multiple category of agents, as they can be interpreted as the number (possibly fraction) of \( (j, s) \)-tasks allocated to \( l \)-category agents. Note that conditions (4.1) and (4.2) are necessary for allocations of decomposable and non-decomposable jobs respectively. These conditions only imply that there exist possible ways of splitting jobs and tasks across different categories of agents to ensure integral number of tasks (jobs) are allocated in case of decomposable (non-decomposable) jobs.

For a system with only flexible jobs, different parts of a task can be allocated to different categories and a category can be allocated parts of tasks. Hence, \( \{z^l_{j,s}\} \) can possibly take any value in \( \mathbb{R}_{+}^{LNS} \). Thus flexible and decomposable (non-decomposable) jobs need to satisfy condition (4.1) (condition (4.2)) which we refer to as \( \text{FD (FND)} \).

For inflexible jobs, a necessary condition the allocation must satisfy is that each category gets the same integral number of \( (j, s) \)-tasks for all \( s, j \), i.e.,

\[
z^l_{j,s} \in \mathbb{Z}_{+} \text{ s.t. } z^l_{j,s} = z^l_{j,s'} \text{ for all } s, s', j.
\]  

(4.3)
An allocation of inflexible and decomposable (non-decomposable) jobs needs to satisfy conditions (4.1) (condition (4.2)) and (4.3), which we refer to as \textit{ID (IND)}.

For simplicity, in this work we focus on systems with only a single one of these four classes of jobs.\textsuperscript{2} For brevity we use the same abbreviations to refer to class of job, as we use for the necessary conditions. Thus, we have FD, FND, ID, and IND systems.

In crowd systems, the scaling of number of job and agent types, rate of job arrivals, and number of available agents is as follows: \( \lambda(N) = \sum_{j=1}^{N} \lambda_j \) scales faster than \( N \), i.e. \( \lambda(N) = \omega(N) \) or \( \lim_{N \to \infty} N/\lambda(N) = 0 \) and the number of skills \( S \) scale slower than \( N \), i.e. \( S = o(N) \). In practice, a job requires at most a constant number of skills \( d \), implying there are \( \Omega(S^d) \) possible job types. On the other hand, the number of skills of an agent \( d' < d \) as a job generally requires more diversity than a single agent possesses, implying \( M = \sum_{l} M^l = O(S^d) \). \( L = O(1) \), as it relates to variation in reputation levels and hourly rates, and so \( M = o(N) \). Beyond these system scalings seen in practice, we assume \( \lambda_j(N) = \omega(1), \forall j \in [N] \) and \( \sum_{j: r_{j,s} > 0} \lambda_j(N) = \Omega(N^c) \) for all \( s \in [S] \), for some \( c > 0 \). In the sequel, we assume these scaling patterns and refer to them as \textit{crowd-scaling}.

\textsuperscript{2}Extension to combinations of multiple classes is not much different but requires more notation.
4.3 Capacity, Outer Region, and Centralized Allocation

In this section we study the limits of a freelance market with centralized allocation and present a centralized algorithm that achieves the limit. We also discuss a simpler upper bound for the capacity region in terms of first-order statistics of the system. These results on ultimate system limits and ways to achieve them are not only important in their own right, but also serve as benchmarks for later discussion of decentralized schemes that provably achieve nearly the same limits.

To formally characterize the maximal supportable arrival rate of jobs we introduce some more notation. For each \( j \in [N] \), let \( Q_j(t) \) be the number of unallocated jobs that are in the crowd system just after allocation epoch \( t - 1 \). As defined above, \( A_j(t) \) is the number of jobs of type \( j \) that arrive between starts of epochs \( t - 1 \) and \( t \). Let \( D_j(t) \) be the number of jobs of type \( j \) that have been allocated to agents at epoch \( t \); we call a job allocated only when all parts have been allocated. Thus the evolution of the process \( Q_j(t) \) can be written as:

\[
Q_j(t + 1) = Q_j(t) + A_j(t) - D_j(t). \tag{4.4}
\]

Note that at any epoch \( t \), at most \( Q_j(t) + A_j(t) \) type \( j \) jobs can be allocated, as this is the total number of type \( j \) jobs at that time and hence \( D_j(t) \leq Q_j(t) + A_j(t) \), implying \( Q_j(t) \geq 0 \).

**Notation and Convention.** We denote the interior and the closure of a set \( C \) by \( \bar{C} \) and \( \bar{C} \), respectively. When we say \( \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N) \in \Lambda \subset \mathbb{R}_{+}^{NS} \)
we mean $\lambda^S = ((\lambda_1, \lambda_1, \ldots, S \text{ times}), (\lambda_2, \lambda_2, \ldots, S \text{ times}), \ldots) \in \Lambda$. Also, whenever we say $\Lambda \subseteq (\supseteq) \Lambda'$ for $\Lambda' \subset \mathbb{R}_+^N$, we mean for any $\lambda \in \mathbb{R}_+^N$, $\lambda \in \Lambda' \iff (\Rightarrow) \lambda^S \in \Lambda$.

**Definition 25.** An arrival rate $\lambda$ is stabilizable if there is a job allocation policy $P$ under which $Q(t) = (Q_j(t), j \in [N])$ has a finite expectation, i.e., $\limsup_{t \to \infty} E[Q_j(t)] < \infty$, for all $j$. The crowd system is called stable under this policy.

**Definition 26.** $\mathcal{C}_\Gamma$, a closed subset of $\mathbb{R}_+^N$ is the capacity region of a crowd system for a given distribution $\Gamma$ of the agent-availability process if any $\lambda \in \mathcal{C}_\Gamma$ is stabilizable and any $\lambda \notin \mathcal{C}_\Gamma$ is not stabilizable.

### 4.3.1 Capacity Region and Outer Region

Let us characterize the capacity regions of different classes of crowd systems. For any given set of available agents $u = (u^l_i : 1 \leq i \leq M^l, 1 \leq l \leq L)$, we define the set of different types of tasks $(a_{j,s})$ that can be allocated in a crowd system. Note that the necessary conditions to be satisfied for tasks to be allocated are specific to the class of crowd system.

Using the explicit conditions (4.1), (4.2), and (4.3) for tasks (jobs) to be allocated, we define $C^{FD}(u)$, $C^{FND}(u)$, $C^{ID}(u)$, and $C^{IND}(u)$ as the set of tasks that can be allocated in FD, FND, ID, and IND systems respectively for given availability $u$. We denote these sets generically by $C(u)$ and refer to conditions FD, IFD, FND, and IND generically as crowd allocation constraint
or CAC.

\[ C(u) := \{ (a_{j,s} \in \mathbb{Z}_+) : \exists (z_{j,s}^l) \text{ satisfying CAC} \}, \]

and \( C(u) \) is the convex hull of \( C(u) \).

The following theorem generically characterizes capacity regions of different crowd systems.

**Theorem 27.** Given a distribution \( \Gamma \) of agent-availability, i.e., \( \Gamma(u) = P(U(t) = u) \), for a \( \lambda \notin \bar{C}(\Gamma) \) there exists no policy under which the crowd system is stable, where

\[
\bar{C}(\Gamma) = \left\{ \lambda = \sum_{u \in \mathbb{Z}_+^M} \Gamma(u) \lambda(u) : M = \sum_l M^l, \lambda(u) \in C(u) \right\}.
\]

For FD, FND, and IND systems, for any \( \lambda \in \bar{C}(\Gamma) \) there exists a policy such that the crowd system is stable.

**Proof.** See Appendix C Sec. C.1.1.

This implies that for FD, FND, and IND systems capacity region \( \mathcal{C}_\Gamma = \bar{C}(\Gamma) \) and for ID systems \( \mathcal{C}_\Gamma \subseteq \bar{C}(\Gamma) \) (possibly strict). Note that the conditions FD, FND, ID, and IND (generically CAC) are necessary conditions for a valid allocation. The above theorem implies these conditions are also sufficient, except for ID systems. In Sec. 4.5, we present an alternate characterization of the capacity region for inflexible systems.

Note \( \bar{C}(\Gamma) \) depends on the distribution of agent availability \( \Gamma \), but it is hard to obtain this distribution for large and quickly-evolving systems in
practice. Hence, a characterization in terms of simpler system statistics is of use. Below is a characterization of a region beyond which no arrival rate can be stabilized. Borrowing terminology from multiterminal Shannon theory, we call this the outer region.

For any set \( J \subset [N] \), define \( \mathcal{N}(J) = \{ l \in [L] : \exists j \in J \text{ s.t. } (j, l) \in E \} \) and the closed subset of \( \mathbb{R}^N_+ \),

\[
\mathcal{C}^\text{out}_\mu = \left\{ \lambda : \forall J \subset [N], \forall s, \lambda_j r_{j,s} \leq \sum_{l \in \mathcal{N}(J)} \sum_{i \in M^l} \mu^l_i h^l_{i,s} \right\}.
\]

**Theorem 28.** For any distribution \( \Gamma \) with mean \( \mu \), \( \mathcal{C}_\Gamma \subseteq \mathcal{C}^\text{out}_\mu \).

**Proof.** See Appendix C Sec. C.1.2. \( \square \)

In general, \( \mathcal{C}_\Gamma \) is a strict subset of \( \mathcal{C}^\text{out}_\mu \) because \( \mathcal{C}^\text{out}_\mu \) only captures the balance of skill-hours in the crowd-system, i.e. average skill-hours requirement is no more than average availability, but partial allocation of a task is not acceptable in a crowd system. Moreover, for non-decomposable jobs all tasks of a job have to be allocated simultaneously. Hence, meeting an average skill-hour balance criterion may be far from being sufficient for stability. For inflexible systems the requirements are even stricter, which is likely to increase the gap between the outer region and the true capacity region. In Sec. 4.5 we present a tighter outer region for inflexible systems.

In certain scenarios \( \mathcal{C}^\text{out}_\mu \) may be non-empty when \( \mathcal{C}_\Gamma \) is empty. For example, consider a simple non-decomposable crowd system with \( N = L = 1 \).
and \( M^1 = S = 2 \). Let each job require 1 hour of both skills, type \( i \) agents have only 1 hour available for skill \( i \) and none for other skills, \( U^1(t) \) be uniformly distributed on \( \{(0,10),(10,0)\} \), and \( \lambda = (4,4) \). Then clearly \( \lambda \in C_{\mu}^{\text{out}} \), but note that at any time there is only one type of skill available, hence no job can be allocated. This implies \( \mathcal{C}_\Gamma = \emptyset \).

4.3.2 Centralized Allocation

Though there exists a policy for each \( \lambda \in \hat{\mathcal{C}}_\Gamma \) that stabilizes the system, these policies may differ based on \( \lambda \) and may depend on the job-arrival and agent availability statistics. Changing policies based on arrival rate and statistics is not desirable in practical crowd systems due to the significant overhead. Below we describe a centralized statistics-agnostic allocation policy which stabilizes any \( \lambda \in \hat{\mathcal{C}}_\Gamma \). Later we discuss computational cost of this policy for different classes of crowdsourcing system and present simpler distributed (or almost distributed) schemes with provable performance guarantees under some mild assumptions.

To describe the scheme we introduce some more notation. Let \( Q_{j,s}(t) \) be the number of \( s \)-tasks (skill \( s \)) of type \( j \) jobs just after the allocation epoch \( t - 1 \) and let \( D_{j,s}(t) \) be the number of \( s \)-tasks (skill \( s \)) of type \( j \) jobs allocated at epoch \( t \). Then

\[
Q_{j,s}(t + 1) = Q_{j,s}(t) + A_j(t) \, \mathbb{1}(r_{j,s} > 0) - D_{j,s}(t).
\]

Note that for all \( t \), due to the CAC condition on allocation, \( D_{j,s}(t) \in \)
Moreover, there is an additional restriction that $D_{j,s}(t) \leq Q_{j,s}(t) + A_j(t)$, as there are $Q_{j,s}(t) + A_j(t)$ part $s$ of job type $j$ in the system at that time, which in turn implies $Q_{j,s}(t) \in \mathbb{Z}_+$ for all $t$. Note that as $D_{j,s}(t) \in C(U(t))$, for non-decomposable systems $Q_{j,s} = Q_{j,s'}$ for all $j, s, s'$, whereas for decomposable systems they may differ.

We propose the MaxWeight Task Allocation (MWTA) policy, Alg. 1, to allocate tasks to agents at epoch $t$ based only on the knowledge of $Q(t)$, $A(t)$, and $U(t)$, and therefore statistics-agnostic. It is based on MaxWeight matching \cite{73,74}.

It is apparent that the MaxWeight part of the algorithm finds a $\{z_{j,s}\}$ that satisfies CAC. The following theorem implies that MWTA allocates tasks optimally. The proof of the theorem is based on adapting the proof of optimality of the MaxWeight algorithm under the constraints and assumptions of crowd systems. It implicitly relies on the following result.

**Proposition 29.** For any $u$ and $Q$, and $\{z_{j,s}\}$ satisfying CAC, the Task Allocation part of MWTA (Alg. 1) gives a feasible allocation for FD, FND, and IND systems.

**Proof.** See Appendix C Sec. C.1.3.

**Theorem 30.** MWTA (Alg. 1) stabilizes FD, FND, or IND crowd systems for any arrival rate $\lambda \in \hat{C}_\Gamma$ (for respective $C_\Gamma$).

**Proof.** See Appendix C Sec. C.1.4.
Algorithm 4 MaxWeight Task Allocation (MWTA)

Input: \( \{Q_{j,s}(t) : j \in [N], s \in [S]\} \), \( A(t) \) and \( U(t) \) at \( t \)
Output: Allocation of jobs to agents

MaxWeight

\[
\left( z^l_{j,s}(t) : l, j, s \right) = \arg \max_{\left( z^l_{j,s} \in \mathbb{Z}_+:l,j,s \right)} \sum_{j,s} Q_{j,s}(t) \Delta_{j,s}
\]

s.t. \( \left( z^l_{j,s} \right) \) satisfy CAC with \( a_{j,s} = \Delta_{j,s}(t) \forall j, s. \)

Task Allocation

for \( j = 1 : N \) do
    Order \( j \)-type jobs arbitrarily, \( O_j \)
    for \( s = 1 : S \) do
        Use order \( O_j \) among non-zero \((j,s)\)-tasks
        \( l = 1 \)
        \( \text{while} \ l \leq L \text{ and } \sum_{k=1}^{l-1} z^k_{j,s} < Q_{j,s}(t) + A_j(t) \text{ do} \)
        Allocate \( \sum_{k=1}^{l-1} z^k_{j,s} \) (\( j \), \( s \)) tasks to category \( l \). Here tasks \([x : x+y]\) are task set \( I = \{[x], \ldots [x+y]\} \) (in the ordering \( O_j \)), \([x] - x\) fraction of task \([x]\) and \( 1+x+y-[x+y] \) fraction of task \([x+y]\).
        \( l \leftarrow l + 1 \)
    end while
end for

for \( l = 1 : L \) do
    Order agents of category \( l \) arbitrarily
    for \( s = 1 : S \) do
        Agents pick maximum (as per availability constraint) tasks (or part) in order from \( \sum_j \min(z^l_{j,s}, Q_{j,s}(t) + A_j(t)) r_{j,s} \) hours
    end for
end for
4.4 Single-category Systems and Decentralized Allocations

There is effectively a single category of agents in many platforms with a large population of new freelancers, whose reputations are based on evaluation tests for skills and who are paid at a fixed rate. Hence designing efficient allocation schemes for single-category systems are of particular interest, as this population of agents are significant in ever-evolving crowd systems. Insights drawn from single-category systems are also useful in controlling multi-category systems, Sec. 4.5.

For a single category system \((L = 1)\), note that \(z^1_{j,s} = a_{j,s} \in \mathbb{Z}_+\) and hence the feasibility condition (4.1) reduces to:

\[
\sum_j a_{j,s} r_{j,s} \leq \sum_i u_i h_{i,s} \text{ for all } s \in [S], a_{j,s} \in \mathbb{Z}_+,
\]

with condition (4.2) additionally requiring \(a_{j,s} = a_{j,s'}\) for all \(j, s, s'\). Thus, \(C(u)\) is the set of \(\{a_{j,s}\}\) satisfying the above conditions for respective classes of jobs and \(\mathcal{C}(\Gamma)\) is the weighted (by \(\Gamma(u)\)) sum of convex hulls of \(C(u)\)s, here \(\mathcal{C}_\Gamma = \bar{\mathcal{C}}(\Gamma)\).

\(\mathcal{C}^\mu_{\text{out}}\) has a simple characterization as well. As for any \(j \in [N], (j, 1) \in E\), and \(\mathcal{N}(J) = 1\) for all \(J \subset [N]\), \(\sum_{l \in \mathcal{N}(J)} \sum_{i \in [M]} \mu_i h_{i,s} = \sum_{i \in [M]} \mu_i h_{i,s}\). Thus it is sufficient to satisfy the inequality for \(J = [N]\), and hence, \(\mathcal{C}^\mu_{\text{out}} = \{\lambda : \sum_{j \in [N]} \lambda_j r_j \leq \sum_{i \in [M]} \mu_i h_i\}\).

The MaxWeight computation in MWTA for single-category systems turns out to be the following integer linear program (ILP), which is related to
knapsack problems.

\[
\arg \max_{\{\Delta_{j,s}, j,s\}} \sum_{j,s} Q_{j,s} \Delta_{j,s}
\]  \hspace{1cm} (4.5)

\[\text{s.t.} \quad \sum_{j} \Delta_{j,s} r_{j,s} \leq \sum_{i} u_i h_{i,s} \forall s \in [S],\]

\[\Delta_{j,s} = \Delta_{j,s'}, \forall s, s', j \text{ (only for ND)}\]

For decomposable and non-decomposable systems, this is a single knapsack and multi-dimensional knapsack problem [87], respectively, and hence NP-hard. There exist fully polynomial time approximations (FPTAS) for single knapsack, whereas for multi-dimensional knapsack only polynomial time approximations (PTAS) are possible [87]. With this approximation, say \(1 - \epsilon\), the MWTA policy stabilizes \((1 - \epsilon) \hat{C}_\Gamma = \{\lambda : \frac{\lambda}{1-\epsilon} \in \hat{C}_\Gamma\}\). Also, note that for large crowd systems each \(\lambda_i\) is large and hence stabilizing any \(\lambda\) with \(\lambda + 1 \in \hat{C}_\Gamma\) is almost optimal. The above ILP can be relaxed to obtain a linear program, an allocation based on which achieves this approximation (see Appendix I.A for details).

4.4.1 Decentralized Allocations

Now we show that due to the structure of the crowd allocation problem and the fact that crowd systems are large, simple allocation schemes with minimal centralized control achieve good performance under mild assumptions on arrival and availability processes. Interestingly, though the centralized optimal allocation requires solving a knapsack problem at each epoch and greedy schemes are known to be sub-optimal for knapsack problems [87], we
propose two simple greedy schemes that are almost optimal with good performance guarantees. One of them, called GreedyAgent allocation provably performs well for decomposable systems and offers the freedom of selection to freelancers. Another, called GreedyJob allocation has provable performance guarantees for both decomposable and non-decomposable systems while allowing customers (job requesters) the freedom of selection. Thus, in some sense, this shows that though greedy algorithms can be suboptimal for an arbitrary allocation instant (at each epoch), for a dynamical system over long time, its performance is good.

Algorithm 5 GreedyAgent Allocation

Input: \( A(t) \)
Output: Job to agent allocations
\( A \): set of agents, \( T \): set of tasks

while \( A \) and \( T \) non-empty do
Agents in \( A \) contend (pick random numbers) and \( a \) wins
for each skill with non-zero skill hour do
\( a \) picks as many integral tasks as it can pick
if \( a \) has remaining available hour then
\( a \) Picks from remaining parts of the partially allocated task
if \( a \) has remaining available hour then
\( a \) picks part of any unallocated task
end if
end if
Remove fully allocated tasks from \( T \)
end for
\( A = A \setminus \{a\} \)
end while
Tasks with partial allocations are not allocated

In GreedyAgent allocation (Alg. 2), agents themselves figure out the
allocation via contention, without any central control. Agents need no knowledge about the agent population, but do need information on the available pool of jobs and have to agree on certain norms. In most freelance market platforms, this information is readily available, and so an algorithm like this is natural. As expected, this scheme may not be able to stabilize any arrival rate in $\hat{C}_t$ for any ergodic job-arrival and agent-availability processes, but it has good theoretical guarantees under some mild assumptions on the job arrival and agent availability processes.

**Definition 31.** A random variable $X$ is Gaussian-dominated if $E[X^2] \leq E[X]^2 + E[X]$ and for all $\theta \in \mathbb{R}$, $E[e^{\theta(X-E[X])}] \leq \exp\left\{\frac{(E[X^2] - E[X]^2)\theta^2}{2}\right\}$

**Definition 32.** A random variable $X$ is Poisson-dominated if for all $\theta \in \mathbb{R}$, $E[e^{\theta(X-E[X])}] \leq e^{E[X](e^\theta - \theta - 1)}$.

Note that these domination definitions imply that the variation of the random variable around its mean is dominated in a moment generating function sense by that of a Gaussian or Poisson random variable.$^3$ Such a property is satisfied by many distributions including Poisson and binomial that are used to model arrival processes for many systems, e.g., telephone networks, internet, call centers, and some freelance markets [59, 60]. It is not hard to show that sub-Gaussian distributions (standard in machine learning [89]) that are symmetric around their mean, are Gaussian-dominated.

$^3$Domination in this sense is used in bandit problems [88].
Theorem 33. If the arrival processes \( \{A_j(t)\} \) and the agent availability processes \( \{U_i(t)\} \) are i.i.d. across time and independent across types (jobs and agents) and all these processes are Gaussian-dominated (and/or Poisson-dominated), then for any given \( \alpha \in (0, 1] \), there exists an \( N_\alpha \) such that GreedyAgent allocation stabilizes any arrival rate \( \lambda \in (1 - \alpha)C^\text{out}_\mu := \left\{ \lambda : \frac{1}{1-\alpha} \lambda \in C^\text{out}_\mu \right\} \) for any single-category decomposable crowd system with \( N \geq N_\alpha \). Moreover, for any arrival rate in \( (1 - \alpha)C^\text{out}_\mu \), at steady state, after an allocation epoch, the number of unallocated tasks is \( O(S \log N) \) with probability \( 1 - o\left(\frac{1}{N^2}\right) \).

Proof. See Appendix C Sec. C.1.5.

As \( C_\Gamma \subseteq C^\text{out}_\mu \), this implies that the greedy scheme stabilizes an arbitrarily large fraction of the capacity region, under the assumptions on the arrival and availability processes. As \( S = o(N) \), more specifically \( O(N^c) \) for \( c < 1 \), the above bound on number of jobs imply that there are \( o(N) \) unallocated tasks at any time. This in turn implies that unallocated tasks per type (average across types) is \( o(1) \), i.e., vanishingly small number of tasks per type are unallocated as the system scales.
In GreedyAgent, there is no coordination among agents while picking tasks within jobs. Hence in a non-decomposable system, many tasks may be picked by agents but only few complete jobs are allocated. As more and more jobs accumulate, the chance of this happening increases, resulting in more accumulation. This can result in the number of accumulated jobs growing without bound, as formalized below.

**Proposition 34.** There exists a class of non-decomposable crowd systems with Poisson-dominated (as well as Gaussian-dominated) distributions of arrival and availability, such that the system is not stable under GreedyAgent allocation.

**Proof.** See Appendix C Sec. C.1.6.

Hence, we propose another simple greedy scheme that works for both decomposable and non-decomposable systems. The GreedyJob allocation scheme (Alg. 3) is completely distributed and hence a good fit for crowd systems. GreedyJob has similar performance guarantees for both decomposable and non-decomposable systems as GreedyAgent has for decomposable systems only.

**Theorem 35.** If the arrival processes \( \{A_j(t)\} \) and the agent availability processes \( \{U_i(t)\} \) are i.i.d. across time and independent across types (jobs and agents), all these processes are Gaussian-dominated (and/or Poisson-dominated) and \( \forall s, s' \), \( |\sum_i \mu_i h_{i,s} - \sum_i \mu_i h_{i,s'}| \) is \( O(\text{subpoly}(N)) \), then for any given \( \alpha \in \)
Algorithm 6 GreedyJob Allocation

Input: $U(t)$
Output: Job to agent allocations
$J$: set of all jobs

while Available skill-hours of agents and $J \neq \emptyset$ do
    Jobs in $J$ contend (pick random numbers) and $J$ wins
    if $J$ finds agents to allocate all tasks then
        Allocate to those agents
    else
        $J$ does not allocate anything
    end if
    $J = J \setminus \{J\}$
end while

$(0, 1], \exists N_\alpha$ such that GreedyJob allocation stabilizes any arrival rate $\lambda \in (1 - \alpha)C_{out}^\mu := \{\lambda : \frac{1}{1-\alpha}\lambda \in C_{\mu}^{out}\}$ for any single-category crowd-system with $N \geq N_\alpha$. Moreover, for any arrival rate in $(1 - \alpha)C_{\mu}^{out}$, at steady state, after an allocation epoch, total number of unallocated jobs (adding all types) is $O(\log N)$ with probability $1 - o\left(\frac{1}{N^\alpha}\right)$.

Proof. See Appendix C Sec. C.1.7.

In Sec. 4.5 we propose a decentralized scheme for multi-category systems that uses the two single-category decentralized schemes as building blocks. At the end of Sec. 4.5 we briefly discuss the suitability of these decentralized schemes for crowd systems in terms of implementability on crowd platforms.
4.5 Multi-Category Systems

Sec. 5.3.1 characterized the capacity region and developed an optimal centralized scheme for crowd systems, whereas Sec. 4.4 discussed simple decentralized schemes for single-category systems. Here we return to multi-category systems, briefly discussing computational aspects of MWTA, followed by an alternate approach to the capacity and outer region of inflexible systems that yields a simple optimal scheme. We also present a decentralized scheme based on insights from the optimal scheme and the decentralized allocations in Sec. 4.4.

The MWTA scheme, which is throughput optimal for FD, FND, and IND systems, involves solving an NP-hard problem for multi-category systems. For multi-category systems this is from the general class of packing integer programs, for which constant factor approximation algorithms exist under different assumptions on the problem parameters [90]. These assumptions do not generally hold for MaxWeight allocation under the CAC constraint. Rather, we follow the same steps of LP relaxation and obtain a scheme that stabilizes any $\lambda$ for $\lambda + 1 \in \tilde{C}_f$, since for large systems this is better than any arbitrarily close approximation scheme (as $\lambda \to \infty$ as $N \to \infty$).

4.5.1 Inflexible System

Below we present a characterization of the capacity region of inflexible systems in terms of the bipartite graph $G = (V,E)$, which captures the restriction of job-agent allocations.
Theorem 36. Any $\lambda$ can be stabilized if $\lambda \in \tilde{\mathcal{C}}^I$, where

$$\mathcal{C}^I = \left\{ \lambda = \sum_{l \in [L]} \lambda^{(l)} : \lambda^{(l)} \in \tilde{\mathcal{C}}^{(l)}_{\Gamma_l}, \lambda^l_j = 0 \text{ for all } (j, l) \notin E \right\},$$

where $\tilde{\mathcal{C}}^{(l)}_{\Gamma_l}$ is the capacity region of a single category system with an agent availability distribution $\Gamma_l = \Gamma \left( {U^l_i : i \in [M^l]} \right)$. Moreover, no $\lambda \notin \tilde{\mathcal{C}}^I$ can be stabilized, i.e., for inflexible systems the capacity region $\mathcal{C}_\Gamma = \tilde{\mathcal{C}}^I$.

Proof. See Appendix C Sec. C.1.8. \qed

This theorem has the following simple consequence. Consider separate pools of agents for each different category, cf. [61], which has agent-availability distributions $\{\Gamma_l : l \in [L]\}$. Each such pool (category) of agents $l$ can stabilize job-arrival rates in $\tilde{\mathcal{C}}^{(l)}_{\Gamma_l}$. Thus if the job arrival process of each job type $j$ can be split in such a way that pool (category) $l$ of agents sees an arrival rate $\lambda^l_j$, where $\lambda^l_j > 0$ only if $(j, l) \in E$, while ensuring that $\{\lambda^l_j : j\} \in \tilde{\mathcal{C}}^{(l)}_{\Gamma_l}$, the system would be stable.

In a server farm where jobs can be placed on any of the server queues, the join-shortest-queue (JSQ) policy stabilizes any stabilizable rate [74]. JSQ gives an arriving job to the server with the shortest queue and each server serves jobs in FIFO order. For multi-category crowd systems, we can draw a parallel between servers and agent pools. In addition we have constraints on job placement given by $G$ and also have to do allocations of jobs among the agents in the pool optimally (unlike JSQ we do not have FIFO/LIFO...
specified). Thus we have to adapt JSQ appropriately based on our insights about optimal operation of crowd systems.

We propose a statistics-agnostic scheme, JLTT-MWTA (Alg. 4) that has two parts: JLTT (join least total task) directs arrivals to appropriate pools of agents and MWTA allocates jobs in each pool separately. Letting $Q_{j,s}^l(t)$ be the number of unallocated $(j, s)$-tasks in $l$th pool just after epoch $t - 1$, JLTT uses these quantities to direct jobs to appropriate pools whereas MWTA uses them to allocate tasks within each pool.

**Algorithm 7** JLTT-MWTA: Divide and Allocate

Input: $A(t), U(t), Q(t)$
Output: Job division and allocation
Create pool $l$ with category $l$ agents ($\forall l \in [L]$)
JLTT: Join Least Total Task

for each $(j, s)$ do
    Count number of unallocated $(j, s)$-tasks in pool $l$: $Q_{j,s}^l(t)$
    Divide $A_j(t)1(r_{j,s} > 0)$ tasks equally among pools
    $\arg\min_{l, (j,l) \in E} \sum_s Q_{j,s}^l(t)$
end for
In each pool $l$ run MWTA for single-category system

The JLTT part is computationally light. The central controller only needs to know $Q(t)$ and has to pick the minimally loaded ($\min_l \sum_s Q_{j,s}^l$) pools of agents to direct jobs (type $j$). To perform MWTA in each pool, a PTAS, FPTAS, or LP relaxation scheme can be used.

Unlike JSQ, where service discipline in each server is fixed and the goal is to place the jobs optimally, we have jobs with multi-dimensional service requirements from time-varying stochastic servers (agent-availability) and have
to place jobs as well as discipline the service in each random and time-varying virtual pool. Thus optimality of JSQ cannot be claimed in our case. But as stated below, JLTT division followed by MWTA allocation is indeed optimal.

**Theorem 37.** JLTT-MWTA stabilizes any $\lambda \in \hat{E}^I$.

**Proof.** See Appendix C Sec. C.1.9.

An important aspect of JLTT-MWTA is that job allocations within each pool can happen independently of each other. The central controller only has to make a decision on how to split the jobs based on the current system state information. This allows a more distributed allocation along the lines of following hierarchical organizational structure [91]. First, the central controller divides jobs for different agent-pools based on $\{Q^I_{j,s}\}$. Then in each agent pool, allocations are according to GreedyJob allocation, which works for both decomposable and non-decomposable single-category systems. The distributed scheme that we propose here is an improvisation of the above JLTT scheme followed by GreedyJob allocation in each pool. We call it Improvised JLTT and GreedyJob Allocation (Alg. 5).

First note that unlike JLTT-MWTA, here we only maintain number of unallocated jobs and do not maintain number of unallocated tasks for each skill $s$. This is because as GreedyJob allocation is used as allocation scheme in each pool, $Q^I_{j,s} = Q^I_{j,s'}$ for all $s, s'$.

This algorithm is also simple to implement. The central controller only needs to track the number of unallocated jobs ($Q^I_j(t)$) from the previous epoch.
Algorithm 8 Improvised JLTT and GreedyJob Allocation

Input: \( A(t), U(t), Q(t) \)
Output: Job division and allocation

**Improvised JSQ for each job-type \( j \) and each skill \( s \):**

\[
n = 0
\]
\[
N^l_j = Q^l_j(t)
\]

While \( n < A_j(t) \)

- Send 1 task to the category \( l^* \) with lowest
- index among \( \arg \min_{(l,j) \in E} N^l_j \)
- Increase \( N^l_j \) and \( n \) each by 1

End While

**Allocations within each pool \( l \):**

- Run GreedyJob allocation

and set \( N^l_j = Q^l_j \). For any arriving job of type \( j \), the central controller sends the job to the pool with minimum \( N^l_j \) and updates \( N^l_j \). This continues until the next epoch, when the \( N^l_j \) are reset to new \( Q^l_j \) values.

Recall that Sec. 5.1 gave a simple example of a fully distributed scheme where jobs pick agents greedily (from the set of feasible agents as per \( G \)) and showed it was not a good scheme. Improvisation of JLTT is proposed for a better performance guarantee, while GreedyJob in each pool is proposed for implementability and freedom of selection for customers. It is not hard to prove Improvised JLTT followed by MWTA is optimal for any arrival and availability process satisfying the assumptions of Sec. 5.2. Below we present performance guarantee for Improvised JLTT and GreedyJob allocation.

To present performance guarantees of the distributed scheme we give an outer region \( \mathcal{C}^O \) for the system, along the lines of the alternative characterization \( \mathcal{C}^I \) of the capacity region \( \mathcal{C}_r \) for inflexible systems.
Theorem 38. Inflexible crowd systems cannot be stabilized for \( \lambda \notin \mathcal{C}^O \), where 
\[
\mathcal{C}^O = \left\{ \lambda : \lambda = \sum_{l \in [L]} \lambda^{(l)} \text{ where } \lambda^{(l)} \in \mathcal{C}^{\text{out}}_{\mu^l} \right\},
\] and \( \mathcal{C}^{\text{out}}_{\mu^l} \) is the outer region for the single category system comprising the \( l \)th category (pool) of agents with \( \mu^l = E[U^l] \).

Proof. See Appendix C Sec. C.1.10.

For job allocation in server farms, extant performance guarantees are mostly for symmetric load, i.e., symmetric (almost) service and job arrival rates and regular graphs, cf. [84, 85]. Unlike server farms, symmetric load (in terms of skill-hours) is not guaranteed in crowd systems by symmetric arrival rates and graphs. This is because different types of jobs have different skill and hour requirements. The following guarantee for crowd systems is for bounded asymmetry (sub-polynomial variation) in agent availability, complete graph, asymmetric job arrival rates, and asymmetric job requirements (extendable to regular graphs with additional assumptions on symmetry of job arrival rates and requirements). Note that because of the inflexibility constraint, a multi-category system with a complete graph is not equivalent to a single-category system.

Theorem 39. Without loss of generality assume the same ordering of agent types in each category, i.e., \( M^l = M^{l'} = M/L \) and \( h^l_i = h^{l'}_i \) for all \( l, l', i \). If the arrival processes \( \{A_j(t)\} \) and the agent availability processes \( \{U^l_i(t)\} \) are i.i.d. across time and independent across types (jobs and agents), all these processes are Gaussian-dominated (and/or Poisson-dominated), 
\[
\sum_i \max_{l,l'} |\mu^l_i - \mu^{l'}_i| \text{ and }
\]
\[
\max_{l,s,s'} | \sum_i \mu^l_i (h^l_{i,s} - h^l_{i,s'}) | \text{ are } O(\text{subpoly}(N)) \text{ and } G \text{ is complete bipartite, then for any given } \alpha \in (0, 1], \exists N_\alpha \text{ such that Improvised JLTT and Greedy-job stabilizes any } \lambda \in (1 - \alpha) \mathcal{C}^O := \{ \lambda : \frac{1}{1-\alpha} \lambda \in \mathcal{C}^O \} \text{ and the maximum number of unallocated jobs (across all types) is } O(\log N) \text{ with probability } 1 - o\left(\frac{1}{N^2}\right).}
\]

Proof. See Appendix C C.1.11.

The proofs of Thm. 33, 35, and 39 are all based on constructing queue-processes (different for the algorithms) that stochastically dominate the number of unallocated jobs, and bounding the steady state distributions of these processes using Loynes’ construction and moment generating function techniques.

4.5.2 Implementation of Decentralized Schemes on Crowd Platforms

We have described the allocation schemes at the level of system abstraction and discussed their performance. These schemes can be easily implemented on crowd platforms as well.

GreedyAgent allocation is completely decentralized, only requiring agents to abide by a norm for picking partial tasks, which can be enforced by randomized vigilance and penalizing norm violators in reputation. If the payments are the same, as it generally is in single-category systems where all jobs require same quality, there is no incentive for agents to deviate from the norm.
Any arbitrary contention method among agents will work for the algorithm, and hence the crowdsourcing platform only needs to ensure that no two allocations are done simultaneously (as practiced in airlines booking). Multiple allocations can also be allowed by the platform if they do not conflict. Here the platform has to ensure that an agent can place requests only for an amount of tasks it can actually perform, given the constraints on available hours. Also, only one agent can request for a task or a certain part of it. Once the agent has been declined, it can place request(s) for task(s) of the same or lesser hours. This can either be enforced by appropriate modification of the portals by keeping tracks of total hours of requests placed or by vigilance.

GreedyJob can also be easily implemented on a crowd platform. The platform has to ensure that jobs request agents and not the other way around. One way to implement this is to allow jobs to place requests for agents while ensuring they do not request more than the required service. Also, the platform has to ensure that skill-hour requests of no two jobs collide. This again can be ensured by serializing the requests as above. Agents are expected to accept a requested task, as there is no difference between tasks involving same skill since payments are the same. This can also be ensured by linking agent rating to rate of task-request acceptance.

It is apparent that GreedyJob offers choice to customers and GreedyAgent offers choice to agents. By allowing a customer (or an agent) to decline an approaching agent (or a customer) request and to explore more options, only one option at a time, the platform can provide freedom of choice to agents and
customers under both schemes while operating at capacity.

In case of multi-category systems, the platform only needs to direct arriving jobs to the appropriate pool of agents, based on current backlog; the rest of the allocation happens as per GreedyJob. Directing a job to a category of agents can be implemented in a crowd platform by making the job visible only to freelancers of that category and vice versa (similar to filtering done by search engines and online social networks) or through explicit hierarchical organization into pools [91].

4.6 Beyond the Capacity Region

We have now characterized the capacity (and outer) regions of different classes of crowd systems, shown the existence of computationally feasible centralized schemes that achieve these regions, and presented simple distributed schemes with minimal centralized intervention and good performance guarantees for any arrival rate within the capacity region. In crowd systems, however, arrival rates may not be within the capacity region, since the platform may have little or no control on resource (freelancer) planning, unlike traditional communication networks or cloud computing systems. Hence, an important aspect of crowd systems is to turn down job requests. Indeed, deciding to decline a job must be done as soon as the job arrives because dropping a job after first being accepted adversely affects the reputation of the crowd platform.

Here, we propose a centralized scheme for a crowd system to decline jobs on arrival in a way that is fair across all job types. Our scheme is statistics-
agnostic and works even for independent but non-stationary arrival and availability processes.

We solve the following problem: given an arrival rate $\lambda$, design a statistics-agnostic policy to accept $(1 - \beta)\lambda$ jobs on average and allocate them appropriately such that $\beta$ is the minimum for which the crowd system is stable. Note that if $\lambda \in \bar{\mathcal{C}}$, the minimum $\beta$ is 0, else, it is strictly positive. We want to design a statistics-agnostic policy without the knowledge of $\lambda$ and $\mathcal{C}_\Gamma$. As a benchmark, we consider the following problem for $\epsilon > 0$, when $\lambda$ and $\mathcal{C}_\Gamma$ are known.

$$\min \beta \in [0, 1] \quad \text{s.t.} \quad (1 - \beta)\lambda + \epsilon 1 \in \mathcal{C}_\Gamma. \quad (4.6)$$

Given $\beta^*$, optimum of $(4.6)$, $(1 - \beta^*)\lambda$ is within $\epsilon$ of the optimal rate of accepted jobs for which the system is stabilizable.

As we want a scheme that is agnostic of $\lambda$ and $\mathcal{C}_\Gamma$, we propose the following simple scheme, for $\nu > 0$ and $\bar{Q}_{j,s}(t)$ is the number of unallocated accepted $(j, s)$-tasks ($(j, s)$-tasks directed to category $l$) in the system.

$$\beta(t) = \arg \min_{\beta \in [0,1]} \min \begin{cases} 
\beta \sum_j A_j(t) - \nu \beta \sum_{j,s:t_j,s > 0} \bar{Q}_{j,s}(t) A_j(t) & \text{(I)} \\
\beta \sum_j A_j(t) - \nu \beta \sum_{j,s:t_j,s > 0} \min_l \bar{Q}^l_{j,s}(t) A_j(t) & \text{(II)} 
\end{cases}$$

Job is accepted w.p. $1 - \beta(t)$, accepted jobs $\tilde{\mathbf{A}}(t)$ (I & II)

For accepted jobs run

$$\begin{cases} 
\text{MWTA} & \text{(I)} \\
\text{JLT-MWTA} & \text{(II)} 
\end{cases} \quad (4.7)$$

Steps marked by I (II) are applicable for FD, FND and IND (ID and IND) systems.
Theorem 40. Crowd system with jobs accepted and allocated according to (4.7) is stable and \( \sum_j \lambda_j (1 - \beta^*) - \frac{1}{T} \sum_{t=1}^{T} E[\sum_j \tilde{A}_j(t)] \) can be made arbitrarily small for an appropriately chosen \( \nu \), for all sufficiently large \( T \).

Proof. See Appendix C Sec. C.1.12.

This theorem demonstrates that by following the job acceptance and allocation method (4.7), the crowd system can be stabilized while ensuring the average number of accepted jobs per allocation epoch is arbitrarily close to the optimal number of accepted jobs per allocation period. Note that as all jobs (across all types) are accepted with the same probability, the above result also implies that \( (1 - \beta^*)\lambda_j - \frac{1}{T} \sum_{t=1}^{T} E[\tilde{A}_j(t)] \) is small. It can be shown that the above scheme works for time-varying systems (\( E[A(t)] = \lambda(t) \) and \( U \sim \Gamma(t) \)) as well guaranteeing small \( \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \sum_j \left( \lambda_j(t)(1 - \beta^*(t)) - E[\tilde{A}_j(t)] \right) \), where \( \beta^*(t) \) is the solution of (4.6) for \( \lambda = \lambda(t) \) and \( c_T = c_{T^*} \).

4.7 Conclusion

Human information processing, structured through freelance markets, is an emerging structure for performing informational work by capturing the cognitive energy of the crowd. It is important to understand the fundamental limits and optimal designs for such systems.

In this work we provide a characterization of the work capacity of crowd systems and present two statistic-agnostic job allocation schemes MWTA (flexible jobs) and JLTT-MWTA (inflexible jobs) to achieve limits. To ensure low
computational load on the crowd platform provider and freedom of choice for job requesters, we present simple decentralized schemes, GreedyAgent, GreedyJob, and Improvised JLT-T-GreedyJob that (almost) achieve capacity with certain performance guarantees. These decentralized schemes are easy to implement on crowd platforms, require minimal centralized control, and offer freedom of self-selection to customers: all desirable qualities for any crowd platform. Due to quick evolution and unpredictability of freelancer resources, crowd systems may often operate outside capacity, which inevitably results in huge backlogs. Backlogs hurt the reputation of the platform, and so we also propose a scheme that judiciously accepts or rejects jobs based on the system load. This scheme is fair in accepting jobs across all types and accepts the maximum number of jobs under which the system can be stable.
Chapter 5

Crowdsourcing with Ordering Constraints

5.1 Introduction

In Chapter 4 we studied crowdsourcing of jobs which require one or multiple skills. We considered the case where each job has multiple parts (tasks) requiring different skills which may (non-decomposable) or may not (decomposable) require coordination between them for successful completion. In the case of decomposable jobs the tasks have no coordination (and hence allocation) constraints between them, whereas for non-decomposable jobs a strict coordination is required. There is another kind of job which is frequently seen in many crowdsourcing platforms. These jobs have multiple steps, each of which is composed of multiple tasks, i.e., each step may require multiple skills, and there is a precedence constraint between the steps of a job in which they ought to be served.

A simple example of such a job is a software/IT development job. Any software/IT product first has to be planned (architecture), then has to be developed (programming) and next has to be tested (testing and quality assurance). In IBM’s Application Assembly Optimization platform such jobs are frequently encountered, and IBM often has to crowdsource these jobs to
its internal employee crowd [92, 93]. In skilled microtasking platforms like SamaHub (operated by SamaSource, a not for profit company which aims to improve lives in developing and third world countries [94]) a majority of jobs have more than one steps. In this chapter we study operating limits and algorithms for crowdsourcing platforms with such jobs.

A job with multiple steps and with a service-order between them poses a new allocation constraint that has not been considered in Chapter 4. In addition, as each step is composed of multiple tasks there may be allocation constraint within each step as well. For example, depending on whether a step is decomposable or not (like decomposable or non-decomposable jobs in Chapter 4) the allocation constraint for a step may change. We can draw a parallel with the scenario in Chapter 4. A step here is similar to a job there and it has multiple parts each involving a different skill (we call it task here). A job here is a batch of multiple steps which have a pre-defined service order or precedence constraint between them.

Job allocation with precedence constraints has been studied in theoretical computer science, as follows. Given several tasks, precedence constraints among them, and one or more machines (either same or different speed), allocate tasks to minimize the weighted sum of completion times or maximum completion time [95]. In crowdsourcing, we have a stream of tasks arriving over time and so we are interested in dynamics.

Dynamic task allocation with precedence constraints has recently been studied in [96] for Bernoulli task arrivals. The setting in [96] is different from
crowdsourcing scenarios, and the optimal scheme is required to search over the set of possible allocations, which is not suitable for crowdsourcing systems due to their inherent high-dimensionality (many types of tasks). Additional challenges in a crowdsourcing platform are: (i) random and time-varying agent availability; (ii) vector-valued service requirements; (iii) fast computation for scaling; and (iv) freedom of choice for customers.

In this work we consider a crowdsourcing system, where a job has one or more steps that have a precedence constraint among them. We characterize the capacity region of such systems and propose a centralized allocation scheme that can be efficiently approximated. Next, for better suitability for crowdsourcing platforms, we propose a fast decentralized scheme. This has minimal centralized intervention and has good performance guarantees for a broad class of systems.

5.2 System Model

The system model here is similar to that in Chap. 4, but there are some importance differences as well. Hence, for clarity, we redefine the system model.

In the crowdsourcing system there are total $S$ kinds of skills available. There are different types of freelancers or agents depending on their available skills and duration of availability. In this chapter we assume that there is only one category of agents. We denote the total number of types of agents by $M$. For an agent of type $m$, we have a vector $h_m = (h_{m,1}, h_{m,2}, \cdots, h_{m,S})$ which
represents the duration of availability of agents for work involving different skills. For each type $m$, $h_m$ is a sparse vector, as an agent has only few of the $S$ skills.

Jobs posted on the platform are of $N$ types. Each type of job $j$ has one or multiple steps associated with it which we denote by $K_j$. A step $k \in \{1, 2, \cdots, K_j\}$ of a job type $j$ which we refer to as $(j, k)$-step needs a skill-hour service $r_{j,k} \in \mathbb{R}_+^S$, i.e. $r_{j,k,s}$ hours of skill $s$. A part of a step of type $(j, k)$ involving skill $s$ is called a $(j, k, s)$-task if $r_{j,k,s} > 0$, which is the size of this task.

In the platform, jobs and/or parts of them are allocated at regular intervals of time to available agents, these epochs are denoted by $t \in \{1, 2, \ldots\}$. Jobs that arrive after an epoch $t$ are considered for allocation at epoch $t + 1$, based on the available agents at that epoch. Unallocated jobs (due to insufficient number of skilled agents) are considered again in the next epoch. We assume that for any task $(j, k, s)$, the size or the time requirement is less than the duration between two scheduling epochs.

Jobs arrive according to a $\mathbb{Z}_+^N$-valued stochastic process $\mathbf{A}(t) = (A_1(t), A_2(t), \ldots, A_N(t))$, where $A_i(t)$ is the number of jobs of type $i$ that arrive between epochs $t - 1$ and $t$.

The stochastic process of available agents at epoch $t$ is $\mathbf{U}(t) = (U_1(t), U_2(t), \ldots, U_M(t))$.

We assume that the processes $\mathbf{A}(t)$ and $\mathbf{U}(t)$ are independent of each
other and that each of these processes is independent and identically distributed for each \( t \). Let \( \Gamma(\cdot) \) be the distribution of \( U(t) \), and let \( \lambda = E[A(t)] \) and \( \mu = E[U(t)] \) be the means of the processes. We also assume that these processes have bounded second moments.

At any epoch \( t \) only an integral allocation of a task (say \((j,k,s)\)) is possible. A set of tasks \( t_1, t_2, \ldots, t_n \) of size \( r_1, r_2, \ldots, r_n \) of skill \( s \) can be allocated to agents \( 1, 2, \ldots, k \) only if available skill-hours for skill \( s \) of these agents \( h_1, h_2, \ldots, h_m \) satisfy

\[
\sum_{p=1}^{n} v_{ip} \leq h_i, \quad \sum_{q=1}^{m} v_{qj} \geq t_j, \quad j \in [n], \quad i \in [m], \quad \{v_{pq} \geq 0\}.
\]

Whether different tasks of a \((j,k)\)-step can be allocated at different epochs depend on the type of the step.

**Definition 41.** A \((j,k)\)-step is called non-decomposable (decomposable) if different tasks comprising it are (are not) constrained to be allocated at the same epoch.

In addition, there is a constraint on the order in which different steps of a job of type \( j \) can be served. For any job of type \( j \) this constraint is given by a directed rooted tree \( T_j \) on \( K_j \) nodes where a directed edge \((k \to k')\), \( k, k' \in [K_j] \) implies that the step \( k' \) of a job of type \( j \) can only be served after the step \( k \) of the same job has been completed.

This system also has the same scaling as in Chap 4, i.e., it follows crowd-scaling.
5.3 System Limits and Algorithms

In this section we characterize the limits of jobs allocation in the system described in Sec. 5.2. We formally characterize the maximal supportable arrival rate of jobs in terms of available resource (agents and hours), design a centralized scheme that achieves it and propose a simple and fast decentralized scheme which is easily implementable on a crowdsourcing platform.

5.3.1 Capacity and Outer Region

We characterize this system based on the notion of stability and capacity region defined in Chapter 4. In this chapter we only discuss the system with non-decomposable steps.

For any given set of available agents \( u = (u_i : 1 \leq i \leq M) \), we define the number of different types of steps \( \{a_{j,k}\} \) that can potentially be allocated in a crowd system as \( C(u) \subset \mathbb{R}^{\sum_{j} K_j} \).

When we say \( \{a_{j,k}\} \) is the number of steps of different types that can potentially be allocated, we consider allocation under the following scenario.

- An infinite number of steps of each type \((j,k), k \in [K_j]\) for a \( j \in [N] \) are available for allocation, i.e., the limitation only comes from the available resource \( u \).

- Precedence constraints among the steps are already satisfied, i.e., all corresponding \((j,k)\)-steps of the available \((j,k')\)-steps have already been allocated. This is equivalent to an absence of any precedence constraint.
For an allocation of $a_{j,k}$ steps of different types to a collection of $R$ agents of type $\{m_r : r \in [R]\}$ and available hours $\{h_{m_r,s} : r \in [R]\}$ Eq. 5.1 must be satisfied, which can also be written as

$$\sum_{j,k} a_{j,k} r_{j,k,s} \leq \sum_{r=1}^{R} h_{m_r,s} \forall s \in [S]. \quad (5.2)$$

Let $\text{conv}C(u)$ be the convex hull of the set $C(u)$. We define another set $C \subset \mathbb{R}_+^{\sum_j K_j}$ as follows.

$$C = \{\sum_u \Gamma(u) a(u) : a(u) \in \text{conv}C(u)\}$$

Based on this we define another set $\mathcal{C} \subset \mathbb{R}_+^N$. Let for any $a \in \mathbb{R}_+^N$, $a^E := (a_1, a_1, \ldots, K_1\text{times}, a_2, a_2, \ldots, K_2\text{times}, \ldots, a_N, a_N, \ldots, K_N\text{times})$.

$$\mathcal{C} = \{a : a^E \in C\}$$

This set characterizes the capacity region of the crowdsourcing system, as stated in the following theorem.

**Theorem 42.** Any arrival rate $\lambda$ is stabilizable if for some $\epsilon > 0 \lambda + \epsilon 1 \in \mathcal{C}$ and no arrival rate $\lambda$ can be stabilized if $\lambda$ is outside the closure of the set $\mathcal{C}$.

Note that in defining $\mathcal{C}$ we ignored precedence constraint. This does not conflict with the fact that the capacity region is a subset of $\mathcal{C}$, but it may not be obvious that $\mathcal{C}$ is in fact the capacity region. We show that with a
scheme that respects the precedence constraints and stabilizes any rate in the interior of \( \mathcal{C} \). \(^1\)

The region \( \mathcal{C} \) depends on the distribution of the availability of the agents. In practice, for a large system it is hard to obtain a high dimensional distribution. Hence, a characterization of the capacity region in terms of first order statistics is useful.

Consider the set \( \mathcal{C}^O \subset \mathbb{R}^N_+ \) defined as follows.

\[
\mathcal{C}^O := \{ \mathbf{a} \in \mathbb{R}^N_+ : \sum_{j,k} a_{j,k,r_{j,k,s}} \leq \sum_m \mu_m h_{m,s} \ \forall s \in [S] \}.
\]

This set is a super-set of the capacity region, or in other words, it is an outer bound of \( \mathcal{C} \).

**Theorem 43.** \( \mathcal{C} \subset \mathcal{C}^O \).

#### 5.3.2 Centralized Allocation

Next, we propose a scheme that stabilizes any arrival rate \( \lambda \) in the interior of \( \mathcal{C} \) and is agnostic of the statistics of the system.

Let \( Q_{j,k}(t) \) be the number of unallocated \((j,k)\) steps just before the allocation epoch \( t \). This includes the number of steps that were not allocated at epoch \( t - 1 \) and the steps that became available for allocation between \( t - 1 \)

\(^1\)Using this approach and the fact that for any directed acyclic precedence graph, there exists a directed rooted tree that comply with the same precedence constraint, our result extends to arbitrary precedence graphs.
and $t$. Thus, if for any $(j, k)$, $D_{j,k}(t)$ $(j, k)$-steps were allocated at epoch $t$ and $A_{j,k}(t + 1)$ new $(j, k)$-steps became available between $t$ and $t + 1$ then,

$$Q_{j,k}(t + 1) = Q_{j,k}(t) - D_{j,k}(t) + A_{j,k}(t + 1).$$

Note that, for any $j$ and $K_j \geq k' > 1$, new $(j, k')$-steps become available only when some $(j, k)$ steps have been completed ($k \rightarrow k'$ in $T_j$). Service times \{$r_{j,k,s}\}$ are strictly less than the duration between two scheduling epochs. So, any step allocated at epoch $t$ is completed before epoch $t + 1$. Hence, for any $j$ and $K_j \geq k' > 1$ and $k \rightarrow k'$ in $T_j$,

$$A_{j,k'}(t + 1) = D_{j,k}(t).$$

On the other hand, for any $j$ and $k = 1$, we have an external arrival $A_j(t + 1)$ between epoch $t$ and $t + 1$.

At any time $t$, for a given resource availability, an allocation rule determines the resources to be allocate for certain number of $(j, k)$-steps. We denote that by $S_{j,k}(t)$. Note that $D_{j,k}(t) = \min(Q_{j,k}(t), S_{j,k}(t))$.

Our goal is to design a scheme that finds a good $\{S_{j,k}(t)\}$ for a given $\{Q_{j,k}(t)\}$ and $U(t) = u$. We propose a centralized allocation procedure for this system.

Centralized Allocation: Inflexible Agents
Input: \( \{Q_{j,s}(t) : j \in [N], s \in [S]\} \) and \( U(t) \) at \( t \)

Output: \( \{S^*_{j,k}(t)\} \) and allocation of steps to agents

Define: \( l_{j,r} \) : number of leaves in the subtree of \( T_j \) rooted at \( r \)

Find \( \{S^*_{j,k}(t)\} \):

\[
\begin{align*}
\{S^*_{j,k}(t)\} &= \\
&= \arg \max_{s_{j,k} \in \mathbb{Z}^+} \sum_{j} \sum_{k=1}^{K_j} \sum_{r : k \rightarrow r \in T_j} s_{j,k} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \\
&\text{s.t. } \{s_{j,k}\} \in C(U(t)) \quad (5.3)
\end{align*}
\]

For each \((j,k)\) allocate \( S^*_{j,k}(t) \) \((j,k)\)-steps

This allocation scheme is statistic agnostic and explicitly computable. It can be shown that this scheme is also optimal, in the sense that any arrival rate that can possibly be stabilized by any policy can also be stabilized by this scheme.

**Theorem 44.** *Centralized allocation procedure stabilizes any \( \lambda \) if \( \lambda + \epsilon \mathbf{1} \in \mathbb{C} \) for some \( \epsilon > 0 \).*

Though the centralized scheme is computable, it cannot always be computed efficiently in polynomial time (in the size of the system). As any allocation in \( C(u) \) has to satisfy the allocation constraint in Eq. 5.2, the optimization problem in Eq. 5.3 can be written as

\[
\begin{align*}
\max_{s_{j,k} \in \mathbb{Z}^+} \sum_{j} \sum_{k=1}^{K_j} w_{j,k} s_{j,k} \text{ s.t.}
\end{align*}
\]
\[
\sum_{j,k} s_{j,k} r_{j,k,s} \leq \sum_{m} u_{m} h_{m,s} \forall s \in [S]. \tag{5.4}
\]

Note that the solution to the problem does not change if we replace \(w_{j,k}\) by \(\max(w_{j,k}, 0)\), as the optimal scheme never allocates any resource to a negative \(w_{j,k}\). Thus, we can assume \(w_{j,k} \geq 0\).

It is not hard to see that the problem in Eq. 5.4 is related to the multi-dimensional knapsack problem [87]. In fact, this problem is a well known modification of the problem called the unbounded multi-dimensional knapsack problem, where the number of available items of a given weight and value are unbounded. This problem is known to be NP-hard and without any FPTAS [97]. PTAS is known for this problem but the complexity is exponential in dimension [97]. Recently extended LP relaxations have been proposed, but they also have the same issues (see [97] and references therein).

We propose the following LP-based scheme that gives a close to optimal centralized allocation for a large crowd system (under crowd scaling). We propose the following LP relaxation based polynomial time (in \(N\) and \(M\)) scheme.

**LP Relaxation**

\[
\{\hat{S}_{j,k}(t)\} = \\
\text{arg max}_{s_{j,k} \in \mathbb{K}_+} \sum_{j} \sum_{k=1}^{K_j} \sum_{r: k \rightarrow r \in T_j} s_{j,k} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \\\n\text{s.t.} \sum_{j,k} s_{j,k} r_{j,k,s} \leq \sum_{m} u_{m} h_{m,s} \forall s \in [S], \tag{5.5}
\]

101
Allocate \( S_{j,k}^r(t) = \lfloor \hat{S}_{j,k}(t) \rfloor \)

For this scheme we cannot give performance guarantee at each allocation epoch for any arbitrary \( Q_{j,k} \). But for a large enough crowd system this scheme stabilizes almost any arrival rate that can be stabilized.

**Theorem 45.** Under crowd scaling, for any \( \alpha \leq 1 \) \( \exists N_0 \) s.t. for any system with \( N \geq N_0 \) the LP-based scheme in Eq. 5.5 stabilizes any arrival rate in \( \alpha \mathcal{C} = \{ a : \frac{a}{\alpha} \in \mathcal{C} \} \).

Obtaining an efficient centralized allocation is not our main goal, we want to obtain a simple and fast distributed scheme.

### 5.3.3 Decentralized Allocation

In this section we discuss a simple decentralized scheme that has good performance guarantees. As discussed before, often one of the main incentives of the customers to go to a crowd platform is to be able to choose workers themselves. Motivated by this we propose a simple greedy scheme that allows the customers the freedom of choice with minimal intervention from the platform operators. This is good for the platform as well, as it reduces its operational cost.

Prioritized Greedy algorithm does a greedy allocation among all the steps across all types of jobs that are at the same level/depth of the respective precedence trees. It starts with the steps that are at the depth 0 of the precedence tree. Once these steps find an allocation (or cannot be allocated
Algorithm 9 Prioritized Greedy

Input: \( \{Q_{j,s}(t) : j \in [N], s \in [S]\} \) and \( U(t) \) at \( t \)
Output: Allocation of steps to agents
Define \( D = \max_j \text{ depth of } T_j \)
\[ S_j = \forall j \in [N] \]
for \( d=1:K \) do
\[ S_j = \{k_j : \text{depth of } k_j \text{ in } T_j \text{ is } d\} \]
Greedy allocation among \( \cup_j \{j, k_j : k_j \in S_j\} \) steps
end for

due to resource scarcity), only then the steps at the next level/depth in the corresponding precedence trees are allowed to allocate themselves.

In greedy allocation, each step compete against others and finds an allocation for all tasks of it. The contentions can be arbitrary, e.g., random, pre-ordered, age-based.

The decentralized greedy algorithm can be efficiently implemented on a crowdsourcing platform with minimal intervention from the platform operator. The operator only has to tag the unallocated steps in the system based on their depth in the rooted precedence tree and show available workers to them only after the steps at a lesser depth have exercised their allocation choice. This can be implemented by introducing filtering in the search engine of the platform.

In addition, this algorithm is fast and has good performance guarantees under the same mild assumptions as in Chapter 4.

Theorem 46. Given, for any \( s, s' \mid \sum_m \mu_m h_{m,s} - \sum_m \mu_m h_{m,s'} \) is sub-poly(\( N \)), i.e., \( o(N^\delta), \forall \delta > 0, \) for any \( \alpha \in (0,1), \exists N_\alpha \) s.t. \( \forall N \geq N_\alpha, \) for any crowdsourcing system of size \( N \) (that follows crowd scaling) and arrival and availability

103
processes being Poisson-dominated (and/or Gaussian-dominated), any arrival rate $\lambda \in \alpha \mathcal{C}$ can be stabilized by Prioritized Greedy and at the steady state the total number of unallocated steps in the system across all types is $O(\log N)$ w.p. $1 - O\left(\frac{1}{N^2}\right)$.

This implies that Prioritized Greedy can stabilize almost any stabilizable arrival rate and ensures that the number of unallocated jobs is much smaller than the system size.

### 5.4 Conclusion

In this chapter we considered allocation of jobs with multiple steps and precedence constraints on a crowdsourcing platform. We characterized the capacity region of such a system and proposed an optimal centralized scheme. We also proposed a decentralized greedy scheme that offers freedom of choice to the customers and has good performance guarantees.
Chapter 6

Conclusion

Study of dynamics in social networks and study of resource allocation in social networks are two important areas of social networks research. This thesis investigates into both of these areas as well as into their intersection. We study opinion dynamics which is central to the study of dynamics in social networks. Under resource allocation in social networks we study optimal job-to-agent allocation for social productions in a crowdsourcing setting. Our work on inference in social networks lies at the intersection of these two important areas. It builds on the understanding of the social dynamics, but its main application lies in optimal resource utilization in social network based campaigning.

First, in Chapter 2, we studied opinion dynamics in social networks. We proposed a generic model of opinion evolution that combines two broad class of dynamics, namely bounded confidence and graph-based linear dynamics into a single framework. Our proposed model captures stochastic opinion dependent opinion exchange in a social network. We analytically studied the conditions under which the opinion dynamics reach a weak consensus and observed that these conditions relate well to observed behaviors in societies.
Next, in Chapter 3, motivated by personalized or targeted campaigning, we studied inference in social networks from interaction data. The problem is founded on the fact that people with close opinions are more likely to accept or agree with each others’ opinions. We considered a generic model of opinion dependent acceptance/agreements and proposed algorithms with provable guarantees, and low computational and sample complexity.

Towards the end of this thesis, in Chapters 4 and 5, we studied job-to-agent allocation in skill-based crowdsourcing platforms. We characterized the limits of these systems and proposed centralized algorithms that achieve them. We also designed simple decentralized schemes with low complexity, easy implementability on crowdsourcing platforms, and good performance guarantees.
Appendices
Appendix A

A.1 Proofs

Before presenting the proofs of the main theorems stated in Section 2.3, we recall notation.

We denote the opinion dynamics process by $X(t) = \{X_i(t) : 1 \leq i \leq n\}$, with $x(t) = \{x_i(t) : 1 \leq i \leq n\}$ is a sample path realization of the process.

Under the assumptions discussed in Section 2.3, $X(t)$ is an irreducible discrete-time Markov chain in $\mathbb{Z}^n$.

Below, we focus on the process of opinion differences, $Y(t) = \{Y_{ij}(t) = X_j(t) - X_i(t), 1 \leq i \neq j \leq n\}$. It turns out that under the assumptions of Section 2.3, $Y(t)$ is also a discrete-time Markov chain. This follows from the update equations (Equations 2.1 and 2.2), from which we get that, for all possible values of $U_{i,j}^t$ and $U_{j,i}^t$, the update of $x_j(t) - x_i(t)$ only depends on $x_j(t) - x_i(t)$ (and not on the individual variables $x_i(t)$ and $x_j(t)$) and from the assumption that $(U_{i,j}^t, U_{j,i}^t)$ has a law that only depends on $|x_j(t) - x_i(t)|$ (through $f_{i,j}$ and $f_{j,i}$).

In fact, though $Y(t)$ is a Markov chain, we use different transformations of this Markov chain that are more convenient. For example, a more convenient chain is $Y_1(t) = \{Y_{ij}(t) = X_j(t) - X_i(t) : 2 \leq j \leq K\}$. From $Y_1(t)$, one can
reconstruct the whole $Y(t)$.

A.1.1 Proof of Theorem 2

The proof is based on the analysis of the process $Y(t) = Y_{12}(t) = -Y_{21}(t)$, which is a Markov chain on $\mathbb{Z}$ that describes $Y(t)$ completely.

We first prove that under the strong influence condition, any sufficiently large compact is positive recurrent. For the converse, we prove that if both agents have moderate influences on each other then there exists a correlation matrix $C$ with all variances less than $\epsilon$ such that the Markov chain is not positive recurrent. Hence $Y(t)$ does not have a stationary distribution; so the same holds true for $Y(t)$, as $Y(t) = Y_{12}(t) \subset Y(t)$.

Stability statement: To prove the stability of this Markov chain we can use the Foster-Lyapunov stability criteria [98]. Towards this aim, we choose the Lyapunov function $L(Y) = Y^2$. To prove the positive recurrence of an large compact, it is sufficient to prove that the drift of this Lyapunov function is upper-bounded everywhere and that this drift is strictly negative outside some compact set.

For the sake of light notation, in the following steps we use $U_{i,j}$ in place of $U^{t}_{i,j}$. In a first step, we bound the Lyapunov drift from above for $y > 0$ (the bound for $y < 0$ is similar). We have

$$
E \left[ L(Y(t + 1)) - L(Y(t)) | Y(t) = y \right]
= P \left[ \min(U_{1,2}, U_{2,1}) = L(Y(t + 1)) | Y(t) = y, \right]
$$
\begin{equation}
\min(U_{1,2}, U_{2,1}) = 1
\end{equation}

\begin{align*}
\begin{aligned}
&+ P[U_{2,1} = 1, U_{1,2} = 0] E[L(Y(t + 1))] \\
&\quad \quad Y(t) = y, U_{2,1} = 1, U_{1,2} = 0 \\
&+ P[U_{2,1} = 0, U_{1,2} = 1] \\
&\quad \quad E[L(Y(t + 1))|Y(t) = y, U_{2,1} = 0, U_{1,2} = 1] \\
&+ P[\max(U_{1,2}, U_{2,1}) = 0] E[L(Y(t + 1))|Y(t) = y, \max(U_{1,2}, U_{2,1}) = 0] \\
&- L(y)
\end{aligned}
\end{align*}

\begin{equation}
(A.1)
\end{equation}

Equation (A.1) follows by breaking the expectation into sum of conditional expectations times probabilities of conditioning events. Equation (A.2) is obtained as follows: given $U_{i,j}$, $Y(t + 1)$ depends only on $Y(t)$ and the endogenous noise. From (2.1) and (2.2), for $\{\min(U_{1,2}, U_{2,1}) = 1\}$, $Y(t + 1) = Y(t) - \gamma_1 Y(t) - \gamma_2 Y(t) + n_2(t) - n_1(t)$, as in this case both agents update their opinions. Similarly, for $\{U_{2,1} = 1, U_{2,1} = 0\}$ and $\{U_{2,1} = 0, U_{2,1} = 1\}$, agents 2 and 1 get closer by a $\gamma_2$ and $\gamma_1$ fraction (up to the rounding) of their opinion difference, respectively. In the remaining case, $\{U_{2,1} = 0, U_{2,1} = 0\}$,
none of the agents moves. All in these cases the opinions of the agents are perturbed by the independent endogenous noise.

We now bound the individual terms by a generic method. Note that all the terms inside the expectation are all of the form \( \mathbb{E} \left[ (g(y) + n_2 - n_1)^2 \right] \), for some function \( g \geq 0 \).

\[
\begin{align*}
\mathbb{E} \left[ (g(y) + n_2 - n_1)^2 \right] \\
= \mathbb{E} \left[ (g(y))^2 + 2g(y)(n_2 - n_1) + (n_2 - n_1)^2 \right] \\
= (g(y))^2 + 2g(y)\mathbb{E} [n_2 - n_1] + \mathbb{E} [(n_2 - n_1)^2] \\
= (g(y))^2 + \mathbb{E} [(n_2 - n_1)^2].
\end{align*}
\]

Note that the functions \( g(y) \) corresponding to our cases are all such that \( g(y) \leq y \). Hence we have

\[
\begin{align*}
\mathbb{E} [L(Y(t + 1)) - L(Y(t))|Y(t) = y] \\
\leq (g(y))^2 + \mathbb{E} [n_2^2 + n_1^2] - L(y) \\
\leq \mathbb{E} [(n_2(n_1)^2)].
\end{align*}
\]

Thus the Lyapunov drift is uniformly upper-bounded, as the noise is of bounded second moment.

For all \( |y| \geq y_0 \) with \( y_0 > 0 \) large enough, \( |y - \lceil \gamma y \rceil| \) is at most \( \nu |y| \) for some \( 0 < \nu < 1 \), if \( \gamma > 0 \) (note that this is true for any rounding and
hence the same proof goes through for every rounding methods). Thus for $|y|$ sufficiently large we have,

\[
\mathbb{E}[L(Y(t+1)) - L(Y(t))|Y(t) = y] \\
\leq f_{1,2}(|y|) f_{2,1}(|y|) \nu y^2 + f_{2,1}(|y|)(1 - f_{1,2}(|y|)) \nu y^2 \\
+ f_{1,2}(|y|)(1 - f_{2,1}(|y|)) \nu y^2 \\
+ (1 - f_{1,2}(|y|))(1 - f_{2,1}(|y|)) y^2 + \mathbb{E}[n_2^2 + n_1^2] - y^2 \\
\leq -(f_{2,1}(|y|)(1 - f_{1,2}(|y|)) + f_{1,2}(|y|)(1 - f_{2,1}(|y|))) \\
(1 - \nu)y^2 + \mathbb{E}[(n_2 - n_1)^2].
\]

When one of the two functions $f_{1,2}(|y|)$ or $f_{2,1}(|y|)$ is asymptotically more than $K|y|^{-\alpha}$ with $\alpha < 2$, it immediately follows from the last relation that $\mathbb{E}[L(Y(t+1)) - L(Y(t))|Y(t) = y] < -a$ for some positive $a$ for $|y|$ large enough. This concludes the proof of the stability result.

**Non-stability statement:** To prove this converse result, the following theorem by Borovkov et. al. [99] is very useful and we briefly discuss this theorem here. Let $S(t)$ be a discrete-time Markov chain on $\{0, 1, \cdots\}$. Let $\{S^\epsilon(t)\}_{\epsilon \geq 0}$ be a collection of Markov chains on the same state space. Assume that the transition probabilities of $S^\epsilon(t)$ converge to that of $S(t)$ as $\epsilon \to 0$, i.e. $\lim_{\epsilon \to 0} \mathbb{P}\{S^\epsilon(t+1) = j|S^\epsilon(t) = i\} = \mathbb{P}\{S(t+1) = j|S(t) = i\}$ for all states $i, j$. Define the following quantities:

\[
\mu(i, \epsilon) = \mathbb{E}[S^\epsilon(1) - i|S^\epsilon(0) = i], \\
\sigma(i, \epsilon) = \mathbb{E}[(S^\epsilon(1) - i)^2|S^\epsilon(0) = i].
\]
We assume that the following limits exist:

\[
\mu = \lim_{i \uparrow \infty} \lim_{\epsilon \downarrow 0} i(\mu(i, \epsilon) + \epsilon), \\
\sigma = \lim_{i \uparrow \infty} \lim_{\epsilon \downarrow 0} \mathbb{E} [ (S^\epsilon(1) - i)^2 | S^\epsilon(0) = i ].
\]

**Theorem 47.** Given \( S, S^\epsilon, \mu(i, \epsilon), \sigma(i, \epsilon), \mu \) and \( \sigma \) defined as above, consider the following conditions:

1. \(-\infty \leq \mu \leq +\infty;\)
2. \(0 < \sigma < \infty \) and \( \sup_{i \geq 0, \epsilon \geq 0} \sigma(i, \epsilon) < \infty;\)
3. \( \sup_{i \geq 0, \epsilon \geq 0} \mathbb{E} [ |S^\epsilon(1) - i|^{2+\gamma} | S^\epsilon(0) = i ] < \infty \) for \( \gamma > 0. \)

If conditions 1, 2 and 3 hold then

a. if \( 2\mu > \sigma \), then \( S \) is positive recurrent;

b. if \(-\sigma \leq 2\mu \leq \sigma \), then \( S \) is null recurrent;

c. if \( \sigma > -2\mu \) then \( S \) is recurrent;

d. if \( 0 < \sigma < -2\mu \) then \( S \) is transient.

We use the above theorem to prove the converse. We use an endogenous noise \( n_1(t) \) and \( n_2(t) \) with \( n_1(t) \) and \( n_2(t) \) independent and with \( P(n_i(t) = 0), P(n_i(t) = +1), P(n_i(t) = -1) > 0 \). This makes the chain irreducible. We choose \( \mathbb{E}[n_1^2] \) and \( \mathbb{E}[n_2^2] \) to be less than \( \epsilon \). Thus to show that the dynamics is
not stable (in the sense of Def. 1), it is sufficient to show that $Y(t)$, which is an irreducible chain, is not positive recurrent.

From $Y(t)$, we first construct the auxiliary process $Y_+(t) \in \mathbb{Z}_+$. Since the noise (per component) takes its values in $\{-C, -C+1, \ldots, C-1, C\}$, the process $Y(t)$ can reach $\mathbb{Z}_+$ from $\mathbb{Z}_-^* = \{-1, -2, \ldots\}$ only through the states $D = \{0, 1, \ldots, 2C-1\}$. Given some initial state for $Y(t)$ of the form $-k$ for some $k > 0$, conditioned on the event that $Y(t)$ reaches $\mathbb{Z}_+$ in one step, let $p^{[C]}(-k, j)$ denote the probability that the first state that $Y(t)$ hits in $\mathbb{Z}_+$ is $j \in D$ (it may be that $\sum_{j \in D} p^{[C]}(-k, j) < 1$ in the transient case). Note that for $j \geq 2C$, $p^{[C]}(-k, j) = 0$. The process $Y_+(t)$ starts in $\mathbb{Z}_+$ and has the same dynamics as $Y(t)$, with the only difference that when it first reaches $\mathbb{Z}_-$, say at $-l$ for some $l > 0$, the process $Y_+(t)$ is set to $j \in \{0, 1, \ldots, C-1\}$ with probability $p^{[C]}(-l, j)$.

The key observation used below is that, if $Y_+$ is not positive recurrent, then $Y(t)$ cannot be positive recurrent. This is because the time to return to $i \in \mathbb{Z}_+$ in the process $Y(t)$ is the time to return to $i$ in the process $Y_+(t)$ plus some additional time which corresponds the time spent by $Y(t)$ in $\mathbb{Z}_-^*$. Hence, the expected time to return to $i \in \mathbb{Z}_+$ in the process $Y_+(t)$ is a lower-bound on the expected time to return to $i$ in the process $Y(t)$. In order to show that $Y(t)$ is not positive recurrent, it is then sufficient to show that $Y_+(t)$ is not positive recurrent.

Note that the state-transition probabilities $p(i, j)$ for the process $Y_+(t)$ are the same as those for $Y(t)$ if $i > 4C$. This fact will be used during the
course of the proof.

We assume that both $f_{12}$ and $f_{21}$ are $\Theta \left( \frac{1}{r^\alpha} \right)$ with $\alpha > 2$ and we also assume that $\gamma_1 = \gamma_2$. It will be apparent in the proof that the case with different exponents and update parameters is very similar.

Consider the sequence of processes $\{Y_+^\epsilon(t)\}_{\epsilon > 0}$ which are constructed in a manner similar to $Y_+(t)$, but with the functions $\Theta \left( \frac{1}{r^\alpha} + \epsilon \right)$ instead of $\Theta \left( \frac{1}{r^\alpha} \right)$. We now apply Theorem 47 to this sequence of processes. We start with proving that Conditions 1-3 hold.

We have, for $i > C$,

$$
\sigma(i, \epsilon) = \mathbb{E} \left[ (Y_+^\epsilon(1) - i)^2 | Y_+^\epsilon(0) = i \right]
\geq \mathbb{E}[n_1^2(t) + n_2^2(t)]\mathbb{P}(\max(U_{1,2}, U_{2,1}) = 0),
$$

which goes to $\mathbb{E}[n_1^2(t) + n_2^2(t)]$ as $i \to \infty$ for all $\epsilon \geq 0$, so that $\sigma > 0$ if it exists.

To check Conditions 1-3, we compute $\lim_{i \to \infty} \lim_{\epsilon \to 0} \sigma(i, \epsilon)$ and $\lim_{i \to \infty} \lim_{\epsilon \to 0} \epsilon \mu(i, \epsilon) \to \infty$. For the particular evolution of our Markov chain $Y(t)$ and the choice of $Y^\epsilon(t)$ in our case, these limits are the same as $\lim_{i \to \infty} \sigma(i, 0)$ and $\lim_{i \to \infty} \epsilon \mu(i, 0)$ respectively. Hence, for simplicity, we compute these limits for $Y_+(t)$ with $i \to \infty$ instead of limits for the process $Y_+^\epsilon$.

For such $i$, we also have

$$
\sigma(i, 0) \leq 2(\gamma i)^2 \mathbb{P}(U_{1,2} + U_{2,1} = 1)
+ 4(\gamma i)^2 \mathbb{P}\{\min(U_{1,2}, U_{2,1}) = 1\}
+ \mathbb{E}[n_1^2(t) + n_2^2(t)]\mathbb{P}(\max(U_{1,2}, U_{2,1}) = 0).
$$
Note that
\[ P(\text{only one of } (U_{1,2}, U_{2,1}) = 1) = 2 \frac{1}{i^{\alpha}} (1 - o(\frac{1}{i^{\alpha}})) , \]
and
\[ P\{ \min(U_{1,2}, U_{2,1}) = 1\} = \frac{1}{i^{2\alpha}} . \]
Hence, the first two terms in the expression for \( \sigma(i, \epsilon) \) are upper-bounded, which implies \( \sigma(i, \epsilon) \) is bounded above. Hence \( \sigma < \infty \) if it exists. It is easy to check that \( \sigma \) exists and is equal to \( E[n_1^2(t) + n_2^2(t)] \). This concludes the verification of Condition 2.

Similarly, we can obtain \( \mu \) as follows.

\[
i\mu(i, \epsilon) = -i\gamma i \Pr(U_{1,2} + U_{2,1} = 1) + i(-i) \Pr(\min(U_{1,2}, U_{2,1}) = 1) .
\]

Using again the fact that \( \Pr(U_{1,2} + U_{2,1} = 1) \) is \( 2 \frac{1}{i^{\alpha+\epsilon}} (1 - o(\frac{1}{i^{\alpha+\epsilon}})) \) and that \( \Pr(\min(U_{1,2}, U_{2,1}) = 1) \) is \( o(\frac{1}{i^{\alpha+\epsilon}}) \), we get the existence of \( \mu \) and the fact that \( \mu = 0 \).

Next, we check Condition 3. By going over steps similar to those used for developing an upper-bound on \( \sigma(i, 0) \), we obtain:

\[
E \left[ |Y_+(1) - i|^{2+c} \mid Y_+(0) = i \right] \leq E[n_1^2(t) + n_2^2(t)] + \Theta \left( \frac{i^{2+c}}{i^{\alpha}} \right) .
\]
For all \( \alpha > 2 \), we can obtain a \( c > 0 \) such that this bound is finite.

Hence, for \( \alpha > 2 \), the three conditions under which Theorem 47 is applicable hold true. Moreover, the values obtained for \( \mu \) and \( \sigma \) imply that,
for $\alpha > 2$, we have $-\sigma \leq \mu \leq \sigma$. Hence, for $\alpha > 2$, the Markov chain $Y_+(t)$ is null recurrent. This in turn implies that $Y(t)$ is not positive recurrent, which concludes the proof when opinions take values in $\mathbb{Z}$.

### A.1.2 Proof of Theorem 5

**Stability statement:**

Consider the opinion difference process $Y_1(t) = \{Y_{ij}(t) : j \neq 1\}$. Note that this completely describes $Y(t)$. As $P(n_i(t) = 0), P(n_i(t) = 1), P(n_i(t) = -1) > 0$ and the interaction graph is complete, $Y_1(t)$ is irreducible and aperiodic. Hence, to prove stability it is sufficient to show that this Markov chain is positive recurrent. Towards this we use the Foster-Lyapunov theorem.

Consider the Lyapunov function $L(Y_1) = \sum_j Y_{ij}^2$. We have

$$
\mathbb{E} \left[ \sum_j Y_{ij}^2(t + 1) | Y_1(t) \right] = \sum_{(i,j)} P(I(t) = (i,j))
$$

$$
\mathbb{E} \left[ \sum_j Y_{ij}^2(t + 1) | Y_1(t), I(t) = (i,j) \right]
$$

(A.3)

Note that

$$
\mathbb{E} \left[ \sum_j Y_{ij}^2(t + 1) | Y_1(t), I(t) = (i,j) \right]
$$

$$
= \sum_{k \notin \{i,j\}} (Y_{ik}(t) + n_k + n_i)^2
$$

$$
+ \mathbb{E} \left[ Y_{ii}^2(t + 1) + Y_{ij}^2(t + 1) | Y_1(t), I(t) = (i,j) \right]
$$

$$
= \sum_{k \notin \{i,j\}} (Y_{ik}^2(t) + \mathbb{E}[n_k^2] + \mathbb{E}[n_i^2]) + \mathbb{E} \left[ Y_{ii}^2(t + 1) \right]
$$
[A.4]

For $i, j \neq 1$, when using the notation $\gamma_{i,j} = \gamma_{j,i} = \gamma$, $U_{i,j}^t = U_{j,i}^t = U$ and $f_{i,j} = f_{j,i} = f$, we have

\[
E [Y_{1 i}^2(t + 1) + Y_{1 j}^2(t + 1) | Y_1(t), I(t) = (i, j)]
= E [(Y_{i i}(t) + U \hat{z}(\gamma Y_{i,j}(t)) + n_i - n_1)^2 | Y_1(t)]
+ E \left[ (Y_{i j}(t) - U \hat{z}(\gamma Y_{i,j}(t)) + n_j - n_1)^2 \right] | Y_1(t)]
= E [2n_i^2 + n_i^2 + n_j^2]
+ E \left[ (Y_{i i}(t) + U \hat{z}(\gamma Y_{i,j}(t)))^2 \right] | Y_1(t)]
+ E \left[ (Y_{i j}(t) - U \hat{z}(\gamma Y_{i,j}(t)))^2 \right] | Y_1(t)]
= E [2n_i^2 + n_i^2 + n_j^2 + Y_{i i}^2(t) + Y_{i j}^2(t)]
+ 2(Y_{i i}(t) - Y_{i j}(t))f(|Y_{i,j}(t)|)\hat{z}(\gamma Y_{i,j}(t))
+ 2f(|Y_{i,j}(t)|)(\hat{z}(\gamma Y_{i,j}(t)))^2
= E [2n_i^2 + n_i^2 + n_j^2 + Y_{i j}^2(t) + Y_{i i}^2(t)]
+ 2f(|Y_{i,j}|)(Y_{i,j} - \gamma(\gamma Y_{i,j}(t)) + (\hat{z}(\gamma Y_{i,j}(t))\hat{z}(\gamma Y_{i,j}(t)))))^2).
\]  

(A.5)

Note that the function $y \to -y\hat{z}(\gamma y) + (\hat{z}(\gamma y))^2$ is bounded above by a constant $k(\gamma)$, for all $y \in \mathbb{R}$, which in turn is bounded by a constant $\kappa = \max_{i,j} k(\gamma_{i,j})$. 

118
Hence,

\[
\mathbb{E} \left[ Y_{1i}^2(t+1) + Y_{1j}^2(t+1) | Y_1(t), I(t) = (i, j) \right] \\
\leq \mathbb{E}[2n_1^2 + n_i^2 + n_j^2] + Y_{1i}^2(t) + Y_{1j}^2(t) + \kappa. \quad (A.6)
\]

Thus for any \((i, j)\) interaction, the drift is bounded above.

Consider now any \((1, j)\) interaction. In that case only \(Y_{1j}\) changes. From the proof of the two agent case, we have that the drift of \(Y_{1j}^2\) is upper-bounded by

\[
\mathbb{E}[n_1^2 + n_j^2] - c|Y_{1j}|^\epsilon, \ c > 0.
\]

It follows from Eq. (A.3) that \(\mathbb{E}[L(Y_1(t+1) - L(Y_1(t)) | Y_1(t)]\) can be written as

\[
C_1 - \sum_j cP(I(t) = (1, j))|Y_{1j}|^\epsilon,
\]

with \(C_1 < \infty\) and \(c > 0\). When \(|Y_1|_2 > B\), for \(B > O\) constant, there exists a \(j\) s.t. \(|Y_{1j}| > \frac{B}{\sqrt{n}}\) and hence we have an overall strictly negative drift for sufficiently large \(B\).

**Non stability statement:**

Let us single out agent 1 from agents \(\{2, 3, \ldots, n\}\). We define an opinion difference process where for \(i \neq 1\) \(n_i \equiv 0\) and \(P(n_1 = 0) = p, P(n_1 = 1) = P(n_1 = -1) = (1-p)/2\).

---

\(^1\)In fact one can show that \(\kappa \leq 0\). This is because, \(\hat{\varepsilon}(\gamma Y)\) has the same sign as of \(Y\) (else, it is 0), so \(Y_1(\gamma Y)\) is positive. Thus, to show \(\kappa \leq 0\) we need to show \(|Y_1| |\hat{\varepsilon}(\gamma Y)| \geq |\hat{\varepsilon}(\gamma Y)|^2\). But this is true as \(|Y| \geq |\hat{\varepsilon}(\gamma Y)|\). When \(|\hat{\varepsilon}(\gamma Y)| = 0\) we have \(\kappa = 0\).
Consider the process \( Y_1 = \{Y_{1i} : i > 1\} \in \mathbb{Z}^{n-1} \) which completely represents the opinion differences process. This is a Markov process.

The proof idea for this part is as follows. We construct another Markov process \( Y'(t) \in \mathbb{Z}^{n-1} \) as well as a coupling between \( Y' \) and \( Y_1 \) where the marginal laws of \( Y' \) and \( Y_1 \) are preserved. Then using a stochastic dominance argument for Markov chains (Lemma 7.13 of [100]) we show that for a sequence of compacts (increasing to \( \mathbb{Z}^{n-1} \)), at any time \( t \), \( Y_1 \) has less probability to be in any of these compacts than \( Y' \) (starting from same initial state). Then we show that \( Y' \) has a dynamics similar to that of the 2-agent opinion difference process with an \( O\left(\frac{1}{x^2 \ln x}\right) \) influence function. Thus, \( Y' \) is not positive recurrent and hence, so is \( Y_1 \). We elaborate these steps in what follows.

Consider another Markov process

\[
Y'(t) = (Y'_2(t), Y'_3(t), \ldots, Y'_n(t)) \in \mathbb{Z}^{n-1}
\]

(the indexing is chosen to be similar to that of \( Y_1 \)) which evolves as follows. It starts with the same initial condition as \( Y_1 \). If \( I(t) = (i,j) \) in the original process \( Y_1(t) \), then, in \( Y'(t) \), a coin with the probability of success \( p'(t) = f_{i,j}(\min_j |Y'_j(t)|) \) is tossed. If there is no success in the coin toss we set

\[
Y'_i(t) = Y'_{\arg \min_j |Y'_j(t)|}(t) + n_1(t) \quad \forall i,
\]

whereas if there is success we set

\[
Y'_i(t) = n_1(t) \quad \forall i.
\]
Note that in the dynamics $Y_1$, whenever $I(t) = (i, j)$, a coin is tossed with the probability $p(t) = f_{i,j}(|Y_{ij}|)$. We now explain how to couple the transitions in $Y_1$ and $Y'_1$ while ensuring that the evolution of the marginals $Y_1$ and $Y'_1$ are by themselves Markovian and with the laws described above: given $I(t) = (i, j)$ (we recall that by construction $I(t)$ is the same in $Y'(t)$ and $Y_1(t)$), in $Y_1$ (resp. $Y'(t)$) rather than tossing independent coins with probability of success $p(t)$ (resp. $p'(t)$), we toss a common coin with probability of success $\max(p(t), p'(t))$, (this coin is common to both processes). If this common coin toss is a success, then we toss two additional independent coins, the first one with probability of success $\frac{p(t)}{\max(p(t), p'(t))}$ and the second one with probability of success $\frac{p'(t)}{\max(p(t), p'(t))}$. The coupling consists in declaring a success in $Y_1(t)$ (resp. $Y'(t)$) if this first (resp. second) coin toss is a success. This preserves the marginal Markov properties for both processes. We use this coupling to prove some stochastic dominance below.

The state space for both processes is $\mathbb{Z}^{n-1}$. We define the following partial order on this state space: for $x, y \in \mathbb{Z}^{n-1}$ we say $x \leq_{cw} y$ if $|x_i| \leq |y_i|$ for $1 \leq i \leq n-1$. We call a set $A$ increasing (see [100]) if $x \in A$ and $x \leq_{cw} y$ implies that $y \in A$. Let $P$ and $P'$ be two probability distributions on $\mathbb{Z}^{n-1}$. One says [100] that $P' \leq_{cw} P$ if

$$P'(A) \leq P(A) \ \forall A \text{ increasing}.$$ 

Let $K$ and $K'$ be two Markov kernels on $\mathbb{Z}^{n-1}$. We say that $K' \leq_{cw} K$ if for
all \( y', y \in \mathbb{Z}^{n-1} \) s.t. \( y' \leq_{cw} y \) and for all increasing sets \( A \)

\[ K'(y', A) \leq K(y, A). \]

Let \( \pi_t \) and \( \pi'_t \) denote the distribution of the Markov chains at time \( t \). If the initial distributions of the two Markov chains are identical and if \( K' \leq_{cw} K \), then by Lemma 7.13 of [100]

\[ \pi'_t \leq_{cw} \pi_t, \; \forall t. \]

We show below that the Kernels \( K \) (of \( Y_1 \)) and \( K' \) (of \( Y'_1 \)) are ordered in the \( \leq_{cw} \) sense. Let us first consider the transition kernels in the case where \( n_i \equiv 0 \) for all \( i \) (including for agent 1) and show that \( K \) and \( K' \) are ordered in this case.

There are two types of interactions: (i) interactions among agents in \([2 : n]\) and (ii) interactions between agent 1 and others. We show that in both cases if \( Y_1 = y \) and \( Y'_1 = y' \) and \( y' \leq_{cw} y \), then the next states (in the \( n_1 = 0 \) case) \( y'_+ \) and \( y_+ \) satisfy \( y'_+ \leq_{cw} y_+ \). Notice that as per the dynamics of \( Y'_1 \), \( Y'_1 \) has all its coordinates equal. So this means that any coordinate of \( y_+ \) has an absolute value no lesser than that of any coordinate of \( y'_+ \).

Consider first an interaction of the form \((i, 1)\). Note that given \( y' \leq_{cw} y \), since \( f_{i,1} \) is non-increasing and since then \( p'(t) > p(t) \), in the coupled version of the processes, any influence in \( Y_1 \) implies that \( Y' \) becomes \( 0 \). Hence we have \( y'_+ \leq_{cw} y_+ \). If there is no influence in \( Y_1 \), either \( y'_+ \) is equal to \( 0 \) (this is possible as \( p'(t) > p(t) \)), so that we still have \( y'_+ \leq_{cw} y_+ \) in this case. Or there is no
influence in $\mathbf{Y}'_+$ either, and the $\leq_{cw}$ ordering is maintained as $y'_+ = x_{\arg\min|y'_i|}$ and $y_+ = y$ respectively.

Consider now an interaction between two agents in $[2 : n]$. It may happen that the distance between these agents is less than $\min_j |Y'_j(t)|$ and hence $Y_1$ may have a higher probability of influence than $\mathbf{Y}'$. This can only happen if agent 1 is not between them. But in that case the influence does not reduce $\min_i |Y_{1i}|$ which is larger than $\min_j |Y'_j(t)|$ (by the assumption that $y' \leq_{cw} y$). As there is no coin toss success in $\mathbf{Y}'$ we set all the coordinates to $\min_j |Y'_j(t)|$, we still have $y'_+ \leq_{cw} y_+$.

Hence, if $n_i \equiv 0$, $y' \leq_{cw} y$ implies that $y'_+ \leq_{cw} y_+$ pathwise (in the coupling).

Note that all coordinates of $y'_+$ are the same by construction (before and after adding $n_1$). On the other hand, for $y$, the coordinates are not same, but we know that each coordinate of $y$ is larger in magnitude than that of $y'$.

Now note that for $|a| \geq |b|$, $P(|a + n_1| \geq c) \geq P(|b + n_1| \geq c)$. This follows from the symmetry of the probability mass function of $n_1$ and noting that the mass of $a + n_1$ in the set $\{|x| \geq c\}$ is more than that of $b + n_1$. As $\min_i |y_{+i}| \geq |y'_{+j}|$ for any $j$, we have $P(|y'_{+j} + n_1| \geq c) \leq P(\min_i |y_{+i} + n_1| \geq c)$, thus implying the desired ordering.

**Remark 48.** By choosing a different noise $n'_1$ for $\mathbf{Y}'_1$ such that $P(n'_1 = 0) = \frac{1+p}{2}$ a path-wise dominance between $\mathbf{Y}_1$ and this $\mathbf{Y}'_1$ can also be shown.

This completes the proof of the ordering of the two kernels.
Consider now a collection of increasing cubes \( \{C_i\} \) centered at origin. Note that complement of each these is an increasing set. So at any time \( t \) \( \pi_t(C_i) \leq \pi'_t(C_i) \) for any \( C_i \). If \( Y' \) is not positive recurrent then it implies that for any of these cubes \( \lim_{t \to \infty} \pi_t(C_i) = 0 \). Thus it only remains to show that \( Y' \) is not positive recurrent.

Note that by construction, \( Y'(t) \) behaves like some one dimensional process after \( t = 1 \), as the values taken by \( Y' \) are on the line \( e = (1, 1, \cdots, 1) \). Hence we can just consider one coordinate, say \( Y'_2(t) \), to investigate its class properties.

Now note that as \( f_{i,j} = O\left(\frac{1}{x^{1+p}}\right) \) for all \( i, j \). At any time \( t > 1 \), \( f_{i,j}(\min_j |Y'_j(t)|) = f_{i,j}(Y'_2(t)) = O\left(\frac{1}{Y'^2} \right) \).

Thus using the similar technique as in the proof of the two agent case that \( Y' \) is null recurrent (because any influence is \( O\left(\frac{1}{x^{1+p}}\right) \) and noise \( n_1 \) has a strictly positive variance as \( 0 < p < 1 \)). Hence, in turn we prove that \( Y_1 \) is not stable.

This completes the proof of Theorem 5.

### A.1.3 Proof of Theorem 6

The proof of Theorem 5 extends to the case where \( U^t_{i,j} \) and \( U^t_{j,i} \) have arbitrary joint distribution but they have the same marginal. This is because the proof only uses the linearity of expectations and the marginals of \( U^t_{i,j} \) and
$U^t_{j,i}$. With obvious notation, we again get

$$
\begin{align*}
\mathbf{E} \left[ Y_{i1}^2(t+1) + Y_{j1}^2(t+1) | Y_1(t), I(t) = (i,j) \right] \\
= \mathbf{E} \left[ (Y_{i1}(t) + U_{i,j}(\gamma Y_{i,j}(t)) + n_i - n_1)^2 | Y_1(t) \right] \\
+ \mathbf{E} \left[ (Y_{j1}(t) - U_{j,i}(\gamma Y_{i,j}(t)) + n_j - n_1)^2 | Y_1(t) \right] \\
= \mathbf{E} [2n_i^2 + n_i^2 + n_j^2] + Y_{i1}^2(t) + Y_{j1}^2(t) \\
+ 2(Y_{i1}(t) - Y_{j1}(t)) f(|Y_{i,j}(t)|) \bar{z}(\gamma Y_{i,j}(t)) \\
+ 2f(|Y_{i,j}(t)|) (\bar{z}(\gamma Y_{i,j}(t)))^2 \\
= \mathbf{E} [2n_i^2 + n_i^2 + n_j^2] + Y_{i1}^2(t) + Y_{j1}^2(t) \\
+ 2f(|Y_{i,j}|)(-Y_{i,j} \bar{z}(\gamma Y_{i,j}(t)) + (\bar{z}(\gamma Y_{i,j}(t)))^2).
\end{align*}
$$

(A.7)

Hence, exactly the same proof for stability works here.

**Non-stability statement:** For the converse the proof is almost similar to the proof of Theorem 5.

Here we have two different coupling techniques for the coin tosses depending on $I(t)$. This is because for interaction with 1 we need only one coin toss in $Y_1$ while for other interactions we need two coin tosses, possibly correlated.

Given $I(t) = (1,j)$, $j \in [2:n]$, we toss a coin for $j$ in $Y_1$ (so in $Y'$). In this case these coin tosses can be coupled exactly in the same way as we couple the coin tosses in the proof of Theorem 5.

Given $I(t) = (i,j)$, $i,j \in [2:n]$ for $Y_1$ we toss two coins (one for $i$ and $j$ each) with the same marginal probability $p(t) = f_{i,j}(|Y_{i,j}|)$ but with an
arbitrary joint distribution. Opinions of $i$ and $j$ are updated based on the success of these coin tosses.

In $Y'$ we toss two coins (for $I(t) = (i, j)$, $i, j \in [2 : n]$) independently with probabilities $f_{i,j}(|Y_j(t)|)$. If there is at least one success we update $Y'$ to $n_11$. So here the success probability is $p'(t) = 2f_{i,j}(|Y_j(t)|) - (f_{i,j}(|Y_j(t)|))^2$.

We again couple the coin tosses in the two processes (for $I(t) = (i, j)$, $i, j \in [2 : n]$) : in $Y'(t)$ note that the kernel remains same if we toss a single coin with success probability $p'(t)$.

Like in the proof of Theorem 5 we first toss a coin with probability $P(t) = \max(p(t), p'(t))$ (common to both processes).

If no success in the common coin toss we do not do anything further for $Y'$ and consider $Y'$ coin toss to be a failure. If there is success, for $Y'$ we toss another coin with probability $\frac{p'(t)}{\max(p(t), p'(t))}$ and if this is success then we consider success in $Y'$ (i.e., update $Y'$ to $n_11$).

For $Y_1$, if there is a success in the common coin toss, we toss a coin with probability $\frac{p(t)}{\max(p(t), p'(t))}$ and based on the success (or failure) of the toss we toss another coin with the right conditional probability from the joint distribution of the two coin tosses. The result of these last two coin tosses are used for updating $Y_1$.

If there is a failure in the common coin toss, we toss a coin with a conditional distribution (obtained from the joint) conditioned on the failure
event. In this case the results of the common coin toss and the second toss are used for updating $Y_1$.

Note that these couplings also preserve the marginal Markov properties of $Y_1$ and $Y'$ respectively. Note that for a given initial condition $x \leq_{cw} y$ (like in the proof of Theorem 5) $p' \geq p$, except if $I(t) = (i, j)$, $i, j \in [2 : n]$ and $i, j$ are on the same side of 1. But in this case (as argued for Theorem 5) $x_+ \leq_{cw} y_+$ before adding $n_1$ (by the same argument). For rest of the cases $p' \geq p$ and the arguments are also exactly the same as in the proof of Theorem 5.

This completes the proof of Theorem 6.

For the rest of the theorems we follow a flow that is different from the way the results are presented in Sec. 2.3. We first prove Theorem 8 as it builds on the proof of Theorem 6.

A.1.4 Proof of Theorem 8

Consider $Y_1 = \{Y_{1j}\}$ and the Lyapunov function $L(Y_1) = \sum_j Y_{1j}^2$.

From the proof of Theorem 7, it follows that, the for all $(i, j)$ interactions, with $i, j \in [2 : n]$, the drift is always upper-bounded by

$$C_1 - f_{i,j}(|Y_{i,j}|)(Y_{i,j}\hat{z}(\gamma Y_{i,j}(t))) - (\hat{z}(\gamma Y_{i,j}(t)))^2),$$
with $C_1 < \infty$. Note that for $|Y_{i,j}|$ sufficiently large and $\gamma > 0$, $\exists \gamma' > 0$ s.t.

$$(Y_{i,j}\mathbb{Z}(\gamma Y_{i,j}(t)) - (\mathbb{Z}(\gamma Y_{i,j}(t))))^2 \geq \gamma' Y_{i,j}^2$$

If an $i,j$ interaction happens (which happens with strictly positive probability), if neither of them is 1, then by the same argument as in the proof of Theorem 6, the drift of the Lyapunov function is always upper-bounded. Moreover, it is strictly negative for sufficiently large $|Y_{ij}|$ if $f_{i,j} = \Omega \left( \frac{1}{x^2} \right)$. On the other hand, if one of them is 1 (say $i$), then only $j$ moves towards 1 (other coordinates of $Y_1$ do not change) resulting in an upper-bounded drift and strictly negative drift for sufficiently large $|Y_{ij}|$ if $f_{j,1} = \Omega \left( \frac{1}{x^2} \right)$ (exactly as in the proof of Theorem 6).

Whenever $Y_1$ is outside the compact $\{x : ||x||^2 \leq B\}$, there exists a $j$ s.t. $|Y_{1j}| > \frac{B}{\sqrt{n}}$. Consider a tree $T$ rooted at $l$ which is a subgraph of $G'$. Then consider the variables $Y_T = \{Y_{i,j} : i$ precedes $j$ in $T\}$. Let $\pi_j$ be the path first from 1 to $l$ and then from $l$ to $j$ on $T$. Then $Y_{1j} = \sum_{(k,k')} \in \pi_j Y_{k,k'}$. As $|Y_{1j}| > \frac{B}{\sqrt{n}}$, there exists a $(k,k') \in \pi_j$ such that $|Y_{kk'}| > \frac{B}{n^{3/2}}$. Note that $(k,k')$ may be $(1,l)$ as well and on every edge on the path $\pi_j$, we have $\Omega \left( \frac{1}{x^2} \right)$ influence.

As $|Y_{kk'}| > \frac{B}{n^{3/2}}$, for sufficiently large $B$, the drift is strictly negative and can be made larger in absolute value than any finite constant. So, for $Y_1$ outside a ball of size $B$ we have a strictly negative drift. Since the drift is upper-bounded everywhere, by the Foster-Lyapunov theorem, $Y_1$ is stable.
A.1.5 Proof of Theorem 7

Consider the Lyapunov function
\[ L(Y) = \sum_{q} \left( \sum_{j \neq q} Y_{qj}^2 \right) \].
Again we want to show that the expected change of this function is bounded above and has a strictly negative drift for \( Y \) outside a sufficiently large compact. We have

\[
\mathbb{E} \left[ \sum_{q} \left( \sum_{j \neq q} Y_{qj}^2(t+1) \right) | Y(t) \right]
= \sum_{p,p'} P(I(t) = (p,p')) \sum_{q} \mathbb{E} \left[ \left( \sum_{j \neq q} Y_{qj}^2(t+1) \right) | Y(t), I(t) = (p,p') \right]
\] (A.8)

Hence, to show that the drift is upper-bounded, it is sufficient to show that

\[
\mathbb{E} \left[ \left( \sum_{j \neq q} Y_{qj}^2(t+1) \right) | Y(t), I(t) = (p,p') \right]
\]

is at most a constant plus \( \sum_{j \neq q} Y_{qj}^2(t) \) for each \( q \not\in \{p,p'\} \) and that

\[
\mathbb{E} \left[ \sum_{j \neq p} Y_{pj}^2(t+1) + \sum_{j \neq p'} Y_{p'j}^2(1) | Y(t), I(t) = (p,p') \right]
\]
is no more than a constant plus \( \sum_{j \neq p} Y_{pj}^2 + \sum_{j \neq p'} Y_{p'j}^2 \) for any \( p,p' \) pair.

Note that, given \( I(t) = (p,p') \), for any \( q \not\in \{p,p'\} \), \( \mathbb{E}[\sum_{j \neq q} Y_{qj}^2(t+1)|Y(t), I(t) = (p,p')] \) is upper-bounded by a constant plus \( \sum_{j \neq q} Y_{qj}^2(t) \). This follows using exactly the same steps as in the proof of Theorem 5. The proof of Theorem 5 only discusses the upper-bounding of \( \mathbb{E}[\sum_{j \neq 1} Y_{1j}^2(t+1)|Y(t), I(t) = (p,p')] \), but we do not use any property of \( \{f_{j,1}\} \) to obtain
Equation (A.7). Hence, the same bound holds, i.e.,

\[
\begin{align*}
\mathbb{E}[\sum_{j \neq q} & Y_{qj}^2(t + 1)|Y(t), I(t) = (p, p')] \\
\leq & \sum_{k \notin \{p, p'\}} (Y_{j}^2(t) + \mathbb{E}[n_k^2] + \mathbb{E}[n_{q_k}^2]) \\
+ & \mathbb{E}[2n_q^2 + n_p^2 + n_{p'}^2] \\
+ & f(|Y_{p, p'}|)(-Y_{p, p'}^2(\gamma Y_{p, p'}(t))) + (\gamma Y_{p, p'}(t))^2).
\end{align*}
\]

(A.9)

Note that we can write \(\sum_{j \neq q} Y_{j}^2(t) + \sum_{j \neq q'} Y_{j}^2(t)\) as \(\sum_{j \notin \{p, p'\}} (Y_{j}^2(t) + Y_{j}^2(t)) + 2Y_{p p'}^2\) which in turn can be written as \(2Y_{p p'}^2 + \sum_{j \notin \{p, p'\}} (Y_{j}^2(t) + Y_{j}^2(t)).\) In addition,

\[
\begin{align*}
\mathbb{E}[Y_{p p'}^2(t + 1)|Y_1(t), I(t) = (p, p')] \\
\leq & Y_{p p'}^2(t) + \mathbb{E}[2n_1^2 + n_p^2 + n_{p'}^2] \\
+ & f(|Y_{p, p'}|)(-Y_{p, p'}^2(\gamma Y_{p, p'}(t))) + (\gamma Y_{p, p'}(t))^2).
\end{align*}
\]

(A.10)

On the other hand,

\[
\mathbb{E} \left[ \sum_{j \notin \{p, p'\}} (Y_{j}^2(t + 1) + Y_{j}^2(t + 1)) |Y(t), I(t) = (p, p') \right] = \\
\sum_{j \notin \{p, p'\}} \mathbb{E} \left[ (Y_{j}^2(t + 1) + Y_{j}^2(t + 1)) |Y(t), I(t) = (p, p') \right].
\]

(A.11)

Using the same steps as those for deriving Equation (A.7) one gets that each term in the last sum is bounded from above by

\[
\begin{align*}
\mathbb{E}[2n_q^2 + n_p^2 + n_{p'}^2] + Y_{j}^2(t) + Y_{j}^2(t) \\
+ & f(|Y_{p, p'}|)(-Y_{p, p'}^2(\gamma Y_{p, p'}(t))) + (\gamma Y_{p, p'}(t))^2).
\end{align*}
\]

(A.12)
As \( f(|Y_{p,p'}|)(-Y_{p,p'}\hat{z}(\gamma Y_{p,p'}(t)) + (\hat{z}(\gamma Y_{p,p'}(t)))^2) \) is bounded from above, the expected drift of \( L \) conditioned on \( I(t) = (p, p') \) is upper-bounded for any \( p, p' \). This proves that the Lyapunov drift is upper-bounded. Next, we show that it is strictly negative outside a sufficiently large compact.

From Equations (A.10), (A.11) and (A.12) we get

\[
\mathbb{E}[L(Y(t+1)) - L(Y(t))|Y(t), I(t) = (p, p')] \leq C_2 + 2(n-1)f(|Y_{p,p'}|)(-Y_{p,p'}\hat{z}(\gamma Y_{p,p'}(t)) + (\hat{z}(\gamma Y_{p,p'}(t)))^2),
\]

where \( C_2 = 4 \sum_j \mathbb{E}[n_j^2] \). Thus,

\[
\mathbb{E}[L(Y(t+1)) - L(Y(t))|Y(t)] \\
\leq C_2 + 2(n-1) \sum_{p,p'} P(I(t) = (p, p')) f(|Y_{p,p'}|)
\]

\[
(-Y_{p,p'}\hat{z}(\gamma Y_{p,p'}(t)) + (\hat{z}(\gamma Y_{p,p'}(t)))^2),
\]

\[
\leq C'_2 + 2(n-1) \sum_{p' \neq 1} P(I(t) = (1, p')) f(|Y_{1,p'}|)
\]

\[
(-Y_{1,p'}\hat{z}(\gamma Y_{1,p'}(t)) + (\hat{z}(\gamma Y_{1,p'}(t)))^2),
\]

where \( C'_2 = C_2 + 2(n-1)\kappa < \infty \) and the last step follows from the fact that \((-Y_{p,p'}\hat{z}(\gamma Y_{p,p'}(t)) + (\hat{z}(\gamma Y_{p,p'}(t)))^2)\) is bounded.

Now, \( P(I(t) = (1, p')) > 0 \) for all \( p' \neq 1 \) and \( f(|Y_{1,p'}|) \geq c/|Y_{1,p'}|^2 - \epsilon \).

Thus we have

\[
\mathbb{E}[L(Y(t+1)) - L(Y(t))|Y(t)] \\
\leq C'_2 - 2(n-1) \sum_{p' \neq 1} P(I(t) = (1, p')) \gamma' |Y_{1,p'}|^{\epsilon},
\]

131
for some $\gamma' > 0$.

Now whenever $Y$ is outside a sufficiently large $L_2$-ball, $\sum_p w_p |Y_{1p}|^\alpha$ is also large for $w_p > 0$ and $\alpha > 0$. Thus, for sufficiently large $Y$ (in e.g. the $L_2$ norm) $C'_2 < 2(n - 1) \sum_{p' \neq 1} P(I(t) = (1, p')) \gamma'|Y_{1,p'}|$, implying that the drift is strictly negative indeed.

### A.1.6 Proof of Theorem 9

This proof builds on the proof of Theorem 7. Considering the same Lyapunov $L(Y) = \sum_q \left( \sum_{j \neq q} Y_{qj}^2 \right)$ and following the exactly same steps we can derive the same bound as in Equation (A.13). Thus we have

$$
\mathbb{E} [L(Y(t + 1)) - L(Y(t)) | Y(t)] 
\leq \ C_2 + 2(n - 1) \sum_{p,p'} P(I(t) = (p, p')) f(|Y_{p,p'}|) 
\left( -Y_{p,p'} \gamma Y_{p,p'}(t) + (\gamma Y_{p,p'}(t))^2 \right).
$$

Hence, the drift is upper-bounded as the function $f(|Y_{p,p'}|)(-Y_{p,p'} \gamma Y_{p,p'}(t)) + (\gamma Y_{p,p'}(t))^2$ is upper-bounded.

Consider a tree $T \subset G$ s.t. for any $(p, p') \in T$ $f_{p,p'}(x) = \Omega \left( \frac{1}{x^{2-\epsilon}} \right)$. Then

$$
\mathbb{E} [L(Y(t + 1)) - L(Y(t)) | Y(t)] 
\leq \ C_2 + 2(n - 1) \sum_{p,p'} P(I(t) = (p, p')) f(|Y_{p,p'}|) 
\left( -Y_{p,p'} \gamma Y_{p,p'}(t) + (\gamma Y_{p,p'}(t))^2 \right).
$$

132
\[ \leq C_2' + 2(n-1) \sum_{(p,p') \in T} P(I(t) = (p,p')) f(|Y_{p,p'}|) \\
( - Y_{p,p'} z(\gamma Y_{p,p'}(t)) + (\gamma Y_{p,p'}'(t))))^2 \\
\leq C_2' - 2(n-1) \sum_{(p,p') \in T} P(I(t) = (p,p')) \gamma |Y_{p,p'}|^e. \]

Whenever \( Y_1 \) is large (in some norm) so is \( Y_T = \{ Y_{pp'} : (p,p') \in T \} \). This is because \( Y_T = A Y_1 \) for some invertible matrix \( A \), so \( c_1 ||Y_T||_2 \leq ||Y_1||_2 \leq c_2 ||Y_T||_2 \) (as all eigenvalues are non-zero). Note that any norm is again bounded below and above by constants times \( ||\cdot||_2 \) norm. Another way to see that large \( Y_1 \) implies large \( Y_T \) is to note the fact that \( Y_1 \) large implies \( \exists j \) s.t. \( |Y_{ij}| \) is large. Consider the same argument as in the proof of Theorem 8, where we consider a path \( \pi_j \) to \( j \) from 1, then on that path there exists an edge \((i,j)\) s.t. \( Y_{i,j} \) is large, so \( Y_T \) is large. Hence, by the same argument as in the proof of Theorem 6, the drift is strictly negative outside a sufficiently large compact.

### A.1.7 Proof of Theorem 10

Like in the case with endogenous noise, here also we use the Foster-Lyapunov stability criterion for irreducible Markov chains [101]. W.l.o.g. we can assume 1 to be the strong influencer. Consider the process \( Y(t) = Y_{12}(t) \).

To show positive recurrence, we use the Lyapunov function \( L(Y) = |Y| \).

The expected drift of the Lyapunov function can be divided into parts as in the proof of Theorem 2, namely into conditional expectation conditioned
on events \((U_{2,1}, U_{1,2}) \in \{0, 1\}^2\).

Note that for a given \(Y(t) = y\) and realizations of \(U_{2,1}\) and \(U_{1,2}\), the updated \(Y(t + 1)\) is a function of \(y, U_{2,1}, U_{1,2}\) and can be written as 
\[g((y, U_{2,1}, U_{1,2})).\]
Hence,
\[Y(t + 1) = g(y, U_{2,1}, U_{1,2}) + n_2 - n_1.\]

Note that
\[E[L(Y(t + 1)) - L(Y(t))|Y(t) = y] = \sum_{u_2, u_1 \in \{0, 1\}} P(u_2, u_1) E[L(Y(t + 1)) - L(Y(t))|Y(t) = y, U_{2,1} = u_2, U_{1,2} = u_1]\]
\[= \sum_{u_2, u_1 \in \{0, 1\}} P(u_2, u_1) E[|g(y, u_2, u_1) + n_2 - n_1| - |y|]\]
\[\leq \sum_{u_2, u_1 \in \{0, 1\}} P(u_2, u_1) (|g(y, u_2, u_1)| - |y|) + E[|n_2|] + E[|n_1|]\]

Note that by the update rule of the dynamics, for any \(u_2, u_1, g(y, u_2, u_1) \leq |y|\). As noise has finite mean, the Lyapunov drift is hence upper-bounded.

Consider the function \(P(u_2, u_1)(|g(y, u_2, u_1)| - |y|)\). If \(u_2 = u_1 = 0\), then \(g(y, u_2, u_1) = y\); if \(u_2 + u_1 = 1\) then \(g(y, u_2, u_1) = y - \hat{z}(\gamma y)\) and if \(u_2 + u_1 = 2\) then \(g(y, u_2, u_1) = y - 2\hat{z}(\gamma y)\). Hence, for \(u_2 = u_1 = 0\), this term is 0. On the other hand, as \(y\) and \(\hat{z}(\gamma y)\) have same signs, for \(u_2 + u_1 \geq 1\) (\(|g(y, u_2, u_1)| - |y|\) is strictly negative:
\[|g(y, u_2, u_1)| - |y| \leq -|\hat{z}(\gamma y)|.\]
Note that \( P(u_2 = 1) = f_{2,1}(|y|) \). Hence,

\[
\sum_{u_1} P(u_2 = 1, u_1)(|g(y, u_2, u_1)| - |y|) \leq -f_{2,1}(|y|)|\hat{z}(\gamma y)|. \quad (A.14)
\]

As \( f_{2,1}(|y|) = \frac{c}{|y|^{1-\epsilon}} \) for sufficiently large \(|y|\), the drift is upper-bounded by

\[
E[|n_2|] + E[|n_1|] - \frac{c|\hat{z}(\gamma y)|}{|y|^{1-\epsilon}}.
\]

Again, for sufficiently large \(|y|\) this is strictly negative. This completes the proof of Theorem 10

### A.1.8 Proof of Theorem 11

W.l.o.g. let us assume 1 has very strong influence on all other agents. The proof of this theorem is similar to that of Theorem 7 with the Lyapunov function being \( \sum_q \sum_{j \neq q} |Y_{qj}| \).

Consider any \( p, p' \) interaction. Note that if \(|Y_{qp}|\) increases in expectation then \(|Y_{qp'}|\) decreases in expectation by the same amount. Hence, the drift of \( \sum_{q \neq \{p, p'\}} \sum_j |Y_{qj}| \) is upper-bounded by the contribution from the endogenous bias:

\[
\sum_{q \neq \{p, p'\}} \sum_j E[|n_j| + |n_q|].
\]

Now consider the remaining part of the Lyapunov function: \( \sum_j (|Y_{pj}| + |Y_{p'j}|) \). Like the proof of Theorem 7 this can be written as

\[
\sum_{j \neq \{p, p'\}} (|Y_{pj}| + |Y_{p'j}|) + 2|Y_{pp'}|.
\]
Note that first part of this expression increases at most by $\sum_{j \notin \{p, p'\}} E[|n_p| + |n_{p'}| + 2|n_j|]$ (by the same argument as above). On the other hand the last term has a drift (following the same derivation for equation A.14)

$$2E[|n_p| + |n_{p'}|] - 2f_{p,p'}(|Y_{pp'}|)|\gamma Y_{pp'}|$$

Hence, the overall Lyapunov drift conditioned on the interaction $I(t) = (p, p')$ is

$$C_3 - 2f_{p,p'}(|Y_{pp'}|)\gamma Y_{pp'}\frac{d}{dt}$$

As this holds for any $p, p'$ interaction, consider the case where agent 1 interacts with any other agent $k$. As $f_{k,1}(|Y_{1k}|) \geq \frac{c}{|Y_{1k}|^{1-\epsilon}}$, so for sufficiently large $|Y_{1k}|$ the conditional drift (conditioned on $I(t) = (1, k)$) is strictly negative. Also, note that as $P(I(t) = (1, k)) > 0$, the contribution to total drift is also strictly negative for sufficiently large $|Y_{1k}|$.

Note that whenever $Y_1$ is outside a sufficiently large compact, for some $k |Y_{1k}|$ is sufficiently large.

A.1.9 Proof of Theorem 12

Consider the same Lyapunov function as in the proof of Theorem 11. By the same argument, conditioned on the interaction $I(t) = (p, p')$, the Lyapunov drift is upper-bounded by

$$C_3 - 2f_{p,p'}(|Y_{pp'}|)\gamma Y_{pp'}.$$
Also, note that as \( P(I(t) = (i,j)) > 0 \) for any \((i,j) \in G'\), the contribution to the total drift (conditional multiplied by \( P(I(t) = (i,j)) \)) is also strictly negative for sufficiently large \( |Y_{1k}| \).

As \( G' \) is connected, there exists a spanning tree \( T \) with all edges having influence \( \Omega \left( \frac{1}{n^2} \right) \). Consider \( Y_T = \{ Y_{ij} : (i,j) \in T \} \) which completely represent \( Y_1 \) (by an invertible linear transformation). If \( Y_1 \) has a sufficiently large \( L_2 \) norm so does \( Y_T \).

Thus, by an argument similar to that in the proof of Theorem 8, there exists a \((k,k') \in T \) such that \( |Y_{kk'}| \) is sufficiently large. Hence, we obtain a strictly negative drift for \( Y_1 \) outside a sufficiently large compact.

### A.1.10 Proof of Theorem 13

W.l.o.g. let us assume that 1 is the root of \( \Lambda \) and that \( n_1 = 0 \) (otherwise we can redefine \( n_j \leftarrow n_j - n_1 \), as we are only interested in opinion difference).

Consider the subtree \( D \) of \( \Lambda \) of depth of 1, which has for root node/agent 1 and with nodes all other agents having directed edges from 1. Let \( Y_D = \{ Y_{ij} : (1,j) \in D \} \). Note that this dynamics is not affected by agents not in \( D \) (as influences are directed). By taking as Lyapunov function \( \sum_{j \in D} |Y_{ij}| \), it can be shown (each term is exactly like the Lyapunov function of a 2-agent biased dynamics) that its drift is upper-bounded by \( 0 < C < \infty \). In fact, from the proof of the 2-agent case, we can show that the upper-bound is \( \sum_{j \in D} E[|n_j|] \). We choose \( C > \sum_{j \in \Lambda} E[|n_j|] \). It can also be shown (by the
same arguments as in the 2-agent case for each \( |Y_{1j}| \) that the drift is upper bounded by \( G = -n^2 C < 0 \) outside a sufficiently large compact (as the drift becomes more and more negative as the compact grows), with \( n \) the number of agents. Hence, we can find a max-norm ball of size \( B > 0 \), such that, for \( B \) sufficiently large, the drift is strictly less than \(-n^2 C\) outside that ball.

We now show by induction that the overall system has a Lyapunov function with a drift upper-bounded and strictly negative outside a max-norm ball. The induction step is as follows. We consider any subtree \( D_1 \) which is rooted at 1 and contains \( D \). We assume that \( Y_{D_1} \) has a Lyapunov function \( L(Y_{D_1}) \) which satisfies the properties of the Lyapunov function of \( \sum_j |Y_{1j}| \) for the dynamics \( Y_D \) (possibly for a different \( B, C \) and \( G \) finite \( G \), say \( B_1, C_1 \) and \( G_1 \) respectively). We then show that if we grow this subtree by adding another agent/node \( a \) (and hence another edge), we obtain a new dynamics \( Y_{D_1,a} \) with a new Lyapunov function which satisfies the same property (for other parameters \( B'_1, C'_1 \) and \( G'_1 \)).

As we have a base case for the induction (dynamics \( Y_D \)), we can show by finite induction (growing the subtree starting from \( D \), by adding agents one by one) the stability of the full dynamics \( Y_1 \equiv Y_\Lambda \).

Consider a subtree \( D_1 \) of \( \Lambda \) s.t. \( D \subset D_1 \). Our induction assumption is that \( D_1 \) has some Lyapunov function \( L(Y_{D_1}) \) which has drift everywhere bounded by \( C_1 \) and such that the drift is strictly less than \( G_1( < 0) \) outside a max-norm ball of size \( B_1 > 0 \). Let \( a \in \Lambda \setminus D_1 \) and such that there exists an \( i \in D_1 \) s.t. \( (i,a) \in \Lambda \). We show that the new system \( Y_{D_1,a} \) has a Lyapunov
function with the same property as $Y_{D_1}$ for some $B_1' = \nu B_1$ ($\nu \geq 1$).

First note that the dynamics $(Y_{D_1}, Y_{1a})$ does not depend on agents not in $D_1 \cup a$ (because any agent outside $D_1 \cup a$ has no influence on agents in $D_1 \cup a$). Also note that $Y_{D_1}$ does not depend on agent $a$ (as $\Lambda$ is a tree $\not\exists i \in D_1$ s.t. $(i, a) \in \Lambda$). $Y_{D_1}$ is a Markov chain and so is $Y_{D_1}$.

For the dynamics $Y_{D_1,a}$ consider the Lyapunov function

$$\widetilde{L}(Y_{D_1,a}) = L(Y_{D_1}) + \hat{L}(Y_{D_1,a}),$$

where $\hat{L}(Y_{D_1,a}) = \max_{j \in D_1 \cup a} |Y_{1j}|$. We have

$$E[\widetilde{L}(Y_{D_1,a}(t+1)) - \widetilde{L}(Y_{D_1,a}(t))|Y_{D_1}(t)] = E[L(Y_{D_1}(t+1)) - L(Y_{D_1}(t))|Y_{D_1}(t)]$$

$$+ E[\hat{L}(Y_{D_1,a}(t+1)) - \hat{L}(Y_{D_1,a}(t))|Y_{D_1,a}(t)]$$

$$\leq C_1 + E[\max_{j \in D_1 \cup a} |n_j|]$$

$$\leq C_1 + \sum_{j \in \Lambda} E[|n_j|]$$

$$\leq C_1 + C := C_1'. \quad (A.15)$$

Note that when $\{Y_{1j} : (1, j) \in D_1 \cup a\}$ is outside a max-norm ball of size $\nu B_1$, with $\nu > 1$, the max-norm of either $\{Y_{1j} : (1, j) \in D_1\}$ (case 1) or $Y_{1a}$ is more than $B_1$ (case 2).

Let us start with preliminary results on these two cases.

If we are in case 1, i.e. $\max_{j \in D_1 \cup a} |Y_{1j}| \geq B_1$, by the induction assumption, we have a negative drift ($< G_1$) for $L(Y_{D_1})$.  

139
If we are in case 2, i.e. \( \max_{j \in D_1 \cup a} |Y_{1j}| \leq B_1 \), then \( |Y_{ia}| > \nu B_1 \) and \( |Y_{ia}| > (\nu - 1)B_1 \). Let us show that we then have a negative drift for \( \tilde{L}(Y_{D_1, a}) \).

We have

\[
E[\tilde{L}(Y_{D_1 \cup a}(t + 1)) - \tilde{L}(Y_{D_1, a}(t))|Y_{D_1, a}(t)]
\]

\[
= P(I(t) = (i, a))E[\tilde{L}(Y_{D_1, a}(t + 1)) - \tilde{L}(Y_{D_1, a}(t))|Y_{D_1, a}(t), I(t) = (i, a)]
\]

\[
+ (1 - P(I(t) = (i, a)))E[\tilde{L}(Y_{D_1, a}(t + 1)) - \tilde{L}(Y_{D_1, a}(t))|Y_{D_1, a}(t), I(t) \neq (i, a)]
\]

\[
\leq P(I(t) = (i, a))E[\tilde{L}(Y_{D_1, a}(t + 1)) - \tilde{L}(Y_{D_1, a}(t))|Y_{D_1, a}(t), I(t) = (i, a)] + E[\max_{j} |n_j|].
\]

We now show that

\[
E[\tilde{L}(Y_{D_1, a}(t + 1)) - \tilde{L}(Y_{D_1, a}(t))|Y_{D_1, a}(t), I(t) = (i, a)]
\]

can be made arbitrarily negative for any \( \nu \) sufficiently large. Note that \( \max_{j \in D_1} |Y_{1j}(t + 1)| \) is upper-bounded by \( B_1 + \max_{j} |n_j| \). Hence, if there no influence from \( i \) to \( a \), then

\[
\max_{j \in D_1 \cup a} |Y_{1j}(t + 1)| \leq \max(|Y_{1a}| + |n_a|, B_1 + \max_{j} |n_j|),
\]

whereas if there is an influence from \( i \) to \( a \), then

\[
\max_{j \in D_1 \cup a} |Y_{1j}(t + 1)|
\]

\[
\leq \max(|Y_{1a}| - |\gamma_{Y_i, a}| + |n_a|, B_1 + \max_{j} |n_j|).
\]
Using this and the fact that $|Y_{1,a}| \geq |\hat{z}(\gamma Y_{i,a})|$, we get

$$
\begin{align*}
&f_{a,i}(|Y_{i,a}|)E[\max(|Y_{1,a}| - |\hat{z}(\gamma Y_{i,a})| + |n_a|, B_1 + \max_j |n_j|)] \\
&+ (1 - f_{a,i}(|Y_{i,a}|))E[\max(|Y_{1,a}| + |n_a|, B_1 + \max_j |n_j|) - |Y_{1,a}|].
\end{align*}
$$

as the maximum of two positive numbers is upper-bounded by their sum. Hence, the conditional drift can be bounded as follows.

$$
\begin{align*}
&\mathbb{E}[\hat{L}(Y_{D1,a}(t+1)) - \hat{L}(Y_{D1,a}(t))|Y_{D1,a}(t), I(t) = (i,a)] \\
&\leq B_1 + \mathbb{E}[\max_j |n_j|] + \mathbb{E}[|n_a| - f_{a,i}(|Y_{i,a}|)|\hat{z}(\gamma Y_{i,a})|] \\
&+ (1 - f_{a,i}(|Y_{i,a}|))E[|Y_{1,a}| + |n_a| - |Y_{1,a}|],
\end{align*}
$$

As, $f_{a,i}(x) = \Omega \left( \frac{1}{x^{1-\epsilon}} \right)$, and $\hat{z}(\gamma y) \geq \frac{\gamma}{2}|y|$ for all $|y| > \frac{2}{\gamma}$, we have

$$
\begin{align*}
f_{a,i}(|Y_{i,a}|)|\hat{z}(\gamma Y_{i,a})| &\geq \gamma'|Y_{i,a}|^\epsilon
\end{align*}
$$

for some $\gamma' > 0$ Note that we can pick a finite $\nu$ (say $\nu_a$, as we add agent $a$) s.t.

$$
B_1 + \mathbb{E}[\max_j |n_j|] + \mathbb{E}[|n_a|] - \gamma'|(\nu - 1)B_1|^\epsilon < G_1 < 0.
$$

We now consider the overall drift for $\tilde{L}$ when $\max_{j \in D_1 \cup a} |Y_{1j}| > \nu_a B_1$.

In case 1,

$$
\begin{align*}
&\mathbb{E}[\tilde{L}(Y_{D1,a}(t+1)) - \tilde{L}(Y_{D1,a}(t))|Y_{D1,a}(t)] \\
&\leq \mathbb{E}[L(Y_{D1}(t+1)) - L(Y_{D1}(t))|Y_{D1,a}(t)]
\end{align*}
$$
\[ + \mathbb{E}[\hat{L}(Y_{D_1,a}(t+1)) - \hat{L}(Y_{D_1,a}(t))|Y_{D_1,a}(t)] \leq G_1 + C. \]

In case 2, by the above preliminary result on \( \hat{L} \)

\[ \mathbb{E}[\tilde{L}(Y_{D_1,a}(t+1)) - \tilde{L}(Y_{D_1,a}(t))|Y_{D_1,a}(t)] \leq \mathbb{E}[L(Y_{D_1}(t+1)) - L(Y_{D_1}(t))|Y_{D_1,a}(t)] + \mathbb{E}[\hat{L}(Y_{D_1,a}(t+1)) - \hat{L}(Y_{D_1,a}(t))|Y_{D_1,a}(t)] \leq C_1 + G_1. \]

Hence, the overall drift for \( \tilde{L}(Y_{D_1,a}) \) outside a max-norm ball of size \( B'_1 = \nu a B \) is bounded above by

\[ G_1' = G_1 + \max(C, C_1) < G_1 + C + C_1. \]

We showed that the drift of \( \tilde{L} \) is upper-bounded everywhere by \( C_1' = C + C_1 \).

This completes the proof by a finite induction argument.

A.1.11 Proof of Theorem 14

The proof of this theorem is based on a dominating Markov chain argument. First consider the Markov chain \( Y(t) = Y_{12}(t) = X_2(t) - X_1(t) \). We consider a simple endogenous noise where \( n_2 - n_1 \in \{\pm 1\} \) w.p. \( p \) being 0 and \( \pm \) with equal probability.
Now we characterize the steady state of the Markov chain $Y(t)$. Note for the Markov chain $Y(t)$ followings are the state transition probabilities,

\[ y \rightarrow y \pm 1 \text{ w.p. } \frac{1}{2}(1 - f_{2,1}(|y|))(1 - f_{1,2}(|y|)) \]

\[ y \rightarrow y - \zeta(\gamma_{1,2}y) \pm 1 \text{ w.p. } \frac{1}{2}(1 - (1 - f_{2,1}(|y|))(1 - f_{1,2}(|y|))) \]

\[ y \rightarrow y - \zeta(\gamma_{2,1}y) \pm 1 \text{ w.p. } \frac{1}{2}f_{2,1}(|y|)f_{1,2}(|y|) \]

Now consider the following Markov chain $\tilde{Y}(t)$ which is a modification of $Y(t)$. The evolution of $\tilde{Y}(t)$ is as follows. Let $p(i, j)$ and $\tilde{p}(i, j)$ be the transition probabilities in $Y(t)$ and $\tilde{Y}(t)$ respectively. Then for $|i| \geq 1$,

\[ \tilde{p}(i, i - \text{sgn}(i)) = \sum_{j:|j|\leq|i-\text{sgn}(i)|} p(i, j) \]

\[ \tilde{P}(i, i + \text{sgn}(i)) = 1 - \sum_{j:|j|\leq|i-\text{sgn}(i)|} p(i, j) \]

In other words, if there is a transition from any state $i$ ($|i| > 0$) to a state $j$ such that $|j| < |i|$ in case of $Y(t)$, then in case of $\tilde{Y}(t)$ the transition is made from $i$ to $i + 1$ or $i - 1$ when $i > 0$ or $i < 0$ respectively. In cases when transitions in $Y(t)$ are from $i$ to $i$ or from $i$ to a state $\text{sgn}(i)(|i| + 1)$, the transition in $\tilde{Y}(t)$ is made to the $\text{sgn}(i)(|i| + 1)$.

Consider the ordering $\leq_{cw}$ as in the proof of Theorem 5. Note that the kernels of $Y(t)$ and $\tilde{Y}(t)$ are ordered in that sense by the construction of $\tilde{Y}$ and the fact that $f_{i,j}$ are non-increasing.

Take any $0 < |y| \leq |\tilde{y}|$ (for $\mathbb{Z}$ this is exactly the $\leq_{cw}$ ordering). Then by the construction of $\tilde{Y}$, $P_Y(|\tilde{y}| \rightarrow |\tilde{y}| - 1)$ is exactly equal to the probability
\( P_Y(|y| \to |y'| \leq |\tilde{y}| - 1) \) which is upper-bounded by \( P_Y(|y| \to |y'| \leq |y| - 1) \). This is because of the following. \( |y| \to |y'| \leq |y| - 1 \) happens either if there is an influence or there is a noise with appropriate sign. As influence is monotone in \(|y|\) (as \( f_{i,j} \) are monotone) and noise is symmetric, the bound holds.

So, we can couple the towards zero transitions in both the processes, as their probabilities are ordered (like the coin tosses for influences in \( Y_1 \) and \( Y' \) are coupled in the proof of Theorem). This will give a sample path dominance in the sense that for \( |\tilde{y}| \geq |y| \) whenever there is a towards zero movement in \( \tilde{Y} \) there is one in \( Y \) and whenever there is an away from zero movement in \( Y \) there is one in \( \tilde{Y} \). Thus like Theorem 5) we get the kernel ordering.

Stochastic dominance between \( Y \) and \( \tilde{Y} \) can also be proved using the following arguments. Let us consider the behavior the chains on \( \mathbb{Z}_+ \).

We use the following result: let \( P \) and \( \tilde{P} \) be two Markov chains on \( \mathbb{Z}_+ \) which admit the representations

\[
Q_{n+1} = Q_n + V(Q_n, U_n), \quad \tilde{Q}_{n+1} = \tilde{Q}_n + \tilde{V}(\tilde{Q}_n, \tilde{U}_n),
\]

(A.16)

with \( U_n \) and \( \tilde{U}_n \) two identically distributed sequences of i.i.d. random variables. Assume that for all \((q, u)\),

\[
V(q, u) \leq \tilde{V}(q, u),
\]

(A.17)

Assume in addition that one of these two chains, say \( \tilde{Q}_n \) is stochastically monotone in the sense that if \( q \leq q' \), then

\[
q + \tilde{V}(q, U) \leq q' + \tilde{V}(q', U).
\]

(A.18)
Then, denoting by $\leq_{st}$ the strong stochastic order, we get that if $Q_0 = \tilde{Q}_0$, then $Q_n \leq_{st} \tilde{Q}_n$ for all $n$. The proof is by induction on $n$. Assume that $Q_n \leq_{st} \tilde{Q}_n$. Then
\[
Q_{n+1} \leq_{st} Q_n + \tilde{V}(Q_n, U_n) \leq_{st} \tilde{Q}_n + \tilde{V}(\tilde{Q}_n, U_n) =_{st} \tilde{Q}_{n+1},
\]
where the first inequality follows from (A.16) and (A.17), and the second inequality follows from (A.18).

Notice that if both chains are ergodic, then, under the above assumptions, their stationary distributions are stochastically ordered too [102]. In our case the stochastic ordering of stationary distribution follows by taking $\tilde{Q}_n$ to be the original chain $Y(t)$ and $Q_n$ to be the constructed chain $\tilde{Y}(t)$ both being restricted to $\mathbb{Z}_+$. Similarly by restricting them to $\mathbb{Z}_-$ the result on the other side of the tail follows. This restriction works because there is no transition in both the chains from $y > 0$ to $y' < 0$ or vice-versa (as $n_1(t) - n_2(t) \in \{\pm 1\}$).

As the transitions in $\tilde{Y}(t)$ are $\pm 1$, this is like a birth-death process, but this is on $\mathbb{Z}$ unlike birth-death processes on $\mathbb{Z}_+ \cup \{0\}$. Note that this Markov chain has a tree like structure, i.e., if we add an (undirected) edge between any two states $(i, j)$ if there is either a transition $i \to j$ or $j \to i$ then the graph obtained for the chain $\tilde{Y}(t)$ is a tree. Hence the detailed balance equations hold true for this Markov chain [103].

Now consider the detailed balance equations for state $y > 0$ such that $\gamma y \geq 2$. Then for these states
\[
y \to y + 1 \text{ w.p. } \leq \frac{1}{2}(1 - \max(f_{2,1}, f_{1,2})(y))
\]
\[ y \rightarrow y - 1 \text{ w.p. } \geq \frac{1}{2} (1 + \max(f_{2,1}, f_{1,2})(y)). \]

This is because \( y \rightarrow y - 1 \) transition in \( Y(t) \) happens when there is an influence (happens with probability \( 1 - (1 - f_{2,1}(|y|))(1 - f_{1,2}(|y|)) \)) or there is no influence but noise is \(-1\) (happens with probability \( \frac{1}{2} (1 - f_{2,1}(|y|))(1 - f_{1,2}(|y|)) \)). Hence, the above transition bounds as

\[
(1 - f_{2,1}(|y|))(1 - f_{1,2}(|y|)) < \min(1 - f_{1,2}(|y|), 1 - f_{2,1}(|y|)).
\]

From a Lyapunov drift analysis (with the square Lyapunov function), it can be seen that this Markov chain is positive recurrent. But here we show positive recurrence by explicitly constructing a steady state distribution (which is also an approach for proving stability). Let \( \pi(\cdot) \) be the steady state distribution. Then

\[
\pi(y) \tilde{p}(y, y + 1) = \pi(y + 1) \tilde{p}(y + 1, y)
\]

\[
\implies \frac{1}{2} (1 + \max(f_{2,1}, f_{1,2})(y)) \pi(y + 1)
\]

\[
\leq \frac{1}{2} (1 - \max(f_{2,1}, f_{1,2})(y)) \pi(y)
\]

\[
\implies \pi(y + 1) \leq \pi(y) \frac{1}{(1 + \max(f_{2,1}, f_{1,2})(y))}.
\]

The rest follows by the fact that \( \max(f_{2,1}(x), f_{1,2}(x)) = \frac{c}{x^\alpha} \) for \( x \geq x_0 > 0 \) and \( \alpha < 1 \).

Thus for any sufficiently large \( y \) we have

\[
\pi(y) \leq \pi(y_0) \prod_{k=y_0}^{y-1} \frac{1}{1 + \max(f_{2,1}, f_{1,2})(k + 1)}
\]

146
\[
\pi(y) \leq \pi(y_0) \left( \left( \frac{1}{1 + \frac{c}{y^\alpha}} \right)^y \right)^{y^{1-\alpha}}.
\]

Thus for some \( y > B > y_0 \) there exists a constant \( 1 < p < 2.71 \) such that
\[
\pi(y + 1) \leq \pi(y_0)p^{-y^{1-\alpha}} \leq e^{-ry^q},
\]
for \( r > 0 \) and \( q \in (0,1) \). This implies that \( \pi(\cdot) \) is summable and hence a stationary distribution exists. Thus
\[
P(\tilde{Y} \geq y) \leq \int_y^\infty e^{-ry^q} = O \left( e^{-ay^b} \right),
\]
for some \( a > 0 \) and \( b \in (0,1) \). Similarly the same bound for \( P(\tilde{Y} \leq -y) \) and in turn for \( P(|\tilde{Y}| \geq y) \) follows. \qed
Appendix B

B.1 Proofs

B.1.1 Proof of Theorem 15

First, we prove the correctness of the algorithm. We do this in two parts. First, we show that Agent Grouping groups the agents as per their opinions. Then we show that given a correct grouping Inferring Opinions infers the opinions correctly.

Lemma 49. $\bigcup_{k=1}^{K} S_k = [n]$ and $i, j \in S_k$ if and only if $x_i = x_j$.

Proof. The proof hinges on the claim that $A_{i,} = A_{j,}$ if and only $x_i = x_j$.

The if part follows because $A_{i,k} = 1$ if and only if $|x_i - x_k| \leq \eta$ and $A_{j,k} = 1$ if and only if $|x_j - x_k| \leq \eta$. As $x_i = x_j$ both the conditions are same and hence $A_{i,k} = A_{j,k}$.

If $x_i \neq x_j$, and $\eta < \frac{L}{2}$, then there exists an $x$ s.t. $|x - x_i| \leq (>\eta$, but $|x - x_j| > (\leq)\eta$. As there is an agent with opinion $x$ (as agents with all opinions are there), $A_{i,} \neq A_{j,}$.

During the iteration, we create a new set for any agent $i$ whose $A_{i,}$ does not match with $A_{j,}$ of any of the agents explored so far. Hence, any agent $i$ must be in one of the groups $S_k$. \qed
Lemma 50. For all \( i \in [n] \), \( \hat{x}_i = x_i \).

Proof. We claim that by Lemma 49, \( K = L \). Otherwise, if \( K > L \) then there must be two agents with the same opinion that have been placed in two different \( S_k \) and \( S_{k'}, k \neq k' \). On the other hand, if \( K < L \) then there must be two agents with different opinions but in the same set \( S_k \). Both of these contradict Lemma 49.

Also, let the opinion of agents in \( S_k \) be \( l_k \).

For \( i, i' \in S_k \), \( A_{i} = A_{i'} \), hence they have same neighbors. Thus if we choose one agent from each \( S_k \), say \( i_k \), then \( i_k \) has edge with agents \( \{i_{k'} : |l_k - l_{k'}| \leq \eta \} \). Hence, \( G_K \) (vertices \( \{1, 2, \cdots, K\} \)) has only those edges for which \( |l_k - l_{k'}| \leq \eta \).

So, \( k, k' \) with \( l_k = 1 \) and \( l_{k'} = L \) have the least degree in \( G_K \), \( \eta \).

So, \( \beta = \eta \). Also note that for neighbors of \( k \) (\( k' \)) their degrees are strictly ordered and degree increases as the opinions of these nodes move away from the extremes (1 and \( L \) respectively).

If \( l_{k}, l_{k'} \) are such \( l_{k'} \) is 1 value closer towards an extreme (1 or \( L \)) compared to \( l_{k} \), then the which is exactly \( \beta \) far from \( l_{k} \) (in a direction opposite to the extreme opinion in consideration) is not a neighbor of \( k_{k'}' \) in \( G_K \) but a neighbor of \( k_{r} \) in \( G_K \).

This implies that we get an order of opinions from the function \( \ell() \) (either increasing or decreasing depending on whether we picked \( k \) with \( l_k = 1 \) or \( k' \) with \( l_{k'} = L \) respectively).

149
Side information at the end chooses the extremes and we keep or flip the order based on our starting opinion \( l_k \) or \( l_{k'} \) (side information resolves between the two).

Note that without the side information we still obtain a correct order up to a reflection about \( L + 1 \).

\[ \square \]

**Lemma 51.** Total computation time is \( O(n^2) \).

**Proof.** To compare \( A_i \) with all the \( S_k \) we need at most \( K \leq L \) comparisons of a pair of \( n \) length vectors. To compare any two \( n \)-length vectors, we need \( n \) operations. Hence, for the algorithm the total computation time is \( O(n^2) \).

This completes the proof of Theorem 15.

### B.1.2 Proof of Theorem 16

The structure of the proof is as follows. We consider few events and show that if all of them happen together then the algorithm infer all the opinions correctly. Then we show that the probability of all the events happening simultaneously is no less than \( 1 - \frac{1}{n^2} \). For computation time, we show that computation time is always upper-bounded by \( cn^2(\log n)^c' \) for some constants \( c,c' \).

First, consider the event \( G \) which is the collection of all the realizations of \( A \) for which the agents are grouped correctly (in the sense that all agents with same opinion are in the same group) by the algorithm StochasticGroup.

\[ G = \{ A : \forall \lambda \in [\Lambda], \forall i,j \in S_\lambda x_i = x_j \} \]
and \( x_i \neq x_j \) if \( i \in S_\lambda, j \not \in S_\lambda \),

and \( \forall l, i \mid \sum_{j:x_j=l} (e_{i,j} - P_{x_i,l}) \leq n^{\frac{1}{2}+\xi', \xi'>0} \}

Consider another event \( E \) which is the collection of all the realizations of \( A \) for which all the estimations \( \hat{P}_{\lambda} \) are \( \frac{\epsilon}{4} \) accurate.

\[
E = \{ A : \forall \lambda, \lambda' a_\lambda \in S_\lambda, a_{\lambda'} \in S_{\lambda'} \}
\[
\frac{1}{|S_{\lambda'}|} \langle A_{a_{\lambda'}}, 1_{S_{\lambda'}} \rangle - P_{x_{a_{\lambda}}, x_{a_{\lambda'}}} < \frac{\epsilon}{4} P_{x_{a_{\lambda}}, x_{a_{\lambda'}}} \}
\]

**Lemma 52.** For \( A \in G \cap E \), \( \hat{x}_i = x_i \) for all \( i \in [n] \).

**Proof.** The proof has some similarity with that of Theorem 15. The proof of the part where we run *Infer Opinions* is exactly same. We show that just before running *Infer Opinions*, under \( G \cap E \), the condition is exactly same as in the proof of Theorem 15.

Note that under \( G \), the grouping of the agents is correct, i.e., if \( x_i = x_j \) then \( i, j \in S_\lambda \) and if \( x_i \neq x_j \), then \( i \in S_\lambda \) and \( j \in S_\lambda' \) for \( \lambda \neq \lambda' \). This follows from the definition of \( G \).

Note that under \( E \), the estimates of \( P_{x_{a_{\lambda}}} \) are correct up to a multiplicative factor of \( \frac{\epsilon}{4} \). Hence, their ratios are correct up to a multiplicative factor of \( \frac{\epsilon}{2} \) (for \( \epsilon \) small). We choose \( 2\epsilon < \min_{p(l,x) > 0; p(l,x') \neq p(l,x)} |\frac{p(l,x')}{p(l,x)} - 1| \). This ensures that if \( \frac{P_{x_{a_{\lambda}}}}{P_{x_{a_{\lambda'}}}} = 1 \) then \( |\frac{P_{x_{a_{\lambda}}}}{P_{x_{a_{\lambda'}}}} - 1| < \epsilon \) and if \( \frac{P_{x_{a_{\lambda}}}}{P_{x_{a_{\lambda'}}}} \neq 1 \) then \( |\frac{P_{x_{a_{\lambda}}}}{P_{x_{a_{\lambda'}}}} - 1| > \epsilon \).

First, consider the case where \( p(l,l \pm z) = p(l,l) \) for some \( |z| > 1 \), i.e., the function is flat around 0 (implying there are some nearby opinions which
agents value equally as their own opinions). In this case, by assumption (iv) \( \exists z' \) with \( L_2 \geq z' > 1, \) s.t. \( p(z') < p(0) \). If we choose \( \epsilon \) as above, for any \( \lambda \) with opinion \( l_\lambda \), all \( \lambda 's \) with opinions in \( |l_\lambda \pm z'| \) are in \( B^1_\lambda \). By condition (i) and (ii) in the assumption, and the choice of \( \epsilon \), \( \lambda ' \in B^1_\lambda \) implies \( \lambda \in B^1_\lambda \), i.e., the graph \( G^1_\Lambda \) is symmetric (directed edges are reciprocated). Also, note that in this case degree of each node is \( \geq 1 \) (but less than \( L_2 \), by assumption iv).

Note that the neighbors of \( \lambda \) are \( \{ \lambda ' : l_{\lambda '} \in [\min(1,l_\lambda - \beta) : \max(l_\lambda + \beta,L)] \} \) for \( \frac{L_2}{2} > \beta \geq 1 \). Also, \( \lambda \) with extreme opinions (1 or \( L \)) have smallest degree. In this case Inferring Opinion is run with \( G^1_\Lambda \).

Rest of the proof is exactly same as that of Theorem 15.

On the other hand, if \( p(l,l+1) < p(l,l) \), then \( G^1_\Lambda \) has degree 0 (under \( G \cap E \) and the above choice of \( \epsilon \)), i.e., it is not connected. Under this condition by the definition of \( B^2_\Lambda \), condition (i) in the assumption, and choice of \( \epsilon \), \( \arg \max_{\lambda ' \in B^1_\lambda } \hat{P}_{\lambda ,\lambda '} = \lambda ' \) s.t. \( |l_{\lambda '} - l_\lambda | = 1 \). Hence, \( G^2_\Lambda \) is connected. Also note that by the assumptions (i), (ii) and (iii), \( G^2_\Lambda \) is symmetric (under \( G \cap E \) and for the choice of \( \epsilon \)), i.e., \( \lambda ' \in B^2_\Lambda \) implies \( \lambda \in B^2_\Lambda \). Also, under \( G \cap E \) and for the above choice of \( \epsilon \), the neighbors of \( \lambda \) in \( G^2_\Lambda \) are \( \{ \lambda ' : l_{\lambda '} \in [\min(1,l_\lambda - \beta) : \max(l_\lambda + \beta,L)] \} \) for some \( \beta \) with \( \frac{L_2}{2} > \beta \geq 1 \) (this \( \beta \) comes from assumption iv). In this case, Inferring Opinion is run with \( G^2_\Lambda \).

Rest of the proof is again exactly same as that of Theorem 15.

\[ \square \]

To complete the proof we need to show that \( P(E \cap G) \geq 1 - \frac{1}{n^2} \). Towards
this we bound the probabilities of $\mathbf{P}(A \notin G)$ and $\mathbf{P}(A \notin E)$.

**Lemma 53.** For any $k \in \mathbb{Z}_+$, there exists a constant $c$ s.t. $|T| = c \log n$ and $\mathbf{P}(A \notin G) \leq \frac{1}{n^k}$.

**Proof.**

\[
\mathbf{P}(G^c) = \mathbf{P}(\{\exists \lambda \in [\Lambda], \exists i, j \in S\lambda \text{ s.t. } x_i \neq x_j\}) \\
\cup \{\exists \lambda \text{ s.t. } x_i = x_j \text{ } i \in S\lambda, j \notin S\lambda\} \\
\cup \{\exists i, l \text{ s.t. } \sum_{j: x_j = l}(e_{i,j} - P_{x_i,l})| > n^{\frac{1}{2}} + \xi\}' \}
\leq \mathbf{P}(\exists \lambda \in [\Lambda], \exists i, j \in S\lambda \text{ s.t. } x_i \neq x_j) \\
+ \mathbf{P}(\exists \lambda \neq \lambda' \text{ s.t. } x_i = x_j \text{ } i \in S\lambda, j \in S\lambda') \\
+ \mathbf{P}(\exists i, l \text{ s.t. } \sum_{j: x_j = l}(e_{i,j} - P_{x_i,l})| > n^{\frac{1}{2}} + \xi\}'\}

Define $G^c_1 = \{\exists \lambda \in [\Lambda], \exists i, j \in S\lambda \text{ s.t. } x_i \neq x_j\}$,

$G^c_2 = \{\exists \lambda \text{ s.t. } x_i = x_j \text{ } i \in S\lambda, j \notin S\lambda\}$.

$G^c_3 = \{\exists i, l \text{ s.t. } \sum_{j: x_j = l}(e_{i,j} - P_{x_i,l})| > n^{\frac{1}{2}} + \xi\}'\}

Note that $G^c_1$ can only happen if there are $i$ and $j$ s.t. $x_i \neq x_j$ but $||d_i - d_j||_1 \leq n^{\frac{1}{2}} + \xi$. Hence,

$G^c_1 \subset \cup_{i,j}\{\exists i, j, x_i \neq x_j, ||d_i - d_j||_1 \leq n^{\frac{1}{2}} + \xi\}$.
Given $i$ and $j$, $x_i \neq x_j$

$$P(|d_i - d_j|_1 \leq n^{\frac{1}{2} + \xi'}) \leq P(\forall W |d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \xi'})$$

Next, to bound this event we show that for any $i$ and $j$ with $x_i \neq x_j$ $\exists W$ s.t. for $c' > 0$,

$$P(|d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \xi'}) \leq \exp(-cn^\xi)$$

Then by union bound this would imply that $P(G_1^c) \leq n^2 \exp(-cn^\xi) \leq \exp(-c_1n^\xi)$.

We show the exponentially small probability of the event $|d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \xi}$ as follows.

Let $P_{l, \cdot} = p(l, \cdot) \Omega(\frac{1}{u(n)})$ for some sub-polynomial $u(n)$, i.e., $u(n) = o(n^\delta)$ for all $\delta > 0$, for $p(l, \cdot) = \Omega(1)$. ¹.

By assumption there exist $\pi_l = \Omega(1), l \in [L], w \leq W$ s.t. there are $n\pi_l$ agents with opinion $l$ and for each $l \neq l'$

$$\left| \sum_{x=1}^{L} \prod_{i=1}^{w} p(l_i, x)(p(l, x) - p(l', x))\pi_x \right| \geq \alpha > 0$$

$$P(\exists W \text{ s.t. } |d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \xi'})$$

¹We stated the theorem for the case where $P_{l, \cdot}$ are $\Omega(1)$, but this proof is done for a more general case where $P_{l, \cdot}$ goes to 0 no faster than an arbitrarily slow polynomial.
\[ \leq P(T \text{ has } \geq W \text{ agents for each opinion}, \exists W \]
\[
\text{s.t. } |d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \frac{\epsilon}{2}}
\]
\[+ P(T \text{ does not have } W \text{ agents with all opinions}) \quad (B.1) \]

First we bound \( P(T \text{ does not have } W \text{ agents with all opinions}) \).

\[
P(T \text{ does not have } W \text{ agents with all opinions})
\leq \sum_{l=1}^{L} P(T \text{ does not have } W \text{ agents with opinion } l)
\leq \sum_{l=1}^{L} P(\cap_{k=1}^{W} \{\text{no opinion } l \text{ in trials } [(k - 1) \left(\frac{|T|}{W}\right) : k \left(\frac{|T|}{W}\right)]\})
\leq \sum_{l=1}^{L} P(\{\text{no opinion } l \text{ in trials } [0 : \left(\frac{|T|}{W}\right)]\})
\leq \sum_{l=1}^{L} \left(1 - \frac{n\pi_l}{|L| - |T|}\right)^\left(|\tau|/W\right),
\]

This is because given there are \( n_l \) agents with opinion \( l \) and there are total \( n \) agents, the chance of picking one is \( \frac{n_l}{n} \). At any time during the picking of \( T \), there are at least \( n\pi_l/|L| - |T| \) agents with opinion \( l \) are left.

Thus,

\[
P(T \text{ does not have } W \text{ agents with all opinions})
\leq \sum_{l=1}^{L} \left(1 - \frac{n\pi_l}{|L| - |T|}\right)^\left(|\tau|/W\right),
\leq \sum_{l=1}^{L} \left(1 - \pi_l/|L| + |T|/n\right)^\left(|\tau|/W\right)
\]

155
\[
\leq \sum_{l=1}^{L} \exp\left(-\frac{\pi_l |T|}{WL} + \frac{|T|^2}{nW}\right) \\
\leq L \exp\left(-\frac{|T| \min_l \pi_l}{WL}\right) \exp\left(\frac{|T|^2}{nW}\right).
\]

Note that \(\min_l \pi_l = \Omega(1)\) as each \(\pi_l = \Omega(1)\). Also, \(|T|^2/n = o\left(\frac{1}{\sqrt{n}}\right)\). Hence, for any \(|T| = \Omega(L \log n)\), we have that

\[
\mathbf{P}(T \text{ does not have } W \text{ agents with all opinions}) \\
\leq \frac{1}{n^k}, k \in \mathbb{Z}_+
\]

Given \(T\) has \(W\) agents for each opinion, for \(x_i = l, x_j = l'\), there exists a set of agents \(W = \{i_1, i_2, \cdots, i_w\}, w \leq W - 1\) with \(x_{i_j} = l_j\). Because in \(T\) there are \(W\) agents with each opinion. So, if we pick any two agents with different opinions (they may even come from \(T\)) we have at least \(W - 1\) agents in \(T\) for each opinion (other than these two), and hence, for the above \(W\)

\[
|\sum_{x=1}^{L} p(l, x) \prod_{i=1}^{w} p(l_i, x)\pi_x - \sum_{x=1}^{L} p(l', x) \prod_{i=1}^{w} p(l_i, x)\pi_x| \geq \alpha.
\]

Let us consider \(d_i(W)\) and define \(d'_i(W)\) to be the number of nodes with opinion \(l\) that form a star with \(i\) and \(W\) as leaves, i.e., \(d_i(W) = \sum_l d'_i(W)\).

Note that the probability an agent \(k (\notin \{i, i_1, i_2, \cdots i_w\})\) has edge to \(\{i, i_1, i_2, \cdots i_w\}\) happens with the following probability

\[
\frac{1}{(u(n))^{w+1}}p(x_i, x_k) \prod_{v=1}^{w} p(x_{i_v}, x_k).
\]

156
Hence,

\[
\mathbb{E}[d^x_i(W)] = \frac{1}{(u(n))^{w+1}} p(x_i, x_k) \prod_{v=1}^{w} p(x_{i_v}, x_k) \pi_x n, \]

up to an error by a constant \(u\) (due to the case where \(k \in \{i, i_1, i_2, \cdots, i_w\}\)).

Note that \(d^x_i(W)\) is a sum of Bernoulli random variables.

Similar expression and arguments are true for \(\mathbb{E}[d^x_j(W)]\) and \(d_j(W)\) respectively.

\[
(d^x_i(W) - \mathbb{E}[d^x_i(W)])
\]

\[
-(d^x_j(W) - \mathbb{E}[d^x_j(W)])
\]

is a zero mean random variable. Moreover, note that \(d^x_i(W)\) can be written as \(\sum_k a_{i,k}(W)\), where \(a_{i,k} = 1\) implies that agent \(k\) has edges to all agents in \(W \cup \{i\}\) (similarly can define \(a_{i,j}\)).

Consider the nodes \([n'] = [n_x] \setminus \{W, i, j\}\), where \(n_x = \pi_x \frac{n}{L}\) is the number of agents with opinion \(x\). Then,

\[
M_{W}(m) = \left( \sum_{k=1}^{m} a_{i,k}(W) - \sum_{k=1}^{m} a_{j,k}(W) \right) - \mathbb{E}[\left( \sum_{k=1}^{m} a_{i,k}(W) - \sum_{k=1}^{m} a_{j,k}(W) \right)]
\]

is a martingale for \(m \leq n'\). This is because \(\{a_{i,k}, a_{j,k}\}\) are independent Bernoulli random variables.
Also, note that $M_W(n')$ is within $\pm W$ of

$$(d^x_i(W) - \mathbf{E}[d^x_i(W)])$$

$$- (d^x_j(W) - \mathbf{E}[d^x_j(W)]).$$

Note that

$$|d_i(W) - d_j(W)|$$

$$\leq \sum_x |d^x_i(W) - d^x_j(W)|$$

By Azuma-Hoeffding inequality

$$\mathbf{P}(|d^x_i(W) - d^x_j(W) - (\mathbf{E}[d^x_i(W)] - \mathbf{E}[d^x_j(W)])| \geq \left( \frac{n}{L} \right) \frac{1}{2} + \xi)$$

$$\leq \exp(- \left( \frac{n}{L} \right) 2^\xi)$$

(B.2)

Hence, by union bound

$$\mathbf{P}(|d_i(W) - d_j(W) - (\mathbf{E}[d_i(W)] - \mathbf{E}[d_j(W)])| \geq L \left( \frac{n}{L} \right) \frac{1}{2} + \xi)$$

$$\leq L \exp(- \left( \frac{n}{L} \right) 2^\xi)$$

Note that

$$|\mathbf{E}[d_i(W)] - \mathbf{E}[d_j(W)]| \geq \frac{(n - W)}{(u(n))^{w+1}} \left( \sum_x p(x_j, x) \prod_{v=1}^w p(x_{i_v}, x) \frac{\pi_x}{L} ight)$$

$$- \sum_x p(x_i, x) \prod_{v=1}^w p(x_{i_v}, x) \pi_x) - W - 2$$

158
\[ \geq \frac{\alpha n}{L(u(n))^{w+1}} - O(1). \]

So,

\[ P(T \text{ has } W \text{ agents for each opinion, } \exists W \]

\[ \text{s.t. } |d_i(W) - d_j(W)| \leq n^{\frac{1}{2} + \frac{\xi}{2}} \]

\[ \leq \exp(-c_0 \frac{n}{\text{sub-poly}(n)}), \]

as \[ |E[d_i(W)] - E[d_j(W)]| \geq \frac{\alpha n}{L(u(n))^{w+1}}. \] This in turn implies that for \( c_1 < c_0, \)

\[ P(G^c_1) \leq \exp(-c_1 n) + \frac{1}{n^k}, k \in \mathbb{Z}_+. \]

Next, we obtain a bound for \( P(G^c_2). \)

Note that for two agents \( i \) and \( j \) with same opinion for any \( W \), there is same chance of forming an edge (simultaneously with \( W \)) to another agent with opinion \( l. \)

Note that \( G^c_2 \) can only happen if there are \( i \) and \( j \) s.t. \( x_i = x_j \) but \[ ||d_i - d_j|| \geq n^{\frac{1}{2} + \xi}. \] Hence,

\[ G^c_2 \subset \cup_{i,j} \{ \exists i, j, x_i = x_j, ||d_i - d_j|| \geq n^{\frac{1}{2} + \xi} \}. \]

\( d^x_i(W) \) and \( d^x_j(W) \) are sum of Bernoulli random variables with same probability of 1 and \( d_i = \sum_{x=1}^L d^x_i. \)

Hence, by Azuma-Hoeffding inequality (again by defining \( a_{i,k}, a_{j,k} \) and martingale \( M_W(m) \) and noting that \( E[d^x_i(W)] = E[d^x_j(W)] \))

\[ P(||d^x_i(W) - d^x_j(W)|| \geq n^{\frac{1}{2} + \frac{\xi}{2}}) \]
\[
\leq \exp(-c'n^\xi)
\]

This in turn implies a similar bound for \(d_i\) and \(d_j\) by union bound:

\[
P(|d_i(W) - d_i(W)| \geq |\mathcal{T}|n^{\frac{1}{2} + \frac{\xi}{2}}) \\
\leq |\mathcal{T}| \exp(-c'n^\xi) \leq \exp(-c''n^\xi)
\]

Note that as \(|\mathcal{T}|\) is sub-poly\((n)\), this is less than \(n^\frac{\xi}{2}\).

There are at most \(|T|^W = O((\log n)^W)\) sets in \(\mathcal{T}\), so by union bound we have that the probability of existence of such a \(W\) for which

\[
|d_i(W) - d_i(W)| \geq n^{\frac{1}{2} + \xi}
\]

for some \(i\) and \(j\) is at most \((\log n)^{W+1} \exp(-c'n^\xi)\). By taking union bound over all such \(i, j\) we have that

\[
P(G_2^c) \leq n^{2} (\log n)^{W+1} \exp(-c'n^\xi) \leq \exp(-c_2 n^\xi).
\]

Note that \(P(G_3^c) \leq \exp(-c_3 n^\xi')\) follows from the Azuma-Hoeffding inequality and union bound. Because, by union bound

\[
P(G_3^c) \leq \sum_{i,l} P\{ \sum_{j:x_j=l} (e_{i,j} - P_{x_i,l}) > n^{\frac{1}{2} + \xi'} \},
\]

and

\[
P\{ | \sum_{j:x_j=l} (e_{i,j} - P_{x_i,l}) > n^{\frac{1}{2} + \xi'} | \} \leq \exp(-c'_3 n^\xi')
\]

by Azuma-Hoeffding inequality. Rest follows because \(nL \exp(-c'_3 n^\xi') \leq \exp(-c_3 n^\xi'), c_3 > 0\).
This in turn proves that $P(G^c) \leq \frac{1}{n^k}, k \in \mathbb{Z}_+$. 

Lemma 54. $P(A \not\in E) \leq \frac{1}{n^k}, k \in \mathbb{Z}_+$.

Proof.

\[
P(E^c) = P(E^c \cap G) + P(E \cap G^c) \geq P(E^c \cap G) + P(G^c)
\]

(B.3)

We next show that $P(E^c \cap G) = 0$.

$E^c$ implies that there exists an agent $a_\lambda \in S_\lambda$ such that

\[
\sum_{j \in S_\lambda} e_{a_\lambda,j} \geq |S_\lambda|(1 + \frac{\epsilon}{4})P_{x_{a_\lambda},x_{a_\lambda}'}.
\]

Now for $G$ we have $|S_\lambda| = \pi_{x_{a_\lambda}} n$ and have that

\[
\left| \sum_{j \in S_\lambda} e_{a_\lambda,j} - \sum_{j \in S_\lambda} E[e_{a_\lambda,j}] \right| < n^{\frac{1}{2}+\xi}
\]

Hence, $E^c \cap G = \emptyset$, as $P_{x_{a_\lambda},x_{a_\lambda}'}|S_\lambda| \geq n^{\frac{1}{2}+\xi}$.

As $P(G^c) \leq \frac{1}{n^k}, k \in \mathbb{Z}_+$, this completes the proof.

Thus we proved that $P(E \cap G) \geq 1 - \frac{1}{n^k}, k \in \mathbb{Z}_+$.

This in turn implies that the algorithm infers correct opinions with probability $\geq 1 - \frac{1}{n^k} \geq 1 - \frac{1}{n^2}$. 

161
Lemma 55. Computation time for the algorithm StochasticGroup followed by StochasticInfer is upper-bounded by $n^{2+\delta}, \forall \delta > 0$.

Proof. In StochasticInfer the only $\Omega(n)$ computation is the estimation of $\hat{P}$ and $L^2$ of them. Hence, StochasticInfer is $O(nL^2)$.

Computing $d_i(W)$ requires $Wn$ computations (product of $W$ vectors of length $n$ each). Also, there are $|\mathcal{T}| \leq |T|^W$ possible $W$s. Hence, for each $i$ we need $\leq W|T|^W n$ computations.

Again for each agent we compare $d_i$ values with $\Lambda$ agents (one for each $S_\lambda$). Each $d_i$ is a vector of size $|\mathcal{T}| \leq |T|^W$. So, each comparison takes time $\leq |T|^W$. Hence, for each $i$ we need $\leq \Lambda|T|^W$ comparisons.

So computation per agent is $\leq W|T|^W n + \Lambda|T|^W$. Note that $|T|^W$ is sub-polynomial in $n$ and hence, we obtain the result for total computation time. \qed

This completes the proof of Theorem 16.

To prove Theorems 18, 19, 21 and 22 the following lemma is useful.

Lemma 56. Let in a population of $n$ agents there are $L = O(poly(\log n))$ types of agents, with each type $l$ being $\frac{n}{L} \pi_l n$ in number, $\pi_l = \Omega(1)$. If we sample $R$ agents uniformly at random, then with high probability ($\geq 1 - \frac{1}{n^k}, k \in \mathbb{Z}_+$) the number of agents of type $l$ in $R$, $\frac{\hat{\pi}_l R}{L}$ satisfies

$$\forall l \in [L] \ |\hat{\pi}_l - \pi_l| < \delta,$$

for any $\delta > 0$, if $R > c_{\delta,k} L \log n$, where $c_{\delta,k}$ depends only on $\delta$ and $k$. 162
Proof. This follows using Azuma-Hoeffding inequality for the sampling process. For \( R = o(n) \), an agent of type \( l \) is picked with a probability \( \frac{\pi_l}{L} \). This is an i.i.d. Bernoulli random variable. So the expected number of type \( l \) agents picked is \( R \frac{\pi_l}{L} \).

On the other hand, \( R \frac{\pi_l}{L} \) is a realization of the sum of the Bernoulli random variables. Hence, for each \( l \) the result follows by direct application of Azuma-Hoeffding for \( R > cL \log n \). Then the lemma follows by taking union bound over all \( L \) (as \( L \) sub-polynomial in \( n \)). \( \square \)

B.1.3 Proof of Theorem 18

This proof follows almost directly from Lemma 56 and the proof of Theorem 15.

Note that \( |U| = \omega(L \log n) \) and hence, with probability \( 1 - \frac{1}{n^k}, k \in \mathbb{Z}_+ \) there are at least \( \frac{\pi_l - \delta}{L} |U| \) agents with opinion \( l \). So, as there are agents of all opinions in \( U \), any two \( \hat{A}_i \) and \( \hat{A}_j \) with \( x_i \neq x_j \) are different. Rest is exactly same as the proof of Theorem 15.

B.1.4 Proof of Theorem 19

We choose \( \delta < \frac{\alpha}{10} \). By Lemma 56 the set \( U \) contains \( n_l \) agents with opinion with \( l \), where w.p. \( 1 - \frac{1}{n^k}, k \in \mathbb{Z}_+, \forall l \)

\[
\frac{\pi_l - \frac{\alpha}{10}}{L} |U| \leq n_l \leq \frac{\pi_l - \frac{\alpha}{10}}{L} |U|.
\]
Note that we have that

\[ P(T \text{ does not have all opinions } W \text{ times}) \leq \frac{1}{n^k}. \]

So from the condition in the theorem we have that for any \( l, l' \) there exists a collection of \( w \) agents in \( T \) s.t.,

\[
\left| \sum_{x=1}^{L} p(l, x) \prod_{i=1}^{w} p(l_i, x) \hat{\pi}_x - \sum_{x=1}^{L} p(l', x) \prod_{i=1}^{w} p(l_i, x) \hat{\pi}_x \right| \geq \frac{4\alpha}{5} > 0,
\]

with probability \( \geq 1 - \frac{1}{n^k} \), where \( \hat{\pi}_l \) is the fraction of agents in \( U \) with opinion \( l \).

Next, we follow the same analysis for agents in \( U \) as we did for agents in \([n]\) in the proof of Theorem 16.

So, we follow the same analysis for \( G^c_1 \) with the difference that instead of \( n \) we have now \( |U| = \phi(n) \) and instead of \( n^{1+\xi} \) we have \( (\phi(n))^{1+\xi} \).

Thus, following the same steps we can show that

\[ P(G^c_1) \leq \exp(-\frac{c_1\phi(n)}{(u(n))^{W+1}L}) + \frac{1}{n^k}. \]

Similarly as in the proof of Theorem 19 we have

\[ P(G^c_2) \leq n^2 |T|^{W+1} \exp(-c_2(\phi(n))^{\xi}). \]

On the other hand for \( G^c_3 \), by union bound

\[ P(G^c_3) \leq \sum_{i,l} \mathbb{P}\{ \sum_{j:x_j=l} (e_{i,j} - P_{x_i,l}) | > n^{1+\xi'} \}, \]

164
and
\[ P\{ \left| \sum_{j:x_j=l} (e_{i,j} - P_{x_i,l}) \right| > n^{\frac{1}{2} + \epsilon'} \} \leq \exp(-c_3' (\phi(n))^{\epsilon'}) \]
by Azuma-Hoeffding inequality. Note that for \( \phi(n)^{\epsilon'} \geq (\log n)^2 \),
\[ \exp(-c_3' (\phi(n))^{\epsilon'}) \leq \exp(-((\log n)(\log n^{c_3'}))) \leq \frac{1}{n^{c_3' \log n}}. \]

Hence, after union bound we have \( P(G_{c_3}^c) \leq \frac{1}{n^k}, \forall k \in \mathbb{Z}_+ \).

The rest of the proof (bounding \( P(E^c) \) and showing correctness for \( G \cap E \)) are exactly the same.

To bound computational complexity note that \( |U| = o(n^\delta), \forall \delta > 0 \) and hence when we compute \( \hat{A}_i, \prod_{w=1}^W \hat{A}_{iw} \), we do at most \( W|U| = o(n^\delta), \forall \delta > 0 \) operations. Hence, the total computation is \( no(n^\delta), \forall \delta > 0 \), implying that the total computation is \( o(n^{1+\delta}), \forall \delta > 0 \).

### B.1.5 Proof of Theorem 21

We first show that opinion recovery is incorrect in a particular group \( U_r \) (among \( R \) groups) with probability \( \leq \frac{1}{n^k}, k \in \mathbb{Z}_+ \). As there are at most \( n \) groups, rest follows from union bound.

Let \( \hat{\pi}_l \) be the corresponding fraction for \( U_r \). As \( |U_r| = \omega((\log n)^{10}) \), by Lemma 56 we have that it is \( \delta \) correct w.p. \( \frac{1}{n^k}, k \in \mathbb{Z}_+ \). When \( \hat{\pi}_l \) is \( \delta \) correct, in group \( U_r \) we have at least one agent with opinion \( l \). Thus following the exact same proof for Theorem 15 we can show that recovery in \( |U_r| \) is correct (w.p. 1) given \( \hat{\pi}_l \) are \( \delta \) correct.
Note that computation is $O(|U_r|^2)$ in each group and hence total computation is $O(n|U_r|^2)$. As $|U_r|$ is sub-polynomial in $n$, so is $|U_r|^2$ and hence the total computation is $o(n^{1+\delta})$, $\forall \delta > 0$.

Note that we collect only $\hat{A}_r$ and each of which has $|U_r|^2$ samples. So total number of samples used is $O(n|U_r|^2) = o(n^{1+\delta})$, $\forall \delta > 0$.

### B.1.6 Proof of Theorem 22

Again, we first show that opinion recovery is incorrect in a particular group $U_r$ (among $R$ groups) with probability $\leq \frac{1}{n^{k}}, k \in \mathbb{Z}_+$. As there are at most $n$ groups, rest follows from union bound.

Again, we have that with probability $\geq 1 - \frac{1}{n^{\pi}}, \hat{\pi}_l$ are $\frac{\alpha}{10}$-correct and the condition in Theorem 16 involving $W$-sized groups is satisfied with $\alpha \leftarrow \frac{4\alpha}{5}$.

Note that for each $U_r$ we have $|T| = c|L|\log n$ chosen randomly from $U_r$ (instead of $[n]$). But note from the derivation of the upper-bound on the probability that it does not contain $W$ agents of each opinion (as in the proof of Theorem 16) we still have the same bound $\frac{1}{n^{\pi}}$. This is because $\frac{|T|^2}{\psi(n)W} = o(1)$ and $|T| = \Omega(L \log n)$.

Rest of the proof for a particular $U_r$ is exactly same as the proof of Theorem 19, because $\psi(n)$ scales similarly as $\phi(n)$ (sub-polynomial but faster than poly-logarithmic).

Time and sample complexity results follow similarly as in the proof of Theorem 21.
Appendix C

C.1 Proofs

In this appendix, we present proofs of the main results in Secs. 5.3.1–4.5. As mentioned earlier, most of these results extend to systems with stationary and ergodic arrival and availability processes, but here we only present results for i.i.d. processes.

C.1.1 Proof of Theorem 27

Here we only prove the converse part, i.e., $\lambda \notin C$ cannot be stabilized by any policy. For the direct part, it is sufficient to prove there exists a scheme that stabilizes any $\lambda \in C$, and so the proof of Thm. 30 below is sufficient.

First we prove that $C$ is a convex subset of $R^N_+$. If $\lambda, \lambda' \in C$, then there exist $(\lambda(u) \in C(u) : u \in Z^M_+)$ and $(\lambda'(u) \in C(u) : u \in Z^M_+)$ such that

$$\sum_u \Gamma(u)\lambda(u) = \lambda, \quad \sum_u \Gamma(u)\lambda'(u) = \lambda'.$$

Thus for any $\gamma \in [0, 1],$

$$\gamma\lambda + (1 - \gamma)\lambda' = \sum_u \Gamma(u)(\gamma\lambda(u) + (1 - \gamma)\lambda'(u)).$$

Note that $C(u)$ is convex since it is the convex hull of $C(u)$; hence $\gamma\lambda(u) + (1 - \gamma)\lambda'(u) \in C(u)$, which in turn implies $\gamma\lambda + (1 - \gamma)\lambda' \in C$. This proves
Thus $\bar{C}$ is a closed convex set. Hence for any $\lambda^O \notin \bar{C}$, there exists a hyperplane $h^T x = c$ that separates $\bar{C}$ and $\lambda^O$, i.e., there exists an $\epsilon > 0$ for any $\lambda \in \bar{C}$ such that $h^T \lambda^O \geq h^T \lambda + \epsilon$.

Hence under any policy:

$$E[h^T Q(t + 1)] = E[h^T (Q(t) + A(t) - D(t))]$$

$$= E[h^T |Q(t) + A(t) - \Delta(t)|^+]$$

where $\Delta(t)$ is the number of possible departure under the scheme if there were infinite number of jobs of each type, and $| \cdot |^+$ is shorthand for $\max(\cdot, 0)$. As $\cdot$ is a convex function of $x$, $h^T |Q(t) + A(t) - \Delta(t)|^+$ is a convex function of $Q(t), A(t)$, and $\Delta(t)$. Thus by Jensen’s inequality:

$$E[h^T |Q(t) + A(t) - \Delta(t)|^+] \geq h^T |E[Q(t)] + E[A(t)] - E[\Delta(t)]|^+$$

Note that any $\lambda$ is a $\Gamma(u)$-combination of some $\{\lambda(u) \in C(u)\}$ and any $\lambda(u)$ is some convex combination of elements of $C(u)$. Also, from the allocation constraints it is apparent that if $a \in C(u)$ then also $a' \in C(u)$ if $a' \leq a$. These two imply that for any $\lambda \in \bar{C}$, if there exists a $\lambda' \leq \lambda$ (component-wise) and $\lambda' \geq 0$, then $\lambda' \in \bar{C}$. That is $C$ is coordinate convex. This in turn implies that for any $\lambda^O \notin \bar{C}$ there exists an $h \neq 0 \in \mathbb{R}^N_+$ such that hyperplane separation holds for this $h$. Thus for $h \geq 0$:

$$E[h^T Q(t + 1)] \geq h^T |E[Q(t)] + E[A(t)] - E[\Delta(t)]|^+$$
\[ \sum_{j} |h_j E [Q_j(t)] + h_j E [A_j(t)] - h_j E [\Delta_j(t)]| \]
\[ \geq \sum_{j} (h_j E [Q_j(t)] + h_j E [A_j(t)] - h_j E [\Delta_j(t)]) \]
\[ \geq E [h^T Q(t)] + h^T E [\lambda^O] - \sup_{\lambda \in \mathcal{E}} h^T \lambda \]
\[ \geq E [h^T Q(t)] + \epsilon \]

Thus we have \( E[h^T Q(t + 1)] \to \infty \). As \( h \geq 0 \), this implies there exists \( j \) such that \( E[Q_j(t)] \to \infty \) as \( t \to \infty \). Hence, the system is not stable.

### C.1.2 Proof of Theorem 28

Consider the dynamics of \( Q_{j,s}(t) \), the unallocated \((j, s)\) tasks at the end of epoch \( t \).

\[
Q_{j,s}(t + 1) = |Q_{j,s}(t) + A_j(t) - D_{j,s}(t)|^+
\]
\[ \geq Q_{j,s}(t) + A_j(t) - D_{j,s}(t) \]
\[ \geq \sum_{k=0}^{t} (A_j(t) - D_{j,s}(t)) . \]

As \( r_{j,s} \geq 0 \),

\[
E[r_{j,s}Q_{j,s}(t + 1)] \geq \sum_{k=0}^{t} E[A_j(t)r_{j,s} - r_{j,s}D_{j,s}(t)]
\]
\[ = \sum_{k=0}^{t} (\lambda_j r_{j,s} - r_{j,s}E[D_{j,s}(t)]) \]

Consider any set \( J \subset [N] \), then at any epoch \( t \), to schedule a certain number of tasks of type \((j, s)\), the system needs that much available usable
skill-hours. This follows from conditions (4.1) and (4.2) and can be written as:

\[ \sum_{j \in J} r_{j,s} D_{j,s}(t) \leq \sum_{l \in N(J)} \sum_{i \in [M^l]} h_{i,s}^l \mu_{i,s}^l. \]

This in turn implies

\[ \sum_{j \in J} r_{j,s} E[D_{j,s}(t)] \leq \sum_{l \in N(J)} \sum_{i \in [M^l]} h_{i,s}^l \mu_{i,s}^l. \]

Hence,

\[ E \left[ \sum_{j \in J} r_{j,s} Q_{j,s}(t + 1) \right] \geq \sum_{k=0}^{t} \left( \sum_{j \in J} \lambda_j r_{j,s} - \sum_{j \in J} r_{j,s} E[D_{j,s}(t)] \right) \]

\[ \geq \sum_{k=0}^{t} \left( \sum_{j \in J} \lambda_j r_{j,s} - \sum_{l \in N(J)} \sum_{i \in [M^l]} h_{i,s}^l \mu_{i,s}^l \right) \quad (C.1) \]

For any \( \lambda \notin C_{\text{out}} \), by definition there exists a \( J \subset [N] \) such that \( \sum_{j \in J} \lambda_j r_{j,s} - \sum_{l \in N(J)} \sum_{i \in [M^l]} h_{i,s}^l \mu_{i,s}^l > 0 \). Thus in that case, \( \limsup_{t \to \infty} E \left[ \sum_{j \in J} r_{j,s} Q_{j,s}(t + 1) \right] = \infty \). Note that since \( J \) is finite and so is \( \max_j r_{j,s} \), there exists a \( j \in J \) such that \( \limsup_{t \to \infty} E \left[ Q_{j,s}(t + 1) \right] = \infty \). This shows the system is not stable for \( \lambda \notin C_{\text{out}} \) and proves \( C_{\Gamma} \subset C_{\text{out}} \).

### C.1.3 Proof of Proposition 29

We consider FD, FND, and IND cases separately.

**FD:** The MaxWeight part chooses \( z_{j,s}^l \) to be integral which implies that integral number of tasks can be allocated if done appropriately. As hours are allocated from tasks in order, a task later in the order only gets allocated
(partially or fully) after the tasks before it are fully allocated. This leads to no partially-allocated tasks.

**FND:** Same ordering is used for all \((j, s)\) tasks and MaxWeight chooses \(z_{j,s}^l\) such that \(a_{j,s} = a_{j,s'}\) (as it satisfies FND) and hence if an \(s\)-task of a job is chosen then also \(s'\) is chosen for \(r_{j,s}, r_{j,s'} > 0\).

**IND:** Same ordering is used for all \((j, s)\) tasks, allocations to different categories are in same order \((l = 1 \text{ to } L)\) and MaxWeight chooses \(z_{j,s}^l\) such that \(z_{j,s}^l = z_{j,s'}^l\) for all \(l, l'\) (as it satisfies IND), hence if a task of a job is allocated to category \(l\) then so are the other tasks.

### C.1.4 Proof of Theorem 30

Note that MaxWeight chooses an allocation \(\{\hat{\Delta}_{j,s}(t)\}\). But, the maximum number of \((j, s)\)-tasks that can be served is \(Q_{j,s}(t) + A_j(t)\). By Prop. 29, Task Allocation does a feasible allocation for FD, FND, and IND systems. Also, note that in the Task Allocation algorithm, the number of allocated \((j, s)\)-tasks is \(\hat{D}_{j,s} = \min(\hat{\Delta}_{j,s}(t), Q_{j,s}(t) + A_j(t))\).

Consider the usual Lyapunov function \(L(Q) = \sum_{j,s} Q_{j,s}^2\).

\[
\mathbb{E}[L(Q(t + 1)) - L(Q(t)) | Q(t)] = \mathbb{E} \left[ \sum_{j,s} (Q_{j,s}^2(t + 1) - Q_{j,s}^2(t)) | Q(t) \right] = \mathbb{E} \left[ \sum_{j,s} \left( (Q_{j,s}(t) + A_j(t) - \hat{D}_{j,s})^2 - Q_{j,s}^2(t) \right) | Q(t) \right]
\]
\[
E \left[ \sum_{j,s} \left( (Q_{j,s}(t) + A_j(t) - \hat{\Delta}_{j,s})^2 - Q_{j,s}^2(t) \right) \right] |Q(t) | \\
\leq E \sum_{j,s} Q_{j,s}(t) A_j(t) |Q(t) | - E \sum_{j,s} Q_{j,s}(t) \hat{\Delta}_{j,s}(t) |Q(t) | + E \sum_{j,s} (A_j^2(t) + \hat{\Delta}_{j,s}^2) |Q(t) |
\]

We bound the last term first.

\[
E \left[ \sum_{j,s} (A_j^2(t) + \hat{\Delta}_{j,s}^2) |Q(t) | \right] = \sum_{j,s} E[A_j^2] + E \left[ \sum_{j,s} \left( r_{j,s} \hat{\Delta}_{j,s} \right)^2 \right] \\
\leq \sum_{j,s} E[A_j^2] + \frac{1}{\min(r_{j,s} > 0)} E \left[ \left( \sum_{j,s} r_{j,s} \hat{\Delta}_{j,s} \right)^2 \right] \\
\leq \sum_{j,s} E[A_j^2] + \frac{1}{\min(r_{j,s} > 0)} E \left[ \left( \sum_{i,l} \left( \sum_{s} h_{i,s}^l \right) U_{i,l}^l \right)^2 \right] \\
\leq \sum_{j,s} E[A_j^2] + \frac{(\max_{i,l,s} h_{i,s}^l)^2}{\min(r_{j,s} > 0)} \max_{i} E \left[ (U_{i}^l)^2 \right] \frac{M(M + 1)}{2} \\
\]

(C.2)

(C.3)

This is a constant \( B < \infty \) independent of \( Q \), as \( E[A_j^2] \) and \( E \left[ (U_{i}^l)^2 \right] \) are finite for all \( j, l, i \).

To bound the first term, note that if \( \lambda + \epsilon 1 \in \mathcal{C} \), then there exist \( \{ \nu(u) \in \mathcal{C}(u) \} \) such that \( \lambda_j \leq \sum_u \Gamma(u) \nu(u) - \epsilon \) for all \( j \in [J] \). Again note that as \( \mathcal{C}(u) \) is the convex hull of \( C(u) \), \( \nu(u) = \sum_k \gamma_k d^k(u) \) for some \( \{ d^k(u) \in C(u) \} \) and \( \gamma_k \geq 0 \) with \( \sum_k \gamma_k = 1 \). So,

\[
E \left[ \sum_{j,s} Q_{j,s}(t) A_j(t) |Q(t) | \right] = \sum_{j,s} Q_{j,s}(t) \lambda_j \\
\leq \sum_{j,s} Q_{j,s}(t) \nu_j - \epsilon \sum_{j,s} Q_{j,s}
\]

172
\[
\leq \sum_{j,s} Q_{j,s}(t) \sum_u \Gamma(u) \sum_k \gamma_k d_j^k(u) - \epsilon \sum_{j,s} Q_{j,s}
= \sum_{u} \Gamma(u) \sum_{j,s} Q_{j,s}(t) \sum_k \gamma_k d_j^k(u) - \epsilon \sum_{j,s} Q_{j,s}
\leq \sum_{u} \Gamma(u) \max_{d(u) \in C(u)} \sum_{j,s} Q_{j,s}(t) d_{j,s}(u) - \epsilon \sum_{j,s} Q_{j,s}
= E \left[ \sum_{j,s} Q_{j,s}(t) \Delta_{j,s}(t) \left| Q(t) \right| \right] - \epsilon \sum_{j,s} Q_{j,s}.
\]

Thus, we have a bound on the Lyapunov drift,

\[
E \left[ L(Q(t+1)) - L(Q(t)) \right| Q(t) \right] \leq B - \epsilon \sum_{j,s} Q_{j,s}.
\]

Hence,

\[
E \left[ L(Q(T)) - L(Q(0)) \right] \leq BT - \epsilon \sum_{t=0}^{T-1} \sum_{j,s} w_{j,s} E[Q_{j,s}(t)].
\]

As \(L(Q(0)) < \infty\) and \(L(Q) \geq 0\) for all \(Q\), we have that for all \(T\),

\[
\frac{1}{T} \sum_{t=0}^{T-1} \sum_{j,s} E[Q_{j,s}(t)] \leq \frac{B}{\epsilon} + \frac{L(0)}{T} < \infty
\]

This in turn implies \(\limsup_{t \to \infty} \sum_{j,s} E[Q_{j,s}(t)] < \infty\), otherwise the time-average cannot be finite. This implies that for all \((j,s)\) with \(r_{j,s} > 0\), \(\limsup_{t \to \infty} E[Q_{j,s}(t)] < \infty\).

Again note that \(Q_j(t) \leq \sum_s Q_{j,s}\), as there can be unallocated jobs with more than one part unallocated. Hence, \(\limsup_{t \to \infty} E[Q_j(t)] < \infty\) for all \(j \in [N]\).
C.1.5 Proof of Theorem 33

In the GreedyAgent algorithm, as each agent with available skill-hours greedily chooses to serve a task, no \((j, s)\) task of size \(r\) can remain unallocated if there is an agent (or agents) with \(s\) skill-hour (total) of at least \(r\). Since at each allocation epoch a task should either be allocated totally or not at all (i.e., \(x < r\) hours cannot be allocated), it may happen that some agent hours are wasted, as that does not meet the task allocation requirement.

Note that since any job requirement is less than \(\bar{r} = \max_{j,s} r_{j,s}\), at most \(\bar{r}\) agent-skill-hours can be wasted.

Let \(H_s(t)\) be the process of unallocated job-hours for skill \(s\) after the allocation at epoch \(t\). Then for all \(t\),

\[
H_s(t + 1) \leq H_s(t) + \sum_j A_j(t) r_{j,s} - \sum_i U_i(t) h_{i,s} + \bar{r}. \]

This implies that process \(G_s(t)\) given by \(G_s(t + 1) = |G_s(t) + \sum_j A_j(t) r_{j,s} - \sum_i U_i(t) h_{i,s} + \bar{r}|^+\) bounds \(H_s(t)\).

\(G_s(t)\) has dynamics of a queue with arrival process \(X_s(t) = \sum_j A_j(t) r_{j,s} + \bar{r}\) and departure process \(Y_s(t) = \sum_i U_i(t) h_{i,s}\). Let \(A_j(\theta) = \mathbb{E}[e^{\theta A_j(t)}]\) and \(U_i(\theta) = \mathbb{E}[e^{\theta U_i(t)}]\) for \(j \in [N]\) and \(i \in [M]\).

For \(\theta \in \mathbb{R}\), then,

\[
\mathbb{E}[e^{\theta(X_s(t) - Y_s(t))}] = \mathbb{E}[e^{\theta X_s(t)}] \mathbb{E}[e^{-\theta Y_s(t)}] = \mathbb{E}[e^{\theta \sum_j A_j(t) r_{j,s} + \bar{r}}] \mathbb{E}[e^{-\theta \sum_i U_i(t) h_{i,s}}]
\]
\[= e^{\theta \bar{r}} \prod_j E[e^{\theta A_j(t)r_{j,s}}] \prod_i E[U_i(t)h_{i,s}] \] (C.4)

\[= e^{\theta \bar{r}} \prod_j A_j(\theta r_{j,s}) \prod_i U_i(-\theta h_{i,s}) \]

\[= \exp \left( \theta \bar{r} + \sum_j \log A_j(\theta r_{j,s}) + \sum_i \log U_i(-\theta h_{i,s}) \right). \]

First consider the Gaussian-dominated case. Since the process variance is no more than the mean and the moment generating function of the variance is upper-bounded by that of a zero-mean Gaussian:

\[\log A_j(\theta r_{j,s}) \leq \lambda_j \theta r_{j,s} + \lambda_j (\theta r_{j,s})^2 \]
\[\log U_i(-\theta h_{i,s}) \leq -\mu_j \theta h_{i,s} + \mu_j (\theta h_{i,s})^2. \] (C.5)

Note that for any two functions \(k_1x^2\) and \(k_2x\), \(\lim_{x \to 0} k_2x/k_1x^2 = \infty\), and hence for any \(\epsilon \in (0,1)\) there exists \(x^* > 0\) such that for all \(x < x^*\), \(k_1x^2/k_2x < \epsilon\). Hence for any \(\epsilon \in (0,1)\), there exist \(\theta^*_{j,s}, \theta^*_{i,s} > 0\), for all \(i, j, s\) such that for all \(\theta < \theta^* = \min_{i,j,s}(\theta^*_{j,s}, \theta^*_{i,s})\),

\[\log A_j(\theta r_{j,s}) \leq \lambda_j \theta^* r_{j,s}(1 + \epsilon) \] (C.5)
\[\log U_i(-\theta h_{i,s}) \leq -\mu_i \theta h_{i,s}(1 - \epsilon) \] (C.6)

Note that since \(N, S, M\) are finite and \(\theta^*_{j,s}, \theta^*_{i,s} > 0\), for all \(i, j, s, \theta^* > 0\). Moreover, note that \(\theta^*\) does not depend on \(\lambda, \mu\) since the ratio of the linear and quadratic terms in the log moment generating functions are independent of \(\lambda\) and \(\mu\).
As \( e^\theta - 1 = \sum_{k=1}^{\infty} \frac{\theta^k}{k!} \), for the Poisson-dominated case we have

\[
\log \mathcal{A}_j(\theta r_{j,s}) \leq \lambda_j \sum_k \frac{(\theta r_{j,s})^k}{k!}
\]

\[
\log \mathcal{U}_i(-\theta h_{i,s}) \leq \mu_j \sum_k \frac{(-\theta h_{i,s})^k}{k!}
\]

Again, by the same argument, we can have a \( \theta^* \) for which (C.16) and (C.17) are satisfied. Thus, for all \( \theta < \theta^* \) we have:

\[
\mathbf{E}[e^{\theta(X_s(t) - Y_s(t))}] \leq \exp \left( \theta \bar{r} + \sum_j \lambda_j \theta^* r_{j,s} (1 + \epsilon) - \sum_i \mu_i \theta h_{i,s} (1 - \epsilon) \right)
\]

\[
\leq \exp \left( \theta \left( \bar{r} - \sum_i \mu_i h_{i,s} (\alpha - \epsilon) \right) \right).
\] (C.7)

Note (C.18) follows from the fact \( \lambda \in (1 - \alpha) \mathbb{C}^{out} \). As \( \epsilon > 0 \) can be chosen arbitrarily small, we can have \( \alpha - \epsilon > 0 \). Since \( \sum_i \mu_i h_{i,s} > \sum_j (1 - \alpha) \lambda_j r_{j,s} \) and \( \sum_j \lambda_j r_{j,s} \) scales with \( \lambda \mathbf{N} \), for sufficiently large \( \lambda \) with \( \lambda_j \geq \lambda_\alpha \) for all \( j \), we have \( \bar{r} - \sum_i \mu_i h_{i,s} (\alpha - \epsilon) \leq -\gamma \sum_i \mu_i h_{i,s} (\alpha - \epsilon) \), for some \( \gamma > 0 \). Thus, we have for some \( \theta > 0 \),

\[
\mathbf{E}[e^{\theta(X_s(t) - Y_s(t))}] \leq \exp (-\theta K), \quad (C.8)
\]

where \( K \) scales with \( \lambda \).

Now by Loynes’ construction:

\[
G_s(t) = \max_{\tau < t} \sum_{k=\tau}^{t} (X_s(t) - Y_s(t)).
\]

Hence,

\[
\mathbf{P}(G_s > g) \leq \frac{\mathbf{E}[e^{\theta G_s}]}{e^{\theta g}}
\]

176
This in turn implies

\[
\mathbb{P}\left( \max_s G_s > c \log N \right) \leq S \mathbb{P}\left( G_s > c \log N \right)
\]

\[
\leq \frac{Se^{-\theta c \log N}}{1 - \exp(-\theta K)}
\]

\[
\leq \frac{S}{N^{\theta c}} \frac{1}{1 - \exp(-\theta K)}.
\]

Note that the total number of unallocated jobs in the system is upper-bounded by \((\min_{j,s} r_{j,s} > 0) \sum_s G_s\), as any unallocated job has at least \(\min_{j,s} r_{j,s} \) skill-hours unallocated. Hence the total number of unallocated tasks in the system, \(Q\), satisfies:

\[
\mathbb{P}(Q > cS \log N) \leq \mathbb{P}\left( \sum_s G_s > c \min_{j,s} r_{j,s} S \log N \right)
\]

\[
\leq \mathbb{P}\left( \max_s G_s > c \min_{j,s} r_{j,s} \log N \right) \frac{S}{N^{\theta c} \min_{j,s} r_{j,s} > 0} \frac{1}{1 - \exp(-\theta K)}.
\]

Note that \(\theta^* > 0\) does not depend on \(\lambda\), so \(\frac{1}{1 - \exp(-\theta K)}\) is \(O(1)\). As \(S = O(N)\), we can choose a \(c > 0\) such that \(\theta^* c \min_{j,s} r_{j,s} > 0 \quad r_{j,s} > 3\) and hence, \(\mathbb{P}(Q > cS \log N) \leq o(N^{-2})\).
C.1.6 Proof of Proposition 34

We prove this proposition by constructing a simple (but general) system and show that the system is not stable via a domination argument with a Markov chain that is not positive recurrent.

Consider \( N = 1, \ M = 2, \) and \( S > 1 \) being even. Let \( r = 1, \ h_1 = (1, 1, \cdots, \frac{S}{2} \text{ terms, 0, 0, } \cdots) \) and \( h_2 = (0, 0, \cdots, \frac{S}{2} \text{ terms, 1, 1, } \cdots) \). Let the arrival to the system be i.i.d. with

\[
\begin{cases}
(1 - \alpha - \delta)\lambda, & \text{w.p. } (1 - \epsilon) \\
2\lambda, & \text{w.p. } \epsilon,
\end{cases}
\]

such that \((1 + \alpha)\epsilon - \delta(1 - \epsilon) = 0\), resulting in mean arrival rate \((1 - \lambda)\) and variance < \(\lambda\). One possible construction is to take \(\epsilon = \frac{1}{2\lambda}\) and \(\delta\) accordingly. The agent-availability process is considered to be \(U_1 = U_2 = \lambda\). Note that both arrival and agent availability processes are Gaussian-dominated as well as Poisson-dominated.

First, consider the case where greedy picking of tasks by the agents may be adversarial. Each agent picks all the tasks that it can take from the job and so agents of different types pick from a different half of the \(S\) skill-parts. Thus if there are \(\geq 2\lambda\) jobs, in worst case (where adversary gives the job to the agent) agents of type 1 may pick \(\frac{S}{2}\) job-parts of \(\lambda\) jobs, while agents of type 2 pick parts of other \(\lambda\) jobs. Hence no job is actually allocated at that allocation epoch. By the next epoch at least \((1 - \alpha)\lambda\) jobs have come and if the agents pick jobs in an adversarial manner, again no job is actually allocated. Thus the number of unallocated jobs keep growing after it hits \(2\lambda\) once. Note
that since there is a positive probability of $\geq 2\lambda$ arrivals, with strictly positive probability, the number of unallocated jobs grows without bound.

Next, we prove the case where greedy picking of the tasks by the agents is random, i.e., an agent picks all $S/2$ parts of a randomly selected job (without replacement). As arrivals and availability are i.i.d., we can describe the number of unallocated jobs by a Markov chain $Q(t)$. Note that for $Q \leq \lambda$, $P(Q \rightarrow 0) = 1$. On the other hand, $0 < P(0 \rightarrow Q) < 1$ for $Q < 2\lambda$.

Consider any $Q = n\lambda$ for $n > 1$. Note that $P(Q \rightarrow x) = 0$ for $x < (n-1)\lambda$ as no more than $\lambda$ jobs can be scheduled, because there are $\lambda$ agents of each type. Again,

$$P(Q \text{ decreases at least by } 1) \leq 1 - \left(1 - \frac{\lambda}{Q + \lambda - \alpha\lambda}\right)^\lambda$$

because there are at least $(1 - \alpha)\lambda$ arrivals and each type picks $\lambda$ agents randomly and this is the probability that the picked sets have a non-empty intersection. Again, as there are $2\lambda$ arrivals w.p. $\geq \epsilon$ we have

$$P(Q \text{ increases by at least } \lambda) \geq \epsilon \left(1 - \frac{\lambda}{Q + 2\lambda}\right)^\lambda.$$ 

Based on computation of transition probabilities for each transition, it follows that for all $k \geq 1$, $P(Q \text{ decreases at least by } 1)$ as well as $P(Q \text{ decreases by } k)$ is decreasing with $Q$, whereas $P(Q \text{ increases by } k)$ increases.

Hence, we can dominate the above chain by another chain $\hat{Q}$ on $\lambda\mathbb{Z}^+$ with transition probabilities

$$P(\lambda n \rightarrow \lambda(n + 1)) = \epsilon \left(1 - \frac{1}{n + 1}\right)^\lambda$$
\[ P(\lambda(n+1) \rightarrow \lambda n) = 1 - \left(1 - \frac{1}{n+1}\right)^\lambda. \]

Now the chain \( \hat{Q}(t) \) is a birth-death chain. If it has a finite (summable over states) invariant, then that is unique. We first assume that the invariant is \( \pi \) and then show that it is not summable to prove that it is not positive recurrent.\(^1\) As this is a birth-death chain the invariant measure must satisfy:

\[ \pi(n+1) = \pi(n) \frac{\epsilon \left(1 - \frac{1}{n+1}\right)^\lambda}{1 - (1 - \frac{1}{n+1})^\lambda}. \]

Since

\[ \frac{\epsilon \left(1 - \frac{1}{n+1}\right)^\lambda}{1 - (1 - \frac{1}{n+1})^\lambda} \rightarrow \infty \]

as \( n \rightarrow \infty \) for any finite \( \lambda > 0 \), this shows that \( \pi \) is not finite.

\[ \Box \]

C.1.7 Proof of Theorem 35

Consider the different types of unallocated jobs. These are given by \[ \{Q_j(t) : j \in [N]\}. \]

Consider the following processes: for each \( s \in [S], Q^s(t) = \sum_{j: r_{j,s} > 0} Q_{j}r_{j,s} \), which represent the number of unserved hours of skills \( s \) over all jobs.

We now construct another process \( \tilde{Q} \) s.t. it dominates the process \( \sum_s Q^s \). So, if we can show upper-bound on \( \tilde{Q} \), then the same bound applies

---

\(^1\) An alternate proof follows from noting that if we take a Lyapunov \( \hat{Q} \) itself, then it has bounded jumps and it is easy to check that after certain \( Q > 0 \) the drift is strictly positive, and invoke the Foster-Lyapunov (converse) theorem for irreducible chain with bounded absolute drift.
for $\sum_s Q^s$. Hence, in turn we get a bound for $\{Q_j(t)\}$ (as $\min\{r_{j,s} > 0\} = \Theta(1)$ by the assumption that $\{r_{j,s}\}$ do not scale with the system size).

Towards constructing a suitable $\tilde{Q}$ we make the following observation about the dynamics of $Q^s$ and $\{Q_j\}$. At each time $t$, $\sum_j A_{j,1}(t)r_{j,s}$ amount of $s$ skill hour is brought to add to $Q^s$. Also, this queue gets some service depending on the available agent hours.

At time $t$, $\sum_m U_m(t)h_{m,s}$ s-skill hour of service is brought by the agents.

For a job to be allocated, all tasks of it must find an allocation. Hence, for a job in type $j$-job to find an allocation it must get $r_{j,s}$ hours of service from each skill $s$. Thus at any time $t$ any skill $s$ queue gets a service of at least

$$\min_{s \in [S]} \sum_m U_m h_{m,s} - \bar{r},$$

where $\bar{r} = \max\{r_{j,k,s}\}$. This is because of the following. For each skill $\sum_s U_m h_{m,s}$ hour is available. Note that a job can be allocated if all its tasks find allocations, converse of which is also true. That is if all tasks of a step found allocation then the step can be allocated. As $\min_{s \in [S]} \sum_s U_m h_{m,s}$ hours of service is brought by the agents for each skill, at least $\min_{s \in [S]} \sum_s U_m h_{m,s} - \bar{r}$ of $s$-skill hours are served (because a maximum of $\bar{r}$ can be wasted, as no task is of size more than $\bar{r}$).

Also, note that the amount of required service brought to the queue $Q^s$ at time $t$ is upper-bounded by

181
\[
\max_{s \in [S]} \sum_j A_j(t) r_{j,s}
\]

Consider a process \( \tilde{Q}^s \) with evolution

\[
\tilde{Q}^s(t + 1) = \max(\tilde{Q}^s(t) + \max_{s \in [S]} \sum_j A_j(t) r_{j,s} - \min_{s \in [S]} \sum_m U_m h_{m,s} + \bar{r}, 0).
\]

Note that given \( \tilde{Q}^s(t_0) \geq Q^s(t_0) \) at some \( t_0 \), the same holds true for all \( t \geq t_0 \).

This is because for \( x, a, b \geq 0 \) and \( x', a', b' \geq 0 \), with \( x \geq x', a \geq a' \) and \( b \leq b' \)

\[
\max(x + a - b, 0) \geq \max(x' + a' - b', 0),
\]

and hence, the monotonicity propagates over time.

Thus, to bound \( \sum_s Q^s \) it is sufficient to bound \( \sum_s \tilde{Q}^s(t) \). Note that each of \( \tilde{Q}^s \) has exactly same evolution, so let us consider

\[
\tilde{Q} := S \tilde{Q}^1,
\]

which bounds \( \sum_s Q^s \).

From the evolution,

\[
\tilde{Q}(t + 1) = \max(\tilde{Q}(t) + S \max_{s \in [S]} \sum_j A_j(t) r_{j,s} - S \min_{s \in [S]} \sum_m U_m h_{m,s} + \bar{r}, 0)
\]
we can write the Loynes’ construction for this process which has the same distribution as this process (and for simplicity we use the same notation, as we are interested in the distribution).

\[
\tilde{Q}^1(0) = \max_{\tau \leq 0} \sum_{\tau \leq t \leq 0} (S \max_{s \in [S]} \sum_j A_j(t)r_{j,s} - S \min_{s \in [S]} \sum_m U_m h_{m,s} + \bar{r}),
\]

(C.10)

assuming that the process started at $-\infty$.

Let us define $X_s(t)$ and $Y_s(t)$ as follows: $X_s(t) := \sum_j A_j(t)r_{j,s}$ and $Y_s(t) := \sum_m U_m h_{m,s}$.

Then,

\[
\tilde{Q}^1(0) = \max_{\tau \leq 0} \sum_{\tau \leq t \leq 0} S(\max_s X_s(t) - \min_s Y_s(t) + \bar{r}).
\]

Now, for any $\theta > 0$

\[
P(\sum_j Q_{j,1} > \bar{r}q)
\leq P(\sum_s Q^*_1 > q)
\leq P(\tilde{Q}_1(0) > q)
\leq P(\theta \tilde{Q}_1(0) > \theta q) = P(\exp(\theta \tilde{Q}_1(0)) > \exp(\theta q))
\leq E[\exp(-\theta q)]E[\exp(\theta \tilde{Q}_1(0))].
\]

183
Now,

\[
\mathbb{E}[\exp(\theta \tilde{Q}_1(0))] = \mathbb{E}\left[\exp(\theta \mathbb{S} \left( \max_{\tau \leq 0} \sum_{\tau \leq t \leq 0} (\max_s X_s(t) - \min_s Y_s(t) + \bar{r}) \right) ) \right] \\
\leq \sum_{\tau \leq 0} \mathbb{E}[\exp(\theta \mathbb{S} \sum_{\tau \leq t \leq 0} (\max_s X_s(t) - \min_s Y_s(t) + \bar{r}))], \quad (C.12)
\]

where the inequality in Eq. C.12 follows because for any random variables \( \{Z_j\} \), \( \exp(\theta Z_j) \) are positive random variables and sum of positives are more than their maximum.

Next, we bound the term within the summation over \( \tau \leq 0 \) in Eq. C.12.

\[
\mathbb{E}[\exp(\theta \mathbb{S} \sum_{\tau \leq t \leq 0} (\max_s X_s(t) - \min_s Y_s(t) + \bar{r}))] \\
\prod_{\tau \leq t \leq 0} \mathbb{E}[\exp(\theta (\max_s X_s(t) - \min_s Y_s(t) + \bar{r}))]. \quad (C.13)
\]

Inequality in Eq. C.13 follows because \( X_s(t), Y_s(t) \) are i.i.d over time.

Next we bound the term within the product \( \prod_{\tau \leq t \leq 0} \) in Eq. C.13,

\[
\mathbb{E} \left[ e^{\theta S(\max_s X_s(t) - \min_s Y_s(t) + \bar{r})} \right] \leq \sum_{s,s'} \mathbb{E} \left[ e^{\theta S(X_s(t)-Y_{s'}(t)+\bar{r})} \right] \quad (C.14)
\]

Inequality in Eq. C.14 is due to the same reason as Eq. C.12.

Let \( A_j(\theta) = \mathbb{E} \left[ e^{\theta A_j(t)} \right] \) and \( U_m(\theta) = \mathbb{E} \left[ e^{\theta U_m(t)} \right] \) for \( j \in [N] \) and \( m \in [M] \).

For \( \theta \in \mathbb{R} \), then,

\[
\mathbb{E} \left[ e^{\theta (X_s(t) - Y_{s'}(t))} \right]
\]
\[ \begin{align*}
&= \mathbb{E} \left[ e^{\theta X_s(t)} \right] \mathbb{E} \left[ e^{-\theta Y_{s'}(t)} \right] \\
&= \mathbb{E} \left[ e^{\theta \sum_j A_j(t) r_{j,s} + \bar{r}} \right] \mathbb{E} \left[ e^{-\theta \sum_i U_i(t) h_{i,s}} \right] \\
&= e^{\theta \bar{r}} \prod_j \mathbb{E} \left[ e^{\theta A_j(t) r_{j,s}} \right] \prod_i \mathbb{E} \left[ e^{-\theta U_i(t) h_{i,s}} \right] \tag{C.15} \\
&= e^{\theta \bar{r}} \prod_j A_j(\theta r_{j,s}) \prod_i U_i(-\theta h_{i,s}) \\
&= \exp \left( \theta \bar{r} + \sum_j \log A_j(\theta r_{j,s}) + \sum_i \log U_i(-\theta h_{i,s}) \right). 
\end{align*} \]

Note that as \( \lambda \in \alpha \mathcal{C} \), by the definition of \( \mathcal{C}_\alpha \), \( \sum_j \lambda_j r_{j,s} < \alpha \sum_m \mu_m h_{m,s} \) and by assumption \( |\sum_m \mu_m h_{m,s} - \sum_m \mu_m h_{m,s'}| \leq \text{subpoly}(N) \) which is used in the following.

First consider the Gaussian-dominated case. Since the process variance is no more than mean and the moment generating function of the variance is upper-bounded by that of a zero-mean Gaussian:

\[ \begin{align*}
\log A_j(\theta r_{j,s}) &\leq \lambda_j \theta r_{j,s} + \lambda_j \frac{(\theta r_{j,1,s})^2}{2} \\
\log U_i(-\theta h_{i,s}) &\leq -\mu_j \theta h_{i,s} + \mu_j \frac{(\theta h_{i,s})^2}{2}.
\end{align*} \]

Note that for any two functions \( k_1 x^2 \) and \( k_2 x \), \( \lim_{x \to 0} k_2 x / k_1 x^2 = \infty \), and hence for any \( \epsilon \in (0,1) \) there exists \( x^* > 0 \) such that for all \( x < x^* \), \( k_1 x^2 / k_2 x < \epsilon \). Hence for any \( \epsilon \in (0,1) \), there exist \( \theta^*_{j,s}, \theta^*_{i,s} > 0 \) for all \( i, j, s \) such that for all \( \theta < \theta^* = \min_{j,s}(\theta^*_{j,s}, \theta^*_{i,s}) \),

\[ \begin{align*}
\log A_j(\theta r_{j,1,s}) &\leq \lambda_j \theta^* r_{j,s} (1 + \epsilon) \tag{C.16} \\
\log U_i(-\theta h_{i,s}) &\leq -\mu_i \theta h_{i,s} (1 - \epsilon) \tag{C.17}
\end{align*} \]
Note that since $N$, $S$, and $M$ are finite and $\theta_{j,s}^*, \theta_{i,s}^* > 0$, for all $i, j, s, \theta^* > 0$. Moreover, note that $\theta^*$ does not depend on $\lambda, \mu$ since the ratio of the linear and quadratic terms in the log moment generating functions are independent of $\lambda$ and $\mu$.

As $e^\theta - 1 = \sum_{k=1}^{\infty} \frac{\theta^k}{k!}$, for the Poisson-dominated case we have

$$
\log A_j(\theta r_{j,s}) \leq \lambda_j \sum_k \frac{(\theta r_{j,s})^k}{k!}
$$

$$
\log U_i(-\theta h_{i,s}) \leq \mu_j \sum_k \frac{(-\theta h_{i,s})^k}{k!}
$$

Again, by the same argument, we can have a $\theta^*$ for which (C.16) and (C.17) are satisfied. As $|\sum_i \mu_i h_{i,s} - \sum_i \mu_i h_{i,s'}| = o(N^\delta), \forall \delta > 0$, and $\sum_i \mu_i h_{i,s} = \Omega(N^c), c > 0$, for all $\theta < \theta^*$ we have:

$$
\mathbb{E} \left[ e^{\theta S(X_s(t) - Y_{s'}(t))} \right] 
\leq \exp \left( \theta^* \bar{r} + \sum_j \lambda_j \theta^* r_{j,s}(1 + \epsilon) - \sum_i \mu_i \theta h_{i,s}(1 - \epsilon) + \theta^* o(\mu_i \theta h_{i,s}) \right)
\leq \exp \left( \theta \left( \bar{r} - \sum_i \mu_i h_{i,s}(\alpha - 2\epsilon) \right) \right). \tag{C.18}
$$

Note (C.18) follows from the fact $\lambda \in (1 - \alpha)\mathcal{E}_{\text{out}}$. As $\epsilon > 0$ can be chosen arbitrarily small, we can have $\alpha - 2\epsilon > 0$. Since $\sum_i \mu_i h_{i,s} > \sum_j (1 - \alpha)\lambda_j r_{j,s}$ and $\sum_j \lambda_j r_{j,1,s}$ scales with $\lambda(N)$, for sufficiently large $\lambda_\alpha$ with $\lambda_j \geq \lambda_\alpha$ for all $j$, we have $\bar{r} - \sum_i \mu_i h_{i,s}(\alpha - \epsilon) \leq -\gamma \sum_i \mu_i h_{i,s}(\alpha - \epsilon)$, for some $\gamma > 0$. Thus, we have for some $\theta > 0$,

$$
\mathbb{E} \left[ e^{\theta S(X_s(t) - Y_{s'}(t))} \right] \leq \exp \left( -\theta SK(N) \right), \tag{C.19}
$$

186
where \( K(N) \) scales with \( N \) no slower than \( \sum_{s,r_j,1,s>0} \lambda_j(N) = \Omega(N^c), \quad c > 0. \)

Thus,

\[
E \left[ e^{\theta S(\max_s X_s(t) - \min_s Y_s(t))} \right] \leq S^2 \exp(-\theta SK(N))
\]

Hence, from Eq. C.12, C.13 and C.14 we have that

\[
E[\exp(\theta \sum_s \tilde{Q}^s(0))] = E[\exp(\theta S\tilde{Q}^1(0))] = E[\exp(\theta \tilde{Q}(0))]
\leq \sum_{\tau \leq 0} S^{2|\tau|} \exp(-\theta SK(N)|\tau|)
\leq c',
\]

because \( S^2 < \exp(\theta^* SK(N)) \) for all sufficiently large \( N \).

Note that though we proved \( E[\exp(\theta Q(t))] < c' \) for \( t = 0 \), but this holds for any finite \( t \) (exactly the same proof).

Hence,

\[
P(Q > c \log N) \leq P(\tilde{Q} > c \log N) \leq c' e^{-c' \theta^* \log N},
\]

which gives the result for an appropriate choice of \( c \).

C.1.8 Proof of Theorem 36

To prove that any \( \lambda \in \mathcal{C}^L \) is stabilizable it is sufficient to invoke Thm. 37 whose proof is below. To show that \( \lambda^O \notin \mathcal{C}^I \) is not stabilizable, we take an
approach similar to the proof of Thm. 27.

Note that the set \( \{ \lambda : \lambda = \sum l \lambda^l, \lambda^l \in \mathcal{C}_l^l \} \) is convex. Also, if \( \lambda' \leq \lambda \) (component-wise) for some \( \lambda \) belonging to the set, then \( \lambda' \in \mathcal{C}^l \). That is, \( \mathcal{C}^l \) is coordinate convex. Hence for any \( \lambda \not\in \text{closure of } \mathcal{C}^L \), there exists a hyperplane \( h \geq 0 \) that strictly separates it from \( \mathcal{C}^L \), i.e., for any \( \{ \lambda^l \in \mathcal{C}^l \} \), for some \( \epsilon > 0 \)

\[
h^T \lambda^0 \geq h^T \sum l \lambda^l + \epsilon
\]

Note that at any epoch \( t \), number of \( j \) jobs allocated \( \Delta_j(t) = \sum l \Delta_j^l(t) \), where \( \Delta_j^l(t) \) is the number of \( j \) jobs allocated to category \( l \) agents. Following similar steps as in the proof of Thm. 27, the result follows.

C.1.9 Proof of Theorem 37

Let \( Q_{l,j,s}(t) \) be the number of unallocated \((j,s)\) tasks in the \( l \)th pool and \( A_j(t) \) be the number of arrived jobs of type \( j \). Let \( A_{l,j}(t) \) be the number of jobs that are sent to pool \( l \) and \( \sum l A_{l,j} = A_j(t) \). At any pool \( l \) MaxWeight is followed and \( \hat{D}_{l,j,s}(t) \) is the number of allocated \((j,s)\) tasks in pool \( l \).

Consider a Lyapunov function \( L(Q) = \sum l,j,s Q_{l,j,s}^2 \). Then:

\[
E [L(Q(t+1)) - L(Q(t)) | Q(t)] \\
\leq E \left[ \sum l,j,s Q_{l,j,s}(t) A_{l,j}(t) | Q(t) \right] - E \left[ \sum l,j,s Q_{l,j,s}(t) \hat{\Delta}_{l,j,s}(t) | Q(t) \right] + E \left[ \sum l,j,s \left( A_{l,j}(t) + \hat{\Delta}_{l,j,s} \right)^2 | Q(t) \right]
\]

The last term can be bounded by noting:

\[
\sum l,j,s \left( A_{l,j}(t) + \hat{\Delta}_{l,j,s} \right)^2 \leq \left( \sum l,j,s \left( A_{l,j}(t) + \hat{\Delta}_{l,j,s} \right) \right)^2
\]

188
\[
\left( \sum_{j,s} A_j(t) + \sum_{l,j,s} \hat{\Delta}_{l,j,s} \right)^2
\leq \left( \sum_{j,s} A_j(t) + \frac{1}{\min(r_{j,s} > 0)} \sum_{l,i,s} h_{i,s} U_{i,s} \right)^2 \quad (C.21)
\]
\[
\leq B' < \infty. \quad (C.22)
\]

Note (C.21) follows similarly as (C.2), whereas (C.22) follows because the arrival and agent-availability processes have bounded second moments.

To bound the first term:

\[
E \left[ \sum_{l,j,s} Q_{l,j,s}(t) A_{l,j}(t) | Q(t) \right] = E \left[ \sum_{l,j} \left( \sum_{s} Q_{l,j,s}(t) \right) A_{l,j}(t) | Q(t) \right]
\leq E \left[ \sum_{l,j,s} \frac{\lambda_{j,l}}{\lambda_{j}} A_j(t) Q_{l,j,s}(t) | Q(t) \right] \quad (C.23)
\leq \sum_l \left( \sum_{j,s} Q_{l,j,s}(t) \lambda_{j,l} \right) \quad (C.24)
\]

where (C.23) is because of the fact JLTT-MWTA sends all arrivals of type \( j \) to the pool \( l \) with minimum \( \sum_s Q_{l,j,s}(t) \).

On the other hand,

\[
E \left[ \sum_{l,j,s} Q_{l,j,s}(t) \hat{\Delta}_{l,j,s}(t) | Q(t) \right] = \sum_l E \left[ \sum_{j,s} Q_{l,j,s}(t) \hat{\Delta}_{l,j,s}(t) | Q(t) \right], \quad (C.25)
\]
because at epoch \( t \) each pool \( l \) runs MaxWeight based on only \( Q_l(t) \) and \( \{\hat{\Delta}_{l,j,s} : j, s\} \) is independent of \( \{A_{l',j}(t), Q_{l',j,s}(t) : l' \neq l\} \) given \( Q_l(t) \).

For every \( l \), we can compute the difference between the \( l \)th term of (C.24) and that of (C.25), which is similar to the first term of (C.2). If
\( \lambda + \epsilon \in C^L \), then \( \lambda^l + \frac{\epsilon}{L} \in C^l \), hence following the same steps as in the proof of Thm. 30 we have

\[
\left( \sum_{j,s} Q_{l,j,s}(t)\lambda^l_j \right) - \mathbb{E}\left[ \sum_{j,s} Q_{l,j,s}(t) \Delta_{l,j,s}(t) | Q(t) \right] \leq -\frac{\epsilon}{L} \sum_{j,s} Q_{l,j,s}(t).
\]

This in turn implies

\[
\mathbb{E}\left[ L(\mathbb{Q}(t + 1)) - L(\mathbb{Q}(t)) | \mathbb{Q}(t) \right] \leq B' - \frac{\epsilon}{L} \sum_{l,j,s} Q_{l,j,s}(t).
\]

Following similar steps as in proof of Thm. 30, we obtain \( \limsup_{t \to \infty} \sum_{l,j,s} \mathbb{E}[Q_{l,j,s}(t)] < \infty \), which in turn implies that for all \( j, s, l \), \( \limsup_{t \to \infty} \mathbb{E}[Q_{l,j,s}(t)] < \infty \). This proves the theorem since for all \( j \), \( Q_j(t) \leq \sum_{l,s} Q_{l,j,s}(t) \).

### C.1.10 Proof of Theorem 38

It is sufficient to prove that \( C^l \subseteq C^O \). Consider any \( \lambda \in C^l \), then by definition of \( C^l \), \( \lambda = \sum_l \lambda^l \), where for all \( l \), \( \lambda^l \in C^l \). By the characterization of the outer region of single category systems \( C^l \subseteq C^l_{\text{out}} \), \( \lambda^l \in C^l_{\text{out}} \). Thus \( \lambda = \sum_l \lambda^l \) for \( \lambda^l \in C^l_{\text{out}} \), for all \( l \). Thus by definition of \( C^O \), \( \lambda \in C^O \), which completes the proof.

### C.1.11 Proof of Theorem 39

We find a high-probability bound on the number of unallocated jobs of type \( j \) and then bound the maximum number of jobs across type. To bound the number of unallocated jobs, we use stochastic domination based on the nature of the method of splitting job arrivals across different pools.
Consider the dynamics of $Q^*_j(t) = \max_l Q_{l,j}(t)$. Let $A_{j,l}(t)$ be the number of jobs of type $j$ that were directed to pool $l$. In the Improvised JSQ step, jobs are sent one-by-one with minimum backlog and hence, the queue $l^*$ with $Q^*_{j}(t) = Q^*_{j}(t)$ gets the minimum number of jobs. Since minimum is less than average, $A_{j,l^*}(t) \leq \frac{A_{j}(t)}{L}$. On the other hand, just before allocation at epoch $t + 1$, the total number of $j$-jobs in $l^*$ cannot be less than the number of jobs in any other $l$ by more than 1. This is because of the Improvised JSQ which allocates jobs one-by-one to the lowest backlogged ($N^l_j$) queue at that time. Hence, we have

$$Q^*_j(t) + \lceil \frac{A_{j}(t)}{L} \rceil + 1 \geq Q_{j,l^*}(t) + A_{j,l^*}(t).$$

Given $Q_{l,j}$, for GreedyJob the number of jobs that can be allocated (assuming number of queued jobs to be infinite) $\Delta_{l,j}$ is monotonic in $U^l$, i.e., if $U^l \geq U'^{l'}$ (component-wise) $\Delta_{l,j} \geq \Delta'_{l,j}$ for all $j$. This property will be useful below.

Consider the following dynamics $Q^j(t), j \in [N]$. Arrivals for each $j$ are according to $\lceil \frac{A_{j}(t)}{L} \rceil + 1$ and agent-availability is according to $U^i(t) = \min_l U^l_i(t)$. This is a single-category system and allocations in this system are according to GreedyJob. As $U^i(t) \leq U^l_i(t)$ for all $i, l$, the number of allocations (assuming queues to be infinite) satisfies $\Delta^j(t) \leq \Delta_{l,j}(t)$. This implies that for each type $j$ $Q_j^j(t)$ dominates $Q^*_j(t)$ as the first queue at any epoch has more number of jobs to be allocated and less number of possible allocations (as $\Delta^j$ is smaller). Thus, it is sufficient to bound $Q^j(t)$.  

191
For this we proceed along the lines of the proof of Thm. 35, replacing arrivals by $A^j(t) = \lceil \frac{A^j(t)}{L} \rceil + 1$ and agent-availability by $U^i(t) = \min_l U^i_l(t)$.

Hence, for each skill $s$, the queue of unallocated skill-hours $H^s(t)$ is the arrival $X^s(t) = \sum_j A^j(t) r_{j,s}$ and the possible amount that can be drained is $Y^s(t) = \sum_i U^i(t) h_{i,s}$. Hence,

$$Q^s(t + 1) = |Q^s(t) + X^s(t) - Y^s(t) + \bar{r}|^+.$$

We can follow similar steps by noting that for $\theta > 0$:

$$E\left[e^{-\theta X^s}\right] = E\left[e^{-\theta \sum_i \min_l U^i_l(t) h_{i,s}}\right] = E\left[e^{-\theta \min_l \sum_i U^i_l(t) h_{i,s}}\right] = E\left[\max_l e^{-\theta \sum_i U^i_l(t) h_{i,s}}\right] \leq \sum_l E\left[e^{-\theta \sum_i U^i_l(t) h_{i,s}}\right]. \quad (C.26)$$

On the other hand,

$$E\left[e^{\theta X^s}\right] \leq e^{\theta(\frac{1}{L} + 1)\bar{r}} \prod_j E\left[e^{\frac{\theta r_j}{L} A^j(t)}\right].$$

Making the assumption on agent arrival rates,

$$\sum_l E\left[e^{-\theta \sum_i U^i_l(t) h_{i,s}}\right] \leq e^{\log L + O(\text{subpoly}(N))} E\left[e^{-\theta \sum_i U^i_l(t) h_{i,s}}\right] \quad (C.27)$$

Since $\max_i \mu^i - \mu^i = \text{subpoly}(N)$ for all $i$, $\lambda \in (1 - \alpha)C^0$ implies that $\frac{\lambda}{L} - \text{subpoly}(N) \in C^{out}$. Thus, we can use the same steps as we did for single-category systems.
For Gaussian-dominated as well as Poisson-dominated cases, following similar steps we can obtain that for a $\theta^* > 0$ (independent of $\lambda$) and sufficiently large $\lambda$,

$$E[e^{\theta(X_s - Y_s^*)}] \leq e^{-\theta^* K(\lambda) + \log L + O(\text{subpoly}(N))}$$

Note that $K(\lambda)$ increases as $\Omega(\min_i \lambda_i) = \Omega(N^c)$ for some $c > 0$, $L = O(1)$, hence there exists $\lambda$ sufficiently large such that $-\theta^* K(\lambda) + \log L + O(\text{subpoly}(N))$ is strictly negative and $E[e^{\theta(X_s - Y_s^*)}] < 1$. The rest follows similarly as the proof of Thm. 33.

C.1.12 Proof of Theorem 40

We first consider FD, FND, and IND systems.

Let $\tilde{A}_j(t)$ be the number of accepted jobs of type $j$ between starts of epochs $t - 1$ and $t$. Let $\tilde{\Delta}_{j,s}(t)$ be the number of allocated $(j, s)$ tasks by the MaxWeight part of MWTA before the execution of Task Allocation. Let $\hat{D}_{j,s}(t)$ be the number of allocated $(j, s)$ tasks at allocation epoch $t$ at the end of Task Allocation. Then:

$$\sum_{j,s,r_{j,s}>0} \left( \tilde{Q}_{j,s}^2(t+1) - \tilde{Q}_{j,s}^2(t) \right) = \sum_{j,s,r_{j,s}>0} \left( (\tilde{Q}_{j,s}(t) + \tilde{A}_j(t) - \hat{D}_{j,s}(t))^2 - \tilde{Q}_{j,s}^2(t) \right)$$

$$\leq \sum_{j,s,r_{j,s}>0} \left( (\tilde{Q}_{j,s}(t) + \tilde{A}_j(t) - \hat{\Delta}_{j,s}(t))^2 - \tilde{Q}_{j,s}^2(t) \right)$$

$$= \sum_{j,s,r_{j,s}>0} Q_{j,s}(t) \left( \tilde{A}_j(t) - \hat{\Delta}_{j,s}(t) \right) + \sum_{j,s,r_{j,s}>0} \left( \tilde{A}_j^2(t) + \Delta_{j,s}^2(t) \right)$$

Expectation of the second summation (conditioned on $Q(t)$) can be bounded by noting that $\tilde{A}_j^2(t) \leq A_j^2(t)$ and the fact that the arrival pro-
cesses have bounded second moment. The value $E\left[\sum_{j,s;r_{j,s}>0} \Delta_{j,s}^2(t)\right]$ can be bounded similarly as in the proof of Thm. 30. Hence, we consider the expectation of the second term to be bounded by $B$ independent of $Q$.

$$
E \left[ \sum_{j,s;r_{j,s}>0} \left( \tilde{Q}_{j,s}(t+1) - \tilde{Q}_{j,s}^2(t) \right) \middle| Q(t) \right]
$$

$$
\leq B + E \left[ \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( \tilde{A}_j(t) - \tilde{\Delta}_{j,s}(t) \right) \middle| Q(t) \right]
$$

$$
\leq B + E \left[ \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta(t))A_j(t) - \tilde{\Delta}_{j,s}(t) \right) \middle| Q(t), A(t) \right] Q(t)
$$

$$
\leq B + E \left[ \frac{\beta(t) \sum_j A_j(t)}{\nu} + \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta(t))A_j(t) - \tilde{\Delta}_{j,s}(t) \right) \right]
$$

$$
- \frac{\beta(t) \sum_j A_j(t)}{\nu} \left| Q(t), A(t) \right| Q(t)
$$

$$
\leq B + E \left[ \frac{\beta(t) \sum_j A_j(t)}{\nu} + \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta(t))A_j(t) - \tilde{\Delta}_{j,s}(t) \right) \right]
$$

$$
- \frac{\beta(t) \sum_j A_j(t)}{\nu} \left| Q(t), A(t) \right| Q(t)
$$

$$
\leq B + E \left[ \frac{\beta(t) \sum_j A_j(t)}{\nu} + \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta(t))A_j(t) - \tilde{\Delta}_{j,s}(t) \right) \right]
$$

$$
- \frac{\beta(t) \sum_j A_j(t)}{\nu} \left| Q(t), A(t) \right| Q(t)
$$

$$
\leq B + E \left[ \frac{\beta(t) \sum_j A_j(t)}{\nu} + \sum_{j,s;r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta^*)A_j(t) - \tilde{\Delta}_{j,s}(t) \right) \right]
$$

(C.28)

194
\[ -\frac{\beta(t)}{\nu} \sum_{j} A_j(t) [Q(t), A(t)] |Q(t)| \]  
\[ \leq B + \frac{1}{\nu} E[(\beta^* - \beta(t)) \sum_j A_j(t)] + E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta^*) A_j(t) - \hat{\Delta}_{j,s}(t) \right) |Q(t)| \right] \]
\[ \leq B + \frac{1}{\nu} E[(\beta^* - \beta(t)) \sum_j A_j(t)] + E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta^*) \lambda_j(t) - \hat{\Delta}_{j,s}(t) \right) |Q(t)| \right] \]

where (C.29) follows as \( \beta(t) \) minimizes \( \beta \sum_j A_j(t) - \nu \beta \sum_{j,s: r_{j,s}>0} \hat{Q}_{j,s}(t) A_j(t) \).

As \( (1 - \beta^*) \lambda + \epsilon 1 \in \mathcal{C} \), following the same steps as in the proof of the Thm. 30:

\[ E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \left( (1 - \beta^*) \lambda_j(t) - \hat{\Delta}_{j,s}(t) \right) |Q(t)| \right] \leq -\epsilon \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \]

As \( \beta^* - \beta(t) \leq 1 \), we have

\[ E \left[ \sum_{j,s: r_{j,s}>0} \left( \hat{Q}_{j,s}^2(t + 1) - \hat{Q}_{j,s}^2(t) \right) \right] \leq B + \frac{1}{\nu} \sum_j \lambda_j - \epsilon E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \right]. \]

Following again the same steps we show that \( \limsup_{t \to \infty} E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \right] < \infty \) which implies the crowd system is stable. Hence, we can assume that

\[ E \left[ \sum_{j,s: r_{j,s}>0} Q_{j,s}(t) \right] < C \text{ for some } C < \infty. \]

Thus we can write

\[ E \left[ \sum_{j,s: r_{j,s}>0} \left( \hat{Q}_{j,s}^2(T) - \hat{Q}_{j,s}^2(0) \right) \right] \leq BT + \frac{1}{\nu} \sum_{t=1}^{T} E[(\beta^* - \beta(t)) \sum_j A_j(t)], \]

which in turn implies

\[ (1 - \beta^*) \sum_j \lambda_j - \frac{1}{T} E \left[ \sum_{t=1}^{T} (1 - \beta(t)) \sum_j A_j(t) \right] \leq \nu B + \frac{\nu}{T} \sum_{j,s: r_{j,s}>0} \hat{Q}_{j,s}^2(0) \]

195
Since $B$ is a constant depending on arrival and availability statistics, $\nu$ and $T$ can be chosen to be small and large respectively to ensure that the left side is arbitrarily small. The desired result follows by noting that

$$
\mathbb{E} \left[ \sum_{t=1}^{T} (1 - \beta(t)) \sum_j A_j(t) \right] = \mathbb{E} \left[ \sum_{t=1}^{T} \sum_j \tilde{A}_j(t) \right].
$$

Also note that since the only requirement is the independence of $A_j(t)$ and $U(t)$ across time, the proof directly extends to settings with non-stationary arrival and availability processes.

Now consider the ID setting and use the Lyapunov function $\sum_{l,j,s} Q_{l,j,s}^2$.

Similar to before, the Lyapunov drift can be bounded by

$$
B + \mathbb{E} \left[ \sum_{l,j,s:r_{j,s}>0} Q_{l,j,s}(t) \left( \tilde{A}_{l,j}(t) - \hat{\Delta}_{l,j,s}(t) \right) |Q(t) \right].
$$

Note that $\sum_{l,j,s:r_{j,s}>0} Q_{l,j,s}(t) \tilde{A}_{l,j}(t)$ is equal to

$$
\sum_{l,j,s:r_{j,s}>0} \min_l \left( \sum_s Q_{l,j,s}(t) \right) \tilde{A}_j(t),
$$

as $\tilde{A}_j(t) = \sum_l \tilde{A}_{l,j}(t)$ and JLT ensures that the jobs are sent to the category with $\min_l (\sum_s Q_{l,j,s}(t))$. So we have

$$
\mathbb{E} \left[ \sum_{l,j,s:r_{j,s}>0} Q_{l,j,s}(t) \left( \tilde{A}_{l,j}(t) - \hat{\Delta}_{l,j,s}(t) \right) |Q(t) \right] \\
= \mathbb{E} \left[ \sum_{l,j:s:r_{j,s}>0} \min_l \left( \sum_s Q_{l,j,s}(t) \right) \tilde{A}_j(t) - \sum_{l,j,s:r_{j,s}>0} Q_{l,j,s}(t) \hat{\Delta}_{l,j,s}(t) |Q(t) \right] \\
\leq \mathbb{E} \left[ \sum_{l,j:s:r_{j,s}>0} Q_{l,j,s}(t) \frac{\lambda_j}{\lambda_j} \tilde{A}_j(t) - \sum_{j,s:r_{j,s}>0} Q_{l,j,s}(t) \hat{\Delta}_{l,j,s}(t) |Q(t) \right],
$$

196
where \((1 - \beta^*)\lambda = \sum_l (1 - \beta^*) \lambda^l\) for \(\lambda^l + \epsilon 1 \in C^l\). The remainder of the proof is similar to the approach for the FD/FND/IND settings, i.e., the MWTA proof for \((1 - \beta^*)\lambda^l + \epsilon 1 \in C^l\) for each \(l\).

C.2 Computation

C.2.1 Computation for Centralized Allocation

For a single category system \((L = 1)\), note that \(z_{j,s}^1 = a_{j,s} \in \mathbb{Z}_+\) and hence the feasibility condition (4.1) becomes

\[
\sum_j a_{j,s} r_{j,s} \leq \sum_i u_i h_{i,s} \text{ for all } s \in [S], a_{j,s} \in \mathbb{Z}_+,
\]

with condition (4.2) additionally requiring \(a_{j,s} = a_{j,s'}\) for all \(j, s, s'\). Thus, \(C(u)\) is the set of \(\{a_{j,s}\}\) satisfying the above conditions for respective classes of jobs (as well as systems) and \(C\) is the weighted (by \(\Gamma(u)\)) sum of convex hulls of \(C(u)s\).

\(C^\text{out}_\mu\) has a simple characterization as well. As for any \(j \in [N], (j, 1) \in E,\) and \(N(J) = 1\) for all \(J \subset [N], \sum_{l \in N(J)} \sum_{i \in [M]} \mu_{i,s}^l h_{i,s} = \sum_{i \in [M]} \mu_i h_{i,s}\). Thus it is sufficient to satisfy the inequality for \(J = [N]\), and hence,

\[
C^\text{out}_\mu = \left\{ \lambda : \sum_{j \in [N]} \lambda_j r_j \leq \sum_{i \in [M]} \mu_i h_i \right\}.
\]

The MaxWeight computation in MWTA for single-category decomposable systems turns out to be the following integer linear program (ILP), which is related to knapsack problems.

\[
\arg \max_{\Delta_{j,s} j,s} \sum_{j,s} Q_{j,s} \Delta_{j,s}
\]
\[
\sum_j a_{j,s} r_{j,s} \leq \sum_i u_i h_{i,s} \forall s \in [S],
\] (C.30)

This problem is an integer program, hence it is not clear whether this problem can be solved efficiently at all instants. In fact, it is a so-called unbounded knapsack problem for a given \(u\) and \(Q\). This problem is known to be NP-hard \([87]\). There is a pseudo-polynomial algorithm based on dynamic programming which solves it exactly, but the runtime may depend on \(Q_{j,s}\).

This dynamic programming-based algorithm can be converted into a fully polynomial time approximation schemes (FPTAS) which achieves any \((1 - \epsilon)\) approximation of the problem in \(\text{poly}(\frac{1}{\epsilon})\) computations. Moreover, there exists faster greedy algorithms that achieve \(\frac{1}{2}\) approximation and can be converted into a polynomial time approximation scheme (PTAS) that achieves \((1 - \epsilon)\) approximation in \(\text{poly}(n^{\frac{1}{\epsilon}})\) computations. Thus we can conclude that though the MWTA algorithm is computationally hard for single-category decomposable system, there exist efficient approximation schemes. It is not hard to show (Prop. 57 below) that an algorithm that gives \((1 - \epsilon)\) approximation of the optimization problem in MWTA can stabilize any \(\lambda\) for which \(\frac{\lambda}{(1-\epsilon)} \in \mathcal{C}_\Gamma\).

For single-category non-decomposable systems, feasibility condition (4.2) of an allocation of \(a_j\) jobs of type \(j\) is given by,

\[
\sum_j a_{j,s} r_{j,s} \leq \sum_i u_i h_{i,s} \forall s \in [S], a_j \in \mathbb{Z}_+.
\]

Hence the stabilizable region changes accordingly to

\[
\mathcal{C}_\Gamma = \left\{ \sum_u \Gamma(u) \lambda(u) : \lambda(u) \in \mathcal{C}^{ND}(u) \right\},
\]

198
\[ C^{ND}(u) = \text{conv}\left\{ a_{j,s} \in \mathbb{Z}_+: \sum_{j} a_{j,s} r_{j,s} \leq \sum_{i} u_i h_{i,s} \text{ and } a_{j,s} = a_j \forall s \right\}. \]

where \( \text{conv}\{\cdot\} \) is the convex hull.

Hence, in this case, the MWTA allocation needs to solve

\[
\arg\max_{\Delta_{j,s}} \sum_j \left( \sum_s Q_{j,s} \Delta_{j,s} \right)
\]

s.t. \( \sum_j \Delta_{j,s} r_{j,s} \leq \sum_i u_i h_{i,s}, \Delta_{j,s} = \Delta_{j,s}', \text{ for all } s, s' \in [S], \quad (C.31)\]

and then divide the allocated jobs arbitrarily among agents while meeting their per skill time-availability, as there is only one category of agents.

This problem is also a knapsack-like integer program, with an additional constraint that the number of \((j, s)\)-items has to be the same as the number of \((j, s')\)-items for all \(j, s\) and \(s'\). Such a problem is called a multi-dimensional knapsack problem. This problem is also NP-hard. Moreover, provably there cannot exist a fully polynomial time approximation scheme for this problem [87].

For a multi-dimensional knapsack problem, there exists an approximation scheme that achieves an approximation factor equal to the dimension \(d\) [87]. This approximation scheme can be converted into a PTAS with complexity \(O(N^{\kappa})\) and an approximation factor of \(1 - \epsilon\), where \(\kappa\) is strictly increasing with dimension and \(\frac{1}{\epsilon}\). In the case of our setting, the number of skills \(S\) (dimension) is large and may scale with \(N\), hence this scheme is not suitable.

Though the total number of skills \(S\) can scale with \(N\), in most cases the number of skill-parts that a type \(j\) job has is a constant, i.e., \(r_j\) has most
coordinates as 0. Thus it is apparent from the objective function that the optimal choice of $\Delta_{j,s}$ is 0 for the corresponding coordinates $s$. This allows us to rewrite the optimization problem as another multi-dimensional knapsack problem with constant dimensions given by $\max_j |\{s : r_{j,s} > 0\}|$.

For this problem we can use the PTAS to obtain arbitrarily close approximation and hence can stabilize a rate-region arbitrarily close to $\mathcal{C}^{ND}$. But the complexity of this algorithm is very high as complexity scales super-exponentially ($N^k$) with the approximation factor (unlike decomposable systems where we have an FPTAS, polynomial in $\frac{1}{\epsilon}$).

Note that our goal is not to solve (C.31) optimally, but to have a fast allocation scheme that can stabilize a large fraction of $\mathcal{C}^{ND}$. In this regard we can take a different approach that exploits basic characteristics of a crowd system. Since $N$ and $\lambda_j(N)$ for $j \in [N]$ are large in most crowd systems, if an allocation scheme stabilizes any rate $\lambda$ for $\lambda + c \mathbf{1} \in \mathcal{C}$, then it stabilizes $\left(1 - \max_i \frac{1}{\lambda_i(N)}\right) \mathcal{C}$. Note that as $\lambda_i(N)$ scales with $N$, this implies that such a scheme would stabilize almost all of $\mathcal{C}$. Motivated by this, we propose the following allocation scheme, which is a modification of (C.31).

\[
\{\hat{x}_j, j\} = \arg \max_{x_j \in \mathbb{R}} \sum_j \left(\sum_s Q_{j,s}\right) x_j \\
\text{s.t. } \sum_j x_j r_{j,s} \leq \sum_i u_i h_{i,s}, \text{ for all } s,
\]

and allocate $\tilde{\Delta}_j = \lfloor \hat{x}_j\rfloor$ jobs of type $j$ to the agents, splitting arbitrarily while meeting time-availability constraints.
Note that since in (C.32), the variables are relaxed to \( \mathbb{R} \) from \( \mathbb{Z} \),
\[
\sum_j (\sum_s Q_{j,s}) \hat{x}_j \geq \sum_j (\sum_s Q_{j,s}) \hat{z}_j.
\]
Again, \( \hat{\Delta}_j = \lfloor \hat{x}_j \rfloor \geq \hat{x}_j - 1 \), hence
\[
\sum_j (\sum_s Q_{j,s}) \hat{\Delta}_j \geq \sum_j (\sum_s Q_{j,s})(\hat{z}_j - 1).
\]

The following proposition guarantees that a proposed LP-relaxation scheme stabilizes any \( \lambda \) with \( \lambda + 1 \in \mathbb{C}^{ND} \).

**Proposition 57.** Let \( \mathcal{P} \) be an allocation scheme that at epoch \( t \) does an allocation \( \{\Delta(t)\} \) instead of \( \{\hat{\Delta}(t)\} \) of the MWTA allocation scheme, which satisfies
\[
\sum_{j,s} Q_{j,s}(t) \hat{\Delta}_{j,s}(t) \leq \sum_{j,s} Q_{j,s}(t) \Delta_{j,s}(t) + \sum_{j,s} Q_{j,s}(t) \delta,
\]
or,
\[
(1 - \epsilon) \sum_{j,s} Q_{j,s}(t) \hat{\Delta}_{j,s}(t) \leq \sum_{j,s} Q_{j,s}(t) \Delta_{j,s}(t),
\]
stabilizes any rate \( \lambda \in \mathbb{C} \) if \( \lambda + \delta \mathbf{1} \in \mathbb{C} \) or \( \frac{1}{1-\epsilon} \lambda \in \mathbb{C} \) respectively.

**Proof.** This proof follows the same steps as the proof of Thm. 30. We first prove the result for an allocation with
\[
\sum_{j,s} Q_{j,s}(t) \hat{\Delta}_{j,s}(t) \leq \sum_{j,s} Q_{j,s}(t) \Delta_{j,s}(t) + \sum_{j,s} Q_{j,s}(t) \delta.
\]
as follows:
\[
\mathbb{E} [L(Q(t+1)) - L(Q(t)) | Q(t)]
\]
\[
\leq \mathbb{E} \left[ \sum_{j,s} Q_{j,s}(t) A_j(t) | Q(t) \right] - \mathbb{E} \left[ \sum_{j,s} Q_{j,s}(t) (\hat{\Delta}_{j,s}(t) - \delta) | Q(t) \right]
\]
\[
+ \mathbb{E} \left[ \sum_{j,s} \left( A_j^2(t) + (\hat{\Delta}_{j,s} - \delta)^2 \right) | Q(t) \right]
\]
201
We can bound the last term as above, because $(\hat{\Delta}_{j,s} - \delta)^2 \leq \hat{\Delta}_{j,s}^2 + \delta^2$. Thus,

$$E[L(Q(t+1)) - L(Q(t)) | Q(t)] \leq B + E\left[\sum_{j,s} Q_{j,s}(t)(A_j(t) + \delta) | Q(t)\right] - E\left[\sum_{j,s} Q_{j,s}(t)\hat{\Delta}_{j,s}(t) | Q(t)\right]$$

Now note that if $\lambda$ is such that $\lambda + \delta 1 \in \mathcal{C}$, then we can write it in terms of convex combinations of $d \in \mathcal{C}(u)$ and follow the same steps as in the proof of Thm. 30.

Similarly for the other case of constant factor approximation we have:

$$E[L(Q(t+1)) - L(Q(t)) | Q(t)] \leq B + E\left[\sum_{j,s} Q_{j,s}(t)A_j(t) | Q(t)\right] - E\left[\sum_{j,s} Q_{j,s}(t)\hat{\Delta}_{j,s}(t)(1 - \epsilon) | Q(t)\right]$$

(C.33)

as $(1 - \epsilon)^2 \hat{\Delta}_{j,s}^2 \leq \hat{\Delta}_{j,s}^2$.

If $\lambda \in (1 - \epsilon)\mathcal{C}$, then by definition of $\mathcal{C}$ and $(1 - \epsilon)\mathcal{C}$, there exist $\nu(u) \in \mathcal{C}(u)$ and $\{d^k(u) \in \mathcal{C}(u)\}$ such that, $\lambda \leq (1 - \epsilon)\nu(u)\Gamma(u)$ and $\sum_k \gamma_k d^k = \nu(u)$ for $\gamma_k > 0, \sum_k \gamma_k = 1$. As for any $d^k$ for a given $u$, $\sum_{j,s} Q_{j,s}(t)\hat{\Delta}_{j,s}(t) \leq \sum_{j,s} Q_{j,s}(t)d^k_{j,s}, \sum_{j,s} Q_{j,s}(t)\lambda_j \leq E\left[\sum_{j,s} Q_{j,s}(t)\hat{\Delta}_{j,s}(t) | Q(t)\right]$. Then following the proof of Thm. 30, the result follows. □
Appendix D

D.1 Proofs

In this section we present the proofs of the main results.

D.1.1 Proof of Theorem 42

Here we only prove that for any $\lambda$ outside the closure of $\mathcal{C}$ cannot be stabilized by any policy. To prove the other way it is sufficient to show that there exists a policy that stabilizes any $\lambda$ in the interior of $\mathcal{C}$. Hence, it is sufficient to prove Theorem 44 in this case, which we do later.

This proof consists of the following steps. We first compare two systems, the original system in question and another in which there is no precedence constraint among different steps of a job. We claim that on any sample path under any policy for the first system, there exists a policy in the second system so that the total number of incomplete jobs across all job types in the second is a lower bound (sample path-wise) for that in the first. Then we show that the second system cannot be stabilized for a $\lambda$ outside the closure of $\mathcal{C}$ and hence, so is true for the first system.

Note that the claim that in terms of number of incomplete jobs across all types the second system is a lower bound on the first system follows by
considering the same policy for the second system as in the first system (this
gives a matching lower bound).

The proof of the result that the second system cannot be stabilized by
any policy is similar to the proof in Sec. C.1.1.

D.1.2 Proof of Theorem 43

This proof is very similar to the proof of Theorem 28 in Sec. C.1.2.

D.1.3 Proof of Theorem 44

The process \( \{Q_{j,k}(t)\} \) is a discrete-time Markov chain on \( \mathbb{Z}_+^{\sum_j K_j} \) under
the centralized scheme. This is because arrival and availability processes are
i.i.d and the centralized allocation at \( t \) does not depend on process values
before \( t \). We show that for this chain all closed classes are positive recurrent
and with probability 1 the chain enters one of the closed classes. Note that this
implies that starting with any initial distribution the Markov chain reaches a
stationary distribution (which may depend on the initial condition) in the sense
that \( \exists d \in \{1, 2, \cdots\} \) (as there may be a closed class which is not aperiodic),
and a distribution \( \pi \) on \( \mathbb{Z}_+^{\sum_j K_j} \) s.t. \( Q(td) \to \pi \) in distribution.

To show stability we need to show \( \limsup_{t \to \infty} E[Q_{j,k}(t)] < \infty, \forall (j, k) \).
Towards this note that it is sufficient to show that \( E[\pi[\sum_{j,k} Q_{j,k}]] \) is finite.
Because this implies that \( \lim_{t \to \infty} E[\sum_{j,k} Q_{j,k}(td)] \) is finite. Note that for any
1 < \( \tau < d \). \( \sum_{j,k} Q_{j,k}(td + \tau) \leq \sum_{j,k} [Q_{j,k}(t) + \sum_{t'=1}^{d} A_j(t')] \) and as arrivals
have finite expectation so \( \limsup_{t \to \infty} E[Q_{j,k}(t)] < \infty, \forall (j, k) \).
Hence, it is sufficient to prove that starting with any initial distribution \( \exists d \in \{1, 2, \cdots \} \) s.t. \( Q(td) \to \pi \) in distribution and \( E_\pi[\sum_{j,k} Q_{j,k}] \) is finite. To prove the convergence in distribution we use a variation of the Foster-Lyapunov theorem presented in [76].

The following lemmas are useful.

**Lemma 58.** For any \( x, y, z \geq 0 \), \((|x - y|^+ + z)^2 \leq x^2 + y^2 + z^2 + 2x(z - y)\).

*Proof.*

\[
\begin{align*}
(|x - y|^+ + z)^2 &= (|x - y|^+)^2 + z^2 + 2z|x - y|^+ \\
&\leq (|x - y|^+)^2 + z^2 + 2xz \\
&\leq (x - y)^2 + z^2 + 2xz \\
&= x^2 + y^2 + z^2 + 2x(z - y)
\end{align*}
\]

(D.1)

where the last inequality follows because \((\max(0,a))^2 \leq a^2\). \(\Box\)

**Lemma 59.** For any allocation \( \{S_{j,k}\} \),

\[
\begin{align*}
\sum_j (l_{j,1}Q_{j,1}(A_{j,1} - S_{j,1}) &
+ \sum_{k>1} l_{j,k} \left( Q_{j,k}(t)S_{j,p_j(k)}^*(t) - Q_{j,k}(t)S_{j,k}(t) \right) \\
= \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t))(A_{j,1} - S_{j,k})
\end{align*}
\]

(D.2)
Proof. First we claim that for any \( j \),

\[
\sum_{k>1} l_{j,k} \left( Q_{j,k}(t)S_{j,p_j(k)}^*(t) - Q_{j,k}(t)S_{j,k}(t) \right)
- \sum_{k=2}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t))S_{j,k}
\]

(D.3)

This can be seen by comparing coefficients of \( Q_{j,k}, k > 1 \) on both sides of the expression. Note that \( \sum_{r \in c_j(k)} l_{j,r} = l_{j,k} \). Also, note that in the sum in the RHS \( Q_{j,k} \) appears twice, once in \( -\sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t))S_{j,k} \) where the coefficient is \( -S_{j,k} \) and again in \( -\sum_{r \in c_j(p_j(k))} l_{j,r}(Q_{j,p_j(k)}(t) - Q_{j,r}(t))S_{j,p_j(k)} \) where the coefficient is \( S_{j,p_j(k)} \).

This implies that for any \( j \),

\[
- l_{j,1}Q_{j,1}S_{j,1}
+ \sum_{k>1} l_{j,k} \left( Q_{j,k}(t)S_{j,p_j(k)}^*(t) - Q_{j,k}(t)S_{j,k}(t) \right)
= - \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t))S_{j,k}
\]

(D.4)

Note that as \( l_{j,k} = \sum_{r \in c_j(k)} l_{j,r} \),

\[
l_{j,1}Q_{j,1} = \sum_{r \in c_j(1)} l_{j,r}(Q_{j,1} - Q_{j,r}) + \sum_{r \in c_j(1)} l_{j,r}Q_{j,r}.
\]

Then again applying the same restructuring of the terms for the sub-trees rooted at \( r \), we eventually obtain
The result follows by combining this (multiplied by $A_{j,1}$) with Eq. D.4.

We consider the following Lyapunov function:

$$L(Q) = \sum_j \sum_k l_{j,k} Q_{j,k}^2,$$

where $l_{j,k}$ is the number of leaves in the subtree of $T_j$ rooted at $k$.

$$\mathbb{E}[L(Q(t + 1))|Q(t)] = \mathbb{E} \left[ \sum_{j,k} (Q_{j,k}(t) - D_{j,k}(t) + A_{j,k}(t))^2 | Q(t) \right].$$

Let $p_j(k)$ denotes the parent of $k$ in $T_j$ and $c_j(k)$ denotes the set of children of $k$ in $T_j$. Note that for all $t$, $k = 1$ $A_{j,1}(t) = A_j(t)$ and for $k > 1$ $A_{j,k}(t) = D_{j,p_j(k)}(t)$. Also note that $S^*_{j,k} \geq D_{j,k}$ for all $t$. So,

$$\mathbb{E} \left[ \sum_{j,k} l_{j,k} (Q_{j,k}(t) - D_{j,k}(t) + A_{j,k}(t))^2 | Q(t) \right]$$

$$= \mathbb{E} \left[ \sum_{j,k} l_{j,k} (|Q_{j,k}(t) - S^*_{j,k}(t)|^+ + A_{j,k}(t))^2 | Q(t) \right]$$

$$\leq \mathbb{E} \left[ \sum_j (l_{j,1}(|Q_{j,1}(t) - S^*_{j,1}(t)|^+ + A_{j,1}(t))^2 \right]$$
\[+ \sum_{k>1} l_{j,k} \left( |Q_{j,k}(t) - S_{j,k}^*(t)|^2 + S_{j,p_{j}(k)}^*(t) \right)^2 |Q(t)| \].

By Lemma 58,

\[
E[L(Q(t+1))|Q(t)] = E \left[ \sum_{j} l_{j,1} \left( A_{j,1}^2(t) + (S_{j,1}^*(t))^2 + Q_{j,1}^2 + 2Q_{j,1}(A_{j,1} - S_{j,1}^*) \right. \\
+ \sum_{k>1} l_{j,k}((S_{j,k}^*(t))^2 + (S_{j,p_{j}(k)}^*(t))^2 \\
+ Q_{j,k}^2 + 2Q_{j,k}(S_{j,p_{j}(k)}^* - S_{j,k}^*)) \right] |Q(t)| \\
\leq C_1 + 2E \left[ \sum_{j,k} l_{j,1}(S_{j,k}^*(t))^2 |Q(t)| \right. \\
+ 2E \left[ \sum_{j} (l_{j,1}Q_{j,1}(A_{j,1} - S_{j,1}) \\
+ \sum_{k>1} l_{j,k}Q_{j,k}(S_{j,p_{j}(k)}^* - S_{j,k}^*) \right] |Q(t)| (D.5) \\
\leq C_2 + 2E \left[ \sum_{j} (l_{j,1}Q_{j,1}(A_{j,1} - S_{j,1}) \\
+ \sum_{k>1} l_{j,k}Q_{j,k}(S_{j,p_{j}(k)}^* - S_{j,k}^*) \right] |Q(t)| (D.6)
\]

Eq. D.5 follows because arrival processes have bounded second moments and are i.i.d and the fact that \((S_{j,k}^*(t))^2 \geq 0\) (so, over-counting them gives an upper-bound). Eq. D.6 is due to the following (where \(K = \max_j K_j\)):

\[
E[\sum_{j,k} l_{j,k}(S_{j,k}^*(t))^2 |Q(t)|]
\]

208
\[ K \mathbb{E}\left[ \left( \sum_{j,k} S_{j,k}^*(t) \right)^2 | Q(t) \right] \leq K \frac{1}{\max_j \left( \sum_s r_{j,k,s} \right)^2} \mathbb{E}\left[ \left( \sum_{j,k,s} S_{j,k}^*(t) r_{j,k,s} \right)^2 | Q(t) \right] \]

\[ \leq K \frac{1}{\max_j \left( \sum_s r_{j,k,s} \right)^2} \mathbb{E}\left[ \left( \sum_{m,s} U_m(t) h_{m,s} \right)^2 | Q(t) \right] \]  

\[ \leq K \frac{\max_m \left( \sum_s r_{m,s} \right)^2}{\max_j \left( \sum_s r_{j,k,s} \right)^2} \mathbb{E}\left[ \sum_m U_m^2 \right] \]

\[ < \infty \]

Eq. D.7 comes from the task allocation constraint and the last step follows as availability processes have bounded second moment.

Consider the last term of Eq. D.6, as \( C_2 \) plus this is the upper-bound for Lyapunov drift \( \mathbb{E}[L(Q(t+1))|Q(t)] - L(Q(t)) \). Then by Lemma 59 and the fact that \( \{A_j(t)\} \) are i.i.d,

\[
\mathbb{E}\left[ \sum_j \left( l_{j,1} Q_{j,1} (A_{j,1} - S_{j,1}^*) \right) \right. \\
+ \sum_k l_j (Q_{j,k}(t) S_{j,p_j(k)}^*(t) - Q_{j,k}(t) S_{j,k}^*(t)) | Q(t) \left. \right] \\
= \sum_j K_j \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k}(t) - Q_{j,r}(t)) \lambda_j \\
- \mathbb{E}\left[ \sum_j K_j \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k}(t) - Q_{j,r}(t)) S_{j,k}^* | Q(t) \right].
\]
Note that for any $Q(t)$ and $U(t)$

\[
\sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) S_{j,k}^* \\
\geq \max_{a \in \mathcal{C}(U(t))} \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) a_{j,k}
\]

Note that in the optimal allocation $\{S_{j,k}^*\}$, $S_{j,k}^* \geq 0$ only if $\sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \geq 0$ (else, just making them 0 gives a better allocation). So,

\[
\sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) S_{j,k}^* \\
\geq \max_{a \in \mathcal{C}(U(t))} \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k} - Q_{j,r}) |^+ a_{j,k}.
\]

Hence,

\[
E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) S_{j,k}^* |Q(t)\right] \\
= E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} |^+ S_{j,k}^* |Q(t)\right] \\
\geq \sup_{a \in \mathcal{C}} E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} |^+ a_{j,k} |Q(t)\right] \\
\geq \sup_{\substack{a \in \mathcal{C}, \ a_{j,k} = a_j, 1 \leq k \leq K_j}} E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} |^+ a_{j,k} |Q(t)\right] \\
= \sup_{\substack{a \in \mathcal{C}}} E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} |^+ a_j |Q(t)\right]
\]
\[ \geq \mathbb{E} \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} \right] + \lambda_j |Q(t)| \]

\[ + \epsilon \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k} - Q_{j,r}) \]

because \( \lambda + \epsilon \in \mathcal{C} \).

As,

\[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} Q_{j,k} - Q_{j,r} \]

\[ \geq \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k}(t) - Q_{j,r}(t)) \lambda_j \]

we have

\[ \mathbb{E}[L(Q(t + 1)) - L(Q(t)) | Q(t)] \]

\[ \leq C_2 - \epsilon \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k} - Q_{j,r}) \]
being the leaves of $T_{k_0}$

$$
\sum_{k \in T_{k_0}} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k} - Q_{j,r})
= \sum_{l \in L_{k_0}} (Q_{j,k_0} - Q_{j,l}).
$$ (D.8)

Hence, we have that

$$
\sum_{k \in T_{k_0}} \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k} - Q_{j,r}) \geq l_{j,k_0} \frac{B}{2} \geq \frac{B}{4}.
$$

Thus we show strictly negative drift for sufficiently large $\{Q_{j,k}\}$ and the drift is bounded by $C_2 < \infty$. Hence, by the Foster-Lyapunov theorem in [76] we have that for any initial distribution $\exists d \in \{1, 2, \cdots\}$ s.t. $Q(td) \rightarrow \pi$ in distribution. To prove finite expectation we consider the following.

$$
\mathbb{E}[L(Q(t + 1)) - L(Q(t))|Q(t)]
\leq C_2 - \epsilon \sum_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k}(t) - Q_{j,r}(t)) \right|^{+}
$$

which implies that

$$
\mathbb{E}[L(Q(t + 1)) - L(Q(t))]
\leq C_2 - \epsilon \mathbb{E} \left[ \sum_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k}(t) - Q_{j,r}(t)) \right|^{+} \right].
$$

Summing both sides from 0 to $T$, we get

$$
\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[ \sum_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r} Q_{j,k}(t) - Q_{j,r}(t) \right|^{+} \right]
\leq \frac{1}{\epsilon} \left( C_2 - \frac{1}{T} \mathbb{E}[L(Q(T + 1))] + \mathbb{E}[L(Q(0))] \right).
$$
As $E[L(Q(0))]$ finite, for any initial condition we have

$$\frac{1}{T} \sum_{t=1}^{T} E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \right] < C_3,$$

for all $T$.

As all terms are positive, for any $d \in \{1, 2, \cdots \}$,

$$\lim_{T \to \infty} \frac{d}{T} \sum_{t=1}^{T} E \left[ \sum_{j} \sum_{k=1}^{K_j} \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(td) - Q_{j,r}(td)) \right] < C_3$$

By the ergodicity of a Markov chain in a positive recurrent class this implies that $E_\pi \left[ \sum_{j} \sum_{k=1}^{K_j} | \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(td) - Q_{j,r}(td)) | \right] < C_3$.

This proves that $E_\pi [Q_{j,L_j}] < C_3$ for any leaf node $l_j$.

By Eq. D.8 we have that for any $k \in T_j$,

$$l_{j,k}Q_{j,k} = \sum_{l_j} Q_{j,l_j} + \sum_{k' \in T_k} \sum_{r \in (k')} l_{j,r}(Q_{j,k'} - Q_{j,r})$$

$$\leq \sum_{l_j} Q_{j,l_j} + \sum_{k' \in T_k} | \sum_{r \in (k')} l_{j,r}(Q_{j,k'} - Q_{j,r}) |.$$

Hence, it follows that $E_\pi [Q_{j,k}] < \infty$.

This implies that $E_\pi \left[ \sum_{j} \sum_{k=1}^{K_j} Q_{j,k} \right] < \infty$ and hence, the proof is complete.
D.1.4 Proof of Theorem 45

In deriving Eq. D.6 we did not use any property of the allocation \( \{S^*_{j,k}\} \) other than the fact that it has to satisfy step allocation constraint. Hence, this upper-bound for Lyapunov drift is valid for any arbitrary feasible allocation \( \{S_{j,k}\} \).

Hence, under the LP-relaxation base allocation \( \{S^R_{j,k}\} \) by Lemma 59 we have

\[
E[L(Q(t + 1)) - L(Q(t))|Q(t)] 
\leq C_2 + 2E \left[ \sum_j \sum_{r \in c_j(k)} l_{j,r} (Q_{j,k} - Q_{j,r}) S^R_{j,k} |Q(t) \right]. \quad (D.9)
\]

Note that for the optimum of the problem in Eq. 5.5, \( \{S^R_{j,k}(t)\} \) the following is true.

\[
\sum_j K_j \sum_{k=1}^{K_j} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) S^*_{j,k} 
= \sum_j K_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \right|^+ S^*_{j,k} 
\leq \sum_j K_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \right|^+ \hat{S}_{j,k}
\]

This is because Eq. 5.5 solves a relaxed problem and the optimal allocation has \( S^*_{j,k} = \hat{S}_{j,k} = 0 \) for \( \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \). As \( S^R_{j,k} = \lfloor \hat{S}_{j,k} \rfloor \) we have that

\[
\sum_j K_j \sum_{k=1}^{K_j} \left| \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \right|^+ S^*_{j,k}
\]
\[
\sum_{j=1}^{K_j} \sum_{k=1}^{L_j} \left| \sum_{r \in c_j(k)} l_{j,r}(Q_{j,k}(t) - Q_{j,r}(t)) \right|^+(S_{j,k}^R + 1)
\]

Hence, for any \( \lambda \) s.t. \( \lambda + 1(1+\epsilon) \in C \) using the same proof as above we can show that the system is stable under the LP-relaxation based policy. Rest follows from the crowd-scaling, because \( N \to \infty \) implies \( \lambda_j \to \infty \forall j \in [N] \).

D.1.5 Proof of Theorem 46

This proof has the following structure. As the total number of incomplete jobs is equal to the total number of unallocated steps, we first show that the total number of unallocated steps at depth 0 (i.e., at the root of each \( T_j \)) across all types have the desirable property. Then we show that this property propagates.

**Proof for depth 0 steps**

This part is very similar to the proof of Theorem 35 in Sec. C.1.7.

Consider the different types of unallocated steps at depth 0. These are given by \( \{Q_{j,1}(t) : j \in [N]\} \).

Consider the following processes: for each \( s \in [S] \), \( Q_{1,s}^s(t) = \sum_{j : r_{j,1,s} > 0} Q_{j,1,s}^s(t) \) which represent the number of unserved hours of skills \( s \) for all steps at depth 0.

We now construct another process \( \tilde{Q}_1 \) s.t. it dominates the process \( \sum_s Q_{1,s}^s \). So, if we can show upper-bound on \( \tilde{Q}_1 \), then the same bound applies for \( \sum_s Q_{1,s}^s \). Hence, in turn we get a bound for \( \{Q_{j,1}(t)\} \) (as \( \min\{r_{j,k,s} > 0\} = \Theta(1) \).
by the assumption that \( \{r_{j,k,s}\} \) do not scale with the system size).

Towards constructing a suitable \( \tilde{Q}_1 \) we make the following observation about the dynamics of \( Q_1^s \) and \( \{Q_{j,1}\} \). At each time \( t \), \( \sum_j A_{j,1}(t)r_{j,1,s} \) amount of \( s \) skill hour is brought to add to \( Q_1^s \). Also, this queue gets some service depending on the available agent hours.

At time \( t \), \( \sum_m U_m(t)h_{m,s} \) s-skill hour of service is brought by the agents.

For a step to be allocated, all tasks of it must find an allocation. Hence, for a step in type \( j \)-job to find an allocation it must get \( r_{j,1,s} \) hours of service from each skill \( s \). Thus at any time \( t \) any skill \( s \) queue gets a service of at least

\[
\min_{s \in [S]} \sum_m U_m h_{m,s} - \bar{r},
\]

where \( \bar{r} = \max \{r_{j,k,s}\} \). This is because of the following. For each skill \( \sum_s U_m h_{m,s} \) hour is available. Note that a step can be allocated if all its tasks find allocations, converse of which is also true. That is if all tasks of a step found allocation then the step can be allocated. As \( \min_{s \in [S]} \sum_s U_m h_{m,s} \) hours of service is brought by the agents for each skill at least \( \min_{s \in [S]} \sum_s U_m h_{m,s} - \bar{r} \) of \( s \)-skill hours are served (because a maximum of \( \bar{r} \) can be wasted, as no task is of size more than \( \bar{r} \)). Note that as depth \( d \) steps have priority in Priority Greedy algorithm over steps at depth \( \geq d + 1 \), they do not have to share resource with higher depth steps. So at depth \( d \) \( \min_{s \in [S]} \sum_s U_m h_{m,s} \) is available for service to steps at depth \( \leq d \).
Also, note that the amount of required service brought to the queue $Q^*_1$ at time $t$ is upper-bounded by

$$\max_{s \in [S]} \sum_j A_{j,1}(t)r_{j,1,s}$$

Consider a process $\tilde{Q}^*_1$ with evolution

$$\tilde{Q}^*_1(t + 1) = \max(\tilde{Q}^*_1(t) + \max_{s \in [S]} \sum_j A_{j,1}(t)r_{j,1,s}$$

$$- \min_{s \in [S]} \sum_m U_m h_{m,s} + \bar{r}, 0).$$

Then following exactly same steps as in the proof of Theorem 35 we can show the following.

$$E[\exp(\theta^* \sum_s \tilde{Q}^*_1(0))]$$

$$= E[\exp(\theta^* S\tilde{Q}^*_1(0))]$$

$$= E[\exp(\theta^* S\tilde{Q}^*_1(0))]$$

$$\leq \sum_{\tau \leq 0} S^{2|\tau|} \exp(-\theta^* SK(N)|\tau|)$$

$$\leq c',$$

because $S^2 < \exp(\theta^* SK(N))$ for all sufficiently large $N$.

*Induction over depths: $d$ to $d + 1*

Now we show that if the total number of unallocated steps at depth $d$ satisfies $E[\exp(\theta Q(0))] < c'$ then the same is true for $d + 1$. 

217
To show the same result for steps at all depths we consider the following process. \( d_j(k) \) be the depth of \( k \) in \( T_j \) then \( Q^s_{d+1}(t) = \sum_{j,k:d_j(k) \leq d+1} Q_{j,k} r_{j,k,s} \) represent the number of unserved hours of skills \( s \) for all steps in the system.

Like in the case of the proof for depth 0, we construct processes \( \tilde{Q}^s_{d+1} \) s.t. \( \sum_s \tilde{Q}^s_{d+1} \) dominates the process \( \sum_s Q^s_{d+1} \). Using the same argument as used previously, at any time \( t \) any skill \( s \) queue gets a service of at least

\[
\min_{s \in [S]} \sum_m U_m h_{m,s} - \bar{r},
\]

and the amount of required service brought to the queue \( Q^s \) at time \( t \) is upper-bounded by

\[
\max_{s \in [S]} \sum_{j,k:d_j(k) \leq d+1} A_{j,k}(t) r_{j,k,s}
\]

Then using the same argument, the process

\[
\tilde{Q}_{d+1}(t+1) = \max(\tilde{Q}_{d+1}(t) + S \max_{s \in [S]} \sum_{j,k:d_j(k) \leq d+1} A_{j,k}(t) r_{j,k,s} - \min_{s \in [S]} \sum_m U_m h_{m,s} + \bar{r}, 0)
\]

upper-bounds the process \( \sum_s Q^s_{d+1} \). Then we can follow the steps that we followed using \( X_s \) and \( Y_s \) previously. Let \( X'_s := \sum_{j,k:d_j(k) \leq d+1} A_{j,k}(t) r_{j,k,s} \) and \( Y'_s := \sum_m U_m h_{m,s} \) respectively. But note that \( A_{j,k} \) for \( k > 1 \) is not an external i.i.d process, rather it is the number of steps of type \( A_{j,p_j(k)} \) that got completed. Hence, we cannot follow the exactly same steps. Note that
\[
\mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (\max_s X'_s(t) - \min_s Y'_s(t) + \bar{r}))]
\leq \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} \max_{s,s' \in [S]} (X'_s(t) - Y'_s(t) + \bar{r}))]
\leq \mathbb{E}[\sum_{s,s' \in [S]} \exp(\theta \sum_{\tau \leq t \leq 0} (X'_s(t) - Y'_s(t) + \bar{r}))]
= \sum_{s,s' \in [S]} \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (X'_s(t) - Y'_s(t) + \bar{r}))].
\]

Note that,
\[
\sum_{\tau \leq 0} \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} \max_{s,s' \in [S]} (X'_s(t) - Y'_s(t) + \bar{r}))]
\leq \sum_{s,s' \in [S]} \sum_{\tau \leq t \leq 0} \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (X'_s(t) - Y'_s(t) + \bar{r}))].
\]

So, we investigate into \( \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (X'_s(t) - Y'_s(t) + \bar{r}))]. \)

\[
\mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (X'_s(t) - Y'_s(t) + \bar{r}))]
= \exp(\bar{r}\theta) \mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (\sum_{j,k} A_{j,k}(t)r_{j,k,s} - \sum_m U_m(t)h_{m,s'}))]
= \exp(\bar{r}\theta) \mathbb{E}[\exp(\theta(\sum_{j,k,d_{j,k}(k) \leq d+1} \sum_{\tau \leq t \leq 0} A_{j,k}(t)
- \sum_m \sum_{\tau \leq t \leq 0} U_{m}(t)h_{m,s'}))].
\]

Note that \( \sum_{\tau \leq t \leq 0} A_{j,k}(t) \) are the creation (or appearance/arrival) of steps of type \((j, k)\) between time \(\tau\) to 0 (and a similar interpretation for agents in case of \( \sum_{\tau \leq t \leq 0} U_m(t) \)), which we denote by \( A_{j,k}(\tau : 0) \) (and \( U_m(\tau : 0) \)) respectively.
Now there is an important observation about $A_{j,k}(\tau : 0)$,

$$A_{j,k}(\tau : 0) \leq Q_{j,p_j(k)}(\tau - 1) + A_{j,p_j(k)}(\tau - 1 : -1)$$  \hspace{1cm} (D.10)

where $p_j(k)$ is the parent of $k$ in $T_j$. This is because of the following. As each job takes one slot to be served, no job whose step $(j, p_j(k))$ completed after $-1$ can have its step $(j, k)$ be available for service at or before 0. Thus by induction on the function $p_j$ we can write that

$$A_{j,k}(\tau : 0) \leq \sum_{w=1}^{d} Q_{j,w}(\tau - 1 - d + w)$$
$$+ A_{j,1}(\tau - d_j(k) : -d_j(k)),$$  \hspace{1cm} (D.11)

as $d_j(k) = d + 1$ by induction assumption. Note that from 1 to $k$ (at depth $d + 1$) there is a unique $d$ length path and hence, on that path w.l.o.g. we denote the respective steps by $(j, w)$ where $w$ is its depth on that path.

Hence,

$$\mathbb{E}[\exp(\theta(\sum_{j,k:d_j(k)\leq d+1} r_{j,k,s} \sum_{\tau \leq t \leq 0} A_{j,k}(t)$$
$$- \sum_{m} \sum_{\tau \leq t \leq 0} U_m(t)h_{m,s'}))]$$

$$\leq \mathbb{E}[\exp(\theta(\sum_{j,k:d_j(k)\leq d+1} r_{j,k,s}(\sum_{w=1}^{d} Q_{j,w}(\tau - 1 - d + w)$$
$$+ A_{j,1}(\tau - d_j(k) : -d_j(k)) - \sum_{m} \sum_{\tau \leq t \leq 0} U_m(t)h_{m,s'}))]$$  \hspace{1cm} (D.12)
Note that $\sum_m \sum_{\tau \leq t \leq 0} U_m(t) h_{m,s'}$ is independent of
\[
\sum_{j,k:d_j(k) \leq d+1} r_{j,k,s} \left( \sum_{w=1}^d Q_{j,w}(\tau - 1 - d + w) + A_{j,1}(\tau - d_j(k) : -d_j(k)) \right),
\]
(D.13)
because $A_{j,1}$ are i.i.d (independent of $U_m$) and $Q_{j,w}(\tau - d + w)$ does not depend
on $U_m(\tau : 0)$ for $d \geq w \geq 1$. Hence,
\[
\mathbb{E}[-\exp(\theta \sum_m \sum_{\tau \leq t \leq 0} U_m(t) h_{m,s'})] 
\]
\[
\leq \mathbb{E}[-\exp(\theta \sum_{j,k:d_j(k) \leq d+1} r_{j,k,s} \left( \sum_{w=1}^d Q_{j,w}(\tau - 1 - d + w) + A_{j,1}(\tau - d_j(k) : -d_j(k)) \right)) \mathbb{E}[\exp(-\theta \sum_m \sum_{\tau \leq t \leq 0} U_m(t) h_{m,s'})].
\]
We use the previously derived bound for $\mathbb{E}[\exp(-\theta \sum_m \sum_{\tau \leq t \leq 0} U_m(t) h_{m,s'})]$. So, we only concern ourselves with
\[
\mathbb{E}[-\exp(\theta \sum_{j,k:d_j(k) \leq d+1} r_{j,k,s} \left( \sum_{w=1}^d Q_{j,w}(\tau - 1 - d + w) + A_{j,1}(\tau - d_j(k) : -d_j(k)) \right)).
\]
(D.14)
Consider any $Q_{j,w}(\tau - 1 - d + w)$ at depth $w$, then $A_{j,1}(\tau - d - 1)$ is independent
of it. As $A_{j,1}$ are i.i.d and future arrivals in a queue are independent of present
and past queue-lengths, we have
\[
\mathbb{E}[-\exp(\theta \sum_{j,k:d_j(k) \leq d+1} r_{j,k,s} \left( \sum_{w=1}^d Q_{j,w}(\tau - 1 - d + w) + A_{j,1}(\tau - d_j(k) : -d_j(k)) \right))].
\]
For the second term we obtain a bound using previous techniques and note that as \( \lambda \in \alpha \mathcal{C} \)

\[
\sum_{j,k : d_j(k) \leq d+1} r_{j,k,s} A_{j,1} \leq \sum_m \mu_m h_{m,s'}
\]

which in the same way as above will imply that for some \( K(N) \) and some \( \theta > 0 \),

\[
\mathbb{E}[\exp(\theta(\sum_{j,k : d_j(k) \leq d+1} r_{j,k,s} A_{j,1}(\tau - d_j(k)) : -d_j(k))) - \sum_m \sum_{\tau \leq t \leq 0} U_m(t) h_{m,s'})] \\
\leq \exp(-\theta K(N) \tau).
\](D.15)

Note that

\[
\mathbb{E}[\exp(\theta(\sum_{j,k : d_j(k) \leq d+1} r_{j,k,s} \sum_{w=1}^{d} Q_{j,w}(\tau - 1 - d + w))) < \infty
\]

by the induction assumption that the number of unallocated steps at depth \( \leq d \) have finite exponential moments.
So we have that

$$\mathbb{E}[\exp(\theta \sum_{\tau \leq t \leq 0} (X_s(t) - Y_s(t) + \bar{r}))] < c_1 \exp(-\theta K(N)\tau),$$

and hence, in turn (using the same steps as above) $Q^*$ has finite exponential moment for some $\theta$. Rest of the steps are similar as above and hence we get the desired result that

$$\mathbb{E}[\exp(\theta \sum_{j,k} Q_{j,k})] < \infty$$

Thus by inducting on $d$ we prove that the total number of unallocated steps over all types of jobs have finite exponential moment (say $c'$).

Hence,

$$\mathbb{P}(\sum_{j,k} Q_{j,k} > q)$$

$$\leq \exp(-\theta q)\mathbb{E}[\exp(\theta \sum_{j,k} Q_{j,k})]$$

$$\leq c' \exp(-\theta q).$$

So, for $q = \frac{3 \log N}{\theta}$, we have the result (as $c'$ is constant).
Bibliography


[33] M. Meadows and D. Cliff, “The relative disagreement model of opinion dynamics: where do extremists come from?”


Vita

Avhishek Chatterjee received his bachelor degree (B.E.) from Jadavpur University, India and master degree (M.E.) in Telecommunication from the Electrical Communication Engineering department of the Indian Institute of Science (I.I.Sc.), Bangalore, India in 2006 and 2008 respectively. Since 2011 he has been working towards his Ph.D. in the Electrical and Computer Engineering department of The University of Texas at Austin.

Permanent email: avhishek@utexas.edu

This dissertation was typeset with $\LaTeX$\textsuperscript{†} by the author.

\textsuperscript{†}$\LaTeX$ is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth’s $\TeX$ Program.