Optimization and Decision-Making in Decentralized Finance, Scheduling, and Graphical Game Theory

Utkarsh Patange

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2024

© 2024

Utkarsh Patange

All Rights Reserved

Abstract

Optimization and Decision-Making in Decentralized Finance, Scheduling, and Graphical Game Theory

Utkarsh Patange

We consider the problem of optimization and decision-making in various settings involving complex systems. In particular, we consider specific problems in decentralized finance which we address employing insights from mathematical finance, in course-mode selection that we solve by applying mixed-integer programming, and in social networks that we approach using tools from graphical game theory.

In the first part of the thesis, we model and analyze fixed spread liquidation lending in DeFi as implemented by popular pooled lending protocols such as AAVE, JustLend, and Compound. Empirically, we observe that over 70% of liquidations occur in the absence of any downward price jumps. Then, assuming the borrowers monitor their loans with exponentially distributed horizons, we compute the expected liquidation cost incurred by the borrowers in closed form as a function of the monitoring frequency. We compare this cost against liquidation data obtained from AAVE protocol V2, and observe a match with our model assuming the borrowers monitor their loans five to six times more often than they interact with the pool. Such borrowers must balance the financing cost against the likelihood of liquidation. We compute the optimal health factor in this situation assuming a financing rate for the collateral. Empirically, we observe that borrowers are often more conservative compared to model predictions, though on average, model predictions match with empirical observations.

In the second part of the thesis, we consider the problem of hybrid scheduling that was faced by Columbia Business School during the Covid-19 pandemic and describe the system that we implemented to address it. The system allows some students to attend in-person classes with social distancing, while their peers attend online, and schedules vary by day. We consider two variations of this problem: one where students have unique, individualized class enrollments, and one where they are grouped in teams that are enrolled in identical classes. We formulate both problems as mixed-integer programs. In the first setting, students who are scheduled to attend all classes in person on a given day may, at times, be required to attend a particular class on that day online due to social distancing constraints. We count these instances as "excess." We minimize excess and related objectives, and analyze and solve the relaxed linear program. In the second setting, we schedule the teams so that each team's in-person attendance is balanced over days of week and spread out over the entire term. Our objective is to maximize interaction between different teams. Our program was used to schedule over 2,500 students in student-level scheduling and about 790 students in team-level scheduling from the Fall 2020 through Summer 2021 terms at Columbia Business School.

In the third part of the thesis, we consider a social network, where individuals choose actions which optimize utility which is a function of their neighbors' actions. We assume that a central authority aiming to maximize social welfare at equilibrium can intervene by paying some cost to shift individual incentives, and that the cost is upper bounded by a budget. The intervention that maximizes the social welfare can be computed using the spectral decomposition of the adjacency matrix of the graph, yet this is infeasible in practice if the adjacency matrix is unknown. We study the question of designing intervention strategies for graphs where the adjacency matrix is unknown and is drawn from some distribution. For several commonly studied random graph models, we show that the competitive ratio of in intervention proportional to the first eigenvector of the expected adjacency matrix, approaches 1 in probability as the graph size increases. We also provide several efficient sampling-based approaches for approximately recovering the first eigenvector when we do not know the distribution. On the whole, our analysis compares three

categories of interventions: those which use no data about the network, those which use some data (such as distributional knowledge or queries to the graph), and those which are fully optimal. We evaluate these intervention strategies on synthetic and real-world network data, and our results suggest that analysis of random graph models can be useful for determining when certain heuristics may perform well in practice.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis would not have been possible but for the love, support and guidance that I was fortunate to receive from those around me. My advisor, Prof Ciamac Moallemi, has been a guiding force steering the shape and the direction of this thesis. His talents, intellect and expertise in the subject matter have been constant sources of motivation for me, and I am incredibly grateful to have been a beneficiary of his mentorship.

I would also like to thank Prof Yash Kanoria, who has always inspired me and encouraged me to better myself. His support and generosity has been humbling and I am eternally grateful to him. I would like to thank the other committee members on my defense committee: Paul Glasserman, Agostino Capponi, and Tim Roughgarden whose feedback and comments helped me improve this dissertation. I am also thankful to Prof Santiago Balseiro and Prof Daniel Guetta for their time and efforts which have been invaluable to me and to Prof Omar Besbes and Prof Jing Dong for their guidance which I was fortunate to receive from time to time.

I am thankful to the Columbia Business School community, including Razwan Popescu and Benny Chang from the research support group, Clara Magram, Kira Grant, Andrew Cassidy, Winnie Leung, Cristina Melo-Moya from the DRO division. Their assistance and support have been greatly appreciated. I am also thankful to Columbia residential for providing a home to me, and to Columbia health services for helping me maintain a healthy lifestyle. The contributions of these institutions have been vital to the completion of this thesis.

I am grateful to Jonah Rockoff and Ben Golub for helpful discussions and inspirations during my projects. I am also thankful to anonymous reviewers for their reviews which have greatly

improved my papers upon which this thesis is based.

To my fellow students and friends — Omar Mouchtaki, Mike Li, Shangzhou Xia, Cecilia Li, Yuri Fonseca, Carlos Bonet, Yuanling Gan, and Sven Sabas — thank you for the stimulating helpful discussions and the sense of camaraderie. Your support has been like a family throughout these years.

Finally, I would like to express my deepest gratitude to my parents, Dr. Sanjay and Shubhangi Patange for their unconditional love. I am also thankful to my brother, Himanshu Patange for always believing in me. My heartfelt thanks go to my wife, Shalaka Sitre, whose love and support have carried me through this work.

Preface

For a long time, I have been interested in practical applications of optimization. Interesting, non-trivial decision-making problems arise in all areas of life, and each of these problems needs to be studied and addressed individually, often employing a diverse set of tools to come up with an (near-)optimal decision. This dissertation is a compilation of three such problems and describes the tools that I, along with my co-authors, used to address them.

Chapter 1 is based on my work in collaboration with my advisor Professor Ciamac Moallemi. In this work, we study fixed-spread liquidation lending platforms, which are among the most widely used lending platforms in decentralized finance. These platforms codify interactions between lenders, borrowers, and liquidators, who each act strategically to optimize their own payoffs. The lenders provide liquidity to the lending pools on the platform from which borrowers are allowed to borrow against a collateral. The liquidators are incentivized to keep the loans solvent by partially liquidating under-collateralized loans. We come up with a model for this interaction, and compute the liquidation cost faced by the borrowers under optimal behavior by the liquidators. We use this cost to compute optimal collateralization ratio that the borrowers must maintain to balance the liquidation and financing costs. We also consider a nefarious agent who can manipulate prices and benefit from the platform, and delineate platform parameters that allow such price manipulation attacks.

Chapter 2 is also based on my work in collaboration with Professor Ciamac Moallemi. This work was motivated by the need to conduct MBA classes in a hybrid mode to comply with social distancing guidelines while also allowing students to attend classes in person during the

COVID-19 pandemic at Columbia Business School. We divide the student population into two categories, and come up with mixed-integer programs to compute the hybrid schedules of these categories by solving them.

Chapter 3 is based on my work in collaboration with William Brown. This evolved from a course project that we did for a class taught by Yash Kanoria. In this work, we consider a social network on which a game is being played. The utility of an agent is a function of their own action, as well as those of their neighbors. A few "central" agents in such a network can have an outsized influence on the actions taken by others in the network. We assume that a third party can incentivize some agents to take an action and examine the problem of identifying central agents in the network and incentivize them so as to maximize the social welfare. [1] study this problem of finding an optimal intervention in the case where the exact social network is known. In real world social networks, however, it may be costly to map the exact graph. The practical decision-making problem in this scenario is to come up with a near-optimal intervention that can be computed without fully mapping the graph. Specifically, we consider this problem for the case where we do not know the full graph, but only the distribution from which it was drawn.

Chapter 1: An Analysis of Fixed-Spread Liquidation Lending in DeFi[1]

1.1 Introduction

Lending is one of the core functions of finance. In traditional as well as decentralized finance, the lenders are protected from counter-party risk through over-collateralization: the practice of using an asset valued more than the lent amount as a guarantee against the loan being repaid. In this paper, we model and study collateralized lending in decentralized finance (DeFi) as it is practiced by the most popular DeFi lending platforms today. In particular, three lending platforms (AAVE, JustLend, and Compound Finance) contribute $10.76 billion in total value locked (TVL) of the $14.73 billion TVL in DeFi lending platforms as of June 8, 2023. All three platforms use a protocol similar to the one we study here.

DeFi lending is most commonly implemented through lending pools (for example, the AAVE V2 lending pool). Users deposit their cryptocurrency tokens in the pool to earn interest. These deposits are lent by the platform to other users. To borrow, a user needs to have sufficient collateral available in their account to support their debt. A user can use multiple assets as collateral against their loan, which in turn, can also consist of multiple cryptocurrencies. A user continues to earn interest on their collateral assets even after borrowing against them. The interest rates charged by the platform, and those earned by the depositors depend on the utilization of the pool, and can change with the supply and demand of the particular cryptocurrencies. In the present work, we shall assume constant interest rates. We shall also assume that the collateral and the debt each consist of a single asset. We formalize our model and assumptions in Section 1.2.

Any collateralized lending protocol needs a mechanism to sell the collateral and repay the loan if the collateral value drops. DeFi lending protocols, handle this by incentivizing third-party

[1]This chapter is based on original work in collaboration with Ciamac Moallemi[2].

liquidators. These liquidators repay a fraction of the loan, and in return receive collateral in a larger amount. Being risk-averse themselves, the liquidators trade this collateral in on-chain decentralized exchanges (DEXes) to receive tokens of a stable currency. A profit-maximizing liquidator, thus, must consider liquidity in these DEXes. In Section 1.3, we consider such a liquidator and compute the optimal fraction of liquidation in different levels of loan health and liquidity.

Because the liquidators receive payment from the borrower's collateral, a rational borrower always has an incentive to prevent liquidations. In particular, a rational borrower who continuously monitors their debt and collateral value would repay part of their loan if the collateral value drops to a certain threshold, possibly by using a flashloan to free up and sell some of the collateral, instead of letting the loan be liquidated and allowing liquidators to receive some of the collateral. Thus, the loan would never get liquidated if the price process is continuous, and the borrower monitors the loan continuously. However, in practice, loans do get liquidated, suggesting either that the price process has jumps or that the borrowers do not monitor their loans. In Section 1.5.3, we establish that a vast majority of liquidations occur without any downward jump in the price process around that time period. Therefore, in this paper, we shall consider a "passive" borrower — one who is unable to monitor the state of their loan continuously — who takes out the loan but does not track the loan state until a later time.

Such a borrower can either face a high risk of liquidation, or deposit a large amount of collateral and weather the resulting opportunity cost. We consider the problem of balancing these trade-offs in Section 1.4. In Section 1.5, we use data to validate our model by comparing observed liquidation costs to model prediction, and the borrowers' health factors with our model optima.

Moreover, these lending pools, by design, are sensitive to prices. In particular, the price of the collateral determines both how much can be borrowed against one unit, and when a loan can be liquidated. As such, in presence of price impacts to trading, predatory agents can manipulate these prices and potentially profit by exploiting the lending pools. Indeed, such price manipulations have been observed in practice. Notably, an individual was able to inflate the price of MNGO coin, borrow against it, and default on the $116 million loan thus acquired [3]. We consider the

possibility of such price manipulation attacks in Section 1.6.

Our goal is to use tools from quantitative finance, namely the risk neutral pricing framework of the Black-Scholes model, in order to better guide usage and design of these lending protocols. As such, throughout the paper, we assume a continuous price process.[2] In this setting, we make the following contributions:

1. We compute the optimal fraction of liquidation for a profit-maximizing liquidator under a liquidity model based on a constant product market maker.

2. We compute the cost of liquidation for a passive borrower and compute optimal quantity of the collateral that balances the cost of capital against the liquidation cost. Our results provide guidance to borrowers regarding the trade-offs between different levels of collateralization based on the volatility in the asset and the borrower's monitoring frequency and cost of capital.

3. Empirically, we observe that a vast majority of liquidations occur in time periods in which the null hypothesis of price continuity cannot be rejected.

4. We compute the average liquidation costs borne by the borrowers for different health factors and compare with model predictions, and observe a match assuming the loans are monitored five to six times more often than the frequency at which borrowers interact with the pool.

5. We compare the health factors maintained by the borrowers with the model optima and observe that the borrowers start with health factors that are, more often, higher than the model optima.

6. When there is a price impact to trading, we observe that predatory agents can benefit from price manipulation. We identify conditions on contract parameters that allow such profitable price manipulations under our assumed price-impact model. Our results can be used by lending pool designers to set and update contract parameters.

[2] In practice, the smart contracts rely on price oracles that update the prices at discrete time instances. We instead assume a continuously updating oracle in addition to a continuous price process.

Related Papers. With growing interest in DeFi protocols, there have been quite a few recent papers that study DeFi lending. A large number of these papers study the inherent fragility of these protocols. [4] systematize the knowledge about lending pools and formally model user interactions with such pools. Using this model, they are able to analyze the vulnerabilities in the lending platforms. [5] study liquidations in such lending protocols from an empirical standpoint. They consider potentials to abuse the existing system and hurt the borrower, and suggest an alternative liquidation mechanism. [6] observe an inherent systemic fragility in the DeFi lending markets when there is price impact to trading. This stems from liquidators selling the collateral and thereby, moving the prices further, causing cascading liquidations of other loans. [7] also study such liquidation spirals, and recommend changes to existing liquidation protocols to prevent their harmful effects. [8] propose a new financial instrument called a "reversible call option" to mitigate liquidations, and potentially strengthen the market.

Other related papers focus on the ways attackers can take advantage of this fragility, and suggest methods to address them. [9] systematize the knowledge about attacks on DeFi protocols in general. [10] present a novel governance attack strategy that would have allowed an attacker to steal 0.5bn USD worth of collateral, and mint an unlimited supply of DAI tokens. [11] argue for redundancy in program logic to minimize severity and frequency of DeFi attacks and provide a novel algorithm to implement it for smart contracts. [12] present a framework for analyzing risk in fixed-spread liquidation lending protocols and show that the liquidation incentive, which is necessary to keep loans solvent, also acts as an incentive for the liquidators to manipulate prices and cause liquidations of loans having a health factor close to one.

Closer to this work are papers that model and analyze the design and usage of DeFi lending platforms. [13] evaluate the economic security of the Compound protocol. [14] empirically examine the data on interest rates, lending pool utilization, and conduct a liquidity study of the markets for DAI, ETH and USDC across AAVE, Compound and dYdX. [15] formally analyze lending pools using a statistical analyzer and show how such an analysis can be used to find threshold and reward parameters that reduce the risk of a loan becoming unrecoverable. [16] investigate how the

Compound protocol is used by its users, and identify systemic risks arising from such a usage. [17] develop an evaluation model for DeFi lending protocols that can be used by participants to decide which protocol to use.

There are also papers that use lending protocols as a tool to design other applications, and analyze them. [18] study non-custodial stablecoins that arise from lending markets, and suggest design improvements for their long-term stability. [19] develop a stochastic model to study such non-custodial stablecoins. [20] suggest allowing liquidity provider (LP) shares on constant function market makers to be borrowed to improve capital efficiency for LPs.

The present work considers the costs and incentives in DeFi lending from a theoretical perspective. While we have not encountered such an approach in other DeFi related papers, the techniques we use are not new and have been used in traditional finance literature for a long time.

1.2 Model

A DeFi loan is a contract between a borrower and a lender. Under the contract, the borrower, at time $t = 0$, borrows V_0 units of currency C (numéraire) against N_0 units of asset (collateral) $\mathcal{B}$. Both the borrowed amount and the collateral accrue interest with time at constant rates γ_V and γ_C respectively from the lending pool. At time t, the loan amount V_t and the collateral amount N_t evolve so that $dV_t = V_t\gamma_V\, dt,\ dN_t = N_t\gamma_C\, dt$. Note here that both the loan and the collateral accrue interest. The lending platform is able to pay this interest to the depositors because it rehypothecates the collateral as a loan to other borrowers. The borrow rate and deposit rate (sometimes known as the "supply" rate) are determined by the supply and demand of the respective assets, and are important tools the platform employs to balance supply and demand of each asset. The risks a platform faces due to this practice is out of the scope of this paper.

Let the spot price of the collateral $\mathcal{B}$ be p_t. The total value of the collateral in the loan at time t is given by $N_t p_t$. While the loan is unpaid, the borrower does not have access to their collateral, and so as long as $V_t < N_t p_t$, the borrower has an incentive to pay back the loan and free up the collateral since the value of the collateral exceeds the value of the loan. Let T be the first time at

the borrower observes the loan after its origination. We call T the monitoring horizon. The idea is that when the loan is originated, the borrower must decide the level of collaterallization, and is unable to adjust the collateral over the interval $(0, T)$. At time T, the borrower again observes the state of the market and the state of the loan, and can revisit this decision.

The lender is exposed to the market risk of the collateral. As is common with collateralized loans, this is addressed by over collateralization. This is implemented by defining a borrowing capacity, c_t, for the borrower given by $c_t := N_t p_t \ell$, where $\ell \in (0, 1)$ is known as the liquidation threshold. Similar to loans in traditional finance, once c_t drops to a value smaller than V_t the loan becomes "unhealthy" and eligible for liquidation. For convenience the ratio $h_t := c_t / V_t$ is defined to be the *health factor*, so that the loan is unhealthy if $h_t < 1$. Having a borrowing capacity less than the value of the collateral, thus, gives the lender some respite from the market risk associated with the collateral. If the value of the collateral drops to below the value of the debt, the loan can be said to be "defaulted" as the borrower no longer has an incentive to pay back the loan. Figure 1.1 depicts the three scenarios: a healthy loan, an unhealthy, but not defaulted loan, and an unhealthy and defaulted loan.

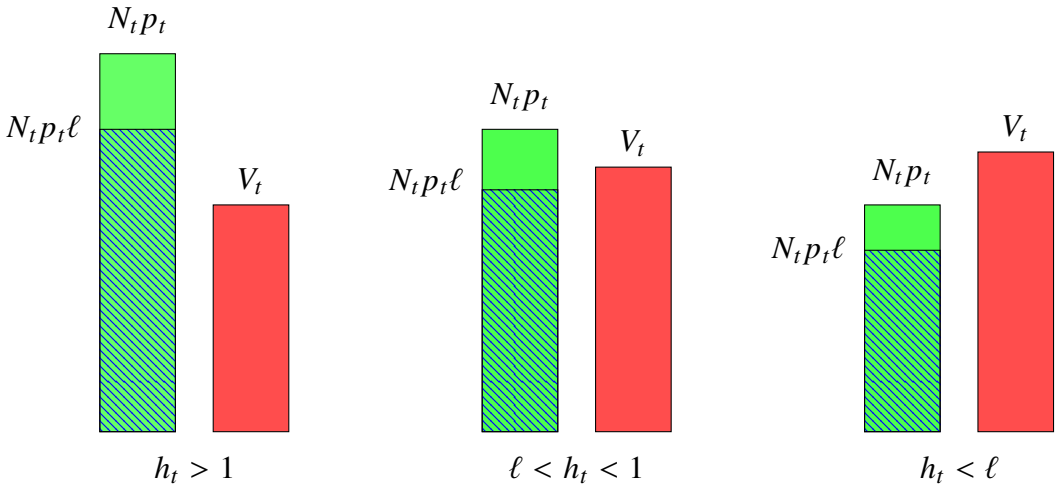


Figure 1.1: Collateral value (shaded+unshaded columns on the left), borrowing capacity (shaded parts of the columns on the left), and debt value in healthy (far left), unhealthy but not defaulted (middle), and unhealthy and defaulted (far right) loans.

The assets used in DeFi can be much more volatile than those in traditional finance and thus

further safeguards are desirable. These safeguards are implemented by paying a third party, known as a liquidator, to liquidate some fraction of the loan once the loan becomes unhealthy. The liquidators are paid in the collateral, and thus assume the market risk. To prevent the borrower from losing too much of their collateral, the fraction of the loan a liquidator can liquidate is bounded from above by $F \in (0, 1]$, known as the *close factor*. Formally, the loan evolves as per the following steps:

1. At time $t = 0$, the borrower starts the loan by borrowing V_0 units of currency $\mathcal{C}$ against a collateral of N_0 units in currency $\mathcal{B}$. The contract parameters ℓ, F, λ, and the interest rates γ_C, γ_V are decided by the platform.

2. At time t, if $c_t < V_t$, a liquidator is allowed to liquidate the unhealthy loan as follows:

   (a) The liquidator decides on a fraction $f \in (0, F)$ to liquidate, and pays $V_t f$ units in currency $\mathcal{C}$ to the lender. The contract pays back the same value in currency $\mathcal{B}$ priced at $p_t/(1 + \lambda)$ per unit. The fraction f is said to be the "fraction of liquidation". If $f = F$, we say that it is a "close-factor liquidation".

   (b) The loan now has debt $V_{t+dt} = V_t(1 - f)$ units in currency $\mathcal{C}$ and collateral $N_{t+dt} = N_t - V_t f(1 + \lambda)/p_t$ in currency $\mathcal{B}$.

Note that the liquidator receives amount $V_t f(1 + \lambda)/p_t$ in collateral, and earns a profit of $\lambda V_t f$ if it can be sold at price p_t. This mechanism is thus referred to as *fixed spread liquidation*, λ being called the "liquidation spread". Collectively, we call F, ℓ, and λ as the contract parameters. The following example demonstrates the liquidation process.

Example 1.1 *Suppose the contract parameters are given by* $\ell = 0.5, F = 0.5, \lambda = 0.08$ *for a loan of USDC against a collateral of WETH. A borrower borrows* $3,000$ *USDC at time* $t = 0$ *against a collateral of* 5 *WETH priced at* $1,800$ *USDC each. The starting health factor is then* $h_0 = \frac{5 \times 1800 \times 0.5}{3000} = 1.5$. *Let us assume that the borrower is passive, and only monitors the loan from time to time. Let us further assume that the interest rates* γ_C *and* γ_V *are 0.0. Figure 1.2 shows the progression of the health factor of this loan over time.*

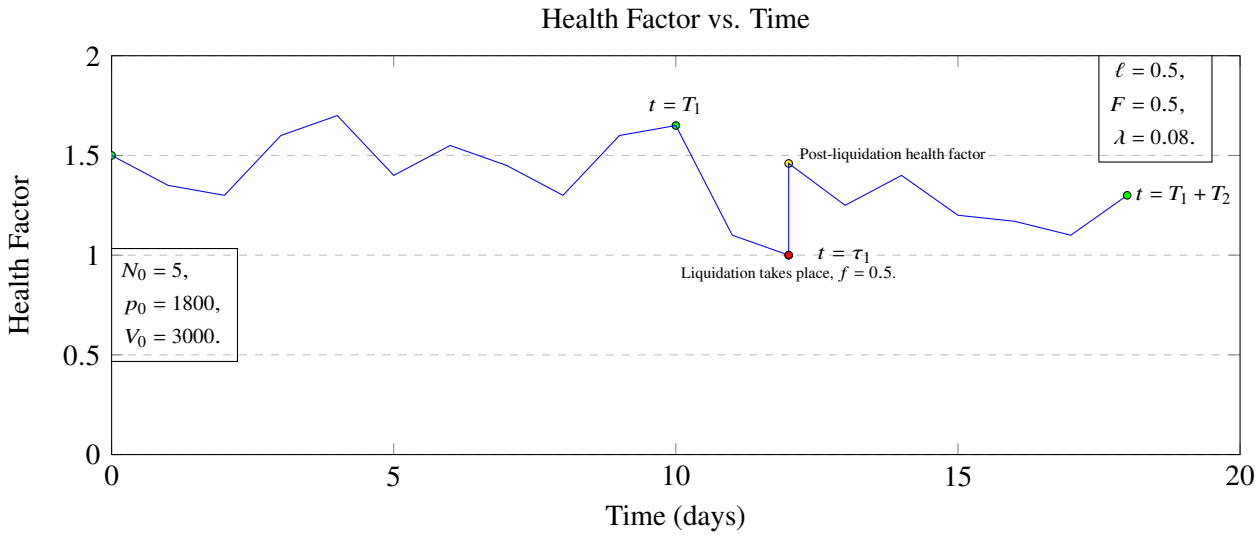


Figure 1.2: Progression of the health factor of the loan in Example 1.1.

The borrower next monitors the loan after $T_1 = 10$ *days. At this time, the price is given by* $p_{T_1} = 1,980$ *USDC/WETH, and hence the health factor is* $h_{T_1} = 1.65$*. Since the health factor has stayed above 1.0, there has been no liquidation so far. At this point, the borrower may want to rebalance the loan by selling some of the collateral, or by borrowing more debt. In our example, the borrower chooses not to do this.*

In the next two days, the price drops to $1,200$ *USDC/WETH, which results in the health factor becoming* 1.0 *and the loan becoming unhealthy and eligible for liquidation. We denote this time by* τ_1*. At this time, a liquidator liquidates the loan at fraction* $f = F = 0.5$*. They pay* $1,500$ *USDC to the lender, and receive* 1.35 *WETH from the collateral of the loan. The remaining debt is thus* $1,500$ *USDC against a collateral of* 3.65 *WETH, giving us a post-liquidation health factor* $h_{\tau_1+dt} = 1.46$*. Thus, the loan has become healthy after liquidation.*

The borrower is unaware of the liquidation, and monitors the loan only after $T_2 = 8$ *days have elapsed since the last monitoring. At this point, the loan is healthy, but the borrower observes less collateral and less debt than before, indicating a prior liquidation. In that liquidation, the borrower had paid the cost of liquidation to the liquidator in the form of the liquidation spread. As this is another point of monitoring, the borrower may again choose to rebalance the loan by either repaying the debt, or putting in more collateral.*

In Example 1.1, the health factor of the loan improved after liquidation. In Appendix A.1, we show that this always happen as long as $\ell(1+\lambda)$ is less than the health factor. Essentially, this condition ensures that there is sufficient collateral to pay for the liquidation spread. In the rest of the paper, we assume that the following condition is satisfied by the contract parameters.

Assumption 1.1 (Collateral Sufficiency) $\ell(1+\lambda) < 1$.

We observe in Appendix A.1 that this condition also implies full recovery of the loan in presence of continuous prices.

1.3 Optimal Liquidations

Before analyzing borrower costs and behavior, we must first consider the behavior of the liquidators. As discussed in Section 1.2, liquidators are motivated by profits, and as such, would not participate in liquidations if doing so is not profitable. Liquidators are paid in risky collateral, and must then sell it immediately lest they be subject to the market risk. To mitigate this risk, they use "flash loans". Flash loans are loans that can be borrowed at a fee and must be repaid in the same transaction. This allows liquidators to atomically obtain a flash loan in the riskless asset C, liquidate the unhealthy loan, sell the collateral thus obtained, and finally repay the initial flash loan along with the associated fee. As the flashloan must be repaid atomically, the liquidator must use on-chain decentralized exchanges (DEXes) to sell the collateral. Automated market makers (AMMs) such as Uniswap, which maintain liquidity pools of pairs of assets and allow one to be exchanged for the other are the most popular DEXes in use as of this writing. These have been analyzed in multiple papers including [21], [22], and [23]. In this section, we assume a constant product market maker which has an initial price equal to the oracle price when liquidation takes place. We now formalize such a market maker.

Suppose that at time t, the $C|\mathcal{B}$ liquidity pool in the market maker platform contains u_t tokens of riskless asset C, and b_t tokens of risky asset $\mathcal{B}$. Ignoring trading fees, the platform ensures that the product $u_t \cdot b_t = L^2$ remains constant. To buy tokens of $\mathcal{B}$, an agent deposits Δc tokens of C,

and they are allocated Δb tokens of $\mathcal{B}$ such that $(u_t + \Delta c) \cdot (b_t - \Delta b) = L^2$. That is,

$$\Delta b = b_t - \frac{L^2}{u_t + \Delta c} = \frac{b_t \Delta c}{u_t + \Delta c}.$$

The instantaneous price $\hat{p}_t$ of an infinitesimal number of tokens of $\mathcal{B}$ in terms of C is thus given by

$$\hat{p}_t = \lim_{\Delta c \to 0} \frac{\Delta c}{\Delta b}\bigg|_t = \frac{u_t}{b_t}.$$

Since price is exogenous, it makes more sense to replace u_t by $L\sqrt{\hat{p}_t}$, and b_t by $L/\sqrt{\hat{p}_t}$. We note that such a constant product market maker incorporates the price slippage since the average price of a trade becomes worse with trade size.

We assume that at time t just before liquidation, the instantaneous price as per the liquidity pool composition, $\hat{p}_t$, is equal to the oracle price given by p_t. When liquidating the loan, the liquidator must take the composition of the liquidity pools into account. Liquidating a large fraction increases the liquidation bonus, but it also costs more to sell it on the DEX. A profit-maximizing liquidator thus does not liquidate the loan beyond a certain point due to liquidity constraints. We note here that we are not concerned about any permanent price impact to the collateral price due to this liquidation, though there certainly can be some that would affect future loan health. The price slippage only concerns us in the amount that it affects the immediate trading costs.

If there is enough liquidity (i.e., the price slippage is small), then the profit maximizing liquidator is still constrained by the available collateral supporting the loan. In general, the liquidator may liquidate a fraction greater than the close factor by performing multiple liquidations while ensuring that the loan in the intermediate stages remains unhealthy. We call the total liquidated fraction from multiple liquidations as the resultant fraction of liquidation (f_{res}), and we denote the optimal feasible resultant fraction of liquidation by f^*_{res}. To compute f^*_{res}, we need to identify cases where multiple liquidations are possible. When the health factor is h, liquidation at fraction f brings the post-liquidation health factor to $\tilde{H}(h, f)$ defined by equation A.1. As per Lemma A.1.1, if $h_t > \ell(1+\lambda)$, then the health factor improves upon liquidation. As observed by [5], in the absence

Condition	$f_{\max} < f_{\text{opt}}$	$f_{\max} > f_{\text{opt}}$
$h_t > \ell(1+\lambda)$	$f^*_{\text{res}} = F + (1-F)f_1(h_t)$ $\mathcal{L} = 0$ Lender may liquidate the loan themselves	$f^*_{\text{res}} = L\sqrt{p_t}\left(\sqrt{1+\lambda}-1\right)/V_t(1+\lambda)$ $\mathcal{L} = 0$ Health factor is guaranteed to be larger than pre-liquidation.
$h_t < \ell(1+\lambda)$	$f^*_{\text{res}} = h_t/(\ell(1+\lambda))$ $\mathcal{L} = V_t - \frac{N_t p_t}{(1+\lambda)}$ Collateral is exhausted.	$f^*_{\text{res}} = L\sqrt{p_t}\left(\sqrt{1+\lambda}-1\right)/V_t(1+\lambda)$ $\mathcal{L} = V_t - N_t p_t + \frac{\lambda u_t\left(\sqrt{1+\lambda}-1\right)}{1+\lambda}$ Liquidity not enough for further profitable liquidations.

Table 1.1: Optimal resultant fractions of liquidations in different levels of liquidity and loan health. Whether there is enough liquidity or not is determined by evaluating the corresponding expressions of f^*_{res}, and checking which is smaller. Lender loss is denoted by $\mathcal{L}$.

of market frictions, the optimal liquidation strategy in this case is to liquidate just enough to bring the health factor to 1, and then following it up with a close-factor liquidation. When $h > \ell(1+\lambda)$, this is the maximal fraction of liquidation $f^{(1)}_{\max}(h)$. Let $f_1(h)$ denote the fraction of liquidation such that $\tilde{H}(h, f_1(h)) = 1$. Then, we get $f^{(1)}_{\max} = f_{\text{res}}(h) = f_1(h) + (1 - f_1(h))F = F + (1-F)f_1(h)$.

Now, such a liquidation is profitable to the liquidator if the price slippage in the AMM is not too large. Recall that upon liquidating a fraction f, the liquidator gets an amount of $\frac{fV_t(1+\lambda)}{p_t}$ in collateral. When traded for the currency $\mathcal{B}$, the liquidator gets an amount $x(f)$ such that

$$(u_t - x(f))\left(b_t + \frac{fV_t(1+\lambda)}{u_t/b_t}\right) = u_t b_t \implies x(f) = \frac{u_t f V_t(1+\lambda)}{u_t + fV(1+\lambda)}.$$

Here, we are assuming that the instantaneous price $\hat{p}_t$ as per the liquidity pool matches the oracle price p_t at the time of liquidation. The profit to the liquidator is then given by $\pi(f) = x(f) - fV_t$. Assuming no upper bound on f, this function is maximized when

$$f = f_{\text{opt}} = \frac{u_t\left(\sqrt{1+\lambda}-1\right)}{V_t(1+\lambda)} = \frac{L\sqrt{p_t}\left(\sqrt{1+\lambda}-1\right)}{V_t(1+\lambda)}.$$

Furthermore, $\pi(f)$ is increasing on $[0, f_{\text{opt}})$, and decreasing on $(f_{\text{opt}}, \infty]$. Hence, when $h_t >$

$\ell(1+\lambda)$, we have $f^*_{\text{res}} = \min\left(f_{\text{opt}}, f^{(1)}_{\text{max}}\right)$. In either case, the health factor improves post liquidation, and thus the borrower continues to have an incentive to pay back the loan. The lender, therefore, does not lose any money in this scenario. Since liquidations improve the health factor of the loan and prevent lenders from incurring any loss, liquidations are beneficial in this scenario as they mitigate lender loss.

If $h_t < \ell(1+\lambda)$, then each liquidation reduces the health factor, until eventually, the entire collateral gets liquidated. This is equivalent to a resultant liquidation equal to $f^{(2)}_{\text{max}} = \frac{h_t}{\ell(1+\lambda)}$, since with this fraction of liquidation, the collateral gained by the liquidator equals N_t. The optimal feasible resultant fraction of liquidation then equals $f^*_{\text{res}} = \min\left(f^{(2)}_{\text{max}}, f_{\text{opt}}\right)$. In the case when $f^*_{\text{res}} = f^{(2)}_{\text{max}}$, the entire collateral is consumed by the liquidator while some debt still remains. The loss incurred by the lender is given by $V_t - \frac{V_t h_t}{\ell(1+\lambda)}$. On the other hand, when $f^*_{\text{res}} = f_{\text{opt}}$, there is still some collateral left in the loan, but the liquidity in the pool is exhausted so that liquidation is no longer profitable. Pricing the remaining collateral at price p_t, the lender loss is given by $\mathcal{L} = V_t - N_t p_t + \frac{\lambda L\sqrt{p_t}\left(\sqrt{1+\lambda}-1\right)}{1+\lambda}$. Table 1.1 summarizes these results.

Alternative Liquidation Spread. We note that when $h_t < \ell(1+\lambda)$, the lender loss is more than the lender loss if there was no liquidation, and the collateral was priced at price $p_t = u_t/b_t$. This is because each liquidation grants the liquidator the liquidation spread, which in this regime results in unbearable costs for the loan. A possible improvement in the lending platform design would then be to disallow any liquidations if $h_t \leq \ell(1+\lambda)$, and allow the lender to possess the collateral so that it can be liquidated. Another alternative is to modify liquidation incentives, and make the liquidation spread a function of the health factor $\tilde{\lambda}(h)$ so that

$$\tilde{\lambda}(h) = \begin{cases} \lambda, & h > \ell(1+\lambda), \\ \frac{h}{\ell} - 1, & \ell \leq h \leq \ell(1+\lambda), \\ 0, & \text{otherwise.} \end{cases}$$

This protocol ensures that the liquidation spread paid to the liquidator does not exceed what the

loan can afford at any given health factor. The lender loss is then not exacerbated by liquidations, though lender loss is still possible. In particular, when $\ell \leq h_t \leq \ell(1+\lambda)$, liquidation does not change the health factor and the lender can recover the loan after infinite liquidations. On the other hand, when $h_t < \ell$, the liquidation incentive is 0, and no third-party liquidator would try to liquidate the loan. However, the lender still has an incentive to perform the liquidation to mitigate their losses, and these losses are less than when the liquidation spread was a positive quantity. Incidentally, the lack of liquidator interest in this scenario would also reduce the required gas fees for the liquidations and allow a less costly liquidation for the lender.

When the assets involved are highly liquid (such as ETH/USDC), the vast majority of liquidations take place when health factor is very close to 1, and thus are at or near close-factor (note that $f_1(1) = 0$). In the subsequent sections, we assume that all liquidations take place at the close-factor.

1.4 Cost of Liquidation

To compute the cost of liquidation, we first assume that the price p_t evolves as per the Black-Scholes model. The price process evolves stochastically and has continuous sample paths. Specifically, consider a filtered probability space $\left(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{Q}\right)$ satisfying the usual conditions. We assume the price of the collateral, p_t, to be an Itô process on this space given by a geometric Brownian motion, i.e., $dp_t = \mu p_t\, dt + \sigma p_t\, dB_t$, where B_t is the standard Brownian motion, μ is the expected rate of return and σ is the return volatility. We also assume that $\mathbb{Q}$ is a risk-neutral measure. In other words, $\mu = r$, the risk-free rate.

We find it convenient to work with the health factor h_t, instead of working with the price. In particular, applying Itô's lemma, we can see that $\log h_t$ satisfies the stochastic differential equation $d\log h_t = \tilde{\mu}\, dt + \sigma\, dB_t$, where $\tilde{\mu} := r + \gamma_C - \gamma_V - \sigma^2/2$. In other words, $h_t = h_0 \exp \sigma B_t + \tilde{\mu} t$. Note that we assume that the liquidator can buy or sell arbitrary quantities of the asset without moving the price. Now, let $\tau_1 := \inf\{t > 0 : h_t \leq 1\}$ denote the stopping time when the loan becomes available for liquidation. As discussed previously, because of sample path continuity,

we have $h_{\tau_1} = 1$. At time τ_1, the liquidator gains collateral worth $\lambda V_{\tau_1} F$ by performing a close-factor liquidation. The loan then continues with debt amount $V_{\tau_1}(1-F)$ and health factor $\tilde{H}(1,F)$. This new loan is subject to the same liquidation risk and we can thus use a recursive formulation to compute the total cost of liquidation of the original loan. To make the calculations tractable, we shall assume that the monitoring horizon is exponentially distributed with rate ν. Due to the memoryless property of the exponential distribution, the cost of further liquidations is independent of the time passed. We can thus write the expected present value of the cost of liquidation as follows:

$$p_q(V,h) = \mathbb{E}_{\mathbb{Q}}\left[e^{-r\tau_1}\left(\lambda V e^{\gamma_V \tau_1} F + p_q\left(e^{\gamma_V \tau_1} V(1-F), \tilde{H}(1,F)\right)\right)\mathbb{1}\left(\tau_1 \leq T\right) \,\middle|\, h_0 = h\right], \quad (1.1)$$

where, $\tilde{H}(1,F)$ is the health factor immediately after liquidation, given by equation (A.1). Before we can compute the above, we need some assumptions on the rate ν. In particular, we observe that, if there is no liquidation, the expected value of the debt amount after the monitoring horizon is given by

$$\mathbb{E}\left[e^{\gamma_V T} V\right] = \begin{cases} \frac{\nu}{\nu - \gamma_V}, & \nu - \gamma_V > 0 \\ \infty, & \nu - \gamma_V \leq 0. \end{cases}$$

To avoid the debt amount (similarly the collateral) diverging to infinity, we shall make the following assumptions about the monitoring frequency ν.

Assumption 1.2 $\nu > \gamma_V, \nu > \gamma_C$.

In practice (cf. Table 1.2), we find that extreme parameter values are required for Assumption 1.2 to be violated. Consider the following example.

Example 1.2 *Let $\gamma_V = 30.0\%, \gamma_C = 25.0\%$ be the annualized interest rates. Then we need $\nu < 0.3$ for Assumption 1.2 to be violated. This is equivalent to the average monitoring horizon being over 3 years.*

Note that the interest rates γ_V, γ_C in Example 1.2 are unusually high, making the divergence pos-

Table 1.2: Values of contract parameters in leading lending platforms on Ethereum Mainnet assuming a loan of USDC against collateral of WETH. All rates are annualized, and all values are current as of Nov 30, 2023. Compound III does not pay any interest on the deposit being used as collateral.

Parameter	Aave V2	Aave V3	Compound III	JustLend
Liquidation threshold (ℓ)	86%	83%	93%	75%
Liquidation spread (λ)	4.5%	5%	3%	8%
Close factor (F)	50%	50%	100%	50%
Deposit rate (γ_C)	1.03%	1.84%	0.0%	0.72%
Borrow rate (γ_V)	5.06%	13.58%	2.86%	3.80%

sible, and even then only when the monitoring horizon is very large in expectation as well. We can now prove the following theorem.

Theorem 1.4.1 *If monitoring horizon T is distributed exponentially with parameter ν that satisfies Assumption 1.2, then the borrower's expected discounted liquidation cost up to time T is given by*

$$p_q(V,h) = \lambda V F h^{-\kappa}\left(1 - \frac{1-F}{\tilde{H}(1,F)^{\kappa}}\right)^{-1},$$

where,

$$\kappa := \frac{\sqrt{\tilde{\mu}^2 + 2\sigma^2(\nu + r - \gamma_V)} + \tilde{\mu}}{\sigma^2}.$$

For convenience, we define $\pi(h) := p_q(V,h)/V$, to be the normalized expected discounted liquidation cost. Table 1.2 lists the values of contract parameters in various popular lending pools. As mentioned earlier, we observe that $r > \gamma_V > \gamma_C$ in all lending pools, except for Compound III. Thus, Assumption 1.2 is satisfied for any $\nu > 0$ in these pools. Example 1.2 shows that ν needs to be unrealistically small for it to be violated with Compound III parameters. We discuss the cost function in further detail in Section 1.5.1. We now consider a control problem where the borrower decides on an initial health factor for the loan. It costs the borrower αh to get a loan with health factor h per unit debt, for some $\alpha > 0$. Here, α determines the financing cost the borrower faces. The borrower aims to minimize the total cost along with the expected payment to the liquidator. In other words, the borrower faces the optimization problem of minimizing $C(h) := \alpha h + \pi(h)$ for $h \geq 1$. We can prove the following theorem.

Theorem 1.4.2 *Under the assumptions of Theorem 1.4.1, the optimal health factor is*

$$h^* = \max\left\{1, \left(\kappa\lambda F\alpha^{-1}\middle/\left(1 - \frac{1-F}{\tilde{H}(1,F)^\kappa}\right)\right)^{\frac{1}{1+\kappa}}\right\}. \tag{1.2}$$

We discuss the optimal health factor in further detail in Section 1.5.1.

1.5 Empirical Analysis

1.5.1 Comparative Statics

Cost Function.

First, we consider the liquidation cost function $\pi(h) = p_q(V,h)/V$. It is instructive to write $\pi(h)$ as an infinite sum.

$$\pi(h) = \frac{\lambda F}{h^\kappa}\left(1 + \frac{1-F}{\tilde{H}(1,F)^\kappa} + \frac{(1-F)^2}{\tilde{H}(1,F)^{2\kappa}} + \dots\right). \tag{1.3}$$

Here, nth term in the sum can be interpreted as the expected cost from the nth liquidation. We can see that the cost decreases as κ increases. We would like to see how the cost changes with the horizon rate ν, and the volatility σ. Clearly, κ increases with ν. After some algebra, we can also see that, as long as $\nu + r - \gamma_V > 0$ as guaranteed by Assumption 1.2, κ decreases with σ. We now fix the contract parameters to their real world values. We use AAVE V2 lending pool to get our contract parameters because as of this writing, it is the lending pool with the highest TVL. For the risk-free rate, we use the 1 month yield curve rate published by US Treasury, which as of this writing is 5.56%. Fixing these parameters, we plot the normalized liquidation cost function, $\pi(h)$ against h for different values of σ and $\mathbb{E}[T] = 1/\nu$ in Figure 1.3. Here, both T and σ are in daily units. The figure shows that $\pi(h) = p_q(V,h)/V$ is a decreasing function of h, which is to be expected since it increases the likelihood of the first liquidation. We see that as volatility increases, the risk of liquidation increases, and hence so does the liquidation cost when everything else is held fixed. We also see that the cost decreases as expected horizon length decreases, and the borrower pays back the loan in a shorter length of time in expectation. Furthermore, as the

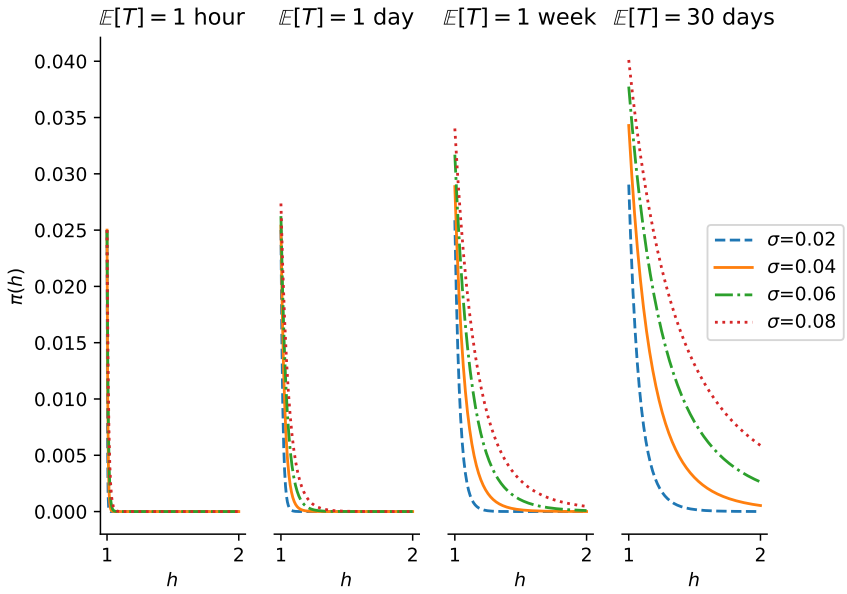


Figure 1.3: Normalized expected discounted cost of liquidation against initial health factor for different values of volatility σ (daily) and expected horizon length.

expected horizon length decreases (ν increases), the curves for different σ come closer indicating that the effect of volatility on the expected cost of liquidation decreases as the horizon becomes shorter in expectation.

Optimal Health Factor. Now, we would also like to know how h^* changes with $\mathbb{E}[T]$, σ, and other variables of interest. To that end, we fix all parameters from Table 1.2 and vary the expected length of monitoring horizon, the liquidation spread, and the financing rate in Figures 1.4, 1.5a, and 1.5b respectively for different values of σ. For all figures, the default financing rate is assumed to be the same as the risk-free rate, and the default expected monitoring horizon is assumed to be 1 day. Based on the financing rate, we set

$$\alpha = \frac{r'}{\ell(\nu - r')},$$

so that αh represents the expected cost of capital borne by the borrower in the monitoring horizon. Here r' is the financing rate. This is because getting a loan of V_0 amount and obtaining a health factor h requires one to deposit total collateral worth $N_0 b_0 = hV_0/\ell$. The cost of financing this

amount of collateral with rate r' is given by,

$$\mathbb{E}\left[e^{r'T}\frac{hV_0}{\ell} - \frac{hV_0}{\ell}\right] = \frac{r'hV_0}{\ell(\nu - r')}.$$

Note that we assume $r' < \nu$ since otherwise, the cost of capital would diverge to ∞.

We see from Figure 1.4 that the borrower needs to start with a larger initial health factor if the asset is more volatile or if the monitoring horizon is longer. This is because in both cases the loan is more likely to get liquidated. We also observe that the rate of increase in the optimal health factor is larger when the asset has a larger volatility.

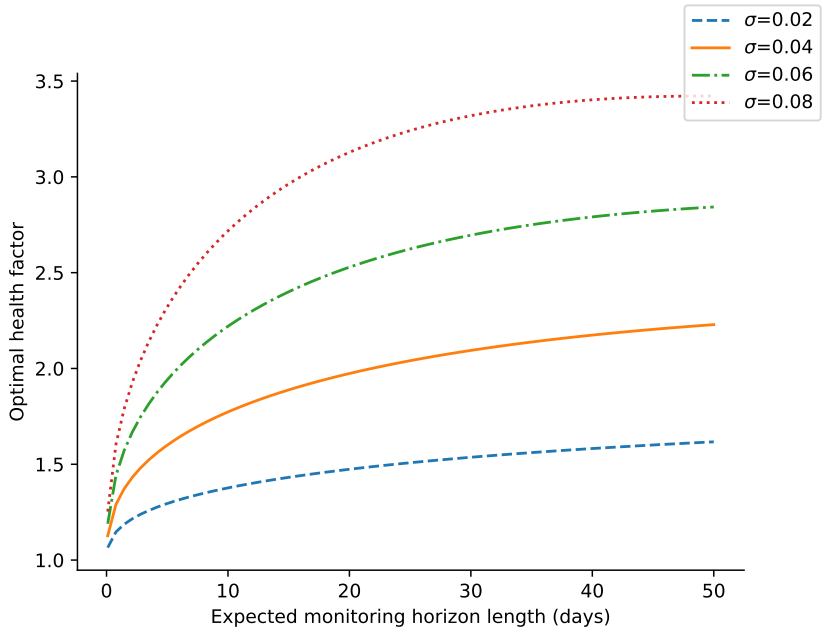
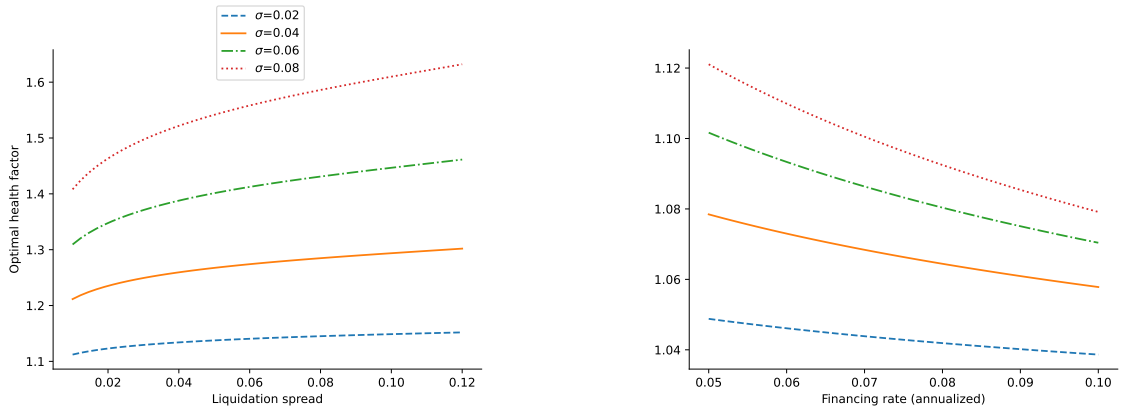


Figure 1.4: h^* vs $\mathbb{E}[T]$

Figure 1.5a shows that the optimal health factor increases with the liquidation spread, and that this increase is steeper for a more volatile asset. This is because the borrower loses more in each liquidation when the spread is large, and the probability of such a liquidation increases with σ.

Figure 1.5b shows that the optimal health factor decreases with the financing rate since it costs more to obtain the necessary capital to get a loan with any given health factor. This decline is also steeper for riskier assets. This is because the optimal health factor of a riskier asset is larger than



(a) h^* vs liquidation spread

(b) h^* vs financing rate

Figure 1.5: h^* vs liquidation spread and financing rate.

that of a less risky asset to begin with, and so the incremental cost due to rising financing rate is higher for a risky asset held in a loan at an optimal health factor.

1.5.2 Data

We get lending data from AAVE V2 lending pool on Ethereum Mainnet [24], risk free interest rates from US treasury daily par yield curve rates for 1 month maturity [25], and minutely prices from Binance Public Data [26]. All data spans from March 15, 2021 through Jan 31, 2024.

From the lending pool, we fetch all deposit, borrow, repay and withdraw events, and filter them so that only users that have borrowed USDC against a collateral of WETH remain. For each event, we ensure that this condition is satisfied for two consecutive blocks: the block in which the event occurred (current block), and the immediately preceding block (prior block). For these two blocks, we get user health factors. We also get user's debt, collateral amount, borrow rates, and the pool's liquidation threshold and liquidation spread for the current block. We further filter this data to remove rows with a debt less than 100 USDC, or a health factor less than 1. We call this the `health-factors` table. This table has 28,876 rows.

For all the users in the `health-factors` table, we separately fetch all their interactions with the pool. These may include setting or unsetting one of their deposits as collateral for their loan, borrowing or depositing currencies other than USDC or WETH, etc. We use this data to figure out

the lifetime of a loan and the average time elapsed between user interactions with the lending pool when a particular loan was active. Lifetime of a loan is defined to be from the first block with a non-zero debt to the last block with a non-zero debt. We assign each distinct loan a unique loan id. We call this the `user-interactions` table. We then join this table with the health factors table by figuring out the loan id from the block number of the current block, so that each row in the health factor table has the loan id, as well as the average intervention time.

From Binance, we get the minutely candles of WETH-USDC prices. Unfortunately, these prices are not available for the entire period as of this writing. Notably, they are missing for the period between September 29, 2022 through March 12, 2023. We use WETH-USDT prices as a proxy for this period. We verify that for the period where both prices are available, the difference between the two is always less than 10%, and is less than 0.2% in 95% of the times. We resample this data to a 15 minute frequency and compute daily volatility for each calendar day, and use it to compute expected liquidation costs and the optimal health factors on that day.

From US Treasury, we get the daily par yield curve rates for 1 month maturity. We use this rate as the risk free rate on that day. These rates are not published on weekends. We use the most recent available rates as proxy for the rates on days on which they are unavailable.

We also get liquidation data from the lending pool. This consists of all loans that covered a USDC debt by grabbing a WETH collateral. For each liquidation, we get the amount of debt liquidated, and the block number. This liquidation table has 1,213 rows. We use this table to compute the liquidation costs between user interactions in the following way: for each user interaction in the user-interaction table, we discount and add up the liquidated amount until the next interaction on that same loan. For discounting, we use the risk-free rate at the interaction time. For interactions that are common between the user-interaction and health-factor tables, we get the liquidated debt. We multiply this by the liquidation spread and divide by the debt in the current block to get the empirical liquidation cost function $\hat{\pi}(h)$.

To summarize, we have the following tables:

`health-factors` : Containing health factors immediately before and after a deposit, withdraw,

borrow, or repay event. Also contains borrow and deposit interest rates, debt and collateral amounts, liquidation cost and loan id. This table has 28,876 rows.

`prices` : Contains WETH-USDC prices at 15 minute intervals. Also contains daily return volatility. This table has 1,620,367 rows.

`liquidations` : For each liquidation event, contains the amount of debt repaid. This table has 1,213 rows.

`user-interactions` : Each lending pool transaction for each user in health-factors table. This table has 155,252 rows.

`interest-rates` : daily risk-free rates. This table has 723 rows.

`firsts` : Rows from the `health-factors` table where the debt changed from 0 in the previous block to > 0 in the current block. After removing the rows with health factors larger than the top 0.5% percentile, this table has 5,285 rows.

1.5.3 Why do Liquidations Occur?

A rational borrower who observes that their loan is close to being liquidated would never allow such a liquidation. They could instead get a flashloan, repay part of their debt, free up and sell some collateral, and repay the flash-loan. Since we observe liquidations, one of two things must be happening:

1. The price process has downward "jumps", leading to liquidations.
2. Users do not monitor their loans at all times.

In this section, we validate our hypothesis that downward price jumps do not cause a majority of liquidations. To do this, we construct a hypothesis test to detect jumps in the price process. First, we resample our price process to a period Δt. Let the price at hour t be p_t. We have for each t:

H0: There is no jump from $t - \Delta t$ to t.

H1: There is a jump from $t - \Delta t$ to t.

To conduct this test, we need an estimate of the volatility at time $t - \Delta t$ that is robust to jumps, and we use returns from n preceding periods to compute it. This methodology is described by [27]. Specifically, we compute the returns $r_t = \log p_t / p_{t-1}$, and the realized bipower variation in the past twenty periods. We obtain a square root and scale it by $\sqrt{\pi/2}$ to get an estimate of volatility $\sigma_{t-1}^{(j)}$ that is robust to jumps. Formally,

$$\sigma_{t-1}^{(j)} = \sqrt{\frac{\pi}{18 \cdot 2} \sum_{i=1}^{18} |r_{t-i} r_{t-i-1}|}$$

We then compute the t-statistic $s_t = r_t / \sigma_{t-1}^{(j)}$. If $|s_t| > 1.96$, we refute the null hypothesis at a 95% confidence level. If $s_t > 1.96$, we say that there was an upward jump, and if $s_t < -1.96$, we say that there was a downward jump at time t.

Thus, we identify the specific hours in which downward jumps occur. For each liquidation in the `liquidations` table, we check whether it occurred in an hour with a downward jump. That is, if the liquidation happened at time $t^{(l)}$, we find i such that $t_i < t^{(l)} \leq t_{i+1}$, and check if there was a downward jump at hour t_{i+1}. Table 1.3 summarizes the results that show a vast majority of liquidations occurring in time periods without any downward jumps.

Δt	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$
30 mins	0.81	0.81	0.82	0.81	0.79
1 hour	0.78	0.77	0.79	0.79	0.79
2 hours	0.77	0.76	0.77	0.76	0.79
12 hours	0.69	0.71	0.71	0.70	0.70
1 day	0.72	0.73	0.75	0.75	0.75

Table 1.3: The fraction of liquidations occurring in periods without downward jumps computed for different period lengths, and different window sizes.

1.5.4 Empirical Liquidation Cost

Now, we compute the empirical liquidation cost and compare it with our closed form expression. To get the expected liquidation cost for a loan k at time t using our closed form expression, we need the following parameters: $h = h^{(k,t)}, F = F^{(t)}, \ell = \ell^{(t)}, \lambda = \lambda^{(t)}, \gamma_C = \gamma_C^{(k,t)}, \gamma_V = \gamma_V^{(k,t)}, r = r^{(t)}, \sigma = \sigma^{(t)}, \nu = \nu^{(k)}$. All of these except for $\nu^{(k)}$ can be found from the `health-factors` and the `prices` tables. Unfortunately, there is no way for us to determine the average monitoring frequency ν for a user. Instead, we use the average interaction frequency $\hat{\nu}$, which is a lower bound on ν. Assuming that users monitor more frequently than $\hat{\nu}$, we expect to overestimate the liquidation costs when we use the closed form to compute them.

To compute the empirical liquidation costs, we divide the `health-factors` table into twenty bins, so that each bin contains an equal number of rows. We then compute an average liquidation cost, as well as the average of the expected liquidation cost for each bin. Figure 1.6

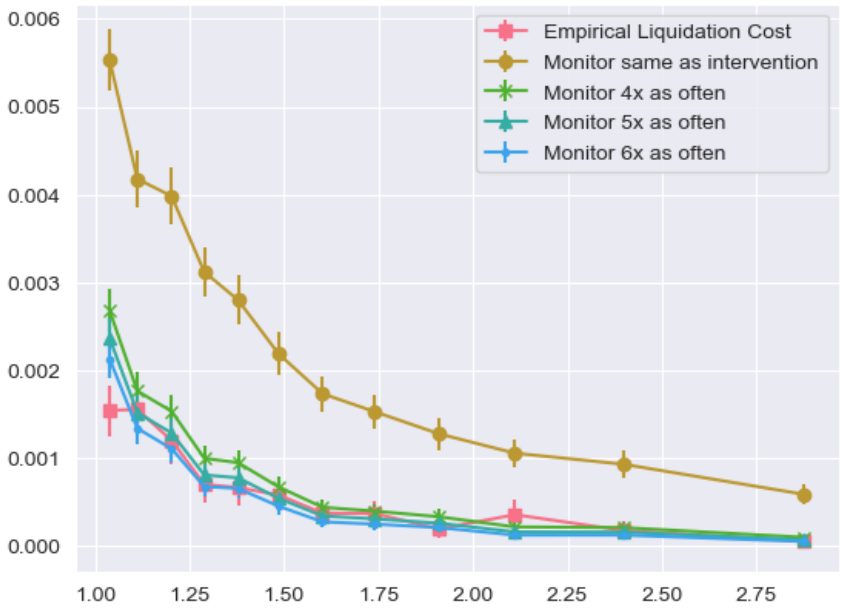


Figure 1.6: The empirical and expected liquidation cost versus health factor. The error bars represent the 95% confidence intervals.

shows the plot of these two costs against the centers of the health factor bins. As we can see, the liquidation cost is indeed overestimated due to our usage of $\hat{\nu}$ instead of ν. However, if we assume

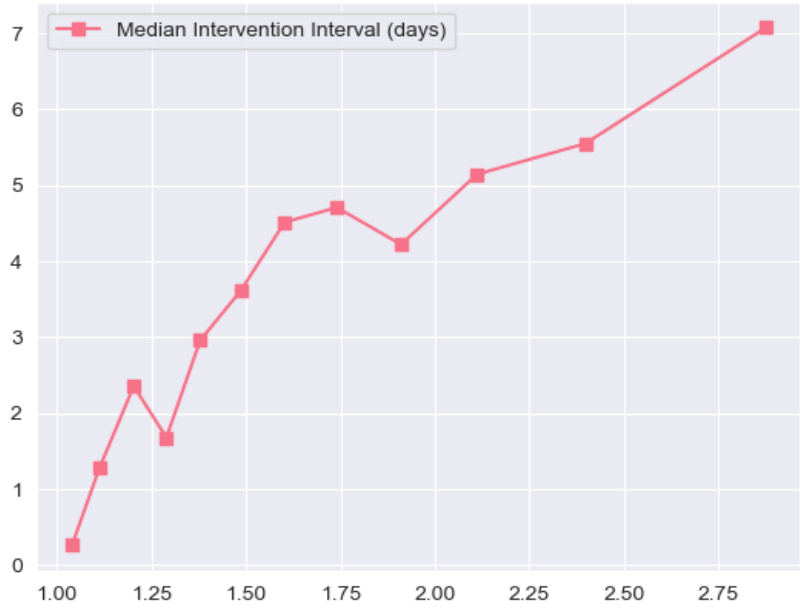


Figure 1.7: Medians of intervention intervals in each health factor bin.

that the monitoring frequency is five or six times the intervention frequency, then the two costs match in most bins. Note that to compute this, we discarded the few ($< 5\%$) cases where the liquidation took place at a health factor less than one. We also discarded the last two health factor bins since they did not have any liquidations. We note a mismatch in the very first bin, and Figure 1.7, which shows the medians of intervention intervals for the same health factor bins, explains the reason: The intervention interval in the first bin is small (median=0.25 days), indicating a monitoring frequency of 20–24 times a day. The fact that oracle updates are hourly can not be ignored when computing the liquidation cost for this bin, and this is the reason our model overestimates the cost.

1.5.5 Observed Health Factors

We now turn to our empirical findings. As before, we use the intervention frequency $\hat{\nu}$ in lieu of the monitoring frequency ν. We therefore expect to overestimate the optimal health factors. Let our estimates be $\hat{h}^*$. Another parameter we need is the financing rate. We use the borrow rate for the collateral (WETH) as the financing rate r'. We are interested in knowing how $\hat{h}^*$ compare with those held by the users. For the user health factors ($\hat{h}$), we only consider the times when a new

loan was opened, that is, we only consider cases where debt amount of the user went from 0 in the previous block to >0 in the current block. After removing the rows with health factors in the top 0.5% percentile as outliers, this table `firsts` consists of 5,285 rows.

To compare the observed health factors with the optimal ones, we use a similar strategy as we used for the liquidation costs. First, we compute $\hat{h}_k^*$ using equation (1.2), assuming the monitoring frequency to be k times the intervention frequency of the borrower. Then, We divide the `firsts` table into 10 bins of equal sizes based on the prevailing volatility on the day the loan was opened. We then compute the medians of $\hat{h}$ and $\hat{h}_k^*$.

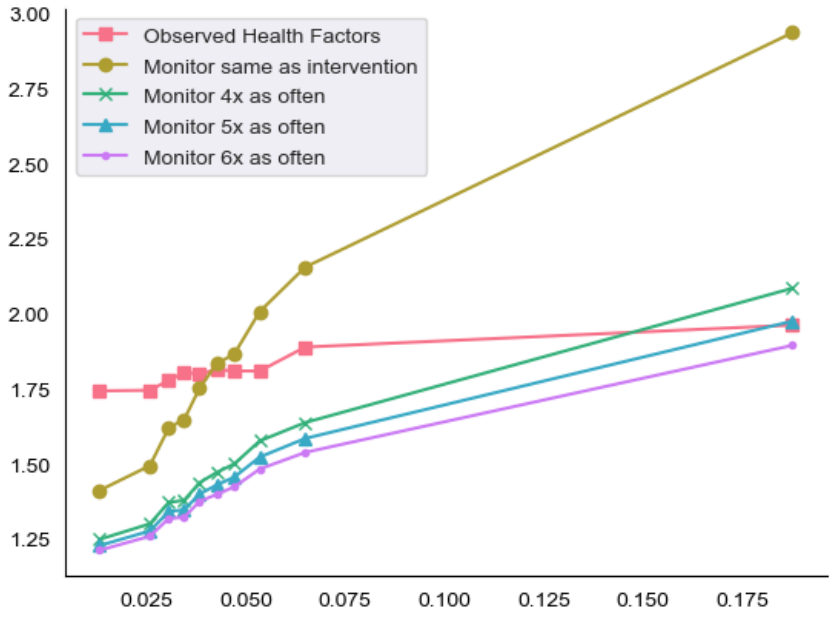


Figure 1.8: Medians of Empirical and Optimal health factors versus volatility

Figure 1.8 shows the graph of the medians of $\hat{h}$ and those of $\hat{h}_k^*$ against the centers of the bins of daily volatility. We observe that, empirically, the health factors are slightly larger than what we predict, especially when the volatility is small, and that the difference becomes smaller as the volatility increases. The plot indicates that borrowers are prepared for a higher volatility than observed.

We formalize our observation by computing an "implied volatility" from the empirical health factors by writing equation (1.2) as a function of σ and then inverting it. Assuming a monitoring

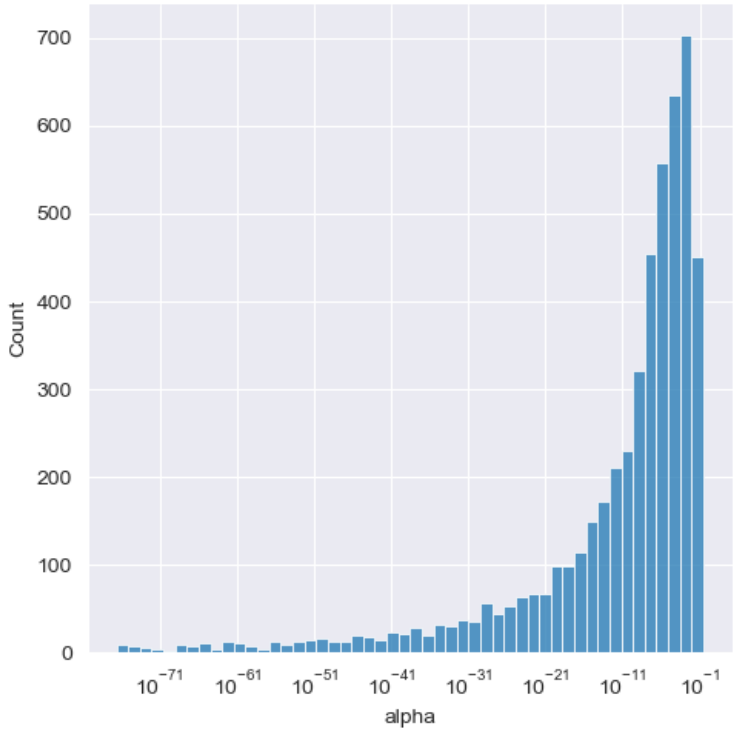


Figure 1.9: Histogram of implied α, the weight given to the cost of capital.

frequency that is four, five, or six times that of the intervention frequency, we compute the implied volatilities $\hat{\sigma}_4^*$, $\hat{\sigma}_5^*$, and $\hat{\sigma}_6^*$ respectively. Starting with the `firsts` table, we first remove all entries where $\hat{\sigma}_k^*$ was undefined or the daily volatility was in the top 0.5% percentile. Formally, we

	$k = 1$	$k = 4$	$k = 5$	$k = 6$
β_k	1.816^{***}	2.904^{***}	3.100^{***}	3.262^{***}
	(0.109)	(0.146)	(0.152)	(0.157)
Count	2,211	3,012	3,171	3,320

Table 1.4: Results of regression of implied volatility against observed volatility for different values of the ratio of intervention frequency and monitoring frequency. Standard errors for each value of β_k are included in brackets. Note: $^{***}p < 0.01$.

estimate the model $\hat{\sigma}_k^* = \beta_k \hat{\sigma} + \epsilon$ for $k \in \{1, 4, 5, 6\}$. Table 1.4 summarizes the results of these regressions. The results indicate that borrowers are prepared for a volatility that is, on average, three times more than the observed daily volatility. Additionally, from the t-statistics, and the positivity of the coefficients, we can infer a positive correlation between the implied volatility and the observed volatility. Indeed, we can explicitly compute the correlations ρ_k between $\hat{\sigma}$ and $\hat{\sigma}_k^*$ to

observe the results in Table 1.5 We note here that the implied volatilities have standard deviations

	$k = 1$	$k = 4$	$k = 5$	$k = 6$
ρ_k	0.109^{***}	0.088^{***}	0.085^{***}	0.083^{***}
	(0.021)	(0.018)	(0.018)	(0.017)
Count	2,211	3,012	3,171	3,320

Table 1.5: Correlations between the observed volatility and the implied volatility for different values of the ratio of intervention frequency and monitoring frequency. Standard errors for each value of the correlation are included in brackets. Note: $^{***}p < 0.01$.

that are an order of magnitude larger than the standard deviation of the daily volatility. This is why the correlations are small, though statistically significant.

In a similar fashion, we also compute the implied α from observed health factors assuming a monitoring frequency that is five times the intervention frequency. Figure 1.9 shows the histogram of this implied α in log scale after removing the bottom 5% percentiles. The values indicate that the borrowers face a very small opportunity costs compared to liquidation costs.

1.6 Price Manipulation

In this section, we consider the case of *predatory borrowing*, where an agent manipulates the price of an asset in the spot market so as to inflate the value, then borrows against it at the inflated value. The agent then walks away with the borrowed amount. First, we need to model price impact. For this, we use the very general model introduced by [28]. Let $B(p, q)$ denote the expected price of trade when quantity $q > 0$ is bought in the market at the starting price p. After the trade, let the expected final price be denoted by $U(p, q) \geq p$. Now consider the following sequence of events: (1) a borrower buys quantity q of an asset at starting price p, and moves it up; (2) then, the borrower uses the asset as collateral and borrows against it. To maximize their profits, they need to find $q > 0$ that maximizes $qU(p, q)\ell - qB(p, q)$. In this scenario, we can prove the following theorem.

Theorem 1.6.1 *The above strategy is profitable in expectation to the borrower for some q if and only if $\ell > \min_q B(p, q)/U(p, q)$.*

Proof: We can rewrite the profit of the borrower as

$$qU(p,q)\left(\ell - \frac{B(p,q)}{U(p,q)}\right).$$

This is positive for some q if and only if

$$\ell > \min_q \frac{B(p,q)}{U(p,q)}.$$

□

Theorem 1.6.1 does not assume anything about the buying process, but only about the price evolution as a function of the quantity being bought. In fact, we can microfound the functions B and U by assuming that the quantity q is traded over a finite interval $[0,t]$ and that the price process follows a price impact function. As an example, consider the model proposed by [29] (the Gatheral model). This is the simplest model in that the price process follows a random walk and its drift is influenced by the trades. We need only consider the expected price process. According to this model, if the price at time 0 is p_0, and the asset is bought at a determinstic rate ν_t, then the price evolves as per the following equation:

$$\mathbb{E}\left[p_t\right] = p_0 + \int_0^t f(\nu_s)G(t-s)\,ds \tag{1.4}$$

where $f(\nu_t)$ represents the immediate impact of trading at rate ν_t and the transience function $G(\tau)$ represents decay of that price impact after time τ.

Corollary 1.6.1 *Under the Gatheral model, if $f : \mathbb{R} \to \mathbb{R}$ is an increasing continuous surjective function, and $G : \mathbb{R}^+ \to \mathbb{R}^+$ is a decreasing continuous function such that $G(0) = 1$, then a predatory borrowing attack with a constant purchase rate is profitable in expectation for some purchase rate if*

$$\ell > \int_0^T tG(T-t)\,dt \Big/ \left(T\int_0^T G(T-t)\,dt\right).$$

Proof: The last term in equation (1.4) signifies the noise due to external market participants. For this discussion, we only consider the effect of price impact in expectation, and so ignore this term. Let the constant rate of trading by the borrower be ν. In this scenario, we have $q = \nu T$, the average trading price is given by

$$B(p_0, q) = \frac{\int_0^T \nu p_t dt}{\nu T} = \frac{p_0 T + f(\nu) \int_0^T \int_0^t G(t-s) ds dt}{T} = p_0 + f(\nu) \int_0^T \frac{(T-t)G(t)}{T} dt,$$

and the final price is given by,

$$U(p_0, q) = p_T = p_0 + f(\nu) \int_0^T G(T-t) dt.$$

Finally, we have,

$$\begin{aligned} \min_{q=\nu T} \frac{B(p_0, q)}{U(p_0, q)} &= \min_{\nu} \frac{p_0 + f(\nu) \int_0^T \frac{(T-t)G(t)}{T} dt}{p_0 + f(\nu) \int_0^T G(T-t) dt} \\ &= \frac{\int_0^T tG(T-t) dt}{T \int_0^T G(T-t) dt}. \end{aligned}$$

Thus, the Gatheral model allows for the predatory borrowing attack if

$$\ell > \frac{\int_0^T tG(T-t) dt}{T \int_0^T G(T-t) dt}.$$

We note that the existence of a profitable rate of purchase does not depend on the price impact function, but only on the decay function. How large the rate should be for profitability does, however, depend on the price impact function. □

We now consider specific examples of the impact and transience functions.

Example 1.3 *The canonical functional form of the transience function $G(\tau)$ as proposed by [29] is $G(\tau) = \tau^{-\gamma}$. In this case we need $\ell > 1/(2-\gamma)$ for the existence of a constant purchase rate that would be profitable to the borrower. Note that such a purchase rate can be high and might*

require access to a large amount of capital. This result can be used by lending pool designers to decide contract parameters. For example, if the price impact is known to follow the Gatheral model with $G(\tau) = \tau^{-0.4}$*, as suggested by [29], then the designers should ensure that* $\ell < 0.625$ *to disallow such an attack.*

Example 1.4 *Another interesting possibility is that* $G(\tau) = e^{-\rho\tau}$ *for some* $\rho > 0$ *as suggested by [30]. In this scenario, we need* $\ell > 1 + \frac{1}{e^{\rho T}-1} - \frac{1}{\rho T}$*. We can also get a similar, but more complicated result when* G *is exponential. Picking a suitable* T *is necessary in this case. We note that the borrower needs to be able to sustain trading at a constant rate for the entire time horizon* T*, and so the designer need to not worry about* T *being arbitrarily large. For example, if* ρ *is estimated to be* 0.5 *per day, and we set* $T = 1$ *day, then* $\ell < 0.541$ *disallows the predatory borrowing attack. With the same* ρ*,* $T = 0.5$ *days require that* $\ell < 0.521$*.*

Example 1.5 *When* ℓ *satisfies the condition in Theorem 1.6.1, we can use our model to compute the quantity needed for such an attack. For example, let* $\ell = 0.93$ *as seen in Compound III. Suppose that the price impact function follows the square-root law, which is widely used in practice (see [31]). For simplicity, suppose that buying is instantaneous and that the price impact is permanent (*$G(\tau) = 1$*). To compute the transaction cost, we use the square-root model as calibrated by [31]. In this model, we have*

$$B(p, q) = p + 1.208 \cdot 10^{-3} p\sigma\sqrt{q/V},$$

where σ *is the daily returns volatility and* V *is the daily traded volume of the asset. This model corresponds to the price update function*

$$U(p, q) = p + 1.812 \cdot 10^{-3} p\sigma\sqrt{q/V}.$$

Now suppose we have $\ell = 0.93$ *as in Compound III. For attacker profitability, we need to find* q *such that* $\ell U(p, q) > B(p, q)$*. Taking* $p = 2000$ *for a WETH-USDC loan and solving, we find that*

q needs to be larger than $34 \cdot 10^{12}$ *times the daily traded volume for this attack to be profitable to the attacker. Such an attack does not seem feasible in practice. However, this example illustrates the utility of our methodology.*

1.7 Conclusion

We modeled fixed-spread liquidation lending platforms and computed the cost borne by a passive borrower as a result of the liquidations in closed form. We assumed an exponentially distributed monitoring interval for ease of computation and interpretation. While it is also possible to compute the cost if the monitoring interval was deterministic, the resulting expression can only be expressed as a sum of an infinite series, and not as a closed form. Based on the cost function we computed, we came up with a novel optimization problem for the passive borrower to help them decide the optimal health factor they should start with.

Empirically, we observed that over 70% of liquidations occur in time periods without price jumps. We also compute the liquidation costs borne by the borrowers in practice and observe that these match with our observations assuming that the borrowers monitor their loans 5–6 times more often than they interact with them. Furthermore, we compare the health factors maintained by borrowers against the model optimal health factors and observe that borrowers are far more conservative in deciding their health factors. This indicates either a very low opportunity cost, or a readiness towards a possible high volatility regime.

Finally, we consider the effects of price impact of trading on such lending platforms. We consider a price manipulation attack and demonstrate that if a carefully calibrated price impact model is available, contract parameters can be designed in such a way so as to make such an attack unprofitable or impractical.

Chapter 2: Hybrid Scheduling with Mixed-Integer Programming at Columbia Business School[1]

2.1 Introduction

The Covid-19 pandemic ushered in an era of online events. After spending some time in a fully online mode, Columbia Business School (CBS) decided to start a hybrid mode of instruction wherein a set of students are invited to attend in-person classes, while others attend online. The intention was to allow MBA students an opportunity to network and extract maximum benefits from their program. The in-person attendance capacity was limited due to social distancing guidelines. We therefore assign each student a mode of attendance for each of their classes: either in person, or online. Since many students travel to campus from afar, if a student was scheduled to attend one class in person, it was desirable to allow them to attend all their classes scheduled on that day in person as much as possible. Furthermore, each student deserved an equal opportunity to attend classes in person an equal number of times. In this paper, we discuss this hybrid scheduling as it was implemented at Columbia Business School starting Fall 2020 through Summer 2021.

At first glance, there are several ways to implement such a schedule. For example, a customized schedule could be designed for each student that lists the days that student should attend classes in person. This would allow considerable flexibility in terms of many relevant objectives that we care about, but would be hard to implement as each student would need to consult their personalized schedule every day, and planning any group activity would need to take into account the combined schedules of everyone in that group. Our objective is to come up with a solution that is easier to implement and communicate, but could be more restrictive. We consider two variations of this problem for different student populations and address each with a separate scheduling approach to

[1]This chapter is based on original work in collaboration with Ciamac Moallemi[32].

better fulfill the needs of the respective populations.

Second-year students at Columbia can have widely varying schedules. Here, we prioritize simple implementation over other objectives. We divide the students into groups, and cycle the groups for in-person attendance throughout the term. While this means that students from one group never get to interact with those from other groups in person, we are willing to allow this loss of interaction in exchange for ease of implementation. An advantage of this approach is that other in-person events can now be arranged using these same groups. For example, suppose that the students are divided into three groups, and that an organizer is organizing an event for all students to attend in person. Such an organizer can schedule this event for three consecutive days. Then, the students scheduled to attend classes in person on these days can attend this event in person as well.

For the first-year students, who are new to the school, networking is arguably more important than it is for second-year students. First-year students at Columbia are divided into "clusters" with identical schedules. Each cluster is further subdivided into learning teams that are expected to spend a significant amount of their time together working on group assignments, etc. Since each cluster has the same number of learning teams, we can schedule these students by their learning teams, and each cluster can follow the same schedule by learning team. We can then try to maximize networking between learning teams.

We refer to the two formulations above as student-level and team-level scheduling, respectively.

2.1.1 Contributions

For student-level scheduling, as discussed above, our solution involves assigning each student to one of M groups, where M is predetermined, and scheduling the groups one by one to attend classes in person on each day of the term. This ensures that each student attends classes in person for exactly a fraction $1/M$ of the time, which is fair since it is the same for all students. In some cases, even if only one group is assigned to attend the day's classes in person, there may be some

classes where in-person attendance would exceed the social distancing capacity. In such a case, we assign the excess students at random to attend those classes online. We formulate the problem of group assignments as an optimization problem that minimizes the total excess (TE), summed over all groups and courses. Because the excess students are still on campus on their scheduled days, the school needs to provide a separate room from which they can attend the class online. Again, due to social distancing considerations, this room is limited in the number of students it can house simultaneously, making it imperative to minimize the amount by which the simultaneous excess (SE), i.e., the maximum excess summed across all classes at the same time, exceeds this limit. We define the amount of SE over the excess-room capacity as the surplus simultaneous excess (SSE). Finally, we should avoid any imbalance between the number of students attending a class in person across different groups. We define total deviation (TD) as a measure of this imbalance and also minimize this quantity. In summary, we make the following contributions to the student-level scheduling problem:

- We present a novel formulation of the problem as a mixed-integer program.
- We analyze the resulting program and, in particular, present a closed-form solution to the relaxed linear program.
- As a preliminary validation, we use our optimization approach to conduct counterfactual group assignment experiments on data before the Covid-19 pandemic (Fall 2019) and present the results in Section 2.2.3.
- We present empirical results from the application of our method to second-year MBA students at Columbia Business School. In Fall 2020, we divided 1,302 second-year students enrolled in 67 hybrid classes into three groups according to our optimization method to get a TE of 20 and an SE of five. In Spring 2021, we divided 1,035 students enrolled in 61 hybrid classes into three groups to get a TE of 0 and an SE of 0. The numbers were much smaller in Summer 2021: we had 226 students enrolled in eight hybrid classes that we divided into two groups to get a TE of 12 and an SE of four. In all terms, the excess room capacity was

140, and so the SSE was 0. We report the final results for Fall 2020 through Summer 2021 in Section 2.2.3

For team-level scheduling, as discussed earlier, each learning team is assigned to attend classes in person or online together for each day of the term. For first-year students, networking is an important part of the MBA experience, and so we schedule to allow students to meet with as many other students as possible while attending classes together. We make the following contributions to the team-level scheduling problem.

- We present a novel formulation of the problem as a mixed-integer program. Our program tries to maximize networking opportunities between students while satisfying fairness criteria that we develop.
- We present empirical results from the application of our method to first-year MBA students at Columbia Business School. Our method was used for hybrid scheduling of 570 first-year students in Fall 2020 and 220 first-year students in Spring 2021.

For both of these problems, we used the Gurobi optimization software [33] version 9.5.1 and Python version 3.9.7 to solve the mixed-integer programs.

2.1.2 Other Applications

Aside from the present setting, our models can be used in other applications as well. The student-level scheduling model can be useful in "mode selection" problems where participants can be assigned to one of two modes, but there is a capacity constraint on the preferred mode. An example is traffic rationing. [34] review the simple license-plate-based driving restrictions that are employed in many cities around the world either for short-term or long-term traffic rationing to combat congestion, reduce pollution, and minimize fuel consumption. From the perspective of minimizing congestion, such restrictions are inefficient, however, since different motorists tend to use different sets of roads. An alternative to such restrictions can be a more complex individual restriction scheme. Under this scheme, each driver would submit a source (e.g., residence) and a

destination (e.g., workplace) for their daily commute. We then compute the geographical regions each driver travels through during their commutes. The drivers are then treated as students and the regions as courses to apply our student-level scheduling model. Once drivers are assigned to a group, groups are scheduled to drive using their own vehicle in a rotational manner. Those not scheduled to drive are required to take public transit or face penalties. The penalties could be administered using technology currently employed by electronic toll collection networks throughout the world.

The team-level scheduling model can be applied to worker assignment to remote versus online modes on a day-to-day basis. [35] observes that only 6% of the remote-capable workers desire a traditional office. Average office use in 10 major US cities is barely above 50% of its pre-Covid level as of February 2023 [36]. Some companies are already moving towards desk sharing [see, for example 37] to save on office real-estate needs. Deciding who works online and who comes to the office in person is a mode selection problem. These workers are often divided into teams, and for maximum intra-team collaboration, it is desirable to schedule everyone in the same team together. Thus our team-level scheduling model can be applied, allowing a planner to schedule a subset of teams to work from the office. Our objective ensures that inter-team interaction is maximized.

2.1.3 Literature Review

Educational scheduling literature can be broadly divided into three categories: timetabling, course allocation, and classroom allocation. Timetabling deals with assigning time slots to various educational activities, include teaching sessions or assessments. By far, most of the scheduling literature deals with timetabling; [38] and [39], survey this area. Solutions that have been proposed for this problem include the work of Burke and Bykov [40], Bellio *et al.* [41], Asín Achá and Nieuwenhuis [42], Abdullah and Turabieh [43], and Mirhassani [44]. On the empirical side, schools have analyzed and applied mixed-integer programming to solve timetabling problems in their specific situations [e.g., 45, 46, 47].

Course allocation is the problem of assigning courses to students or teachers. In many cases, a

course is divided into smaller sections for manageability, and students and teachers are assigned to these sections. We consider these to be instances of the course allocation problem. Optimization has been applied to this problem as well [e.g., 48, 49, 50, 51]. On the applied side, this problem has been studied along with timetabling [e.g., 52] and classroom allocation [e.g., 53, 54, 55].

Classroom allocation, the problem of assigning appropriate classrooms to educational activities, has also been studied individually [e.g., 56, 57], and in tandem with timetabling and course allocation [e.g., 54, 55, 58, 45].

Closer to our work are papers that deal with challenges posed by the Covid-19 pandemic. One line of literature broadly considers various problems experienced by different schools related to the pandemic. [59] apply mixed-integer programming models to redesign routes and schedules for the campus bus system during the pandemic. [60] study Covid-19 related interventions in university populations. [61] considers the problem of classroom design with social-distancing considerations.

Specifically with respect to scheduling, there have been a number of different approaches taken by various schools to implement social distancing. These range from fully online to fully in person classes [62]. [63] use integer optimization for timetabling and term planning during the pandemic. They expand their university's two-term schedule by adding a third term and redistribute the courses in these three terms so as to reduce in-person student population on campus. They also decide timetables for these courses to implement required social distancing protocols while also allowing as many students as possible to attend their classes in person. [64] study the problem of deciding course modes for each course. These could be fully remote (online), fully in person, or a hybrid of the two. They also reassign classrooms under the capacities, newly-reduced to implement social distancing, to the non-remote courses.

The Three Cohort Model (3CM) of [65] is closest to part of our work in that 3CM considers a similar problem as our student-level scheduling problem. However, our contribution differs from their work in the following ways:

- The excess (GE) of 3CM is the sum of "course excesses," where each "course excess" is defined to be the maximum of the excess for that course over all cohorts (groups). In contrast,

our TE metric is the sum of all excesses over courses and groups. TE is more operationally relevant since each instance of excess is an instance where a student is invited to campus, but was asked to attend a teaching session online, and thus creates disutility for that student. Minimizing the total number of such instances would better serve the students, since it more directly corresponds to maximizing an aggregate measure of welfare.

Ensuring that SE does not exceed the capacity of the excess room is also directly operationally relevant. Otherwise, the excess students would not have a Covid-safe place to attend their classes online while present on campus.

Finally, a low TD is desirable to create a more equal learning and teaching experience for students and teachers.

The 3CM objective is to minimize the GE, whereas our objective is to minimize a linear combination of the TE, the TD, and to ensure that the SE does not exceed a specified constant — namely the capacity of the excess room.

- In their setting, the number of groups is held fixed at 3, whereas we take this as a parameter, and discuss the trade-offs in deciding this value. Based on these trade-offs, we allow for changing this value from term-to-term. In practice, we divided the students in three groups in Fall 2020 and in Spring 2021, and two groups in Summer 2021.

- We provide theoretical insight by studying the relaxed linear program and give conditions that ensure that the optimal objective value of the relaxed program is 0. In other words, we identify necessary conditions so that zero excess can be achieved.

- [65] state that their model "may be well suited for situations in which all courses meet on the same set of days," but we do not believe this to be a requirement to apply this model. In fact, CBS course schedule does not satisfy this requirement, and yet we were able to apply a very similar model while being fair to the students. The concern cited by [65] is that "students would not be guaranteed an in-person course meeting each week" with this model. However, over the term, each student can still be guaranteed to attend each of their classes in person an

equal number of times regardless of their group. Based on the counterfactual performance of the 3CM, [65] say that the model "is inappropriate if we desire to maximize the number of students attending some in-person classes." In contrast, we are able to guarantee that each student enrolled in hybrid classes and willing to attend them in person, is scheduled to attend each of their classes in person at least once every M class meetings throughout the term.

- [65] ultimately abandon the 3CM approach for three reasons:

  1. They could not guarantee that each student attends each of their classes each week. This was true for us as well. However, we were still able to guarantee that over the term, each class was attended by each student an equal number of times.
  2. Their approach did not accommodate all classes, such as those that meet once a week. This was not an issue for us. In fact, in Fall 2020, almost 75% of classes scheduled to be hybrid met once a week.
  3. They found that the optimal solution produced by the 3CM exceeded the designated social distancing capacities of the classrooms by up to 40%. We eliminate this potential issue by recommending course enrollment capacities and number of groups based on our experiments on past data.

  The solution finally implemented at Clemson University is markedly different from the 3CM approach or our student-level scheduling approach. It consists of assigning each student to an in-person attendance group individually for each course. This allows for schedules in which some students would need to attend one class in person and others online on the same day of the week for the entire term, and so represents a set of trade-offs that we do not consider here.

As in many of the papers cited above, we apply mixed-integer programming techniques. Mixed-integer programming has been extensively studied. [66] and [67] are two excellent books that provide a comprehensive overview of mixed-integer programming and its applications to optimization. It has also been applied to scheduling problems in a wide variety of domains. [68] applies

mixed-integer programming to the problem of scheduling a workforce to meet varying demand, [69] apply it to the parallel machine-scheduling problem, and [70] apply it to scheduling problems in healthcare.

2.2 Student-Level Scheduling

We first consider the student-level scheduling problem. In this setting, different students can enroll in different classes and hence they are scheduled individually.

2.2.1 Formulation

We have a set $\mathcal{S}$ of students and a set C of classes. For each class $k \in C$, $\mathcal{A}_k$ is the set of students enrolled in it and c_k is its social distancing capacity. We denote by C_t the set of classes scheduled to be ongoing at time t, where t denotes an hour in a week. Sunday midnight corresponds to $t = 0$ and $t = T \triangleq 167$ corresponds to Saturday at 11:00 PM. $\mathcal{T}$ is defined to be the set of hours in a week. A class is ongoing at $t \in \mathcal{T}$ if t falls between its start-time and end-time with a 10-minute buffer on each side.

We need to assign each student to exactly one of M groups where M is some positive integer. Let $\pi_{i,j}$ be a variable denoting the assignment of student $i \in \mathcal{S}$ to group j, $1 \leq j \leq M$. Since each student can be assigned to only one group, we have

$$\sum_{j=1}^{M} \pi_{i,j} = 1, \ \forall i \in \mathcal{S}. \tag{2.1}$$

When $\pi_{i,j}$ are all assigned integer values, we call the assignment an integral one. A fractional student assignment is one where each student is fractionally assigned to multiple groups and the mass of this student in these groups sums up to unity (as per constraint 2.1). A relevant fractional student assignment is the uniform one, which we now define.

Definition 2.1 (Uniform fractional student assignment) *A uniform fractional student assignment is a fractional student assignment where each student is assigned to each of the M groups such*

that $\pi_{i,j} = 1/M \ \forall i \in \mathcal{S}, 1 \leq j \leq M$.

The uniform fractional student assignment is ideal in that it is perfectly balanced and symmetric across both classes and students, and hence is a useful benchmark. However, being fractional, it cannot be implemented.

For each day of the term d, we pick a group j to be assigned to in-person attendance for all of the day's classes. We cycle through all groups one by one for each day of the term. In a fair allocation, each student will have an opportunity to attend each of their classes in person an equal number of times. Because each class in the school follows a schedule that repeats weekly, this can be achieved only if each student attends class in person each day of the week a roughly equal number of times throughout the term. This is straightforward if M is co-prime with the number of days classes are scheduled. Otherwise, we need to vary the repeating sequence appropriately to accomplish this. Figure 2.1 shows the group assignment calendar for the first half of Fall 2020,

Figure 2.1: Group assignment calendar for the first half of Fall 2020. Here, for each teaching day (Monday–Thursday), one of the 3 groups (C, B, or S) is assigned to attend classes in person. Note that September 18th was included as a teaching day (running a Monday schedule), in order to make up for the Labor Day holiday (September 7th).

with three groups (named C, B, and S) as it was implemented at CBS. Note that classes take

place on 4 days of the week (Monday–Thursday) and the assignment is rotated through the three groups — C, followed by B, followed by S. Then, for any day of the week, the group assigned for that day also rotates through the same sequence (e.g., group C is assigned for the first Monday, followed by group B for the second Monday, etc.). This ensures that each day of the week sees an equal number of each of the three groups scheduled throughout the term. In Summer 2020, there were two groups scheduled over the same four teaching days. Since two and four are not co-prime numbers, we need a different approach to group scheduling. In particular, the two groups (named A and B) are scheduled in an ABAB sequence in one week and a BABA sequence the next, alternating throughout the term.

As can be seen from Figure 2.1, on any day of the week, any one of the M groups may be assigned an in-person schedule. Hence, when considering in-person attendance for any class, we must assume that any one of the M groups may be assigned in-person attendance for that class. We also note that once we have an integral assignment, the protocol described above produces an inherently fair schedule that ensures that every student attends classes in person for a fraction $1/M$ of the days. Our problem then reduces to that of computing a group assignment. We now discuss ways to evaluate a given assignment by defining some quantities of interest.

Let $\mathcal{T}_d \subseteq \mathcal{T}$ be the set of hours on day d. Then, the set of students assigned to in-person attendance when group j is assigned to day d is $\left\{i \in \mathcal{S} : \pi_{i,j} = 1, i \in \cup_{t\in\mathcal{T}_d} \cup_{k\in C_t} \mathcal{A}_k\right\}$. If a class has more assignments than the social distancing capacity, we assign some students randomly to an "excess room" for the duration of that class. At time t, if group j is assigned to in-person attendance, the excess room will have s_j^t students, where

$$s_j^t \triangleq \sum_{k\in C_t} \left(\sum_{i\in\mathcal{A}_k} \pi_{i,j} - c_k \right)^+ . \tag{2.2}$$

The excess room has capacity E, and if s_j^t exceeds E at any time t for group j, the surplus students would not be able to attend classes online while being on campus.

Now, it is imperative that we assign groups in a way that minimizes the excess occupancy.

Furthermore, it is undesirable to have a large imbalance in attendance in any given class when different groups are scheduled to attend classes in person. We define the following performance metrics for a group assignment.

- **Surplus Simultaneous Excess (**SSE**)**. SE is the maximum number of students assigned to the excess room at any given time. We have

$$\text{SE} \triangleq \max_{1\le t\le T, 1\le j\le M} s_j^t. \tag{2.3}$$

  We are interested in how much SE exceeds E, the excess-room capacity. That is, SSE = $(\text{SE} - E)^+$, is our metric of interest.

- **Total Excess (**TE**)**. Every time a student is scheduled to attend classes in person, but is assigned to an excess room, it creates disutility for that student. TE measures this disutility by summing up the excesses across all classes assuming each of the groups is scheduled to in-person attendance. We have

$$\text{TE} \triangleq \sum_{1\le j\le M} \sum_{k\in C} \left(\sum_{i\in\mathcal{A}_k} \pi_{i,j} - c_k \right)^+. \tag{2.4}$$

- **Total Deviation (**TD**)**. It is undesirable to have too many in-person students on one day and too few on another. To avoid such outcomes, we also need to minimize TD, which is the sum of the absolute difference between an assignment and the uniform fractional student assignment that divides each student's mass equally among all groups (see definition 2.1). That is,

$$\text{TD} \triangleq \sum_{1\le j\le M} \sum_{k\in C} \left| \sum_{i\in\mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M} \right|. \tag{2.5}$$

Figure 2.2 shows an example assignment of 13 students, each enrolled in a subset of three specific classes divided into two groups: group 1 (green) and group 2 (white). In this example, we see an excess of one each from classes 2 and 3 when group 1 is scheduled, and an excess of one

from class 2 when group 2 is scheduled. The TE is thus three. We note that classes 2 and 3 overlap on Thursdays between 13:00 and 13:30, and if group 1 is scheduled to attend classes in person, both these classes would have an excess of one each. The SE is thus two. Since the excess-room capacity is two, the SSE is 0. The contribution to the TD from class 1 is 0, from class 2 is 0, and from class 3 is two, making the TD equal to two.

Clearly, a desirable assignment would have a low SSE, a low TE, and a low TD. To achieve such an assignment, we formulate an optimization problem that tries to minimize a linear combination of these three quantities while being constrained so that each student is assigned to exactly one group. Succinctly, our optimization problem (P) can be described as follows:

$$\begin{aligned} &\text{minimize} && \text{A weighted sum of SSE, TE, and TD} \\ &\text{subject to} && \text{Each student being assigned to exactly one group.} \end{aligned} \tag{2.6}$$

A formal integer programming formulation with auxiliary variables follows:

$$\text{minimize} \quad \sum_{k \in C} \sum_{j=1}^{M} e_{jk} + \lambda \sum_{k \in C} \sum_{j=1}^{M} \delta_{jk} + \mu s \tag{2.7}$$

$$\text{subject to} \quad \sum_{j=1}^{M} \pi_{ij} = 1 \qquad \forall i \in \mathcal{S} \tag{2.8}$$

$$\sum_{i \in \mathcal{A}_k} \pi_{ij} - c_k \leq e_{jk} \qquad 1 \leq j \leq M,\ \forall k \in C \tag{2.9}$$

$$-\delta_{jk} \leq \sum_{i \in \mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M} \leq \delta_{jk} \qquad 1 \leq j \leq M,\ \forall k \in C \tag{2.10}$$

$$s \geq s_j^t - E \qquad 0 \leq t \leq T,\ 1 \leq j \leq M \tag{2.11}$$

$$s_j^t \geq \sum_{k \in C_t} e_{jk} \qquad 0 \leq t \leq T,\ 1 \leq j \leq M \tag{2.12}$$

$$\pi_{ij} \in \{0, 1\} \qquad \forall i \in \mathcal{S},\ 1 \leq j \leq M \tag{2.13}$$

$$\delta_{jk},\ e_{jk}, s \geq 0 \qquad 1 \leq j \leq M,\ \forall k \in C, \tag{2.14}$$

where, $e_{jk}, \delta_{jk}, s_j^t$, and s are auxiliary variables. At the optimum, e_{jk} is set to the excess in class k

when group j is scheduled to attend classes in person; i.e., $e_{jk} = \left(\sum_{i \in \mathcal{A}_k} \pi_{i,j} - c_k\right)^+$, δ_{jk} is set to the contribution to TD by class k when group j is scheduled in-person; i.e., $\delta_{jk} = \left|\sum_{i \in \mathcal{A}_k} \pi_{ij} - \frac{|\mathcal{A}_k|}{M}\right|$, s_j^t follows equation (2.2), and s is set to $\left(\max_{t,j} s_j^t - E\right)^+$. Thus, at the optimal solution, we have TE = $\sum_{j,k} e_{jk}$, TD = $\sum_{j,k} \delta_{jk}$, SE = $\max_{t,j} s_j^t$, and SSE = $s = (\text{SE} - E)^+$. By equation (2.7), the above mixed-integer program's objective is to minimize a weighted sum of TE, TD, and SSE

We note that, for us, the SSE is the most important metric since, if it is positive, that would indicate our inability to house excess students in a Covid-safe environment. TE is the next most important metric since it directly corresponds to social welfare. By comparison, TD is the least important metric. The objective of our integer program is to minimize a weighted sum of these three metrics. This weighted sum is our objective function. In practice, however, we observe that SE takes very small values relative to the excess-room capacity even when we do not include it in the objective function. Therefore, we set the weight of SSE to 0 in the objective function, and compute it after the resulting optimal solution is obtained. We weight the rest of the metrics as follows:

- The weight of the TE metric is normalized to be one.
- The weight of the TD metric, λ, needs to be set to be between zero and one. In practice, we set it to[2] 0.25.

Finally, the choice of M is important as well. M should be kept small enough so that students can hope to attend as many classes in person as possible. However, it should not be so small that the excess is large.

2.2.2 Analysis

Let us consider a simplified version of the problem: one without the integer constraints (called the relaxed linear program). Essentially, this allows for each student to be fractionally assigned to multiple groups. One such assignment would be the uniform fractional student assignment. In fact,

[2]In practice, the optimization program managed to achieve the minimum possible TE, and so the weight proved not to be so important.

it turns out that this is the optimal fractional student assignment for the relaxed linear program. We define the TE in the uniform fractional student assignment as the uniform excess (UE). That is,

$$\text{UE} \triangleq \sum_{k \in C} M \left(\frac{|\mathcal{A}_k|}{M} - c_k \right)^+ . \tag{2.15}$$

Clearly, UE is 0 when, for each $k \in C$, we have

$$|\mathcal{A}_k| \leq M c_k. \tag{2.16}$$

If this holds for class k, we say that class k satisfies the "minimal-capacity condition."

This gives us insights into the original problem as the optimal objective value of the integer program must be lower bounded by the optimal objective value of the relaxed linear program. We can prove the following theorem.

Theorem 2.2.1

1. *If TE is the only metric with non-zero weight in the objective function, and integer constraints are relaxed, the optimal objective value is equal to the UE. Therefore, the TE of any feasible allocation for the integer program is lower bounded by the UE.*

2. *The optimal objective value of the relaxed linear program is attained by the uniform fractional student assignment.*

The proof follows from applying strong duality to the relaxed linear program. See Appendix B.1.1 for the detailed proof.

Example 2.1 below shows that even when we restrict the objective function so that TE is the only metric with non-zero weight, the optimal objective value may not always be equal to the UE.

Example 2.1 *Suppose that we have three students, each enrolled in two of three classes, and class has a social distancing capacity of one. The enrollments are such that for every pair of students, there is a unique class that pair is enrolled in. Suppose that we set M to two. Clearly, we must have*

at least one group with at least two students. Then, we will have at least one class with an excess of one. Thus the TE *is at least one, whereas the* UE *is* 0 *since assigning half of each student's mass to each of the two groups results in an excess of* 0*.*

We observe that the integer constraints are binding in this case, since otherwise both the TE and the UE would be equal to 0. Note that in the same example, if we increase the number of groups to three, we can assign each student to a different group, achieving a TE of 0. This suggests that the enrollment of students to classes and the choice of M both play a role in making the integer constraint binding.

We can also come up with a lower bound on TD. If we consider each course in isolation, the problem of minimizing its contribution to TD is a fair division problem — the problem of dividing its students into M groups equally as much as possible. This problem can be solved by assigning students one-by-one to a group in a round-robin fashion. This way, we can compute the minimum possible contribution to the TD from each course and aggregate it to get the minimal deviation (MD). Formally, we define the minimal deviation of a course k as

$$\mathrm{MD}_k \triangleq 2 \cdot |\mathcal{A}_k| \bmod M \cdot \left(1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}\right), \tag{2.17}$$

where $\{x\}$ denotes the fractional part. That is, $\{x\} = x - \lfloor x \rfloor$, where $\lfloor x \rfloor$ is the greatest integer less than or equal to x. The minimal deviation for the problem is given by

$$\mathrm{MD} \triangleq \sum_{k \in C} \mathrm{MD}_k \,. \tag{2.18}$$

We can prove the following theorem.

Theorem 2.2.2 *The TD is lower bounded by the MD.*

See Appendix B.1.2 proof of Theorem 2.2.2.

Getting a TE of 0: From the above discussion, we conclude that to get a TE of 0, we must first have a UE of 0, but that alone does not guarantee a TE of 0. To get a UE of 0, it suffices to set a

registration limit on each class so that dividing each student's mass equally among M groups gives an excess of 0 from that course. This is the minimal-capacity condition in equation (2.16). In the next section, we look at past data, and observe that this may actually lead to a TE of 0 in reality even though that is not theoretically guaranteed.

2.2.3 Empirical Results

We tested our methodology on Fall 2019 data before using it in production in the Fall 2020 term. The data (provided to us by the CBS Dean's Office) consisted of 739 students, each enrolled in 5–6 classes for a total of 4,204 enrollments. Each class had an average of ≈ 60 students enrolled in it. The data was high quality, requiring little to no cleaning. When conducting the experiments, we assumed all 117 classes to be in a hybrid mode. In contrast, in Fall 2020 and Spring 2021, there were over a thousand students enrolled in 115 and 113 classes, respectively. Not all of these classes were held in a hybrid mode, however: about half of the classes were held fully online, while the rest were held in a hybrid mode. Based on our Fall 2020 data indicating a low turnout of students to in-person classes, we conducted a survey prior to the beginning of classes in Spring 2021 to find out if any students wanted to opt out of in-person attendance. Of the 1,553 students registered, 1,035 opted to attend classes in person. In Summer 2021, we had only 336 students enrolled in eight classes, of which 226 opted to attend classes in person, and only those were considered for hybrid scheduling. Table 2.1 summarizes these statistics.

Term	Students	Total Classes	Hybrid Classes	Total Enrollments
Fall 2019	739	117	117	4204
Fall 2020	1302	115	67	5815
Spring 2021	1035 (of 1553)	133	61	7544
Summer 2021	226 (of 336)	8	8	565

Table 2.1: Student enrollment statistics. Our method was tested on pre-Covid Fall 2019 data. For this test, all classes were assumed to be in a hybrid mode of attendance.

We implemented the optimization program using the gurobipy [33] library version 9.5.1 in the Python programming language version 3.9.7. To run the programs, we used AMD EPYC 7532 3.2–

3.3 GHz processors. Each program used up to 30 threads and was allocated up to 100 GB RAM. An optimal solution was found in almost all cases within a tolerance of 10^{-4} in the objective value. Only in one case, the program ran out of memory before reaching the optimal solution. The run time of each program ranged from two hours to four days on the Fall 2019 data set, but was less than five minutes on Fall 2020, Spring 2021, and Summer 2021 data sets.

While testing with the Fall 2019 data, we considered three potential social-distancing protocols (the final protocol had not been decided at the time of these experiments), namely,

`6ft` : six feet of distance between every two students.

`4ft` : four feet of distance between every two students.

`50sf` : fifty square feet of area given to each student.

In general, classroom capacities in the `6ft` social distancing protocol were less than that in the `50sf` social distancing protocol, which in turn were approximately equal to the capacities in the `4ft` social distancing protocol. These experiments are summarized in Table 2.2. The MIP Gap column reports the maximum difference between the objective value of the solution obtained and the optimal objective value in that setting. Note that in all except one case, the optimizer reached an optimal solution (MIP Gap = 0). In the case when we used the `50sf` social distancing protocol and divided the students into six groups, the optimizer ran out of memory before reaching an optimal solution.

As can be expected, the TE, UE and SE decrease with an increasing number of groups. We observe that keeping the number of groups constant, these excesses decrease when going from `6ft` to `4ft`, and from `4ft` to `50sf` social-distancing protocols. Figure 2.4 shows this trend for TE.

Two more interesting observations can be made from the results. The first observation is that the TD in the solution obtained is equal to the MD in almost all the experiments. The only case where we did not obtain a solution with TD equal to MD is also the only case when the optimizer ran out of memory before reaching the optimal solution.

Social-Distancing Protocol	M	Total Excess	Uniform Excess	Simultaneous Excess	Total Deviation	Minimal Deviation	Objective Value	MIP Gap
6ft	6	1	1	1	252.00	252.00	64.00	0
6ft	5	25	25	2	180.80	180.80	70.20	0
6ft	4	263	263	10	158.00	158.00	302.50	0
6ft	3	753	753	33	100.00	100.00	778.00	0
6ft	2	1595	1595	79	60.00	60.00	1610.00	0
4ft	6	0	0	0	252.00	252.00	63.00	0
4ft	5	0	0	0	180.80	180.80	45.20	0
4ft	4	3	3	1	158.00	158.00	42.50	0
4ft	3	58	58	5	100.00	100.00	83.00	0
4ft	2	601	601	41	60.00	60.00	616.00	0
50sf	6	0	0	0	253.00	252.00	63.25	0.25
50sf	5	0	0	0	180.80	180.80	45.20	0
50sf	4	0	0	0	158.00	158.00	39.50	0
50sf	3	1	1	1	100.00	100.00	26.00	0
50sf	2	417	417	28	60.00	60.00	432.00	0

Table 2.2: Summary of experimental results of hybrid scheduling with different values of M, and different social distancing policies. All assignments were computed setting the weight of the TD metric to 0.25, and the weight of the surplus SE to 0.0 in the objective function. The capacity of the excess room was determined to be 140, and hence, the surplus SE was 0 in all cases. An optimal solution was reached in all except for the one case with 50sf social distancing protocol and six groups. The MIP Gap indicates the maximum difference between the objective value of the obtained solution and the optimal objective value.

The second, and more operationally relevant, observation is that the UE is equal to the TE of the assignment computed by our optimizer in all of our experiments on the past data, meaning that the relaxation of the integer constraint is tight as far as minimizing the TE is concerned. We could hope, then, that the optimal TE would be 0 if the UE were set to 0.

To ensure a UE of 0, we need to ensure that the minimal-capacity condition is satisfied for each class. That is, the enrollment limit of each class is upper bounded so that the social distancing capacity is sufficient to allow a fraction $1/M$ of the enrolled students to attend that class in person. In this scenario, a large M allows more students to enroll in classes of their choosing, and a small M gives more in-person attendance opportunities to students.

When the social distancing protocol and the corresponding social distancing capacities were finalized (these turned out to be between the 6ft and 50sf/4ft capacities described above), we

observed that these were 25%–30% of the original capacities of almost all of the classrooms. This meant that if M was set to four, enrollment capacities would have increased or remained the same as previous years. Furthermore, based on the finalized classroom capacities, we expected a low impact on TE when going from four groups to three groups. A policy decision was made to set M to three. This caused a reduction in enrollment capacities while allowing students to attend classes in person once every three days.

Another important decision was the setting of the weights of the different metrics in the objective function. We normalized the weight of the TE metric to one. As discussed earlier, the SSE metric is important only if the SE values in different settings, computed on an assignment that was optimal without the SE metric, are comparable to the social distancing capacity of our excess room. The social distancing capacity of the designated excess room was determined to be 140 seats. As can be seen in Table 2.2, all of the settings we tried on the past data resulted in an optimal assignment (computed without the SE metric in the objective function) having a simultaneous excess much less than the excess room capacity. Therefore, we concluded that the weight of the SSE did not matter in our case, and we set it to 0. The weight of the TD metric was set to 0.25, though we observe that the exact value did not matter in the resulting problem since we always managed to achieve a TE equal to the UE.

Table 2.3 shows the final results of hybrid scheduling as implemented in the Fall 2020, Spring 2021, and Summer 2021 terms at Columbia Business School for second-year MBA students. Students enrolled only in fully online classes were also assigned to a group since groups were used for non-instructional gatherings among students as well (for example, if a club activity was scheduled on a day assigned to group 1, only group 1 students could attend it in person). It can be seen that in all years, the TE was always equal to the UE. While care was maintained in making sure that the minimal-capacity condition was satisfied by controlling registration for most courses, this constraint was slackened for some, which led to the positive UE and TE in the Fall 2020 and Summer 2021 terms.

In Fall 2020, we observed that many students decided to stay at home even when given an

option to attend classes in person. In light of this, a survey was conducted before the start of the Spring 2021 term, where students indicated their preferred mode of attendance. Only those who were interested in attending classes in person were considered for scheduling. While we hoped to be able to set M to two for scheduling in Spring 2021, it turned out to be infeasible since many students opted to attend classes in person. In Summer 2021, however, the number of students who opted to attend classes in person was low enough to allow us to set M to two, while still maintaining higher enrollment limits.

Term	M	Total Excess	Simultaneous Excess	Uniform Excess	Total Deviation
Fall 2020	3	20	5	20	382.67
Spring 2021	3	0	0	0	719.33
Summer 2021	2	12	4	12	6

Table 2.3: Summary of the final results of hybrid scheduling in the Fall 2020, Spring 2021, and Summer 2021 terms.

2.2.4 Deployment

The deployment took place as per the following steps:

1. As noted earlier, not all classes were held in a hybrid mode. The first step was to determine which classes were to be held fully online and which in a hybrid mode. This was determined based on course content and instructor preference.

2. In the next step, the social distancing capacities were determined for each classroom. The meeting times, and classroom assignments for each class were also finalized.

3. A tentative number of groups, M, was determined. Along with classroom capacities, this decided the enrollment limit for each class as per the minimal-capacity constraint.

4. Students were allowed to enroll in classes. There enrollments were not in our control except for deciding the enrollment limits.

5. Students indicated whether they wanted to opt out of in person classes altogether for Spring 2021 and Summer 2021 terms only.

6. We computed group assignments for students who did not opt out, and were enrolled in at least one hybrid class, based on actual enrollments with M groups. We also checked if the number of groups could be lowered without increasing TE and SE to unmanageable levels. We did this by recomputing the assignments with lower values of M.

7. Based on the results, we decided the final value of M.

8. We randomly assigned a group to the students who were not assigned one so far.

9. The Dean's Office disseminated the group assignments for each student along with the group calendar for the term.

In all three terms, the tentative number of groups in Step 3 was determined to be three. In Fall 2020 and Spring 2021 terms, this was also the final number of groups. As discussed in the previous section, based on low student turnout in Fall 2020, students were given an option to opt out of in-person classes in Spring 2021 and Summer 2021 terms. In both terms, we tried group assignments with two groups in Step 6. In Spring 2021 this resulted in a high value of TE and the number of groups was finalized to be three. In Summer 2021, a group assignment with two groups produced a manageable level of TE and SE, and the number of groups was finalized to be two. As a result of this, we were able to accommodate more willing students to attend in-person classes without reducing class enrollment limits. Because each computation only took less than five minutes to run, such repeated computations were not a challenge.

To increase efficiency further, students were given another chance to opt out of in-person classes on any day of their choosing through an app. Other students were then chosen to attend those classes in person on that day. This was handled by the Dean's Office.

2.3 Team-Level Scheduling

MBA programs prepare students for the world of business by teaching them collaboration and networking. To aid collaboration, first-year MBA students at Columbia are divided into learning teams (LTs), which are assigned group tasks and attend classes together. Networking is usually not an issue since in-person class attendance automatically encourages that. With a hybrid model, however, we need to ensure that every student has a chance to meet every other student in person as well. We call this **mixing** among students. Mixing is particularly important for first-year students, and we need to encourage it along with considerations such as fairness. We formally define these considerations in the next section.

2.3.1 Problem Formulation

Since all first-year students attend the same set of classes, we can treat each team as an individual entity, drastically reducing the number of variables. Suppose that we have N teams, and decide to ask K of them to attend classes in person each day for a term comprising T days, where $N \mod K \equiv 0$. In our case, we have $N = 12$, $K = 4$, and $T = 44$. Let $\mathcal{S}$ be the set of all subsets of size K of $\{1, 2, \ldots, N\}$. We call each such subset an LT group. Then $|\mathcal{S}| = \binom{N}{K} = 495$. Exactly one LT group should attend classes in person on each day of the term, and we can again use integral variables to ensure this. We define $\pi_{s,t}, s \in \mathcal{S}, t \in \{1, 2, \ldots, T\}$ to be the variable denoting the assignment of group s to an in person class-attendance on day t. Since we intend to invite exactly one subset of size K to attend classes in person each day, we need,

$$\sum_{s \in \mathcal{S}} \pi_{s,t} = 1 \qquad 1 \leq t \leq T + p. \tag{2.19}$$

In this case, a fractional group assignment is one where on any given day, multiple LT groups are attending classes in person fractionally, while their total mass sums up to unity. A relevant fractional group assignment is the uniform one, which we now define.

Definition 2.2 (Uniform Fractional Group Assignment) *A uniform fractional group assignment is a fractional group assignment in which on each day of the term, a fraction* $1/495$ *of each LT group is attending classes in person.*

We have two notions of fairness.

- Longitudinal balance. Days of in-person attendance should be equally distributed throughout the term. One way to ensure this is to make sure that each team attends classes in person once every $N/K = 3$ days. More formally, we define $\mathcal{B}$ to be the set of $\left\lceil \frac{T}{N/K} \right\rceil$ blocks, each consisting of N/K consecutive days. Note that this may require the creation of additional dummy days. For example, in our case, we have $\mathcal{B} = \{\{1, 2, 3\}, \{4, 5, 6\}, \ldots, \{43, 44, 45\}\}$, where $t = 45$ is a dummy day added to make every element of $\mathcal{B}$ of equal size. We have $|\mathcal{B}| = \left\lceil \frac{T}{N/K} \right\rceil = 15$. Longitudinal balance then requires that

$$\sum_{s:i\in s} \sum_{t:\mathrm{bl}(t)=b} \pi_{s,t} = 1 \quad \forall b \in \mathcal{B}, 1 \le i \le N, \tag{2.20}$$

  where $\mathrm{bl}(t)$ is the block in $\mathcal{B}$ containing the day t (which may be a dummy day). We assume that we need to create p dummy days. As noted above, in our case, $p = 1$.

- Balance over days of the week. Since classes follow a schedule that repeats weekly, each team should attend classes in person on each day of the week an equal number of times. Let $\mathcal{D}$ be the set of days of the week, and let $\mathrm{dow}(t)$ denote the element of $\mathcal{D}$ corresponding to day t (day of the week on day t). For $d \in \mathcal{D}$, let $\mathrm{dow}^{-1}(d) \triangleq \{t \in [T] : \mathrm{dow}(t) = d\}$. Then each team should have between $l^d_{\min} \triangleq \left\lfloor \frac{\left|\mathrm{dow}^{-1}(d)\right|K}{N} \right\rfloor$ and $l^d_{\max} \triangleq \left\lceil \frac{\left|\mathrm{dow}^{-1}(d)\right|K}{N} \right\rceil$ days of in-person attendance falling on day d of the week. That is,

$$l^d_{\min} \le \sum_{s:i\in s} \sum_{t:\mathrm{dow}(t)=d} \pi_{s,t} \le l^d_{\max} \quad \forall d \in \mathcal{D}, 1 \le i \le N. \tag{2.21}$$

  For example, in our case $\mathcal{D} = \{M, T, W, R\}$ and $\left|\mathrm{dow}^{-1}(M)\right| = 11$. This means that each team should be able to attend classes in person on Mondays between $l^M_{\min} = 3$ and $l^M_{\max} = 4$

times.

As mentioned earlier, our objective is to encourage mixing among students. We can do this by maximizing the minimum number of times any two teams attend classes in person together. That is, we maximize the quantity

$$\min_{1\le i,j\le N} \sum_{s:i,j\in s} \sum_{t\in[T]} \pi_{s,t}. \tag{2.22}$$

Note that dummy days are not to be counted when computing the number of times any two teams meet. Succinctly, we can write the optimization problem as

$$\begin{aligned} &\text{maximize} && \text{Minimum number of in-person meetings between any two teams} \\ &\text{subject to} && \text{Exactly four LTs attending classes in person on each day} \\ &&& \text{Longitudinal balance being maintained} \\ &&& \text{Balance over days of the week being maintained.} \end{aligned} \tag{2.23}$$

A mathematical formulation using auxiliary variables follows:

$$\max_{\pi, m_{\min}} \quad m_{\min} \tag{2.24}$$

$$\text{subject to} \quad \sum_{s\in\mathcal{S}} \pi_{s,t} = 1 \qquad 1 \le t \le T+p, \qquad \text{(2.19 revisited)}$$

$$\sum_{s:i\in s} \sum_{t\in[T+p]:\mathrm{bl}(t)=b} \pi_{s,t} = 1 \qquad \forall b \in \mathcal{B}, 1 \le i \le N, \qquad \text{(2.20 revisited)}$$

$$l^d_{\min} \le \sum_{s:i\in s} \sum_{t\in[T]:\mathrm{dow}(t)=d} \pi_{s,t} \le l^d_{\max} \qquad \forall d \in \mathcal{D}, 1 \le i \le N, \qquad \text{(2.21 revisited)}$$

$$\sum_{s:i,j\in s} \sum_{t\in[T]} \pi_{s,t} \ge m_{\min} \qquad 1 \le i, j \le N, \tag{2.25}$$

$$\pi_{s,t} \in \{0,1\} \qquad \forall s \in \mathcal{S}, \forall t \in [T+p], \tag{2.26}$$

where $m_{\min}$ is an auxiliary variable that, at the optimum, is set to the minimum number of days

any two teams are together in person. That is, at the optimum, we have

$$m_{\min} = \min_{1 \le i,j \le N} \sum_{s: i,j \in s} \sum_{t \in [T]} \pi_{s,t},$$

which is consistent with our choice of objective function according to (2.22).

2.3.2 Insights

Again, we can draw useful insights into the problem by solving the relaxed linear program. In effect, this would allow fractional group assignments. We are able to prove the following theorem.

Theorem 2.3.1

1. *The team-level scheduling problem with relaxed integer constraints has an optimum when each day of the term has a uniform fractional group assignment, that is, one where* $\pi_{s,t} = \frac{2}{N(N-1)}$, $s \in \mathcal{S}, 1 \le t \le T + p$.
2. *The optimal objective value when integer constraints are relaxed is* $\frac{TK(K-1)}{N(N-1)}$.
3. *In any optimal solution, each team meets with every other team an equal number of times.*

We defer the proof to Appendix B.2.1. We note that the optimal objective value as per Theorem 2.3.1 Part 2 evaluates to 4.0 in our case. We are now able to observe that we must have an optimality gap when solving the corresponding integer program. Suppose to the contrary that there is a solution where every pair of teams meet at least four times. For any given team, we know that it meets with exactly three other teams every day. Then, to meet each of the 11 other teams four times, the team needs to attend classes in person at least 15 days (rounded up from 14.67). Thus, we would need each of the 12 teams to attend classes in person for at least 15 days of the term. Since every day four teams attend classes in person, this would require at least 45 days in the term, while we have only 44 days. Thus, an objective value of four is impossible and we are left with an optimality gap. In fact, as we discuss in the next section, the optimizer was able to find a solution with an objective value of three as the optimum.

2.3.3 Results

The only data required for this problem was the number of days of the term, the number of learning teams, and the number of teams which are to be scheduled to attend classes in person on each day. We formulated the optimization program using the gurobipy [33] library version 9.5.1 in Python version 3.9.7, and used AMD EPYC 7532 3.2 GHz processor to compute a solution. The optimizer was allocated up to 30 threads and up to 100 GB RAM. The optimal schedule thus computed had an objective value of three. That is, our schedule ensured that each team met with every other team at least three times throughout the term. Table 2.4 shows the number of times

Learning Team	1	2	3	4	5	6	7	8	9	10	11	12
1	15	4	4	5	5	3	7	3	4	3	4	3
2	4	15	3	3	5	6	6	4	5	3	3	3
3	4	3	15	5	5	4	3	5	3	4	4	5
4	5	3	5	14	4	4	4	3	4	3	3	4
5	5	5	5	4	15	3	4	5	4	3	4	3
6	3	6	4	4	3	15	4	4	5	5	3	4
7	7	6	3	4	4	4	15	4	3	3	3	4
8	3	4	5	3	5	4	4	15	4	4	4	5
9	4	5	3	4	4	5	3	4	15	6	4	3
10	3	3	4	3	3	5	3	4	6	14	5	3
11	4	3	4	3	4	3	3	4	4	5	14	5
12	3	3	5	4	3	4	4	5	3	3	5	14

Table 2.4: Meeting matrix between teams according to the optimal schedule. Each element of the matrix shows the number of times the learning team numbered in the top row meets with the team numbered in the leftmost column in person.

each team met with every other team. The diagonal entries show the number of times each team attended classes in person throughout the term.

We observe that, while every pair of teams meet at least three times, the actual number of meetings varies quite a bit from pair to pair, ranging from three to seven. Also, we observe that the problem as it is stated in (2.23) can have multitudes of distinct optimal solutions that all have the same objective value. Deciding which of these solutions to implement can then be an interesting problem, though in our implementation, we did not pursue this. Since increasing the number of

interactions between students is our goal, we can come up with many different objective functions instead of the one given by equation (2.24). We list some of them here and discuss how their choice affects the imbalance noted above. Let m_{ij} denote the number of times teams i and j both attend classes in person. That is,

$$m_{ij} \triangleq \max_{1 \le i,j \le N} \sum_{s: i,j \in s} \sum_{t \in [T]} \pi_{s,t}, \tag{2.27}$$

and let $m_{\max} \triangleq \max_{1 \le i,j \le N} m_{ij}$.

F1: Maximize $m_{\min}$. This is the same as equation (2.24), and was used in real life. While it does encourage different teams meeting multiple times, it may lead to an imbalance due to some pairs of teams meeting many more times than others.

F2: Maximize $m_{\min} - \alpha m_{\max}$. This would mitigate the imbalance between different pairs of teams in terms of their number of meetings.

F3: Minimize $\max_{1 \le i,j \le N} \left| m_{ij} - 4 \right|$. This again reduces the imbalance between different pairs of teams by forcing each $m_{i,j}$ to be as close to four as possible. Here, four is picked because it is the value of $m_{ij}, 1 \le i \ne j \le N$ in the optimal solution of the relaxed linear program as per Theorem 2.3.1 Part 3.

Figure 2.5 shows a comparison of these objective functions. We see that both of the alternate objective functions F2 and F3 mitigate the imbalance by restricting the number of pairwise meetings to a number between three and five. While we see that the two lead to different solutions, we also observe that they need not do so by the nature of the respective functions alone. Both solutions have the same objective values when evaluated with either of the objective functions. The difference, then, seems to be due to some quirk of the optimizer itself. This again points to the fact that there are many degrees of freedom in this problem that we have not explored so far.

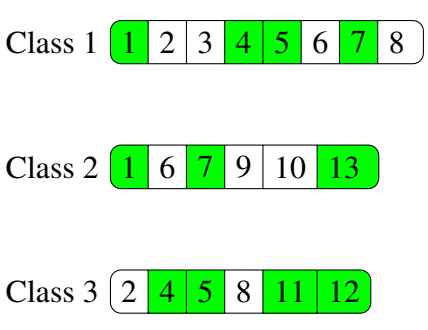
Another possible objective function, we designate as F4 is one that minimizes the sum of the absolute difference between the number of times each pair of teams meet and four (instead of minimizing the maximum absolute difference as in F3). This would then penalize each pair of

teams for meeting more or less than four times. The optimizer when trying to solve the problem with objective function F4 ran out of memory before converging to a solution.

2.4 Conclusion

We were able to use mixed-integer programming to formulate and solve the hybrid scheduling problem. We considered two settings of the problem: student-level and team-level scheduling. In the case of student-level scheduling, we were able to use insights obtained from our analysis of the relaxed linear program to make policy decisions that helped us get to a more efficient solution, i.e., one with low excess. The solution we computed was used to schedule over 2,500 second-year MBA students enrolled in 136 hybrid classes across three terms at Columbia Business School.

In the case of team-level scheduling, we were able to use the greater control such scheduling affords due to the homogeneity of the data in order to obtain an even more efficient schedule. Specifically, we were able to guarantee that each learning team met with every other learning team at least three times during the course of the 44-day term. Our model was used to schedule 790 first-year MBA students to their classes across two terms.



Excess-room capacity = 2.

(a) Class enrollments, timetable, capacities and group assignments.

Class	Group 1	Group 2	Total
Class 1	0	0	0
Class 2	1	1	2
Class 3	1	0	1
Total	2	1	TE=3

(b) TE computation. Each cell contains the excess from the class on the left when the group in the top row is scheduled to attend classes in person. Rightmost column contains contribution to TE from each class. Bottom row contains contribution to TE from each group. The bottom right cell contains the TE.

Class	Group 1	Group 2	Total
Class 1	0	0	0
Class 2	0	0	0
Class 3	1	1	2
Total	1	1	TD=2

(c) TD computation. Each cell contains the contribution to TD from the class on the left when the group in the top row is scheduled to attend classes in person. Rightmost column contains contribution to TD from each class. Bottom row contains contribution to TD from each group. The bottom right cell contains the TD.

Day and	Group 1				Group 2			
Time	Class 1	Class 2	Class 3	Total	Class 1	Class 2	Class 3	Total
Monday All day	0	0	0	0	0	0	0	0
Tuesday All day	0	0	0	0	0	0	0	0
Wednesday 10:00–11:30	0	0	0	0	0	0	0	0
Wednesday 12:00–13:30	0	1	0	1	0	1	0	1
Thursday 12:00–13:00	0	1	0	1	0	1	0	1
Thursday 13:00–13:30	0	1	1	2	0	1	0	1
Thursday 13:30–16:00	0	0	1	1	0	0	0	0
Maximum	0	1	1	2	0	1	0	1

(d) SE computation. Excess for each class is computed for each time instant assuming each group to be attending classes in person. Group totals are then computed for each time instant, followed by a maximum over these in the bottom row. Maximum of this row is the SE. In this example, SE = 2. Since excess room has capacity 2, we have SSE = 0.

Figure 2.2: An example showing 13 students enrolled in 3 classes along with the weekly schedules and social distancing capacities of these classes. The colors depict one possible assignment of the students to groups: the students in group 1 are indicated in green and the students in group 2 are indicated in white. In this example, we have TE=3, an SE=2, SSE=0, and TD=2.

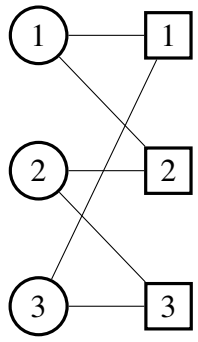


Figure 2.3: Graph showing student enrollments in Example 2.1.

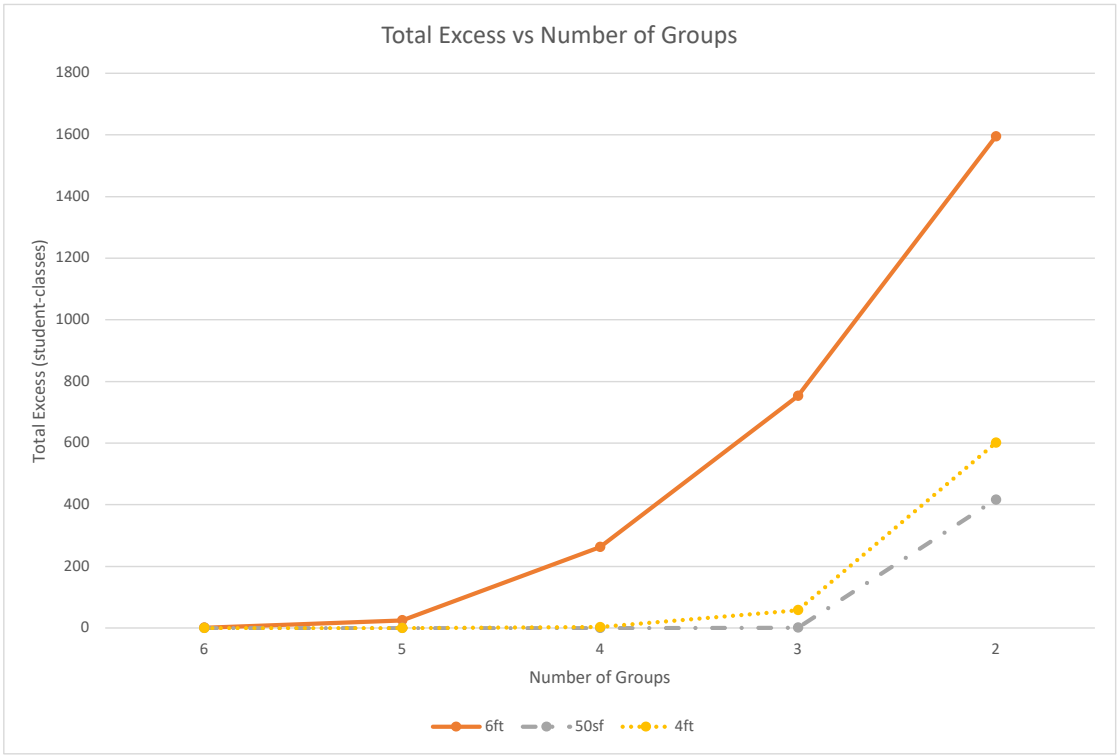


Figure 2.4: Effect of reduction in the number of groups on total excess for different social-distancing protocols. The TE increases with a decreasing number of groups. For the same number of groups, `6ft` social-distancing protocol has the largest TE, followed by the `4ft` social-distancing protocol, followed by the `50sf` social-distancing protocol. In the final implementation, the classroom capacities were somewhere between `6ft` and `50sf`/`4ft` protocols, and the number of groups was set to 3.

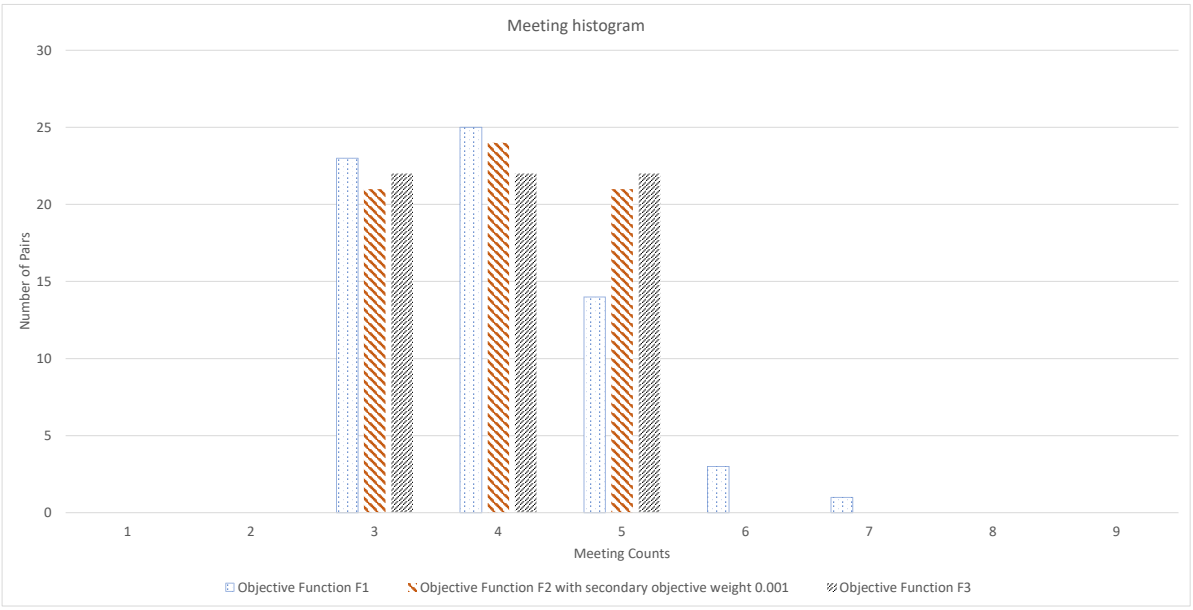


Figure 2.5: Histogram depicting number of meetings between any pair of distinct learning teams when the schedule is evaluated using different objective functions.

Chapter 3: Targeted Intervention in Random Graphs[1]

3.1 Introduction

Individual decision-making in many domains is driven by personal as well as social factors. If one wants to decide a level of time, money, or effort to exert on some task, the behaviors of one's friends or neighbors can be powerful influencing factors. We can view these settings as games where agents in a network are playing some game, each trying to maximize their individual utility as a function of their "standalone value" for action as well as their neighbors' actions. The actions of agents who are "central" in a network can have large ripple effects. Identifying and understanding the role of central agents is of high importance for tasks ranging from microfinance [72] and vaccinations [73], to tracking the spread of information throughout a community [74]. We view our work as providing theoretical support for heuristic approaches to intervention in these settings.

A model for such a setting is studied in recent work by Galeotti, Golub, and Goyal [1], where they ask the natural question of how a third party should "intervene" in the network to maximally improve social welfare at the equilibrium of the game. Interventions are modeled by assuming that the third party can pay some cost to adjust the standalone value parameter of any agent, and must decide how to allocate a fixed budget. This may be interpreted as suggesting that these targeted agents are subjected to advertizing, monetary incentives, or some other form of encouragement. For their model, they provide a general framework for computing the optimal intervention subject to any budget constraint, which can be expressed in terms of the spectral decomposition of the graph. For large budgets, the optimal intervention is approximately proportional to the first eigenvector of the adjacency matrix of the graph, a common measure of network centrality.

[1]This chapter is based on original work in collaboration with William Brown[71].

While this method is optimal, and computable in polynomial time if the adjacency matrix is known, it is rare in practice that we can hope to map all connections in a large network. For physical networks, edges representing personal connections may be far harder to map than simply identifying the set of agents, and for large digital networks we may be bound by computational or data access constraints. However, real-world networks are often well-behaved in that their structure can be approximately described by a simple generating process. If we cannot afford to map an entire network, is optimal targeted intervention feasible at all? A natural target would be to implement interventions which are competitive with the optimal intervention, i.e. obtaining almost the same increase in social welfare, without access to the full adjacency matrix. Under what conditions can we use knowledge of the distribution a graph is drawn from to compute a near-optimal intervention without observing the realization of the graph? Without knowledge of the distribution, how much information about the graph is necessary to find such an intervention? Can we ever reach near-optimality with no information about the graph? These are the questions we address.

3.1.1 Contributions

Our main result shows that for random graphs with independent edges, the first eigenvector of the "expected adjacency matrix", representing the probability of each edge being included in the graph, constitutes a near-optimal intervention simultaneously for almost all generated graphs, when the budget is large enough and the expected matrix satisfies basic spectral conditions. We further explore graphs with given expected degrees, Erdős-Rényi graphs, power law graphs, and stochastic block model graphs as special cases for which our main result holds. In these cases, the first eigenvector of the expected matrix can often be characterized by a simple expression of parameters of the graph distribution.

Yet in general, this approach still assumes a fair amount of knowledge about the distribution, and that we can map agents in the network to their corresponding roles in the distribution. We give several sampling-based methods for approximating the first eigenvector of a graph in each of the

aforementioned special cases, which do not assume knowledge of agent identities or distribution parameters, other than group membership in the stochastic block model. These methods assume different query models for accessing information about the realized graph, such as the ability to query the existence of an edge or to observe a random neighbor of an agent. Using the fact that the graph was drawn from *some* distribution, we can reconstruct an approximation of the first eigenvector more efficiently than we could reconstruct the full matrix. The lower-information settings we consider can be viewed as assumptions about qualitative domain-specific knowledge, such as a degree pattern which approximately follows an (unknown) power law distribution, or the existence of many tight-knit disjoint communities.

We evaluate our results experimentally on both synthetic and real-world networks for a range of parameter regimes. We find that our heuristic interventions can perform quite well compared to the optimal intervention, even at modest budget and network sizes. These results further illustrate the comparative efficacies of interventions requiring varying degrees of graph information under different values for distribution parameters, budget sizes, and degrees of network effects.

On the whole, our results suggest that explicit mapping of the connections in a network is unnecessary to implement near-optimal targeted interventions in strategic settings, and that distributional knowledge or limited queries will often suffice.

3.1.2 Related Work

Recent work by Akbarpour, Malladi, and Saberi [75] has focused on the challenge of overcoming network data barriers in targeted interventions under a diffusion model of social influence. In this setting, for $G(n, p)$ and power law random graphs, they derive bounds on the additional number of "seeds" needed to match optimal targeting when network information is limited. A version of this problem where network information can be purchased is studied in [76]. Another similar model was employed by Candogan, Bimpikis, and Ozdaglar [77] where they study optimal pricing strategies to maximize profit of a monopolist selling service to consumers in a social network where the consumer experiences a positive local network effect, where notions of centrality play a

key role. Similar targeting strategies are considered in [78], where the planner tries to maximize aggregate action in a network with complementarities. [79] studies the efficacy of blind interventions in a pricing game for the special case of Erdős-Rényi graphs. In [80], targeted interventions are also studied for "linear-quadratic games", quite similar to those from [1], in the setting of infinite-population graphons, where a concentration result is given for near-optimal interventions.

Our results can be viewed as qualitatively similar findings to the above results in the model of [1]. While they have showed that exact optimal interventions can be constructed on a graph with full information, we propose that local information is enough to construct an approximately optimal intervention for many distributions of random graphs. It is argued in [81] that collecting data of this kind (aggregate relational data) is easier in real networks compared to obtaining full network information. We make use of concentration inequalities for the spectra of random adjacency matrices; there is a great deal of work studying various spectral properties of random graphs (see e.g. [82, 83, 84, 85, 86]). Particularly relevant to us is [86], which characterizes the asymptotic distributions of various centrality measures for random graphs. There is further relevant literature for studying centrality in graphons, see e.g. [87]. Of relevance to our sampling techniques, a method for estimating eigenvector centrality via sampling is given in [88], and the task of finding a "representative" sample of a graph is discussed in [89].

3.2 Model and Preliminary Results

Here we introduce the "linear-quadratic" network game setting from [1], also studied in e.g. [80], which captures the dynamics of personal and social motivations for action in which we are interested.

3.2.1 Setting

Agents are playing a game on an undirected graph with adjacency matrix A. Each agent takes an action $a_i \in \mathbb{R}$ and obtains individual utility given by:

$$u_i(a, A) = b_i a_i - \frac{1}{2} a_i^2 + \beta \sum_j A_{ij} a_i a_j$$

Here, b_i represents agent i's "standalone marginal value" for action. The parameter β controls the effect of strategic dynamics, where a positive sign promotes complementary behavior with neighbors and a negative sign promotes acting in opposition to one's neighbors. In this paper we focus on the case where each value A_{ij} is in $\{0, 1\}$ and $\beta > 0$. The assumption that $\beta > 0$ corresponds to the case where agents' actions are complementary, meaning that an increase in action by an agent will increase their neighbors' propensities for action. [2] We assume that $b_i \geq 0$ for each agent as well.

The matrix $M = (I - \beta A)$ can be used to determine the best response for each agent given their opponents' actions. The best response vector a^*, given current actions a, can be computed as:

$$a^* = b + \beta A a.$$

Upon solving for $a^* = a$, we get that $a^* = (I - \beta A)^{-1} b = M^{-1} b$, giving us the Nash equilibrium for the game as all agents are simultaneously best responding to each other. We show in Appendix C.2 that when agents begin with null action values, repeated best responses will converge to equilibrium, and further that the new equilibrium is likewise reached after intervention.

Our results will apply to cases where all eigenvalues of M are almost surely positive, ensuring invertibility. [3] The social welfare of the game $W = \sum_i u_i$ can be computed as a function of the

[2] When $\beta < 0$, neighbors' actions act as substitutes, and one obtains less utility when neighbors increase levels of action. In that case, the optimal intervention for large budgets is approximated by the last eigenvector of the graph, which measures its "bipartiteness".

[3] If $\beta > 0$ and M is not invertible, equilibrium actions will be infinite for all agents in some component of the graph.

equilibrium actions:

$$W = \frac{1}{2}(a^*)^\top a^*$$

Given the above assumptions, equilibrium actions a_i^* will always be non-negative.

3.2.2 Targeted Intervention

In this game, a central authority has the ability to modify agents' standalone marginal utilities from b_i to $\hat{b}_i$ by paying a cost of $(b_i - \hat{b}_i)^2$, and their goal is to maximize social welfare subject to a budget constraint C:

$$\max \sum_i u_i \quad \text{subject to} \quad \sum_i (b_i - \hat{b}_i)^2 \leq C.$$

Here, an *intervention* is a vector $y = \hat{b} - b$ such that $\|y\|^2 \leq C$. [4] Let $W(y)$ denote the social welfare at equilibrium following an intervention y. It is shown in [1] that the optimal budget-constrained intervention for any C can be computed using the eigenvectors of A, and that in the large-budget limit as C tends to infinity, the optimal intervention approaches $\sqrt{C} \cdot v_1(A)$. Throughout, we assume $v_i(A)$ is the unit ℓ_2-norm eigenvector associated with λ_i, the ith largest eigenvalue of a matrix A. We also define $\alpha_i = \frac{1}{(1-\beta\lambda_i)^2}$, which is the square of the corresponding eigenvalue of M^{-1}. Note that we do not consider eigenvalues to be ordered by absolute value; this is done to preserve the ordering correspondence between eigenvalues of A and M^{-1}. A may have negative eigenvalues, but all eigenvalues of M^{-1} will be positive when $\beta\lambda_1 < 1$, as we will ensure throughout.

The key result we use from [1] states that when β is positive, as the budget increases the cosine similarity between the optimal intervention y^* and the first eigenvector of a graph, which we denote by $\rho(v_1(A), y^*)$,[5] approaches 1 at a rate depending on the inverted spectral gap of the adjacency

[4] Unless specified otherwise, $\|\cdot\|$ refers to the ℓ_2 norm. When the argument is a matrix, this denotes the associated spectral norm.

[5] The cosine similarity of two non-zero vectors z and y is $\rho(z, y) \triangleq \frac{z \cdot y}{\|z\|\|y\|}$. For unit vectors x, y, by the law of cosines, $\|x - y\|^2 = 2(1 - \rho(x, y))$, and so $1 - \frac{\|x-y\|^2}{2} = \rho(x, y)$. Thus $\|x - y\| < \epsilon$ for $\epsilon > 0$ if and only if

matrix, which is defined as follows.

Definition 3.1 (Inverted Spectral Gap of a Matrix) *Inverted spectral gap of a matrix A is the smallest value κ such that $|\lambda_i(A)| \leq \kappa \cdot |\lambda_1(A)|$, where λ_i is the ith largest eigenvalue, in magnitude, of the matrix A.*

Our results will involve quantifying the *competitive ratio* of an intervention y, which we define as $\frac{W(y)}{W(y^*)}$, where $W(\cdot)$ denotes the social welfare at equilibrium after an intervention vector is applied, and where $y^* = \arg\max_{x \,:\, \|x\|=\sqrt{C}} W(x)$. This ratio is at most 1, and maximizing it will be our objective for evaluating interventions.

3.2.3 Random Graph Models

We introduce several families of random graph distributions which we consider throughout. All of these models generate graphs which are undirected and have edges which are drawn independently.

Definition 3.2 (Random Graphs with Independent Edges) *A distribution of random graphs with independent edges is specified by a symmetric matrix $\overline{A} \in [0,1]^{n \times n}$. A graph is sampled by including each edge (i,j) independently with probability $\overline{A}_{ij}$.*

Graphs with given expected degrees ($G(w)$, or Chung-Lu graphs) and stochastic block model graphs, often used as models of realistic "well-behaved" networks, are notable cases of this model which we will additionally focus on.

Definition 3.3 ($G(w)$ Graphs) *A $G(w)$ graph is an undirected graph with an expected degree sequence given by a vector w, whose length (which we denote by n) defines the number of vertices in the graph. For each pair of vertices i and j with respective expected degrees w_i and w_j, the edge (i,j) is included independently with probability $\overline{A}_{ij} = \frac{w_i w_j}{\sum_{k \in [n]} w_k}$.*

$\rho(x,y) > 1 - \epsilon^2/2$.

Without loss of generality, we impose an ordering on w_i values so that $w_1 \geq w_2 \geq \ldots \geq w_n$. To ensure that each edge probability as described above is in $[0, 1]$, we assume throughout that for all vectors w we have that $w_1 \leq \sqrt{\sum_{k\in[n]} w_k}$.

$G(n, p)$ graphs and power law graphs are well-studied examples of graphs which can be generated by the $G(w)$ model.[6] For $G(n, p)$ graphs, w is a uniform vector where $w_i = np$ for each i. Power law graphs are another notable special case where w is a *power law sequence* $\{w_i\}_{i=1}^n$ such that $w_i = c\,(i + i_0)^{-\frac{1}{\sigma-1}}$ for $\sigma > 2$, some constant $c > 0$, and some integer $i_0 \geq 0$. In such a sequence, the number of elements with value x is asymptotically proportional to $\frac{1}{x^\sigma}$.

Definition 3.4 (Stochastic Block Model Graphs) *A stochastic block model graph with n vertices is undirected and has m groups for some $m \leq n$. Edges are drawn independently according to a matrix $\overline{A}$, and the probability of an edge between two agents depends only on their group membership. For any two groups i and j, there is an edge probability $p_{ij} \in [0, 1]$ such that $\overline{A}_{kl} = p_{ij}$ for any agent k in group i and agent l in group j.* [7]

For each graph model, one can choose to disallow self-loops by setting $\overline{A}_{ii} = 0$ for $1 \leq i \leq n$, as is standard for $G(n, p)$ graphs. Our results will apply to both cases.

3.3 Approximately Optimal Interventions

The main idea behind all of our intervention strategies is to target efforts proportionally to the first eigenvector of the expected adjacency matrix. Here we assume that this eigenvector is known exactly. In Section 3.5, we discuss cases when an approximation of the eigenvector can be computed with zero or minimal information about the graph. Our main theorem for random graphs with independent edges shows conditions under which an intervention proportional to the first eigenvector of the expected matrix $\overline{A}$ is near-optimal.

[6] There are several other well-studied models of graphs with power law degree sequences, such as the BA preferential attachment model, as well as the fixed-degree model involving a random matching of "half-edges". Like the $G(w)$ model, the latter model can support arbitrary degree sequences. We restrict ourselves to the independent edge model described above.

[7] If $m = n$, the stochastic block model can express any distribution of random graphs with independent edges, but will be most interesting when there are few groups.

We define a property for random graphs which we call (ϵ, δ)-*concentration* which will ensure that the expected first eigenvector constitutes a near-optimal intervention. In essence, this is an explicit quantification of the asymptotic properties of "large enough eigenvalues" and "non-vanishing spectral gap" for *sequences* of random graphs from [86]. Intuitively, this captures graphs which are "well-connected" and not too sparse. One can think of the first eigenvalue as a proxy for density, and the (inverse) second eigenvalue as a proxy for regularity or degree of clustering (it is closely related to a graph's mixing time). Both are important in ensuring concentration, and they trade off with each other (via the spectral gap condition) for any fixed values of ϵ and δ.

Definition 3.5 ((ϵ, δ)-Concentration) *A random graph with independent edges specified by $\overline{A}$ satisfies (ϵ, δ)-concentration for $\epsilon, \delta \in (0, 1)$ if:*

1. *The largest expected degree $d_{\max} = \max_i \sum_{j \in [n]} \overline{A}_{ij}$ is at least $\frac{4}{9} \log(2n/\delta)$*
2. *The inverted spectral gap of $\overline{A}$ is at most κ*
3. *The quantity $\lambda_1(\overline{A}) \cdot (1 - \kappa^2)$ is at least $\frac{1024\sqrt{d_{\max} \log(2n/\delta)}}{\epsilon^2}$*

Theorem 3.3.1 *If $\overline{A}$ satisfies (ϵ, δ)-concentration, then with probability at least $1 - \delta$, the competitive ratio of $y = \sqrt{C} v_1(\overline{A})$ for a graph drawn from $\overline{A}$ is at least $1 - \epsilon$ for a sufficiently large budget C if the spectral radius of the sampled matrix A is less than $1/\beta$.*

The concentration conditions are used to show that the relevant spectral properties of generated graphs are almost surely close to their expectations, and the constraint on β is necessary to ensure that actions and utilities are finite at equilibrium.[8] The sufficient budget will depend on the size of the *spectral gap* of $\overline{A}$, as well as the standard marginal values. For example, if $\lambda_1 > 2|\lambda_i|$ holds in the realized graph for all $i > 1$, then a budget of $C = 256 \cdot \|b\|^2 / (\epsilon \beta \lambda_1(\overline{A}))^2$ will suffice. Intuitively, a large β would mean more correlation between neighbors' actions at equilibrium. A large $\lambda_1\left(\overline{A}\right)$ would mean a denser graph (more connections between agents) in expectation and

[8] The spectral radius condition holds with probability $1 - \delta$ when $1/\beta$ is at least $\lambda_1(\overline{A}) + \sqrt{4 d_{\max} \log(2n/\delta)}$ (follows from e.g. [83], see Section C.3.2 for details).

a large ϵ would mean that the realized graph is more likely to be close to expectation. All of these conditions reduce the required budget because a small intervention gets magnified by agent interaction. Further, the smaller the magnitude of initial b, the easier it is to change its direction.

The proof of Theorem 3.3.1, along with proofs for the lemmas stated in this work, is deferred to Appendix C.1, At a high level, our results proceed by first showing that the first eigenvector is almost surely close to $v_1(\overline{A})$, then showing that the spectral gap is almost surely large enough such that the first eigenvector is close to the optimal intervention for appropriate budgets. A key lemma for completing the proof shows that interventions which are close to the optimal intervention in cosine similarity have a competitive ratio close to 1.

Lemma 3.3.1 *Let b be the vector of standalone values, and assume that $C > \max(\|b\|^2, 1)$. For any y where $\|y\|^2 = C$ and $\rho(y, y^*) > \gamma$ for some γ, the competitive ratio of y is at least $1 - 4\sqrt{2(1-\gamma)}$.*

The main idea behind this lemma is a smoothness argument for the welfare function. When considering interventions as points on the sphere of radius $\sqrt{C}$, small changes to an intervention cannot change the resulting welfare by too much. This additionally implies that when a vector y is close to y^*, the exact utility of y^* for some budget C can be achieved by an intervention proportional to y with a budget C' which is not much larger than C.

In Section 3.4 we give a specialization of Theorem 3.3.1 to the case of $G(w)$ graphs. There, the expected first eigenvector is proportional to w when self-loops are not removed. We give more explicit characterizations of the properties for $G(w)$, $G(n, p)$, and power law graphs which ensure the above spectral conditions (i.e. without relying on eigenvalues), as well as a budget threshold for near-optimality. We discuss the steps of the proof in greater detail, and they are largely symmetric to the steps required to prove Theorem 3.3.1.

3.4 Graphs with Given Expected Degrees

In this section, we show a method for obtaining near-optimal interventions in graphs generated by the $G(w)$ model. We show that the first eigenvector of a $G(w)$ graph is almost surely close to w, and our intervention will simply be proportional to w. This indicates that degree estimates are often sufficient for near-optimal intervention. We assume that w is sorted in descending order and that each entry is strictly positive. Our main theorem for this section holds for all $G(w)$ distributions which satifsy the following specialization of (ϵ, δ)-concentration. The second condition corresponds to requiring a sufficiently large first eigenvalue, as was the case for general random graphs; $G(w)$ graphs do not exhibit clustering on average, and so we do not need an additional condition for the second eigenvalue.

Definition 3.6 ((ϵ, δ)-Concentration for $G(w)$ Graphs) *A $G(w)$ graph satisfies (ϵ, δ)-concentration for $\epsilon, \delta \in (0, 1)$ if:*

1. *The largest expected degree w_1 is at least $\frac{4}{9}\log(2n/\delta)$ and at most $\frac{\|w\|}{6}$*
2. *The second-order average of the expected degree sequence $\tilde{d} = \frac{\sum_i w_i^2}{\sum_i w_i}$ is at least $\frac{256\left(\sqrt{4w_1 \log(2n/\delta)}+1\right)}{\epsilon^2}$*

Theorem 3.4.1 *For $G(w)$ distributions satisfying (ϵ, δ)-concentration, and for C at least $\frac{256\|b\|^2}{(\epsilon\beta\tilde{d})^2}$, with probability at least $1 - \delta$,*

$$\frac{W\left(\sqrt{C} \cdot \frac{w}{\|w\|}\right)}{W(y^*)} \geq 1 - \epsilon,$$

where b is the vector of standalone marginal values and y^* is the optimal intervention for a budget C, if the spectral radius of A is less than $1/\beta$.[9]

Our conditions ensure that the first eigenvalue of the graph is not too small, and that the other eigenvalues are not too large in magnitude. It is worth noting that $\frac{1}{\beta\tilde{d}}$ is small unless the network

[9]The spectral radius of βA is less than 1 with probability $1 - \delta$ when $\frac{1}{\beta} < \frac{1}{\tilde{d}+\sqrt{4w_1 \log(2n/\delta)}}$.

effects in the game are negligible, in which case we should not expect eigenvector centrality to be important for small budgets. In Section 3.4.3, we consider applications to $G(n, p)$ and power law graphs. Here we assume the vector w which parameterizes the graph distribution is known, and in fact our intervention will simply be proportional to w.

In the proof of Theorem 3.4.1, we proceed by observing that the first eigenvector of $\overline{A}(n)$ is proportional to w, and then prove that when the spectral conditions hold, the first eigenvector of $A(n)$ is nearly proportional to w with high probability. We then determine sufficient budget sizes such that near-optimality of the intervention follows from Lemma 3.3.1.

3.4.1 Proportionality of Eigenvector Centrality and Degree

Here we show that the first eigenvector of the adjacency matrix A is almost surely close to the unit vector rescaling of w. We first observe that this holds in the standard version of the $G(w)$ model which allows for self-loops, and then we show that the first eigenvector does not change by much upon pruning loops.

Lemma 3.4.1 *For $G(w)$ graph distributions, any eigenvector of $\overline{A}$ corresponding to a non-zero eigenvalue is proportional to w.*

It can be checked that up to scaling, w is the unique vector which satisfies the eigenvalue equation for a non-zero eigenvalue. As all rows and columns of $\overline{A}$ are proportional to w, there is only one non-zero eigenvalue. From the eigenvalue equation it is simple to check that the non-zero eigenvalue $\lambda_1(\overline{A})$ will be equal to the second-order average degree $\frac{\sum_i w_i^2}{\sum_i w_i}$.

In this formulation of the $G(w)$ model, agents are allowed to have self-loops with positive probability. We note now that even if we remove the possibility of self-loops by setting the diagonal entries of $\overline{A}$ to 0, which in turn will remove the rank-deficiency, the spectral norm between the expected matrix (with loops) and the realized matrix (with or without loops) will be small. This in turn will imply that the first eigenvectors of these matrices are close.

A key tool in this proof is a bound on the difference in first eigenvectors of matrices which are close in norm when one of them has a small second eigenvalue. We only make use of this in the

case where the second eigenvalue of one matrix is zero, but a more general version of the result, used to prove Theorem 3.3.1, is included in Appendix C.1.

Lemma 3.4.2 *Let A be a symmetric $n \times n$ matrix with largest absolute eigenvalue λ_1 (with multiplicity 1) and all other eigenvalues equal to 0. Let B be a symmetric $n \times n$ matrix with largest absolute eigenvalue μ_1, and suppose $\|A - B\| \leq \eta$. Then,*

$$\|v_1(A) - v_1(B)\| \leq \sqrt{2\left(1 - \frac{\mu_1 - \eta}{\lambda_1}\right)}.$$

This follows from considering a decomposition of the first eigenvector of B into a component proportional to $v_1(A)$ and one proportional to some vector orthogonal to $v_1(A)$. Given that $A - B$ has a small spectral norm, the image of $v_1(B)$ in A will be large, showing that the orthogonal component is small, which we can use to show that the eigenvectors are close.

We can then show that norm difference of the expected and realized matrices is almost surely small using a matrix concentration bound from [83], allowing us to apply Lemma 3.4.2 to bound the difference in their eigenvectors, as the second eigenvalue of $\overline{A}$ is 0.

Lemma 3.4.3 *If the above assumptions about the $G(w)$ distribution are satisfied, then with probability $1 - \delta$ it holds that:*

$$\left\| v_1(A) - \frac{w}{\|w\|} \right\| \leq \epsilon/8.$$

As the first eigenvector of $\overline{A}$ is proportional to w, this shows that that the first eigenvector will be nearly proportional to w regardless of whether we allow self-loops, and so our intervention will be close to the true graph's first eigenvector. Next, we will see that this implies near-optimality for a sufficiently large budget.

3.4.2 Bounding Suboptimality of Interventions

The previous results indicate that the first eigenvector will be close to its expectation when our assumptions hold, even upon removing self-loops. We now give a similar results for the first and second eigenvalues, which allows us to guarantee a sufficiently large spectral gap for M^{-1}. First, we give a bound on the second eigenvalue of $\overline{A}$ with the diagonal removed.

Lemma 3.4.4 *Let D be the matrix which is equal to $\overline{A}$ along the diagonal and 0 elsewhere, and let λ_2 denote the second-largest absolute eigenvalue of $\overline{A} - D$. Then,*

$$\lambda_2 \leq \frac{2w_1\lambda_1}{\|w\|} + 1.$$

This follows from a similar orthogonal decomposition approach to the proof of Lemma 3.4.2, as well as another direct application of Lemma 3.4.2. We can then get an absolute bound on the first and second eigenvalues of A.

Lemma 3.4.5 *With probability at least $1 - \delta$, the second eigenvalue of A is at most*

$$\frac{2w_1\tilde{d}}{\|w\|} + 1 + \sqrt{4w_1 \log(2n/\delta)}$$

and the first eigenvalue of A is at least

$$\tilde{d} - 1 - \sqrt{4w_1 \log(2n/\delta)}.$$

This follows from applying the triangle inequality to Lemma 3.4.4 and Theorem 1 from [83] as well as from our observation about the first expected eigenvalue of $\overline{A}$. To complete the proof of Theorem 3.4.1, we can combine the previous results to show that $\lambda_1 > 2\lambda_2$ when the stated conditions hold. Proposition 2 from [1] allows us to use this fact to show that when budget is above our lower bound, the first eigenvector is close to the optimal intervention in cosine similarity. We can then show that w and the optimal intervention are close in cosine similiarity, using Lemma 3.4.3

as a key step. Plugging this into Lemma 3.3.1 gives us the theorem.

3.4.3 Examples: $G(n, p)$ and Power Law Graphs

$G(n, p)$ graphs are perhaps the most well-studied family of random graphs, which we can interpret as a special case of the $G(w)$ model and give explicit conditions for when (ϵ, δ)-concentration holds. These conditions are lower bounds on n and p which guarantee the requirements for applying Theorem 3.4.1, and for clarity we restate the eigenvector similarity and near-optimality results for the case of $G(n, p)$ graphs.

Lemma 3.4.6 *For $G(n, p)$ graphs with $p \geq \frac{4\log(2n/\delta)}{9(n-1)}$ and n at least $\Omega(1/\epsilon^4)$,*

$$\left\| v_1(A) - \frac{1}{\sqrt{n}} \right\| \leq \epsilon/8.$$

Theorem 3.4.2 *For $G(n, p)$ graphs with $p \geq \frac{4\log(2n/\delta)}{9n-1}$ and n at least $\Omega(1/\epsilon^4)$, with probability at least $1 - \delta$, the uniform intervention $y = \frac{\sqrt{C}}{\sqrt{n}} \cdot \mathbf{1}$ achieves utility within $1 - \epsilon$ of the optimal intervention for budgets C at least $\frac{256\|b\|^2}{(\epsilon\beta np)^2}$.*

The constant factor for the lower bound on n we obtain in the proof of Lemma 3.4.6 is large, but can likely be optimized, and our empirical results in Section 3.6 indicate the uniform intervention is close to optimal on reasonably small $G(n, p)$ graphs.

This theorem also applies directly to stochastic block model graphs where all blocks are the same size and each has the same ingroup and outgroup probabilities. Here, all agents are equally central in expectation, and it is simple to check that the first eigenvector of $\overline{A}$ will be uniform. This allows a near-optimal intervention for sufficiently large and dense graphs without any knowledge of the edges, connectivity probabilities, group memberships, or degrees.

We can also show that our approach is near-optimal for many power law graphs, as introduced in Section 3.2. Power law graphs are a notable special case of the $G(w)$ model, and are often

studied as models of real-world networks. Our results hold for power law graphs where $\sigma \in (2, 2.5)$, a range containing many observed examples [90, 91, 85].

Lemma 3.4.7 *For power law graphs with $2 < \sigma < 2.5$, if we have that w_1 is at least $\Omega\left(\left(\frac{\log(2n/\delta)}{\epsilon^4}\right)^{\frac{1}{5-2\sigma}}\right)$, then with probability at least $1 - \delta$,*

$$\left\| v_1(A) - \frac{w}{\|w\|} \right\| \leq \epsilon/8.$$

Theorem 3.4.3 *For power law graphs with $2 < \sigma < 2.5$, if we have that w_1 is at least $\Omega\left(\left(\frac{\log(2n/\delta)}{\epsilon^4}\right)^{\frac{1}{5-2\sigma}}\right)$, if β is small enough to ensure that $\beta\lambda_1(A) < 1$, then with probaility at least $1 - \delta$, the intervention $y = \sqrt{C} \cdot \frac{w}{\|w\|_1}$ achieves utility within a $1 - \epsilon$ factor of the optimal intervention for budgets C at least $\frac{256\|b\|^2}{\left(\epsilon\beta\lambda_1(\overline{A})\right)^2}$, where $\lambda_1(\overline{A}) = \Theta(w_1^{3-\sigma})$.*

3.5 Centrality Estimation

The previous sections show that interventions proportional to $v_1(\overline{A})$ are often near-optimal simultaneously for almost all graphs generated by $\overline{A}$. While we often may have domain knowledge about a network which helps characterize its edge distribution, we still may not be able to precisely approximate the first eigenvector of $\overline{A}$ *a priori*. In particular, even if we believe our graph comes from a power law distribution, we may be at a loss in knowing which vertices have which expected degrees.

In this section, we discuss approaches for obtaining near-optimal interventions without initial knowledge of $\overline{A}$. We first observe that "blind" interventions, which treat all vertices equally in expectation, will fail to approach optimality. We then consider statistical estimation techniques for approximating the first eigenvector which leverage the special structure of $G(w)$ and stochastic block model graphs. In each case, we identify a simple *target intervention*, computable directly from the realized graph, which is near-optimal when (ϵ, δ)-concentration is satisfied. We then give efficient sampling methods for approximating these target interventions. Throughout this section,

our focus is to give a broad overview of these techniques rather than to present them as concrete algorithms, and we frequently omit constant-factor terms with asymptotic notation.

3.5.1 Suboptimality of Blind Interventions

Here we begin by showing that when the spectral gap is large, all interventions which are far from the optimal intervention in cosine similarity will fail to be near-optimal even if the budget is very large.

Lemma 3.5.1 *Assume that C is sufficiently large such that the role of standalone values is negligible. For any y where $\|y\|^2 = C$ and $\rho(y, y^*) < \gamma$, the competitive ratio is bounded by*

$$\gamma^2 \left(1 - \frac{\alpha_2}{\alpha_1}\right) + \frac{\alpha_2}{\alpha_1} + 2\sqrt{\frac{\alpha_2}{\alpha_1}},$$

where α_i is the square of the ith largest eigenvalue of M^{-1}.

This tells us that if one were to design an intervention without using any information about the underlying graph, the intervention is unlikely to do well compared to the optimal one for the same budget unless eigenvector centrality is uniform, as in the case of $G(n, p)$ graphs. Thus, there is a need to try to learn graph information to design a close-to-optimal intervention. We discuss methods for this next.

3.5.2 Degree Estimation in $G(w)$ Graphs

For $G(w)$ graphs, we have seen that expected degrees suffice for near-optimal interventions, and we show that degrees can suffice as well.

Lemma 3.5.2 *If a $G(w)$ graph specified by $\overline{A}$ satisfies (ϵ, δ)-concentration, then with probability at least $1 - O(\delta)$,*

$$\|w - w^*\| \leq O(\epsilon \|w\|),$$

where w^* is the empirical degree vector, and the intervention proportional to w^* obtains a competitive ratio of $1 - O(\epsilon)$ when the other conditions for Theorem 3.3.1 are satisfied.

Thus, degree estimation is our primary objective in considering statistical approaches. As we can see from the analysis in Theorems 3.3.1 and 3.4.1, if we can estimate the unit-normalized degree vector w^* to within ϵ ℓ_2-distance, our competitive ratio for the corresponding proportional intervention will be $1 - O(\epsilon)$. Our approaches focus on different query models, respresenting the types of questions we are allowed to ask about the graph; these query models are also studied for the problem of estimating the average degree in a graph [92, 93]. If we are allowed to query agents' degrees, near-optimality follows directly from the above lemma, so we consider more limited models.

Edge Queries.

Suppose we are allowed to query whether an edge exists between two vertices. We can then reduce the task of degree estimation to the problem of estimating the mean of n biased coins, where for each vertex, we "flip" the corresponding coin by picking another vertex uniformly at random to query. By Hoeffding and union bounds, $O\left(\frac{n}{\epsilon^2}\log\left(\frac{n}{\delta}\right)\right)$ total queries suffice to ensure that with probability $1 - \delta$, each degree estimate is within ϵn additive error. Particularly in the case of dense graphs, and when ϵ is not too small compared to $1/n$, this will be considerably more efficient than reconstructing the entire adjacency matrix. In particular, if $\|w\|_1 = \Theta(n^2)$, the above error bound on additive error for each degree estimate directly implies that the estimated degree vector $\hat{w}$ is within ℓ_1 (and thus ℓ_2) distance of $O(\epsilon \|w\|_2)$.

Random Neighbor Queries.

Suppose instead we are restricted to queries which give us a uniformly random neighbor of a vertex. We give an approach wherein queries are used to conduct a random walk in the graph. The stationary distribution is equivalent to the the first eigenvector of the *diffusion matrix* P =

AD^{-1}, where D is the diagonal matrix of degree counts.[10] We can then learn estimates of degree proportions by sampling from the stationary distribution via a random walk.

The mixing time of a random walk on a graph determines the number of steps required such that the probability distribution over states is close to the stationary distribution in total variation distance. We can see that for $G(w)$ graphs satisfying (ϵ, δ)-concentration with a large enough minimum degree, mixing times will indeed be fast.

Lemma 3.5.3 *For $G(w)$ graphs satisfying (ϵ, δ)-concentration and with $w_n \geq \frac{1}{\epsilon}$, the mixing time of a random walk to within ϵ total variation distance to the stationary distribution is $O(\log(n/\epsilon))$. Further, the largest connected component in A contains $n(1 - \exp(-O(1/\epsilon))$ vertices in expectation.*

If a random walk on our graph has some mixing time t to an approximation of the stationary distribution, we can simply record our location after every t steps to generate a sample. Using standard results on learning discrete distributions (see e.g. [94]), $O\left(\frac{n+\log(1/\delta)}{\epsilon^2}\right)$ samples from ϵ-approximate stationary distributions suffice to approximate w^* within ℓ_1 distance of $O(\epsilon \|w^*\|)$ with probability $1 - \delta$, directly giving us the desired ℓ_2 bound. Joining this with Lemma 3.5.3, our random walk takes a total of $O\left(\frac{n+\log(1/\delta)}{\epsilon^2} \log\left(\frac{n}{\epsilon}\right)\right)$ steps (and thus queries) to obtain our target intervention, starting from an arbitrary vertex in the largest connected component.

3.5.3 Matrix Reconstruction in SBM Graphs

There is a fair amount of literature on estimation techniques for stochastic block model graphs, often focused on cases where group membership is unknown [95, 96, 97, 98]. The estimation of eigenvectors is discussed in [99], where they consider stochastic block model graphs as a limit of a convergent sequence of "graphons". Our interest is primarily in recovering eigenvector centrality efficiently from sampling, and we will make the simplifying assumption that group labels are

[10] The stationary distribution of a random walk on a simple connected graph is $\frac{d_i}{\sum_j d_j}$ for all vertices i, where d_i is the degree. While $G(w)$ graphs may fail to be connected, in many cases the vast majority of vertices will belong to a single component, and we can focus exclusively on that component. We show this in Lemma 3.5.3.

visible for all vertices. This is reasonable in many cases where a close proxy of one's primary group identifier (e.g. location, job, field of study) is visible but connections are harder to map.

In contrast to the $G(w)$ case, degree estimates no longer suffice for estimating the first eigenvector. We assume that there are m groups and that we know each agent's group. Our aim will be to estimate the relative densities of connection between groups. When there are not too many groups, the parameters of a stochastic block model graph can be estimated efficiently with either edge queries or random neighbor queries, From here, we can construct an approximation of $\overline{A}$ and compute its first eigenvector directly. In many cases, the corresponding intervention is near-optimal.

A key lemma in our analysis shows that the "empirical block matrix" is close to its expectation in spectral norm. We prove this for the case where all groups are of similar sizes, but the approach can be generalized to cover any partition.

Lemma 3.5.4 *For a stochastic block model graph generated by $\overline{A}$ with m groups, each of size $O(\frac{n}{m})$, let $\hat{A}$ denote the empirical block matrix of edge frequencies for each group. Each entry per block in $\hat{A}$ will contain the number of edges in that block divided by the size of the block. With probability at least $1-\delta$,*

$$\left\|\overline{A}-\hat{A}\right\| \leq O\left(\max\left(\frac{m\sqrt{\log(n/\delta)}}{\sqrt{n}},\ \log^2(n/\delta)\right)\right).$$

The same bound will then apply to the difference of the first eigenvectors, rescaled by the first eigenvalues (which will also be close). Similar bounds can also be obtained when group sizes may vary, but we stick to this case for simplicity.

Edge Queries.

If we are allowed to use edge queries, we can estimate the empirical edge frequency for each of the $O(m^2)$ pairs of groups by repeatedly sampling a vertex uniformly from each group and querying for an edge. This allows reconstruction of the empirical frequencies up to ϵ error for each

group pair, with probability $1 - \delta$, with $O\left(\frac{m^2}{\epsilon^2}\log(m/\delta)\right)$ samples. For the block matrix $\hat{A}$ of edge frequencies for all group pairs, Lemma 3.5.4 implies that this will be close to its expectation when there are not too many groups, and so our estimation will be close to $\overline{A}$ in spectral norm as well. If $\overline{A}$ satisfies (ϵ, δ)-concentration and the bound from Lemma 3.5.4 is small compared to the norm of $\overline{A}$, then the first eigenvectors of A, $\overline{A}$, and $\hat{A}$ will all be close, and the corresponding intervention proportional to $v_1(\hat{A})$ will be near-optimal.

When all group pairs may have unique probabilities, this will only provide an advantage over a naive graph reconstruction with $O(m^2)$ queries in the case where $m = o(n)$. If we know that all out-group probabilities are the same across groups, our dependence on m becomes linear, as we can treat all pairs of distinct groups as one large group. If in-group probabilities are the same across groups as well, the dependence on m vanishes, as we only have two probabilities to estimate.

Random Neighbor Queries.

We can also estimate the empirical group frequency matrix with random neighbor queries. For each group, the row in $\overline{A}$ corresponding to the edge probabilities with other groups can be interpreted as a distribution of frequencies for each group. $O(\frac{m}{\epsilon^2}\log(\frac{m}{\delta}))$ samples per row suffice to get additive error at most ϵ for all of the relative connection probabilities for our chosen group. This lets us estimate each of the m rows up to scaling, at which point we can use the symmetry of the matrix to recover an estimate of $\overline{A}$ up to scaling by some factor. Again, when (ϵ, δ)-concentration holds and the bound from Lemma 3.5.4 is small, the first eigenvector of this estimated matrix will give us a near-optimal intervention.

3.6 Experiments

Our theoretical results require graphs to be relatively large in order for the obtained bounds to be nontrivial. It is natural to ask how well the heuristic interventions we describe will perform on relatively small random graphs, as well as on real-world graphs which do not come from a simple generative model (and may not have independent edges). In Section 3.6.1, we evaluate our

described interventions on real and synthetic network data, by adjusting b_i values and computing the resulting welfare at equilibrium, and find that performance can be quite good even on small graphs. Section 3.6.2, we present experimental results on synthetic networks.

3.6.1 Real Networks

To test the usefulness of our results for real-world networks which we expect to be "well-behaved" according to our requirements, we simulate the intervention process using network data collected from villages in South India, for purposes of targeted microfinance deployments, from [72]. In this context, we can view actions a_i as indicating levels of economic activity, which we wish to stimulate by increasing individual propensities for spending and creating network effects. The dataset contains many graphs for each village using different edge sets (each representing different kinds of social connections), as well as graphs where nodes are households rather than individuals. We use the household dataset containing the union of all edge sets. These graphs have degree counts ranging from 77 to 365, and our experiments are averaged over 20 graphs from this dataset. We plot competitive ratios while varying C (scaled by network size) and the spectral radius of βA, fixing $b_i = 1$ for each agent.

The expected degree intervention is replaced by an intervention proportional to exact degree. We also fit a stochastic block model to graphs using a version of the approach described in Section 3.5.3, using exact connectivity probabilities rather than sampling. Our group labels are obtained by running the Girvan-Newman clustering algorithm [100] on the graph, pruning edges until there are either at least 10 clusters with 5 or more vertices or 50 clusters total. We evaluate the intervention proportional to the first eigenvector of the reconstructed block matrix. All interventions are compared to a baseline, where no change is applied to b, for demonstrating the relative degree in social welfare change.

In Figure 3.1, we find that degree interventions perform quite well, and are only slightly surpassed by first eigenvector interventions. The stochastic block model approach performs better than uniform when the spectral radius sufficiently large, but is still outperformed by the degree

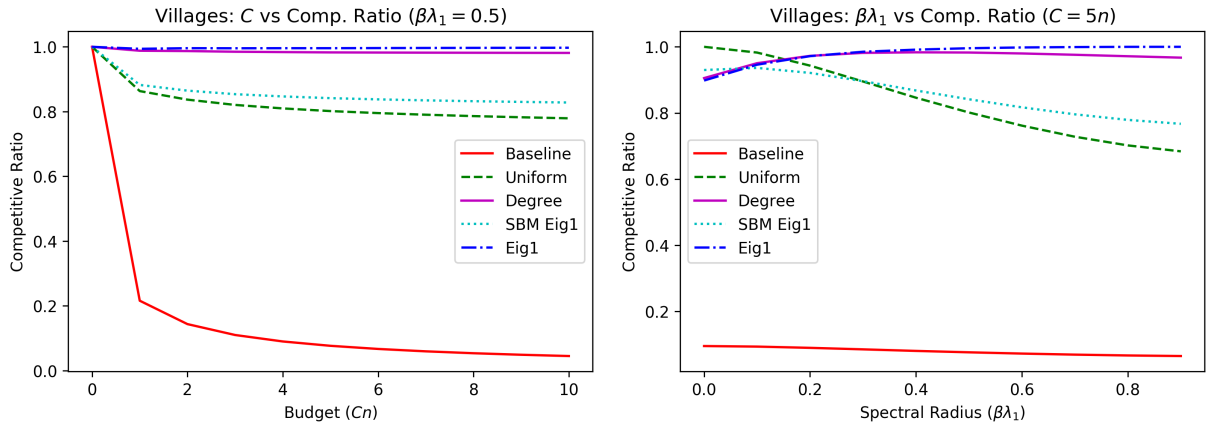


Figure 3.1: Performance of Different Intervention Schemes in Village Graphs

and first eigenvector interventions. Upon inspection, the end result of the stochastic block model intervention was often uniform across a large subgraph, with little or no targeting for other vertices, which may be an artifact of the clustering method used for group assignment. On the whole, we observe that minimal-information approaches can indeed perform quite well on both real and simulated networks.

3.6.2 Random Networks

We evaluate our results in simulated $G(n, p)$ and power law graphs, analyzing the competitive ratio; we compare a baseline (no intervention), expected degree interventions, the first eigenvector intervention (computed from the realized graph), and the optimal intervention for each graph (computed via quadratic programming).

For all experiments, we fix $n = 100$ and $b_i = 1$ for each agent. For both graph families, we experiment by independently varying the budget size C, a distribution parameter (p or σ), and the spectral radius of βA. We plot the competitive ratio of each heuristic intervention with the optimal intervention as parameters are varied, and each parameter specification is averaged over 10 graph samples. We generate power law graphs according to the $G(w)$ model with a power law sequence, where the maximum expected degree is fixed at 25 and the minimum is fixed at 1 for all exponent values.

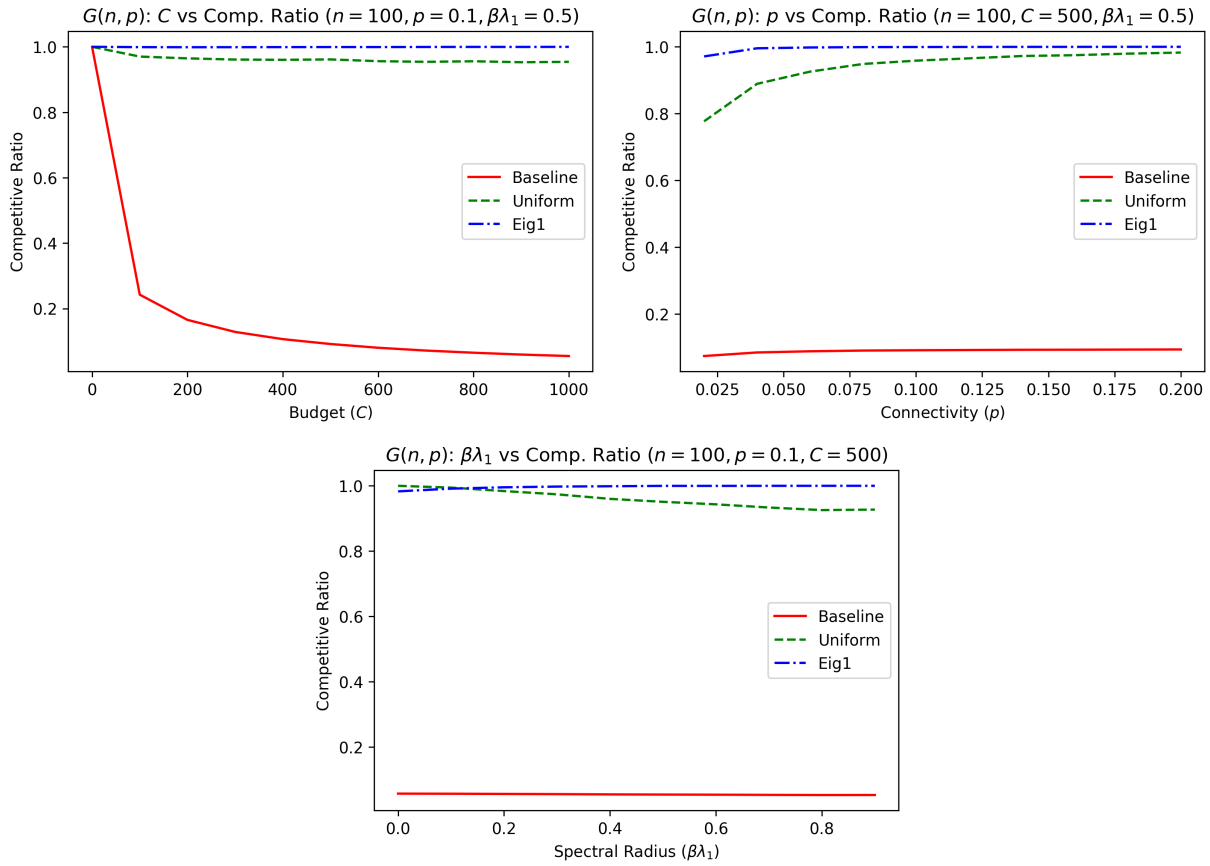


Figure 3.2: Performance of different intervention schemes in $G(n, p)$ graphs

In $G(n, p)$ graphs (see Figure 3.2), we see that the first eigenvector intervention is close to optimal in almost all cases, and that the uniform intervention is quite competitive as well, particularly when the graph is increasingly dense. When the spectral radius is small, the uniform outperforms the first eigenvector intervention as expected, it is optimal when β is 0. The small baseline values at most points indicate that the change to social welfare from our interventions is indeed quite drastic, even when $C = \|b\|^2$.

In power law graphs (see Figure 3.3), the uniform intervention does not perform nearly as well unless the spectral radius is small, where it outperforms other approaches. The expected degree intervention does considerably better in general. When $\sigma > 2.5$, where our theoretical results do not hold, we see that heuristics still perform well. We expect that this is an artifact of the fixed expected degree bounds and the small graph size. In large power law graphs, larger

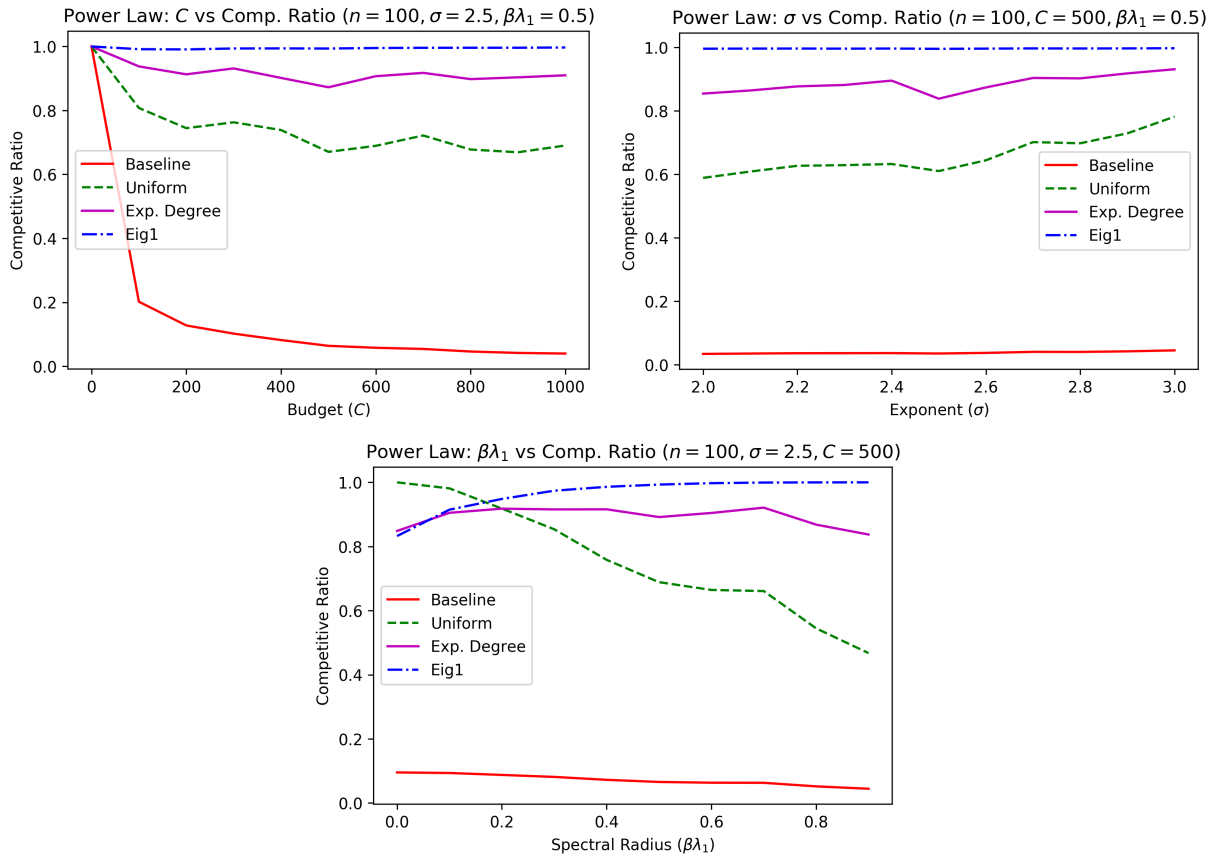


Figure 3.3: Performance of different intervention schemes in power law graphs

exponents correspond to the graph having a smaller "core" of dense connectivity, which can be quite important for influencing the rest of the network. Our experiments suggest that smaller cores are less important in graphs of this size, which more closely resemble sparse $G(n, p)$ graphs with a few well-connected vertices.

Acknowledgments.

We thank Ben Golub, Yash Kanoria, Tim Roughgarden, Christos Papadimitriou, and anonymous reviewers for their invaluable feedback.

References

[1] A. Galeotti, B. Golub, and S. Goyal, "Targeting interventions in networks," *Econometrica*, vol. 88, no. 6, pp. 2445–2471, 2020. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.3982/ECTA16173.

[2] C. C. Moallemi and U. Patange, "An analysis of fixed-spread liquidation lending in defi," Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4803547. Last Accessed: Aug 14, 2024, Proceedings of the 4th Workshop on Decentralized Finance (DeFi) in Association with Financial Cryptography 2024, (Forthcoming).

[3] L. Husney, "Mango markets madness: A case study on the mango markets exploit," *Infotrend*, Mar. 3, 2023, Accessed: 2023-06-07.

[4] M. Bartoletti, J. H. yu Chiang, and A. L. Lafuente, "SoK: Lending Pools in Decentralized Finance," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12676 LNCS, Springer Science and Business Media Deutschland GmbH, 2021, pp. 553–578, ISBN: 9783662639573. arXiv: 2012.13230.

[5] K. Qin, L. Zhou, P. Gamito, P. Jovanovic, and A. Gervais, "An empirical study of DeFi liquidations: Incentives, risks, and instabilities," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 2021, pp. 336–350, ISBN: 9781450391290. arXiv: 2106.06389.

[6] A. Lehar and C. A. Parlour, "Systemic Fragility in Decentralized Markets," *SSRN Electronic Journal*, vol. 2022, 2022.

[7] J. Warmuz, A. Chaudhary, and D. Pinna, *Toxic Liquidation Spirals*, 2022. arXiv: 2212.07306.

[8] K. Qin, J. Ernstberger, and L. Zhou, "Mitigating Decentralized Finance Liquidations with Reversible Call Options," in *Proceedings of Financial Cryptography and Data Security*, 2023.

[9] L. Zhou *et al.*, "SoK: Decentralized Finance (DeFi) Attacks," in *Proceedings - IEEE Symposium on Security and Privacy*, vol. 2023-May, 2023, pp. 2444–2461, ISBN: 9781665493369. arXiv: 2208.13035.

[10] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais, "The Decentralized Financial Crisis," in *Proceedings - 2020 Crypto Valley Conference on Blockchain Technology,*

CVCBT 2020, Institute of Electrical and Electronics Engineers Inc., 2020, pp. 1–15, ISBN: 9781728193908. arXiv: 2002.08099.

[11] L. J. F. Gudgeon, *On the brink of a second financial system: modelling and mitigating risk in decentralised finance*, 2023.

[12] S. N. Cohen, M. Sabate-Vidales, L. Szpruch, and M. Gontier Delaunay, "The Paradox of Adversarial Liquidation in Decentralised Lending," *SSRN Electronic Journal*, 2023.

[13] H.-T. Kao, T. Chitra, R. Chiang, and J. Morrow Gauntlet, *An Analysis of the Market Risk to Participants in the Compound Protocol*, 2019.

[14] L. Gudgeon, S. Werner, D. Perez, and W. J. Knottenbelt, "DeFi Protocols for Loanable Funds: Interest Rates, Liquidity and Market Efficiency," in *AFT 2020 - Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, Association for Computing Machinery, Inc, 2020, pp. 92–112, ISBN: 9781450381390. arXiv: 2006.13922.

[15] M. Bartoletti, J. Chiang, T. Junttila, A. Lluch Lafuente, M. Mirelli, and A. Vandin, "Formal Analysis of Lending Pools in Decentralized Finance," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13703 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 335–355, ISBN: 9783031197581. arXiv: 2206.01333.

[16] K. Saengchote, "Decentralized lending and its users: Insights from compound," *Journal of International Financial Markets, Institutions and Money*, vol. 87, 2023.

[17] S. Yang and W. Cui, *An evaluation system for defi lending protocols*, 2023. arXiv: 2303.01022 [cs.SI].

[18] A. Klages-Mundt and A. Minca, "(In)Stability for the Blockchain: Deleveraging Spirals and Stablecoin Attacks," *Cryptoeconomic Systems*, 2021. arXiv: 1906.02152.

[19] A. Klages-Mundt and A. Minca, "While stability lasts: A stochastic model of noncustodial stablecoins," *Mathematical Finance*, vol. 32, no. 4, pp. 943–981, 2022. arXiv: 2004.01304.

[20] T. Chitra, G. Angeris, A. Evans, and H. T. Kao, "A Note On Borrowing Constant Function Market Maker Shares," in *DeFi 2022 - Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security, co-located with CCS 2022*, Association for Computing Machinery, Inc, 2022, pp. 55–61, ISBN: 9781450385404.

[21] J. Milionis, C. C. Moallemi, T. Roughgarden, and A. L. Zhang, "Quantifying loss in automated market makers," in *Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security*, ser. DeFi'22, Los Angeles, CA, USA: Association for Computing Machinery, 2022, 71–74, ISBN: 9781450398824.

[22] C. McMenamin, V. Daza, and B. Mazorra, "An automated market maker minimizing loss-versus-rebalancing," in *Mathematical Research for Blockchain Economy*, P. Pardalos, I. Kotsireas, W. J. Knottenbelt, and S. Leonardos, Eds., Cham: Springer Nature Switzerland, 2023, pp. 95–114, ISBN: 978-3-031-48731-6.

[23] L. Heimbach, E. Schertenleib, and R. Wattenhofer, "Risks and returns of uniswap v3 liquidity providers," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, ser. AFT '22, Cambridge, MA, USA: Association for Computing Machinery, 2023, 89–101, ISBN: 9781450398619.

[24] AAVE, *Aave v2 lending pool*, last accessed: 2023-12-08.

[25] US Department of Treasury, *Daily treasury par yield curve rates*, `https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_yield_curve&field_tdr_date_value_month=202312`, last accessed: 2023-12-08.

[26] Binance, *Binance public data*, last accessed: 2023-12-08.

[27] S. S. Lee and P. A. Mykland, "Jumps in Financial Markets: A New Nonparametric Test and Jump Dynamics," *The Review of Financial Studies*, vol. 21, no. 6, pp. 2535–2563, Dec. 2007. eprint: `https://academic.oup.com/rfs/article-pdf/21/6/2535/24453758/hhm056.pdf`.

[28] G. Huberman and W. Stanzl, "Price manipulation and quasi-arbitrage," *Econometrica*, vol. 72, no. 4, pp. 1247–1275, 2004.

[29] J. Gatheral, "No-dynamic-arbitrage and market impact," *Quantitative Finance*, vol. 10, no. 7, pp. 749–759, 2010.

[30] A. A. Obizhaeva and J. Wang, "Optimal trading strategy and supply/demand dynamics," in *American Finance Association Meetings*, 2006.

[31] A. S. Kyle and A. A. Obizhaeva, "Market microstructure invariance: Empirical hypotheses," *Econometrica*, vol. 84, no. 4, pp. 1345–1404, 2016.

[32] C. C. Moallemi and U. Patange, "Hybrid scheduling with mixed-integer programming at columbia business school," *INFORMS Journal on Applied Analytics*, vol. 54, no. 3, pp. 222–240, 2024. eprint: `https://doi.org/10.1287/inte.2022.0070`.

[33] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2022.

[34] E. Guerra, A. Sandweiss, and S. D. Park, "Does rationing really backfire? a critical review of the literature on license-plate-based driving restrictions," *Transport Reviews*, vol. 42,

no. 5, pp. 604–625, 2022. eprint: https://doi.org/10.1080/01441647.2021.1998244.

[35] G. Tsipurkey, "The four horsemen of the mandated return to office," *Forbes*, Apr. 1, 2023, Accessed: 2023-03-03.

[36] J. Pisani and K. Rhone, "U.S. return-to-office rate rises above 50% for first time since pandemic began," *The Wall Street Journal*, Feb. 1, 2023, Accessed: 2023-03-02.

[37] J. Elias, "Google asks some employees to share desks amid office downsizing," *CNBC*, Feb. 23, 2023, Accessed: 2023-03-03.

[38] H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," *Computers and Industrial Engineering*, vol. 86, pp. 43–59, 2015.

[39] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, "An overview of curriculum-based course timetabling," *TOP*, vol. 23, no. 2, pp. 313–349, 2015.

[40] E. K. Burke and Y. Bykov, "An adaptive flex-deluge approach to university exam timetabling," *INFORMS Journal on Computing*, vol. 28, no. 4, pp. 781–794, 2016.

[41] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf, and T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," *Computers and Operations Research*, vol. 65, pp. 83–92, 2016. arXiv: 1409.7186.

[42] R. Asín Achá and R. Nieuwenhuis, "Curriculum-based course timetabling with SAT and MaxSAT," *Ann Oper Res*, vol. 218, no. 1, pp. 71–91, 2014.

[43] S. Abdullah and H. Turabieh, "On the use of multi neighbourhood structures within a tabu-based memetic approach to university timetabling problems," *Inf Sci*, vol. 191, pp. 146–168, 2012.

[44] S. A. Mirhassani, "A computational approach to enhancing course timetabling with integer programming," *Appl Math Comput*, vol. 175, no. 1, pp. 814–822, 2006.

[45] Á. García-Sánchez, A. Hernández, E. Caro, and G. Jiménez, "Universidad Politécnica de Madrid Uses Integer Programming for Scheduling Weekly Assessment Activities," *INFORMS Journal on Applied Analytics*, vol. 49, no. 2, pp. 104–116, 2019.

[46] A. Tripathy, "School timetabling - a case in large binary integer linear programming.," *Management Science*, vol. 30, no. 12, pp. 1473–1489, 1984.

[47] J. Stallaert, "Automated timetabling improves course scheduling at UCLA," *Interfaces*, vol. 27, no. 4, pp. 67–81, 1997.

[48] H. A. Yekta and R. Day, "Optimization-based mechanisms for the course allocation problem," *INFORMS Journal on Computing*, vol. 32, no. 3, pp. 641–660, 2020.

[49] E. Budish, G. P. Cachon, J. B. Kessler, and A. Othman, "Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation," *Operations Research*, vol. 65, no. 2, pp. 314–336, 2017.

[50] J. J. Dinkel, J. Mote, and M. A. Venkataramanan, "Efficient decision support system for academic course scheduling," *Operations Research*, vol. 37, no. 6, pp. 853–864, 1989.

[51] A. Kannan, G. Van Den Berg, and A. Kuo, "ISchedule to personalize learning," *Interfaces*, vol. 42, no. 5, pp. 437–448, 2012.

[52] T. R. Hinkin and G. M. Thompson, "SchedulExpert: Scheduling Courses in the Cornell University School of Hotel Administration," *Interfaces*, vol. 32, no. 6, pp. 45–57, 2002.

[53] G. Gonzalez, C. Richards, and A. Newman, "Optimal course scheduling for United States Air Force Academy cadets," *Interfaces*, vol. 48, no. 3, pp. 217–234, 2018.

[54] C. H. Martin, "Ohio University's College of Business uses integer programming to schedule classes," *Interfaces*, vol. 34, no. 6, pp. 460–465, 2004.

[55] B. A. Kassa, "Implementing a class-scheduling system at the College of Business and Economics of Bahir Dar University, Ethiopia," *Interfaces*, vol. 45, no. 3, pp. 203–215, 2015.

[56] A. E. Phillips, H. Waterer, M. Ehrgott, and D. M. Ryan, "Integer programming methods for large-scale practical classroom assignment problems," *Comput Oper Res*, vol. 53, pp. 42–53, 2015.

[57] M. W. Carter and C. A. Tovey, "When Is the Classroom Assignment Problem Hard?" *Operations Research*, vol. 40, no. 1-supplement-1, S28–S39, 1992.

[58] O. Strichman, "Near-Optimal Course Scheduling at the Technion," *Interfaces*, vol. 47, no. 6, pp. 537–554, 2017.

[59] G. Chen, X. Fei, H. Jia, X. Yu, and S. Shen, "The University of Michigan Implements a Hub-and-Spoke Design to Accommodate Social Distancing in the Campus Bus System Under COVID-19 Restrictions," *Interfaces*, vol. 52, no. 6, pp. 539–552, 2022.

[60] P. I. Frazier *et al.*, "Modeling for Covid-19 college reopening decisions: Cornell, a case study," *Proceedings of the National Academy of Sciences*, vol. 119, no. 2, e2112532119, 2022. eprint: `https://www.pnas.org/doi/pdf/10.1073/pnas.2112532119`.

[61] S. Kacapyr, *The unsung engineering behind Cornell's fall 2020 schedule*, https://www.engineering.cornell.edu/spotlights/unsung-engineering-behind-cornells-fall-2020-schedule, Accessed: 2022-05-12, 2021.

[62] UNESCO, *Covid-19 response – hybrid learning*, https://en.unesco.org/sites/default/files/unesco-covid-19-response-toolkit-hybrid-learning.pdf, Accessed: 2022-05-11, 2020.

[63] C. Barnhart, D. Bertsimas, A. Delarue, and J. Yan, "Course Scheduling Under Sudden Scarcity: Applications to Pandemic Planning," *Manufacturing and Service Operations Management*, vol. 24, no. 2, pp. 727–745, 2022.

[64] M. Navabi-Shirazi, M. El Tonbari, N. Boland, D. Nazzal, and L. N. Steimle, "Multicriteria Course Mode Selection and Classroom Assignment Under Sudden Space Scarcity," *Manufacturing & Service Operations Management*, vol. 24, no. 6, pp. 3252–3268, 2022.

[65] A. B. Gore, M. E. Kurz, M. J. Saltzman, B. Splitter, W. C. Bridges, and N. J. Calkin, "Clemson University's Rotational Attendance Plan During COVID-19," *INFORMS Journal on Applied Analytics*, vol. 52, no. 6, p. 553, 2022.

[66] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer Programming* (Graduate texts in mathematics). Springer, 2014, vol. 271, ISBN: 9783319110073.

[67] G. L. Nemhauser and L. A. Wolsey, *Integer programming and combinatorial optimization*. Springer, 1988, vol. 191.

[68] N. Beaumont, "Scheduling staff using mixed integer programming," *European Journal of Operational Research*, vol. 98, no. 3, pp. 473–484, 1997.

[69] T. Saraç and B. Tutumlu, "A mix integer programming model and solution approach to determine the optimum machine number in the unrelated parallel machine scheduling problem," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 37, no. 1, pp. 329–345, 2022.

[70] N. Yadav and A. Tanksale, "An integrated routing and scheduling problem for home healthcare delivery with limited person-to-person contact," *European Journal of Operational Research*, vol. 303, no. 3, pp. 1100–1125, 2022.

[71] W. Brown and U. Patange, "Targeted intervention in random graphs," in *Algorithmic Game Theory*, T. Harks and M. Klimm, Eds., Cham: Springer International Publishing, 2020, pp. 211–225, ISBN: 978-3-030-57980-7.

[72] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, "The diffusion of microfinance," *Science*, vol. 341, no. 6144, p. 1 236 498, 2013.

[73] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, "Using Gossips to Spread Information: Theory and Evidence from Two Randomized Controlled Trials," *The Review of Economic Studies*, vol. 86, no. 6, pp. 2453–2490, Feb. 2019.

[74] A. Banerjee, A. G a, E. Duflo, and M. Jackson, *Gossip: Identifying central individuals in a social network*, Jun. 2014.

[75] M. Akbarpour, S. Malladi, and A. Saberi, "Diffusion, seeding, and the value of network information," in *Proceedings of the 2018 ACM Conference on Economics and Computation*, ser. EC '18, Ithaca, NY, USA: ACM, 2018, pp. 641–641, ISBN: 978-1-4503-5829-3.

[76] D. Eckles, H. Esfandiari, E. Mossel, and M. A. Rahimian, *Seeding with costly network information*, 2019. arXiv: 1905.04325 [cs.SI].

[77] O. Candogan, K. Bimpikis, and A. Ozdaglar, "Optimal pricing in networks with externalities," *Operations Research*, vol. 60, no. 4, pp. 883–905, 2012.

[78] G. Demange, "Optimal targeting strategies in a network under complementarities," *Games and Economic Behavior*, vol. 105, pp. 84–103, 2017.

[79] J. Huang, A. Mani, and Z. Wang, "The value of price discrimination in large random networks," *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019.

[80] F. Parise and A. Ozdaglar, *Graphon games: A statistical framework for network games and interventions*, 2018. arXiv: 1802.00080 [cs.GT].

[81] E. Breza, A. G. Chandrasekhar, T. H. McCormick, and M. Pan, *Using aggregated relational data to feasibly identify network structure without network data*, 2017. arXiv: 1703.04157 [stat.ME].

[82] F. Chung, L. Lu, and V. Vu, "Eigenvalues of random power law graphs," *Annals of Combinatorics*, vol. 7, no. 1, pp. 21–33, 2003.

[83] F. Chung and M. Radcliffe, "On the spectra of general random graphs," *Electr. J. Comb.*, vol. 18, Oct. 2011.

[84] F. Chung and L. Lu, "Connected components in random graphs with given expected degree sequences," *Annals of Combinatorics*, vol. 6, no. 2, pp. 125–145, 2002.

[85] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ser. STOC '00, Portland, Oregon, USA: Association for Computing Machinery, 2000, 171–180, ISBN: 1581131844.

[86] K. Dasaratha, "Distributions of Centrality on Networks," arXiv.org, Papers, Sep. 2017.

[87] M. Avella-Medina, F. Parise, M. T. Schaub, and S. Segarra, "Centrality measures for graphons," *CoRR*, vol. abs/1707.09350, 2017. arXiv: 1707.09350.

[88] N. Ruggeri and C. D. Bacco, *Sampling on networks: Estimating eigenvector centrality on incomplete graphs*, 2019. arXiv: 1908.00388 [cs.SI].

[89] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06, Philadelphia, PA, USA: Association for Computing Machinery, 2006, 631–636, ISBN: 1595933395.

[90] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins, "The web as a graph: Measurements, models, and methods," in *Computing and Combinatorics*, T. Asano, H. Imai, D. T. Lee, S.-i. Nakano, and T. Tokuyama, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 1–17, ISBN: 978-3-540-48686-2.

[91] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[92] O. Goldreich and D. Ron, "Approximating average parameters of graphs," *Random Structures & Algorithms*, vol. 32, no. 4, pp. 473–493, 2008.

[93] A. Dasgupta, R. Kumar, and T. Sarlos, "On estimating the average degree," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14, Seoul, Korea: Association for Computing Machinery, 2014, 795–806, ISBN: 9781450327442.

[94] C. L. Canonne, *A short note on learning discrete distributions*, 2020. arXiv: 2002.11457 [math.ST].

[95] S.-Y. Yun and A. Proutiere, "Optimal sampling and clustering in the stochastic block model," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 13 422–13 430.

[96] E. Abbe, "Community detection and stochastic block models: Recent developments," *Journal of Machine Learning Research*, vol. 18, no. 177, pp. 1–86, 2018.

[97] T. Tabouy, P. Barbillon, and J. Chiquet, *Variational inference for stochastic block models from sampled data*, 2017. arXiv: 1707.04141 [stat.ME].

[98] M. T. Schaub, S. Segarra, and J. N. Tsitsiklis, *Blind identification of stochastic block models from dynamical observations*, 2019. arXiv: 1905.09107 [cs.LG].

[99] M. Avella-Medina, F. Parise, M. T. Schaub, and S. Segarra, *Centrality measures for graphons*, 2017. arXiv: 1707.09350.

[100] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[101] A. Sinclair, "Improved bounds for mixing rates of markov chains and multicommodity flow," in *LATIN '92*, I. Simon, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 474–487, ISBN: 978-3-540-47012-0.

Appendix A

A.1 Contract Parameters and Lender Recovery

First, we establish what effect a liquidation has on the health of a loan. Clearly, it would be desirable to the borrower and the lender that the loan becomes healthier after liquidation. Lemma A.1.1 establishes such a condition. We note that this condition was also observed by [5] and [7].

Lemma A.1.1 *The health factor improves upon liquidation iff $h_t > \ell(1+\lambda)$, then any liquidation.*

Proof: Let $\tilde{H}(h_t, f)$ denote the post-liquidation health factor when a loan having a health factor $h_t \leq 1$ is liquidated at factor f. We have,

$$\tilde{H}(h_t, f) := \frac{p_t \ell}{V_t(1-f)} \left(N_t - \frac{f V_t (1+\lambda)}{p_t} \right) = \frac{h_t - f(1+\lambda)\ell}{1-f}. \tag{A.1}$$

Then, we have,

$$\tilde{H}(h_t, f) > h_t \iff h_t > \ell(1+\lambda).$$

□

Let us now consider liquidator incentives and establish their optimal behavior. Assuming that the liquidation occurs at time t_0, a profit-maximizing liquidator faces the following optimization problem (M):

$$\max_{0 \leq f \leq F} \quad \lambda V_{t_0} f \tag{A.2}$$

$$\text{subject to} \quad V_{t_0} f (1+\lambda)/p_{t_0} \leq N_{t_0}. \tag{A.3}$$

Here, f is factor of liquidation, and is bounded above by the close factor. The constraint (A.3) limits the payment to the liquidator by the available collateral.

Lemma A.1.2 *It is optimal to liquidate the maximum amount possible.*

Proof: Since the objective function increases with f, the profit is maximized when f is set as large as possible. □

Optimal liquidator behavior was also considered by [5], but their setting allowed for jumps in the price process, and thus health factors could become strictly less than 1 at the time of liquidation. Consequently, their setting allowed for multiple successive liquidations on the same loan, which would not be possible here.

We now assume that there are a large number of liquidators competing to liquidate unhealthy loans. Due to Lemma A.1.2, this means that an unhealthy loan gets liquidated as soon as its health factor drops to 1. We are now ready to prove that whenever the "collateral sufficiency" condition (Assumption 1.1) is satisfied, the lender is able to recover the loan.

Theorem A.1.1 *If the collateral sufficiency condition is satisfied, then*

1. $h_t \geq 1$ *for all* t.
2. *Lender never loses any money, and*
3. *Each liquidation takes place at the close factor.*

Proof: For part 1, it suffices to prove that each time there is a liquidation, the health factor becomes larger than 1. To that end, suppose that the first liquidation takes place at time t. Then, $\ell(1+\lambda) < 1 = h_t$, and hence by Lemma A.1.1, the post-liquidation health factor is greater than 1. Combined with the fact that the initial health-factor is greater than 1, the liquidators have an incentive to liquidate, and that the prices are continuous, we get the result.

For part 2, we note that as long as $N_t p_t > V_t$, the borrower has no endogenous incentive to default. This is equivalent to the condition that $h_t > \ell$. Since $\ell < 1$, it is always satisfied due to part 1.

For part 3, we note that constraint (A.3) is redundant due to our assumption, i.e., it is satisfied by any $f \in [0, F]$. To see this, we observe that

$$\begin{aligned}
\frac{N_{t_0} p_{t_0}}{V_{t_0}(1+\lambda)} &= \frac{h_{t_0}}{\ell(1+\lambda)}, \\
&= \frac{1}{\ell(1+\lambda)}, && (h_{t_0} = h_{\tau_1} = 1) \\
&\geq 1. && \text{(due to the assumption.)}
\end{aligned}$$

Thus, constraint (A.3) is satisfied by any $f \leq F \leq 1$. The maximum allowable factor of liquidation as prescribed by Lemma A.1.2 then must be the close factor. □

A.2 Proof of Theorem 1.4.1

Proof: First, we note by a sample path argument that $p_q(\alpha V, h) = \alpha p_q(V, h)$ for $\alpha \geq 0$. This is because, in any sample path, keeping h constant, multiplying V by α results in each payoff to the liquidator being multiplied by α. Thus, we can rewrite equation (1.1) as

$$p_q(V, h) = (1 - F) p_q(V, \tilde{H}(1, F)) E_1(h) + \lambda V F E_1(h),$$

where,

$$E_1(h) = \mathbb{E}_{\mathbb{Q}} \left[e^{(\gamma_V - r)\tau_1} \mathbb{1}\left(\tau_1 \leq T\right) \,\middle|\, h_0 = h \right].$$

We observe that,

$$E_1(h) = \mathbb{E}_{\mathbb{Q}} \left[e^{(\gamma_V - r - \nu)\tau_1} \mathbb{1}\left(\tau_1 < \infty\right) \,\middle|\, h_0 = h \right],$$

by conditioning the expectation on τ_1 and applying tower law. Define

$$\begin{aligned} X_t(h) &:= \frac{\log h_0 - \log h_t}{\sigma}, \\ &= (-B_t) + \left(-\frac{\tilde{\mu}}{\sigma}\right) t, \\ &\stackrel{d}{=} B_t + ct, \end{aligned} \tag{A.4}$$

where $c := -\frac{\tilde{\mu}}{\sigma}$, and consider again the expression,

$$E_1(h) = \mathbb{E}_{\mathbb{Q}}\left[e^{(\gamma_V - r - \nu)\tau_1} \mathbb{1}\left(\tau_1 < \infty\right) \,\middle|\, h_0 = h\right].$$

We can redefine τ_1 to be the first hitting time of the drifted Brownian motion to the level $a := \frac{\log h_0}{\sigma} > 0$. By Girsanov Theorem, the process X_t is a standard Brownian motion on $0 \leq t \leq \tau_1$ under measure $\tilde{\mathbb{Q}}$, where

$$\frac{d\tilde{\mathbb{Q}}}{d\mathbb{Q}} = \exp -cB_{\tau_1} - c^2\tau_1/2.$$

Denoting by $\mathbb{E}_{\tilde{\mathbb{Q}}}$ the expectation with respect to measure $\tilde{\mathbb{Q}}$, we can write,

$$\begin{aligned} E_1(h) &= \mathbb{E}_{\tilde{Q}}\left[\exp B_{\tau'_a} c + \frac{c^2}{2}\tau'_a + (\gamma_V - r - \nu)\,\tau'_a \mathbb{1}\left(\tau'_a < \infty\right) \,\middle|\, h_0 = h\right] \\ &= \mathbb{E}_{\tilde{Q}}\left[\exp X_{\tau'_a} c - \frac{c^2}{2}\tau'_a + (\gamma_V - r - \nu)\,\tau'_a \mathbb{1}\left(\tau'_a < \infty\right) \,\middle|\, h_0 = h\right] \\ &= e^{ac}\mathbb{E}_{\tilde{Q}}\left[\exp\left(-\frac{c^2}{2} + \gamma_V - r - \nu\right)\tau'_a \mathbb{1}\left(\tau'_a < \infty\right) \,\middle|\, h_0 = h\right], \end{aligned}$$

where τ'_a is the hitting time of a standard Brownian motion (X_t under measure $\tilde{\mathbb{Q}}$) to the level $a > 0$. Since τ'_a is not integrable, and finite with probability 1, $E_1(h)$ is infinite whenever,

$$\begin{aligned} &-\frac{c^2}{2} + \gamma_V - r - \nu > 0 \\ \iff\ &\tilde{\mu}^2 + 2\sigma^2(\nu + r - \gamma_V) < 0. \end{aligned}$$

Note that due to Assumption 1.2, we have

$$\tilde{\mu}^2 + 2\sigma^2(\nu + r - \gamma_V) > 0.$$

Now, we can use the fact that,

$$M_t = \exp \eta X_t - \eta^2 t/2,$$

is a Martingale for $\eta \in \mathbb{R}$ with respect to $(\mathcal{F}_t)_{t\geq 0}$ under the measure $\tilde{\mathbb{Q}}$ to prove that,

$$\mathbb{E}_{\tilde{Q}}\left[\exp -\frac{\eta^2}{2}\tau'_a\right] = \exp -|\eta|\, a. \tag{A.5}$$

Then, we get that,

$$\begin{aligned} E_1(h) &= e^{ac} e^{-a\sqrt{c^2+2(\nu+r-\gamma_V)}} \\ &= h^{-\kappa}. \end{aligned}$$

To summarize, we have

$$E_1(h) = \begin{cases} h^{-\kappa}, & \left(\tilde{\mu} + \sigma^2\right)^2 + 2\sigma^2(\nu - \gamma_C) \geq 0, \\ \infty, & \left(\tilde{\mu} + \sigma^2\right)^2 + 2\sigma^2(\nu - \gamma_C) < 0. \end{cases} \tag{A.6}$$

Finally, we have

$$\begin{aligned} p_q(V,h) &= h^{-\kappa}\left(\lambda VF + (1-F)p_q(V,\tilde{H}(1,F))\right), \\ \implies p_q(V,\tilde{H}(1,F)) &= \tilde{H}(1,F)^{-\kappa}\left(\lambda VF + (1-F)p_q(V,\tilde{H}(1,F))\right), \\ \implies p_q(V,\tilde{H}(1,F)) &= \frac{\lambda VF\tilde{H}(1,F)^{-\kappa}}{\left(1-(1-F)\tilde{H}(1,F)^{-\kappa}\right)}, \\ \implies p_q(V,h) &= \frac{\lambda VF}{h^{\kappa}\left(1-\dfrac{1-F}{\tilde{H}(1,F)^{\kappa}}\right)}. \end{aligned}$$

□

A.3 Proof of Theorem 1.4.2

Proof: We note that the function $C(h)$ is convex and therefore, apply first order conditions to get the maxima. We further note that if the maxima is less than 1, then the cost functions is increasing beyond 1, and $h = 1$ is optimal feasible. □

Appendix B

B.1 Student-Level Scheduling

B.1.1 Proof of Theorem 2.2.1

Proof: We observe that Part 2 implies Part 2.2.1 since TE is equal to the objective function of the integer program when $\lambda = \mu = 0$, which must always be greater than or equal to the optimal objective value of the relaxed linear program. To prove Part 2, consider first the relaxed linear program with $\lambda = \mu = 0$. For simplicity, we can omit constraints (2.10), (2.11), and (2.12) as the objective function no longer has δ or s in it. Let α_i, for $i \in \mathcal{S}$, be the dual variables corresponding to (2.8) and let $\beta_{jk}, 1 \leq j \leq M, k \in C$ be the dual variables corresponding to (2.9). We have the dual problem

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{S}} \alpha_i - \sum_{1 \leq j \leq M} \sum_{k \in C} c_k \beta_{jk} & \\
\text{subject to} \quad & \alpha_i \leq \sum_{k : i \in \mathcal{A}_k} \beta_{jk} & \forall i \in \mathcal{S}, 1 \leq j \leq M \\
& 0 \leq \beta_{jk} \leq 1 & 1 \leq j \leq M, k \in C.
\end{aligned}$$

At the dual optimal solution, we will have $\alpha_i = \min_{1 \leq j \leq M} \sum_{k : i \in \mathcal{A}_k} \beta_{jk}$. We can then rewrite the dual problem as,

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{S}} \min_{1 \leq j \leq M} \sum_{k : i \in \mathcal{A}_k} \beta_{jk} - \sum_{k \in C} c_k \sum_{j=1}^{M} \beta_{jk} & \\
\text{subject to} \quad & 0 \leq \beta_{jk} \leq 1 & 1 \leq j \leq M, k \in C.
\end{aligned}$$

We observe that if β is optimal, then so is $\bar{\beta}_{\cdot k} = \frac{1}{M}\sum_{j'=1}^{M}\beta_{j'k}$. First, it is obviously feasible. We can also see that the objective value does not decrease with this replacement. The first term is

$$\begin{aligned}\min_{1\le j\le M}\sum_{k\in C}\bar{\beta}_{jk}\mathbb{1}\left(i\in\mathcal{A}_k\right) &= \min_{1\le j\le M}\frac{1}{M}\sum_{j'=1}^{M}\sum_{k\in C}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right)\\ &= \frac{1}{M}\sum_{j'=1}^{M}\sum_{k\in C}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right)\\ &\ge \min_{1\le j'\le M}\sum_{k\in C}\beta_{j'k}\mathbb{1}\left(i\in\mathcal{A}_k\right).\end{aligned}$$

Thus, without loss of generality, we may assume that $\beta_{jk} = \beta_k$. Then, the dual problem objective becomes,

$$\begin{aligned}\max_{\beta\in[0,1]}\sum_{i\in\mathcal{S}}\sum_{k\in C}\left(\beta_k\mathbb{1}\left(i\in\mathcal{A}_k\right)-M\beta_k\right) &= \max_{\beta\in[0,1]}\sum_{k\in C}\left(|\mathcal{A}_k|-c_kM\right)\\ &= \sum_{k\in C}\left(|\mathcal{A}_k|-c_kM\right)^+\\ &= \text{UE}.\end{aligned}$$

Since UE is attained by the uniform fractional student assignment, we conclude that the uniform fractional student assignment is optimal when $\lambda = \mu = 0$.

Incidentally, for this assignment, $\delta_{j,k} = 0$, $1 \le j \le M$, $\forall k \in C$. Thus, the uniform fractional student assignment is also optimal when $\lambda > \mu = 0$. To show this for the case $\mu, \lambda > 0$, it suffices to argue that s is minimized by the uniform fractional student assignment as well. For any t, we observe that $\sum_{j=1}^{M} s_j^t$ is the TE for a problem (P_t) where $C = C_t$. For problem P_t, again, the uniform fractional assignment is optimal and sets $s_j^t = \sum_{k\in C_t}\left(\frac{|\mathcal{A}_k|}{M} - c_k\right)^+$, $1 \le j \le M$. Since s_j^t's have the same values for all j and $\max_{1\le j\le M} s_j^t \ge \sum_{j=1}^{M} s_j^t/M$, this assignment also minimizes $\max_{1\le j\le M} s_j^t$. As this holds for each t, we get that $s = \left(\max_{t,j} s_j^t - E\right)^+$ is also minimized by the uniform fractional student assignment. Hence, the uniform fractional student assignment is optimal for all $\lambda, \mu \ge 0$. □

B.1.2 Proof of Theorem 2.2.2

Proof: It suffices to prove that MD_k is, in fact, the minimum contribution to the TD by course k. To that end, we fix a course k having non-zero number of students, and consider an optimization problem ($\mathrm{P}^{(k)}$) that minimizes its contribution to the TD. It is given by,

$$\text{minimize} \quad \sum_{j=1}^{M} \left| d_j - \frac{|\mathcal{A}_k|}{M} \right|, \tag{B.1}$$

$$\text{subject to} \quad \sum_{j=1}^{M} d_j = |\mathcal{A}_k|\,, \tag{B.2}$$

$$d_j \in \mathbb{Z}_+ \cup \{0\}\,, \qquad 1 \le j \le M. \tag{B.3}$$

Here, d_j is the number of students of class k assigned to group j. We note that any feasible solution $\{\pi\}_{i\in\mathcal{S},1\le j\le M}$ to (P) corresponds to a feasible solution $\{d\}^{(k)}_{1\le j\le M}$ to ($\mathrm{P}^{(k)}$) for each k: we can set $d_j = \sum_{i\in\mathcal{S}} \pi_{ij}$. The sum of objectives of ($\mathrm{P}^{(k)}$)'s is equal to the TD of the solution $\{\pi\}_{i\in\mathcal{S},1\le j\le M}$. Thus, the sum of optimal objective values of ($\mathrm{P}^{(k)}$)'s is a lower bound on the TD of any solution to (P).

To solve ($\mathrm{P}^{(k)}$), we observe that an optimal solution must exist since the constraint set is non empty. For example, the solution setting $d_1 = |\mathcal{A}_k|$ and $d_j = 0, j > 1$ is feasible.

If $|\mathcal{A}_k|$ is a multiple of M, we can divide the students into M groups of size $\frac{|\mathcal{A}_k|}{M}$ each to get an objective value of 0. The minimum contribution to TD by course k in this case is 0, which is also the value of MD_k.

If $|\mathcal{A}_k|$ is not a multiple of M, we claim that there exists an optimal solution $\left\{d^*_j\right\}_{1\le j\le M}, 1 \le j \le M$ such that for each j, d_j is either $\left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil$ or $\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor$. Here $\lceil x \rceil$ denotes the smallest integer greater than or equal to x. To see this, consider an optimal feasible solution $\left\{d_j\right\}_{1\le j\le M}$ where this does not hold. That is, $\exists 1 \le j_1 \le M$, such that $d_{j_1} \notin \left[\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil\right]$. First, suppose that $d_{j_1} < \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor$. We must then have at least one $1 \le j_2 \le M$ such that $d_{j_2} \ge \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil$. We can create a new feasible solution $\left\{d'_j\right\}_{1\le j\le M}$ by setting $d'_j = d_j, 1 \le j \le M, j \notin \{j_1, j_2\}$, and $d'_{j_1} = d_{j_1} + 1, d'_{j_2} = d_{j_2} - 1$. We note that this solution is also feasible since $\sum_{j=1}^{M} d'_j = \sum_{j=1}^{M} d_j$

and d'_j is a non-negative integer for $1 \leq j \leq M$. We also observe that the objective value for the solution $\left\{d'_j\right\}_{1\leq j\leq M}$ is strictly less than that for the solution $\left\{d_j\right\}_{1\leq j\leq M}$. Proceeding this way, we converge to a feasible solution $\left\{\tilde{d}_j\right\}_{1\leq j\leq M}$ having an objective value no greater than the optimal objective value and the property that $\tilde{d}_j \geq \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, 1 \leq j \leq M$. In a similar fashion, we can get to a solution $\left\{d^*_j\right\}_{1\leq j\leq M}$ such that $d^*_j \in \left[\left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor, \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil\right], 1 \leq j \leq M$ without increasing the objective value from the optimal objective value. Thus, our claim holds.

We now compute the value of the objective at the solution $\left\{d^*_j\right\}_{1\leq j\leq M}$. Let $L = \left\{j : d^*_j = \left\lfloor \frac{|\mathcal{A}_k|}{M} \right\rfloor\right\}$, and $G = \left\{j : d^*_j = \left\lceil \frac{|\mathcal{A}_k|}{M} \right\rceil\right\}$. To satisfy the feasibility constraint (B.2), we must have $|G| = |\mathcal{A}_k| \bmod M$, and $|L| = M - |\mathcal{A}_k| \bmod M$. The contribution to the objective value by a group $j \in L$ is given by $\left\{\frac{|\mathcal{A}_k|}{M}\right\}$, while for a group $j' \in G$, it is given by $1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}$. Thus, the total objective value is

$$\left\{\frac{|\mathcal{A}_k|}{M}\right\}(M - |\mathcal{A}_k| \bmod M) + (|\mathcal{A}_k| \bmod M)\left(1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}\right) = 2\,(|\mathcal{A}_k| \bmod M)\left(1 - \left\{\frac{|\mathcal{A}_k|}{M}\right\}\right).$$

Here, we used the fact that $\left\{\frac{a}{b}\right\} b = a \bmod b$, for any positive integers a and b. Thus, we conclude that the minimum contribution by any course k to the total deviation is given by MD_k. The total deviation is, therefore, bounded below by the minimal deviation, which is defined as $\sum_{k\in C} \mathrm{MD}_k$.

□

B.2 Team-Level Scheduling

B.2.1 Proof of Theorem 2.3.1

Proof: Consider the dual of this problem after relaxing the integer constraints. It is given by

$$\min_{\alpha,\beta,\gamma,\delta,\eta} \quad \sum_{i=1}^{N}\sum_{d\in D} l^d_{\max}\alpha_{i,d} - \sum_{i=1}^{N}\sum_{d\in D} l^d_{\min}\beta_{i,d} + \sum_{i=1}^{N}\sum_{b\in B}\gamma_{i,b} + \sum_{t=1}^{T+p}\delta_t, \tag{B.4}$$

$$\text{subject to} \quad \delta_t + \sum_{i\in s}\left(\alpha_{i,\text{dow}(t)} - \beta_{i,\text{dow}(t)} + \gamma_{i,\text{bl}(t)} - \sum_{j\in s}\eta_{ij}\right) \geq 0 \qquad \forall s \in \mathcal{S}, \forall t \in [T], \tag{B.5}$$

$$\delta_{T+r} + \sum_{i\in s}\gamma_{i,\text{bl}(T+r)} \geq 0 \qquad \forall s \in \mathcal{S}, 1 \leq r \leq p, \tag{B.6}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\eta_{ij} = 1, \tag{B.7}$$

$$\alpha_{i,d} \geq 0, \quad \beta_{i,d} \geq 0, \quad \eta_{ij} \geq 0 \qquad \forall d \in D, 1 \leq i, j \leq N. \tag{B.8}$$

First, we propose a solution to the dual. We can set,

$$\begin{aligned}
\alpha^*_{i,d} = \beta^*_{i,d} &= 0, && 1 \leq i \leq N, d \in \mathcal{D},\\
\gamma^*_{i,b} &= 0, && b \in \mathcal{B},\\
\delta^*_t &= \frac{K(K-1)}{N(N-1)}, && 1 \leq t \leq T,\\
\delta^*_{T+r} &= 0, && 1 \leq r \leq p,\\
\eta^*_{ij} &= \frac{1}{N(N-1)}, && 1 \leq i \neq j \leq N,\\
\eta^*_{ii} &= 0, && 1 \leq i \leq N.
\end{aligned}$$

We observe that in this setting,

$$\sum_{i\in}\sum_{j\in s}\eta_{ij} = \frac{K(K-1)}{N(N-1)} \qquad \forall s \in \mathcal{S},$$

and constraint (B.5) is satisfied. Likewise, all the other constraints are also satisfied. We observe that the objective value is $\frac{TK(K-1)}{N(N-1)}$.

Now, we need only prove that the uniform fractional group assignment is feasible and achieves the same objective value in the primal. If true, this would imply that it is optimal for the primal and we would have proved Parts 1 and 2.

To that end, we go back to the definition of the uniform fractional group assignment (Definition 2.2). In this solution, we have that $\pi^*_{s,t} = \frac{1}{\binom{N}{K}}, \forall s \in \mathcal{S}, 1 \le t \le T+p$ and the constraint (2.19) is automatically satisfied. Now, any $i \in \{1,\dots,N\}$ is present in $\binom{N-1}{K-1}$ LT groups, and for any $b \in \mathcal{B}$, $|b| = \frac{N}{K}$. Thus, the LHS of constraint (2.20) becomes

$$\frac{N\binom{N-1}{K-1}}{K\binom{N}{K}} = 1,$$

and it is also satisfied. Finally, for any $d \in \mathcal{D}, 1 \le i \le N$,

$$\sum_{s:i\in s}\sum_{t\in[T]:\mathrm{dow}(t)=d}\pi^*_{s,t} = \sum_{t\in[T]:\mathrm{dow}(t)=d}\frac{\binom{N-1}{K-1}}{\binom{N}{K}} = \frac{\left|\mathrm{dow}^{-1}(d)\right|K}{N},$$

and by the definitions of $l^d_{\min}$ and $l^d_{\max}$, constraint (2.21) is also satisfied. We observe that for any $1 \le i \ne j \le N$,

$$\sum_{s:i,j\in s}\sum_{t\in[T]}\pi^*_{s,t} = \frac{T\binom{N-2}{K-2}}{\binom{N}{K}} = \frac{TK(K-1)}{N(N-1)}.$$

For $i = j$, this quantity will be higher, and so we need to set $m^*_{\min} = \frac{TK(K-1)}{N(N-1)}$ which is also the value of the objective. Thus, we have proved Parts 1 and 2.

The solution to the dual tells us an interesting thing about the relaxed LP. Because we found an optimal solution with $\eta_{ij} > 0, 1 \le i \ne j \le N$, we can conclude, by complementary slackness, that

in any primal optimal solution $\tilde{\pi}$, we must have $\sum_{s:i,j\in s} \sum_{t\in[T]} \tilde{\pi}_{s,t} = m_{\min}$ for every pair of distinct i and j, since η_{ij} is the dual variable for primal constraint (2.25). Thus, Part 3 is also proved. □

In our case, the dual optimal objective value is equal to $\frac{44\times4\times3}{12\times11} = 4$. Suppose that there exists a solution to the original mixed-integer program with $m_{\min} = 4$. Since this is also an optimal solution of the relaxed LP, we must have $m_{ij} = 4, 1 \leq i \neq j \leq N$. This would mean that every team i is attending classes in person at least $\frac{11\times4}{3} = 14.67$ times. But, every team attending classes in person 15 times is impossible since we would need at least $\frac{15\times12}{4} = 45$ days. Thus, we are left with an optimality gap in our solution. More generally, a necessary condition for a vanishing optimality gap is that both $\frac{TK}{N}$ and $\frac{TK(K-1)}{N(N-1)}$ be integers.

Appendix C

C.1 Omitted Proofs

First, we prove a result about general matrices that we will use later in the proofs.

Lemma C.1.1 *For any two $n \times n$ symmetric matrices A and B, let λ_1 and λ_2 be the largest (according to magnitude) eigenvalues of A, and let μ_1 be the largest (according to magnitude) eigenvalue of B (Note: By definition, $||B|| = \mu_1$ and $||A|| = \lambda_1$). Let κ be the inverted spectral gap, i.e. $|\lambda_2| \leq \kappa|\lambda_1|$. Then,*

$$\|v(A) - v(B)\|^2 \leq 2\left(1 - \frac{1}{\lambda_1}\sqrt{\frac{(\mu_1 - \eta)^2 - \lambda_1^2\kappa^2}{1 - \kappa^2}}\right)$$

Where $v(\cdot)$ is the unit eigenvector with the largest absolute eigenvalue of the matrix $\cdot$, and $\eta = \|A - B\|$.

Proof: First, we observe that,

$$\|Av(B)\| \geq \|Bv(B)\| - \|B - A\| \cdot \|v(B)\| = \mu_1 - \eta, \tag{C.1}$$

by the triangle inequality. Also, we can write, $v(B) = \phi_1 v(A) + \phi_2 w$ where $w \perp v(A)$ is a unit vector, and $\phi_1^2 + \phi_2^2 = 1$. Since $w \perp v(A)$, its image under A can at most have a magnitude of λ_2. That is,

$$||Aw|| \leq \lambda_2$$

From our assumption, we can write

$$\begin{aligned}\|Av(B)\| &\leq \sqrt{\lambda_1^2\phi_1^2 + \lambda_2^2\phi_2^2} \\ &\leq \sqrt{\lambda_1^2\phi_1^2 + \kappa^2\lambda_1^2\phi_2^2} \\ &= \lambda_1\sqrt{\phi_1^2 + \kappa^2\phi_2^2}\end{aligned}$$

Combining with (C.1), we get

$$\begin{aligned}\frac{\mu_1 - \eta}{\lambda_1} &\leq \sqrt{\phi_1^2 + \kappa^2\phi_2^2} \\ \implies \left(\frac{\mu_1 - \eta}{\lambda_1}\right)^2 &\leq \phi_1^2 + \kappa^2(1 - \phi_1^2) \\ \implies \phi_1 &\geq \frac{1}{\lambda_1}\sqrt{\frac{(\mu_1 - \eta)^2 - \lambda_1^2\kappa^2}{1 - \kappa^2}}\end{aligned}$$

Using the fact that $\|v(A) - v(B)\| = \|(1 - \phi_1)v(A) + \phi_2 w\| = \sqrt{(1 - \phi_1)^2 + \phi_2^2} = \sqrt{2(1 - \phi_1)}$, we get

$$\|v(A) - v(B)\|^2 \leq 2\left(1 - \frac{1}{\lambda_1}\sqrt{\frac{(\mu_1 - \eta)^2 - \lambda_1^2\kappa^2}{1 - \kappa^2}}\right).$$

□

Proof: [Proof of Theorem 3.3.1] From Lemma C.1.1, for matrices $\overline{A}$ and A, we have that

$$\left\|v_1(\overline{A}) - v_1(A)\right\|^2 \leq 2\left(1 - \frac{1}{\lambda_1}\sqrt{\frac{(\mu_1 - \eta)^2 - \lambda_1^2\kappa^2}{1 - \kappa^2}}\right)$$

where λ_1 is the first eigenvalue of $\overline{A}$, $\left|\lambda_i(\overline{A})\right| \leq \kappa\lambda_1$ for all $i > 1$, μ_1 is the first eigenvalue of A, and $\eta = \left\|\overline{A} - A\right\|$. By the triangle inequality $\mu_1 - \eta \geq \lambda_1 - 2\eta$, and so

$$\begin{aligned}
\left\|v_1(\overline{A}) - v_1(A)\right\|^2 &\leq 2\left(1 - \frac{1}{\lambda_1}\sqrt{\frac{(\lambda_1 - 2\eta)^2 - \lambda_1^2\kappa^2}{1-\kappa^2}}\right) \\
&\leq 2\left(1 - \frac{1}{\lambda_1}\sqrt{\frac{\lambda_1^2(1-\kappa^2) - 4\eta\lambda_1}{1-\kappa^2}}\right) \\
&\leq 2\left(1 - \sqrt{1 - \frac{4\eta}{(1-\kappa^2)\lambda_1}}\right) \\
&\leq \frac{8\eta}{(1-\kappa^2)\lambda_1} \qquad \left(\sqrt{1-x} > 1 - x, x \in [0,1]\right).
\end{aligned}$$

Since $d_{\max} \geq \frac{4}{9}\log(2n/\delta)$, we have, from the proof of Theorem 1 in [83] that $\eta \leq 2\sqrt{d_{\max}\log(2n/\delta)}$. If $\lambda_1 \geq \frac{1024\sqrt{d_{\max}\log(2n/\delta)}}{(1-\kappa^2)\epsilon^2}$, we have that

$$\begin{aligned}
\left\|v_1(\overline{A}) - v_1(A)\right\|^2 &\leq \frac{16\sqrt{d_{\max}\log(2n/\delta)}}{(1-\kappa^2)\lambda_1} \\
&\leq \epsilon^2/64,
\end{aligned}$$

and so $\left\|v_1(\overline{A}) - v_1(A)\right\| \leq \epsilon/8$. For a sufficiently large budget, by Proposition 2 in [1], $\left\|\frac{y^*}{\|y^*\|} - v_1(A)\right\| \leq \epsilon/8$, where y^* is the optimal intervention. By the triangle inequality this gives us that $\left\|\frac{y^*}{\|y^*\|} - v_1(\overline{A})\right\| \leq \epsilon/4$. Applying Lemma 3.3.1 and law of cosines gives a competitive ratio of $1 - \epsilon$. □

Proof: [Proof of Lemma 3.3.1] First we give a general expression of the change in welfare observed at equilibrium after some intervention y.

$$\begin{aligned}
\Delta W(y; b, A) &= W(y; b, A) - W(\mathbf{0}; b, A) \\
&= \frac{1}{2}\left(\left(M^{-1}y\right)^\top\left(M^{-1}y\right) + 2\left(M^{-1}b\right)^\top\left(M^{-1}y\right)\right) \\
&= \frac{1}{2}\left(\left\|M^{-1}y\right\|^2 + 2\left(M^{-1}b\right)^\top\left(M^{-1}y\right)\right)
\end{aligned}$$

By the law of cosines, there is some vector y' with norm $\sqrt{2C\,(1-\gamma)}$ such that $y^* - y = y'$. Let $\lambda_M = \lambda_1(M^{-1})$. We can then give an additive bound on the utility loss $\Delta U = W\,(y^*; b, A) - W\,(y; b, A)$:

$$
\begin{aligned}
\Delta U &= \Delta W\,(y^*; b, A) - \Delta W\,(y; b, A) \\
&= \frac{1}{2}\left(\left\|M^{-1}y^*\right\|^2 + 2\left(M^{-1}b\right)^\top\left(M^{-1}y^*\right)\right) \\
&\quad - \frac{1}{2}\left(\left\|M^{-1}y\right\|^2 + 2\left(M^{-1}b\right)^\top\left(M^{-1}y\right)\right) \\
&= \frac{1}{2}\left(\left\|M^{-1}y^*\right\|^2 - \left\|M^{-1}\,(y^* - y')\right\|^2\right) && \text{(substituting } y = y^* - y'\text{)} \\
&\quad + \left(M^{-1}b\right)^\top M^{-1}\,(y^* - y) \\
&\leq \frac{1}{2}\left(\left\|M^{-1}y^*\right\|^2 - \left(\left\|M^{-1}y^*\right\| - \left\|M^{-1}y'\right\|\right)^2\right) \\
&\quad + \left(M^{-1}b\right)^\top M^{-1}\,(y^* - y) && \text{(triangle inequality)} \\
&\leq \left\|M^{-1}y^*\right\| \cdot \left\|M^{-1}y'\right\| + \left(M^{-1}b\right)^\top M^{-1}\,(y^* - y) \\
&\leq \lambda_M^2 C \cdot \sqrt{2\,(1-\gamma)} + \left\|M^{-1}b\right\| \cdot \left\|M^{-1}\,(y^* - y)\right\| && \text{(Cauchy-Schwarz)} \\
&\leq \lambda_M^2 C \cdot \sqrt{2\,(1-\gamma)} + \lambda_M\,\|b\| \cdot \left\|M^{-1}\,(y^* - y)\right\| && \text{(spectral rad. def.)} \\
&= \lambda_M^2 C \cdot \sqrt{2\,(1-\gamma)} + \lambda_M^2\,\|b\| \cdot \sqrt{2C\,(1-\gamma)} && \text{(law of cosines)}
\end{aligned}
$$

From our lower bound on C we then have:

$$W(y^*; b, A) - W(y; b, A) \leq 2\lambda_M^2 C\sqrt{2(1-\gamma)}$$

Further, we have that

$$W(y^*; b, A) \geq \frac{1}{2}\lambda_M^2 C,$$

as this is obtainable even if $b = \mathbf{0}$ by letting y^* be the first eigenvector of M^{-1} associated with the spectral radius λ_M. Recall that social welfare at equilibrium is increasing in b. We can then give a multiplicative bound on utility loss:

$$\begin{aligned}\frac{W(y^*; b, A) - W(y; b, A)}{W(y^*; b, A)} &\leq \frac{2\lambda_M^2 C\sqrt{2(1-\gamma)}}{\frac{1}{2}\lambda_M^2 C} \\ &= 4\sqrt{2(1-\gamma)}\end{aligned}$$

and so

$$\frac{W(y; b, A)}{W(y^*; b, A)} \geq 1 - 4\sqrt{2(1-\gamma)}.$$

□

Proof: [Proof of Lemma 3.4.1] Let v be an eigenvector of $\overline{A}$, and let λ be the corresponding eigenvalue. It suffices to show that for any v and any i, k, if $\lambda \neq 0$, then:

$$\frac{v_i}{v_k} = \frac{w_i}{w_k}. \tag{C.2}$$

In $G(w)$ graphs, an edge between vertices i and j is constructed with a probability proportional to $w_i w_j$. As such, for any i, j, k it holds that:

$$\frac{\overline{A}_{ij}}{\overline{A}_{kj}} = \frac{w_i}{w_k}. \tag{C.3}$$

By the eigenvalue equation, for any i,

$$\sum_{j=1}^{n} \overline{A}_{ij}\, v_j = \lambda v_i,$$

and so for any i, k, by (C.3),

$$\begin{aligned}\lambda \cdot \frac{w_k}{w_i} v_i &= \frac{w_k}{w_i} \sum_{j=1}^{n} \overline{A}_{ij} v_j \\ &= \sum_{j=1}^{n} \overline{A}_{kj} v_j \\ &= \lambda v_k\end{aligned}$$

which implies (C.2) when λ is non-zero. □

Proof: [Proof of Lemma 3.4.2] This follows from Lemma C.1.1 with $\kappa = 0$. We also give a simpler direct alternate proof here, which may be of independent interest.

First, we observe that

$$\|Av(B)\| \geq \|Bv(B)\| - \|B - A\| \cdot \|v(B)\| = \mu_1 - \eta, \tag{C.4}$$

by the triangle inequality. We can write $v(B) = \phi_1 v(A) + \phi_2 x$, where x is a unit vector orthogonal to $v(A)$, and $\phi_1^2 + \phi_2^2 = 1$, for some values ϕ_1 and ϕ_2. Since $x \perp v(A)$, its image under A can at most have a magnitude of $\lambda_2 = 0$. That is,

$$\|Ax\| = 0.$$

From our assumption, we can write

$$\begin{aligned}\|Av(B)\| &= \|\phi_1 Av(A) + \phi_2 Ax\| \\ &= \|\phi_1 Av(A)\| \\ &= \lambda_1 \phi_1.\end{aligned}$$

Combining with (C.4), we get that

$$\frac{\mu_1 - \eta}{\lambda_1} \leq \phi_1.$$

We can then conclude that

$$\begin{aligned}
\|v(A) - v(B)\| &= \|(1-\phi_1)v(A) + \phi_2 w\| \\
&= \sqrt{(1-\phi_1)^2 + \phi_2^2} \\
&= \sqrt{1 - 2\phi_1 + (\phi_1^2 + \phi_2^2)} \\
&= \sqrt{2(1-\phi_1)} \\
&\leq \sqrt{2\left(1 - \frac{\mu_1 - \eta}{\lambda_1}\right)}.
\end{aligned}$$

□

Proof: [Proof of Lemma 3.4.3] Let D be the matrix that has the same diagonal entries as $\overline{A}$ and 0 everywhere else. First we observe that

$$\left\| \overline{A} - (\overline{A} - D) \right\| \leq 1,$$

as all entries of D are bounded by 1. Whether or not we are deleting self-loops, by applying the triangle inequality to the previous observation and Lemma C.3.1,

$$\left\| A - \overline{A} \right\| \leq \sqrt{4w_1 \log(2n/\delta)} + 1,$$

which implies again by triangle inequality that

$$\|A\| \geq \tilde{d} - \sqrt{4w_1 \log(2n/\delta)} - 1.$$

By our distributional assumption,

$$\tilde{d} \geq \frac{256\left(\sqrt{4w_1 \log(2n/\delta)} + 1\right)}{\epsilon^2}$$

By Lemma 3.4.2, it follows that

$$\begin{aligned}
\left\| v_1(A) - v_1(\overline{A}) \right\| &\leq \sqrt{2\left(1 - \frac{\|A\| - \left\|A - \overline{A}\right\|}{\left\|\overline{A}\right\|}\right)} \\
&= \sqrt{2 \cdot \frac{\left\|\overline{A}\right\| - \|A\| + \left\|A - \overline{A}\right\|}{\left\|\overline{A}\right\|}} \\
&\leq 2\sqrt{\frac{\left\|A - \overline{A}\right\|}{\left\|\overline{A}\right\|}} \\
&\leq 2\sqrt{\frac{\sqrt{4w_1 \log(2n/\delta)} + 1}{\tilde{d}}} \\
&\leq 2\sqrt{\frac{\epsilon^2}{256}} \\
&= \epsilon/8.
\end{aligned}$$

□

Proof: [Proof of Lemma 3.4.4] The second-largest absolute eigenvalue can be defined as

$$\begin{aligned}
\lambda_2 &= \sup_{\substack{x \perp v_1\left(\overline{A} - D\right) \\ \|x\| = 1}} \left\| \left(\overline{A} - D\right) x \right\| \\
&\leq \sup_{\substack{x \perp v_1\left(\overline{A} - D\right) \\ \|x\| = 1}} \left\| \overline{A}\, x \right\| + \sup_{\substack{x \perp v_1\left(\overline{A} - D\right) \\ \|x\| = 1}} \|Dx\| \qquad \text{(C.5)}
\end{aligned}$$

Any vector x that is perpendicular to $v_1\left(\overline{A}-D\right)$ can be written as

$$x = \phi_1 v_1\left(\overline{A}\right) + \phi_2 u$$

Where u is perpendicular to $v_1\left(\overline{A}\right)$, and $\phi_1^2 + \phi_2^2 = 1$. Let $d := v_1\left(\overline{A}-D\right) - v_1\left(\overline{A}\right)$. Since $\overline{A}$ has only one non-zero eigenvalue, we can apply Lemma 3.4.2 to get

$$\|d\| \leq \sqrt{2\left(1 - \frac{\mu_1 - \eta}{\lambda_1}\right)} \tag{C.6}$$

Where $\mu_1 = \left\|\overline{A}-D\right\|$, $\lambda_1 = \left\|\overline{A}\right\|$, $\eta = \|D\|$. We have, from the triangle inequality,

$$\left\|\overline{A}\right\| - \|D\| \quad \leq \quad \left\|\overline{A}-D\right\|$$

Or,

$$\begin{aligned} \lambda_1 - \eta &\leq \mu_1 \\ \implies 1 - \frac{\mu_1 - \eta}{\lambda_1} &\leq \frac{2\eta}{\lambda_1} \end{aligned}$$

Putting the values for $\eta = \frac{w_1}{\sum_{i=1}^n w_i}$, $\lambda_1 = \frac{\sum_{i=1}^n w_i^2}{\sum_{i=1}^n w_i}$, and using (C.6), we get

$$\|d\| \leq \frac{2w_1}{\|w\|}$$

Since, x is, by definition perpendicular to $v_1\left(\overline{A}-D\right)$, we can write

$$\begin{aligned} v_1\left(\overline{A}-D\right)\cdot x = 0 &\implies d\cdot x + v_1\left(\overline{A}\right)\cdot x = 0 \\ &\implies \phi_1 = v_1\left(\overline{A}\right)\cdot x = -d\cdot xb \end{aligned}$$

Using Cauchy-Schwarz, we get that $|\phi_1| \leq \|d\| \leq \frac{2w_1}{\|w\|}$. Thus,

$$\overline{A}\, x = \phi_1 \lambda_1 v_1\left(\overline{A}\right) + \phi_2 \cdot 0 \qquad \left(\because u \perp v_1\left(\overline{A}\right) \implies u \in \ker\left(\overline{A}\right)\right)$$
$$\implies \left\|\overline{A}\, x\right\| \leq \frac{2w_1\lambda_1}{\|w\|}$$

Combining with (C.5), and noting the fact the eigenvalues of D are all less than 1, we get the result.

□

Proof: [Proof of Theorem 3.4.1] If the stated conditions hold, then with probability at least $1-\delta$, all desired bounds on eigenvalues and norms are obtained simultaneously by Theorem C.3.1. First we see that this implies a constant-factor separation between λ_1 and λ_2 for the realized adjacency matrix. This ensures that $\beta\lambda_1 < 1$ by our condition on β. From the proof of Lemma 3.4.3 we know that

$$\lambda_1 \geq \tilde{d} - \sqrt{4\, w_1 \log(2n/\delta)} - 1,$$

and from Lemma 3.4.5, we know that

$$\begin{aligned}\lambda_1 &\leq \frac{2w_1\tilde{d}}{\|w\|} + \sqrt{4\, w_1 \log(2n/\delta)} + 1 \\ &\leq \frac{\tilde{d}}{3} + \sqrt{4\, w_1 \log(2n/\delta)} + 1.\end{aligned}$$

For any $\epsilon < 1$ and $\delta < \frac{1}{2}$ this implies that $\lambda_1 > \frac{5}{6}\tilde{d}$ and $\lambda_2 < \frac{5}{12}\tilde{d}$, and so $\frac{\lambda_1}{2} > \lambda_2$. We can use this to show a bound on the spectral gap of M^{-1} which allows us to apply Proposition C.3.1:

$$\begin{aligned}
\left(\frac{\alpha_2}{\alpha_1 - \alpha_2}\right)^2 &= \left(\frac{1/(1-\beta\lambda_2)^2}{1/(1-\beta\lambda_1)^2 - 1/(1-\beta\lambda_2)^2}\right)^2 \\
&= \left(\frac{1}{\left(\frac{1-\beta\lambda_2}{1-\beta\lambda_1}\right)^2 - 1}\right)^2 \\
&\leq \left(\frac{1}{\left(\frac{1-0.5\beta\lambda_1}{1-\beta\lambda_1}\right)^2 - 1}\right)^2 && (\lambda_1 \geq 2\lambda_2) \\
&\leq \frac{1}{(\beta\lambda_1)^2}. && \text{(C.7)}
\end{aligned}$$

where the final inequality holds by noting that equality holds only at $\beta\lambda_1 = 1.25$, and that (C.7) is larger for all $\beta\lambda \in (0, 1)$. Thus, if

$$\begin{aligned}
C &\geq \frac{2\|b\|^2}{(\epsilon/(8\sqrt{2}))^2} \cdot \left(\frac{\alpha_2}{\alpha_1 - \alpha_2}\right)^2 \\
&= \frac{256\|b\|^2}{\left(\beta\tilde{d}\epsilon\right)^2},
\end{aligned}$$

then $\rho(y^*, v_1(A)) \geq \sqrt{1 - (\epsilon/(8\sqrt{2}))^2}$ by Proposition C.3.1. Applying the inequality $\sqrt{1-x} \geq 1 - x$ for $x \in [0, 1]$, we have that $\rho(y^*, v_1(A)) > 1 - (\epsilon/(8\sqrt{2}))^2 = 1 - \epsilon^2/128$. Scaling y^* to the unit vector $\frac{y^*}{\|y^*\|}$ and applying the law of cosines, we get that $\left\|\frac{y^*}{\|y^*\|} - \frac{v_1(A)}{\|v_1(A)\|}\right\| \leq \epsilon/8$. From Lemma 3.4.3, we also have that $\left\|v_1(A) - \frac{w}{\|w\|}\right\| \leq \epsilon/8$. By the triangle inequality this gives us that $\left\|\frac{y^*}{\|y^*\|} - \frac{w}{\|w\|}\right\| \leq \epsilon/4$. Applying Lemma 3.3.1 and law of cosines gives a competitive ratio of $1 - \epsilon$. □

Proof: [Proof of Lemma 3.4.6] When considering $G(n, p)$ graphs where self-loops are added with probability p, this is equivalent to the $G(w)$ graph distribution with $w_i = np$ for all i. We give sufficient conditions for applying Lemma 3.4.3, which holds even if we do not allow self-loops.

We can compute the second-order average expected degree as:

$$\begin{aligned}\tilde{d} &= \frac{\sum_i w_i^2}{\sum_i w_i} \\ &= \frac{n^2 p^3}{np^2}\end{aligned}$$

and then can see that

$$\begin{aligned}\frac{256\left(\sqrt{4w_1 \log(2n/\delta)} + 1\right)}{\epsilon^2} &\leq \frac{256\sqrt{6w_1 \log(2n/\delta)}}{\epsilon^2} \\ &\leq \frac{768\sqrt{6}}{\epsilon^2} \cdot \sqrt{n}p, \qquad\qquad (p \geq \frac{4}{9}\log(2n/\delta))\end{aligned}$$

and so whenever n is at least $(786^2 \cdot 6/\epsilon^4)$, we will have that $\tilde{d} = np$ is sufficiently large. We already have the required bound on the maximum expected degree, which in fact holds for all expected degrees, and so we can directly apply Lemma 3.4.3 to obtain the result. □

Proof: [Proof of Theorem 3.4.2] In order to apply Theorem 3.4.1, we simply need to show that the spectral gap is not too small, which previously followed from the condition that $w \leq \frac{\|w\|}{6}$. If n is at least 36, this will hold for the $G(w)$ interpretation of $G(n, p)$ graphs (prior to removing the diagonal), as $\|w\| = n^{1.5}p$, so then $\frac{\|w\|}{w} = \sqrt{n} \geq 6$. □

Proof: [Proof of Lemma 3.4.7] In [82] it is proven that whenever $\sigma \in (2, 2.5)$, the expected largest eigenvalue of a random power law graph is $\Theta\left(w_1^{3-\sigma}\right)$, where w_1 is the highest degree. If w_1 is at least $\Omega\left(\left(\frac{\log(2n/\delta}{\epsilon^4}\right)^{\frac{1}{5-2\sigma}}\right)$, then the largest eigenvalue will be $\Omega\left(\frac{\sqrt{w_1 \log n}}{\epsilon^2}\right)$. With appropriate constants, we obtain the required bound on $\tilde{d} = \lambda_1(\overline{A})$ to ensure that deviations from expectation are sufficiently small such that we can apply Lemma 3.4.3. □

Proof: [Proof of Theorem 3.4.3] By Lemma 3.4.7, the first eigenvector of the graph is sufficiently

close to its expectation. For power law graphs larger than some constant, $\frac{w}{\|w\|} < \frac{1}{6}$ will hold, and so the second eigenvalue will be small enough such that, when applying Theorem 1 from [83], a spectral gap factor of $1/2$ will be obtained almost surely. The bound above gives us a constant factor spectral gap for sufficiently large constants, which allows us to obtain the same budget lower bound as in Theorem 3.4.1. We can then combine these results as in Section 3.4 to obtain the theorem. □

Proof: [Proof of Lemma 3.5.1] If cosine similarity is small, much of the mass of the intervention is in components orthogonal to y^*, whose efficacy can be upper-bounded by the second eigenvalue of M^{-1}. This bound will depend on the first and second eigenvalues of βA. For some ϕ_1, ϕ_2 where $\phi_1^2 + \phi_2^2 = 1$, and for some unit vector u orthogonal to y^*, we can write:

$$\begin{aligned}
y &= \phi_1 y^* + \phi_2 \sqrt{C} u \\
\therefore W(y) &= \left\| M^{-1} y \right\|^2 = \left\| \phi_1 M^{-1} y^* + \phi_2 \sqrt{C} M^{-1} u \right\|^2 \\
&= \phi_1^2 \left\| M^{-1} y^* \right\|^2 + C\phi_2^2 \left\| M^{-1} u \right\|^2 + 2\phi_1\phi_2 \left(M^{-1} y^* \right)^\top \left(M^{-1} u \right) \\
\therefore \frac{W(y)}{W(y^*)} &\leq \phi_1^2 + \phi_2^2 \frac{\alpha_2}{\alpha_1} + 2\phi_1\phi_2 \sqrt{\frac{\alpha_2}{\alpha_1}} \\
&\leq \gamma^2 \left(1 - \frac{\alpha_2}{\alpha_1} \right) + \frac{\alpha_2}{\alpha_1} + 2\sqrt{\frac{\alpha_2}{\alpha_1}}
\end{aligned}$$

□

Proof: [Proof of Lemma 3.5.2] From the $G(w)$ version of the (ϵ, δ)-concentration condition, we have that

$$\frac{\sqrt{w_1 \log(n/\delta)}}{\epsilon^2} \leq O\left(\frac{\|w\|_2^2}{\|w\|_1} \right)$$

which implies that

$$\begin{aligned}\sqrt{nw_1 \log(n/\delta)} &\leq O\left(\epsilon^2 \|w\|_2\right) \\ &\leq O\left(\epsilon \|w\|_2\right)\end{aligned}$$

as $\|w\|_1 \leq \sqrt{n}\,\|w\|_2$ for any vector of length n. The difference in degree vectors $\|w - w^*\|$ can be expressed as

$$\begin{aligned}\left\|\overline{A}\,\mathbf{1} - A\mathbf{1}\right\| &\leq \left\|\overline{A} - A\right\| \cdot \|\mathbf{1}\| \\ &\leq O\left(\sqrt{nw_1 \log(n/\delta)}\right),\end{aligned}$$

where the last line follows from Lemma C.3.1 and the (ϵ, δ)-concentration conditions. From the analysis of Theorems 3.3.1 and 3.4.1, we can see that approximating the first eigenvector of A to within $O(\epsilon)$ ℓ_2 distance is sufficient for near-optimality, which we have for the true first eigenvector by triangle inequality with w. □

Proof: [Proof of Lemma 3.5.3]

To see that there will be a giant component containing almost all vertices, note that this follows from analysis of the giant component in $G(n, p)$ graphs with $p = \frac{1}{\epsilon n}$, as all probabilities in $\overline{A}$ are at least this large. A $G(n, p)$ graph with $p > \frac{1}{n}$ almost surely has a giant component of size $\Theta(n)$. Consider the sampling of all edges in such a graph except for those incident on vertex i. The expected number of edges from i to the $\Theta(n) = n^*$ size component will still be $\Theta(1/\epsilon)$. Each edge is included or excluded independently, and so the probability that no edges are included is at most $\left(1 - \frac{O(1/\epsilon)}{n^*}\right)^{n^*} \leq \exp\left(-O(1/\epsilon)\right)$. As this holds for each vertex, the expected number of vertices outside the giant component is $n \cdot \exp\left(-O(1/\epsilon)\right)$. A will be a block diagonal matrix with a block for each component, and the largest eigenvalue will be associated with this giant component. We concern ourselves only with this component in our mixing time analysis.

The mixing time of a graph A can be analyzed using the second eigenvalue of its diffusion

matrix. The first eigenvalue of P will be 1, and from Proposition 1 in [101], it follows that random walks on connected graphs have mixing time of $O(\log(n/\epsilon))$ to within additive probability $2\epsilon/n$ for each vertex, and thus ϵ total variation distance, if $\lambda_2(P)$ is bounded by a constant factor below 1.

Observe that the second eigenvalue of $\overline{P} = \overline{A}\,\overline{D}^{-1}$ is 0, as $\overline{P}$ has rank 1. A and D will be close in spectral norm to their expectations for graphs satisfying (ϵ, δ)-concentration, and the same will hold for D^{-1} when the minimum degree is large enough. Note that $n = \Omega\left(\frac{\sqrt{w_1 \log(n/\delta)}}{\epsilon^2}\right)$ by the (ϵ, δ)-concentration conditions, as the first eigenvalue of $\overline{A}$, which will always be less than n, must be at least this large. Given the lower bound on w_1, we then have that $n = \Omega\left(\frac{\log(n/\delta)}{\epsilon^2}\right)$, with a constant larger than 100. We can then apply Hoeffding's inequality to see that the true degree of each vertex will be within $\frac{1}{2\epsilon}$ of its expectation, simultaneously with probability at least $1 - O(\delta)$. As such, for any vector v, the standard basis components of $D^{-1}v$ and $\overline{D}^{-1}$ are all within a factor of 2. We have already seen that when a $G(w)$ graph $\overline{A}$ satisfies (ϵ, δ)-concentration, Av will be close to the projection of v onto $v_1(A)$, as $\lambda_i(A) \ll \lambda_1(A)$ for $i > 1$. Together with the previous observation, this implies that the image of any vector will be close under P and $\overline{P}$. Thus, with high probability $\lambda_2(P)$ will be bounded by a constant less than 1, giving us our desired mixing time.

□

Proof: [Proof of Lemma 3.5.4] Our proof will proceed using Theorem 5 from [83] for bounding the deviation of sums of Hermitian random matrices. $\hat{A}$ can be viewed as a sum of $O(m^2)$ independent matrix random variables, with one variable for each pair of groups. For each pair of groups, the matrix $\hat{A}$ with zeros for all other group pair entries is symmetric and Hermitian, and the sum of all of these matrices is $\hat{A}$.

First, we show that each of these individual block matrices $\hat{A}^{ij}$ has an empirical frequency close to its expectation. By considering random variables for each of the $O(n^2/m^2)$ edges and applying Chernoff bounds, we have that

$$\left|\hat{p}_{ij} - p_{ij}\right| \leq O\left(\frac{m}{n}\sqrt{\log(m/\delta)}\right)$$

for all groups with probability $1 - \delta/2$.

To apply the theorem, we need to obtain a bound on $\left\|\overline{A}^{ij} - \hat{A}^{ij}\right\|$, the expected and empirical block matrices for each group. All $O(n^2/m^2)$ non-zero entries in such a matrix equal will be, and will be bounded by $O(m/n \cdot \sqrt{\log(m/\delta)})$ as we have just seen. The first eigenvector for such a matrix will have $O(n/m)$ non-zero entries, each $O(\sqrt{m/n})$, and its product with the matrix will again have $O(n/m)$ non-zero entries, each of which will be $O(\sqrt{m/n \cdot \log(m/\delta)})$. Thus the spectral norm, equivalent to the corresponding first absolute eigenvalue, will be $O(\sqrt{\log(m/\delta)})$.

Further, we need to bound the norm of the sum of the variance matrices to apply the theorem, which we denote $V^2 = \left\|\sum_{ij} \text{var}(\hat{A}^{ij})\right\|$. For a matrix A, $\text{var}(A) = \mathbb{E}\left[(A - \mathbb{E}[A])^2\right]$. Each entry in $(\hat{A}^{ij} - \overline{A}^{ij})^2$ will be $O(k(p_{ij} - \hat{p}_{ij})^2)$ where k is the number of edges in the block. The expected value of this is simply $k \cdot \text{var}(\hat{p}_{ij}) = p_{ij}(1 - p_{ij})/k$. Summing over all block variance matrices gives us a matrix where all entries (other than the diagonal if self-loops are ignored) are $O(m^2/n^2)$; this matrix is n-dimensional, and so the spectral norm will be $O(m^2/n)$.

By Theorem 5 from [83], with $K = O\left(\max\left(\frac{m\sqrt{\log(n/\delta)}}{\sqrt{n}}, \log^2(n/\delta)\right)\right)$:

$$\begin{aligned}
\Pr\left[\left\|\hat{A} - \overline{A}\right\| > K\right] &\leq O\left(n \cdot \exp\left(-\frac{K^2}{m^2/n + K \cdot \log(m/\delta)}\right)\right) \\
&= O\left(n \cdot \exp\left(-\frac{K \log(n/\delta)}{K}\right)\right) \\
&\leq \delta/2. && \text{(approp. constants)}
\end{aligned}$$

This completes the proof. □

C.2 Analyzing Best Response Dynamics

We can see that the social welfare of the game acts as a potential function when all agents' actions are below equilibrium levels, and so we should expect selfish agents to reach the new equilibrium after applying our intervention.

Theorem C.2.1 (Convergence of Best Response Dynamics) *When $\beta > 0$, $b > 0$ and $a_0 < a^*$, repeated best responses of agents will converge to equilibrium.*

Proof: [Proof of Theorem C.2.1] Suppose β and all b_i values are positive, and all agents start with initial action levels $a_i = 0$. This is below the optimal action level for all agents, and each agent's utility function is concave upon fixing all other action values. When each agent best responds, they will increase their action level towards the optimum value. Because β is positive, these increases can only improve the welfare of all other agents. Social welfare increases upon every best response, and thus best response dynamics will converge. The optimal intervention $\beta > 0$ will always be non-negative for the graphs we consider, as the social welfare is an increasing function of b. We can again see that Best Response Dynamics would result in asymptotic convergence to the new Nash equilibrium after updating the b_i values; all updates will be non-negative, and thus the social welfare of the network remains a potential function for the game. If agents only best-respond by more than ϵ in a given round, convergence will be polynomial in $\frac{1}{\epsilon}$ and other relevant parameters. □

C.3 Further Preliminaries

C.3.1 Eigenvalue Transformations

We note some observations about the eigenvalues of graphs which satisfy the assumptions from Section 3.2. For a graph A, eigenvalue λ_i corresponds to eigenvalue $\frac{1}{1-\lambda_i}$ in $(I - A)^{-1}$ and $\frac{1}{1-\beta\lambda_i}$ in $(I - \beta A)^{-1}$; corresponding eigenvectors will be identical for both A and $(I - \beta A)^{-1}$. We will refer to this latter matrix $(I - \beta A)^{-1}$ as M^{-1}. Given that the spectral radius of βA is less than 1, $\frac{1}{1-\beta\lambda_i}$ is decreasing in λ_i, and so the eigenvalues of M^{-1} are ordered according to their corresponding eigenvalues in A. Further, all eigenvalues of M^{-1} will be positive. It follows that the spectral radius of M^{-1}, which we denote by λ_M, is $\frac{1}{1-\beta\lambda_1(A)}$.

C.3.2 Imported Results

For completeness, we restate important results which we use in our analysis. A key proposition from [1] shows that in the setting we consider, for large enough budgets, the optimal intervention for a graph is close in cosine similarity to the first eigenvector.

Proposition C.3.1 (Proposition 2 in [1]) *Suppose A is symmetric, the spectral radius $\lambda_1(A)$ is less than $1/\beta$, and $\beta > 0$. Then for any $\epsilon > 0$, if*

$$C > \frac{2\|b\|^2}{\epsilon}\left(\frac{\alpha_2}{\alpha_1 - \alpha_2}\right)^2$$

then $\rho(y^*, \sqrt{C}v_1(A)) > \sqrt{1-\epsilon}$.

We also state a key theorem about the spectra of random graphs.

Theorem C.3.1 (Theorem 1 in [83]) *For a random graph with edges constructed independently according to $\bar{A}$ and maximum expected degree $\Delta \geq \frac{4}{9}\ln(2n/\delta)$, with probability $1-\delta$ for sufficiently large n,*

$$|\lambda_i(A) - \lambda_i(\bar{A})| \leq \sqrt{4\Delta \ln(2n/\delta)}$$

for all $1 \leq i \leq n$.

As noted in [86], implicit from the proof of Theorem C.3.1 in [83] (where it is Theorem 1) is a deviation bound on the matrix norm. The result holds for all random graphs with independent edges, and we restate it here for the special case of $G(w)$ graphs.

Lemma C.3.1 (Restatement of Theorem 1 in [83]) *For a random graph with independent edges A drawn from $\overline{A}$, if the maximum expected degree $d_{\max}$ is at least $\frac{4}{9}\log(2n/\delta)$, then with probability at least $1-\delta$ it holds that*

$$\left\|A - \overline{A}\right\| \leq \sqrt{4d_{\max}\log(2n/\delta)}.$$