A Strategic Agent-Based Analysis of Economic and Technological Changes in Financial Networks

by

Katherine Mayo

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2024

Doctoral Committee:

Professor Michael P. Wellman, Chair
Professor Peter Adriaens
Professor Atul Prakash
Assistant Professor Jeffery Zhang

Katherine Mayo

kamayo@umich.edu

ORCID iD: 0000-0002-8142-1856

For Mama and Baba and all we can ever know

ACKNOWLEDGMENTS

Nothing significant is ever achieved alone. The past five years were made possible by the support of wonderful colleagues, friends, and family.

First, I must thank my advisor Michael Wellman, without whom none of this would be possible. His steadfast guidance and support throughout this challenging process allowed me to flourish. Mike's invaluable feedback could always be counted on to ensure my work was of the highest caliber. I greatly appreciated his patience, and the space he gave me to explore my interests and to run with my ideas. It was an honor and a privilege to work with Mike and I am a better researcher because of his immeasurable influence.

Thank you also to the members of my dissertation committee: Peter Adriaens, Atul Prakash, and Jeffery Zhang. Their feedback and comments throughout the proposal and dissertation process undoubtedly made this dissertation stronger. The collaboration with Danial Dervovic, Sumitra Ganesh, Alec Koppel, and Haibei Zhu from the JP Morgan Chase AI Research team was very beneficial for developing and improving the final project of this dissertation.

My time in the Strategic Reasoning Group was greatly enhanced by working alongside some incredibly talented lab mates including Frank Cheng, Xintong Wang, Megan Shearer, Max Smith, Zun Li, Yongzhao Wang, Christine Konicki, Madelyn Gatchel, Austin Nguyen, Chris Mascioli, Therese Nkeng, and Arshdeep Singh. I was also fortunate to work with a stellar research scientist, Mithun Chakraborty. I could always count on any of these individuals to help me work through problems or to provide constructive feedback on practice talks, but their most important contribution was making this stressful time more enjoyable. I will forever miss the chaos of Thursdays in BBB 3945. A special thank you to Frank, Megan, and Xintong whose early guidance was crucial in getting my first research project off the ground. Special thanks also to Madelyn for her endless support, feedback, and encouragement in my later years. Her friendship has meant more to me than she will ever know. And to my fellow Minions: we'll always have Haymaker trivia night.

I am grateful to have also worked with some incredibly talented undergraduate students over the past few years, both in research and in the classroom. Thank you to my research assistants Shaily Fozdar and Nicholas Grabill for their tireless efforts in pursuit of my research ideas. Additionally, thank you to the undergraduate students who let me teach them the wonders of Python once a week for an entire semester. Teaching EECS 110 pushed me way outside of my comfort zone, but these students made coming to class each week such a joy and turned this experience into an unexpected highlight of my graduate school career. A big thank you also to Laura Burdick for offering me this opportunity, and to my IA Sarah Kurata for taking this leap with me.

I also want to take this opportunity to thank the College of Engineering and CSE administrative staff for keeping everything running smoothly. A special thanks to Kimberly Clark and Cindy Estell for making sure I got my 30 minutes each week with Mike, no matter how hectic our schedules.

These past five years I have received a tremendous amount of support from many family and friends outside of the Michigan bubble. Thank you especially to Amelia Hecsh, David Jiang, Spencer McEwen, Verity Scheel, Adam Keester, and Mishy Wang. Aside from patiently listening to my graduate school woes, they provided me a much needed lifeline to a world outside of research. A reminder that there is more to life than publish–or–perish and reviewer 2's comments. I also want to thank Tom and Monica Egan for their unwavering support, encouragement, and genuine interest in my research.

Last, but certainly not least, I want to thank my family. I owe everything I am to my parents. They nurtured my curiosity, my independence, and my ambition that have gotten me this far. Their endless support, not only these past five years, but in everything I do allows me to move mountains. Thank you to my father, the original Dr. Mayo, for inspiring me to embark on this journey. Thank you to my siblings, Liz and James, who have always believed in me (sometimes more than I believe in myself), but are never afraid to knock me down a peg or two when needed. Their support in everything I do gives me the confidence to do hard things. Finally, thank you to my husband Alex, who has been there for every high and every low of the last five years. My victories were sweeter and my trials less arduous because I got to share them with him. I am fortunate to have a partner willing to make the sacrifices required for me to follow my dreams.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Economic events and advancements in technology have drastically transformed the financial system over the past 20 years. This includes the implementation of new policies post-2008 financial crisis, as well as new methods for customers to engage with the system from cryptocurrency to faster processing payments. This system is a complex network, vital for connecting individuals, businesses, and financial institutions for the purposes of monetary transactions. Thus, it is important to carefully consider how changes may impact and transform the system in the future. This dissertation applies a computational approach to study strategic decisions that arise from economic and technological changes in financial networks. I introduce the *extended financial credit network* model capable of expressing diverse financial scenarios and enabling the creation of agent-based models for studying strategic interactions. To study the strategic decisions of agents within the model, I apply methods from empirical game-theoretic analysis to identify equilibrium behavior.

I apply these methods to four case studies of strategic behavior in financial networks. First, I analyze *portfolio compression*, a method for eliminating cycles of debt, as a strategic decision. Compressing cycles offers both potential benefit and harm to the network, and banks must weigh both possibilities carefully. I find that the compression decision is best made using simple, local information available to banks. Further, I note the importance of the recovery rate of insolvent banks in the perceived affect of cycles on systemic risk. Second, I investigate the strategic *use of costly fraud detection systems* by banks. I observe the behavior of banks is subject to the relative, not specific, capabilities of their fraud detectors. In particular, those with strong detectors are better able to adjust their behavior in response to rising costs due to the existence of weaker banks in the system. The third study addresses *bank allowance of real-time payments by customers*. Customer overdrafts pose a potential credit risk to banks, who assume short-term liability, and lead to strategic decisions regarding which customers should be allowed use of these new payments. My analysis shows banks choose to allow most, though not all, customers to send real-time payments. I discover the strategic choice of banks will not lead to the socially optimal outcome in this scenario. Lastly, I extend the previous two works to study banks' strategic *mitigation of fraud risk in real-time payments*. In particular, I focus on the strategic trade-off banks make between restricting customer access to these payments and investing in costly fraud detection. I find banks value the ability to control customer access, though they never invoke overly strict restrictions. Instead, they balance some constraints with the use of fraud detection. I observe these strategic measures limit the negative effects from fraudulent actors with minimal disruption to customers.

Broadly, this dissertation demonstrates the effectiveness of agent-based modeling and empirical game-theoretic analysis to gain insight into strategic interactions in financial networks. It also illustrates the flexibility of the extended financial credit network model, which is employed in all four studies. Finally, I establish the new strategic feature gains assessment, a method for assessing the benefit of strategies in the strategy space. The assessment uncovers valuable insights into strategic behavior in the studies on portfolio compression and fraud risk in real-time payments.

CHAPTER 1

Introduction

1.1 Motivation

The financial system is a vast, complicated network connecting individuals, businesses, and financial institutions (e.g., banks) through monetary transactions. The network itself is composed of a web of underlying subsystems and services, each designed to meet various financial needs (e.g., payments, investments). In today's global economy, the financial system plays an increasingly critical role in connecting people and businesses. It is also constantly evolving as economic and technological forces impact the behavior of participants and the services offered by the system.

Economic pressures may lead network participants to alter their behavior and adopt new principles to guide their interactions. In the wake of the 2008 recession, many financial institutions adjusted how they conducted their business, seeking better ways to manage their risk [Yellen, 2013, Schuldenzucker and Seuken, 2021]. This included implementing new measures designed to decrease their exposure to other network participants and limit systemic risk. More recently, the COVID-19 pandemic was a source of global economic distress [Susskind and Vines, 2020, Padhan and Prabheesh, 2021]. Health and safety concerns forced mass reliance on card and digital-based payments, which may prove to be a permanent shift [Toh and Tran, 2020, Jonker et al., 2022]. As these changes are reactions to specific events, it is important to analyze how such behavioral changes will continue to affect the financial system in the future.

Advancements in technology have also been responsible for driving significant changes, especially in the payments landscape. The once popular cash and checks are increasingly being replaced by card-based methods of transaction [Board of Governers of the Federal Reserve System, 2023]. Mobile wallets, block-chain based payments, and faster processing payments are revolutionizing the system by offering new ways to complete transactions. Technological innovations also lead to improvements in existing infrastructure. For example, increased automation has lead to great improvements in timing and accuracy of bank fraud detection systems. These new technologies warrant further study to better understand how they complement and transform the existing financial system.

1.2 Studying Financial Networks

Studying the ever-evolving financial system, however, poses several challenges. Often security and privacy concerns lead to limited availability of real-world financial data for research outside of financial institutions. Absent of such concerns, data to address new or hypothetical scenarios may be limited or non-existent. Consider the goal of stress testing for a yet unseen economic shock or exploring the entrance of a new payment service. Historical data will only contain past behavior of participants and will not include how such changes may influence future behavior. In such situations, employing simulation-based methods to conduct the analysis may offer a solution.

Agent-based modeling is a computational approach that leverages simulation to study complex interactions. It can be a particularly effective method when relevant data is difficult to obtain, scarce, or non-existent and when the phenomena of interest are especially complex. Agent-based modeling has been used to simulate and study a variety of social systems including the spread of epidemic diseases in populations [Chopra et al., 2023, Hoertel et al., 2020], school choice [Diaz et al., 2019], and financial markets [Cheng and Wellman, 2017, Wellman and Wah, 2017, Wang et al., 2021]. This computational approach can also serve as a valuable complement to data-driven analyses that construct mathematical models of systems, when both approaches are feasible. In an agent-based model, one or more *agents* interact within an environment according to some defined sets of rules and actions. From these interactions, macro-level behaviors organically emerge which are studied to better understand how complex systems form and operate.

It may be that these behaviors are the result of strategic decisions made by the agents, which should be considered in the model. Complicating the analysis, the strategic decision made by one participant is often dependent on the strategic decisions of other participants. Game theory offers a mathematical framework for analyzing such strategic interactions. It defines an environment in which strategic players interact according to sets of strategies that govern their behaviors. The players derive utility based on personal preferences over outcomes that occur as a result of the strategic choices. Analyzing the environment involves identifying a stable point in which each player is employing the best strategy given the environment's circumstances and player beliefs about other participants' strategic decisions, referred to as an *equilibrium*. Unfortunately, classical game-theoretic reasoning requires an analytical model of the environment and interactions, which may be intractable when studying large, complex systems such as financial networks. For such applications, a method known as empirical game-theoretic analysis may be better suited. Empirical game theoretic-analysis employs simulation of strategies in an agent-based model to obtain a mapping of strategies to utility for outcomes, referred to as payoffs. The agents in the model are equivalent to players in a game theoretic-game and results of simulation are used to identify the game's equilibrium.

This dissertation combines methods from agent-based modeling and empirical game-theoretic analysis to gain insights into strategic decisions arising from economic and technological changes in financial networks. I extend the financial credit network model with the addition of financial node types allowing the model to differentiate between the behaviors of various types of network participants (e.g., banks, customers). This results in a more expressive model capable of capturing a wider variety of financial scenarios. The extended financial credit network enables the creation of agent-based models, by serving as an environment in which strategic agents may interact according to behavior defined by heuristic strategies. I employ methods from empirical game-theoretic analysis to identify the equilibrium behavior of these agents with the goal of addressing the following questions:

1. What factors drive the strategic decisions of financial network participants?

2. How do the decisions of strategic participants affect outcomes for the entire network?

More specifically, I analyze the equilibrium strategies to determine the information most beneficial to an agent making the strategic decision and how variation in environmental parameters may play a role. Once the equilibrium is identified, I seek to understand how agents following this behavior affects the network as a whole, as well as any non-strategic participants. I address these questions as they pertain to strategic debt compression, fraud detection, and real-time payments. An overview of the remaining chapters within this dissertation follows.

1.3 Dissertation Overview

Chapter 2 outlines the representations and methods employed for modeling and analyzing financial networks, as well as provides the necessary background information for understanding these methods. Section 2.1 explains the financial credit network model and introduces the extended financial credit network with financial node types. I provide details on game theoretic-reasoning and empirical game-theoretic analysis in Section 2.2. The particular method for conducting empirical game-theoretic analysis employed, Subgame Search, is discussed in Section 2.2.3. Finally, Section 2.3 introduces a new methodology for assessing the value provided by subsets of the strategy space referred to as the strategic feature gains assessment.

The next four chapters demonstrate use of the extended financial credit network and empirical game-theoretic analysis in four case studies of strategic decision-making in financial networks.

Chapter 3 explores portfolio compression, a method for eliminating cycles of debt in a network, as a strategic decision made by banks. Bank nodes strategically vote to accept or reject a potential compression, which is performed only if nodes on the cycle vote unanimously in favor. I find that the compression decision is best made using strategies based on simple, local network information rather than simply unconditionally accepting or rejecting a proposal. Furthermore, as measured by several metrics, a strategic compression vote does not negatively impact the network. By performing the strategic feature gains assessment, I uncover an important link between the recovery rate of insolvent nodes and the way banks perceive the potential effects of a cycle on systemic risk. The work reported in this chapter appears in the proceedings of the International Conference on AI in Finance [Mayo and Wellman, 2021].

Chapter 4 investigates strategic use of costly fraud detection systems for credit card payments. Bank nodes strategically determine if a payment should be *flagged* and sent to the fraud detector, while malicious actors attempt fraudulent payments by strategically exploiting weaknesses in the bank nodes. The analysis shows a strategic dependency between equilibrium behavior and a bank node's relative fraud detection capabilities. Banks with weaker fraud detection systems (*weak* banks) flag most payments regardless of fraud detection costs, while banks with stronger detection capabilities (*strong* banks) will take into account the rising costs of fraud detection. Strong banks flag most payments when costs are low, but switch to flagging only highly suspicious payments when costs rise. Malicious actors will always frequently attempt high value fraudulent payments. Analyzing the breakdown of costs experienced by banks shows that strong banks are better off when flagging strategically, while weak banks should flag all payments. Part of the contents of this chapter was presented at the AAAI workshop on Modeling Uncertainty in the Financial World [Mayo et al., 2023].

Chapter 5 studies banks' strategic allowance of real-time payments use by customers at risk of overdrafts. Bank nodes strategically set minimum thresholds on deposit holdings required for a customer to qualify to send these new payments. Analyzing the results, I note that banks select thresholds that allow most, though not all, customers to send real-time payments. Banks tend to increase allowance as the processing delay of standard payments increases or the likelihood of customer overdrafts decreases. I observe both customers and banks are better off, as measured by several outcomes, when all customers are allowed to send real-time payments. Thus, I conclude the strategic outcome in this situation is not socially optimal. The work presented in this chapter appears in the proceedings of the International Conference on AI in Finance [Mayo et al., 2021].

Chapter 6 extends work from the previous two chapters by focusing on banks' strategic allowance of real-time payments given the risk of fraud. Bank nodes strategically select maximum thresholds on payment values above which a payment cannot be sent in real time. Malicious actors attempt fraudulent payments in the network for which banks may be liable. My analysis finds bank nodes maintain high thresholds, but balance this with high investment in fraud detection when liable for fraud. The use of the strategic feature gains assessment highlights the importance of thresholds as a risk mitigation technique for banks. The malicious actors strategically exploit banks where they have historically been most successful committing fraud and generally attempt fraud using any payment type. I find the measures employed by banks to alleviate fraud risk drastically affect fraudulent payments with little disruption to customers. A portion of the work reported in this chapter will appear in the proceedings of the International Joint Conference on Artificial Intelligence [Mayo et al., 2024].

Chapter 7 concludes the dissertation with a summary of contributions and discussion.

CHAPTER 2

Network Representations and Methods for Analysis

This chapter describes the financial network representation and methods for analysis employed in this dissertation. First, I discuss the modeling of financial networks as a set of nodes connected by directed edges leading to the financial credit network (FCN) model. I introduce an extension to this FCN model by defining *financial node types* and illustrate how they allow for a more expressive model capable of capturing a wider variety of financial relationships. Then, I explain the main analysis method, empirical game-theoretic analysis (EGTA), and its connection to agent-based modeling. Finally, I introduce a novel use of EGTA to assess the benefit of strategies in a strategy space, which I refer to as the strategic feature gains assessment.

2.1 Financial Network Modeling

2.1.1 Financial Credit Networks

The ***credit network*** formalism was introduced as a way to represent a trust relationship between individuals in a network [DeFigueiredo and Barr, 2005, Ghosh et al., 2007, Dandekar et al., 2011] modeled as a graph. The nodes in the graph represent individuals participating in the network. Nodes are connected by directed edges referred to as ***credit edges*** representing the willingness of a node to accept IOUs from another node in the network. The weight on each edge represents the capacity for trust in terms of how many IOUs the node is willing to accept. An example of a relationship between two nodes, *A* and *B*, in a network is shown in Figure 2.1. Node *A* is willing to accept up to 60 IOUs from node B, while B is willing to accept up to 100 IOUs from A. The respective credit edges are represented in the figure as solid red arrows.

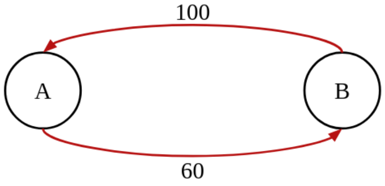


Figure 2.1: An example of a simple credit network with two nodes and two credit edges represented by the solid, red arrows.

This establishment of trust allows the nodes to call in ***favors*** from one another, such as the procurement of goods or rendering of services, in exchange for IOUs. For example, node A may give node B 10 IOUs in exchange for B providing some service to A. After the exchange, node A is now willing to do 10 additional IOUs worth of favors for node B in the future. We represent this transaction as an increase in the value on the credit edge from A to B of 10 IOUs. On the other hand, since A has called in 10 IOUs worth of favors, B is now willing to do 10 IOUs fewer favors in the future for A. This is captured by the decrease in the value on the credit edge from B to A by the 10 IOUs. After this exchange, the network appears as in Figure 2.2.

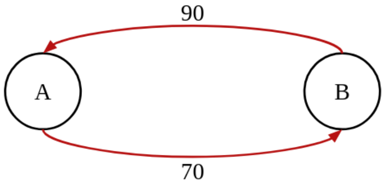


Figure 2.2: An example of a simple credit network after a transaction of 10 IOUs from node A to node B.

Such an exchange of IOUs represents a transaction in which one party incurs a debt of IOUs for a service (favor) with a party they trust to complete the transaction. For this reason, the credit network formalism offers a way to model and study various financial relationships. General properties of credit networks have been studied by Dandekar et al. [2011, 2015]. Credit networks have previously been used to study recommender systems [DeFigueiredo and Barr, 2005], multi-unit auctions [Ghosh et al., 2007], and informal borrowing [Karlan et al., 2009].

The credit network model was first expanded by Cheng et al. [2016] with the inclusion of interest rates on the network edges, creating the ***financial credit network***. Similar to the credit network formalism, a ***credit edge*** in an FCN represents willingness of one node to transact with another. In a financial network, this willingness is often due to the potential to charge interest rates. An FCN credit edge is specified by its ***capacity***, the amount of credit extended, and its ***interest rate***, the *minimum* rate demanded for drawing on the line of credit. An example of an FCN with solid red credit edges is shown in Figure 2.3. In this example, node A extends 50 units of credit to B with an interest rate of *at least* 10%. Meanwhile, C extends 60 units of credit to A with an interest rate of *at least* 20%.

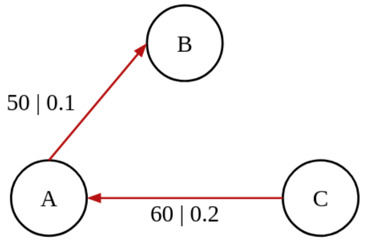


Figure 2.3: An example of credit edges in a financial credit network.

To incorporate interest rates, it is necessary for FCNs to distinguish between willingness to transact and existing transactions accomplished with the introduction of *debt edges*. A ***transaction*** takes place in the network when a node draws upon credit, creating a debt owed from one node to another. This debt is represented by a ***debt edge*** which specifies a ***value*** of the debt and the interest rate incurred. Continuing the FCN example, consider the scenario in which B borrows 20 units from node A. This is feasible due to the credit edge from A to B, though recall A demands an interest rate of at least 10%. Assume A and B agree to the minimum interest rate, drawing on the credit now means B owes a debt of 20 units with an interest rate of 10% to A in the future. This is modeled by the dashed blue debt edge from B to A of 20 units and an interest rate of 0.1 shown in Figure 2.4. Since B has drawn on some of the credit A is willing to extend, the additional credit A is willing to extend to B in the future must decrease by the amount already transacted. This is represented by the decrease in the capacity on the credit edge from A to B from 50 to 30 units.

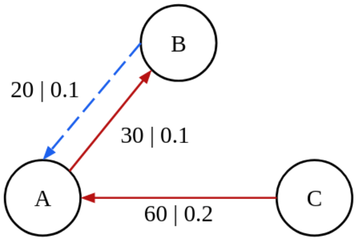


Figure 2.4: An example of node *B* drawing on 20 units of credit at an interest rate of 10% creating a debt from *B* to node *A*.

This simple example demonstrates the FCN's ability to deliver a richer model of financial interactions than the simpler credit network formalism. The inclusion of interest rates and separation of credit and debt edges allow for the modeling of credit cards, loans, and other such financial transactions. Thus, a simple addition to the credit network model greatly expands the modeling capabilities to increasingly complex financial relationships.

2.1.2 Extending FCNs with Financial Node Types

I introduce a new extension to the functionality of FCNs by including ***node types***. A node's type defines its behaviors and edge connections in the network, further expanding modeling capabilities. In this work I will introduce three types of nodes: bank nodes, customer nodes, and fraudster nodes; however the model can support additional node types. The edges in the network will be as described in the FCN model, but by assuming identical interest rates throughout the network interest rates can be omitted to simplify edge specification. A ***credit edge*** is defined as *(A, B, credit, v)* denoting *A* extending *v* units of credit to node *B* and will be depicted in figures as a solid red arrow connecting nodes. An edge *(A, B, debt, v)* denotes a ***debt edge*** with node *A* owing *v* units to node *B*. Debt edges will be depicted in figures as a dashed blue arrow connecting nodes. I refer to the set of all edges within the network as *E*. The inclusion or exclusion of specific node types and directionality of edges allows the model to express varied financial scenarios as I will now describe.

2.1.2.1 Bank Nodes

Bank nodes represent banks and other financial institutions. The credit relationships between banks forms the ***interbank network***, allowing banks to transact with one another on behalf of themselves and their customers, creating debts. In the models presented here, banks are fully connected by credit edges representing a willingness by every bank to transact in some capacity with every other bank in the network. I refer to the set of bank nodes in the network as $B = \{B_1, \ldots, B_b\}$. An example of an interbank network is shown in Figure 2.5. In this example, the amount of credit B_1 extends to B_2 is equal to the amount of credit B_2 extends to B_1 (40 units). This creates two credit edges, $(B_1, B_2, \mathit{credit}, 40)$ and $(B_2, B_1, \mathit{credit}, 40)$, which is condensed to a double sided arrow between the two nodes to simplify the diagram. Similarly, B_1 extends 60 units of credit to B_3 and is extended 60 units of credit in return. Such equal relationships are not necessary, however, as demonstrated by the relationship between B_2 and B_3. Bank node B_2 extends B_3 60 units of credit, while B_3 extends B_2 only 50 units.

Just as in the FCN example, a node in this network can transact with any other trusted node in the network, represented by the existence of a credit edge. Transactions create debt edges between nodes, as described below in Section 2.1.2.3.

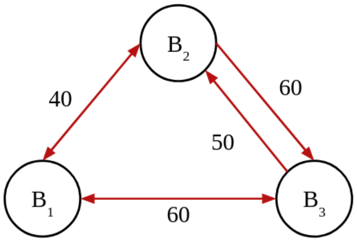


Figure 2.5: An example of an interbank network of three bank nodes fully connected by credit edges. In this example, a double sided arrow describes two bank nodes extending an equal amount of credit to one another.

2.1.2.2 Customer Nodes

Customer nodes represent customers, either individuals or businesses, and allow the model to explicitly distinguish between specific financial retail relationships. I refer to the set of all customer nodes in the network as $C = \{C_1, \ldots, C_n\}$. In financial networks, customers interact with one another through various financial institutions rather than directly, such as through the use of a credit card. Thus, customer nodes are restricted to direct edge connections with bank nodes only, relying on the interbank network to conduct business with other customer nodes. The manner in which these edge connections are defined allow for the expression of different customer and bank relationships.

For example, Figure 2.6 models a customer node C_1 with 100 units deposited in an account at bank B_1. The deposits are modeled as a debt edge from the bank to the customer with the value on the edge symbolizing that C_1 may withdraw some or all those deposits at any time, at which point B_1 is obligated to return them. A customer also has a *willingness to hold additional deposits* in their account, subject to FDIC insurance limits or personal preferences. I model this as credit extended by the customer to the bank creating a credit edge from C_1 to B_1 with a value equal to the amount of additional deposits, in this case 50 units. Together with the current deposit holdings, this forms an upper bound on the total amount of deposits a customer is willing to hold in this account at any given time. In this example, C_1 is willing to hold up to 150 units in its account at any given time.

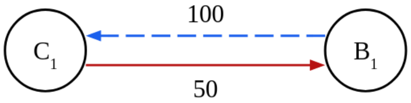


Figure 2.6: An example of a customer node C_1 with 100 units deposited in an account at bank node B_1 and a willingness to hold up to 150 units in the account at any given time.

On the other hand, the relationships between banks and customers could be reversed. Figure 2.7 depicts a debt owed by customer node C_1 to bank node B_1, who has issued credit to C_1. Such a relationship arises when the bank issues credit to the customer, such as through a loan or credit card, which the customer draws on creating a debt. A credit edge represents the future willingness of the bank to extend credit to the customer, while the debt edge represents credit already drawn on by the customer. Using the capacity on the edges, it can be concluded that B_1 originally offered 200 units of credit to C_1 of which C_1 has already drawn 100 units. Thus, B_1 is willing to extend up to 100 units of credit to C_1 in the future.

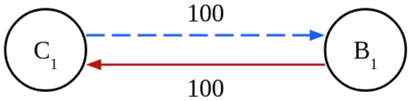


Figure 2.7: A customer node C_1 with a line of credit (e.g., credit card) issued by bank node B_1. The customer has drawn on 100 units previously to make payments and 100 additional units remain on the credit line.

In these examples, only one relationship between a bank and customer node is represented at a time. However, this is not a requirement and multiple relationships between banks and customers may be expressed in the same model.

2.1.2.3 Transactions

Transactions may occur in this extended FCN network in many forms between the various parties. In this section, I will describe two types of transactions captured in this model: bank–to–bank and customer–to–customer. Both transaction types utilize the interbank network.

Banks transact with each other on behalf of themselves or their customers. A bank–to–bank transaction is modeled by the creation of a debt edge between the two transacting nodes and an adjustment to the value on the accompanying credit edge. An example of B_1 transacting 10 units with B_2 is shown in Figure 2.8. A debt edge from B_1 to B_2 is created with value 10 units and the value on the credit edge from B_2 to B_1 has decreased by 10 units. The credit edge from B_1 to B_2 would remain unaffected by this transaction, and has been omitted from the figure for simplicity. Recall that, by definition, all bank nodes are willing to transact with all other bank nodes in the network.

When the transaction takes place between two customers, it is referred to as a ***payment***. The customer making the payment is the ***sender***, while the customer accepting the payment is the ***receiver***. Senders may draw on credit, such as a loan or credit card, or on deposits in a bank account to make a payment. In either case, the process unfolds over a series of defined steps referred to as initialization, clearing, and settlement. First, the sender declares the intended recipient and amount

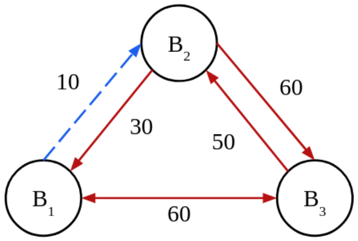


Figure 2.8: A transaction of 10 units from B_1 to B_2 within the interbank network shown in Figure 2.5.

of the payment to its bank. From there, the sender's bank will confirm payment detail validity and exchange messages with the receiver's bank. Finally, funds will be transferred between parties, ending with the final receipt of funds by the receiver. These steps are incorporated into the extended FCN model as a series of updates to the edges connecting the payment sender, sender and receiver banks, and the receiver.

An example of before and after a payment is made in a small network of two customer and two bank nodes is shown in Figure 2.9. Customer node C_1 is as described in Figure 2.6. A second customer node, C_2, has an account at bank node B_2 with 150 units currently deposited and a willingness to hold 100 additional units. Consider the case where C_1 makes a payment of 10 units to C_2 by drawing on its current deposit holdings. C_1 withdraws 10 units from its account to make the payment, decreasing the value on the debt edge from B_1 to C_1 by the value of the payment. With 10 units fewer in its account, C_1's willingness to hold additional deposits increases by the value of the amount withdrawn. This is reflected in an increase in the credit edge from C_1 to B_1 by 10 units. In this case, the payment receiver's account is at a different bank requiring B_1 to transact through the interbank network on behalf of its customer. Note that if the sender and receiver belonged to the same bank, this step would be omitted. As described above, B_1 routes the payment through the interbank network resulting in creation of a debt edge and a change to the credit edge according to the value of the payment. Finally, B_2 deposits the funds into the receiver's account, increasing C_2's deposits by 10 units as captured by the increase on the debt edge from B_2 to C_2. With 10 additional units in its account, C_2's willingness to hold additional deposits must decrease by the same amount reflected with a decrease in the value on the credit edge from C_2 to

B_2 by 10 units. These steps complete a payment made using funds held in a bank account from customer node C_1 to C_2.

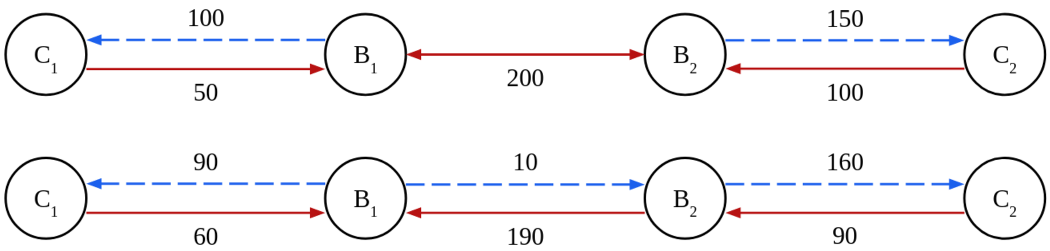


Figure 2.9: An example payment of 10 units between customer node C_1 and customer node C_2 in a model of customer nodes with bank deposits.

If instead the sender were to draw on a line of credit to make a payment, the change would be reflected in a decrease on the credit edge and an increase in the debt edge. The receiver, however, would still receive the payment as an increase in debt from the bank to the receiver and the interbank transaction would be unchanged. A detailed description of payments using credit is saved for Chapter 4. The timing of the edge updates described may also be varied to allow for the distinction between different payment types, as presented in Chapter 5 and Chapter 6.

2.1.2.4 Fraudster Nodes

The model may also incorporate malicious actors committing fraudulent payments within the network. Such actors are represented by the ***fraudster node*** type, while their actions are expressed with the addition of the fraud edge. A ***fraud edge*** $(F_1, C_1, fraud, v)$ is a non-directed edge connecting F_1 and C_1 representing F_1 committing *v* units of fraud using C_1's account. In this case, customer node C_1 is referred to as the ***victim***. A fraudster may draw on either a customer's bank account or credit line, depending on the financial scenario modeled, to commit fraud. Figure 2.10 shows an example of fraudster node F_1 after committing 10 units of fraud using customer node C_1's bank account as reflected by the fraud edge between F_1 and C_1.

When a fraudster makes a fraudulent payment in the network, the processing and outcome remain the same as in the non-fraudulent case. Thus, the only change in the edge updates is the

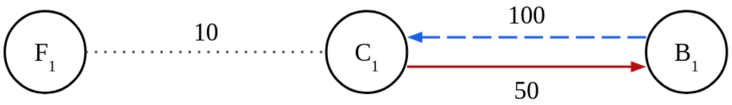


Figure 2.10: An example of the relationship between a malicious actor, the fraudster, and a customer in which the fraudster has committed 10 units of fraud using the customer node's account.

creation of the fraud edge connecting the fraudster and the victim. An example of the full effect of fraudster node F_1 making a payment of 10 units to customer node C_2 using C_1's account is shown in Figure 2.11. Note this payment is similar to the payment shown in Figure 2.9 with the modification of a fraudster, rather than customer initializing the payment. As can be seen in the figure, the edge updates between the victim, banks, and receiver appear the same as the non-fraudulent case.

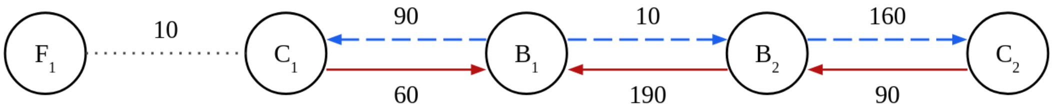


Figure 2.11: The result of fraudster node F_1 making a payment of 10 units using customer node C_1's account. Note with the exception of the fraud edge, the resulting model appears the same as Figure 2.9.

2.1.2.5 Summary

The extended FCN model with node types introduced in this section provides greater flexibility for expressing financial scenarios than previous financial network models. This is achieved through the inclusion or exclusion of specified node types and the defined connections between nodes, allowing for different *versions* of the model. As I will demonstrate in later chapters, these versions along with the definitions of their mechanics and evolution will serve as the environments for studying strategic interactions. By studying these environments with game-theoretic analysis as described below, I seek to understand strategic decision-making within financial networks and the effects of those decisions on network outcomes.

2.2 Empirical Game-Theoretic Analysis

This section discusses the main methods for performing strategic analysis on the extended FCN model. I describe empirical game-theoretic analysis, which achieves game-theoretic reasoning with the use of simulation in agent-based models, and the specific analysis method Subgame Search. I also introduce the novel strategic feature gains assessment which employs EGTA to assess the benefit of strategies in the strategy space for making strategic decisions.

2.2.1 Game Theory Concepts

Many interactions between individuals are governed by underlying strategic behaviors. This is perhaps obvious in games such as chess or poker, but also present in a range of other areas from finance (e.g., auctions, financial markets) to politics (e.g., voting, diplomacy). Economics provides the ***game theory*** framework to analyze such strategic interactions between individuals.

A ***game*** $\Gamma = (P, (S_p), (u_p))$ consists of *P* players indexed by *p*, each having a non-empty strategy set S_p. The strategies available to a player define how it behaves and interacts with other players and the environment in which they exist. Each player has a payoff function, $u_p : \Pi_p S_p \rightarrow \mathbb{R}$, which quantifies the utility player p receives for outcomes resulting from the strategic choices of all players in the game.

If all P players in the game have the same strategy sets and payoff functions, we say they are interchangeable and the game is referred to as a ***symmetric game***. Symmetric games allow for more concise representations. For example, we can simply refer to the strategy set of all players S and the utility function u. In some games, the players can be partitioned into two or more groups with symmetry holding within the groups. These groups are referred to as ***roles*** and the games as ***role symmetric***. For a role symmetric game, we define a set of strategies and a utility function for each role. In this dissertation, I deal with symmetric or role-symmetric games and will introduce the remaining definitions for symmetric games.

We define a ***mixed-strategy profile*** σ as a probability distribution over the strategy set *S* assigning the probability a player employs each strategy. The payoff a player p receives for playing according to σ is $u_p(\sigma)$, while the payoff to some player p for deviating to strategy s while the remaining players play according to σ is $u_p(\sigma_{-p}, s)$. Comparison between strategy profiles can be measured using ***regret***, the maximum gain a player p receives for unilaterally deviating from σ to another strategy s written:

$$\mathsf{REGRET}(\sigma) = \max_{s \in S} u_p(\sigma_{-p}, s) - u_p(\sigma) \tag{2.1}$$

Recall, we are studying symmetric games in which players are interchangeable and thus any player p may be chosen to deviate without loss of generality.

There exist several game-theoretic solution concepts, but for the purposes of this work I focus on Nash equilibrium. A ***Nash equilibrium*** is a strategy profile σ^* from which no single player is better off unilaterally deviating, or $\mathsf{REGRET}(\sigma^*) = 0$. An equilibrium is referred to as a ***pure-strategy Nash equilibrium*** if it assigns a single best strategy and a ***mixed-strategy Nash equilibrium*** if it assigns a probability distribution over the strategy space.

Classical game theory studies an analytical, mathematical model of the game to reason about the decisions of rational agents. However, for complex environments constructing an analytical model may not always be possible. It can be difficult to completely define the outcomes of interactions for a large number of players or for a large strategy space. Furthermore, there may be complexity in the environment such as the effects of uncertain events (e.g., a stock market crash) that may also disproportionately affect players. Fortunately, analysis of such complex interactions may be achieved with a computational method known as ***empirical game-theoretic analysis*** [Wellman et al., 2024]. EGTA combines agent-based modeling, multi-agent simulation, and equilibrium computation to reason about complex strategic decision-making.

2.2.2 EGTA Concepts

EGTA employs agent-based models to simulate the behavior of agents interacting in an environment and uses data gathered from the simulations to conduct a game-theoretic analysis [Wellman, 2016]. In an ***agent-based model***, agents adopt strategies that define their behaviors and interactions within the model. These strategies can range from simple rule-based heuristics to those incorporating more complex implementation such as deep learning. Agents receive a payoff as a result of the performance of their strategies in the environment defined by a payoff function. Thus, it may be understood that an agent-based model possesses the necessary definitions of a game-theoretic *game* $\Gamma(P, S, u)$. Specifically, it defines a set of players (agents), a set of strategies for players, and payoff functions mapping outcomes to valuations for each player. By defining the agent-based model as a game, the model may be used to conduct game-theoretic analysis. Since the game is induced from simulation, the resulting game model is referred to as an ***empirical game model***.

To construct the empirical game model, payoff data is gathered for strategies using simulation in the agent-based model. The agent-based model is considered a black-box oracle $\mathcal{O}$ mapping a strategy profile to payoffs for each agent. Simulating a strategy profile in the agent-based model is referred to as *applying* $\mathcal{O}$ to the strategy profile. To account for uncertainty in the environment, for example variation in agents' private information (e.g., asset holdings), $\mathcal{O}$ is applied to a given profile multiple times. Thus, the payoff for each strategy profile is calculated as the sample average of payoffs observed in each application of $\mathcal{O}$. The general EGTA process is shown in Figure 2.12.

As the number of agents and/or strategies increases, the number of strategy profiles grows exponentially large and only a fraction can be practically evaluated. For example, the game with 10 symmetric agents and 16 strategies described in Chapter 3 has over 3.2 million distinct profiles. Fortunately, EGTA methodology is generally able to identify and confirm approximate equilibria far short of an exhaustive evaluation of the profile space [Fearnley et al., 2015]. This is often achieved by careful implementation of search criteria determining which payoff data to gather. In my experiments, I employ the ***Subgame Search*** method and describe its iterative strategy profile search procedure below.

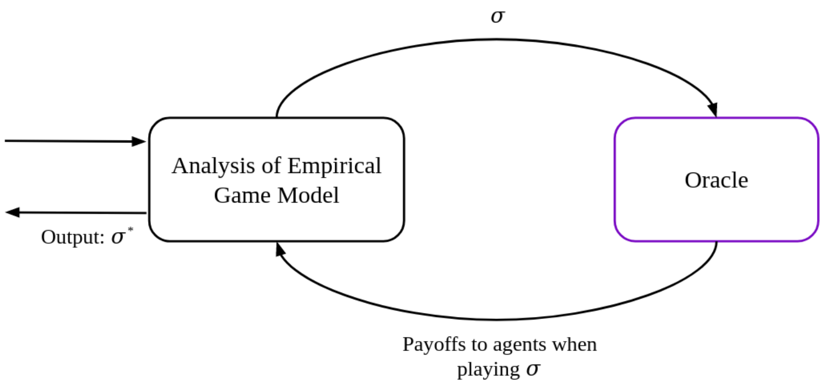

σ
Analysis of Empirical Game Model
Oracle
Output: σ*
Payoffs to agents when playing σ

Figure 2.12: An overview of the iterative EGTA process. The analysis method identifies Nash equilibria and determines additional strategy profiles data to gather. The oracle maps strategy profiles to payoffs using simulation in an agent-based model, which is used to induce the empirical game.

2.2.3 Subgame Search

Subgame Search employs selective simulation to incrementally construct the empirical game model with the goal of identifying Nash equilibria [Cassell and Wellman, 2013]. It does so by continually constructing and searching maximal subgames. A ***subgame*** is defined as a game restricting agents to a subset of the strategies from the full strategy set *S* and for which there exists payoff data for all strategy profiles in the subgame. Conversely, the ***full game*** is the game allowing agents to play any strategy in *S*. Subgame Search begins by analyzing small subgames restricting agents to a few available strategies before potentially moving on to larger subgames. The search is guided by a ***required games queue*** specifying the subgames yet to be evaluated, which is initialized with subgames limited to all singleton strategies $\{s\} \forall s \in S$.

For each initial subgame in the queue, evaluation consists of the following: removal of a subgame from the queue, gathering of payoff data, and analysis of the subgame. Payoff data for all strategy profiles in the subgame is gathered by applying oracle $\mathcal{O}$ to each profile. Since each initial subgame is a singleton set, only one strategy profile exists in each subgame constituting all agents playing only the given strategy. In the analysis phase, a Nash equilibrium of the subgame, σ^c, is identified and considered a ***candidate equilibrium*** for the full game. To confirm or refute this candidate, $\mathcal{O}$ is applied to all one-player deviation profiles from the candidate to strategies outside of the subgame. In a ***one-player deviation*** profile from σ, a single ***deviating agent*** deviates to a different strategy while the remaining agents play according to σ.

These deviating profiles are evaluated to determine if any beneficial deviations from the candidate exist. A ***beneficial deviation***, b, is a strategy providing a higher payoff to the deviating agent than playing according to the candidate equilibrium. In other words, the candidate equilibrium has non-zero regret, as defined in Equation 2.1. If a candidate has no beneficial deviations, the regret is 0, and by definition it can be confirmed as an equilibrium of the full game.

Simulation-based games often gather noisy payoff data and to account for this, Subgame Search considers ϵ-Nash equilibrium. An ***ϵ-Nash equilibrium*** is a strategy profile with regret less than ϵ, considered an approximate Nash equilibrium of the game. Thus, as long as the regret is less than ϵ, Subgame Search will consider the candidate a confirmed equilibrium of the full game. Otherwise, the candidate equilibrium and the beneficial deviations are used to create new subgames to be added to the required games queue. The ***support*** of a strategy profile, $supp(\sigma)$, is the set of strategies assigned a non-zero probability of being played. The strategy offering the largest payoff gain, the most beneficial deviation, is defined as b^* and the set of all additional beneficial deviations is B. Then new subgames are created and added to the queues as follows:

- Add subgame formed by $supp(\sigma^c) \cup b^*$ to the required games queue.
- Add subgames formed by $supp(\sigma^c) \cup b \; \forall b \in B$ to the ***back-up queue***.

The processing of subgames added to the required games queue follows the same steps outlined for the initial subgames. Payoff data collected while processing a given subgame is stored and accessible throughout the search process, making later processing more efficient. When gathering the payoff data for new subgames, $\mathcal{O}$ only needs to be applied to profiles not previously encountered in the search process. The remaining profiles' data can be accessed in the stored data base for the analysis step. Similarly, $\mathcal{O}$ is only applied to new one player deviation profiles. Thus, as exploration continues, the number of profiles required to simulate for evaluation of subgames decreases.

Subgame Search terminates when at least one Nash equilibrium of the full game has been identified *and* the required games queue is empty. In the event an equilibrium has not yet been found and the queue is empty, the search process proceeds by pulling subgames from the *back-up games queue*. The search process is designed to minimize the number of strategy profiles required to search and thus is not guaranteed to return multiple Nash equilibrium if more than one exists. The complete steps of the Subgame Search process are outlined in Figure 2.13. These steps can be contextualized within the general EGTA framework according to the color of their outline. Steps outlined in purple perform calls to the oracle, while those outlined in black perform analysis on the constructed game model.

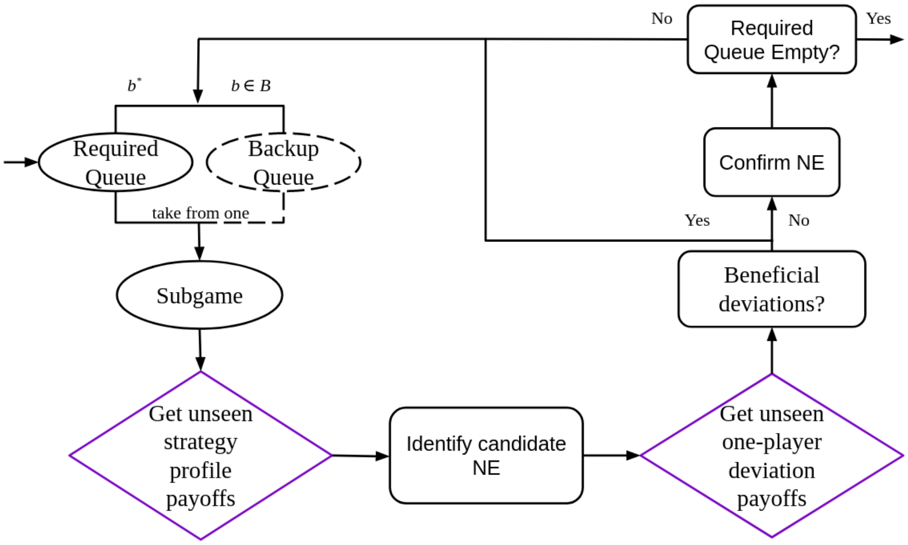


Figure 2.13: The Subgame Search process iteratively evaluates subgames to identify candidate Nash equilibria and confirms or refutes them as Nash equilibria of the full game. Steps are outlined in color corresponding to their context within the general EGTA framework presented in Figure 2.12.

2.3 Strategic Feature Gains Assessment

EGTA also serves as the basis for a novel method of assessing the strategy space I introduce as the strategic feature gains assessment. Often, we wish to analyze a strategic context beyond simply identifying strategic choices of rational agents in equilibrium. In particular, we may wish to quantitatively assess and compare the suitability of strategies for making the strategic decision. Such an analysis may provide additional context to the strategic selection, a ranking of the strategies, or help constructing new strategies.

A Nash equilibrium may provide some measure of answering such questions, though it is rather limited providing at best a binary partition of the strategy space. In particular, strategies in the support may be considered beneficial for making the strategic decision, while the remaining strategies are deemed not beneficial. While the probability assigned to strategies in equilibrium may be seen as a natural ranking, this may prove a problematic comparison. In particular, under such an interpretation all strategies outside the support are considered equally ranked. To illustrate the problem, consider such a strategy S_1 that always performs poorly against any other strategy in the set. A so called ***dominated*** strategy would never logically be selected and its poor performance should result in a very low ranking. However, by using probabilities assigned in equilibrium, such a strategy would be ranked similarly to a non-dominated strategy outside the support. Therefore, Nash equilibrium alone does not necessarily provide sufficient meaningful comparison of individual strategies.

Prior works address analysis of the strategy space with several approaches. Shearer et al. [2023] compares an outcome of interest, specifically total and market profits, under different strategies for manipulating financial benchmarks. In other cases, the focus is not on individual strategies but rather on *groups* or *types* of strategies categorized by similarities such as the basis for strategic decision-making. Such analyses compare the usefulness of different categories in equilibrium [Mayo and Wellman, 2021, Wang et al., 2021], as well as comparing outcomes based on the inclusion or exclusion of particular strategy groups [Brinkman and Wellman, 2017, Wang et al., 2021]. Finally, Jordan [2010] introduces the concept of ***NE-Regret*** to measure the benefit of a strategy calculated by the regret to a player for deviating from the equilibrium to a given strategy. This leads to the ***NE-response ranking*** which can be used to rank strategies according to the benefit they provide for strategic decision-making. NE-Regret has been applied to studying poker bots [Jordan, 2010], auction bidding strategies [Wellman et al., 2012], and graph security games [Nguyen et al., 2017]. Balduzzi et al. [2018] introduces a similar method for ranking agents and single-agent tasks called Nash averaging.

I present the ***strategic feature gains assessment*** to understand the benefit to a player in a game provided by the inclusion of a specified set of strategies, deemed the deviation set. The deviation set is defined such that the encompassing strategies are united by commonalities such as the basis for decision making or their effect on agent behavior. The assessment quantifies benefit to players as the maximum payoff gain achieved by deviating to strategies in the deviation set from a Nash equilibrium of a base set identified using the Subgame Search procedure described above.

The assessment is defined for a symmetric game $\Gamma = (P, S, u)$ with *P* players indexed by *p*, a strategy set *S*, and a utility function *u*. The ***deviation set*** is defined as a set of strategies, $\Delta \subset S$, to be assessed. The set of initial strategies players may access is the ***base set*** $\Omega \subset S$, which is disjoint from Δ.

The assessment steps are as follows:

1. Define the sets Δ and Ω to be used for the assessment

2. Obtain $\sigma^*(\Omega)$ using Subgame Search

3. Calculate the gain of Δ as $\max_{s \in \Delta \cup \Omega} u_i(\sigma^*_{-i}, s_i) - u_i(\sigma^*)$

If a game is role symmetric with the assessment performed on only one role, players of the non-assessed role retain use of their full strategy sets throughout the assessment. Performing the assessment with different base and deviation sets provides a point of comparison between different scenarios or information sets. This allows for a broader understanding of the important factors in player decision making.

2.4 Summary

In this chapter, I describe the methodological components employed in this dissertation. The extended financial credit network model with node types introduces a flexible environment for expressing a variety of financial situations. With the addition of rules governing node interaction, strategy sets, and payoff functions, the model fits the definition of an agent-based model and allows for the application of empirical game-theoretic analysis with nodes as *players* in a game. Subsequent chapters will specify these additions for analysis of specific financial scenarios. Empirical game-theoretic analysis is performed using Subgame Search to identify Nash equilibria in these financial network games.

CHAPTER 3

Portfolio Compression

3.1 Introduction

The idea of ***portfolio compression*** is to simplify a web of financial obligations by canceling a cycle of debt, leaving each party's net position the same. Consider the situation in Figure 3.1 where bank A owes money to bank B, B owes a like amount to bank C, and C owes the same amount to A. Furthermore, suppose the debt contracts comprising the cycle over these institutions have the same interest rates and maturity dates. Intuitively, canceling the three debts leaves all parties in the same net position, while simplifying their balance sheets. If all banks on a cycle are solvent and able to pay their debts when due, the operation of compressing portfolios by canceling the debts has no effect and the compression decision is of little interest. With risk of default, however, portfolio compression is not generally neutral and may affect stability in the network.

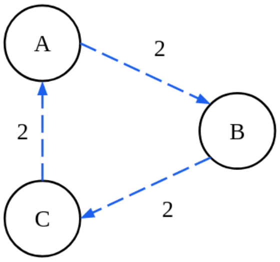


Figure 3.1: A simple example of a debt cycle.

For instance, if bank A in Figure 3.1 turns out to be insolvent, then B would be better off had the compression occurred than had it not. C might similarly be better off if B's ability to pay its debts depends on receiving payment from A. Thus, compression may serve to limit contagion in financial networks by removing paths of default propagation. This potential to limit financial contagion while simplifying balance sheets is the reason compression has gained prominence since the 2008 financial crisis.

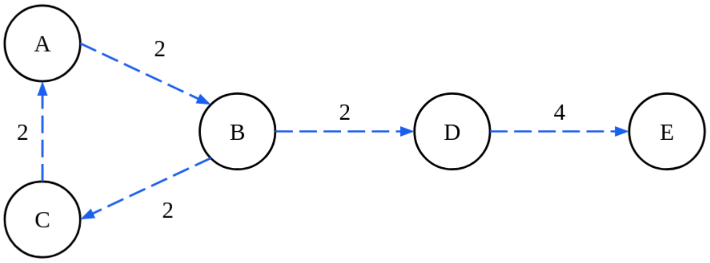


Figure 3.2: An example where insolvency in the network leads to harmful compression effects in the network [Schuldenzucker and Seuken, 2021].

On the other hand, portfolio compression can also impede the ability for parts of the network to absorb losses from an insolvent bank, thus enabling spread that may have been avoided. Schuldenzucker and Seuken [2021] present an example, shown in Figure 3.2, where eliminating a cycle causes previously solvent institutions to become insolvent. The network starts with a cycle like the one described above with the addition of two edges: B owes money to D and D owes money to E. Each bank holds some quantity of external assets outside of the network. Suppose the cycle were compressed and then B experiences a shock that reduces its external assets to zero. In this scenario, B would have no assets to pay its remaining creditor, D. To absorb the loss from B's insolvency, D must hold enough external assets to pay E, otherwise D would also default. In contrast, suppose this cycle were not compressed prior to the shock reducing B's external assets to zero. If C's external asset holdings are sufficiently large, C could absorb the loss from B's insolvency and prevent its spread to A. In turn, A would pay B the full debt amount. B would then use its internal assets to make a partial payment to its creditors, which could prevent D from being insolvent. Note that the realization of this outcome depends on the value of the partial payment made by B. In this case, the cycle cushions the shock and so compression has the harmful effect of allowing default to spread elsewhere.

Although compression in this case is harmful to the network, the optimal outcome of rejecting the compression may not be realized. In this example, bank *C* would prefer compression, while *A* and *B*'s indifference towards compression may result in their acceptance of the proposal. However, imagine a slight modification to the network so there is a liability from *D* to *A*. If *D*'s ability to pay its liabilities depends on receiving a payment from *B*, compression would allow insolvency to spread to *A*. In this case, *A* would no longer be indifferent about the compression. Thus, we can see that the qualitative effects of compression are dependent on characteristics of the network.

Some of the prior work on compression has focused on analyzing which characteristics of a network determine whether a compression will be harmful or not. One important factor is the ***recovery rate***: the fraction of assets an insolvent node is able to recover and use to pay back creditors, with the remaining assets being lost to default. Veraart [2019] finds that when recovery rates are nonzero, portfolio compression may be harmful to the network. Schuldenzucker and Seuken [2021] build on these results by studying a broader range of network aspects to determine when financial institutions are incentivized to perform compression and when compression is a Pareto improvement.

These analyses illuminate the interplay of network structure and effects of compression from an *ex-post* perspective where details of the external shock are known. In practice, decisions about whether to compress debts are made by institutions with a limited view of the network and prior to resolution of uncertain financial events. Evaluation of compression decisions, therefore, necessitates a model accounting for uncertainty and incomplete information on the state of the network. Further, prediction of what these firms would decide should also consider strategic effects, recognizing that optimal policies generally depend on how the other firms make their decisions.

To analyze the broader context of compression decisions, I frame portfolio compression as a strategic decision made by individual agents with incomplete information in an extended FCN model. I define a one-shot game played by bank nodes strategically deciding to accept or decline a proposed compression. I consider a variety of heuristic strategies based on the information available to nodes about the network, which may provide useful indications of potential negative effects of compression to the voting node. These strategies are evaluated under different insolvent node recovery rates using EGTA to identify Nash equilibria. Additionally, I apply the strategic feature gains assessment to identify the value of local information for making the compression decision. Finally, I conduct an analysis of the effects of compression as a strategic decision on the network, including quantifying systemic risk. Bank nodes in the network make the compression decision according to the Nash equilibria identified through EGTA and various outcomes in the network are measured. Systemic risk is calculated three ways using measures of total equity and number of insolvent nodes.

From these experiments I find evidence that for an individual node, the optimal policy is generally conditional on features of the local network and the individual's balance sheet. Through the strategic feature gains assessment, I find that at higher recovery rates, nodes perceive a greater potential for cycles to cushion shocks in the network. When the compression decision is made strategically, the likelihood of a given cycle being compressed is lower than in the social optimum. Compression decisions generated by strategic consensus may be better or worse than no compression. However, the price of anarchy of a strategic compression is high for all recovery rates.

3.2 Related Work

Eisenberg and Noe [2001] pioneered the modeling of financial debt networks as a set of nodes representing banks, with directed edges forming interbank liabilities, for the purpose of studying systemic risk. Rogers and Veraart [2013] expand the representation by adding recovery rates to the network, allowing insolvent banks to recover part of their assets to pass onto creditors. Resolution of debts in such a network is defined by a ***clearing vector*** specifying payments required by each node. The authors present an iterative algorithm for deriving clearing vectors consistent with the debt obligations and recovery rates applied to insolvent nodes. They use their expanded model of the financial debt network and clearing algorithm to study the rescue of failing banks in the network through mergers with solvent banks. The authors find that solvent banks do not always have both the means and incentives to rescue insolvent banks in the network.

Studies of contagion and systemic risk in financial networks have emphasized the impact of network structure [Acemoglu et al., 2015, Elliott et al., 2014, Glasserman and Young, 2015]. These works do not specifically address the issue of debt compression. Kanellopoulos et al. [2023] study debt transfers, which also operate on the premise of debt cancellation, however the cancellation is not necessary cyclic in nature. Zhou et al. [2024] analyze a strategic prepayments game, in which insolvency and cycles in the network may incentivize a prepayment. A prepayment is made when a bank pays a liability in full before its maturity date, clearing the debt from the network. While multiple prepayments may occur, they will not necessarily eliminate a cycle of debt in the network.

Prior studies of compression tend to focus on algorithms for identifying candidate compressions, rather than the decision of whether to compress particular candidates [D'Errico and Roukny, 2019, O'Kane, 2017]. For example, D'Errico and Roukny [2019] use transaction data on over-the-counter derivatives called credit default swaps (CDSs) to study efficiency of different types of portfolio compression. A CDS is a contract that enables a creditor to hedge the risk of a debt obligation with another member of the network. Efficiency of a compression method is determined by the amount of the excess–the total amount eligible for compression–the resulting compression is able to eliminate. Of the types of compression analyzed, conservative and hybrid compression are the most efficient at removing excess from the network.

Veraart [2019] analyzes the effect of portfolio compression on systemic risk, specifically focusing on the relationship between systemic risk and recovery rate. Systemic risk is defined by the set of banks that become insolvent with compression compared to the set in the same network without any compression. Compression is defined as ***beneficial*** when the set of banks defaulting with compression is a strict subset of the banks defaulting without compression, ***weakly beneficial*** when beneficial or the two sets are the same, and ***harmful*** if some node defaults under compression which would not have defaulted without compression. They prove that when the recovery rate is nonzero, compression can be harmful to the network. Conversely, compression with a zero recovery rate is always weakly beneficial. Furthermore, when no nodes on the cycles become insolvent, the network is indifferent to compression.

Schuldenzucker and Seuken [2021] study compression from the ex-post perspective, analyzing the relationship between a variety of network characteristics and the effects of compression on systemic risk. The analysis focuses on whether a compression is beneficial, either in a Pareto or social welfare sense. For example, they establish a connection between homogeneity of the network and compression being a Pareto improvement.

These analyses of portfolio compression have demonstrated the potential benefits of compression both quantitatively and qualitatively, as well as potential harm to systemic risk. Given that compression has the potential to have a positive or negative effect on the financial network, it seems worthwhile to consider how financial institutions faced with a potential compression should make their decision.

3.3 Compression Model and Definitions

I define the compression model as an instance of the interbank network described in Section 2.1.2.1. The model consists of a set of bank nodes $B = \{B_1, \ldots, B_b\}$ and a set of edges *E* between them. In this section I will introduce additional terminology for specifically discussing interbank debt and compression.

I refer to an edge $(B_i, B_j, debt, v)$ as a ***liability*** of *v* units B_i owes to B_j. This same edge represents an asset for B_j and since it is within the network it is referred to as an ***internal asset***. A bank node in this model may owe multiple liabilities to another node, arising from multiple contracts, and thus I denote the total liability owed by node B_i to B_j as:

$$l_{B_iB_j} = \sum_{(B_i, B_j, debt, v) \in E} v.$$

Then I can define a given node, B_i's ***total liability*** in the network by:

$$L_{B_i} = \sum_{j=1}^{b} l_{B_i B_j}$$

Each node is also endowed with some ***external assets*** e and I define a node B_i's ***total assets*** A_{B_i} as the sum of external and internal assets:

$$A_{B_i} = e_{B_i} + \sum_{j=1}^{b} l_{B_j B_i}.$$

A bank node is considered ***insolvent*** when $A_{B_i} - L_{B_i} < 0$.

Using these definitions, I can formally define a portfolio compression. Given a cycle $c = (B^c, E^c)$ with $B^c \subseteq B$ and $E^c \subseteq E$, let $\mu^c = \min_{(B_i, B_j, debt, v) \in E^c} v$ denote the smallest liability on the cycle. Then a ***compression*** of a cycle c is defined by the revised set of liabilities for edges on the cycle:

$$\forall (B_i, B_j, debt, v) \in E^c. \quad v^c = v - \mu^c.$$

It can be seen that compression reduces the assets and liabilities of the nodes on the cycle while maintaining their net positions. An example of a potential cycle and its subsequent compression can be seen in Figure 3.3. Note credit edges between the bank nodes are omitted for clarity, though would exist between nodes to support the creation of debt. The cycle *c* consists of nodes B_1, B_2, and B_3 and the set of edges: [$(B_1, B_2, debt, 3)$, $(B_2, B_3, debt, 2)$, $(B_3, B_1, debt, 5)$]. The μ^c value is 2. In this example, compressing the cycle allows node B_3 to eliminate its exposure to node B_2 and decreases the liabilities of B_1 and B_3.

An interbank network as the one described is also associated with recovery rate parameters $\alpha, \beta \in [0, 1]$, which govern payments of an insolvent node when its debts reach maturity. The α parameter denotes the fraction of an insolvent node's external assets that are recovered. β controls the fraction of an insolvent node's internal assets that are recovered. The assets not recovered are lost to default costs. The total assets recovered by an insolvent node B_i are:

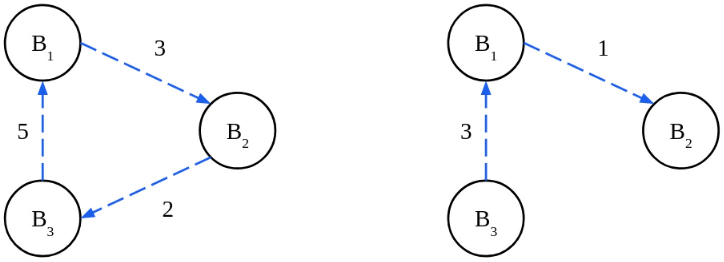


Figure 3.3: A simple debt cycle before and after compression by two units.

$$\alpha e_{B_i} + \beta \sum_{j=1}^{b} l_{B_j B_i}.$$

I make several simplifying assumptions for this current analysis. As presented in Section 2.1.2.1, nodes in the interbank network are fully connected representing established trust between all parties for transacting. The first simplification is all credit edges have infinite value, allowing for credit edges to be omitted from the discussion. Additionally, all debts in the network have the same maturity date and interest rate. [1] We also assume a financial institution never has a liability to itself, $l_{B_i B_i} = 0$. Finally, we assume that $\alpha = \beta$, and refer to α as the parameter controlling the fraction of an insolvent bank node's total assets that are recovered in the event of a default.

3.4 Compression Game

Adopting the *ex-ante* perspective, we introduce a compression game where b bank nodes in an interbank network decide whether to agree to a given compression proposal using heuristic strategies based on local information. If all accept, the cycle is compressed. After a negative external asset shock, nodes receive a payoff based on the resolution of liabilities in the network.

[1] Under this assumption we can also limit each ordered pair of nodes to at most one debt edge.

3.4.1 Initialization

The network is initialized with b bank nodes, each with a random endowment of external assets $e_{B_i} \sim U\{10, \ldots, 20\}$. Liabilities between bank nodes are randomly generated according to the following procedure:

- Randomly select two nodes B_i and B_j.
- If $l_{B_iB_j} = 0$ and $e_{B_j} > 0$, create edge $(B_i, B_j, debt, v)$ in E with $v \sim U\{1, \ldots, e_{B_j}\}$.
- Update the external assets of both nodes: $e_{B_i} \leftarrow e_{B_i} + v$; $e_{B_j} \leftarrow e_{B_j} - v$.

This procedure repeats until the network contains at least one cycle of three or more edges. Note that throughout, all nodes in the network remain solvent by construction. During the initialization process, nodes also select their chosen strategy from those described below.

3.4.2 Heuristic Strategies

The decision of interest for my analysis is a node's choice to accept or reject a proposal to compress a given debt cycle. In principle, nodes may take into account anything they know about the network. For this study, I define a set of heuristic strategies that consider different forms of information about a node's situation and the network, including various measures of assets and liabilities both total and associated with particular nodes. Each strategy maps this information to general agreement or disagreement with a proposed compression. I then analyze the strategic interaction of nodes in the network by their selection from among the available heuristics. Table 3.1 lists all strategies, defined as the condition under which node B_i chooses to accept the proposed compression. Several of the strategies reference a single node B_j in the decision context. This should be understood to be the bank node directly preceding B_i on the cycle $(B_j, B_i, debt, v) \in E^c$) in the case of the Internal Ratio strategy and its negation. For the Liability Ratio and Relative Liability strategies and their negations, B_j is the node directly following B_i on the cycle ($(B_i, B_j, debt, v) \in E^c$).

Strategy Name	Vote ”yes” iff
YES	Always
NO	Never
Asset Ratio	$\frac{\sum_{j=1}^{b} l_{B_j B_i}}{A_{B_i}} > 0.5$
Neg Asset Ratio	$\frac{\sum_{j=1}^{b} l_{B_j B_i}}{A_{B_i}} \leq 0.5$
Expected Internal	$0.5 \times (\sum_{j=1}^{b} l_{B_j B_i}) - L_{B_i} < 0$
Neg Expected Internal	$0.5 \times (\sum_{j=1}^{b} l_{B_j B_i}) - L_{B_i} \geq 0$
External	$e_{B_i} - L_{B_i} < 0$
Neg External	$e_{B_i} - L_{B_i} \geq 0$
Internal Ratio	$\frac{l_{B_j B_i} - \mu^c}{\sum_{j=1}^{b} l_{B_j B_i}} > 0.5$
Neg Internal Ratio	$\frac{l_{B_j B_i} - \mu^c}{\sum_{j=1}^{b} l_{B_j B_i}} \leq 0.5$
Internal	$(\sum_{j=1}^{b} l_{B_j B_i}) - L_{B_i} < 0$
Neg Internal	$(\sum_{j=1}^{b} l_{B_j B_i}) - L_{B_i} \geq 0$
Liability Ratio	$\frac{l_{B_i B_j} - \mu^c}{L_{B_i}} > 0.5$
Neg Liability Ratio	$\frac{l_{B_i B_j} - \mu^c}{L_{B_i}} \leq 0.5$
Relative Liability	$\frac{l_{B_i B_j} - \mu^c}{l_{B_i B_j}} \geq \frac{L_{B_i} - \mu^c}{L_{B_i}}$
Neg Relative Liability	$\frac{l_{B_i B_j} - \mu^c}{l_{B_i B_j}} < \frac{L_{B_i} - \mu^c}{L_{B_i}}$

Table 3.1: A complete list of heuristic strategies denoting the conditions under which node B_i accepts compression proposals.

The first strategy listed in Table 3.1 is the unconditional acceptance, a strategy I label YES. This is meant to cover as a baseline the common practice of simply accepting any compression proposed. The tendency to agree may be due to a number of factors, including a desire to simplify balance sheets or to comply with regulation [Schuldenzucker and Seuken, 2021]. For symmetry I also include its complement, the unconditional reject strategy NO. The remaining heuristic strategies are defined by inequalities over assets and liabilities in various ways. The complements are defined by simple negation, or flipping the inequality. For example, the Asset Ratio strategy, $\frac{\sum_{j=1}^{b} l_{B_j B_i}}{A_{B_i}} > 0.5$ has a negation Neg Asset Ratio, $\frac{\sum_{j=1}^{b} l_{B_j B_i}}{A_{B_i}} \leq 0.5$. This collection of strategies is far from exhaustive, but is designed to cover a variety of strategically relevant features.

Understanding the relative usefulness of different network features is the main object of applying game-theoretic analysis.

3.4.3 Game Steps and Payoffs

The game then proceeds with a compression vote. For cycle c, the B^c nodes independently cast their vote for the proposed compression cycle using their selected strategy. A compression is performed if and only if all B^c nodes vote in favor of the compression. If a network contains more than one cycle, they are addressed in sequence. I consider compression on cycles with $|B^c| \geq 3$ before cycles with $|B^c| = 2$. Cycles within these size classes are ordered randomly. The order is relevant, as decisions for a given cycle may be affected by the outcome of compressing cycles considered earlier in the sequence.

As the goal is to understand the effect of strategic compression on systemic risk, I follow the compression decision with a financial stress event. Absent an insolvent node on the cycle, compression has no effect on systemic risk [Veraart, 2019, Theorem 3.10]. I therefore subject the network to a shock on external assets, designed to throw some nodes into insolvency. This asset shock is modeled as the failed realization of outside investments. Nodes invest their external assets that remain after liability creation in one of two ways based on whether they are part of a cycle or not. Nodes not on a cycle are randomly assigned to either one of two investments, or to not make an investment, each with probability 1/3. Conversely, nodes on a cycle are randomly assigned to one of the two investments, each with probability 1/2. This process is designed to increase the likelihood that a non-empty subset of nodes on a cycle become insolvent. In the final step, the external shock is triggered by uniformly randomly choosing an investment option to fail. When an investment fails, nodes assigned to that investment do not recoup any of their investment.

The network is then *resolved* by forcing all nodes to pay off their debts. Solvent nodes pay in full and insolvent nodes pay creditors from their recovered assets proportional to their respective liabilities. Following the algorithm by Rogers and Veraart [2013], I define a proportional liability matrix Π with entries $\Pi_{B_i B_j} = \frac{l_{B_i B_j}}{L_{B_i}}$. A payment *p* made by an insolvent bank node B_i to one of its creditors B_j is then given by:

$$p_{B_iB_j} = \alpha \left(e_{B_i} + \sum_{k=1}^{b} p_{B_kB_i} \right) \Pi_{B_iB_j}.$$

At the end of the game, the payoff to node B_i for playing the selected strategy is its remaining asset holdings after resolving all liabilities. Nodes that become insolvent receive a payoff of zero.

3.5 Empirical Game-Theoretic Analysis

3.5.1 Identifying Nash Equilibria

I analyze the compression decision using EGTA to identify Nash equilibria in games with $b = 10$ bank nodes and 16 heuristic strategies. Payoff data for strategy profiles is gathered by repeated simulation of profiles in the compression game described above. I test multiple configurations of the game defined by the recovery rate parameter $\alpha \in \{0, 0.1, 0.3, 0.5, 0.7, 1.0\}$. Each of these games has over 3.2 million distinct profiles, but employing the Subgame Search methodology necessitates only evaluating a fraction of these. Table 3.2 presents the extent of simulation required for the Nash equilibrium computation across the game configurations.

α	Min # runs/profile	# profiles
0	1.5M	12,109
0.1	1.0M	78,247
0.3	1.5M	25,665
0.5	2.5M	62,517
0.7	1.5M	51,746
1.0	2.0M	63,538

Table 3.2: Simulation effort for EGTA applied to six games. For each game, I tabulate the minimum number of runs per profile (M=million) and the number of profiles evaluated.

This analysis attempts not to emphasize the use of specific strategies, but to identify local information most useful for making the compression decision. To this end, I group strategies sharing similar characteristics into four sets referred to as ***features***. The assets feature includes strategies that base the compression decision on a node's asset positions, while the liabilities feature includes strategies focusing on a node's liabilities. The dynamic feature strategies consider possible changes that may occur in the network. Finally, the unconditional set includes only the YES and NO strategies that do not use any network information to inform decision making. A list of the features, their descriptions, and encompassing strategies can be found in Table 3.3.

I report the results of the Nash equilibrium search in the context of the feature sets. The total ***weight*** assigned to a feature in equilibrium is the sum of the probabilities assigned in equilibrium to strategies belonging to the feature. Feature weights are normalized such that if a game configuration has multiple equilibria, the average feature weights across equilibrium profiles is reported.

The weight of all features in equilibrium is reported in Figure 3.4. Note that features may comprise overlapping sets of strategies allowing total weights across features to exceed one for some games. Generally speaking, I find that strategies employing simple, local network information are preferred to the unconditional strategies by nodes making the compression decision. The clear exception occurs when $\alpha = 0$, for which YES is a pure strategy Nash equilibrium, aligning with the theorem stating compression cannot be harmful with a zero recovery rate [Veraart, 2019]. Strategies belonging to the assets or dynamic features are assigned the majority of the weight in equilibrium for games with $\alpha > 0$. This indicates anticipating network changes and assessing asset holdings may be most informative when faced with a compression decision.

3.5.2 Strategic Feature Gains Assessment

Aside from understanding the strategic choices in equilibrium, I also wish to evaluate the heuristic strategy space to further understand the benefit of different features for making the compression decision. To conduct this analysis, I employ the strategic feature gains assessment. In particular I perform four assessments on each game configuration to analyze the effects of including or excluding features. Note that the $\alpha = 0$ case is omitted as the Nash equilibrium in that case is to not use local information, rending an exploration of strategies using such information trivial. The first three assessments (A1, A2, A3) measure the benefit of access to a given feature when previously

Feature name	Description	Strategies
Assets	makes decision based on asset position	asset ratio, expected internal, external, internal ratio, internal
Liabilities	makes decision based on liability position	liability ratio, relative liability
Dynamic	considers changes that may occur in the network	internal ratio, liability ratio, relative liability, expected internal
Unconditional	does not use any information to make the decision	Yes, No

Table 3.3: The feature groups created for analysis including their description and a list of the encompassing strategies. The negations of each listed strategy are similarly included in each feature group.

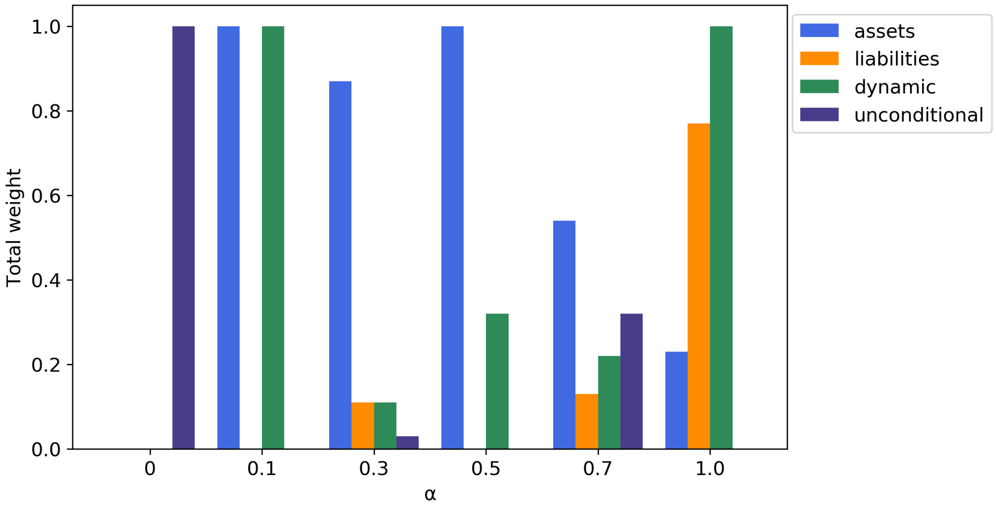


Figure 3.4: When $\alpha \neq 0$, voting based on asset holdings or how a negative shock or compression might affect the network is often most helpful.

nodes only had access to all other strategies. The last assessment, A4, uses only the unconditional strategies as the base set and measures deviation to all features. This assessment is designed to uncover the strategy offering the most gains to a node previously unable to use local information to make the compression decision. The base and deviation sets for each assessment are outlined in Table 3.4. Note that when performing the assessments, the membership for strategies belonging to multiple features is overridden in favor of the base set. For example, a strategy S_1 belonging to a feature in the base set and deviation set will only be used in the base set.

Assessment name	Base set	Deviation set
A1	Liabilities, Dynamic, Unconditional	Assets
A2	Assets, Dynamic, Unconditional	Liabilities
A3	Assets, Liabilities, Unconditional	Dynamic
A4	Unconditional	Assets, Liabilities, Dynamic

Table 3.4: The four strategic feature gains assessments reported as the features belonging to each assessment set.

Figure 3.5 displays the results from assessments A1, A2, and A3, reporting the respective deviation sets in parentheses. From these results, it can be seen that allowing strategies that use asset holdings (assets feature) or strategies that consider possible changes to the network (dynamic feature), offer the largest gains. This aligns with the equilibrium weight findings, further demonstrating the importance of such local information in the compression decision.

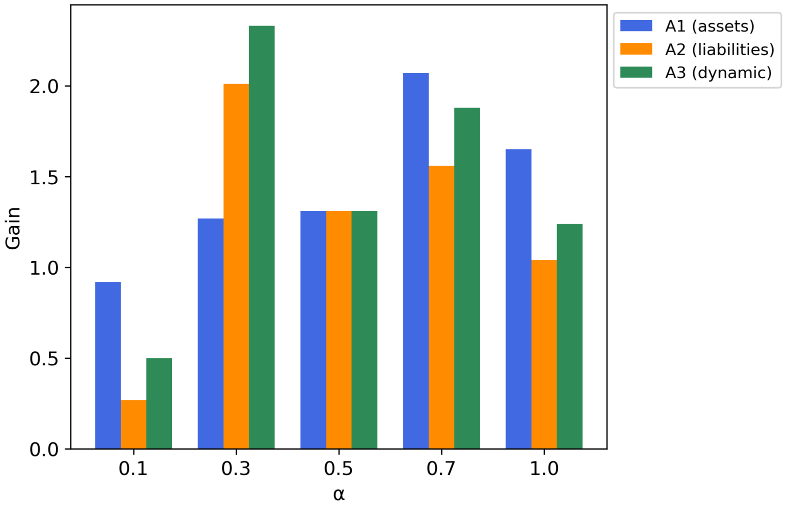


Figure 3.5: An agent experiences the most benefit when allowed to use asset information or information about how a negative shock or compression might affect the network in the compression decision.

For assessment A4 I report the strategy offering the greatest gains and its feature categorization for each game configuration. The results in Table 3.5 show that generally the feature assets provides the most gains. I also find that as α increases, the strategies with the maximum gain in payoff switch from preferring compression if asset holdings are strictly greater than liabilities to preferring compression if asset holdings are smaller than liabilities. This may be because a larger α value allows insolvent nodes to recover a larger portion of their assets, which they can pass on to their creditors. Therefore, when α is smaller, nodes place a higher premium on compression's benefit of limiting contagion. But when α is larger, nodes may tend to view the cycle as a means of cushioning shocks. At $\alpha = 0.5$, a strategy that belongs to both the liabilities and dynamic feature is the most useful.

In comparing the results of the feature assessments, I find that in the cases of $\alpha \in \{0.1, 0.7, 1\}$, using the asset feature provided the greatest gains in all analyses. When $\alpha = 0.3$ the most beneficial feature depends on what other information nodes previously had available to them. Nodes with prior access to other local information gain the most from using the dynamic feature. Start-

α	Feature with max gain	Strategy with max gain
0.1	Assets	Internal
0.3	Assets	External
0.5	Liabilities/Dynamic	Neg Relative Liability
0.7	Assets	Neg External
1.0	Assets	Neg Internal

Table 3.5: The most useful feature to an agent making the compression decision is Assets.

ing from no local information (A4), nodes gain the most by using the assets feature. A similar situation holds for $\alpha = 0.5$.

3.6 Effects of Strategic Compression

3.6.1 Compression Occurrence

I introduce an experiment designed to understand how compression as a strategic decision affects how often compression is performed. For each game configuration, nodes play according to the identified Nash equilibrium in 10,000 random instances of the game described in Section 3.4. To measure the impact of a strategic decision on compression, I track several metrics over the many random instances: number of cycles, number of cycles compressed, length of cycles, number of insolvent nodes, total equity, and number of nodes on a cycle voting in favor of compression. If the equilibrium for a given configuration is mixed, the heuristic strategy is randomly selected from a weighted sample of the strategies in the support. When a game configuration has multiple equilibria, each equilibrium is tested separately and results are reported as an average over the equilibria.

Table 3.6 shows the impact of allowing a strategic decision on the number of compressions performed. Schuldenzucker and Seuken [2021] note that in practice, banks presented with a compression proposal generally agree to perform compression. The reason for this may be that banks are following the YES policy due to regulation or a desire to simplify balance sheets. However,

α	% cycles compressed	% "YES" when no compression
0	100	–
0.1	32	53
0.3	56	62
0.5	12	41
0.7	30	39
1.0	36	53

Table 3.6: When $\alpha \neq 0$ and the compression decision is made strategically, the percentage of cycles compressed remains low.

these findings show that when the nodes in this network use local information to make the compression decision, the majority of possible compressions are not performed. This may be because nodes put a higher premium on keeping the cycle to absorb possible losses than compressing to limit exposure to insolvent nodes.

Compression is performed only when the cycle nodes vote unanimously in favor; one dissenting vote vetoes the proposal. Therefore, I questioned whether the number of performed compressions remained low because the majority consensus was against compression or because compression was perceived as harmful by only a small subset of the voting nodes. From my results, it can be seen that even though on average the majority of voting nodes agree to compression, it is often not a strong majority and in several cases the majority on average do not want compression.

3.6.2 Systemic Risk

Additionally, I explore the impact of strategic compression on all banks by measuring systemic risk using the information gathered above. I will introduce three methods by which systemic risk can be measured in these networks using total equity and the set of insolvent nodes. For this analysis, I gather the described data under three scenarios: nodes follow the Nash equilibrium to decide compression, cycles are universally compressed, and cycles are never compressed.

The first measure of systemic risk is the ***price of anarchy***. I define total equity in the network as the sum of equity for all network participants:

$$\hat{E} = \sum_{i=1}^{b} \left[e_{B_i} + \sum_{j=1}^{b} l_{B_j B_i} - L_{B_i} \right].$$

I also define the ***optimal choice*** as the decision for a given cycle (compress or not compress) yielding the greatest total equity:

$$\hat{E}^{*} = \max\{\hat{E^c}, \hat{E^{nc}}\},$$

where $\hat{E}^c$ is the total equity in the network where cycle *c* is compressed and $\hat{E}^{nc}$ is the total equity in the network where *c* is not compressed. Then the price of anarchy can be defined as:

$$PoA = \frac{\hat{E}^s}{\hat{E}^*},$$

where $\hat{E}^s$ is the total equity in the network where the decision to compress cycle *c* comes from a strategic vote.

The results of the price of anarchy computation for all game configurations is shown in Figure 3.6. It can be seen that the average price of anarchy is high and in fact close to 1 for all configurations. Thus, the strategic choice is often the optimal one for the network and becomes closer to optimal as the recovery rate of insolvent nodes increases. When $\alpha = 0$ the price of anarchy is 1, since the optimal strategy is to always compress and compression in this case is never harmful.

Another method for analysis categorizes strategic compression as beneficial, neutral, or harmful to the network. A beneficial compression reduces systemic risk. Thus, the relative number of cases in which compression is not harmful may serve as an indicator for strategic compression's effect on systemic risk. I will introduce two possible methods for distinguishing between a beneficial, neutral, or harmful case.

The first method classifies these cases in accordance with the definitions of Veraart [2019]. Let *SC* be the set of nodes that become insolvent in the network with a strategic compression decision, and *NC* be the set of nodes that become insolvent in the same network without compression. The

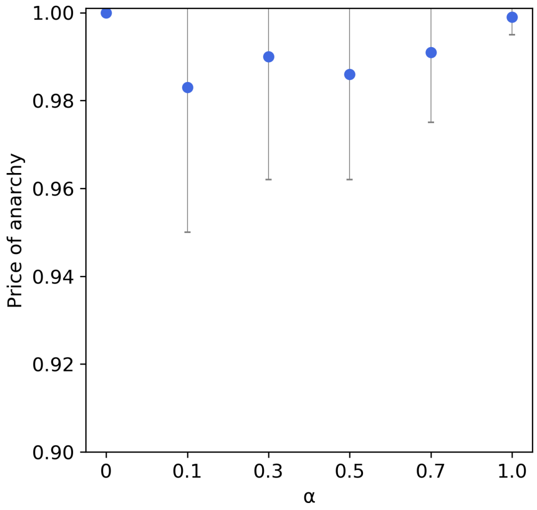


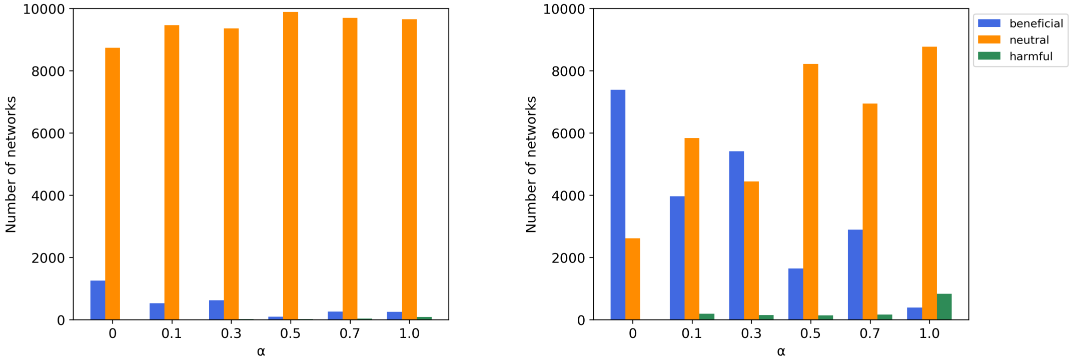
Figure 3.6: The average price of anarchy of the Nash equilibria is high for all α values.

strategic decision is beneficial and reduces systemic risk if $SC \subsetneq NC$; neutral if $SC \subseteq NC$, and harmful if $SC \setminus NC = \varnothing$.

The second method, following Schuldenzucker and Seuken [2021], defines social benefit in terms of total equity in the network instead of the set of defaulting nodes. As above, let $\hat{E}^s$ be the total equity in the network when compression is decided by strategic vote and $\hat{E}^{nc}$ be the total equity in the same network with no compression. The strategic decision is beneficial is $\hat{E}^s > \hat{E}^{nc}$, neutral if $\hat{E}^s = \hat{E}^{nc}$, and harmful if $\hat{E}^s < \hat{E}^{nc}$.

The results of categorizing each network's resulting compression by effects on systemic risk can be seen in Figure 3.7. When using the resulting set of insolvent nodes as the metric for effects on systemic risk, most of the strategic decisions had a neutral effect. In these cases, either compression did not affect the set of banks becoming insolvent or the strategic vote resulted in no compression. When the strategic vote resulted in compression with an effect on the network, in very few cases was this effect negative. However, when looking at the total equity remaining in the system after resolving the network as the metric for systemic risk, the number of cases where the strategic compression is beneficial or harmful to the network increases. Although in most situations the same set of nodes are becoming insolvent, the amount of equity in the system varies.

I further examine the difference in the number of networks where strategic compression was



(a) categorizing by set of insolvent nodes

(b) categorizing by total equity

Figure 3.7: The number of networks where the result of the strategic compression decision was beneficial or harmful depends on the metric used for categorization.

considered beneficial by total equity to those considered beneficial by the set of insolvent nodes. It must be the case that there are networks where the effect is beneficial by total equity, but not by the set of insolvent nodes. Though note that not every network where the effect was beneficial by insolvent nodes is beneficial by total equity.

The results in Table 3.7 show that such networks are overwhelmingly classified as neutral by the set of insolvent banks. Thus, while compression is changing the amount of equity in the system, it is often not affecting the set of nodes that become insolvent. Performing some amount of compression in the network, while not saving insolvent nodes from defaulting, is able to increase the amount of equity among the remaining nodes. This may be due to a decrease in exposure of the remaining nodes to the insolvent ones as a result of compression.

α	Beneficial	Neutral	Harmful
0	1216	6167	0
0.1	500	3469	0
0.3	590	4814	8
0.5	90	1544	12
0.7	234	2629	27
1.0	22	359	14

Table 3.7: When a network is classified as beneficial by total equity, it is generally classified as neutral by the set of insolvent nodes.

3.7 Summary

This study examines the strategic decision facing nodes in a financial network presented with opportunities to remove cycles of debt by a process of debt compression. I focus on the implications for financial contagion, anticipating how compression may insulate or expose an institution to the effects of an asset shock. Taking a perspective ex-ante to the potential shock, I ask what types of strategies nodes with imperfect information should use to make such a decision. This model captures uncertainty regarding the shock and the financial situation of other nodes in the network. I compare strategies that use different types of local information that may be available to an individual node.

From my results, it can be confirmed that adopting a heuristic strategy based on privately held information is often better than unconditionally agreeing or disagreeing for an individual node. In particular, features related to the asset holdings of a node are found to be most useful. When these heuristics are employed, compression proposals are often declined. While based on a stylized model, these results suggest that accepting compression proposals based solely on simplifying balance sheets should be questioned, as institutions may have sufficient basis to account for risks as well as benefits of compression.

I also illustrate the use of the strategic features gains assessment for further analyzing the importance of features in the heuristic strategy space. This analysis revealed an interesting pattern as I varied a key environmental parameter, the recovery rate. In particular, I found that for higher recovery rates, the potential for cycles to cushion shocks was increased, and as a result the compression decision was more inclined to hinge on variables that evaluated this potential.

My experiments also show the effects of a strategic compression decision on systemic risk. I find that while strategic compression can be harmful to the network, it is more often beneficial compared to unconditional acceptance. Furthermore, I can compare the strategic choice to the optimal decision and see that the price of anarchy remains high. In other words, allowing compression to be a decision made strategically by nodes in the network does not substantially increase systemic risk.

As noted, this model is an oversimplification of true financial networks. While it effectively demonstrates unconditional acceptance is not the best strategy for banks, an assessment of additional information available in the true decision context may be necessary to directly inform policy. Further elaborations of the model could account for varieties in debt terms, regulatory constraints, and other realities. Additional types of contracts and instruments, such as credit default swaps, could also be incorporated into the network model [Schuldenzucker et al., 2020]. The analysis approach presented here could likewise be applied in the more elaborated setting, to inform decision making and regulation in specific scenarios.

My analysis in this study treats the compression decision as a one-shot game. An interesting direction for future work would be to extend this model in time, considering a sequence of compression opportunities, asset shock events, debt network evolution, and so on. By extending the model in this way, I could capture signals nodes in the network may be able to gather over time about the stability of the other nodes, similar to a bank's ability to use past payment information to make borrowing and lending decisions.

CHAPTER 4

Flagging Payments for Fraud Detection

4.1 Introduction

Credit card fraud occurs when a malicious party, the fraudster, makes unauthorized purchases using a credit card that does not belong to them. The fraudster may obtain the card's details in a variety of ways such as physically stealing the card, or through electronic database breeches. While a historically notorious problem, increases in online transactions as a result of the COVID-19 pandemic have exacerbated fraud [Stripe, 2023a]. In 2022, $33.45 billion was lost worldwide to payment card fraud, which included credit card fraud [Nilson Report, 2023]. Losses are expected to increase in the future with projected losses of $39.60 billion in 2026.

The majority of these fraud losses are borne by card issuers. As a result, many banks have enacted a myriad of fraud prevention measures. This includes the development of systems, called ***fraud detectors***, for detecting attempted fraudulent charges. Much current research focuses on the development and improvement of these detectors, the latter of which is often focused on increasing the role of automation. My work complements this by seeking to understand the decision context and broader effects of fraud detection on the payment system. Specifically, noting that fraud detection often carries a cost for human and/or technical capital. Spending on financial crime prevention, including fraud prevention, in 2023 was $22.1 billion [Maynard, 2023], while projected spending on artificial intelligence based detection methods for online payments is projected to reach $11.3 billion by 2025. Given this, I am interested in studying how such costs might impact the adoption of fraud detection technology and the resulting effect on patterns of fraud in the network.

I define payment ***flagging*** as a bank action sending a credit card payment to the fraud detector. This introduces the bank's ***flagging problem*** which is, in an environment in which invoking the fraud detector is costly, to decide decide which payments to send to the detector. I investigate this problem in a variant of the extended FCN model where fraudster and customer nodes route payments through a banking network. Each bank node in the model has access to its own fraud detector, which is treated as a black box defined only by its accuracy and cost characteristics. Banks are characterized by the capabilities of their fraud detection systems. A bank node with high fraud detection accuracy is referred to as a strong bank, while a bank node with lower fraud detection accuracy is a weak bank. However, the costs incurred by both bank types remain the same. Within this model, I define a ***flagging game*** played by strategic bank and fraudster nodes. Bank nodes decide which payments to flag, trading off costs of missing a fraudulent payment and of using the detector. Fraudster nodes decide the value of attempted fraudulent payments and how often to attempt fraud. I analyze the strategic behavior of bank and fraudster nodes in this game using EGTA.

From my experiments, I find that high costs for using the fraud detector deter strong bank nodes from flagging many payments, but this is not the case for weak bank nodes. This aligns with results from an analysis of the costs experienced by bank nodes as a result of fraudulent payments in the environment. For strong bank nodes, these costs tend to be balanced between liability for undetected fraud and costs of fraud detection. On the other hand, weak bank nodes' primary costs are the losses from undetected fraud. When banks flag all payments, I find slightly more customer nodes are victims of attempted fraud than if banks strategically flag payments.

4.2 Related Work

Several prior works attempt to categorize instances of fraud by the origin or manner of fraudulent actions [Delamaire et al., 2009, Sakharova, 2012, Onwubiko, 2020]. Beals et al. [2015] introduce a classification framework for describing instances of financial fraud, including payments fraud, to allow for reporting in the National Crime Victimization Survey. In the general fraud detection space, prior work notes the challenges of detection such as concept drift, imbalanced data sets, and timeliness of detection [Abdallah et al., 2016].

The majority of prior work in the fraud detection space, however, focuses on developing systems for detecting fraud more quickly and accurately than human auditors. These systems have evolved from primarily rule-based to statistical data mining and machine learning based techniques. Supervised learning has often been used to train models to discern fraudulent from non-fraudulent payments. However, the challenge of accessing labeled data has made unsupervised methods relying on techniques such as anomaly detection also popular [Ryman-Tubb et al., 2018, West and Bhattacharya, 2016]. Recent literature focuses on leveraging the power of deep neural networks to learn patterns of fraudulent behavior and identify potentially fraudulent payment attempts [Roy et al., 2018, Lin et al., 2021, Zheng et al., 2020]. Several prior works implement a cost sensitive approach for misclassification in the credit card fraud detection case to account for bank liability for fraud [Sahin et al., 2013, Bahnsen et al., 2016]. To my knowledge, these works do not generally address constraints that might limit the use of these systems.

In contrast, Dervovic et al. [2021] study fraud detection as a constrained resources problem. Financial institutions have a limited number of payments they are able to review for fraudulent activity in a given time window. Such constraints may be caused, for example, by a limit in the number of auditors available to review payment information. Thus, institutions must decide as payments are processed, whether to select a given payment for review. Each payment carries a reward for being selected and the goal is to maximize the total reward received. The authors propose the Non-parametric Sequential Allocation Algorithm to address this problem and demonstrate its effectiveness for selecting the most valuable transactions using a public fraud data set. By examining a hard constraint, this work can be seen as addressing short-term constraints in resources. I extend such a problem to the long-term in which costs are the primary constraint.

4.3 The Credit Card Network Model

I define the credit card network model within an extended FCN with fraudster, customer, and bank nodes. The set of bank nodes $B = \{B_1, \ldots, B_b\}$, set of customer nodes $C = \{C_1, \ldots, C_n\}$, and set of fraudster nodes $F = \{F_1, \ldots, F_m\}$ are defined such that $m \ll b \ll n$. Within the network, banks issue credit cards to customers represented by a credit edge extended from a bank node to a customer node as shown in Figure 2.7. Customers draw on their credit cards to make payments to other customers in the network through the interbank network.

An example payment of 10 units from customer node C_1 to C_2 is shown in Figure 4.1. Drawing on the credit card issued by B_1 to make the payment results in a decrease in C_1's available credit for future payments. This is reflected by a decrease in the value on the solid red credit edge from B_1 to C_1 by the value of the payment. By using some credit, C_1 increases the debt it owes to bank B_1 by the value of the payment captured by an increase in the value on the dashed blue debt edge from C_1 to B_1. The payment is routed to the receiver through the interbank network as described in Section 2.1.2.3, creating a debt from B_1 to B_2. Finally, B_2 provides the funds from the payment to C_2 as a debt owed from B_2 to C_2 equal to the payment value. Note the credit edge from B_1 to B_2 and the debt edge from C_2 to B_2 remain unchanged as a result of the payment and have been omitted from the updated network for clarity.

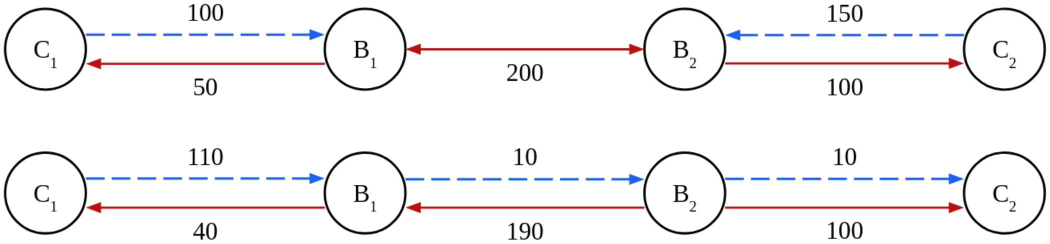


Figure 4.1: Customer C_1 makes a payment of 10 units to customer C_2 by drawing on the line of credit issued as a credit card by bank B_1.

When it is a fraudster initiating the payment, there is only a slight variation in the payment modeling. Figure 4.2 shows a 10 unit payment to C_2 originating with fraudster F_1. A fraudulent payment creates a fraud edge as explained in Section 2.1.2.4 between the fraudster and its victim.

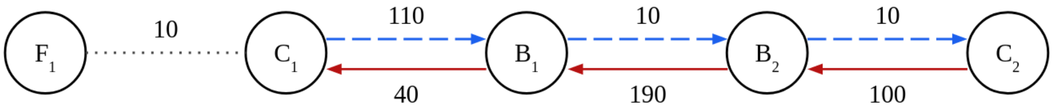


Figure 4.2: The result of a fraudster drawing on C_1's credit card to make a payment of 10 units to customer C_2. With the exception of the fraud edge, the resulting edge updates are the same as those shown in Figure 4.1.

Otherwise, the payment steps remain unchanged and the result of the payment for the remaining parties is the same as the non-fraudulent case.

To simplify the model, I assume bank nodes have infinite willingness to engage with each other on behalf of their customers. Thus, the model is implemented with arbitrarily large values on the credit edges forming the interbank network.

4.4 Game Initialization and General Behavior Overview

The game begins with the initialization of the bank, customer, and fraudster nodes described along with general behaviors below. After node initialization, the customer and fraudster nodes are organized into the initial payments queue.

4.4.1 Bank Nodes

Bank nodes in the model belong to one of two types, strong or weak, which help define their capabilities in the network. When initializing the game the number of bank nodes that should be assigned to each type is taken as input. Banks are then randomly assigned to a type in accordance with this parameter.

The fraud detection capabilities of each bank node are set according to the bank's type during initialization and remain for the entirety of the game. Recall the fraud detectors for each bank are defined as black boxes represented by their parameters. Let $\gamma_{B_i}^{tp}$ denote the probability a bank node B_i's fraud detector accurately labels a fraudulent payment. That is, if a payment is truly fraudulent, B_i's fraud detector labels it fraudulent with probability $\gamma_{B_i}^{tp}$ and as non-fraudulent with probability $1 - \gamma_{B_i}^{tp}$. Likewise, define $\gamma_{B_i}^{tn}$ as the probability B_i's fraud detector accurately labels a non-fraudulent payment as not fraudulent. For this analysis, I will simplify the parameters such that $\gamma_{B_i}^{tp} = \gamma_{B_i}^{tn}$ and will simply refer to the probability B_i's fraud detector correctly labels a payment relative to its true label as γ_{B_i}. The γ_{B_i} value for each bank node is set by a random draw from $\gamma_{B_i} \sim U[0.75, 0.9]$ if B_i is a strong bank and $\gamma_{B_i} \sim U[0.5, 0.65]$ otherwise.

4.4.2 Customer Nodes

The behavior of customer nodes in the network is calibrated from the JP Morgan AI Research (JPMAIR) payments data for fraud detection synthetic data set [1]. A variety of information is available for each payment in the data set including its time step, sender, receiver, value, and whether it is fraudulent or not. Since this set is used only to define customer behavior, I limit use to the non-fraudulent payments resulting in a data set of 47,570 transactions.

During initialization, each customer node is assigned a sender identification number, datasetID, from the data set by a uniform random draw without replacement. For example, customer node C_5 in the network may be assigned datasetID $= 127352$ which is used to establish the credit line and customer payment patterns in the network. Upon initialization, a customer node will be uniformly randomly assigned to a bank node at which point the bank B_i extends a line of credit to the customer creating a credit edge from the bank to the customer. The value on the edge is equal to $2P$ where *P* equals the total value of payments the assigned datasetID sends in the JPMAIR data set. Each customer will attempt payments with a frequency following a Poisson distribution defined by frequency parameter λ. Customer node C_i's λ_{C_i} parameter is calculated during initialization using the number of payments sent by C_i's datasetID, relative to the total number of time steps in the data set.

Customer nodes may experience attempted fraud using their credit lines over the course of the game. It is assumed the means by which fraudsters obtain credit card details does not deter the customer from continuing to use their card. For example, card details could unknowingly have

[1] This publication includes or references synthetic data provided by J.P. Morgan.

been leaked from a database breach while the customer maintains physical possession of their card.

4.4.3 Fraudster Nodes

This model assumes it is equally possible for a fraudster to compromise the credit card information of any given customer node in the network with negligible cost. Thus, a fraudster is not motivated to exploit the credit lines of any one particular customer. Fraudster nodes do, however, differentiate between bank nodes in terms of their ability to successfully commit fraud. To that end, each fraudster is initialized with an empty history of successful attempts at each bank node which will be updated throughout the game. At the start of the game when there is no recorded history, fraudster nodes will randomly select their victim uniformly from the set of all customer nodes. As the history is updated, fraudsters may make more informed selections of bank nodes to target with fraud attempts. It is assumed fraudsters also have negligible cost access to customers' bank nodes assignments and will often favor targeting any customer belonging to a bank where they have historically been most successful. Fraudsters' targeting behavior will be described in more detail in Section 4.6.

Fraudster nodes are also particularly inclined to make payments to a certain subset of customer nodes in the network, who may be more vulnerable to fraud as a result of business practices [Stripe, 2023b]. This set of customers, the ***suspicious set***, is initialized by a uniform random draw of the customer set during initialization. The size of the set, $|TS|$, is drawn such that $|TS| \sim U\{10, \ldots, 20\}$. While the true identity of the suspicious set members is unknown to banks, it is assumed they know such a group exists and have a noisy estimate of its size. Such information could have been obtained through previous analysis of payment history conducted by the bank. Each bank node maintains its own estimates of the suspicious set, referred to as a bank's ***perceived set***. Bank node B_i's perceived set is noisy with respect to both size and membership. The size of the perceived set PS_{B_i} is randomly drawn such that $|PS_{B_i}| \sim U\{|TS| + 1, \ldots, |TS| + 5\}$. Each customer in the suspicious set is included in B_i's perceived set with independent probability 0.9

if B_i is a strong bank and 0.6 otherwise. The remaining spots in the perceived set are filled by a random draw of customer nodes in the network who are not members of *TS*.

4.4.4 Payments Queue

The final step of initialization is the creation of the ***payments queue***, which schedules payment attempts in the network. Each slot in the queue is equivalent to a single time step in the game amounting to a length of *T*. Thus, sequentially processing the queue moves the game forward in time. The queue is initialized by placing customer and fraudster nodes in time slots, referred to as their *arrival* times. For a customer node, this is calculated from the Poisson distribution using its frequency parameter λ. A fraudster's payment frequency is determined by its selected strategy, and ultimate placement in the queue further determined by the calculated arrival time. In particular, the *T* time steps are comprised of alternating, equal length time periods. An ***active*** time period is one in which fraudsters are more likely to be active. Conversely, a ***non-active*** time period is one in which fraudsters are less likely to be active. If the calculated arrival time for a fraudster is during an active period, the fraudster will be added to the queue in this slot. Otherwise, with probability 0.2 the fraudster will arrive in this slot and with probability 0.8 a later time slot is calculated. In the case any node's calculated arrival slot is already occupied, the node will be inserted in the next available time slot. After all customer and fraudster nodes' initial arrival times are calculated, the queue is fully initialized.

4.5 Strategies

After initialization, banks and fraudsters select their strategies from their respective sets defined below.

4.5.1 Bank Strategies

The strategic decision for a bank node in the network is determining which payments to flag when invoking the fraud detector is costly. The flagging process is considered a quick review of surface-level details of payments and thus does not use customer-specific information such as the sender's average behavior, which is often used for a more in-depth analysis by the fraud detector. The strategies of the bank nodes are modeled as logistic functions calculating the probability of flagging based on attributes of the payment. The probability that a bank node flags a payment, ρ, is determined by:

$$\rho = \frac{1}{1 + e^{-k(v-(cx))}} \tag{4.1}$$

where v is the value of the payment being evaluated, k is a game configuration parameter, and x is determined by the bank's strategy.

In particular, when determining whether a payment may warrant flagging, the payment's value may be an important indicator. If a payment's value is high, the bank node may prefer it be reviewed by the fraud detector, since banks are liable for fraud that occurs. Consider $c = 1$ in Equation 4.1 for now. The value of x in the equation determines the tipping point, absent other information, at which payments are considered high and should be more likely to be flagged. Bank nodes will consider three possible values for x:

- *500*: the average of possible payment values in the model
- *pay_avg*: the average payment value of all customers of the given bank node
- *credit_est*: an estimated average payment value calculated from the average ratio of payment value to initial credit line

In addition to its value, two other factors may help indicate whether a payment warrants flagging: the receiver and timing of the payment. Bank nodes use their perceived sets to determine if they believe a payment's receiver is suspicious, and thus more likely to receive fraudulent payments than other members of the network. In this model, bank nodes also know when fraudsters are more likely to attempt fraud. During these so-called active periods, bank nodes may be more interested in flagging payments.

	Active period = T	Active period = F
Receiver in *PS* = T	$c = 0.001$	$c = 0.5$
Receiver in *PS* = F	$c = 0.5$	$c = 1$

Table 4.1: The *reg* matrix increases the likelihood of flagging a payment as more suspicious markers are present.

	Active period = T	Active period = F
Receiver in *PS* = T	$c = 0.001$	$c = 0.001$
Receiver in *PS* = F	$c = 0.001$	$c = 0.001$

Table 4.2: The *most* suspicion matrix is very likely to flag any payment.

	Active period = T	Active period = F
Receiver in *PS* = T	$c = 2$	$c = 2$
Receiver in *PS* = F	$c = 2$	$c = 2$

Table 4.3: The *rare* matrix, is unlikely to flag most payments and does not consider attributes other than value.

	Active period = T	Active period = F
Receiver in *PS* = T	$c = 0.001$	$c = 2$
Receiver in *PS* = F	$c = 2$	$c = 2$

Table 4.4: The *rare_sus* matrix is unlikely to flag payments unless timing and receiver are suspicious, and the value is considered high.

While any of these discussed conditions—high payment values, suspicious receivers, and higher risk timing—may warrant flagging, as more of these characteristics appear in a payment it is more likely banks may wish to flag it. This notion grounds the creation of ***suspicion matrices***, which assign different c values for different combinations of these suspicious events. As a result, the logistic function is shifted allowing for increased or decreased likelihood of flagging. For example, Table 4.1 shows a matrix where the number of suspicion markers present for a payment of value v determines the likelihood it is flagged. As more markers appear, a payment of value v is increasingly likely to be flagged. I refer to this suspicion matrix as the *reg* matrix, and define three additional matrices. The *most* matrix is listed in Table 4.2, the *rare* matrix in Table 4.3, and the *rare_sus* matrix in Table 4.4. These four matrices can be summarized by their general behavior in the presence of suspicious conditions:

- *most*: sends the majority of payments to the detector and only considers payment value
- *reg*: as more suspicious conditions are present, it becomes increasingly likely a payment goes to the detector
- *rare_sus*: sends only payments with a high degree of suspicion to the detector
- *rare*: has a low probability of sending any payment to the detector and only considers payment value

Thus, the matrices control the use of bank fraud detectors based on both general desires for fraud detector use and payment characteristics.

To illustrate the general difference in outcomes under these matrices, consider a payment of 800 units sent to a member of the bank's perceived set during an active period. The *most*, *reg*, and *rare_sus* matrices all return a very high probability of flagging this payment because there are three suspicion markers present. On the other hand, the *rare* matrix returns a very low probability. Now consider a similar payment of 800 units sent to a receiver who is not a member of the bank's perceived set during a non-active period. The *most* matrix is still very likely to flag the payment and the *rare* is still very unlikely to flag the payment. However, with only a suspiciously high value, the *reg* matrix flags this payment with probability less than in the original example. The absence of other suspicion markers results in the *rare_sus* matrix returning a low probability for flagging this payment. Finally, if the payment was sent to a member of the perceived set *or* during an active period, only the *reg* matrix's results would change. With only one suspicious condition true, the payment would be sent the detector with a likelihood between the outcomes in the two previous examples.

A bank node's strategy is a combination of one x value and one suspicion matrix for a total of 12 possible strategies for bank nodes. Strategies are referred to in the form *[x, matrix]*, for example *[500, reg]* refers to a strategy in which the bank node considers a payment value high, absent other information, if it is greater than 500 and calculates the probability of flagging by considering all suspicious conditions.

4.5.2 Fraudster Strategies

Fraudsters aim to maximize the total amount of fraud they commit in the network. To achieve this, fraudsters may use a number of strategies from attempting to evade detectors to taking advantage of their less than perfect accuracy. In the former case, for example, if a fraudster knows larger payment values attract greater scrutiny they may keep payment values below a certain amount. And in the latter, they may exhibit bolder behavior, attempting a high number of payments to increase the odds of success.

To model this, fraudster nodes in this game select from a set of strategies comprised of two components defining fraudster payment attempts: frequency and value. Each component is defined as an interval of possible values from which a fraudster node's behavior is drawn. The frequency with which a fraudster attempts payments is characterized as either *low* or *high* and maps to an interval of values from which to draw the λ parameter, shown in Table 4.5. Recall, the λ parameter defines the arrival rate of a node according to the Poisson distribution. After a strategy is selected, this λ is drawn and remains the fraudster's frequency for the duration of the game. There are four possible characterizations for the payment value, defined as:

- *low*: an interval of low values
- *high*: an interval of high values
- *random*: an interval containing all low and high values
- *alternating*: alternates attempts between a value from *low* and a value from *high*

Each time a fraudster payment is made, the value of the payment is randomly selected from the interval defined by the value component of the selected strategy. The mapping of payment value options to their respective intervals is listed in Table 4.6.

In total, there are eight possible fraudster strategies combining the frequency and payment value choices. Fraudster strategies will be referred to in the format *[frequency, value]*. For example, *[low, high]* refers to the strategy with infrequent payment attempts of high value.

Frequency	Interval
low	$U \sim [200, 1000]$
high	$U \sim [10, 100]$

Table 4.5: A mapping of the frequency component of the fraudster strategy to the interval from which the fraudster's λ parameter is drawn. This parameter is set at the beginning of the game and remains the fraudster's frequency for the duration of the game.

Value	Interval
low	$U \sim \{1, \ldots, 400\}$
high	$U \sim \{600, \ldots, 1000\}$
random	$U \sim \{1, \ldots, 1000\}$
alternate	$v_1 = U \sim \{1, \ldots, 400\}; v_2 = U \sim \{600, \ldots, 1000\}$

Table 4.6: The corresponding intervals from which to draw payment value based on the value component of the fraudster's strategy. This interval is randomly drawn from each time the fraudster makes a payment.

4.6 Fraud Flagging Game Steps

After initialization and strategy selection, the game unfolds over T time steps. Each time step follows the same procedure:

1. The next payment sender is removed from the queue.

2. A payment is generated according to whether the sender is a customer or a fraudster as described below.

3. The sender's bank processes the payment by first ensuring its validity. A valid payment must be less than or equal to the sender's current available credit line and an invalid payment is canceled. Processing of valid payments continues with the following steps:
    (a) The bank uses its strategy to calculate the probability of flagging the payment, ρ. If the payment is flagged:
        - The bank correctly labels the payment relative to its true label with probability γ, as described in Section 4.4.1.
        - A payment labeled as fraudulent by the detector is canceled regardless of the true label.

    (b) Payments that continue processing are used to update the network as described in Section 4.3.
4. The sender's next arrival time is calculated as outlined for customers and fraudsters respectively below.

4.6.1 Customer Payments

Customer node payments are generated by a noisy draw from the JPMAIR data set. For a sender C_i, a payment by C_i's $\mathsf{datasetID}$ is randomly selected from the data set. The value for the customer node's payment, v, is then drawn from the normal distribution $v \sim N(v_{ds}, 10)$, where v_{ds} is the value of the drawn payment. To create the network, only a subset of the $\mathsf{datasetID}$s are selected and thus the payment receiver in the data set is not guaranteed to be present in the network. Therefore, I obtain an index in the network for the receiver by applying a hashing function to the receiver's ID listed in the data set. Specifically, the receiver index is calculated as $\mathsf{datasetID} \mod n$, with n equal to the number of customers in the network. For example, a payment received by $\mathsf{datasetID} = 128372$ would be routed to customer node C_{172} in a network of 200 customers. After attempting a payment, the customer is reinserted into the queue at a new arrival time calculated similarly to the process described during initialization.

With this method, customer node behavior is calibrated from the data set under the assumption customers tend to follow similar payment patterns over time, though with some variation. While not strictly necessary as the JPMAIR data set is itself synthetic, this method is useful for maintaining patterns in the data while preserving privacy of individuals in the actual data set.

4.6.2 Fraudster Payments

Fraudsters attempt payments according to their selected strategies as described in Section 4.5. The value of the payment is determined by a uniform draw from the strategy's payment value interval. While fraudsters may be more likely to make payments to those in the suspicious set, there may be instances in which payments are sent to others. Thus, the receiver of a fraudster's payment will be a member of the suspicious set with probability 0.8, and otherwise a non-member. Among the customers belonging to the selected set, one will be uniformly randomly selected.

If the payment is not canceled at any point in the processing steps, it is used to update the network as described for the fraudulent payment case. Then, the fraudster will increment its number of historical successes at the current victim's bank. To aid in payoff calculations at the end of the game, the total value of these undetected fraudulent payments are recorded for each bank node. On the other hand, if the payment attempt is canceled for any reason, the fraudster node will select a new victim. As mentioned previously, fraudsters target bank nodes and do not differentiate between individual customer nodes. The fraudster will target the bank where it has historically been most successful in the past with probability 0.7 and otherwise target any other bank. Once a target bank has been selected, the fraudster will uniformly randomly select its new victim from among the customer nodes of that bank.

The fraudster's next arrival time will be calculated using its payment frequency as described in Section 4.4.4. If the next arrival time is in an active period the fraudster will be added to the queue at that time step. If the next arrival time is in a non-active period, the fraudster is added to the queue at that time step with low probability and otherwise a new arrival time is calculated.

4.6.3 Payoffs

At the end of T time steps, the fraudster and bank nodes receive payoffs based on the performance of their selected strategies. Fraudster node payoffs are equal to the total amount of fraud they successfully commit in the network. For a given fraudster node F_i the payoff is calculated as:

$$\sum_{j=1}^{n} \sum_{(F_i, C_j, fraud, v) \in E} v$$

The payoff for a bank node in the network, regardless of its type, is the sum of costs it is subject to given the presence of fraudsters in the network, with the best strategy minimizing these costs. The costs for the presence of fraudsters is two-fold. First, banks are liable for undetected fraud that occurs in the network. This is simply the total value of undetected fraud the bank node experiences over the course of the game, which I refer to as ***FC***. The second cost is due to the employment of fraud detection measures. In particular, this decomposes into two sub-costs, the first of which is the cost of sending payments through the fraud detector. I model this cost as a flat fee, β, for each flagged payment similar to the fee structure of several third-party fraud detection management services [Amazon AWS, 2024, Microsoft, 2024, Stripe, 2024]. The total number of payments flagged by a given bank node throughout the course of the game is ***NFD***. Fraud detection technology is also imperfect with a side effect of erroneously labeling payments as fraudulent. These false positives are the second sub-cost to banks in the form of lost transaction fees and inconvenience to customers, which could translate to a loss of business to the bank. To capture this additional cost, I define α_1 as the cost for lost transaction fees, modeled as a percentage of the payment's value the bank node would have received if the payment attempt had not been terminated [Shy and Wang, 2010]. The inconvenience to customer nodes is modeled as a flat cost per terminated payment, α_2. The total value of a bank's false positive payments is written ***VFP*** and the total number of false positive payments is written ***NFP***. Thus, for a bank node B_i their payoff is calculated as:

$$-FC_{B_i} - \alpha_1 VFP_{B_i} - \alpha_2 NFP_{B_i} - \beta NFD_{B_i} \tag{4.2}$$

4.7 Experiments Overview

The flagging game is analyzed using EGTA to identify the Nash equilibrium for bank and fraudster nodes. I analyze networks with $n = 200$ customer nodes, $b = 4$ banks nodes, and $m = 1$ fraudster node. There are two bank nodes of each type in the network. The game is played over $T = 4,320$ time steps and the active and non-active periods last for $A = 72$ time steps. For the bank strategy logistic functions, I set $k = 0.002$. The transaction fee bank nodes could have obtained for routing payments marked as false positives is $\alpha_1 = 0.01$. Experiments are conducted on nine game configurations defining varying costs to all bank nodes for employing fraud detectors. In particular, I test: $\alpha_2 = \{0.5, 1, 2\}$ and $\beta = \{0.1, 1, 2\}$. The lowest value for β is set to be smaller than α_2 to cover the case where fraud detection is quite cheap due to increased automation or other methods, leading the chief cost for banks to be customer satisfaction. A summary of these parameters can be found in Table 4.7.

Parameter	Setting
n	200
b	4 (2 strong, 2 weak)
m	1
T	4,320
A	72
k	0.002
α_1	0.01
α_2	{0.5, 1, 2}
β	{0.1, 1, 2}

Table 4.7: A list game parameters and their settings used in experiments.

After identifying the equilibria, I conduct an additional experiment to understand the effects of the equilibrium decisions on outcomes in the network. In particular, on the fraudster success rate, number of customers experiencing attempted fraud, and the relative costs banks incur. First, I measure these outcomes when bank and fraudster nodes must play according to their equilibrium strategies in each game configuration. In the event of a mixed strategy equilibrium, nodes are randomly assigned to play a pure strategy by a weighted draw according to the probability distribution defined by the equilibrium. For each game configuration, 100 random network instances are simulated and the results are averaged over all runs. If a configuration has multiple equilibria, the results are also averaged across the equilibria. Second, the same methodology is followed in the case where bank nodes flag all payments with probability $\rho = 1$. This reflects a scenario in which all bank nodes may elect, or be forced by policy or regulation, to send all payments through fraud detection. In this case, the fraudster still exhibits strategic behavior and an additional EGTA analysis is conducted to determine its equilibrium strategy in this case for each game configuration.

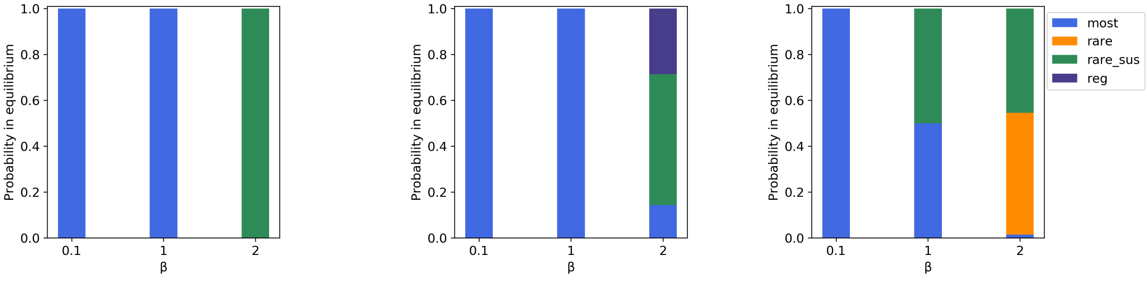
4.8 Experiment Results

4.8.1 Nash Equilibria

4.8.1.1 Bank Nodes

I present an analysis of the equilibrium of bank nodes separately by strategy component, that is the selected x value and suspicion matrix. First, I find that all bank nodes, regardless of type, tend to choose $x = 500$ in equilibrium. Thus, absent other information, as payment values become greater than 500, there is an increased likelihood of flagging. It is possible the other values, which focus on historical averages, are less useful if present payment behaviors deviate enough from the past. In particular, it should be noted fraudster activity is not included in the historical averages. Additionally, from a bank's perspective, minimizing fraud liability often requires gauging the relative risk of the payment by its value. In this environment, the bounds on the value of the payment are known. Thus, the notion of a high payment value may be easily measured relative to the possible payment values rather than requiring additional metrics.

Second, in analyzing the selected suspicion matrices I identify an equilibrium dependence on bank node type. Weak bank nodes almost always select the *most* suspicion matrix for all game configurations. Strong bank nodes, as shown in Figure 4.3, select the *most* matrix when $\beta < 2$ and tend towards limited detector use when $\beta = 2$. That is, when costs are lower all bank nodes flag most payments for review. However, when costs increase, strong bank nodes become more selective and only flag very suspicious payments. Simply put, a better fraud detector is used less frequently than a weaker one when costs for invoking fraud detection are high.



(a) $\alpha = 0.5$ (b) $\alpha = 1$ (c) $\alpha = 2$

Figure 4.3: When detector costs are low, strong bank nodes select strategies that flag most payments with high probability. However, when costs rise they tend to favor strategies that result in few flags.

The contrast in behavior of strong and weak banks is explained by the behavior of the fraudster node. Recall, the fraudster node has a tendency to target bank nodes where it has historically been most successful committing fraud. Other things being equal, this will be a weak bank due to its poorer fraud detection capabilities. As a result, strong banks are less often the targets of attempted fraud. With less of a concern for fraud and rising costs of detectors, these banks lower their use of fraud detection in an effort to limit their overall costs. In this case, the behavior of fraudsters and existence of banks with worse fraud detection capabilities in the network allow strong bank nodes to adjust their strategic behavior.

4.8.1.2 Fraudster Nodes

In all game configurations, the fraudster's equilibrium strategy is *[high, high]*. Under this strategy, the fraudster will very frequently attempt payments of high value using customer nodes' funds. In doing so the fraudster node is hoping to take advantage of the inability for imperfect fraud detectors to identify all its payment attempts. The fraudster is playing the odds and making it worthwhile with large payment values

To aid in the analysis of outcomes in the network, I obtain the fraudster's equilibrium in the case when all banks flag all payments. I find in this case the fraudster's equilibrium remains the *[high, high]* strategy. Even though all payments are flagged, fraud detectors remain imperfect and thus the fraudster is continuing to exploit this as explained previously.

4.8.2 Impact on Network Outcomes

My analysis of fraudster outcomes identifies an average success rate of 38% across all game configurations when nodes play according to the Nash equilibria. Despite this, the fraudster is still able to commit an average of 23,083 units of fraud by attempting very frequent, high value payments. Thus, the fraudster's gamble with the imperfect nature of fraud detectors pays off. When banks flag all payments with probability 1, the average success rate drops to 28% with 18,051 units of fraud committed. Thus, while increased flagging prevents some fraud, it will not completely eliminate it since fraudsters can continue to gamble with imperfect detection.

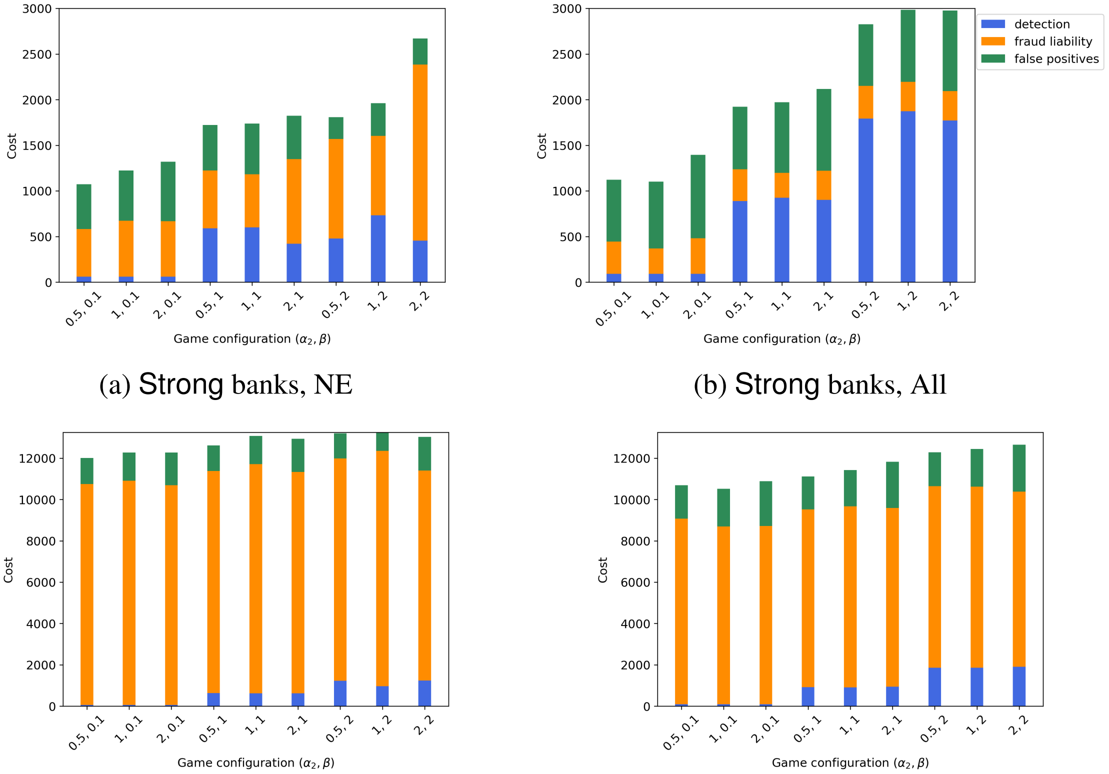
Fraudsters target less than 10% of their payment attempts at strong bank nodes, providing quantitative evidence for the intuition behind weak banks' strategic behavior. Given the fraudster's proclivity for targeting such banks, their desire to flag most payments is logical. The presence of fraudsters in the network does not widely impact customer nodes. On average, only 17% of customers fall victim to attempted fraud when nodes follow the equilibrium strategies. This average increases slightly to 20% when banks flag all payments as a result of fraudster behavior. Recall, the fraudster will switch victims anytime its payment attempt is canceled. In my earlier analysis, I determined that the success rate of the fraudster is lower when banks flag all payments. Thus, the fraudster experiences more canceled payments resulting in increased victim switching and a greater number of customers experiencing attempted fraud. The increase indicates banks and customers may be slightly more inconvenienced when banks take a stricter approach to fraud detection. In the real world, when a fraud attempt is discovered it usually triggers the bank to issue new credit cards to the victim. This introduces a cost to banks for issuing the new cards, and for customers in time delay to receive new cards and begin to use them (e.g., updating automatic payments). Generally speaking, this may introduce a potential trade-off between customer experience and fraud liability for banks. These fraudster and customer outcomes are summarized in Table 4.8.

	NE	All
Fraudster success rate	38%	28%
Total fraud committed (units)	23,083	18,051
Proportion of customers who are victims	0.17	0.2

Table 4.8: Outcomes for fraudster and customer nodes in the network when nodes play according to equilibrium (NE) and when banks flag all payments (All).

The total cost to bank nodes for the presence of fraudsters in the network is three-fold: liability for undetected fraud, cost of using the detector, and the cost of possible false positives. I explore the distribution of these costs experienced by bank nodes, both under equilibrium and in the case all payments are flagged, in Figure 4.4. From this analysis, two important conclusions can be drawn. First, comparing the cost break down for strong and weak banks provides strong evidence for the intuition behind banks' strategies in equilibrium. It can be seen in Figure4.4c, liability for undetected fraud is by far the largest proportion of costs experienced by weak bank nodes. Logically, this follows from the findings that fraudsters are much more likely to attempt fraud at these banks and the knowledge that their fraud detection capabilities are poor. With fraud posing a much greater threat to weak banks, they are less inclined to take the rising cost of fraud detection into consideration when strategically flagging. Thus, they opt to flag most payments, regardless of the current cost of fraud detection. In contrast, strong bank nodes experience more balanced costs, as shown in Figure 4.4a, and so rising fraud detection costs influence the strategic flagging process.

Second, I find strong bank nodes are better off when strategically flagging. The total costs for these banks in almost all game configurations are less under Nash equilibrium (Figure 4.4a) than when all payments are flagged (Figure 4.4b). This is especially evident further along the x-axis where the cost of fraud detection is highest. When strong banks flag all payments, the majority of the cost is attributed to fraud detection for all but the lowest detection cost. Strategically flagging allows strong banks to redistribute their costs in a more favorable way and decrease overall costs. Weak banks, as favored targets of fraudsters, are better off flagging all payments with probability 1 as evidenced by the slightly lower costs in this case compared to under the Nash equilibria. Note, while the most matrix has a higher likelihood of flagging payments, it will still take into consideration payment value and not all payments will be flagged.



(a) Strong banks, NE

(b) Strong banks, All

(c) Weak banks, NE

(d) Weak banks, All

Figure 4.4: A breakdown of the average bank costs when banks play according to the Nash equilibrium (NE) and when they flag all payments (All).

4.9 Summary

In this chapter, I propose the flagging problem in which banks incur a non-negligible cost for invoking credit card fraud detectors. As a result, banks may wish to strategically flag payments for review by detectors so as to balance the cost of using the detector with potential losses from fraudulent payments. To analyze the flagging problem's affect on the adoption of detectors and patterns of fraud, I introduce the flagging game played by fraudster and bank nodes in the extended FCN. The bank nodes in the model differ in their detection capabilities, but all experience the same costs for use of their detectors. Bank nodes consider these costs when selecting a strategy that determines the probability a given payment is flagged. Fraudster nodes, with the singular goal of maximizing the amount of fraud they can commit, select a strategy defining the frequency and value of their payment attempts.

My experiments show fraudster nodes always prefer to frequently attempt payments of high value, likely to take advantage of the imperfect fraud detectors employed by bank nodes. An analysis of outcomes in the network finds this strategy generally pays off for the fraudster, who is able to successfully commit fraud in the network.

All bank nodes flag most payments when detector costs are low. However, when costs become high, strong bank nodes become more selective with their flagging and weak bank nodes continue to flag most payments. This deviation in behavior is attributed to the fraudster's more frequent targeting of weak bank nodes. As a result, weak banks are primarily concerned with the costs from fraud liability and rising detection costs do not factor into their strategic decision. Strong banks nodes experience more balanced costs and find the threat of fraud to be minor compared to the incurred costs of invoking the detector. Therefore, when detection costs are high, strong banks benefit from the existence of weak banks in the system and the tendency for fraudster nodes to target such banks. This behavior demonstrates the importance of considering other banks' capabilities when making fraud-related strategic decisions.

This model demonstrates, to my knowledge, the first known attempt at explicitly capturing the strategic interaction between banks and fraudsters. The results indicate there are useful insights to be gained from studying the fraud problem in this manner and further expansions of the model could broaden our understanding of these interactions. For example, including additional dimensions along which the flagging and fraudster decisions are made, expanding the study to different cost schemes for banks, and adding a cost for fraudsters to acquire customer card details. The current model assumes equal accuracy on fraudulent and non-fraudulent payments, which could be relaxed to better reflect the realities of learning fraud detection from imbalanced data sets. Simply reducing the accuracy of detectors on non-fraudulent payments may not greatly effect the results of this work, as it is likely to simply increase the successful fraud at weak banks due to the behavior of the fraudster node. However, expanding the specifications of fraud detectors may be useful in understanding the strategic implications for bank nodes employing different overall goals for detection (e.g., stressing recall over precision). The model could also be extended from the black box fraud detectors to more sophisticated implementations, and could even support comparing specific fraud detection models. Such an analysis may provide greater insight into interactions between systems deployed in the real world.

CHAPTER 5

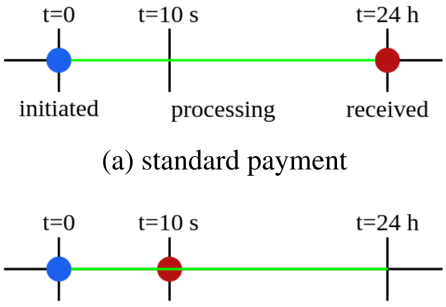
Adoption of Real-Time Payments with Credit Risk

5.1 Introduction

The distinction between a standard payment and real-time payment lies in its speed and availability. Standard payments often experience a processing delay in the clearing and settlement steps as a result of batch processing and interbank messaging protocols [Committee on Payments and Market Infrastructures, 2016]. This leads to delays in payment receivers' access to funds, which can range from a couple of hours to several days. These delays can be particularly significant for payments initiated outside of business hours. Conversely, a ***real-time payment*** (RTP) is characterized by immediate receipt of the funds by the receiver [Committee on Payments and Market Infrastructures, 2016]. This immediacy is guaranteed even when the payment is initialized outside of regular business hours.

Many RTP systems are in use today around the world, including the Faster Payments Services (FPS) in the United Kingdom, the Real-Time Clearing (RTC) system in South Africa, Swish in Sweden [Committee on Payments and Market Infrastructures, 2016], and FedNow in the United States [Federal Reserve Board, 2021]. While these systems are required to provide immediate receipt of funds, their implementation of clearing and settlement may vary. This work particularly focuses on real-time payments with deferred settlement, the method employed by the FPS and RTC systems. In the case of ***deferred settlement***, an irrevocable credit of funds to the receiver's account occurs before the clearing and settlement steps complete [Committee on Payments and

Market Infrastructures, 2016]. The clearing and settlement steps themselves often occur in batches similar to standard payments. An illustration of a standard payment versus an RTP with deferred settlement is shown in Figure 5.1. In the standard processing case, a payment is initiated in time step $t = 0$, followed by processing, and the final receipt of the payment in time step $t = 24$h. Conversely, the RTP with deferred settlement is initiated in $t = 0$ and results in receipt of payment in $t = 10$, followed by processing time akin to a standard payment.



(a) standard payment

(b) real-time payment with deferred settlement

Figure 5.1: A depiction of the standard payment and RTP with deferred settlement timelines from initiation to final receipt of funds. The standard payment's funds are not received until processing completes, but an RTP releases the funds immediately and finishes processing the payment in a time frame similar to a standard payment.

The increasing provision of these systems is predominantly driven by customer demand, especially in developing countries [Mastercard, 2020]. The immediacy of RTPs offers a great benefit to customers. In an emergency or otherwise sensitive situation, knowledge of immediate receipt can provide the sender peace of mind. Even in non-emergency scenarios, payment senders can benefit from RTPs. Consider an ecommerce merchant who insists on receiving funds before the purchased goods will be shipped. An RTP allows the customer to receive their goods in a more timely fashion, while also benefiting the merchant who gains the opportunity to immediately employ the funds back into their business. RTPs are not without risks, however, especially for banks. The expediency of these payments may make it difficult to catch potential problems, for instance due to fraud or other errors, before payment processing begins. The deferred settlement case, in particular, necessitates banks assume a credit risk and potential liability in the event problems arise.

I seek to understand how credit risk might impact bank adoption of RTP systems with deferred settlement. In particular, which customers will be offered real-time payment options. To this end, I implement the extended FCN model with customer and bank nodes, and the ability to distinguish between standard and real-time payments. Customer nodes in the network have a general preference for sending RTPs and will prefer to conduct business at a bank that offers them. On occasion, customer nodes in the model will initiate a payment with a value exceeding their deposit holdings at the time of settlement. So called ***overdrafts*** in this case are a byproduct of the deferred settlement mechanism and introduce a credit risk for banks, who extend short-term credit to senders enabling these payments. As a result, short-term liability is assumed by banks. Within this model, a game is played by bank nodes who must decide which customers, if any, should be allowed to send RTPs in the network. This is handled by selecting from strategies that set the prerequisite minimum threshold of initial deposits required for customers to be allowed to send RTPs over the course of the game. When selecting their threshold, banks must strategically balance the benefits of offering RTPs, including the ability to attract customers, with the cost of potential exposure to overdrafts.

I analyze the game using EGTA to identify the Nash equilibrium for various model parameter settings controlling customer behavior and standard payment delays. Outcomes when banks follow the equilibria are compared to the case when all customers are allowed to send RTPs to understand the impact of strategic choices on the network. I find that banks in the model tend to select strategies that set positive, but low thresholds on the initial deposit holdings required to send RTPs. As a result, most—but not all—customers have access to the service. In aggregate, banks are generally better off when all customers are allowed RTPs, as it increases the overall volume of successful payments and decreases overdrafts. Nevertheless, individual banks are generally unwilling to assume a level of risk to grant all their customers the use of RTPs leading to the sub optimal outcome instead.

5.2 Related Work

Bech [2008] employ game theory to study liquidity in the real-time gross settlement (RTGS) system, which handles settlement of wholesale payments between banks. The study analyzes bank management of intraday liquidity under different credit policies of the central bank. The authors find this scenario leads to two well-known games: the prisoner's dilemma and the stag hunt. Johnson et al. [2004] specifically study liquidity in the deferred settlement instance of the RTGS system by using historical data from the US Federal Reserve's Fedwire Funds Service.

To better understand adoption of RTPs, two prior works analyze related proxy systems. Bech et al. [2017] identify parallels in the adoption of RTGS systems and RTPs, particularly noting both experienced a 15-year gap between first introduction and prominent worldwide up-take. Hayashi and Toh [2020] note the importance of mobile banking as a gateway for accessing RTPs and identify gaps in usage that will need to be addressed to promote wide adoption of RTPs.

Much prior literature on the topic of RTPs provides introductions to these relatively new additions to the payments space. Such works tend to focus on their distinction from standard payments and explain benefits and risks for participants [Committee on Payments and Market Infrastructures, 2016, Hartmann et al., 2019, Santamaría, 2019]. In particular, several works focus on the potential increased liquidity demand these payments may place on banks [Galbiati and Soramaki, 2008, Hellqvist and Korpinen, 2021]. There also exist several studies focusing on the design of RTP systems [Guo et al., 2015, Kulk, 2021, Guo and Ma, 2023], including proposing blockchain implementations [Arshadi, 2019, Zhong et al., 2019].

Several authors provide analyses of the effects of existing RTP systems on the broader payments ecosystem. A study of the FPS in the United Kingdom finds heavy use of RTPs for standing orders and credit card bills, but almost no effect on point-of-sale transactions [Greene et al., 2014]. Another study comparing the adoption of RTP systems in six countries finds the greatest impact of RTPs is on traditional credit transfers and the use of checks [Hartmann et al., 2019]. It also identifies the most important factors for successful adoption of RTPs.

To my knowledge, none of the existing literature specifically address how banks might decide who can use RTP systems. Given the potential risk to banks, it is worth exploring how they may place a limitation on customer use and explore how such a limitation may be chosen.

5.3 Payment Model

The payment model employed is an extended FCN consisting of a set of bank nodes $B = \{B_1, \ldots, B_b\}$ and customer nodes $C = \{C_1, \ldots, C_n\}$. Bank nodes in the network hold deposits on behalf of customer nodes as shown in Figure 2.6, who draw on those deposits to make payments to other customers in the network. Customers may have the opportunity to make one of three types of payments: standard, real time, or overdraft. The latter, as I will explain, is a special case of a real-time payment in this model. To differentiate between standard and real-time payments, I must introduce time more explicitly into transactions within the payment model. All payments will follow the same edge updates as detailed in Section 2.1.2.3. However, the order and timing of the updates will be defined to reflect characteristics unique to each payment type. This will be aided by the use of a ***payments queue***, ***Q***, for each bank node. In this case, the queue will store *payments* sent by customers of a given bank node that have yet to finish processing.

5.3.1 Standard Payments

Standard payments are subject to batch processing, introducing a time delay. To capture this, standard payments sent in the model will not be used to update the edges in the network until a later time period. Until then, the payment is stored in the sender bank's payments queue. At regularly scheduled intervals, referred to as the ***clearing period***, all queues are cleared by removing all payments from the queues and using them to update the appropriate edges in the network.

An example of this process is shown in Figure 5.2. Time step $t = 0$ shows the network before a payment is initiated. In time step $t = 1$, customer node C_1 initiates a standard payment of 10 units to customer node C_2. Note, this is the same payment modeled in Figure 2.9. At this point, nothing in the network changes and the payment is stored in B_1's payments queue, Q_{B_1}. It is not until the next clearing period in time step $t = X$ that the payment is removed from the queue and used to update the network. After the edge updates, the network appears the same as in Figure 2.9.

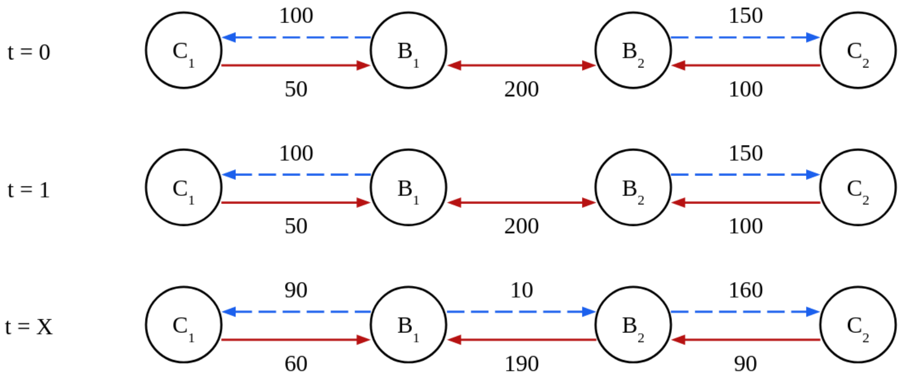


Figure 5.2: An example of a standard payment of 10 units initiated at time step $t = 1$ by customer node C_1. The network remains unaffected by the payment until processing finishes in the next clearing period.

5.3.2 Real-Time Payments with Deferred Settlement

Real-time payments with deferred settlement are characterized by both immediate receipt of payment and the use of batch processing. To capture immediate receipt of payment, the edges between the receiver and its bank are updated in the same time step in which the payment is initiated. Deferred settlement is modeled by adding the remaining payment steps to the sender bank's payments queue. As in the standard payment case, the payment is removed and used to update the network during the next clearing period.

An example of customer node C_1 making an RTP of 10 units to customer node C_2 is shown in Figure 5.3. When the payment is initiated in time step $t = 1$, the edges between B_2 and C_2 are immediately updated to reflect C_2 receiving the payment funds. The remaining edge updates, however, do not occur until the next clearing period $t = X$ to reflect deferred settlement between the remaining payment parties. Note that after the clearing period, standard and real-time payments have the same effect on the network.

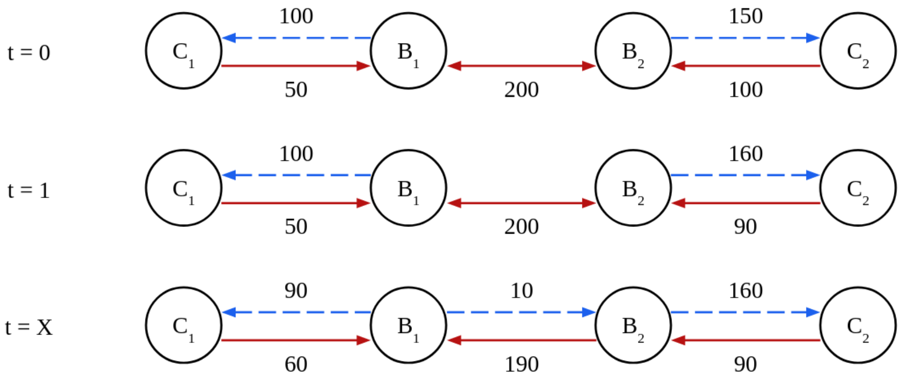


Figure 5.3: Customer node C_1 makes a real-time payment of 10 units to customer node C_2. The payment creates an immediate change to the edges between the receiver and its bank, but the remaining payment steps are not implemented until the next clearing period.

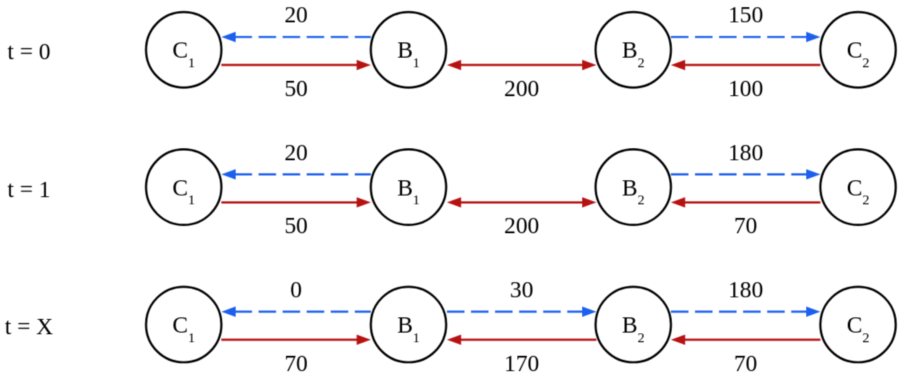


Figure 5.4: An example overdraft payment of 30 units from customer node C_1 to customer node C_2. In this instance, bank node B_1 supplies the remainder of the payment value to allow the payment to process.

5.3.3 Overdraft Payments

On occasion, a customer may make a payment whose value exceeds the sender's current account holdings referred to as an overdraft [Parrish and Frank, 2011]. When overdrafts occur, banks supply the difference to allow the payment to process. Banks are willing to engage in this practice in the short term. However, in the long term, liability is passed back to customers through overdraft fees.

In the payments model, overdrafts are a byproduct of the expediency of RTPs, which do not allow bank nodes time to conduct thorough checks on the customer's account. Such payments occur when the value of a customer's RTP exceeds its deposit holdings *during settlement*. When an insufficient payment occurs in the model, the sender's bank will supply the difference to allow the payment to process. The total difference supplied by bank nodes on behalf of their customers is called the ***overdraft coverage***.

Figure 5.4 shows an example of customer node C_1 making an RTP of 30 units. C_1's current deposit holdings are 20 units, which is insufficient to cover the payment amount. Since the payment is sent in real time, the receipt of the payment occurs in the time step in which it was initiated as in the previous RTP example. However, when the payment finishes processing during the next clearing period, there are some differences. In an overdraft situation, the sender's bank node will take the sender's remaining deposits for the payment. In this case, the dashed blue debt edge from B_1 to C_1 will now have a value of 0 units. With 20 fewer units in its account, C_1's willingness to hold additional deposits increases by the value removed from the account. Bank node B_1 will supply the remaining 10 units to the payment and pass on the full value, 30 units, to the receiver's bank. Thus, an overdraft creates an asymmetry in the settlement amounts between the sender and its bank, and the sender and receiver banks.

5.4 Payments Game

The payments game is played by bank nodes in the extended FCN described above. Bank nodes must decide which of their customer nodes, if any, are allowed to send RTPs by selecting from one of six available strategies. After payments are made in the network over T time steps, banks receive a payoff for the performance of their chosen strategy.

5.4.1 Initialization

During the initialization phase, bank and customer nodes are created to form the financial network. As part of their initialization, bank nodes will select their strategy for allowing customer access to RTPs which is also used to initially attract customers to their bank.

5.4.1.1 Customer Nodes

Customer node C_i is initialized with a random amount of initial deposits ID_{C_i}, drawn from an exponential distribution with average ID_a. As a result, the majority of the customers in the network will start with approximately average deposit holdings while a minority will be part of very wealthy group of individuals. Each customer node is also assigned uniformly randomly to a preference for receiving payments. Either a customer will be willing to accept any kind of payment (*Any*) or will only wish to accept RTPs (*RTP-only*). Lastly, customer nodes are initialized with an ***attentiveness*** representing the likelihood a customer node checks for the possibility of an overdraft when initializing an RTP. The attentiveness of customer nodes is a game configuration parameter in the range of $a \in [0, 1]$ which applies to all customer nodes equally. In the event the customer identifies a possible overdraft, it will adjust its payment as described in Section 5.4.2.

5.4.1.2 Bank Nodes

Bank nodes are initialized to form the interbank network with infinite willingness to route payments on behalf of customer nodes. Upon initialization, each bank node selects its strategy to determine which customers will be allowed to send RTPs. Each strategy sets a different minimum threshold on initial deposit holdings a customer node must have to be allowed to send RTPs. Strategy thresholds are referenced relative to the average initial deposits held by customer nodes in the network (ID_a). They range from requiring initial deposit holdings below average to those requiring greater than average deposits, as well as including all and no customer allowance scenarios. A list of all bank strategies can be found in Table 5.1. As discussed, bank nodes are liable in the short term for any overdrafts that occur. On the other hand, bank nodes can use the promise of RTPs to attract business from customers. Thus, the strategic real-time payments decision for bank nodes is to select a strategy balancing the benefits of RTPs with the credit risk banks assume in the event of overdrafts.

Strategy name	Customer C_i allowed to send RTPs if:
0%	all customers allowed
25%	$ID_{C_i} > 0.25 \times ID_a$
50%	$ID_{C_i} > 0.5 \times ID_a$
100%	$ID_{C_i} > ID_a$
200%	$ID_{C_i} > 2 \times ID_a$
infinity	no customers allowed

Table 5.1: A mapping of bank strategy name to the minimum threshold of initial deposits it requires to allow customer node C_i the use of RTPs.

5.4.1.3 Network

Customer nodes are assigned to bank nodes under the assumption that they prefer banks that allow them to send RTPs, but do not further differentiate between banks in any way. It is further assumed that the selected strategies of bank nodes are public knowledge at the time of assignment. The specific assignment procedure is as follows:

- If no bank would allow a customer to send RTPs, the customer is uniformly randomly assigned to any bank.
- If there exists one or more banks that would allow the customer to send RTPs, the customer is uniformly randomly assigned to any such bank.

Upon assignment, a debt edge is created representing the customer depositing their initial deposits into an account at the bank. I assume customers have an infinite willingness to hold additional deposits in their accounts, creating a credit edge from the newly assigned customer to its bank of an arbitrarily high value. These assignments last for the duration of the game.

5.4.2 Game Steps

The game then unfolds over T discrete time steps. Each time step begins by checking if it is a clearing period and if so, clearing all payments from the bank queues to update the network as described in Section 5.3. Clearing periods occur every χ time steps. Next, η new payments are created in the network in the following manner:

1. Randomly select the sender from the customers in the network.
2. Randomly select the receiver from the remaining customers in the network.
3. Draw the payment value, v, from a uniform $v \sim U\{1, \ldots, 100\}$.
4. Determine if the payment is sent as a standard payment or RTP by the receiver's preference such that:
    - If the sender can send RTPs, it is sent in real time.
    - If the receiver is *RTP-only* and the sender cannot send RTPs, the payment attempt is terminated.
    - Otherwise the payment is sent as a standard payment.

To aid with payment calculations, I define customer node C_i's ***available funds*** (A_{C_i}) as its current deposit holdings minus the value of any pending payments C_i has sent in the network. When an attempted payment's value is greater than the sender's available funds ($v > A_{C_i}$), I consider it a ***potential overdraft***. This is only a *potential* overdraft because it is possible for the sender to receive enough funds between its initiation and final settlement to render this payment valid. However, if it were sent in the present time step, there would be insufficient funds to cover the full amount. Bank nodes in the network will cancel a potential overdraft attempted as a standard payment, but a potential overdraft RTP can only be caught by the customer. With probability equal to customer node attentiveness, a customer node sending an RTP will check its payment information before sending. If customer node C_i checks a potential overdraft RTP, it will adjust the value to be $v = max(0, A_{C_i})$. A payment value of 0 triggers termination of the payment attempt. Any payment attempts not canceled during the creation phase will be processed as described in Section 5.3.

The final time step consists of only the queue clearing step and no new payments are created. After the game concludes, bank nodes will receive a payoff for the performance of their strategies.

5.4.3 Payoffs

Bank payoffs consider the benefits of RTPs and the short-term liability for overdraft payments. In particular, the deposits a bank holds for its customers may be viewed as a representation of the bank's ability to conduct business. However, the deposits themselves are a liability that must be paid upon demand. I model the utility a bank node receives from attracting the initial deposits of all its customer nodes (***ID***) as a fraction of their total value. Similarly, the utility a bank derives from routing payments in real time for its customers is modeled as a fraction of the payment's value representing customer satisfaction and bank benefits such as fees that could be charged for routing these special payments [Deloitte, 2019]. Under this payoff scheme, customers are assumed to derive slightly more satisfaction or are willing to pay slightly more in fees for the ability to send a 100 unit payment in real time than a 5 unit payment. I refer to the total value of RTPs routed by bank B_i as $\mathbf{R_{B_i}}$. Finally, the loss to bank nodes in the game is equal to the total overdraft coverage required (***OC***) from credit provided to customers to allow for the processing of overdraft payments. In the short term, banks hold full liability for these payments, even if they are able to transfer liability to customers in the long run.

Thus, for a bank node B_i, the payoff is:

$$\alpha ID_{B_i} + \beta R_{B_i} - OC_{B_i}$$

5.5 Empirical Game-Theoretic Analysis

5.5.1 Model Parameters and Game Configurations

I analyze different configurations of the payments game using the EGTA methodology to identify Nash equilibrium strategies employed by bank nodes. My experiments are conducted on games with $n = 225$ customer nodes and $b = 3$ bank nodes. Customer nodes have an average initial deposit holding of $ID_a = 1,000$. All games are played over $T = 720$ time steps with $\eta = 45$ payments attempted in each. The game configurations are differentiated by the parameter setting for the clearing period length and customer attentiveness. I run experiments with $\chi \in \{4, 6, 12, 24\}$ and $a \in \{0, 0.25, 0.5, 0.75\}$ for a total of 16 different configurations. Bank payoffs are calculated using $\alpha = 0.5$ and $\beta = 0.02$. These parameter settings are listed in Table 5.2.

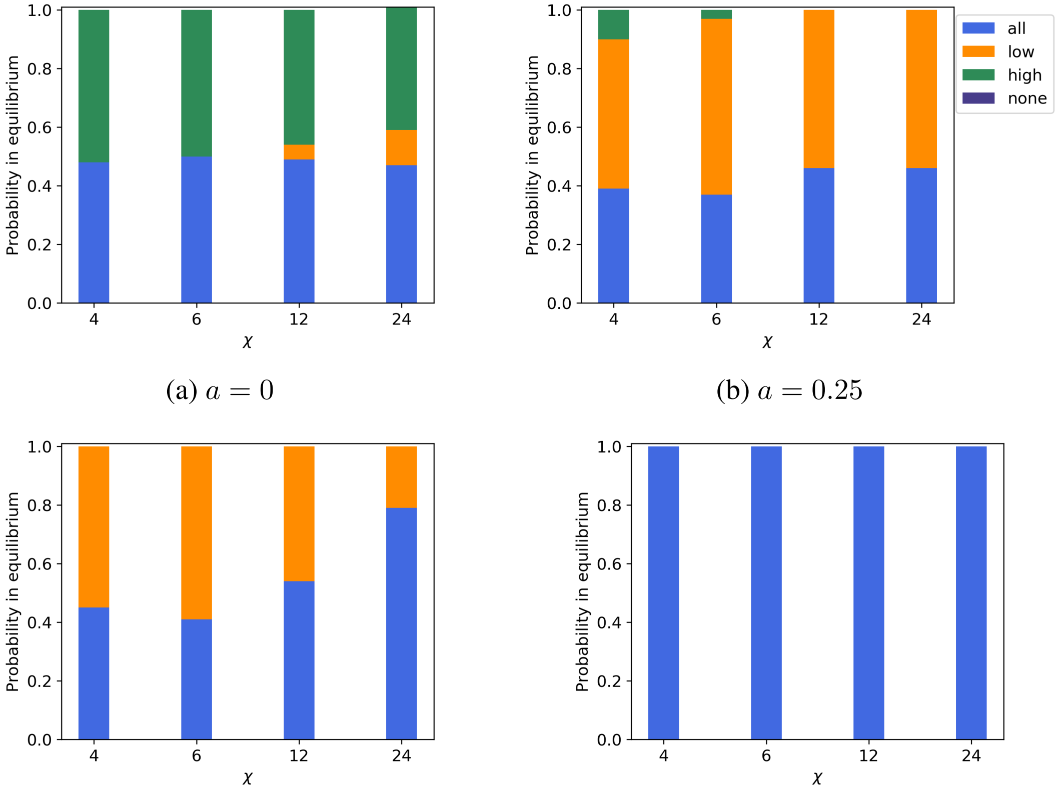
Parameter	Setting
n	225
b	3
T	720
η	45
χ	{4, 6, 12, 24}
ID_a	1,000
a	{0, 0.25, 0.5, 0.75}
α	0.5
β	0.02

Table 5.2: A list of parameter settings for the payments game.

5.5.2 Nash Equilibria

To aid in analyzing the equilibria, I sort the bank strategies into four groups describing their allowance of RTPs. The all group contains only the strategy that allows all customers to send RTPs, while the none group contains the strategy that does not allow RTPs to any customers. Strategies in the low group require fewer than the average amount of initial deposits to send RTPs. Conversely, strategies in the high group require greater than average initial deposits.

The total probability assigned in equilibrium to playing strategies from each group is reported in Figure 5.5 for each game configuration. The results show that as the clearing period and customer attentiveness increases, banks nodes are more likely to adopt lower thresholds. And when customer attentiveness is 0.75, bank nodes allow all customers the use of RTPs. Intuitively, as customer nodes are more likely to confirm validity of their payments, the number of required overdrafts decreases. Thus, bank nodes may be more willing to allow broader use of RTPs. On the other hand, a shorter clearing period creates a shorter delay for processing of standard payments. The difference between an RTP and a standard payment becomes smaller for customers. In such a case, it may not be as worthwhile for bank nodes to provide as many customers with RTP use.



(a) $a = 0$

(b) $a = 0.25$

(c) $a = 0.5$

(d) $a = 0.75$

Figure 5.5: The total probability bank nodes employ each strategy group under Nash equilibrium for each game configuration.

5.6 Comparison of Impact on the Network

5.6.1 Experiment Overview

I aim to understand how the bank node equilibria affect outcomes in the network. To study this, bank nodes in the network play according to the equilibrium identified in the EGTA analysis in 1,000 random instances for each payment game configuration. For mixed-strategy equilibria, the bank nodes are assigned to play a pure strategy in the support according to a weighted draw based on the probability assigned to each strategy in equilibrium. Over the many runs of each game configuration, I measure customer nodes' access to RTPs, the success of payment attempts, and metrics related to overdraft payments.

To further understand the impact of the Nash equilibria, I compare these results to two additional cases. In the first case, all bank nodes allow all customers to send RTPs regardless of initial deposits, equivalent to all bank nodes selecting the 0% strategy. This represents a situation in which banks might be required, for instance by federal regulation, to allow everyone access to RTPs. Conversely, in the second case, bank nodes do not allow any customer nodes to send RTPs, representing the status quo before RTP systems were introduced. Note, some customers in the network will still be initialized to demand RTPs-only. In this case, such a preference can be viewed as an unwillingness to accept payments through the banking system without an RTP system in place. I will refer to these cases as the all and none cases respectively.

5.6.2 Impact Experiment Results

First and foremost, the strategies available to bank nodes may lead to limited customer access of RTPs. To investigate the impact of the equilibria on customers, I examine the proportion of customers allowed to send RTPs in each game configuration. The results are shown in Figure 5.6. I find that no fewer than 90% of customer nodes have access to RTPs in the configurations tested and in a few cases, all customer may send RTPs. Furthermore, the proportion tends to increase as the clearing period lengths increase and customers become more attentive to RTPs. Logically, these trends align with patterns in the equilibrium selection of bank nodes. As both the clear period and customer attentiveness increase, bank nodes become more likely to select strategies with lower threshold requirements for RTP use. This leads to an increased number of customer nodes meeting the qualifications for sending RTPs, as reflected in the increasing proportions shown. The exception to these trends occur when customer attentiveness is 0.75 and the pure strategy Nash equilibrium is to allow all customers RTP access for all clearing period lengths.

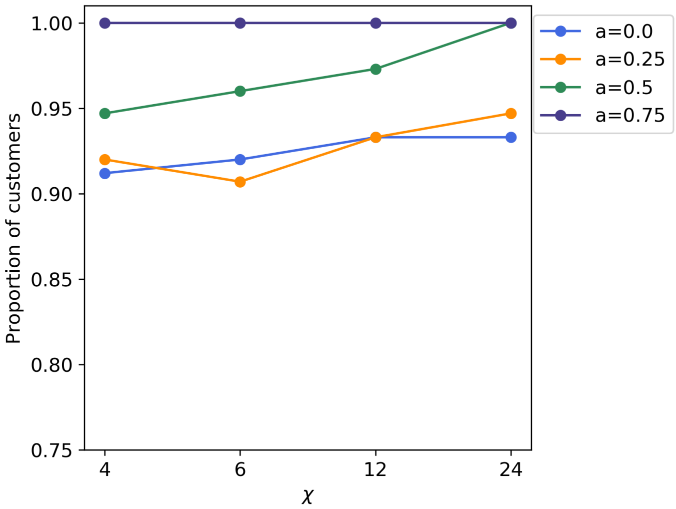


Figure 5.6: The proportion of customer nodes allowed to send RTPs in the network tends to increase as the clearing period and customer attentiveness increases. This trend aligns with patterns in the Nash equilibrium of the bank nodes.

An additional measure of the impact on customer nodes is the success of payment attempts in the network. Recall, issues with customer compatibility and deposit holdings may lead to early termination of a payment attempt. I analyze the proportion of payment attempts that do not experience early termination, deemed successful, for all game configurations. For comparison, I also gather the results for the all and none cases as well. I calculate the total expected number of payment attempts from the T and η game parameters. Figure 5.7 shows the results for all game configurations following the Nash equilibrium and the all case.

The results show that payment attempts in the network experience high success rates regardless of whether banks play by the Nash equilibrium or allow all customers use of RTPs. In the case of Nash equilibrium, the success of payments tends to slightly increase as the clearing period

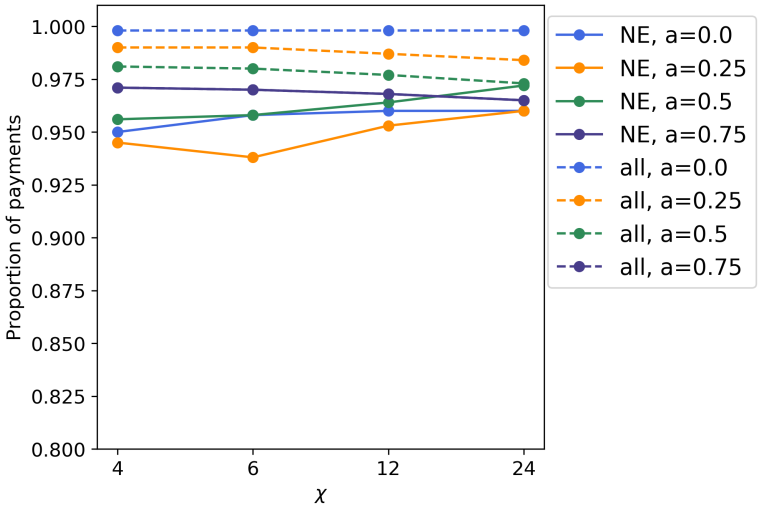


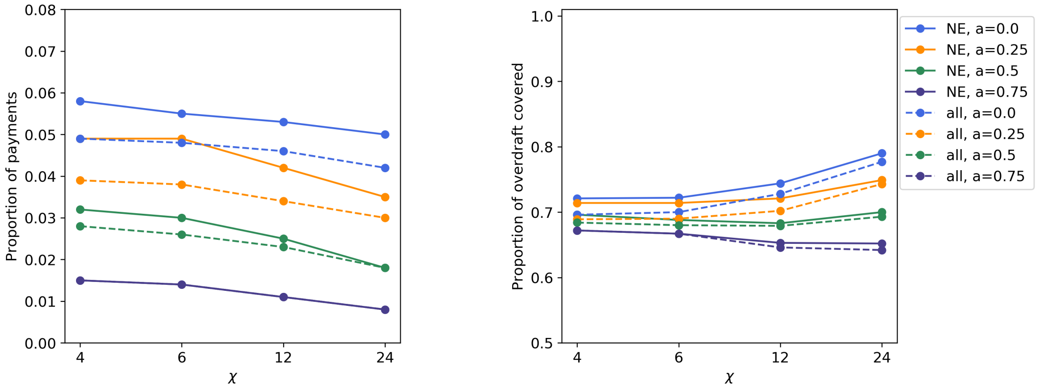
Figure 5.7: Customer nodes experience high success with payment attempts in both the Nash equilibrium and all cases. However, for all game configurations, success is greatest when all customers can send RTPs.

increases. At lower clearing periods, bank nodes tend to allow fewer customer nodes access to RTPs which may lead to cases of sender and receiver incompatibility. Such cases would increase the number of unsuccessful payments. However, more attentive customers allows banks to set lower thresholds which increase RTP use. As a result, compatibility issues would *decrease* leading to more successful payments.

On the other hand, in the all case, increases in attentiveness or clearing period length slightly decrease the success of payments. This can be attributed to an increased likelihood of insufficient funds for making payments. As the clearing period increases, the number of payments left unprocessed between clearing periods increases. As a result, a customer node is more likely to have insufficient available funds for a new payment. When this customer is also very attentive, it is more likely to catch the insufficiency and cancel the attempt due to a lack of funds.

In comparing the success rates for the Nash equilibrium and all cases, I find that allowing all customers to send RTPs leads to more successful payments in almost all configurations. The only exception is when the Nash equilibrium is the all case. This aligns with the analysis of the trends in success rates above. Under the equilibria, success rates are affected by customer incompatibility and deposit value issues. When all customers are allowed to send RTPs, incompatibility is no longer an issue and the only risk to success comes from rare deposit issues. I also find the average proportion of successful payments across all configurations in the none case to be 0.32, well below the success in the other two cases. This highlights the importance of allowing RTPs to promote customer liquidity. The best outcomes occur when all customers are allowed RTPs, however, even allowing access to only some participants appears to promote increased participation in the banking system.

Bank nodes select their strategies to balance customer experience with their exposure to credit risk from overdrafts. I evaluate the impact of the Nash equilibria on bank nodes by measuring the portion of their RTPs that are overdrafts and, when a overdraft occurs, the portion of the payment covered by the bank node. Figure 5.8 shows these outcomes when bank nodes play according to the Nash equilibria and in the all case for each game configuration. In the none case, overdrafts in the model are caught and terminated before processing and their prevalence and required coverage are 0.



(a) Proportion of RTPs that are overdrafts

(b) Proportion of overdraft coverage

Figure 5.8: Bank nodes tend to see few overdraft payments, but are often required to cover the majority of such payments. Allowing all customers the use of RTPs tends to result in slightly fewer overdrafts and requires less coverage.

The results show several interesting trends in overdraft payments mirrored in both the Nash equilibrium and all cases. First, few RTP attempts result in overdrafts, however, when they occur bank nodes tend to cover the majority of the payment. This trend is evident regardless of whether bank nodes adopt the equilibrium strategies or allow all customer nodes to send RTPs. Second, as customer nodes become increasingly attentive, both the number of overdrafts and required coverage tend to decrease. A more attentive customer is more likely to check and correct a payment value, which decreases the possibility of an overdraft occurring.

Notably, as the clear period length increases, the two overdraft measures trend in different directions. In particular, the number of overdrafts tends to show a slight decrease, while the required coverage shows a slight increase. To explain this phenomenon, consider customer nodes as both payment senders and receivers. A more frequent clearing period length provides fewer opportunities for a given customer node to receive funds before the next clearing period. Now, when the next clearing period occurs, this customer has accumulated fewer deposits to use when settling payments for which it is a sender than they might have if the clearing period were longer. Thus, there is an increase in the number of instances in which an RTP's value is strictly larger than the sender's deposit holdings at the time of settlement. On the other hand, a more frequent clearing period also limits the number of payments a customer node may send before the next clearing period. With fewer opportunities to make payments between clearing periods, the *total* amount owed in payments of a given customer node at the next clearing period will be smaller with a clearing period length of 4 than 24. Thus, while there may be an increase in the number of payments for which a customer node does not have enough deposits, the amount of deposits a customer node does have tends to be closer to the total value of payments made.

The case with customer attentiveness of 0.75 diverges from this trend. In this instance, both the number of overdrafts *and* the required coverage slightly decrease as the clearing period length increases. Similar to the logic described above, a longer clearing period length provides more opportunities for a customer to receive payments between clearing periods. When the attentiveness is high, customer nodes are more likely to check and correct a potential overdraft payment. By definition, a correction will decrease the value of the payment relative to its original value. As a result, it is more likely the sender has the required funds during settlement.

Finally, I also note the number of overdrafts and the overdraft coverage required in the all case is slightly less than, or in some cases equal to, the results under the Nash equilibrium. However, since bank nodes still choose to implement some limits on RTP usage in equilibrium, it appears these slight differences are enough to affect the bank nodes' decision.

In comparing these various network outcomes in the Nash equilibrium and all cases, I uncover an important friction between strategic choice and optimal network outcomes. From the analysis, it can be seen that all parties involved tend to be better off when all customer nodes are able to send their payments in real time. This is evident across all measured outcomes. Customer nodes are always more successful attempting payments in the network when all sending RTPs. Bank nodes tend to see fewer overdrafts, and when they do, cover a slightly smaller amount of the payment. However, an individual bank node playing the payments game is often more likely to play a strategy that limits the number of customers with access to RTPs. I conclude that, while bank nodes may be better off when all customers in the network are allowed to send RTPs, a strategic bank node would prefer not to assume the individual risk required to realize this outcome.

5.7 Summary

In this chapter, I analyze the effect of credit risk on the adoption of real-time payments with deferred settlement by banks. I introduce a payment model that supports customer nodes sending both standard and real-time payments with deferred settlement through the interbank network. Credit risk experienced by bank nodes is modeled as overdraft payments, which occur when the value of an RTP is greater than the deposit holdings of the sender during settlement and result in the inability for proper checks to occur before payment processing of an RTP begins. When overdrafts occur, bank nodes extend temporary credit to customers to continue processing the payment. In this way, bank nodes become liable for the difference between the customer's deposits and the payment's value. Such a scenario captures both the risks of the expediency of RTPs and the risk borne by banks in the deferred settlement case. I ask which customers banks should allow to send RTPs in this scenario by capturing the decision in a strategic game played by bank nodes.

Bank nodes select a strategy setting a minimum threshold of initial deposits a customer must have to qualify for RTP use. The strategic decision for bank nodes requires balancing the benefits of attracting customers with RTPs and a desire to limit their resulting liability.

The results show that while bank nodes never choose strategies with high thresholds, the likelihood of allowing all customers to send RTPs is dependent on game configuration parameters. When customer nodes are less likely to have overdrafts, bank nodes are willing to allow all, or nearly all, customers the use of RTPs. However, if customer nodes may be at a higher risk of overdrafts, bank nodes become more likely to select a strategy imposing a non-zero threshold. Bank nodes also tend to prefer to set thresholds when the clearing period is lower and standard payments become similar to RTPs. In this case, it is less worthwhile for bank nodes to risk allowing as many customers the use of RTPs.

I also analyze the effects of bank nodes employing the Nash equilibria to various outcomes in the network and compare them to cases in which banks allow everyone or allow no one RTPs. The results of this analysis show that both bank and customer nodes are better off when all customers are allowed to send RTPs. This is evident in the success of payment attempts in the network, the number of overdrafts, and the coverage required by overdraft payments. However, the Nash equilibrium in many situations for banks in the game is to select a strategy that may place a limitation on RTPs. From these results, I infer that although outcomes for banks are more favorable when all customers send RTPs, strategic banks are unwilling to assume the necessary liability risk to provide all their own customers with RTPs.

While this work offers some insights into the strategic decision faced by banks in the real-time payments space and the effects on the broader network, it is important to note this model is rather simplistic. Indications from the current work imply that regulation or mandate may be necessary to force bank nodes to reach the optimal outcome for this scenario. However, modeling additional externalities in future work may explore how other factors affect the RTP decision. For example, customers may be endowed with additional features that would impact their payments, the network may include more than just customers and banks, and issues with payments could stem from other scenarios such as fraud. I begin to address the question of fraud in the next chapter and continuing to address these and other potential additions should help paint a more complete picture of the real-time payments landscape.

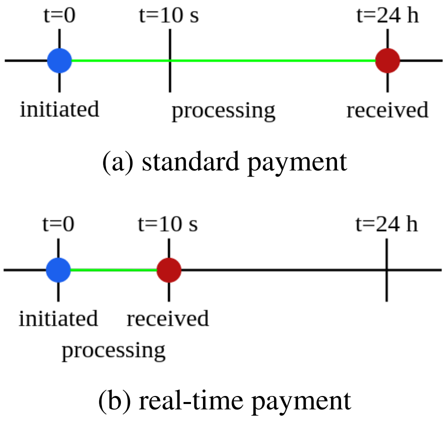
CHAPTER 6

Mitigating Fraud Risk in Real-Time Payment Systems

6.1 Introduction

In the previous chapter, the risk to banks for allowing customers to engage in RTPs was attributed to unintentional consequences of the speed of these payments. However, risks extend beyond such innocent errors. Explicitly malicious behavior such as the fraudulent activity in Chapter 4 remains a threat in real-time payment systems. Drawing on aspects of the previous two chapters, I turn my attention to analyzing strategic behavior related to fraud in real-time payments. I study the case of RTPs with real-time settlement, in which the clearing and settlement steps also occur almost immediately [Committee on Payments and Market Infrastructures, 2016] as shown in Figure 6.1. The payment steps occur in the same sequence as a standard payment, however over a much shorter period of time. The total processing time from initiation to final receipt of funds for these RTPs is about 10 seconds [Diadiushkin et al., 2019, European Central Bank, 2023].

Fraudsters target these payments with the intention of exploiting the limited ability of fraud detection systems to handle the speed necessary for real-time settlement. A survey of payment service providers finds current manual review processes for fraud detection average 5 to 10 minutes [Diadiushkin et al., 2019], too slow for RTPs. Thus, fraud detection for RTPs necessitates reliance on fully automated quick reviews, which tend to be less accurate. A study of the Faster Payment Service in the United Kingdom (UK) found a 132% increase in fraud the year FPS was introduced,



(a) standard payment

(b) real-time payment

Figure 6.1: A comparison of the clearing and settlement processing steps for a standard and real-time payment with real-time settlement.

though numbers for FPS alone were not available. RTPs also attract ***authorized push payments*** (APP) fraud, wherein a customer is tricked into authorizing a fraudulent payment. APP fraud is difficult to detect because of the direct involvement of the lawful account holder and was the second largest type of payments fraud in the UK in 2018 [Taylor and Galica, 2020].

Given the demonstrated risk of fraud in RTPs, I seek to understand how banks may strategically mitigate their fraud risk while providing this beneficial payments service to customers. I study this question in a variant of the extended FCN model capturing interactions between customers, banks, and fraudsters. The model distinguishes between real-time payments with real-time settlement and standard payments when updating the edges in the network. Within this model, I define an RTP fraud game played by bank and fraudster nodes. To mitigate risks—-including fraud risk—of RTPs, maximum threshold values above which a payment cannot be sent in real time have been found to be useful tools for banks [Weyman, 2016]. Thus, bank nodes in the game make strategic choices regarding threshold value setting and investment in RTP fraud detection. Fraudsters in the game make strategic decisions determining targets of their fraudulent payment attempts.

The game is analyzed using EGTA to identify the Nash equilibria for various game configurations. To evaluate the benefit provided by the inclusion of each risk mitigation technique, I apply the strategic feature gains assessment as described in Section 2.3. Finally, I study outcomes in the network under strategic equilibrium to understand the impact on network participants.

I find that as bank liability for RTP fraud increases, banks adopt both threshold values and fraud detection with higher probability. When banks utilize both mitigation techniques, fraudsters switch from attempting fraud only with RTPs to considering any payment type. In selecting banks to target, fraudsters tend to heavily rely on historical success as a basis for decision making. The strategic feature gains assessment highlights the importance of controlling the bank threshold value for mitigating initial fraud risk. By analyzing equilibrium outcomes, I find that bank strategies are effective at disrupting fraudsters with minimal negative impact to customers.

6.2 Related Work

As this chapter is an extension of the previous two, the related works introduced in those chapters remain relevant here. Prior literature in real-time payments is discussed in Section 5.2 and work related to fraud detection can be found in Section 4.2.

Diadiushkin et al. [2019] focus specifically on applying fraud detection in the real-time payments space, noting the challenges required by the short processing time. Additional works introduce methods for faster fraud detection procedures applicable to the general payments space [John et al., 2016, Said and Hajami, 2021, Madhuri et al., 2023]. While not all of these works are motivated by RTPs, the benefits of exploring faster fraud detection methods have clear benefits in the RTP space. There also exist several works discussing various concerns in RTP systems which may affect fraudulent activity. These concerns include difficulties with performing maintenance and repair due to their availability [Weyman, 2016] and with verifying customer identity [Dugauquier et al., 2023].

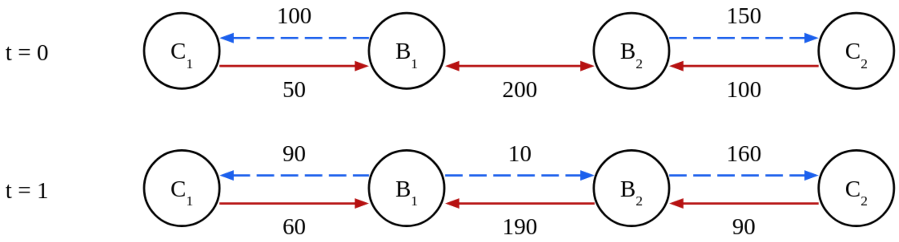


Figure 6.2: Customer node C_1 sends an RTP of 10 units to C_2 in time step $t = 1$, leading to an immediate update to the values on the edges of all payment participants.

6.3 RTP Fraud Model

6.3.1 Bank and Customer Nodes

The RTP fraud model is an extended FCN with a set of bank nodes, $B = \{B_1, \ldots, B_b\}$, and a set of customer nodes, $C = \{C_1, \ldots, C_n\}$. Each customer node owns deposits held in an account at a bank node and draws on them to make payments to others in the network as described in Section 2.1.2. Bank nodes are responsible for routing these payments, which may be standard or real-time payments, on behalf of customers through the interbank network. I assume in this model banks have an infinite willingness to route such payments for customers.

6.3.1.1 Real-Time Payments

As in Chapter 5, I will differentiate between payment types by the timing of the edge updates representing the payment. In a real-time settlement setting, all payment steps from initiation to final receipt of funds are executed immediately. I model this by updating all edges between payment participants in the network during the same time step in which the payment is initiated as shown Figure 6.2. Time step $t = 0$ reflects the network before the payment is made. At time step $t = 1$, customer node C_1 makes a payment of 10 units to C_2 which is immediately reflected in changes to all edges between the two nodes.

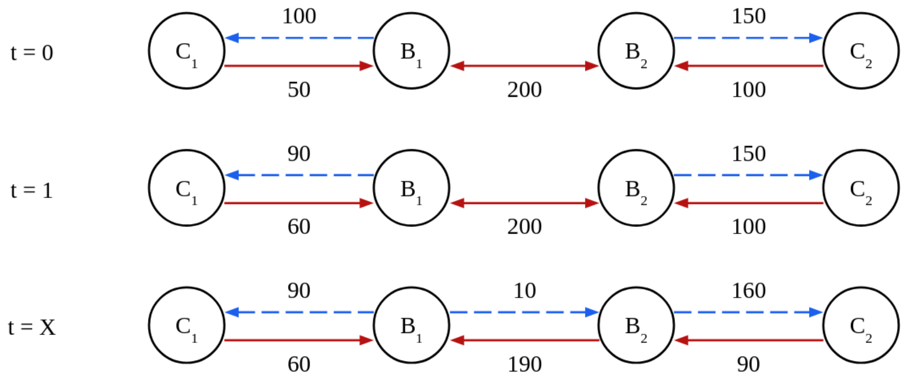


Figure 6.3: An example of customer node C_1 making a standard payment of 10 units to C_2. A standard payment updates only the edges between the sender and its bank initially, with the remaining edges updating during the next clearing period.

6.3.1.2 Standard Payments

Standard payments, as mentioned in the previous chapter, are subject to batch processing which leads to delays. To model batch processing, each bank node will have a payments queue for storing unprocessed payments. When a standard payment in initiated, only the credit and debt edges between the sender and its bank are updated before the payment is placed in the sender bank's payments queue. Note, this diverges from the standard payment modeling in the previous chapter. At a later time period, the clearing period, the bank queues are cleared and the remaining edge updates reflecting the payment are applied to the network. The same payment of 10 units from C_1 to C_2 is shown in Figure 6.3, this time as a standard payment. The edges between C_1 and B_1 are updated at time step $t = 1$ to represent the beginning of payment processing. However, it is not until the clearing period $t = X$ that the edges between the two bank nodes and the receiver and its bank are updated. After the clearing period, both the standard and real-time payment have the same effect on the network.

6.3.2 Fraudster Nodes and Fraudulent Payments

As in Chapter 4, fraud is committed in the network by fraudster nodes. Importantly, fraud may occur with any payment type. A fraudster draws on the deposits of a customer node, the victim, to make payments in the network. When this occurs, a fraud edge is created connecting the fraudster and its victim as described in Figure 2.10. Other nodes in the network are generally unaware of the payment's fraudulent nature, leaving the remaining payment steps as described for the non-fraudulent cases above. In particular, a fraudulent RTP will immediately create the fraud edge and update all edges between victim and receiver in the same time step the fraudster initiates the payment. On the other hand, when a fraudulent standard payment is initiated, only the fraud edge, and edges between the victim and its bank are updated immediately. The remaining edge updates representing the payment are reflected in the network after the next clearing period.

6.4 RTP Fraud Game Initialization

In this section, I describe the initialization of all nodes, the strategy sets available to strategic nodes, and the formation of the initial payments model. As the game unfolds, the initial network will dynamically update as a result of behavior described in Section 6.5.

6.4.1 Customer Nodes

The n customer nodes are initialized with deposits drawn from an exponential distribution with an average value of 10,000. Each customer node is also initialized with its own preference for RTP usage as both a sender and receiver. A given customer node will only be willing to receive RTPs (*RTP-only*) or be willing to accept any type of payment (*Any*). This is initialized as a random draw with probabilities λ and $1-\lambda$ respectively. Furthermore, some customers may also have an aversion to utilizing a new, unfamiliar technology to send payments. However, I assume that customers are aware others may demand payments in real time and thus are always willing to engage to some degree with RTPs. To capture this, each customer node has its own personal threshold for payment values above which they are unwilling to send in real time. Customer thresholds are uniformly randomly chosen during initialization from $CT \in \{200, 400, 600, 800, 1000\}$. The maximum payment value in the model is 1,000, so a customer with threshold $CT = 1,000$ has no restrictions on willingness to send RTPs.

Lastly, I note that customers may not necessarily always desire to send an eligible payment in real time due to personal preference and payment requirements [Greene et al., 2014, Hartmann et al., 2019]. I capture this with the ***urgency*** parameter (u), such that with probability u an eligible payment is sent in real time and with probability $1-u$ it is sent as a standard payment. The urgency parameter universally applies to all customer nodes.

6.4.2 Bank Nodes

A set of b bank nodes is initialized to form the interbank network and with their own payments queues. Each bank node has access to two fraud detectors, one for each type of payment. Since the focus of this work is to understand the use, rather than emphasize a particular method of fraud detection, I abstract away implementation details by modeling both detectors as black boxes as in Chapter 4. A given detector is characterized only by the probability γ it correctly labels a payment as fraudulent or not fraudulent relative to its true label. In this instance, I again consider the detection abilities for fraudulent and non-fraudulent payments equal and simply refer to one γ value for a given payment type. Bank node B_i's standard payments fraud detector $\gamma^S_{B_i}$ is initialized with a random draw from $U \sim [0.8, 0.9]$ and its RTP detector $\gamma^R_{B_i}$ by the bank node's chosen strategy as described below.

6.4.2.1 Bank Strategy Set

During initialization, each bank node selects a strategy from a set of two-part strategies controlling RTP usage and fraud detection. The first part of the strategy establishes the bank's threshold $BT_{B_i} \in \{0, 200, 400, 600, 800, 1000\}$, which is the maximum payment value bank node B_i will allow a customer to send in real time. A threshold of $BT = 0$ is equivalent to prohibiting any RTPs, a threshold of $BT = 1000$ allows unrestricted RTPs, and the remaining choices offer varying degrees of limitation. The threshold choice of bank nodes is assumed to be public information that might attract customers to the bank.

The second part of the strategy is the bank's chosen investment level in fraud detection for RTPs (***FDI***). The investment level determines the bank node's RTP fraud detection capabilities for the duration of the game and incurs a one-time cost. A higher investment allows for a more accurate fraud detector (higher γ^R), but at a higher cost to the bank. To capture the challenges of speed, fraud detection accuracy for RTPs at every investment level is strictly less than that of standard payments. Table 6.1 maps the investment level choices to one-time cost and γ^R setting, which is derived relative to the bank's standard payment fraud detection capabilities. The strategy set for bank nodes contains all 24 combinations of threshold choices and investment levels.

FDI level	Cost (units)	$\gamma^R =$
none	0	–
low	1,000	$0.65 \times \gamma^S$
high	3,000	$0.9 \times \gamma^S$

Table 6.1: A mapping of RTP fraud detection investment level to one-time incurred cost and detection capability (γ^R) relative to the bank's standard payment fraud detector (γ^S). The detection capability of a bank remains set for the duration of the game.

6.4.3 Fraudster Nodes and Strategy Set

The network is initialized with m fraudster nodes. Each fraudster node maintains a history of its success committing fraud at each bank node, which is used to inform its strategic behavior. Before the game begins, fraudsters select their strategies. Each strategy determines the rules the fraudster will follow for selecting a bank and payment type to target for fraudulent activity. I assume given the opportunity, fraudster nodes will always prefer to attempt RTPs to exploit the limits of fraud detection in the quicker payment scheme. However, it may be beneficial for fraudsters to consider attempting standard payments as well. If banks severely restrict RTPs, fraudsters may wish to gamble on the imperfect nature of standard payment fraud detection rather than completely miss an opportunity to attempt fraud. Thus, fraudsters in the game will choose between one of two rules: attempt RTPs only (RTP-only) or be willing to attempt a standard payment if an RTP is not possible (Any).

Fraudsters in the game target bank nodes at which to commit fraud, but do not differentiate between customer nodes of a targeted bank. I assume information regarding customer and bank relationships is accessible to fraudsters at negligible cost. A fraudster node selects the bank node to target using their strategy, and randomly selects a victim from customers with accounts at the target bank. Whenever a fraudster's payment attempt is caught, it will switch to a new victim node by following the same procedure of first selecting the bank target and then a random customer of that bank. Note depending on the rule, the target bank may change over the course of the game.

There are two types of targeting rules fraudsters may choose from: threshold and success. The threshold rule simply targets the bank with the highest bank threshold, $\mathrm{argmax}_{B_i} BT_{B_i}$. The success rule selects the bank at which the fraudster has been most successful in the past, modeled as a *multi-armed bandit problem* and implemented using the *Upper Confidence Bound* algorithm [Auer et al., 2002].

Specifically, the target bank b^t is calculated by:

$$\underset{B_i}{\mathrm{argmax}} \frac{\sum_{s=1}^{t} I_{b^s=B_i} X_{B_i}(s)}{N_{B_i}(t)} + \rho \times \sqrt{\frac{log(t)}{N_{B_i}(t)}},$$

where t is the current time step and $N_{B_i}(t)$ is the number of times bank B_i has been targeted prior to t. I set X_{B_i}, the reward gained by targeting B_i, to 1 if the payment attempt was successful and 0 otherwise. A larger value of ρ indicates greater exploration of bank nodes than exploitation. I test three variants of this strategy defined by $\rho \in \{0.2, 1, 2\}$, which are referred to as $\mathsf{success}_\rho$. Following both rules may result in more than one bank node fitting the target criteria. In this case, one such bank is uniformly randomly chosen as the target each time.

I also include the random targeting rule, which randomizes over threshold and success rules each time a new bank target is selected. I test two variants of this rule defined by the ρ value used for the success rule. In particular, I test $\rho \in \{0.2, 2\}$ and refer to them as random_ρ.

Overall, there are two possible payment targeting rules and six possible bank targeting rules. A single strategy is a combination of one payment and one bank targeting rule creating 12 possible strategies available to fraudster nodes in the game.

6.4.4 Network Formation

The initial network is formed by assigning each customer node to a bank node on the assumption that customers prefer banks that allow them to send RTPs. If possible, a given customer node C_i is assigned to the bank node that allows the customer to send RTPs at or above C_i's desired level. That is, a bank B_j such that $CT_{C_i} \leq BT_{B_j}$. If no such bank exists, the customer is assigned to the bank node offering the highest possible threshold ($\text{argmax}_{B_j} BT_{B_j}$). In the event multiple banks fit the criteria, one such bank will be randomly selected. Upon assignment to a bank, the customer deposits all its initial funds in an account creating a debt edge from the bank node to the customer as outlined in Section 6.3. For simplicity, I assume customer nodes in the game have an infinite willingness to hold additional deposits in their account.

6.5 RTP Fraud Game

The fraud game unfolds over T time steps. Each time step t follows the same procedure:

1. Check if t is a clearing period and if so, clear all bank queues and use the unprocessed payments to update the network as described in Section 6.3.1.
2. Generate η new payments in the network as described below in Section 6.5.1.
3. All payments are sent through the respective sender bank's fraud detector and update the network accordingly.

After the last time step, all payment queues are cleared for the final time, and bank and fraudster nodes receive a payoff for the performance of their selected strategies.

6.5.1 Payment Creation

During the payment creation phase, η payment senders are randomly selected from the network without replacement. With probability μ, one of these η senders is a fraudster. The value of a given payment is drawn from $U \sim \{1, \ldots, \min(1000, d)\}$, where d is the amount of deposits currently held in the sender's account. A payment receiver is obtained by randomly selecting from the other customer nodes in the network. Whether the payment is sent in real time or not is determined by the sender, its bank, and the value of the payment as described below.

6.5.1.1 Customer Node Payment Type

Consider a payment of value v sent by customer node C_i with an account at bank B_j. This payment is only eligible to be sent in real time if $v \leq CT_{C_i}$ *and* $v \leq BT_{B_j}$. That is, if the sender is willing and able, respectively, to send it in real time. If the receiver only accepts RTPs, the payment must be sent in real time. Payment attempts ineligible for real time transmission are terminated in this case due to incompatibility and considered a missed opportunity for a transaction within the network. If the receiver is wiling to accept any type of payment and the payment is ineligible for real time, it is sent as a standard payment. Otherwise, a payment eligible to be sent in real time will be sent as an RTP subject to the sender's urgency as described in Section 6.4.1. This determination process is summarized in Figure 6.4.

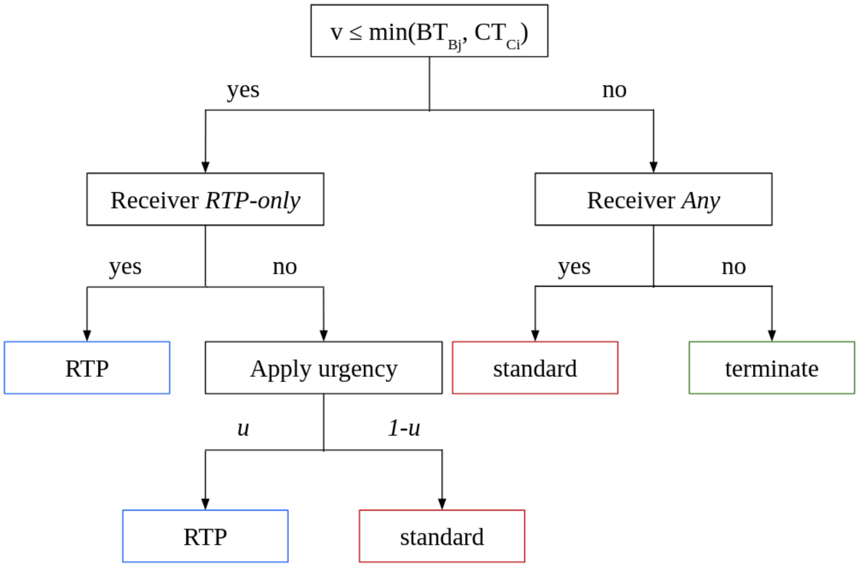


Figure 6.4: A flow chart determining the payment type for a payment sent by customer node C_i belonging to bank B_j of value v. A customer's payment type is determined by the value of the payment, the sender's bank, the receiver, and customer urgency.

6.5.1.2 Fraudster Node Payment Type

Unlike customer nodes, fraudsters are only constrained by bank thresholds and always prefer RTPs when possible. A fraudster making a payment of v units using a victim with an account at B_j will attempt an RTP if $v \leq BT_{B_j}$. When the payment's value exceeds B_j's threshold, the receiver's preference and fraudster's strategy determine the course of action. If the receiver is willing to accept any payment type and the fraudster is following the Any rule, a standard payment is attempted. Otherwise, the payment attempt is terminated, at which point the fraudster selects a new victim following the outlined procedure. Figure 6.5 summarizes the determination process.

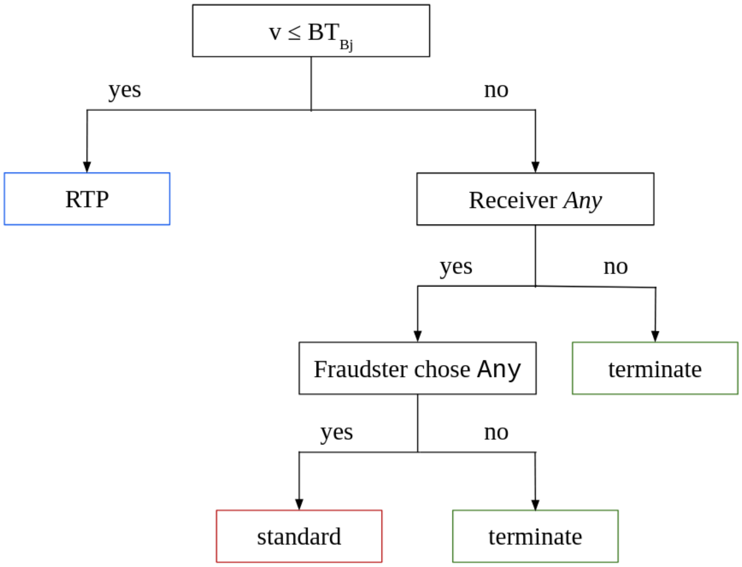


Figure 6.5: A flow chart determining the payment type for a fraudster's payment of value v using a victim belonging to bank B_j.

6.5.2 Fraud Detection and Network Updates

All payment attempts not terminated during the creation phase undergo fraud detection. Bank node B_i will invoke its black box fraud detector for the corresponding payment type as described in Section 6.4.2. A standard payment will be correctly labeled with respect to its true label with probability $\gamma^S_{B_i}$ and an RTP with probability $\gamma^R_{B_i}$. Any payment labeled fraudulent is canceled regardless of its true label. The remaining payments update the network as outlined in Section 6.3.1.

6.5.3 Payoffs

A bank's payoff is composed of the effects of RTP services offered and the effects of invoking fraud detection. By offering RTPs, a bank attracts some total amount of initial customer deposits, ***ID***. Though the deposits themselves are a bank liability, they represent partial assets in the form of continued business from customers, which is modeled as a fraction of the deposit value. Bank nodes route some total value of payments in the network on behalf of their customers, denoted ***VA***, on which banks may be able to charge a small fee δ_2. When fraud occurs, bank nodes are liable for the full amount for standard payments (***VSF***) and may hold some liability for RTPs (***VRF***) as described in Section 6.6.1. Choosing to invest in fraud detection results in a one-time cost to banks (Table 6.1) denoted by ***RFI***. Lastly, imperfect fraud detectors may lead to erroneous labeling of some customer payments and their subsequent cancellation. So called false positives represent missed opportunities for routing and may lead to frustration and less frequent business by customers. To account for this potential cost, I define ***VFP*** as the total value of false positive payments and use β to account for these potential costs.

Thus, bank node B_i receives the following payoff:

$$\delta_1 ID_{B_i} + \delta_2 VA_{B_i} - \alpha VRF_{B_i} - VSF_{B_i} - RFI_{B_i} - \beta VFP_{B_i}.$$

A fraudster node, on the other hand, seeks only to maximize their success committing fraud. This is reflected in their payoff, which is simply the total value of fraudulent payments the fraudster commits across all banks and payment types during the game. For a given fraudster node, F_i, this value is calculated as:

$$\sum_{j=1}^{n} \sum_{(F_i, C_j, fraud, v) \in E} v$$

6.6 Experiments Overview

6.6.1 Identifying Nash Equilibria

I employ EGTA to identify Nash equilibrium strategies for bank and fraudster nodes in games with $n = 200$ customers, $b = 4$ banks, and $m = 1$ fraudster. The game is played over $T = 2,880$ time steps, with clearing periods occurring every $\chi = 96$ time steps. In each time step, $\eta = 20$ payments are generated with probability $\mu = 0.7$ one of the payments is made by a fraudster node. A customer will send an eligible payment in real time with urgency $u = 0.5$. When calculating bank payoffs, the value of initially attracted deposits is scaled by $\delta_1 = 0.5$, the fee for routing payments is set to $\delta_2 = 0.01$ [Deloitte, 2019], and the cost of false positives is $\beta = 0.2$. To understand the effect of network attributes on the Nash equilibrium, I perform the analysis on nine different configurations of the RTP fraud game defined by customer demand for RTPs and bank liability for RTP fraud. Customer demand for RTPs is controlled by the number of customers willing to receive only RTPs, which is determined by the probability a given customer is initialized as such. I test three different probabilities, $\lambda \in \{0.25, 0.5, 0.75\}$. I run experiments for the cases where banks have no liability, partial liability, and full liability for RTP fraud captured by $\alpha = \{0, 0.5, 1\}$ respectively. These parameter settings are summarized in Table 6.2. Each strategy profile selected for simulation is simulated in 3,000 random instances of the RTP fraud game.

6.6.2 Strategic Feature Gains Assessment

To better assess the benefit to bank nodes of each fraud risk mitigation technique, I employ the strategic feature gains assessment introduced in Section 2.3. In particular, I seek to understand the individual and relative benefit of controlling RTP fraud detection investment and payment thresholds for banks.

I perform the four assessments listed in Table 6.3 on the nine configurations of the RTP fraud game described in Section 6.6.1. Assessments A1 and A2 measure the gains a bank node experiences from accessing both mitigation techniques after initially only controlling one: either

Parameter	Setting
n	200
λ	{0.25, 0.5, 0.75}
u	0.5
b	4
m	1
T	2,880
η	20
χ	96
μ	0.7
α	{0, 0.5, 1}
δ_1	0.5
δ_2	0.01
β	0.2

Table 6.2: A list of the parameter settings used for an EGTA analysis of the RTP fraud game.

threshold setting or investment in fraud detection. The base set of A1, ***BT only***, contains strategies controlling the RTP threshold, but that always set the fraud detection investment level to *none*. The deviation set contains the remaining strategies in the bank strategy set. For assessment A2, the strategies in the base set, ***FDI only***, control the level of fraud detection investment, but set the threshold to $BT = 1,000$. Again, the deviation set contains the remaining bank node strategies. Note, I consider control over a technique to include the choice to not invoke it. Thus, controlling the threshold includes the option to set $BT = 1,000$ and control of fraud detection investment includes the option to choose *none*.

The last two assessments, A3 and A4, address the scenario where banks begin without control over any mitigation measures to see which single technique they will employ first, when forced to choose only one. Both assessments contain only one strategy in their base set, neither of which allows investment in fraud detection. However, A3's strategy sets $BT = 0$ to represent the no RTPs case and A4 sets $BT = 1,000$ to represent the unrestricted RTPs case. Both deviation sets contain the strategies which control *only* one of the mitigation techniques. That is, if a strategy employs a threshold, the investment level must be *none* and if it invests in RTP fraud detection it must set $BT = 1,000$. The analysis of this assessment will emphasize the technique employed by the strategy offering the greatest gains. A full list of the strategies included in each base and deviation set can be found in Table 6.6. Note, while some of the assessments share the same description of the deviation set, the membership will vary slightly. This is because the exact membership of the deviation set is relative to the base set, as the assessment requires the two sets be disjoint.

Assessment name	Base set	Deviation set
A1	BT only	BT *and* FDI
A2	FDI only	BT *and* FDI
A3	no RTPs	FDI *or* BT
A4	BT = 1000, FDI = none	FDI *or* BT

Table 6.3: A list of each assessment performed, described by the strategies in the base and deviation sets. *BT* denotes threshold setting and *FDI* denotes fraud detection investment setting.

6.6.3 Effects on the Network

Finally, I analyze the effects of bank and fraudster nodes following the equilibrium strategies on the network. I measure various outcomes for customer and fraudster nodes over 1,000 runs of each game configuration. In the event of a mixed-strategy equilibrium, nodes are assigned to play a pure strategy from a weighted draw according to the equilibrium distribution. I compare the outcomes under Nash equilibrium to two additional settings: when banks do not offer RTPs and when bank nodes do not restrict RTPs, nor invest in fraud detection.

6.7 Experiment Results

6.7.1 Nash equilibria

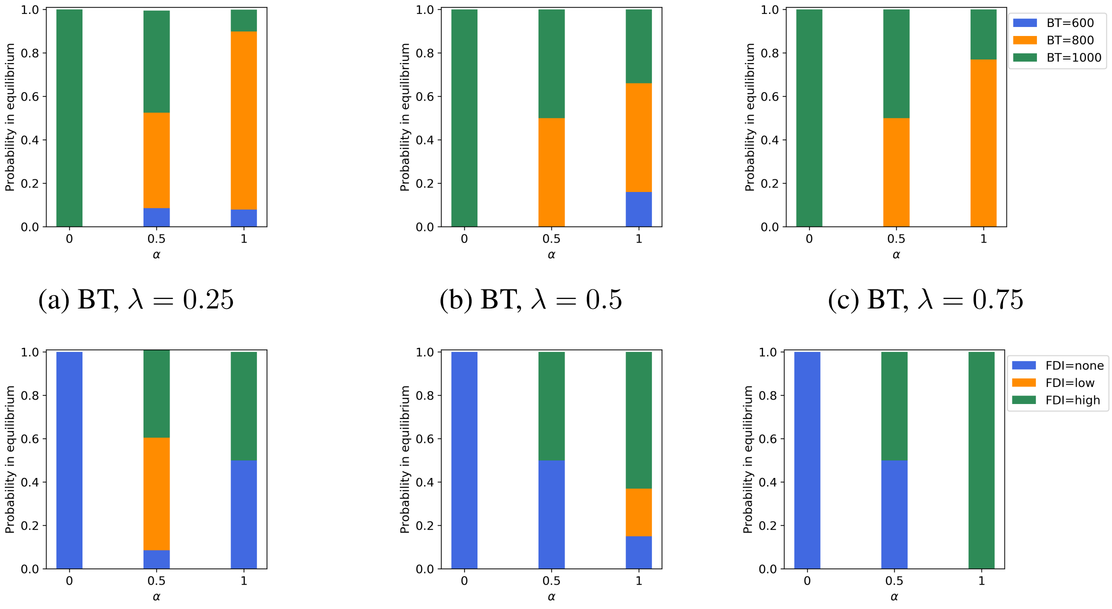
Figure 6.6 shows the Nash equilibria of bank nodes in the RTP fraud game, with each component of the strategy shown separately. The results are presented as the total probability assigned to playing each component of the strategy according to the equilibria for all game configurations.

Note, bank nodes never select a threshold below $BT = 600$, and so for simplicity I omit those lower thresholds from the figure.

I find that bank nodes in the RTP fraud game select strategies with high thresholds, but often balance this with the use of fraud detection when bank are liable for RTP fraud. Specifically, when banks are not liable for RTP fraud, they do not restrict customer access to RTPs, nor do they employ fraud detection. With partial liability, bank nodes begin to set thresholds and invoke fraud detection. Interestingly, analysis of the equilibria shows a strategic trade-off between customer access to RTPs and the cost of fraud detection in this setting. Bank nodes tend to either invoke fraud detection to avoid restricting customer RTP usage *or* set a threshold value instead of incurring the cost for fraud detection. When bank nodes become liable for the full amount of RTP fraud, they exhibit a higher probability of both setting a threshold and investing in good fraud detection. In this case, the trade-off between fraud detection costs and customer RTP restriction is only seen when customer demand for RTPs is low. We find that customer demand has a smaller effect on bank equilibria than RTP fraud liability, with greater customer demand resulting in only slight increases in threshold values selected.

In response to the banks' strategic decisions, fraudsters in the network tend to target banks based on historical success, either with the success or random strategies, with a high degree of exploitation. Fraudsters will prefer to target RTP-only unless bank nodes implement both fraud mitigation measures, invoking fraud detection *and* setting a threshold value. In this case, more barriers to successful RTP fraud force the fraudster to adopt the Any strategy.

The identified equilibria show that bank nodes will begin invoking at least one fraud risk mitigation technique when only liable for half the value of the fraud that occurs. This begs the question of whether bank nodes will tolerate any level of fraud liability before invoking such measures. To answer this, I also conducted the EGTA analysis when bank liability for fraud is non-zero, but low ($\alpha = 0.25$). At this level of liability and even with low customer demand, there does exist an equilibrium in which bank nodes do not restrict customer real-time payment use, nor invoke fraud detection. Thus, I determine that bank nodes are willing to take on a non-zero amount of liability



(a) BT, $\lambda = 0.25$

(b) BT, $\lambda = 0.5$

(c) BT, $\lambda = 0.75$

(d) FDI, $\lambda = 0.25$

(e) FDI, $\lambda = 0.5$

(f) FDI, $\lambda = 0.75$

Figure 6.6: The total probability assigned to each component of a bank node's strategy in equilibrium for all game configurations. *BT* denotes RTP thresholds and *FDI* indicates RTP fraud detection investment.

for fraudulent RTPs before they worry about potential fraud.

6.7.2 Strategic Feature Gains Assessment

While both mitigation techniques often appear in the equilibria of bank nodes, the strategic feature gains assessment highlights the particular importance of control over customer RTP use. The results from A1 and A2, shown in Figure 6.7, demonstrate greater gains to bank nodes when starting by setting a threshold (*BT*) and gaining the ability to invest in RTP fraud detection (*FDI*) compared to the reverse scenario. There are no gains from any strategies in the deviation set when $\alpha = 0$ because the pure strategy Nash equilibrium occurs in the base sets of both A1 and A2. When thresholds are set first, the addition of *FDI* allows banks to deviate to a strategy with a higher threshold value, expanding RTP usage. Conversely, starting with *FDI* results in a later restriction of RTP usage and a lower payoff gain for banks. Though it should be noted bank nodes almost always benefit from the ability to use both mitigation techniques, with the exception of the cases where the equilibrium is to not restrict RTPs nor offer fraud detection.

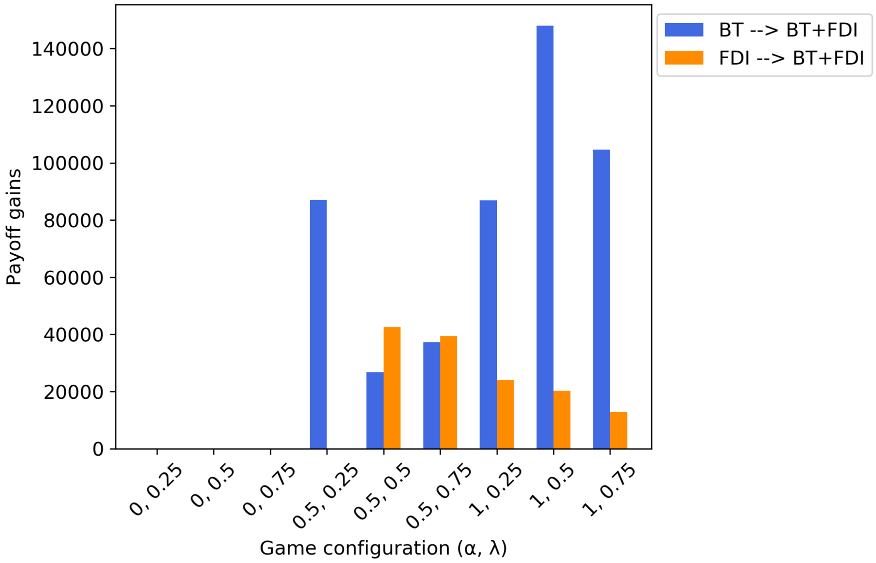


Figure 6.7: The results from A1 and A2 show that a bank node benefits the most by gaining the ability to control fraud detection investment level (*FDI*) *after* initially controlling only the threshold value (*BT*).

The importance of initial threshold setting is further supported by the results of A3 and A4. In both cases I find a bank node in all game configurations will choose to deviate to a strategy controlling the threshold value when given the option to choose only one mitigation technique. When banks initially do not offer RTPs, the best deviation is to allow real-time payments with a threshold of $BT = 200$ if banks hold some liability for fraud and $BT = 1000$ otherwise. The low threshold allows a bank node to enjoy the benefits of offering customers RTPs without assuming too great of a risk from fraud. On the other hand, if banks begin with unrestricted RTPs, the first restriction they would place if they hold liability for fraud is setting the threshold to $BT = 800$. This is a small limitation on customers, so bank nodes are not overly penalized, but helps keep the amount of fraud manageable.

Figure 6.8 shows the payoff gains for assessments A3 and A4. The results show the payoff gains in the no RTPs scenario are much higher than the gains in the unrestricted case. And in fact, are also larger than those gains experienced in assessments A1 and A2. These results indicate that bank nodes in the model benefit greatly from beginning to offer RTPs to their customers, in spite of the potential risks from fraud.

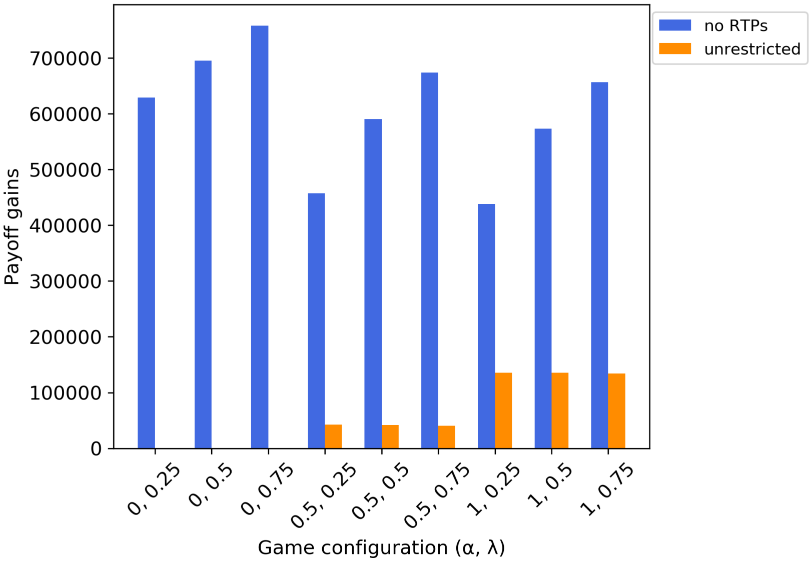


Figure 6.8: The payoff gains for deviating to a strategy in the deviation set for assessments A3 and A4 show the largest gains occur when a bank node begins to offer RTPs, even at a low threshold.

6.7.3 Network Effects

When nodes play according to the Nash equilibria, all bank nodes are targeted by the fraudster over the course of the game and at least 1/3 of customer nodes are victims of the fraudster. The number of victims increases as bank nodes become more concerned with fraud and implement stricter mitigation measures. This leads to less success by the fraudster and forces it to change customers more often to accomplish its goal. In these cases, up to 80% of customer nodes become victims.

Theoretically, the fraud risk mitigation techniques employed by banks could negatively impact customers' ability to successfully make payments in the network. Thresholds could increase customer incompatibility issues, while fraud detection could result in some false positives. To explore these effects in practice, I compare the customer nodes' rate of successful payment attempts under the Nash equilibria to the case when bank nodes do not restrict RTPs, nor use fraud detection. In addition, I report the same measure for the fraudster node. Figure 6.9 shows the results of this comparison, reported as the difference between the success with no RTP restrictions and the outcome under Nash equilibrium for each game configuration. The results show only a small gain in success for customer nodes when bank nodes switch to unrestricted RTPs without fraud detection and a much larger gain for fraudsters. This indicates the strategic choices of banks greatly reduce fraudster success with minimal interruptions to customers.

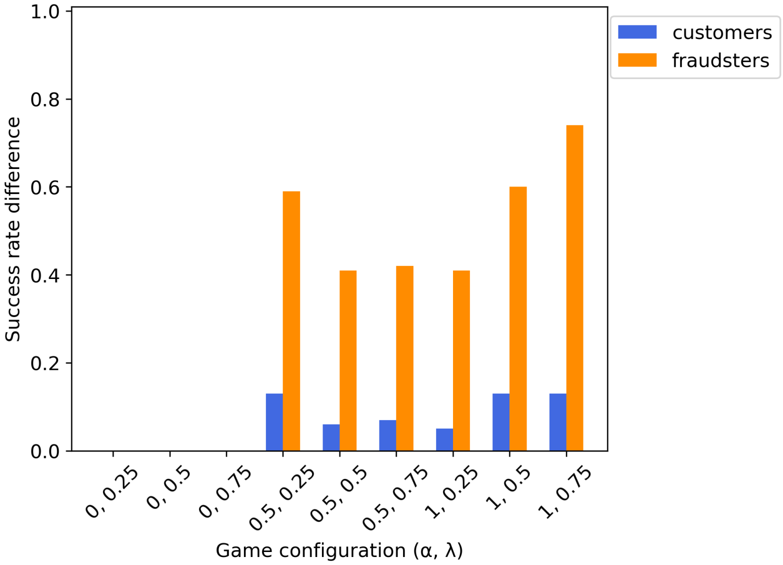


Figure 6.9: The gain in success rate for fraudsters and customers when banks allow unrestricted RTPs compared to under equilibrium. Fraudsters experience a much larger gain, indicating the strategic choice of banks effectively mitigates fraud risk with limited impact on customers.

6.8 Preliminary Analysis of Strategic Customer Nodes

Within the current RTP fraud game, customer nodes exhibit non-strategic behavior and their preferences are selected during initialization by a simple stochastic process. It should be noted, however, that customers may also have strategic preferences with respect to RTP use. In this section, I offer a preliminary analysis of customer nodes' strategic preferences for sending and receiving RTPs in the context of the RTP fraud game. The experiments presented here analyze customer decision making in isolation, but offer some preliminary commentary on customers' strategic choices.

6.8.1 Receiver Preferences

The first study examines customers in their role as payment receivers. Customer nodes in the model may choose to accept RTPs only (*RTP-only*) or be willing to accept any type of payment (*Any*). When making this decision, customer nodes may make trade-offs between flexibility and liquidity. A willingness to accept any payment type allows a customer node to engage with all other customers in the network. On the other hand, the immediacy of RTPs offers customers a high degree of liquidity. To understand how customer nodes make these trade-offs, I introduce a simplified version of the RTP fraud game and analyze it using EGTA to identify the Nash equilibrium.

6.8.1.1 Receiver Experiment Overview

The receiver experiment model contains only bank and customer nodes. Since the RTP fraud game does not model recovery of fraudulent funds, the presence of a fraudster does not affect customer demand as payment receivers. Banks nodes are initialized as members of the interbank network and with payments queues as described in Section 6.4.2. Customers nodes are initialized with some initial deposit holdings following the same distribution reported in Section 6.4.1. Each customer node will be initialized with an unrestricted ability to send RTPs with probability k and will be unable to send any RTPs with probability $1 - k$. During initialization, each customer node will also select its strategy from $\{\textsf{RTP-only}, \textsf{Any}\}$. Customer nodes in the network are also defined by an urgency parameter, which operates in the same manner as in the RTP fraud game. The network is formed by uniformly randomly assigning customers to bank nodes.

The receiver game is played over T time steps, with each time step creating η new payments. For each payment the sender, receiver, and value are randomly selected as in the original RTP fraud game. The payment type is determined with a simple modification to the RTP fraud game procedure. If the receiver demands RTP-only, the sender must sent a real time payment and if unable, the payment attempt is terminated. If the sender cannot send RTPs and the receiver is willing to accept any payment type, the payment is sent as a standard payment. Otherwise, if the sender can send RTPs and the receiver is willing to accept any payment, the customer urgency determines the payment type as described in Section 6.4.1.

After the T time steps, customer nodes receive a payoff for the performance of their strategy. The payoff to customer node C_i is:

$$\delta\, VSR_{C_i} + VRR_{C_i}$$

with $\mathbf{VSR_{C_i}}$ representing the total value of standard payments and $\mathbf{VRR_{C_i}}$ representing the total value of RTPs received by C_i over the course of the game. As mentioned, customer nodes value the immediacy provided by real-time payments and place a premium on receiving RTPs over standard payments. This is modeled with the δ cost attributed to the delay of standard payments.

The parameters of the receiver game follow the settings of the RTP fraud game except for those listed in Table 6.4. In particular, this game is played with only $n = 10$ customer nodes and $b = 2$ bank nodes. Each time step creates only $\eta = 2$ payments and bank payoffs are calculated with $\delta = 0.7$. Customer nodes' strategic decision for receiving payments must factor in the availability of RTPs for payment senders. If no customer can send RTPs, it would not be logical for a customer node to demand payment in only RTPs. Thus, I conduct the receiver experiment for three settings of k, the probability a customer node is assigned unrestricted access to RTPs.

Parameter	Setting
n	10
b	2
m	0
η	2
k	{0.25, 0.5, 0.75}
δ	0.7

Table 6.4: The parameter settings employed for the receiver game. Relevant parameters not listed follow the same settings as the RTP fraud game.

6.8.1.2 Receiver Nash Equilibrium

My experiments find that receivers in this game always choose to accept any payment type. Thus, customer nodes prefer flexibility and are not overly concerned with liquidity. These results indicate that simply considering these two factors alone are not enough to sway a customer node to only conduct business in RTPs. If there is reason for a customer to choose RTPs only, it is likely due to addition reasons unaccounted for in this model.

It is worth noting that in the existing game, a receiver following the Any strategy will receive RTPs subject to the customer urgency parameter. Future experiments may test different settings for this parameter, as the likelihood of a sender unnecessarily engaging in RTPs may affect the receiver's strategic choice.

6.8.2 Sender Preferences

In some scenarios of the RTP fraud game, bank nodes hold limited or no liability for fraud that occurs. Instead, liability is assumed by the victim node. Naturally, this potential liability may affect customer nodes' willingness to engage with RTPs. To study this, I introduce the sender game, in which customers may strategically select their personal thresholds, CT. Customer thresholds in this case are imposed at the account level and apply to customer and fraudster nodes making payments from the account equally. When setting their thresholds, customers balance the demand of receivers with potential liability for fraud. I explore the strategic choices of customer nodes in the sender game described below by applying EGTA in several game configurations.

6.8.2.1 Sender Experiment Overview

Bank nodes are initialized as described in Section 6.4.2, however, bank investment in RTP fraud detection is handled with a slight modification. Rather than a strategic choice, the investment level of bank nodes is treated as a game configuration parameter applied to all bank nodes in the model. Thus, all bank nodes will invest the same amount in RTP fraud detection. The initialization of their fraud detectors follows the same procedure as the RTP fraud game, so there still exists some variation in detection capability between bank nodes. Customer node initialization follows the RTP fraud game steps, with the exception of customer RTP thresholds. Each customer node will strategically select its threshold from $CT = \{0, 200, 400, 600, 800, 1000\}$. In the RTP fraud game, all customer nodes preferred to send RTPs and thus no customers were assigned a threshold $CT = 0$. I relax this assumption in the sender model to allow strategic customers this choice. All customers abide by the same urgency parameter as described in the RTP fraud game. Fraudsters in this model do not follow any particular strategies and randomly select their victim from a uniform draw of the customer nodes.

The sender game is played very similarly to the RTP fraud game, unfolding over T time steps. In each time step, η payments are made, with probability μ one of these payments is made by a fraudster node. Each payment is randomly generated as outlined in the RTP fraud game. The payment's type is determined similarly to the RTP fraud game case, but eliminates the need to check the bank threshold. A payment sent by customer C_i of value v is eligible to be sent as an RTP if $v \leq CT_{C_i}$. An eligible payment is automatically sent in real time if the receiver demands real-time payments. If the receiver is willing to accept any payment type, customer urgency is used to determine its type. When a payment is not eligible to be sent in real time, it will be sent as a standard payment if the receiver allows. However, if the receiver demands RTPs, the attempt must be terminated. These same payment type rules apply to the fraudster node when its victim is C_i.

After the T time steps, each customer node receives a payoff for the performance of its threshold strategy. The payoff is determined by the total value of payments of any type sent in the network by the customer (***VS***) and the customer's liability for fraudulent real-time payments. The latter is calculated from the total value of fraudulent RTPs sent from the customer's account, ***VRF***. The payoff for customer node C_i is:

$$VS_{C_i} - \alpha VRF_{C_i}$$

I apply EGTA to identify equilibria for games with $n = 10$ customer nodes, $b = 2$ bank nodes, and $m = 1$ fraudster node. The experiments are conducted for several settings of bank fraud detection investment level and customer liability for fraud to understand how these factors affect customers' strategic choices. I test $\alpha = \{0.5, 1\}$ and *FDI* = $\{$*none*, *high*$\}$ for a total of four game configurations. The settings for the remaining game parameters are listed in Table 6.5. Any unlisted parameter retains its setting from the RTP fraud game.

Parameter	Setting
n	10
b	2
m	1
α	{0.5, 1}
FDI	{*none*, *high*}

Table 6.5: A list of the parameter settings employed for the sender game. Unlisted parameters follow the settings outlined for the RTP fraud game.

6.8.2.2 Sender Nash Equilibrium

The Nash equilibria show that customers tend to adopt an all–or–nothing approach to real-time payments. All equilibria are pure strategy equilibria either selecting $CT = 0$ or $CT = 1000$. From these findings, I conclude that customers nodes are not likely to only partially adopt RTPs (e.g., $0 < CT < 1000$) as a reaction to the threat of fraud. It is likely if a customer were to do this, there would be other intrinsic motivations, assuming the customer were behaving rationally.

Customer nodes with partial liability for fraud are still willing to engage with real-time payments, especially if bank nodes employ fraud detection. However, if customers nodes are fully liable, their willingness to use RTPs depends on the measures banks take to protect them. Specifically, customers will only send RTPs if banks employ a high level of fraud detection. This behavior was confirmed by analyzing the $FDI = low$ setting when $\alpha = 1$ as well. Comparing these results to findings from the RTP fraud game suggest it may be reasonable for banks and customers to share liability for RTP fraud. Customer nodes in the sender model are willing to take on some, though not all, liability for fraud. Likewise, in the RTP fraud game, I note bank nodes appear willing to assume a small amount of liability for RTP fraud before they begin to limit customer access to those payments. However, it is unlikely banks would be able to force customers to assume full liability and still expect customers to engage with RTPs. In the sender model, customers demand a high investment in RTP fraud detection when customers assume full liability for RTP fraud. Unfortunately, in the RTP fraud model, banks do not implement any fraud detection measures for RTPs when they have no liability for fraud occurring with these payments.

While these initial observations yield some interesting insights into RTP fraud liability, further exploration is necessary to confirm these results. It is likely, the strategic choices of bank nodes may be altered by interaction with strategic customer nodes. Further work exploring strategic decisions as a result of liability assignment should employ a single model accounting for strategic behavior of both banks and customers.

6.9 Summary

In this chapter, I explore banks' strategic mitigation of fraud risk in RTPs, balancing benefit to customers with vulnerability to fraud. To maintain this balance, banks often supplement the use of fraud detection with restrictions to RTP use. I investigate how banks may strategically trade-off between the use of these two mitigation methods and the strategic response of fraudsters.

I study this problem in an extended FCN modeling a payment system in which an RTP fraud game is played among banks and fraudsters. The game is analyzed using EGTA to identify Nash equilibria for nine game configurations defined by bank liability for RTP fraud and customer demand for RTPs. The results indicate that with no liability, bank nodes are willing to allow unrestricted access to RTPs and do not invoke fraud detection. However, as banks take on more liability, they become more likely to employ higher fraud detection measures and place restrictions on RTP use. In response, fraudster nodes must adjust from targeting only RTPs to a willingness to commit fraud with any payment type. I further study the strategic decision of bank nodes by employing the strategic feature gains assessment to gauge the relative and individual importance of each risk mitigation technique. The results of the assessments identify the importance of initial restrictions to RTPs for mitigating fraud risk. By exploring outcomes for network participants, I find bank strategic decisions in equilibrium effectively lower fraud risk with little impact on customer outcomes.

Lastly, I acknowledge that customer nodes may also approach their use of real-time payments strategically. I introduce a preliminary analysis of strategic customers which studies customers separately as payment senders and receivers. The Nash equilibria from these experiments show customer nodes in these models are willing to accept Any type of payment and when willing to engage with RTPs, do so without personal limitations. Comparing the preliminary customer analysis to results from the RTP fraud game indicate it may be possible for customers and banks to share liability for fraud without major disruptions to RTP use. Confirming this behavior in a model incorporating both strategic banks and customers is left for future work.

Set name	All strategies
BT-only	0_none, 200_none, 400_none, 600_none, 800_none, 1000_none
BT and FDI (A1)	0_low, 0_high, 200_low, 200_high, 400_low, 400_high, 600_low, 600_high, 800_low, 800_high 1000_low, 1000_high
FDI-only	1000_none, 1000_low, 1000_high
BT and FDI (A2)	0_none, 0_low, 0_high, 200_none, 200_low, 200_high, 400_none, 400_low, 400_high, 600_none, 600_low 600_high, 800_none, 800_low, 800_high
no RTPs	0_none
FDI or BT (A3)	200_none, 400_none, 600_none, 800_none, 1000_none, 1000_low, 1000_high
FDI or BT (A4)	0_none, 200_none, 400_none, 600_none, 800_none, 1000_low, 1000_high

Table 6.6: A list of bank strategies included in the base and deviation sets listed in Table 6.3.

CHAPTER 7

Conclusion

The financial system plays a vital role in connecting customers and financial institutions for the purposes of financial transactions. This system is ever-evolving as economic and technological changes impact the system's various subsystems and services. Often, these changes introduce strategic decisions for network participants such as whether or not to embrace a new service. Such changes and ensuing strategic choices require careful analysis to understand how they may affect and transform the existing system. This dissertation employs computational methods to study the effects of strategic decision-making in the financial system. I capture the complex interactions of financial network participants in agent-based models and apply empirical game-theoretic analysis to identify equilibrium agent behavior. Analysis of the equilibrium and outcomes when agents play according to it provide an understanding of strategic choices and their impacts on the network. I demonstrate the effectiveness of these methods for gaining insight into strategic decision-making in four case studies of the financial network. As each study employs the extended FCN as a basis for the agent-based model, these studies also illustrate its capacity to capture diverse financial scenarios.

7.1 Portfolio Compression

In this work, I study compression as a strategic decision made by bank nodes in the extended FCN model. Compression is decided with a vote by bank nodes on the cycle and performed only when all nodes unanimously vote in favor of the compression. I find that banks nodes confronted with a compression decision prefer strategies that use simple, local information to the unconditional accept or reject strategies. When bank nodes make the compression decision strategically, the network rarely experiences an increase in systemic risk. The analysis also discovers a connection between the recovery rate of insolvent banks and the perceived role of debt cycles in systemic risk. When insolvent banks recover few assets, banks see cycles as modes for propagating systemic risk within the network. On the other hand, when insolvent banks recover a larger portion of their assets, banks view cycles as beneficial for cushioning the network from potential negative shocks. This analysis—the first study of compression from the ex-ante perspective—demonstrates the benefit of strategic compression for bank nodes and the importance of the insolvent bank recovery rate in this decision.

7.2 Flagging Payments for Fraud Detection

This work addresses strategic use of fraud detection by bank nodes in an extended FCN model when invoking the fraud detector may be costly. Bank nodes strategically flag payments for review by their detectors, attempting to balance the incurred cost with potential liability for fraud. My analysis finds strong bank nodes are able to take rising costs into account, switching from flagging most payments to only highly suspicious payments when detector costs become high. Weak banks, on the other hand, are concerned only with liability for fraud and always flag most payments. This is further supported by examining the breakdown of costs experienced by bank nodes. Strong banks are better off when strategically flagging, but outcomes for weak banks are best when all payments are flagged. These findings demonstrate the importance of considering relative ability in strategic decision-making. Strong banks do not make their strategic choice on their fraud detection capabilities alone, but with the knowledge that banks with weaker capabilities exist in the network.

7.3 Adoption of Real-Time Payments with Credit Risk

In this work, I analyze the adoption of real-time payments when there is risk of customer overdrafts. Bank nodes in an extended FCN model strategically allow customers the ability to send RTPs by setting minimum thresholds on deposit holdings required for RTP access. I find that bank nodes set thresholds that allow most, but not all, customers to send RTPs. In particular, banks are more likely to adopt lower thresholds when standard payments face longer processing delays and customer overdrafts are less likely. I also discover that all network participants tend to be better off when RTPs have no restrictions compared to the outcomes under equilibrium. Thus, if unrestricted access to sending RTPs is a desirable outcome, additional incentives must be provided for banks to reach this decision or it must come in the form of a mandate.

7.4 Mitigating Fraud Risk in Real-Time Payments Systems

This work extends the previous studies of fraud detection and real-time payments to investigate fraud risk in real-time payment systems. I employ the extended FCN model to study bank nodes attempting to limit their vulnerability to fraudsters by strategically investing in fraud detection and controlling allowance of RTP use by customers. Bank nodes in this scenario set high thresholds, but generally also invest highly in fraud detection. This behavior drives fraudsters to follow strategies defined by willingness to attempt any payment type and high exploitation of banks based on historical success rates. I identify threshold setting as an important technique for banks in the fight against fraud. By adopting measures to mitigate fraud, banks could negatively impact customer experience in the network. However, I find this is not the case, as the equilibrium of bank nodes drastically impacts fraudsters with minimal disruption to customers. These results offer a timely analysis of fraud prevention in this newer payments space.

7.5 Strategic Feature Gains Assessment

I introduce the strategic feature gains assessment as a method for further analyzing the strategy space and illustrate its ability to contribute to our understanding of strategic interactions in two works. As a methodology, the strategic feature gains assessment offers two particular advantages over other methods such as NE-regret [Jordan, 2010]. First, the assessment focuses on the benefit provided by a group of strategies, rather than individual strategies. In some cases, this may align better with the goals of the analysis. For example, consider performing the assessment to inform new strategy generation. In this case, one might test groups of strategies united by common elements, determine which are most useful, and generate new variations of those beneficial strategies for future experiments. Note, however, that the assessment is still capable of measuring gains for individual strategies by defining singleton strategy deviation sets. Second, the strategic feature gains assessment provides greater control of the analysis by defining disjoint base and deviation sets. In doing so, the assessment effectively compares two different scenarios: one before and the second, after access to a group of strategies is granted. Furthermore, the union of these disjoint sets need not be the full strategy set. This allows the assessment to calculate benefit with respect to specific hypothetical contexts of interest. For example, calculating the benefit to banks when real-time payments are introduced to a system previously devoid of such payments. When applying NE-regret, the comparison point is the Nash equilibrium of the game with the full strategy set. Thus, we can not necessarily completely omit strategies meeting a specific criteria if one or more are in the support of the Nash equilibrium.

7.6 Discussion

This dissertation lays the foundations for studying potential changes in financial networks using agent-based modeling and empirical game-theoretic analysis. Future directions may address some of the limitations present in this work to continue providing valuable understanding of strategic decision-making in financial networks. First, all of these works employ simplified models of the network and expanding these models may uncover additional factors for strategic decision-making. However, additional complexity will increase the parameter space of these models, perhaps beyond what is computationally feasible to analyze using the current methods. For such analysis, methods employing deep learning to learn agent payoffs may be more suitable. Gatchel and Wiedenbeck [2023] learns payoffs over a range of possible game parameters using a single neural network, demonstrating such methods can be quite data efficient. Second, the agent-based models in this work do not incorporate real-world data, which was unavailable for this work. Design decisions are instead grounded in real-world observations and logic. Though if possible, future work might consider calibrating these models using real-world data. The use of a synthetic data set in Chapter 4 may provide an example of how such calibration can be achieved.

BIBLIOGRAPHY

Aisha Abdallah, Mohd Aizaini Maarof, and Anazida Zainal. Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113, 2016.

Daron Acemoglu, Asuman Ozdaglar, and Alirea Tahbaz-Salehi. Systemic risk and stability in financial networks. *American Economic Review*, 105(2):564–608, 2015.

Amazon AWS. Amazon Fraud Detector. https://aws.amazon.com/fraud-detector/, 2024.

Nasser Arshadi. Blockchain Platform for Real-Time Payments: A Less Costly and More Secure Alternative to ACH. *Technology & Innovation*, 21(1):3–9, 2019.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 2002.

Alejandro Correa Bahnsen, Djamila Aouada, Aleksandar Stojanovic, and Bjorn Ottersten. Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*, 51: 134–142, 2016.

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating Evaluation. In *32nd Conference on Neural Information Processing Systems*, 2018.

Michaela Beals, Marguerite DeLiema, and Martha Deevy. Framework for a Taxonomy of Fraud. *Stanford Center on Longevity*, 2015.

Morten L. Bech. Intraday Liquidity Management: A Tale of Games Banks Play. *FRBNY Economic Policy Review*, 2008.

Morten L. Bech, Yuuki Shimizu, and Paul Wong. The Quest for Speed in Payments. *BIS Quarterly Review*, 2017.

Board of Governers of the Federal Reserve System. The Federal Reserve Payments Study: 2022 Triennial Initial Data Release. https://www.federalreserve.gov/paymentsystems/fr-payments-study.htm, 2023.

Erik Brinkman and Michael P. Wellman. Empirical Mechanism Design for Optimizing Clearing Interval in Frequent Call Markets. In *ACM Conference on Economics and Computation*, 2017.

Ben-Alexander Cassell and Michael P. Wellman. EGTAOnline: An experiment manager for simulation-based game studies. In Francesca Giardini and Frederic Amblard, editors, *Multi-Agent Based Simulation XIII*, volume 7838 of *Lecture Notes in Artificial Intelligence*, pages 85–100. Springer, 2013.

Frank Cheng and Michael P. Wellman. Accounting for strategic response in an agent-based model of financial regulation. In *18th ACM Conference on Economics and Computation*, pages 187–203, 2017.

Frank Cheng, Junming Liu, Kareem Amin, and Michael P. Wellman. Strategic Payment Routing in Financial Credit Networks. In *ACM Conference on Economics and Computation*, pages 721–738, 2016.

Ayush Chopra, Alexander Rodriguez, Jayakumar Subramanian, Arnau Quera-Bofarull, Balaji Krishnamurthy, and B. Aditya Prakash. Differentiable Agent-based Epidemiology. In *22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

Committee on Payments and Market Infrastructures. Fast payments: Enhancing the speed and availability of retail payments. Technical Report 154, Bank for International Settlements, 2016.

Pranav Dandekar, Ashish Goel, and Ramesh Govindan. Liquidity in credit networks: A little trust goes a long way. In *12th ACM Conference on Electronic Commerce*, pages 147–156, 2011.

Pranav Dandekar, Ashish Goel, Michael P. Wellman, and Bryce Wiedenbeck. Strategic Formation of Credit Networks. In *ACM Transactions on Internet Technology*, volume 15, pages 1–41, 2015.

D. B. DeFigueiredo and E. T. Barr. TrustDavis: A non-exploitable online reputation system. In *Seventh IEEE International Conference on E-Commerce Technology*, pages 274–283, 2005.

Linda Delamaire, Hussein Abdou, and John Pointon. Credit card fraud and detection techniques: a review. *Banks and Bank Systems*, 4, 2009.

Deloitte. Economic impact of real-time payments. Technical report, Mastercard, 2019.

Marco D'Errico and Tarik Roukny. Compressing Over-the-Counter Markets. https://ssrn.com/abstract=2962575, 2019. European Systemic Risk Board Working Paper Series No 44.

Danial Dervovic, Parisa Hassanzadeh, Samuel Assefa, and Prashant Reddy. Non-Parametric Stochastic Sequential Assignment With Random Arrival Times. In *30th International Joint Conference on Artificial Intelligence*, pages 4214–4220, 2021.

Alexander Diadiushkin, Kurt Sandkuhl, and Alexander Maiaitin. Fraud Detection in Payments Transactions: Overview of Existing Approaches and Usage for Instant Payments. *Complex Systems Informatics and Modeling Quarterly*, pages 72–88, 2019.

Diego A. Diaz, Ana Maria Jimenez, and Cristian Larroulet. An agent-based model of school choice with information asymmetries. *Journal of Simulation*, 2019.

Damien Dugauquier, Geertjan va Bochove, Alain Raes, and Jean-Julien Ilunga. Digital payments: Navigating the landscape, addressing fraud, and chartering the future with Confirmation of Payee solutions. *Journal of Payments Strategy & Systems*, 17:359–371, 2023.

Larry Eisenberg and Thomas H. Noe. Systemic risk in financial systems. *Management Science*, 47(2):236–249, 2001.

Matthew Elliott, Benjamin Golub, and Matthew O. Jackson. Financial networks and contagion. *American Economic Review*, 104(10):3115–3153, 2014.

European Central Bank. What are instant payments? https://www.ecb.europa.eu/paym/integration/retail/instant_payments/html/index.en.html, 2023.

John Fearnley, Martin Gairing, Paul Goldberg, and Rahul Savani. Learning Equilibria of Games via Payoff Queries. *Journal of Machine Learning Research*, 16:1305–1344, 2015.

Federal Reserve Board. About the FedNow Service. https://www.frbservices.org/financial-services/fednow/about.html, 2021.

Marco Galbiati and Kimmo Soramaki. An Agent-Based Model of Payment Systems. *Journal of Economic Dynamics and Control*, 35(6):859–875, 2008.

Madelyn Gatchel and Bryce Wiedenbeck. Learning Parameterized Families of Games. In *22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

Arpita Ghosh, Mohammad Mahdian, Daniel M. Reeves, David M. Pennock, and Ryan Fugger. Mechanism Design on Trust Networks. In *Third International Workshop on Internet and Network Economics*, pages 257–268, 2007.

Paul Glasserman and H. Peyton Young. How likely is contagion in financial networks? *Journal of Banking & Finance*, 50:383–399, 2015.

Claire Greene, Marc Rysman, Scott D. Schuh, and Oz Shy. Costs and Benefits of Building Faster Payment Systems: The U.K. Experience and Implications for the United States. Technical Report 14-5, Federal Reserve Bank of Boston Research Paper Series Current Policy Perspectives, 2014.

Zhiling Guo and Dan Ma. Catching the Fast Payments Trend: Optimal Designs and Leadership Strategies of Retail Payment and Settlement Systems. *MIS Quarterly*, 47(2), 2023.

Zhiling Guo, Rob Kauffman, Mei Lin, and Dan Ma. Near Real-Time Retail Payment and Settlement Systems Mechanism Design. Working paper, SWIFT Institute, 2015. URL https://swiftinstitute.org/wp-content/uploads/2015/11/WP-No-2014-004-1.pdf.

Monika Hartmann, Lola Hernandez van Gijsel, Mirjam Plooij, and Quentin Vandeweyer. Are instant payments becoming the new normal? A comparative study. *European Central Bank Occasional Paper Series*, 2019.

Rumiko Hayashi and Ying Lei Toh. Mobile Banking Use and Consumer Readiness to Benefit from Faster Payments. *Federal Reserve Bank of Kansas City Economic Review*, 105(1):1–5, 2020.

Matti Hellqvist and Kasperi Korpinen. Instant Payments as a new normal: Case study of liquidity impacts for the Finnish market. Technical Report 7, BoF Economics Review, 2021.

Nicolas Hoertel, Martin Blachier, Carlos Blanco, Mark Olfson, Marc Massetti, Marina Sanchez Rico, Frederic Limosin, and Henri Leleu. A stochastic agent-based model of the SARS-CoV-2 epidemic in France. *Nature Medicine*, 26(9):1417–1421, 2020.

S.N. John, Okokpujie Kennedy O., C. Anele, F. Olajide, and Chinyere Grace Kennedy. Realtime Fraud Detection in the Banking Sector Using Data Mining Techniques/Algorithm. In *International Conference on Computational Science and Computational Intelligence*, 2016.

Kurt Johnson, James J. McAndrews, and Kimmo Soramaki. Economizing on Liquidity with Deferred Settlement Mechanisms. *FRBNY Economic Policy Review*, 2004.

Nicole Jonker, Carin van der Cruijsen, Michiel Bijlsma, and Wilko Bolt. Pandemic payment patterns. *Journal of Banking & Finance*, 143, 2022.

Patrick R. Jordan. *Practical Strategic Reasoning with Applications in Market Games*. PhD thesis, University of Michigan, 2010.

Panagiotis Kanellopoulos, Maria Kyropoulou, and Hao Zhou. Debt Transfers in Financial Networks: Complexity and Equilibria. In *22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

Dean Karlan, Markus Mobius, Tanya Rosenblat, and Adam Szeidl. Trust and Social Collateral. *Quarterly Journal of Economics*, 124(3):1307–1361, 2009.

Erwin Kulk. Request to pay: Monetising the instant payments investment. *Journal of Digital Banking*, 5(3):193–203, 2021.

Wangli Lin, Li Sun Qiwei Zhong, Can Liu, Jinghua Feng, Xiang Ao, and Hao Yang. Online credit Payment Fraud Detection via Structure-Aware Hierarchical Recurrent Neural Network. In *30th International Joint Conference on Artificial Intelligence*, pages 3670–3676, 2021.

T. Sirisha Madhuri, E. Ramesh Babu, B. Uma, and B. Muni Lakshmi. Big-data driven approaches in materials science for real-time detection and prevention of fraud. *Materials Today*, 81:969–976, 2023.

Mastercard. The importance of dependability in real-time payments. Technical report, Mastercard, 2020.

Nick Maynard. Financial Crime Prevention: Segment Analysis, Key Trends, & Market Forecasts 2023-2027. Technical report, Juniper Research, 2023.

Katherine Mayo and Michael P. Wellman. A strategic analysis of portfolio compression. In *2nd ACM International Conference on AI in Finance*, pages 1–8, 2021.

Katherine Mayo, Shaily Fozdar, and Michael P. Wellman. An agent-based model of strategic adoption of real-time payments. In *2nd ACM International Conference on AI in Finance*, pages 1–9, 2021.

Katherine Mayo, Shaily Fozdar, and Michael P. Wellman. Flagging Payments for Fraud Detection: A Strategic Agent-Based Model. In *3rd International Workshop on Modeling Uncertainty in the Financial World*, 2023.

Katherine Mayo, Nicolas Grabill, and Michael P. Wellman. Fraud Risk Mitigation in Real-Time Payments: A Strategic Agent-Based Analysis. In *33rd International Joint Conference on Artificial Intelligence*, 2024.

Microsoft. Dynamics 365 Fraud Protection pay-as-you-go pricing. https://dynamics.microsoft.com/en-us/ai/fraud-protection/pricing/, 2024.

Thanh H. Nguyen, Mason Wright, and Michael P. Wellman. Multi-Stage Attack Graph Security Games: Heuristic Strategies with Emprical Game-Theoretic Analysis. In *Workshop on Moving Target Defense*, 2017.

The Nilson Report. Card Fraud Losses Worldwide. *Nilson Report*, 1254:6–8, 2023.

Dominic O'Kane. Optimising the multilateral netting of fungible OTC derivatives. *Quantitative Finance*, 17(10):1523–1534, 2017.

Cyril Onwubiko. Fraud matrix: A morphological and analysis-based classification and taxonomy of fraud. *Computers & Security*, 96, 2020.

Rakesh Padhan and K.P. Prabheesh. The economics of COVID-19 pandemic: A survey. *Economic Analysis and Policy*, 70:220–237, 2021.

Leslie Parrish and Josh Frank. An Analysis of Bank Overdraft Fees: Pricing, Market Structure and Regulation. *Journal of Economic Issues*, 45, 2011.

L.C.G. Rogers and L.A.M Veraart. Failure and rescue in an interbank network. *Management Science*, 59(4):882–898, 2013.

Abhimanyu Roy, Jingyi Sun, Robert Mahoney, Loreto Alonzi, Stephen Adams, and Peter Beling. Deep learning detecting fraud in credit card transactions. In *Systems and Information Engineering Design Symposium*, 2018.

Nick F. Ryman-Tubb, Paul Krause, and Wolfgame Garn. How Artifical Intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artifical Intelligence*, 76:130–157, 2018.

Yusuf Sahin, Serol Bulkan, and Ekrem Duman. A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, 40:5916–5923, 2013.

Marouane Ait Said and Abdelmajid Hajami. AI Methods Used for Real-Time Clean Fraud Detection in Instant Payment. In *13th International Conference on Soft Computing and Pattern Recognition*, 2021.

Irina Sakharova. Payment card fraud: Challenges and solutions. In *IEEE International Conference on Intelligence and Security Informatics*, 2012.

Javier Santamaría. Developments in instant payments. *Journal of Payments Strategy & Systems*, 13(3):190–193, 2019.

Steffen Schuldenzucker and Sven Seuken. Portfolio Compression in Financial Networks: Incentives and Systemic Risk. https://ssrn.com/abstract=3135960, 2021. Working Paper.

Steffen Schuldenzucker, Sven Seuken, and Stefano Battison. Default ambiguity: Credit default swaps create new systemic risks in financial networks. *Management Science*, 66(5):1981–1998, 2020.

Megan Shearer, Gabriel Rauterberg, and Michael P. Wellman. Learning to Manipulate a Financial Benchmark. In *Fourth ACM International Conference on AI in Finance*, 2023.

Ozy Shy and Zhu Wang. Why Do Payment Card Networks Charge Proportional Fees? Technical report, The Federal Reserve Bank of Kansas City, 2010. URL https://www.kansascityfed.org/documents/5325/pdf-rwp08-13.pdf. Research Working Papers.

Stripe. Online and ecommerce fraud statistics that are predicting the future of fraud. https://stripe.com/resources/more/online-and-ecommerce-fraud-statistics, 2023a.

Stripe. Credit card fraud detection and prevention: Best practices and tactics for businesses. https://stripe.com/resources/more/credit-card-fraud-detection-and-prevention, 2023b.

Stripe. Protect your business from fraud. https://stripe.com/radar/pricing, 2024.

Daniel Susskind and David Vines. The economics of the COVID-19 pandemic: an assessment. *Oxford Review of Economic Policy*, 36:S1–S3, 2020.

John L. Taylor and Tony Galica. A New Code to Protect Victims in the UK from Authorized Push Payments Fraud. *Banking & Finance Law Review*, 35:327–332, 2020.

Ying Lei Toh and Thao Tran. How the COVID-19 Pandemic May Reshape the Digital Payments Landscape. *Federal Reserve Bank of Kansas: Payments System Research Briefing*, 2020.

Luitgard A.M. Veraart. When does portfolio compression reduce systemic risk? https://ssrn.com/abstract=3311176, 2019. Working Paper.

Xintong Wang, Christopher Hoang, Yevgeniy Vorobeychik, and Michael P. Wellman. Spoofing the Limit Order Book: A Strategic Agent-Based Analysis. *Games*, 12(2), 2021.

Michael P. Wellman. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems*, 30:1175–1189, 2016.

Michael P. Wellman and Elaine Wah. Strategic agent-based modeling of financial markets. *RSF: The Russell Sage Foundation Journal of the Social Sciences*, 3(1):104–119, 2017.

Michael P. Wellman, Eric Sodomka, and Amy Greenwald. Self-confirming price-prediction strategies for simulataneous one-shot auctions. In *28th Conference on Uncertainty in Artificial Intelligence*, 2012.

Michael P. Wellman, Karl Tuyls, and Amy Greenwald. Empirical Game-Theoretic Analysis: A Survey. https://arxiv.org/abs/2403.04018, 2024. Working Paper.

Jarrod West and Maumita Bhattacharya. Intelligent financial fraud detection: A comprehensive review. *Computers & Security*, 57:47–66, 2016.

Julius Weyman. Risks in Faster Payments. *Retail Payments Risk Forum Working Paper*, 2016.

Janet L. Yellen. Interconnectedness and Systemic Risk: Lessons from the Financial Crisis and Policy Implications, 2013. Speech at the American Economic Association/American Finance Joint Luncheon, San Diego, California.

Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. Federated Meta-Learning for Fraudulent Credit Card Detection. In *29th International Joint Conference on Artificial Intelligence*, pages 4654–4660, 2020.

Lin Zhong, Qianhong Wu, Jan Xie, Zhenyu Guan, and Bo Qin. A secure large-scale instant payment system based on blockchain. *Computers & Security*, 84:349–364, 2019.

Hao Zhou, Yongzhao Wang, Konstantinos Varsos, Nicholas Bishop, Rahul Savani, Anisoara Calinescu, and Michael Wooldridge. A strategic analysis of prepayments in financial credit networks. In *33rd International Joint Conference on Artificial Intelligence*, 2024.