Convergence and Equilibria of Pooling Games in Proof-of-Stake Blockchains

Christina Ovezik

Master of Science

Advanced Technology for Financial Computing

School of Informatics

University of Edinburgh

2021

Abstract

Blockchains have stirred a lot of interest both in industry and academia during the past decade, as their tamper-proof nature and promise of decentralisation opened new doors for financial systems, and beyond. As with any peer-to-peer network, blockchains are reliant on the participants' voluntary choice to maintain them, therefore they need to provide incentives to their members, to ensure sufficient participation and, subsequently, stability of their network. To that end, all blockchain protocols include some sort of incentive mechanism, but as several studies have pointed out, not all of those mechanisms are effective in promoting desired properties of a system, such as a high degree of decentralisation. In this work, we extend the game-theoretic model that was proposed in the context of the reward scheme of a Proof-of-Stake blockchain, with the ultimate goal of bringing the theoretical framework closer to the real-life system. We also develop a configurable simulation engine that plays out the "game" under several different settings, and we use it to run experiments that help us extend our knowledge on the convergence and equilibria of such systems, and gain insights on the behaviour of their participants.

Acknowledgements

I would like to thank my supervisor, Prof. Aggelos Kiayias FRSE, who introduced me to the mesmerising world of blockchains through his lectures and guided me through the complex —but ultimately fun!— modelling process of Proof-of-Stake systems. I would also like to thank Aikaterini-Panagiota Stouka for dedicating her time to help me get a better understanding of the work I was trying to extend.

A very special thanks is owed to my mom, my dad and my sister for always being by my side —even from thousands of miles away— and to all my loved ones, in Greece and in Edinburgh, for helping me get through this challenging year and for believing in me every step of the way —even at times when I was losing faith in myself.

Last, I would like to express my gratitude towards the Edinburgh Blockchain Technology Laboratory for funding this peoject.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Christina Ovezik*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

More than a decade after the first blockchain protocol was established [1], we now appreciate that there exist several applications for this technology beyond its original scope, and we observe that it has taken up a key role both in industry and academia. Named as a disruptive technology, blockchains are expected to radically influence many sectors in the near future, including healthcare and governance [2].

An important aspect of this technology is decentralisation, namely removing control from individual trusted parties and distributing it across a large number of independent actors [3]. This transition from a central entity to a peer-to-peer network successfully eliminates the necessity for trust, but it is reliant on the participants' voluntary choice to maintain the network. In order for such systems to thrive and remain stable they, thus, need to provide the right incentives for people to get involved —and stay involved— with them.

This is where the field of Blockchains meets Game Theory and Behavioural Economics, as it is necessary to analyse and understand how people behave under certain conditions, in order to design mechanisms that align the interests of the network participants with the interests of the system [4].

Even though incentive mechanisms have been used in blockchain technology since its first appearance in Bitcoin [1], their design was initially not backed up by relevant research, hence there were no guarantees for their effectiveness. In fact, numerous studies that analysed the Bitcoin protocol after it was already in use concluded that its incentive mechanism may not be sufficient to convince participants to engage with the protocol in the specified (honest) way, or that it may steer the system towards

unfavourable, fairly centralised states [5, 6, 7, 8, 9].

With regard to a newer, more energy-efficient type of blockchain protocols — referred to as Proof-of-Stake blockchains— that differ from the first ones —referred to as Proof-of-Work blockchains— in the way they achieve consensus among the network participants, less research work has been conducted about their incentive mechanisms and their eventual stability. One of the few formal analyses of such systems was performed by Brünjes et al [10] in the context of the Cardano network[1], which makes use of the Ouroboros Proof-of-Stake blockchain protocol [11]. Their game-theoretic analysis proved that the Reward Sharing Scheme used in Cardano incentivises participants to behave in an honest manner and leads the system towards a state of equilibrium with favourable properties, such as a satisfactory degree of decentralisation.

Brünjes et al additionally ran simulations to put their theory to test and the results they got were in line with their analysis. However, several assumptions were made through the process and restrictions were added to the model, begging the question of whether it is comparable to the real-life system.

1.2 Objectives

The aim of this project is to extend the game-theoretic model that was introduced by Brünjes et al for the analysis of the broad class of reward schemes of Cardano [10]. Our main focus will be to align the theoretical framework as much as possible with the real-life conditions of the system and to develop a configurable simulation engine, that will allow us to conduct experiments under several different settings. This will give us the chance to further explore the properties of the incentive mechanisms of Proof-of-Stake blockchains and to draw conclusions about the emergent properties of such systems.

The immediate objective is to determine whether the simulations converge to the theoretical equilibrium described in [10], where the system stabilises to a favourable state, and to explore how this potential convergence depends on the different parametrisations of the system and on any potential assumptions about the behaviour of its participants. What are the necessary conditions for the system to stabilise? Can we remove some of the assumptions that were made, some knowledge from the players or some degree of rationality and still attain satisfying results? What happens if participants behave rationally but myopically? Is the final state of the system fair to all

[1] https://cardano.org

players and sufficiently decentralised? And what is the trajectory that the system follows until the point of equilibrium, if such exists? These are some of the questions we set out to resolve, with the ultimate goal of providing insights into the behaviour of Proof-of-Stake blockchain participants and extending our general knowledge about the equilibria and convergence properties of such systems.

1.3 Structure

This dissertation comprises four additional chapters, that focus respectively on the required background knowledge, the methodology that was employed, the experiments that were conducted and the conclusions that were drawn.

Specifically, in the next chapter, we provide an overview of the background knowledge that is needed for one to fully understand this piece of work, introducing concepts from Game Theory, Behavioural Economics and Blockchain Technologies. Chapter 3 then addresses the specific techniques that were used to accomplish our goals and the design choices that were made. In chapter 4, we present the different experiments that were performed and we analyse the results that were attained. Lastly, in the final chapter, we conclude by summarising our work and proposing several directions for future research to extend it.

Chapter 2

Background & Related Work

In this chapter, we introduce the preliminaries that are needed for one to better understand this piece of work. These include concepts from Game Theory, Behavioural Economics and Blockchain Technologies. We also review here relevant research work that influenced the design choices that were made later on.

2.1 Game theory

The field of Game Theory seeks to formally describe interactions among decision-making agents (be it humans, animals or computer systems) and the strategic play that arises within such interactions. Since the introduction of modern concepts in the subject area during the 20_{th} century [12, 13, 14, 15], game-theoretic models have been used to analyse and better understand numerous real-life scenarios, including, but not limited to, financial markets, traffic congestion and voting.

Game Theory acknowledges several types or classes of games, which can be categorised based on the way they transpire (simultaneous or sequential), the degree of knowledge that the players[1] hold about the system's state (perfect or imperfect information) , the interdependence of the players (coalitional / cooperative or non-cooperative), or the way they are represented (strategic / normal-form games which are described by a matrix or extensive form games which require more complex structures such as trees) [2] [16, 17].

[1] Note that the terms agent and player are used interchangeably in the context of Game Theory.

[2] Note that this is a non-exhaustive list of the different game types in Game Theory.

2.1.1 Strategies, Utilities & Equilibria

For a strategic-form game G with n players, we define a set of possible actions that the players can choose from, which we call *pure strategies*. Let $S_i = \{s_1, ..., s_{m_i}\}$ denote the set of m_i pure strategies that are available to player i and $S = S_1 \times ... \times S_n$ the set of possible combinations of all the players' pure strategies. A *mixed strategy* $\mathbf{x_i} = (x_i(s_1), ..., x_i(s_{m_i}))$ for player i is a vector that defines a probability distribution over their pure strategies S_i and implies that the player uses randomness to decide which strategy to play, based on the probabilities in $\mathbf{x_i}$ [3]. A *strategy profile* $\mathbf{X} = (\mathbf{x_i}, ..., \mathbf{x_n})$ is then defined as an n-tuple of the players' chosen mixed strategies. If all the strategies in $\mathbf{X}$ are pure strategies, then $\mathbf{X}$ is called a *pure strategy profile*.

To evaluate a profile of strategies, player i makes use of a *utility* or *payoff* function $u_i : S \mapsto \mathbb{R}$ that assigns a real value to each combination of pure strategies of the players (the higher the value of u_i the better this profile is considered for player i). For mixed profiles, the player calculates the *expected utility* based on the probabilities with which each strategy profile is expected to occur [17].

A crucial assumption that is made during every game-theoretic analysis is that all players seek to maximise their expected utility, which is why they are often referred to as *utility maximisers*. Under this assumption, a player's *best response* to the strategies of other players is defined as the player's strategy which yields the highest expected utility under that scenario.

When every player is playing a best-response strategy, then we arrive at a so-called *Nash Equilibrium*, where no player can increase their utility by *unilaterally* deviating from the current profile of strategies. This is a very important solution concept, as it guarantees the *stability* of a game. According to Nash's theorem, every game has a Nash equilibrium in mixed strategies [13].

2.1.2 Rationality as Common Knowledge

A fact F is considered *common knowledge* among the players of a game G if all the players in G know F and in addition know that all other players also know F, know that all players know that all players know F, and so on [17].

Since we have assumed that the goal of each player is to maximise their individual utility, then a *rational player* is expected to always choose the strategy that yields the highest expected utility for them. However, to calculate their expected utility, the

[3] Note that mixed strategies are a superset of pure strategies.

players also need to have information about the other players' (future) choices, which is not available to them. At this point, if we additionally assume that the utility function of each player is common knowledge, as well as the fact that the players are rational, then we make it easy for a players to "guess" the other players' choices, as they assume that each player will pick the strategy that generates the highest utility for them. Then, they can make their own decisions based on this information.

2.1.3 Population games

A class of games that has been defined to model interactions within *large populations* of agents is that of *population games*. Formally, a population game comprises a society $P = \{1, ..., p\}$ of $p \geq 1$ populations of agents. Agents that belong to the same population ρ have the same set of available strategies S_ρ and select a (pure) strategy from that set throughout the game play.

Population games are guaranteed to have at least one Nash equilibrium in pure strategies, which can be reached through a best-response dynamics play [18].

2.1.3.1 Inertial equilibria

Gentile introduce the concept of an *inertial equilibrium* in the context of population games [19]. In short, their approach accounts for the fact that a player may incur costs when switching their strategy from action A to action B, therefore action B would be preferred only if the increase in utility it promises is sufficient to offset this switching cost. In reality, these could be any kind of costs, such as the time cost of learning a new skill or the monetary cost of a transaction fee that needs to be paid for every stock purchase. Additionally, this concept captures the notion of psychological or *decision inertia*, which dictates that people tend to stick to their previous choices, even if they are suboptimal [20].

The inertial equilibrium is formally defined as "a distribution over the action space, where no agent has any incentive to unilaterally switch action, when accounting not only for utility gain but also for switching cost". Gentile et al also prove that inertial equilibria are a superset of Nash equilibria and proceed by describing a "better-response dynamics" algorithm for population games that takes switching costs into consideration and converges to an inertial equilibrium. They refer to this algorithm as the "natural dynamics" of the game, as it simply dictates that an agent will choose any arbitrary action, as long as it offers an improvement to their current utility, after

subtracting from the gains the cost that the switch will induce.

2.2 Behavioural Economics

Neoclassical economics assume fully rational decision-makers who have perfect knowledge of the world around them. However, more recent schools of thought suggest that such conditions are so far away from reality that it does not make sense to take them for granted when trying to model or predict real-life behaviour related to economic choice.

2.2.1 Bounded rationality

The concept of *bounded rationality* was formed to describe the conditions under which real people operate. It encapsulates the fact that, in most cases, people are not equipped with perfect information about their environment or unbounded time and computational resources to come up with all possible alternative solutions, evaluate their consequences and make meaningful comparisons between them [21].

2.2.2 Optimising vs Satisficing

Herbert Simon introduces the concept of "satisficing" as an alternative to optimising in the decision-making process of boundedly-rational agents, such as humans [21]. Satisficing (satisfy + suffice) involves choosing an option that satisfies certain constraints and is sufficiently good, e.g. in terms of its expected utility. According to studies and lab experiments, the actual thought process of humans is closer to a heuristic search than to an optimisation task and therefore can be better modelled by a "satificing" mechanism.

2.3 Blockchains

Though still considered a nascent technology, blockchains have found their way into numerous research fields and industrial applications. Their tamper-proof nature and promise of decentralisation opened new doors for financial systems and beyond.

2.3.1 Overview

A blockchain is a distributed database of transactions, that satisfies certain properties, such as *immutability*, *auditability* and *anonymity* [3]. Simply put, once a transaction has been registered in the database, it can never be removed, the status of a transaction is easily verifiable by anyone with access to that database, and the transaction parties do not have to provide any identifiable information for their transactions to be recorded[4]. It owes its name to the fact that the involved transactions are grouped in "blocks" and each block has a link to the previous one, thereby forming a "chain".

This distributed transaction ledger is shared among the nodes of a peer-to-peer network, that are also responsible for its maintenance. Carrying out this task involves expenses (hardware purchases, electricity bills, etc.), hence those members are typically compensated by the protocol with a reward of some sort that incentivises them to look after the network. Though not a requirement, blockchains are typically *public* and *permissionless*, meaning that anyone can choose to participate as a node in their network[5] [3].

2.3.2 Resource pooling & (de)centralisation

Another key property of blockchains is *decentralisation*, as they eliminate the need for a central trusted authority, such as a bank, to verify the transactions of the network [3]. Instead, all transaction blocks are broadcast to and verified by the entire network. However, a network controlled by 10 entities is not *as* decentralised as one controlled by 1000, and so on, so it makes sense to examine the *degree of decentralisation* of a blockchain system. Gencer et al measure the decentralisation of two popular blockchains, Bitcoin [1] and Ethereum [23], and conclude that both exhibit tendencies towards centralisation and that "further research is required to decentralise permissionless consensus protocols" [6].

This tendency towards centralisation stems from the fact that it makes economic sense for the network participants to form groups and combine their resources, in order to reduce personal costs and maximise profits. In the context of Bitcoin —and Proof-of-Work blockchains in general— these multi-participant entities are called mining pools, while in Proof-of-Stake systems they are referred to as *stake pools*. In most cases, the distribution of the revenue among the pool members is not defined by the

[4]Research suggests that the "de-anonymisation" of blockchain transactions is possible in certain cases, but for the average user the provided degree of anonymity is sufficient [22]

[5]From now on, we will use the term blockchain to refer to a public, permissionless blockchain.

protocol itself, but rather left for each pool do decide upon. There are several different mechanisms that have been proposed and used by pools to divide the rewards, but the common part is that, traditionally, all pools require a service fee to be paid to them by the pool members, in exchange for the participation in the pool.

2.3.3 Incentives in Blockchain systems

Every blockchain protocol includes an incentive mechanism, with the ultimate goal of securing user engagement and, subsequently, network stability. Though the general recipe is the same (issue rewards for the parties that produced transaction blocks), the specifics of each mechanism can prove of paramount importance when studying the emergent properties of those systems (for example in terms of security or decentralisation).

Incentives in Bitcoin: As the oldest blockchain protocol, Bitcoin is also the most studied one, which is why we have so much information about its components, including multiple analyses of its reward mechanism. Eyal and Sirer prove that the Bitcoin protocol is not incentive compatible[6], as there is a strategy that miners can adopt — called selfish mining— that deviates from the protocol but yields higher rewards for them [9]. Kiayias et al prove that rational participants of the Bitcoin network who need to decide between joining an existing pool or creating a new one will always centralise to one pool, owned by the player who can guarantee the lowest service costs [8].

Taxonomy of incentive mechanisms: The analysis of the different incentive mechanisms that have been used in blockchain protocols over the years revealed two prevalent categories: the *unimodal* ones, where resource holders have only one option for engaging with the protocol, and the *multimodal* ones, which allow for resource-holders to take up different roles that entail different responsibilities and result in different rewards [25].

A *linear unimodal* reward scheme suggests that an entity that commands x% of the resources will receive x% of the available rewards, in expectation. This is the general approach followed in Bitcoin, Ethereum and others [1, 23, 26]. An important issue that has been identified in this approach is the tendency towards centralisation, as resource holders expect to receive more and more rewards if they keep merging their resources.

[6] A mechanism is considered incentive compatible if it can not be strategically manipulated by anyone, and thus, results in honest behaviour being a dominant strategy [24]

In line with previous research, Brünjes et al demonstrate that linear unimodal reward schemes, such as the one used in Bitcoin, have no theoretical equilibrium that includes more than one pool [10].

In the bimodal spectrum, two subcategories have been proposed: in the *representative* approach, the resource holders vote for operators to represent them and the ones who get elected run the protocol, while the *delegative* approach allows resource holders to either engage with the protocol directly or delegate their resources to entities of their choice.

2.3.3.1 Reward Sharing Scheme of Cardano

In an attempt to incentivise high engagement with the protocol and increase the degree of the sysetm's decentralisation, Brünjes et al propose a delegative bimodal reward scheme with *capped rewards* and *incentivised pledging* [10]. Both of these concepts will be explained in this section.

Reward distribution among pools: As a first step, the total rewards of each epoch are distributed among the active pools of the system. While in general the rewards grow with the size of the pool, they stop doing so after a threshold is reached, as is dictated by this piecewise function, which is responsible for calculating the rewards that correspond to a certain pool:

$$r(\sigma,\lambda) = \frac{R}{1+\alpha} \cdot (\sigma' + \lambda' \cdot \alpha \cdot \frac{\sigma' - \lambda' \cdot \frac{1-\sigma'}{\beta}}{\beta}) \tag{2.1}$$

where

- σ is the stake of the pool at the given snapshot.
- λ is the stake that the pool owner has pledged to the pool.
- $R \in \mathbb{R}$ are the total rewards for this epoch.
- $\alpha \in [0,\infty)$ is a parameter of the reward scheme that determines the importance of the owner's pledge in the calculation of the pool's rewards.
- $\sigma' = min\{\sigma,\beta\}$
- $\lambda' = min\{\lambda,\beta\}$

- $\beta = \frac{1}{k}$ is the saturation threshold, with $k \in \mathbb{N}$ denoting the desired number of pools in the system ($k < n$).

  Note that R, α and k are fixed for a system, therefore the rewards of the pools vary based on their specific stake and pledge.

Capped rewards: With regard to the size of a pool (i.e. the stake that it is responsible for), the reward function (2.1) dictates that as the size grows, the reward grows as well, until a threshold is reached, denoted by β. After that threshold, the pool's rewards stabilise, regardless of its increase in size. Indirectly, this imposes a cap on the sizes of the pools, as there is no incentive (in fact there is counter-incentive) to have pools larger than the threshold. However, the protocol still allows for a bigger pool to exist, hence the we can view β as a *soft cap* on the pools' size. The threshold is given the value $\beta = \frac{1}{k}$, where k is the desired number of pools for the system. A pool with stake equal to or greater than β is called *saturated*.

By disincentivising the creation of large pools, the authors of [10] aim to end up with a more decentralised pool formation than past attempts. However, they acknowledge that restricting the pool sizes in this manner might result in some participants engaging in Sybil behaviour, namely assuming multiple identities in the system [27] —which in this case is equivalent to an operator splitting their pool to form multiple pools. In such a setting, the decentralisation of the system would be compromised, as the number of pools in the final configuration would not represent independent entities. To battle this issue, they introduce the concept of incentivised pledging, which is explained below.

Incentivised pledging: Pool operators in Cardano are encouraged to "pledge" a certain amount of stake when opening a pool. This pledged stake gets "locked" and can not be retrieved by the pool owner while the pool is still functioning. To give a reason to the pool operators to provide a high pledge, Brünjes et al introduce the α parameter that was seen in formula 2.1. This parameter, when given a non-zero value, results in higher-pledged pools receiving more rewards (the higher the value of α the bigger the difference in the rewards of a low-pledged and a high-pledged pool).

By favouring higher-pledged pools through the reward mechanism, the authors believe that higher commitment to the protocol will be achieved and that misuse of power due to the implicit amplification of pool operators' stake will be limited. Additionally, incentivised pledging provides counter-incentives for operators to create multiple

pools —because splitting their pledge into multiple pools would result in lower total rewards— and could therefore help in the prevention of Sybil behaviour.

However, a very high value of α exacerbates the "rich getting richer" phenomenon, which is an inherent issue in Proof-of-Stake protocols [28], therefore the parameter is tuned so that a trade-off is made between egalitarianism and Sybil resilience of the system.

Reward distribution within pools: As a further step, the protocol also handles the distribution of rewards to the individual pool members. Note that this is different from previous approaches in blockchains, which were only concerned with distributing rewards to a pool as a whole, leaving the pool owner responsible for the redistribution. This new approach of distributing rewards directly to the pool members removes the need for additional trust in the face of the pool owner.

As mentioned before, the pool owners need to declare their operational costs upon the creation of their pools. When a pool's reward gets calculated, an amount that corresponds to the declared cost is first set aside for the pool operator[7], to offset that cost. If the reward is not sufficient to cover the pool's cost, then no rewards are distributed to its members. Note that, in this case, the pool operator suffers a loss (as they have to pay the operational costs anyway), whereas the pool members do not (in the worst case they get zero rewards, but they never go negative, so there is no danger involved for them).

To compensate the operators for the added risk they have to bear and to further incentivise pool creation, the protocol allows them to set a value of their choice that determines the fraction of the rewards (after cost deduction) that will be further set aside for the operator before any additional distribution. This value is called the pool's *margin* and it is important for the operators to set it carefully, as it has the power to attract or drive away delegators.

After the operating rewards have been allocated, the remaining fraction of the pool's rewards get distributed to its members proportionally to the stake they contribute. Remember that as a pool's size get larger, its rewards increase and therefore the rewards of its members increase. However, if a pool's size exceeds the saturation point, then its total rewards stay constant and the rewards of individual members decrease, as they are allocated a smaller fraction of the same pot.

[7] According to the latest specifications of Cardano (hydra.iohk.io/build/delegation_design_spec.pdf), the terms pool operator and pool owner have a slightly different signification, but in this setting, we will use them interchangeably, to refer to the (single) person who receives the special pool rewards.

Formally put, if a pool has stake σ, cost c, margin m and pledge λ, then:
Each delegator that contributes stake a_d to the pool receives:

$$r_d = \begin{cases} (1-m)\cdot(r(\sigma,\lambda)-c)\cdot\frac{a_d}{\sigma} & \text{if } r(\sigma,\lambda) > c \\ 0 & \text{otherwise} \end{cases}$$

And the pool owner, who contributes stake $a_o = \lambda$ to the pool, receives:

$$r_o = \begin{cases} c + m\cdot(r(\sigma,\lambda)-c) + (1-m)\cdot(r(\sigma,\lambda)-c)\cdot\frac{\lambda}{\sigma} & \text{if } r(\sigma,\lambda) > c \\ r(\sigma,\lambda) & \text{otherwise} \end{cases}$$

where $r(\sigma,\lambda)$ are the rewards that a pool with stake σ and pledge λ is entitled to.

Brünjes et al formally define a game to model the above reward scheme, which we are going to refer to as *Pooling Game*, because of its involvement with the formation of pools among the resource holders[8]. They conduct a game-theoretic analysis of the game and prove that there exists an equilibrium point, where there are exactly k pools, run by the players with the highest potential profits and saturated by the delegations of other players [10].

This is precisely the game that our work focuses on, and extends, where deemed necessary, therefore more information about it will be given in the next chapter, which describes our methodology.

[8]Brünjes et al use the term "Stake Pools Game" but we choose the more general "Pooling Game" to account for the fact that the model could easily be extended to analyse systems that do not run on Proof-of-Stake.

Chapter 3

Methodology

In this chapter, we go over the specific techniques that were used to accomplish our goals, namely modelling the behaviour of Proof-of-Stake blockchain participants and analysing it through configurable simulations. First, we give a description of the game model that was developed and we justify any design choices that were made, and then we provide an overview of how the simulation of the game unfolds.

3.1 Extending the Model

Our model builds upon the Pooling Game that was introduced in [10], keeping the same building blocks, but extending it where deemed necessary. A lot of the definitions that we use in this section are taken straight from [10], but our additions to the model (most prominently the introduction of pool splitting as a strategy) necessitated some adjustments in the notation of the game's components. Note that we do not make any changes whatsoever to the reward scheme itself (described in 2.3.3.1), but rather in the way we allow players to move, granted that such a reward scheme is in place.

3.1.1 Formal description

As mentioned in paragraph 2.1.1, a game can be described by a set of players, their possible strategies and the utility functions they use to evaluate each combination of strategies. In this game, we have a set of players $N = \{1, \dots n\}$, alternatively referred to as *stakeholders*, as they represent the asset holders in the case of Proof-of-Stake blockchains. Each player i is allocated some stake s_i and a cost c_i that determines the operational cost of player i's first pool, should they choose to open one (the cost of sub-

sequent pools will be defined below). It holds that $\sum_{i=1}^{n} s_i = 1$, meaning that the total stake of the system (before the distribution of the rewards) is equal to 1 (alternatively we can consider s_i to be player i's *relative stake*). We also set $R = 1$, meaning that the total rewards that can be distributed to the players are equal to 1. Note at this point that for convenience, we use absolute values for the rewards and the costs, contrary to the stake, which is in line with the model in [10].

Strategies: We define the strategy S_i of player i as the following quadruple:

$$S_i = (\xi_i, \mathbf{m_i}, \lambda_\mathbf{i}, \mathbf{a_i})$$

where

- $\xi_i \in \mathbb{N}$ is the number of pools that player i wishes to operate. (Note that we can then easily calculate the total number of pools in the system as $\xi = \sum_{i=1}^{n} \xi_i$).
- $\mathbf{m_i} = (m_{i,1}, \ldots, m_{i,\xi_i})$ with $m_{i,j} \in [0,1]$ are the margins that the player imposes to their ξ pools.
- $\lambda_\mathbf{i} = (\lambda_{i,1}, \ldots, \lambda_{i,\xi_i})$ with $\lambda_{i,j} \geq 0$ and $\sum_{j=1}^{\xi} \lambda_{i,j} \leq s_i$ are the pledges that the player commits to their pools.
- $\mathbf{a_i} = (\mathbf{a_{i,1}}, \ldots, \mathbf{a_{i,n}})$ describes the allocation of player i's stake to the pools of players $1, \ldots, n$, where $\mathbf{a_{i,j}} = (a_{i,j_1}, \ldots, a_{i,j_{\xi_j}})$ describes the allocation of player i's stake to the ξ_j different pools of player j. It holds that $\sum_{j=1}^{n} \sum_{k=1}^{\xi_j} a_{i,j_k} \leq s_i$, meaning that each player can allocate part or all of their stake (but obviously no more than that) to the pools of the system.

  If $\xi_i > 0$ then $a_{i,i_k} = \lambda_{i,k}$ for $k \in \{1, \ldots, \xi_i\}$ else $\mathbf{a_{i,i}}$ is a null vector (as is every $\mathbf{a_{i,j}}$ when $\xi_j = 0$ for $j \in \{1, \ldots, n\}$).

Note that this strategy definition allows for a player to both operate a pool and delegate stake to other pools at the same time. As a reminder, the pooling game transpires over several rounds, so the above definition of a strategy represents the player's strategy during one round of the game.

A pool that is owned by player j is denoted by π_{j_k}, $k \in [1, \xi_i]$, and the stake of the pool is $\sigma_{j_k} = \sum_{i=1}^{n} a_{i,j_k}$. Remember that if $\sigma_{j_k} \geq \beta$, namely if the pool's stake exceeds the reward scheme's saturation threshold, then pool π_{j_k} is called *saturated* (or also *oversaturated* if the inequality is strict).

Utility functions: A natural choice for the utility function of the players would be the reward they receive from the protocol, based on their strategy, minus any costs they have to bear. Brünjes et al refer to that value as the *myopic utility* of a player, as it only takes into consideration the current state of a pool and does not look ahead to what the pool could accomplish in the future. To incorporate a more far-sighted element in the logic of the players, they go one step further and define the *non-myopic utility* of a player, which in principle is the same as the myopic utility (rewards minus costs), but calculates the rewards of a pool based on its expected future stake, instead of its current one [10]. Note that using this type of utility function assumes that rationality is common knowledge among the players of the game (as defined in 2.1.2).

To estimate the future (non-myopic) stake of a pool, they define the following notions, which we borrow for our model. Note that, for simplicity, in this section we keep the definitions as they are, with only one index to refer to a pool's properties (e.g. λ_j instead of λ_{j_k} for the pledge of a pool owned by player j), but they can all be trivially extended to match our multi-pool-strategy notation from the previous paragraph.

- The *potential profit* of a pool π_j with pledge λ_j and cost c_j:

$$P(\lambda_j, c_j) = r(\beta, \lambda_j) - c_j \tag{3.1}$$

  represents the highest profit that a pool with this pledge and operational cost can yield (basically the profit it would get at saturation). Note that r represents the reward function that was described in formula 2.1.

- The *desirability* of a pool π_j:

$$D_j = \begin{cases} (1 - m_j) \cdot P(\lambda_j, c_j) & \text{if } P(\lambda_j, c_) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

  represents the maximum profits that the delegators could get from this pool. Note that this value depends on the margin m_j of π_j, so by choosing lower margins, pool owners make their pools more desirable and vice versa.

- The *rank* of a pool is then defined based on the desirabilities of all pools, so that the pool with the best (highest) desirability is assigned rank 1, the second best is assigned rank 2, and so on.

Then, the *non-myopic stake* of a pool π_j with current stake σ_j, pledge λ_j, and rank rank_j is defined as follows:

$$\sigma_j^{NM} = \begin{cases} max(\beta, \sigma_j) & \text{if } \text{rank}_j \leq k \\ \lambda_j & \text{otherwise} \end{cases} \tag{3.3}$$

This means that the top k pools of the system are expected to become saturated, while the rest are expected to end up only with their owner's pledge, as delegators would choose the higher-ranked pools over them.

Having defined the non-myopic stake of a pool, we can now proceed to the definition of the non-myopic utility of a player with regard to that pool. Specifically, if pool π_j has current stake σ_j, cost c_j, margin m_j, pledge λ_j and rank rank_j, then:

Each delegator i that contributes stake $a_{i,j}$ to π_j has an expected payoff of:

$$u_{i,j} = \begin{cases} 0 & \text{if } \lambda_j = 0 \text{ (inactive pool)} \\ \max((1-m_j)\cdot(r(\beta,\lambda_j)-c_j)\cdot\frac{a_{i,j}}{\sigma_j^{NM}},\ 0) & \text{if } \text{rank}_j \leq k \text{ and } \lambda_j \neq 0 \\ \max((1-m_j)\cdot(r(\lambda_j+a_{i,j},\lambda_j)-c_j)\cdot\frac{a_{i,j}}{\lambda_j+a_{i,j}},\ 0) & \text{otherwise} \end{cases} \tag{3.4}$$

And the owner of pool π_j, who contributes pledge λ_j to the pool, can calculate their payoff from π_j as:

$$u_{j,j} = \begin{cases} 0 & \text{if } \lambda_j = 0 \text{ (inactive pool)} \\ r(\sigma_j^{NM},\lambda_j)-c_j & \text{if } r(\sigma_j^{NM},\lambda_j) < c_j \text{ and } \lambda_j \neq 0 \\ (r(\sigma_j^{NM},\lambda_j)-c_j)\cdot(m_j+(1-m_j)\cdot\frac{\lambda_j}{\sigma_j^{NM}}) & \text{otherwise} \end{cases} \tag{3.5}$$

The total utility of a player i is then calculated as the sum of the individual utilities from the pools they participate in: $u_i = \sum_{j=1}^{n} u_{i,j}$. Note that 3.4 does not yield negative values, while 3.5 does, which represents the fact that, unlike pool operators, delegators never risk having any losses.

While the previous approach dictated that all players use the non-myopic utility as a driver for their decisions [10], we add heterogeneity to the model, by allocating different utility functions to different sets of players. Specifically, we consider the partition of the players into three populations:

- The *non-myopic players* $N_{NM} = \{1, \ldots n_1\}$, whose goal is to maximise their non-myopic utility, as it was introduced in [10] and described by equations 3.4 and 3.5 above.

- The *myopic players* $N_M = \{n_1 + 1, \dots n_2\}$, whose goal is to maximise their myopic utility, which is the same as the non-myopic one, but uses only the current stake of the pools instead of their expected future stake.

- The *abstainers* $N_A = \{n_2 + 1, \dots n\}$, who stay inactive throughout the entire gameplay. In practice, that means that if player i is an abstainer, then their strategy is always: $S_i = (0, \mathbf{0}, \mathbf{0}, \mathbf{0})$. Alternatively, we can say that these players are using a constant function $u = c$, $c \in \mathbb{R}$ as their utility function.

It holds that $N_{NM} \cup N_M \cup N_A = N$ and $N_{NM} \cap N_M \cap N_A = \varnothing$, meaning that each player necessarily belongs to one and only one of the above populations.

Game configuration: After defining the players' strategies and utility functions, we augment the model, to account for concepts that were introduced in Chapter 2, such as the bounded rationality or inertia that characterises agents in such a setting as ours.

For the active players $N' = N_{NM} \cup N_M$, we introduce an *inertia ratio* ρ, which is taken into consideration when determining the utility of a player's currently established move. Specifically, if player i played strategy S_i during round q, then during round $q+1$, i will calculate the final utility of strategy S_i as $(1+\rho) \cdot u(S_i)$, where $u(S_i)$ is the value that is returned from player i's utility function for strategy S_i. In practice, this means that in order for a new strategy S_i' to be chosen, it is not only required to yield higher utility than the previous one, but it needs to yield *sufficiently higher* utility, i.e. $u(S_i') > (1+\rho) \cdot u(S_i)$.

This is in line with the concepts of decision inertia and inertial equilibria that were introduced in paragraph 2.1.3.1 and embody the fact that real-life actors are averse to changing their chosen course of action. For the sake of simplicity, we chose to have the same value of inertia ratio for all the players, but it would also make sense to tailor this value to the different players, to account for the heterogeneity of the real-world agents. Alternatively, we could use switching costs (i.e. scalars instead of ratios) like they did in [19], but ultimately we expect that the big picture in all cases would be the same.

On a related note, we also impose a restriction on the pool owners, so that after they open a new pool, they are not allowed to close it (or any other of their pools if they have more) for a certain number of rounds. This is to account for the fact that a real-life pool operator would give their pool a chance to grow and would not abandon it after the first hurdle. In practice, a variable is included in the model that determines

the minimum number of steps that a player needs to keep a pool; let us denote this by δ.

Therefore, if at step $q-1$, player i's strategy includes $\xi_i = x$ pools and at step q it includes $\xi_i = x+1$ pools (meaning that i opened a new pool), then at step $q+\delta$ and at every step in between, it should hold that $\xi_i \geq x+1$. Note that the player is still allowed to create new pools during these rounds or change their strategy in any way other than reducing the number of pools (for.example by changing the margin of their pools).

Another variable that is added to the model is that of a *fixed cost per pool*, γ. The need for this variable arises when we consider that we allow players to open multiple pools each. As mentioned above, each player i is assigned an individual cost value c_i that corresponds to the operational costs that owning a pool would potentially incur on i. However, it would not be wise to assume that a second pool operated by i would cost an additional c_i, as there are costs that can be shared between multiple pools, such as any specialised equipment. Therefore, the extra pool would be expected to only add to the total cost by a small factor.

This is a common concept in business and economics, first introduced by Adam Smith and commonly known by the term *economies of scale*, implying that it is possible to produce more units of a good or service with lower costs per unit on average [29]. We can, thus, view the pool operation activity as an economy of scale and assume that c_i represents the first big investment and that every additional pool only increases the total cost by γ. It is important to choose the value of γ carefully, so that it is lower than any player i's starting cost c_i, as operating two or more pools is assumed to cost less per pool for every player than operating one pool would.

3.2 Playing out the Game

Based on the game that was defined above, we create an engine that can simulate the way the game could transpire under certain (configurable) circumstances.

We simulate the game as a *better-response dynamics* interplay, where at every round a player looks for a move that is better (in terms of expected utility) than their current one, but not necessarily the "best". This is different from the approach used in [10], where in each round the players looked for their best response to the other players' strategies, meaning that they searched for the strategy that yielded the highest

possible utility for them.

From a computational perspective, looking for a "good-enough" solution is a lot less expensive than looking for the best, therefore our simulation is expected to have improved performance over the previous one. Additionally, we claim that this will not come at the expense of the simulation's accuracy, but rather, based on the Behavioural Economics concepts that were described in section 2.2, we expect that the chosen method of "satisficing" in terms of the players' strategies will yield more realistic results than optimising would. On the Game-Theory side, it has been shown that, under certain circumstances, better-response dynamics can even find equilibria that best-response dynamics fail to [30], so it is deemed to be a very promising approach.

3.2.1 Determining delegation moves

In order to choose which pool(s) to delegate their stake to, players look at the state of the system and calculate the *desirability* of the existing pools (refer to 3.1.1 for definition). As mentioned earlier, the desirability of a pool represents the fraction of the profits that is left for the delegators, so it makes sense for players to want to join pools with high desirability. Therefore, the prospective delegators use a desirability-related heuristic to choose where to delegate their stake to, namely they rank the pools based on their desirability and choose to delegate their stake to the pool(s) with the highest desirability that are not already saturated.

We note here that the definition of desirability given by formula 3.2 makes use of a pool's *potential* profit, which is a value related to long-term thinking and, thence, a myopic delegator can not be expected to calculate it. Instead, myopic players use a corresponding *myopic desirability*, which is calculated in the same way as the desirability but using a pool's current profit instead of the potential one. This value expresses how desirable a pool appears at that very moment in time and is subject to many fluctuations, as the stake distribution changes.

3.2.2 Determining pool moves

The players use a combination of heuristic and greedy methods to determine potential pool-operator moves, namely strategies where a player i plans to have ξ_i pools, with $\xi_i \geq 1$.

Potential for pool: First of all, it should be noted that not all players "go through the trouble" of determining a suitable pool-operator strategy. For some, it is rather obvious that their combination of personal stake and operational cost is not competitive enough to even try to open a pool. To determine whether they have "potential for a pool", players once again resort to the useful metric of desirability. Essentially, they pose the question to themselves of whether they have the capacity to create a pool that could potentially be more desirable than any of the currently open pools.

To that end, the player calculates their potential profit —which is equal to the desirability that the player's pool would have if its margin was set to 0— and concludes that their pool would stand a chance within the current landscape only if this value is higher than the desirability of at least one active pool. The logic behind that is that the player in that case seeks to "steal" the delegators from that "inferior" pool by opening one with better prospects. If the desirabilities of all existing pools are higher, then the player only considers delegation moves for that round.

In the special case that the current pools are not enough to cover the total stake of the system without getting oversaturated —with the current parameter setting of $\beta = \frac{1}{k}$ this is equivalent to having less than k active pools— then having a positive potential profit is a necessary and sufficient condition for a player to have "potential for a pool", as it means that there is stake that is forced to remain undelegated or delegated to an oversaturated pool that yields suboptimal rewards, therefore there is incentive for it to be moved to a new pool, granted that the pool's potential rewards are sufficient to offset its costs.

Note that the the above process takes place only for players who do not have any pools and are considering opening their first one. Players who already own pools use more "greedy" methods to decide whether to "expand" their operations, as we will specify in the paragraph below.

Number of pools: As has been made clear in several sections of this work, our extended model of the Pooling Game allows players to operate an arbitrary number of pools each. This has the unfortunate side effect of increasing the complexity of a move, in the sense that a potential pool owner now has an additional decision to make, namely determine the appropriate number of pools to operate. To avoid excessive complexity and to keep the process more realistic, we only have the players assess a limited number of options, the specifics of which depend on their personal situation.

If a player i, who does not own a pool, determines that they have potential to

open one, then they form strategy S_i with $\xi_i = 1$, meaning that they only consider running one pool at that point. In practice, this corresponds to "testing the water" before making any larger commitments. On the other hand, player j, who is already a pool owner, considers at each step the possibilities of increasing by one, decreasing by one —if applicable— or maintaining the same number of pools; independent of their situation, all pool owners also evaluate the possibility of "going back to the base case" of operating one pool. The final decision is then taken based on the expected utilities of the above options.

It should be noted that in some cases, namely when they have "recently" opened a new pool, the pool owner has restricted options, in the sense that they are not allowed to close any pools. Therefore, those players would consider only the possibilities of keeping the same number of pools or increasing it by one.

Margin: The pool owner calculates the margin of each potential pool separately, so it is possible for them to run multiple pools with different margins. To determine a favourable value for a pool's margin, the player performs a "local binary search" around the current margin of the pool m, trying out values above and below it, namely in the range $[0, 2m]$, calculating the relevant expected utility, and continuing the search in the most promising direction. Since the margins are in a continuous space, we only allow for a certain number of calculations (currently set to 5), to guarantee a result in finite time. For new pools, the search is performed around an initial margin $m = 0.25$.

Pledge: If, at any given point, player i with stake s_i decides to operate ξ_i pools, then they split their stake equally as pledge among the pools, so that each pool π_{i_j} has a pledge $\lambda_{i_j} = \frac{s_i}{\xi_i}$. If $\frac{s_i}{\xi_i} > \beta$, meaning if the potential pledge is higher than the saturation point, then each pledge becomes $\lambda_{i_j} = \beta$, as it does not make any sense for the operators to include stake to their pool beyond the saturation point. If a pool owner has remaining stake even after saturating their pool(s), then they delegate the remaining $s_i - \xi_i \cdot \beta$ stake to another player's pool(s) (allowing a combination of operator and delegator moves comes very handy here, as choosing only one of them would not be realistic).

In the case that the player saturates their pools with pledge, then they do not have anything to gain by receiving delegations from other players, as they already get the highest rewards they can. On the contrary, it is possible for them to be harmed by delegators, as a malicious player could launch an "oversaturation attack" against them, by adding stake to their pool beyond the saturation point and subsequently reducing

the pool's (and the pool owner's) rewards.

To avoid this phenomenon, those stakeholders are allowed to set their pool's status to *private*, meaning that no other player has the authority to allocate their stake to them. In practice, the simulation considers a pool to be private if and only if it has pledge $\lambda \geq \beta$. Note that, by default, a private pool has its margin set to zero, to avoid unnecessary computations, as any value would yield the same result for such a pool.

At this point, it should be noted that the above procedure is followed by all active players, regardless of their characterisation as "myopic" or "non-myopic". This is because we assume that when making such an important decision as operating a pool, any rational stakeholder will at least attempt to plan ahead. Future research is welcome to find a way to lift this assumption, but for now we take it for granted, and, therefore, for this context, we do not define a separate, myopic play.

3.2.3 Simulation set up

Setting up the simulation involves initialising the n players, as well as defining the state of the system at round 0.

Initialise players

Stake: We considered two options for distributing stake among the players, namely:

- Pareto distribution: We assume that the players' stake follows a Pareto distribution, as in most cases of wealth distributions. Therefore, we create a pareto distribution with a certain (configurable) shape value and sample n values from it and allocate them to the players in random order. Note that this is the method used in the previous simulation tool [10] and the default method used here.
- Actual distribution of Cardano: We sample n values from Cardano's actual stake distribution and allocate them to the players in random order. The data was collected through a collaboration with another student, Leonidas Triantafyllou, who queried the full node of Cardano during epoch 275.

Cost: We assume that the players' costs are uniformly distributed within a certain (configurable) range. Therefore, during the set-up, we sample n values from U(min_cost, max_cost) and allocate them to the players in random order.

Initialise system state: We start with no pools in the system, meaning that at step 0 all players play the "null" strategy $S = (0, \mathbf{0}, \mathbf{0}, \mathbf{0})$.

The remaining parameters of the simulation, such as the inertia ratio γ or the fraction of myopic players $\frac{|N_M|}{n}$ are all configurable and no further assumptions were made about them. The default values of all the variables of the simulation can be seen in Figure 4.1 in the next chapter.

3.2.4 Termination criteria

In each simulation execution, we run the model until convergence or until a predefined maximum number of iterations has been reached. In each step, all players get the chance to make a move.

Convergence is defined as follows: if no changes occur in the composition of pools for a predefined number of steps, then we assume that players have no incentive to deviate from their chosen strategies and conclude that an equilibrium point has been reached. To ensure that everyone gets a chance to update their strategy in any way they wish, we impose that the number of "idle" steps needed for convergence is always higher than δ, i.e. the number of steps during which an operator might be restricted from closing their pools.

3.2.5 Simulation step

Briefly put, during a step of the simulation each player examines a handful of strategies, including delegation and pool strategies, and chooses the one that yields the highest utility, with the current strategy being given an additional benefit because of inertial forces.

In more detail, the following process takes place, which can also be seen through the flowchart of Figure 3.1:

In each round, all players are given the chance to make a move; to ensure fairness and make the process more realistic, players are always activated in random order[1]. Each player i makes their move as follows, upon their turn:

- If $i \in N_A$, i.e. if player i is an abstainer, then they do not engage in any strategic thinking and they end their turn right away, letting the simulation proceed with

[1]To get reproducible results, we set a seed number for generating pseudo-randomness, but to ensure that the results are not biased towards this input, we later run the simulation with multiple seeds during some experiments.

the next player. Otherwise, they consider changing their strategy as outlined in the steps below.

- The player calculates the expected utility from their current strategy (myopic or non-myopic utility, depending on the type of player) and augments it by the specified inertia factor, so that this move has an advantage when compared with the utilities of other moves, as dleness is preferred over moves that offer minimal improvement.
- The player forms a potential delegation strategy:
  - They look at the current pools (besides their own if they have any) and rank them based on their desirability. They consider an allocation of their stake to the pool(s) with the highest desirability that are not saturated.
  - They calculate the expected utility of this delegation strategy.
- In some cases, the player forms a set of potential operator strategies:
  - If the player is not a pool operator, then they explore the possibility of opening a new pool:
    - They examine their personal situation and the current pool landscape and they determine whether they have the potential to open a pool, based on the process that was described in paragraph 3.2.2.
    - If the player decides that opening a pool is a viable option, then they determine the parameters of their potential pool, using the processes outlined in 3.2.2 and then calculate the expected utility of this pool strategy.
  - If the player is already a pool operator, then they form an alternative operator strategy:
    - They consider up to 4 different pool strategies, using the process described in 3.2.2.
    - They calculate the expected utilities of these strategies and keep as a final candidate the highest-utility one.
- The player then compares the utility of their (up to) 3 possible strategies (keeping current status, being a pool operator or being a delegator), chooses the role that yields the highest utility and executes the respective strategy. If there is a tie

between two moves, the one that requires less effort is chosen (so current status is preferred over delegation and delegation is preferred over pool operation).

- If the player chooses to open a pool, then they are obligated to keep it for at least the predefined number of steps, δ, to give their pool(s) a chance to grow before making any further decisions.

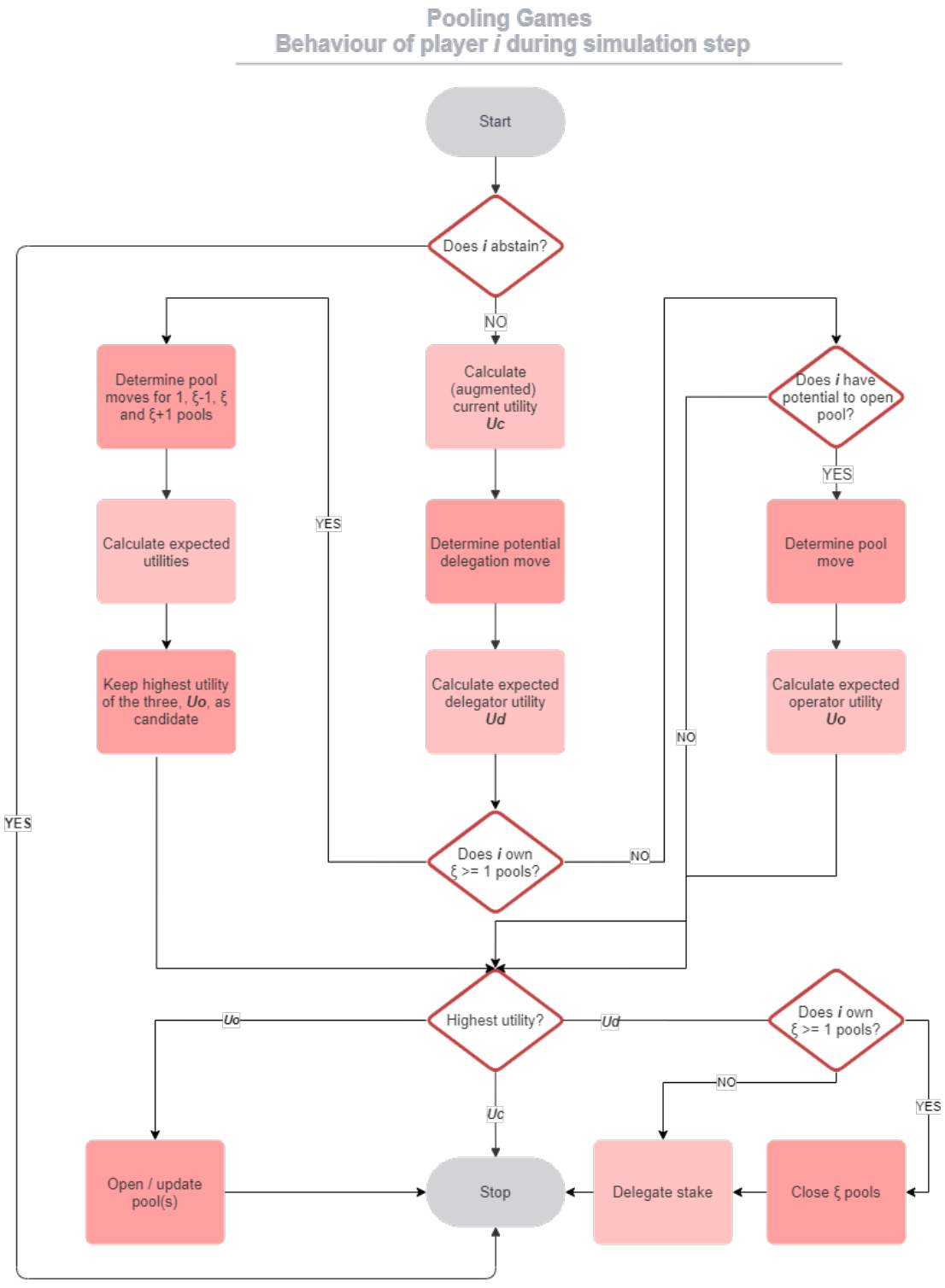


Figure 3.1: Flowchart of player activity during a step of the simulation. Note that the δ parameter has been omitted for tje purposes of this illustration.

3.3 Comparison with previous approach

Ultimately, our objective was to extend the model of the Pooling Game that was introduced in [10], in order to make it more realistic, and to develop a new simulation engine for that model, capable of playing out any scenario that may be considered useful.

The main addition of our model is the adjustment in the definition of the players' strategies, which enables each player to operate an arbitrary number of pools. In the previous model, it was assumed that players could only open one pool each, which is not in line with the "rules" of the real-life system.

Furthermore, this time we do not impose an upper bound on the players' wealth by truncating the stake distribution. Instead, to solve the issue of oversaturated pools that could arise when such "rich" stakeholders are in place, we allow them to open multiple pools or to combine pool operation and delegation moves (depending on their personal situation, one or the other might be more favourable). This highlights another difference of our simulation engine, as the previous one did not support combined moves of operating a pool and delegating to other pools (even though the theoretical model did).

We also integrated heterogeneity in the model, by allowing players in the same instance of the game to adhere to different utility functions, compared to the previous one which used the non-myopic utility for all players. Abstention was also introduced as a possible strategy of the players, to account for the fact that in real life 100% participation is never the case. On top of that, we formalised the notion of inertia and included it as a variable in the model, along with other variables, such as the minimum number of steps that a pool operator has to keep their pool.

It should be noted at this point that the previous simulation also included a similar logic, namely restricting what the operators can do after they open a pool, but their restrictions were more strict, as they didn't allow operators to even update the margin of their pool (they were basically completely blocked for a number of steps). By relaxing this restriction and allowing the operators to do anything besides closing pools during that time period, we capture the fact that opening a pool is a commitment, but one that does not remove all freedom from the operator (for example changing the pool's margin is an easy process and does not influence the longevity of the pool).

Last, but by far not least, we removed the unrealistic assumptions with regard to the players' knowledge about the personal situation of each other. In the previous

simulation, all players were expected to know the exact stake and operational cost of every other player, information which they then used to determine their optimal strategy. However, one can not expect such "omniscience" from real-life actors (even if they analyse the blockchain to get data about the players' stake, predicting the personal costs of other players would be a very tough task). In our case, players only use information that in real life is publicly available, such as the stake and cost of pools that are already in operation. A key element that allowed us to reduce the players' assumed knowledge was the substitution of "best-response" with "better-response" dynamics, namely heuristically looking for a sufficiently good solution, instead of trying to determine the absolute best one.

The experiments of next section will reveal the impact of these changes on the final outcome of the simulation.

Chapter 4

Experiments & Results

In this chapter, we present the different configurations that the simulation was run with and the results that were produced. We aggregate this information to interpret how the different parameter values or assumptions can influence the course of the game and what the repercussions for the real-life systems are.

4.1 Baseline configuration & assumptions

We define the baseline scenario for the simulation, which can be seen in Figure 4.1. In all other experiments, if we do not specify some of these values, it can be taken for granted that the values used are the ones mentioned in that figure.

It should be noted at this point that certain parameter values were set arbitrarily, in lack of any relevant research to support the choice —an example of such a parameter is the inertia ratio. For that reason, although our goal is to align the model with the real-life system as much as possible, we must acknowledge the possibility of diverging from it, in case the values we test turn out not to be representative.

As mentioned in the previous chapter, we assume that the costs of the players are uniformly distributed in a certain range and that their stake follows a Pareto distribution.

4.1.1 Baseline Analysis

Our baseline simulation converges to the desired number of pools, 10, which is in line with the theoretical analysis and the experiments conducted in [10]. These pools are generally owned by the stakeholders with the highest potential profits, as we can see

Baseline parameter values for the experiments

Game parameters

- $\mathbf{n = 100}$: we have 100 stakeholders in the system (same as in [10]).
- **cost** $\mathbf{c_i} \in [\mathbf{0.001}, \mathbf{0.002}]$: the players' costs are uniformly distributed in this range. These values are also taken from [10].
- **common cost** $\gamma = \mathbf{0.0001}$: for every additional pool a player opens, they will bear this additional cost.
- **inertia ratio** $\rho = \mathbf{0.1}$: a player switches from strategy S_A to S_B only if the increase in utility is substantial, namely: $u_{S_B} > 1.1 \cdot u_{S_A}$.
- **minimum steps to keep pool** $\delta = 10$: after opening a pool, an operator has to refrain from closing any pools for at least 10 rounds.
- **abstaining fraction** $\frac{|\mathbf{N_M}|}{|\mathbf{N}|} = \mathbf{0.1}$: 10% of the players abstain completely from the game.
- **myopic fraction** $\frac{|\mathbf{N_M}|}{|\mathbf{N}|} = \mathbf{0.1}$: 10% of the players are short-sighted (use myopic utility and desirability as a driver for their decisions).The remaining 80% are non-myopic.

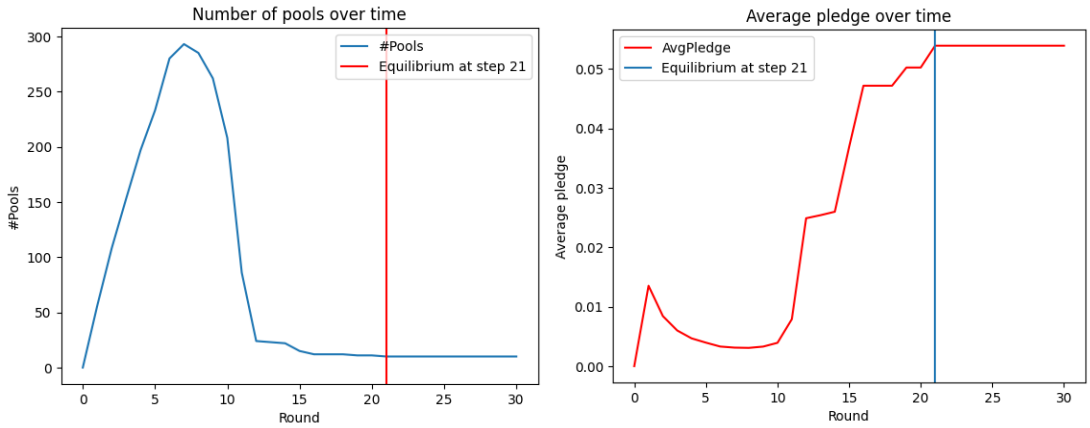
Reward scheme parameters

- $\mathbf{k = 10}$: the desired number of pools is 10.
- $\alpha = \mathbf{0.3}$: the value that is currently used in the real-life system of Cardano, to trade between efficiency and Sybil resilience.

Simulation parameters

- **Pareto shape value** $\alpha = \mathbf{2}$: the players' stake is sampled from a Pareto distribution with this parameter. This is in accordance with the experiments in [10].
- **pool splitting = on**: players are allowed to operate multiple pools each.
- **max iterations = 1000**: if the players do not reach an equilibrium after each having the chance to move for 1000 times, then the simulation stops.
- **random seed = 42**: the pseudorandom generator is initialised with this seed, for reproducibility purposes.

Figure 4.1: Parameter values used in the baseline scenario, grouped in categories.



(a) Number of open pools per round.

(b) Average pledged stake per round.

Figure 4.2: Results from the baseline execution.

on Table 4.1, with minor exceptions that can be attributed to abstention, myopic play or other reasons. We also note that, in general, the potential profit rank is closer to the owner's stake rank than to their cost rank, which can be attributed to the relatively high value of the α parameter, which assigns higher rewards to higher-pledged pools. There are four private pools, owned by the "richest" players, who were able to saturate them with pledge ($\lambda = 0.1$), while the rest are open to delegations, with pledge equal to their owner's stake and small —but non-zero— margins.

We observe that at the equilibrium there is no pool splitting behaviour, other than the case of the richest player, who could saturate two pools with their pledge. However, taking a look at Figure 4.2a, we can understand that most, if not all, pool owners engaged in pool splitting at some point through the game, as the number of pools grows beyond the number of players for some rounds. From Figure 4.2b, we can see that the average pledged stake of the system generally grows over time, which is consistent with the fact that pool splitters close down their extra pools and concentrate their entire stake into a single pool.

Another observation from Table 4.1 is that, although the majority of the pools are saturated, there are a few exceptions, meaning that the total stake that is controlled by pools is less than the total stake of the system. This happens for two reasons. The obvious one is the abstaining fraction of the players (10% in our baseline execution), as the stake of those players remains undelegated throughout the gameplay. The less obvious reason is the inertia ratio in combination with the "rich" stakeholders: when a pool operator saturates their pool with pledge and still has remaining stake, then the rational thing to do is delegate the remaining stake to another pool, which is what these

Pool id	Pool stake	Margin	Pledge	Owner stake	Owner stake rank	Pool cost	Owner cost rank	Owner PP rank
8	0.1000	0.0132	0.0215	0.0215	8	0.0010	6	8
21	0.1000	0.0000	0.1000	0.1060	3	0.0011	16	1
31	0.1000	0.0000	0.1000	0.2110	1	0.0008	43	2
51	0.1000	0.0000	0.1000	0.1188	2	0.0019	92	3
57	0.1000	0.0183	0.0293	0.0293	5	0.0016	62	5
63	0.1000	0.0000	0.1000	0.2110	1	0.0008	43	2
111	0.0709	0.0082	0.0153	0.0153	13	0.0014	50	14
123	0.0471	0.0065	0.0252	0.0252	7	0.0017	72	7
170	0.1000	0.0141	0.0315	0.0315	4	0.0018	84	4
172	0.1000	0.0026	0.0161	0.0161	11	0.0011	14	9
Total	**0.9180**	-	**0.5389**	-	-	**0.0132**	-	-

Table 4.1: Results from the baseline scenario. The pools are arranged in rows based on the order in which they were created and PP stands for potential profit.

players do when first devising their strategy; however, if the pools of their choice close down, then re-delegating that stake is not a move that yields much higher utility for them compared to their current status, so they do not "bother" doing it, leaving part of their stake inactive.

It should be noted that the baseline scenario was also run with different random seeds, but all of them exhibited similar behaviour (convergence to around k —if not exactly k— pools with the properties described above).

4.2 Additional experiments

To examine the effect of the different parameters on the dynamics of the game, we run 16 additional scenarios, that differ by at least one value from the baseline (see Table 4.2 for details). In the paragraphs below, we summarise the results of these experiments and analyse the ones that were of greatest interest. We included some of these results in the appendix, where we also added aggregate results from some more experiments.

Experiment No	Differences from baseline scenario	Default values & Objectives
1	0% myopic players	Examine the effect
2	50%	of myopic play.
3	90% myopic players	*Default is 10%*
4	0% abstaining players	Examine the effect of
5	30%	abstention. *Default is 10%*
6	inertia ratio $\rho = 1\%$ & iterations = 10 000	Examine the
7	$\rho = 5\%$	effect of inertia.
8	$\rho = 15\%$	*Default is 10%*
9	$\delta = 0$	Examine the effect of restricting
10	$\delta = 20$	operators. *Default is 5*
11	pool splitting not allowed	Examine the setting of restricting operators to having one pool each.
12	n = 1000	Examine how the simulation scales. *Default is 100*
13	stake sampled from real data	Examine the effect of our assumptions. *Default involves sampling from Pareto distribution with* $\alpha = 2$.
14	k = 30	Examine the effect of the
15	$\alpha = 0.01$	reward scheme parameters.
16	$\alpha = 1$	*Defaults are 10 and 0.3*

Table 4.2: List of conducted experiments.

4.2.1 Myopic play

The first 3 experiments focus on myopic play, by adding to or subtracting myopic players from the game and observing the effects. As a reminder, myopic play implies that players look at the current state of the system (current stake of each pool, etc.) and not at its expected future state, when making decisions, such as which pool to delegate their stake to.

We expected that this behaviour could lead to less optimal final configurations of

the system, however this was not the case, as even the results from cases 2 and 3 that included high fractions of myopic players had only negligible differences from the baseline. We can interpret this in two ways: either the system is not influenced by myopic play (which would be a very positive result for its real-life equivalent) or (unfortunately more likely) our representation of myopic play is not sufficient (remember that only delegator moves can be made myopically in our simulation, while pool moves always assume a degree of far-sightedness). In defence of our simulation, we observe that in the cases considered so far, the stake rank of the players was very similar to their potential profit rank (see Table 4.1 for an example of that), which implies that the myopic thinking of choosing the pools with the highest stake would be fairly aligned with the non-myopic thinking of choosing the ones with the highest potential profit.

To get a better understanding of the effect of myopic play on the system, we believe that more work is needed, to investigate the different possibilities listed above.

4.2.2 Abstention

Experiments No 4 and 5 focus on the effect of abstention, by tweaking the fraction of the player population that remains inactive throughout the game. When no players abstain from the game (case 4), the only observed difference from the baseline is that the total stake controlled by the pools increases, approaching the total stake of the system. This was an expected result, as the previously inactive players were free to delegate their stake or even open pools of their own now.

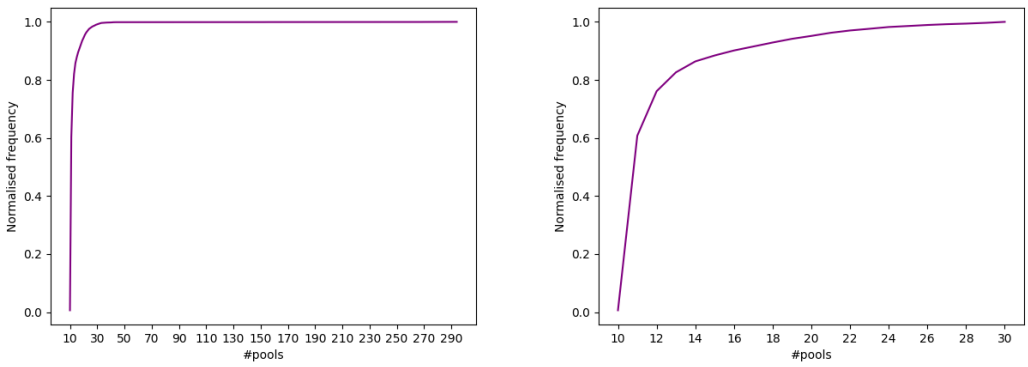
When a significant fraction of the players abstain (30% in case 5), the total stake controlled by pools is significantly lower (around 0.71 compared to the baseline's 0.92) and we observe more pool owners with lower potential profits. In addition, the total pledge of the system is decreased and the total costs are slightly increased. These results were expected, as a higher abstention rate results in higher probability of leaving well-situated players out of the game (the abstaining players are chosen at random among all players, so at the beginning of the game they all have equal probability of abstaining).

4.2.3 Inertia

Perhaps the most interesting set of experiments, cases 6, 7 and 8 revealed the significant impact of decision inertia (and subsequently of our assumptions) on the convergence of the system.

While an inertia ratio of 5% (case 6) or 15% (case 7) only yielded minimal differences from the baseline's 10% (e.g. slightly higher total stake in the first case and quicker convergence in the latter) , setting a value as low as 1% (case 6) prevented the simulation from reaching a point of equilibrium —at least with the definition of equilibrium that we have given so far.

This is because players in our game never determine their "optimal" strategy, so it is very likely that they can always make at least a very small improvement to their current strategy, which can create room for improvement in another player's strategy, and so on. However, this continuous search for a better alternative is not necessarily a bad result —in fact, one might say that it is more realistic than the absolute stability that the other experiments yielded. A lot depends though on the state of the system during this infinite play: was the system *relatively* stable (e.g. by having a consistent number of active pools) or did it keep going through major alterations?



(a) Cumulative frequencies of pool numbers.

(b) Area of interest from (a).

Figure 4.3: Results from experiment No 6. The frequencies shown are normalised, but the absolute frequencies can also be easily determined by multiplying by the number of rounds, namely 10 000.

To get a grasp on the above, we perform a statistical analysis of the simulation's output, and specifically of the time series that reveals the number of open pools during each round. Figure 4.3a depicts the cumulative frequencies of pool numbers, i.e. what percentage of rounds includes at least x pools, with x taking all the values that occurred during the simulation run. Evidently, even though the simulation did not "converge", the number of pools remained relatively stable throughout the most part. Figure 4.3b zooms into the area of highest interest, to facilitate our understanding of the situation. From there, we can see that in more than 75% of all rounds, we have 12 or less pools,

which is very close to the desired number of pools (10 for this run). We also observe that more than 90% of the time we have 17 pools or less and that more than 99% of the time, the number of pools does not exceed 30. This is a satisfying result, as it shows that, even when the majority of stakeholders are extremely opportunistic and hardly ever "stay put", the system does not diverge too much from its target state.

Our research on the subject also revealed that there exists a concept of an equilibrium that could potentially capture the results from this experiment. Grazzini and Richiardi define the notion of a *transient equilibrium*, in the family of *statistical equilibria*, for given time windows of agent-based models [31]. In simple terms, a statistical equilibrium is described by the mean value of a model's output series of interest, within a certain time window, and it is considered transient if there exists another time window where that mean value has changed. The catch is that the time series in question needs to be (weakly) stationary in order for the above to hold, so we would have to first analyse the output for stationarity, to be able to characterise any such equilibria within [32]. This analysis is outside the scope of this work, but it opens up an interesting direction for future research, namely to explore the emergent properties of the model's output, examine whether the concept of a transient equilibrium can successfully describe our data and, if yes, what that means for the real-life system.

The results we reach from these executions are critical in understanding how behavioural traits of a system's stakeholders, such as a strong tendency to stick to past decisions, can influence its eventual stability. We speculate that inertia is a determining factor in the stability of the system, so we could direct more research into measuring the real inertia ratios that people (subconsciously) use to make decisions, to see which one of our scenarios is more representative of reality.

We note that experiments 9 & 10 can also fall under the broad category of "inertia", as they represent the degree to which a pool operator is determined to keep their pool up and running, regardless of its performance. The results obtained from these executions, however, imply that this form of inertia is not as strong as the previous one, since the final configurations that were generated for the different values were almost identical to the baseline; only the speed of convergence was impacted to a meaningful degree.

4.2.4 No pool splitting

Though our theoretical model allows players to create multiple pools each, we can choose to disable this behaviour in the simulation, to observe how players would adapt.

In general, the expected behaviour arises and our results are consistent with the simulations in [10], which did not allow for pool splitting. Some small differences between this execution (No 11) and the baseline is that the total pledge of the system is a bit lower (0.48 vs 0.54) and the total operating costs are a bit higher (0.016 vs 0.013).

4.2.5 Reward scheme parameters

Brünjes et al confirmed through simulations that their model produces the expected results, namely that the system converges to k pools for an arbitrary value of k and that the value of α influences the pool properties of the final formation, while at the same time, other factors, such as the number of stakeholders in the system or the exact distribution of their stake, had no impact on the result [10]. In this paragraph, we examine whether the changes we made to the model had an effect on any of the above and we explore how pool splitting behaviour can be influenced by these parameters.

In general, we observe that the conclusions that were drawn in [10] also hold in our simulations. The execution with a higher value for k (case 14) yielded more pools than the baseline [1], while sampling the players' stake from real data instead of a Pareto distribution (case 13) had minimal impact on the final configuration and same goes for an increased number of stakeholders in the system (case 12).

The most interesting results were achieved through our last experiments (No 15 and No 16), which proved experimentally what was described in theory, namely that a higher value of α prevents pool splitting behaviour, while a low value does not. In the case of a low α value, while k remained the same as in the baseline (10 desired pools), the final configuration at the equilibrium involved 16 pools, run by a mere 4 of the players (see Table A.1 in the Appendix for details). On the other hand, the higher α prevented pool splitting behaviour, resulting in a configuration similar to the baseline, which also included a slightly increased total pledge. These results imply that the theoretical analysis of [10] was correct and that the value currently used in Cardano ($\alpha = 0.3$) is big enough to disincentivise Sybil behaviour (see also Figure A.2 in the Appendix for aggregate results from simulations with varying values of α that highlight the effect of the parameter on pool splitting behaviour).

[1] see Figure A.1 in the Appendix for more results related to k.

Chapter 5

Conclusions

5.1 Summary

In this project, we modelled the stake pool operation and delegation process of the Cardano blockchain as a game, both theoretically —by extending the model that was introduced by Brünjes et al in [10]— and empirically, through configurable simulations. We brought the model closer to the real-life system in several ways, including lifting assumptions about the "players" and not requiring excessive knowledge from them.

Our most prominent addition was the inclusion of pool splitting as a potential strategy, which lifted the previous restriction that each player could only operate one stake pool at a time. Other adjustments entailed having a heterogeneous population of players, removing excessive knowledge from them and integrating the notions of bounded rationality and decision inertia in their behaviour.

This broader model helped us examine the efficiency of Cardano's reward scheme under a diverse set of conditions, through simulations. Our experimental analysis yielded results that were in line with the ones in [10], and also confirmed theoretical claims that could not be evaluated with the previous simulation tool, such as the expectation that an increased value of the reward scheme's α parameter can effectively prevent pool splitting and Sybil behaviour among the stakeholders of the system.

5.2 Limitations & Future Work

There exist several ways in which this work can be built upon. First of all, one potential course of action is to conduct additional, structured series of experiments using

the simulation engine that was developed, to extract more insights about the emergent properties of the system in question. Due to the time constraints of the current project, a very large part of the work was dedicated to the modelling and development processes, therefore our design and analysis during the experimentation phase was not as comprehensive as it could be, possibly leaving the rigorous reader with questions about directions that were not thoroughly investigated. Another option would be to extend the simulation tool itself, for example by defining an explicit way with which players who have been characterised as "myopic" make their pool moves or by allowing players to evaluate a bigger range of strategies during each round.

A potentially more interesting direction would be to extend even more the theoretical model of the (extended) Pooling Game that we described in section 3.1. There are many ways in which this can be accomplished, such as adding more heterogeneity among the players (e.g. by introducing a new set of players that are active but are predisposed to being delegators and therefore never form pool-operation strategies) or refining our notion of inertia (e.g. by assigning different inertia ratios / switching costs to different players or different pairs of strategies of the game).

A somewhat challenging addition to the model might be the incorporation of exchange rates, to account for the fact that in real life the costs of pool operators are not measured in the same unit as the rewards that the system distributes (one of them is typically paid in fiat currency while the other in cryptocurrency). In a setting where this distinction is modelled, it would be interesting to simulate events, such as volatility or shocks in the market, and observe the impact they have on the system.

A natural extension of the model would be to make it compatible with other similar systems, for example with other Proof-of-Stake blockchains, such as Ethereum 2.0 [33] or Algorand [26]. Though outside the scope of the current work, care was taken in the development of the simulation engine, to ensure that extending it in such a way would not be troublesome.

In an effort to improve our model, without necessarily extending it, one could also attempt to "tune" its parameters, so that their values are as close as possible to their real-life equivalents. A simple example of a parameter that needs tuning is that of the inertia ratio. As we saw through our experiments in Chapter 4, different values of inertia can lead to very different results for the system, therefore it is important to understand which value represents best the decision-making process of real-life actors. We appreciate though that this is not an easy task to perform, as it requires extensive research on human behaviour.

Bibliography

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.

[2] Julie Frizzo-Barker, Peter A Chow-White, Philippa R Adams, Jennifer Mentanko, Dung Ha, and Sandy Green. Blockchain as a disruptive technology for business: A systematic review. *International Journal of Information Management*, 51:102029, 2020.

[3] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *2017 IEEE international congress on big data (BigData congress)*, pages 557–564. IEEE, 2017.

[4] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying Chang Liang, and Dong In Kim. A survey on blockchain: A game theoretical perspective. *IEEE Access*, 7, 2019.

[5] Arthur Gervais, Ghassan O. Karame, Vedran Capkun, and Srdjan Capkun. Is bitcoin a decentralized currency? *IEEE Security and Privacy*, 12, 2014.

[6] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert Van Renesse, and Emin Gün Sirer. Decentralization in bitcoin and ethereum networks. In *International Conference on Financial Cryptography and Data Security*, pages 439–457. Springer, 2018.

[7] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security*, pages 477–498. Springer, 2016.

[8] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 365–382, 2016.

[9] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security*, pages 436–454. Springer, 2014.

[10] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward sharing schemes for stake pools. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 256–275. IEEE, 2020.

[11] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In *Annual International Cryptology Conference*, pages 357–388. Springer, 2017.

[12] J v Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[13] John F Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[14] Lloyd Shapley. A value for n-person games. *Ann. Math. Study28, Contributions to the Theory of Games, ed. by HW Kuhn, and AW Tucker*, pages 307–317, 1953.

[15] John Maynard Smith. *Evolution and the Theory of Games*. Cambridge university press, 1982.

[16] Sergiu Hart. Games in extensive and strategic forms. volume 1 of *Handbook of Game Theory with Economic Applications, Chapter 2*, pages 19–40. Elsevier, 1992.

[17] Michael Maschler, Eilon Solan, and Shmuel Zamir. Game theory (translated from the hebrew by ziv hellman and edited by mike borns). *Cambridge University Press, Cambridge, pp. xxvi*, 979:4, 2013.

[18] William H Sandholm. *Population games and evolutionary dynamics*. MIT press, 2010.

[19] Basilio Gentile, Dario Paccagnan, Bolutife Ogunsola, and John Lygeros. The nash equilibrium with inertia in population games. *IEEE transactions on automatic control*, pages 1–1, 2020.

[20] Carlos Alós-Ferrer, Sabine Hügelschäfer, and Jiahui Li. Inertia and decision making. *Frontiers in psychology*, 7:169, 2016.

[21] Herbert Alexander Simon. *Models of bounded rationality: Empirically grounded economic reason*, volume 3. MIT press, 1997.

[22] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[23] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37), 2014.

[24] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V.Editors Vazirani, editors. *Introduction to Mechanism Design (for Computer Scientists)*, page 209–242. Cambridge University Press, 2007.

[25] Aggelos Kiayias. Blockchain reward sharing - a comparative systematization from first principles. Available at: `https://iohk.io/en/blog/posts/2020/11/30/blockchain-reward-sharing-a-comparative-systematization-from-first-principles/`.

[26] Jing Chen and Silvio Micali. Algorand. *arXiv preprint arXiv:1607.01341*, 2016.

[27] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[28] Pramod Viswanath and Gerui Wang. Compounding of wealth in proof-of-stake cryptocurrencies. In *Financial Cryptography and Data Security: 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18–22, 2019, Revised Selected Papers*, volume 11598, page 42. Springer Nature, 2019.

[29] Adam Smith. *The wealth of nations [1776]*, volume 11937. na, 1937.

[30] Ben Amiet, Andrea Collevecchio, and Kais Hamza. When "better" is better than "best". *Operations research letters*, 49(2):260–264, 2021.

[31] Jakob Grazzini and Matteo Richiardi. Estimation of ergodic agent-based models by simulated minimum distance. *Journal of Economic Dynamics and Control*, 51:148–165, 2015.

[32] Jakob Grazzini. Analysis of the emergent properties: Stationarity and ergodicity. *Journal of Artificial Societies and Social Simulation*, 15(2):7, 2012.

[33] Vitalik Buterin, Daniël Reijsbergen, Stefanos Leonardos, and Georgios Piliouras. Incentives in ethereum's hybrid casper protocol. *International Journal of Network Management*, 30(5):e2098, 2020.

Appendix A

Additional results

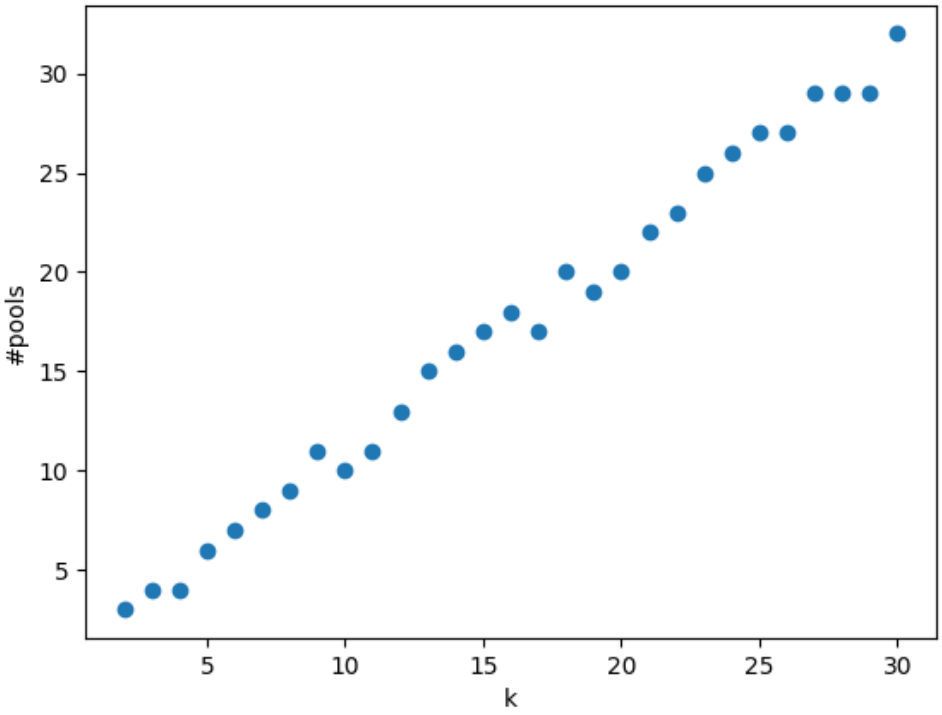


Figure A.1: Number of pools at equilibrium for different values of k.

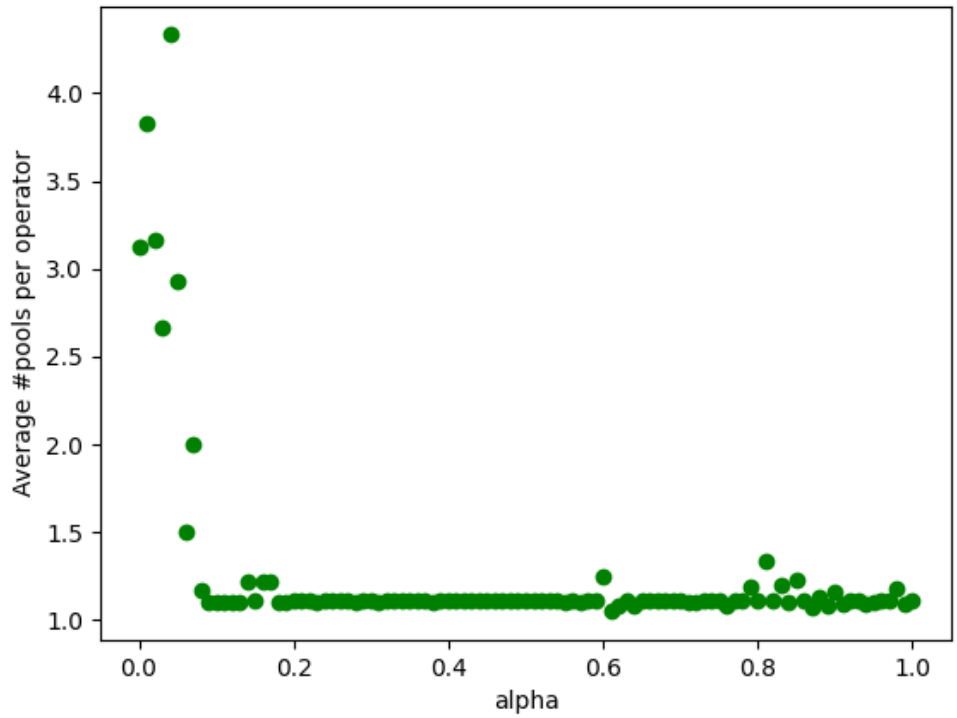


Figure A.2: Average number of pools per operator at equilibrium for different values of α.

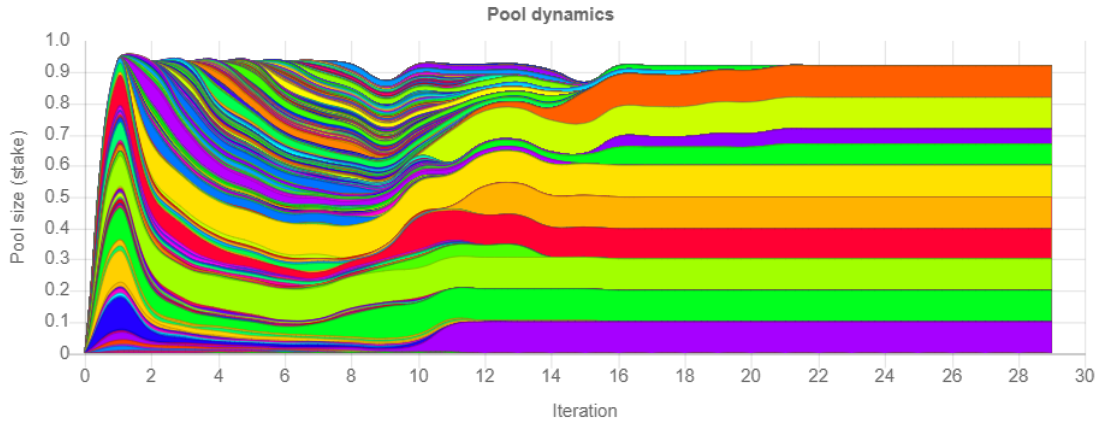


Figure A.3: Pool dynamics of the baseline simulation.

Pool id	Pool stake	Margin	Pledge	Owner stake	Owner stake rank	Pool cost	Owner cost rank	Owner PP rank
8	0.1000	0.0002	0.0036	0.0215	8	0.0003	6	3
25	0.1000	0.0000	0.0177	0.1060	3	0.0003	16	1
36	0.1000	0.0000	0.1000	0.2110	1	0.0008	43	2
47	0.1000	0.0000	0.1000	0.1188	2	0.0019	92	5
54	0.1000	0.0000	0.1000	0.2110	1	0.0008	43	2
60	0.0177	0.2228	0.0177	0.1060	3	0.0003	16	1
75	0.1000	0.0002	0.0036	0.0215	8	0.0003	6	3
79	0.0177	0.2226	0.0177	0.1060	3	0.0003	16	1
113	0.0140	0.0073	0.0140	0.0140	16	0.0012	24	16
137	0.0177	0.2206	0.0177	0.1060	3	0.0003	16	1
138	0.0168	0.0002	0.0036	0.0215	8	0.0003	6	3
228	0.1000	0.0002	0.0036	0.0215	8	0.0003	6	3
240	0.0177	0.2336	0.0177	0.1060	3	0.0003	16	1
265	0.0177	0.2266	0.0177	0.1060	3	0.0003	16	1
360	0.1000	0.0002	0.0036	0.0215	8	0.0003	6	3
530	0.0036	0.0002	0.0036	0.0215	8	0.0003	6	3

Table A.1: Results from experiment No 15. Note that all pools are run by only 4 players.