# Reward Sharing Schemes for Stake Pools

Lars Brünjes* Aggelos Kiayias† Elias Koutsoupias‡ Aikaterini-Panagiota Stouka†

Monday 30th July, 2018

Notes by Sebastien Guillemot (twitter.com/SebastienGllmt)

**Abstract**

We introduce and study reward sharing schemes that promote the fair formation of *stake pools* in collaborative projects that involve a large number of stakeholders such as the maintenance of a proof-of-stake (PoS) blockchain. Our mechanisms are parameterised by a target value for the desired number of pools. We show that by properly incentivising participants, the desired number of stake pools is a non-myopic Nash equilibrium arising from rational play. Our equilibria also exhibit an efficiency / security tradeoff via a parameter that allows them to be calibrated and include only the pools with the smallest possible cost and/or provide protection against Sybil attacks, the setting where a single stakeholder creates a large number of pools in the hopes to dominate the collaborative project. We also experimentally demonstrate the reachability of such equilibria in dynamic environments where players react to each others strategic moves over an indefinite period of interactive play.

## 1 Introduction

Consider a society of agents that have stake in a joint effort that is recorded in a ledger and want to run a *collaborative project* (which might be maintaining the ledger itself). Each player decides whether to participate directly or delegate its stake to other stakeholders. Our primary example of a collaborative project is maintaining a blockchain in a proof-of-stake (PoS) blockchain system, e.g., [13, 7]. In such situations, there is a tradeoff between efficiency and trust. Efficiency is achieved by a dictatorial solution, in which a unique delegate makes all decisions. At the other extreme, trust in the system is maximized by a direct democracy approach when every stakeholder participates in the project proportionally to their stake. We are interested in the intermediate solution of representative democracy or polyarchy [5] in which there exists a relatively large number of delegates. A direct way to achieve a fixed number $k$ of delegates is to select $k$ of them with probability proportional to their stake as typically suggested in many PoS protocols [1, 13, 7, 11, 6]. Unfortunately there are a few technical problems with this solution: first, not all participants are always present, want to, or have the capacity to act as delegates, and second, randomly selected delegates that need to run the system may be inefficient. There are also other problems with the random solution: no accountability for delegates is build into the system and therefore somebody can try to subvert the system by simply paying some selected delegates to act in a particular way the moment they become known.

Instead, in this work we put forth a novel *stake-pool* approach: we set a target $k$ of pool-leaders and then an incentives-based mechanism will provide appropriately calculated rewards to all participants to propose or support already proposed stake-pools. The mechanism allows all agents to allocate freely their stake to any pool they prefer and it provides rational incentives for agents to assign stake in a such a way so that $k$ stake pools will be formed in the end. The number $k$ of pools should be as large as technical considerations, such as a continuous network presence and efficiency, allow (for

*IOHK, lars.bruenjes@iohk.io
†University of Edinburgh & IOHK, Aggelos.Kiayias,A.stouka@ed.ac.uk
‡University of Oxford, elias.koutsoupias@cs.ox.ac.uk

instance it will be needed that any of the $k$ pool leaders can transmit a message received by most other participants in a reasonable time window for the underlying PoS protocol to operate properly).

There are certain properties that such a mechanism must satisfy which are direct requirements of the nature of the collaborative project of maintaining a blockchain, and there are desirable properties that are part of a design philosophy. As an example of the former properties, we usually do not know the number $n$ of agents, and therefore the mechanism should not rely on such knowledge. As an example of latter, a desirable property is that the formed pools have are efficient in terms of cost and they are of similar size.

A stake pool participates in the collaborative project through its leader; each participating pool leader $j$ incurs a cost $c_j$ (e.g., corresponding to the cost of running a server to execute a PoS protocol). In order to incentivize the stakeholders and pool leaders to form pools and work for the collaborative project, we introduce a *reward scheme*. In this work, we assume that there is a *fixed reward* to be distributed among all pools at each stage of the maintenance process.

When an agent declares its candidacy to become a pool leader, we assume that certain aspects of its profile, such as its stake and its operational costs, are announced to everyone else (e.g., in a PoS setting this can be achieved by signing a "pool creation" certificate with their key[1]). We want these aspects either to be verifiable, as for example is the stake committed to its own pool, or to be reported truthfully. The latter means that the mechanism must be incentive compatible.

Given the above, the setting that we consider is simple: we assume that there are $n$ agents or players with publicly known stakes $s = (s_1, \ldots, s_n)$ and costs for running a pool $c = (c_1, \ldots, c_n)$. We set up a game so that the agents are incentivized to run pools or allocate their stake to pools created by others. The game is determined by a reward scheme. A *reward scheme* determines the way by which the total reward $R$ is distributed to the pools and how the pool reward is distributed to the pool members.[2] *The central issue of this work is to determine reward schemes with desired properties.*

In this work, we focus on a simple class of reward schemes. Simplicity is an important property as complicated schemes impose extra burden to players for selecting their best strategies. One component of a reward scheme determines how the total reward is distributed to pools. We focus on the class of reward schemes that allocate reward $r(\sigma, \lambda)$ to a pool of total stake $\sigma$ and allocated pool leader stake $\lambda$, and we call $r$ the *reward function*. The other component of a reward scheme determines how the pool reward $r(\sigma, \lambda)$ is distributed to the pool leader and pool members. It makes sense that the reward for the pool leader is different from the reward for pool members to compensate the pool leader for the cost it incurs by contributing to the collaborative project as well as to incentivize them to take the initiative to form a pool. We focus on reward schemes that distribute the pool reward as follows: the pool leader gets an amount to cover its cost of running the project as well as a fraction $m_j$ of the remaining amount. The remaining is distributed among the pool members, including the pool leader, proportionally to the stake that they contributed to the pool.

Given a reward function $r$ that belongs to the above class, the players will pick their strategy that determines whether they will run a pool or not and whether they will allocate some or all of their stake to pools created by other players. Natural questions about these games are: Do they have pure equilibria? Can we compute them efficiently? Do the best-response dynamics converge fast?

Suppose that the reward margins $\vec{m}$ and the allocated pool leaders stake $\vec{\lambda}$ to their potential pools are fixed. $\vec{m}, \vec{\lambda}$ determine a game in which the players decide to create or not their own pool with stake $\lambda_j$ and/or how to allocate their remaining stake among the available pools.

An important observation here is that the notion of Nash equilibrium does not seem appropriate for this type of games. The reason is that all Nash equilibria (if they exist) will have reward margins $m_j = 1$ for a simple reason: once the other players select their strategies and in particular the allocation of their stake, the best response of a pool leader is to increase its margin as much as possible. However, a choice of such a high reward margin will not happen in reality since it will not result in a competitive

[1] In fact multiple stakeholders may be allowed to back the same pool creation certificate by signing it.

[2] Expectedly, rewards will not be guaranteed but can only be claimed by pools that are operating properly in terms of doing their part in the collaborative project.

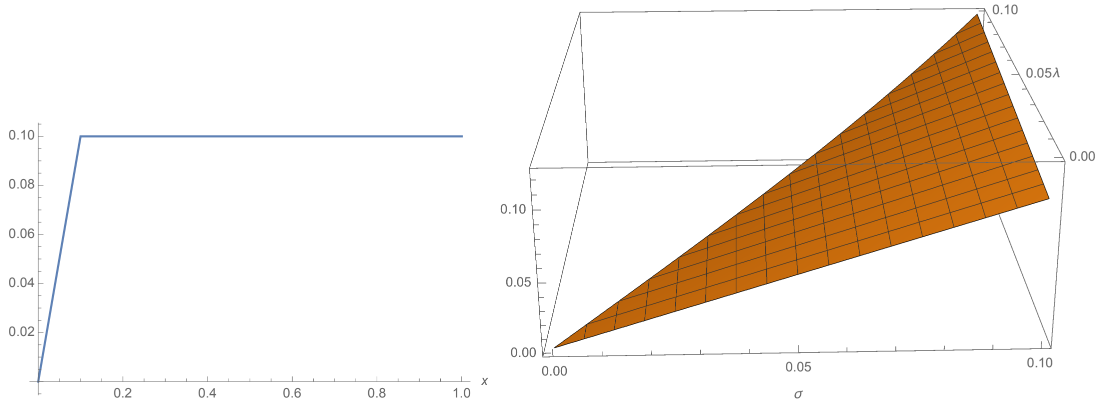



Figure 1: Reward function for $z_0 = 1/10$ with $a_0 = 0$ (left) and $a_0 = 1/4$ (right).

stake pool that will be able to attract member stakeholders. Similar situations occur in other games, such as the Cournot competition [10]. The appropriate framework for such games is to consider *non-myopic* playing, i.e., consider equilibria in a setting where utility is defined in a non-myopic fashion, accounting for the effects that a certain move of a player will incur anticipating a strategic response by the other players.

**Our results.** Let us first consider a reward function $r(\sigma,\lambda) = r(\sigma,0)$ that depends only on the total stake $\sigma$ of the pool (note we assume without loss of generality that the stake of any agent or pool belongs to $(0,1)$ and represents the fraction of the total stake controlled by the specific entity) and it is independent of the stake $\lambda$ of the pool leader. The natural choice is to select $r(\sigma,0)$ proportional to $\sigma$, which has the nice property that it rewards all players proportionally to their stake. However, it leads to dictatorial equilibria in which a single pool is created. Although there may be other equilibria, it should be clear that this reward scheme cannot achieve the goal of creating approximately $k$ pools because it is independent of the target $k$. Instead, a simple modification of this reward scheme goes a long way in meeting this target. Consider the modification

$$r(\sigma,0) \sim \min\{\sigma, z_0\},$$

where $z_0$ is a constant and $\sim$ indicates proportionality with a multiplier that guarantees that the total reward is sufficient to pay all pools[3] (see Figure 1). If we call a pool *saturated* when its total stake $\sigma$ is at least $z_0$, we can say that the reward function discourages oversaturated pools. By setting $z_0 = 1/k$, this reward scheme seems to provide the right incentives to create pools of size up to $z_0 = 1/k$, which naturally leads to $k$ pools of equal size. However, this picture is to a large extend misleading because the usual myopic best-response dynamics create a single pool instead of $k$, because even with this reward function, for a pool member a saturated pool is preferable to a pool whose reward is mainly used to cover the cost of its leader. The good news is that, as we will show, dynamics of non-myopic best response achieves the goal by leading to an equilibrium of $k$ pools of equal size, given a reasonable definition of an appropriate non-myopic notion of utility.

To evaluate the quality of a reward scheme, we should compare the resulting equilibrium with an optimal solution. An optimal solution when all participants act honestly and selfishly is to have $k$ pools of equal size that are run by pool leaders of minimal cost. This would make the system efficient, in both computational and economic sense. But besides efficiency, we want the system to withstand attacks from some players that try to run many pools, even at a loss. A player that wants to subvert the system may try to attack the system by creating multiple identities and declaring minimal cost (in what sometimes is referred to as a Sybil attack [8]). We can prevent this attack, to a large degree, if we can guarantee that players can attract stake from other players only if they commit substantial stake

[3]A smooth function that approximates this reward function may be preferable to improve the dynamics of convergence to equilibrium.

to their own pool. This is precisely the reason for considering reward functions that depend, besides the total stake of the pool, on the stake of the pool leader.

Ideally, we want the pools to be created by the players ranked best according to $a_0 s_j - c_j$ (a linear combination of their stake $s_j$ and their cost $c_j$), where $a_0$ is a nonnegative parameter that can be adapted to trade between efficiency and Sybil security. By selecting $a_0 = 0$ we get the most efficient solution, and on the other extreme, by selecting a very large $a_0$, we can obtain a potentially inefficient solution in which the pool leaders are the $k$ "wealthiest" while Sybil attacks are totally de-incentivised.

The objective is to design a reward scheme that provides incentives to obtain an equilibrium that compares well with the above optimal solution. On the other hand, we feel that it is important that the mechanism is not unnecessarily restrictive and all players have the "right" to become pool leaders.

The natural way to accomodate this in our scheme, would be to use the above reward function but apply it to $\sigma + a_0\lambda$, a weighted sum of the total pool stake $\sigma$ and the allocated pool leader stake $\lambda$. With this in mind, the reward function becomes $r(\sigma,\lambda) \sim \min\{\sigma, z_0\} + \overset{\text{(Unbounded)}}{\underline{a_0\lambda}}$. Again this reward function goes some way towards meeting the objective but the best response dynamics, even non-myopic best response dynamics, do not lead to equilibria that resemble the optimal solution and in particular, it may create pools of very large size. The reason is that the influence of the stake $\lambda$ of the pool leader when a pool has not enough stake is very significant. Given that the ideal size of the pool is $z_0$, one way to alleviate this effect is to change the influence factor $a_0$ to be proportional to the stake that the pool has already attracted, that is to change the influence factor to $a_0' = a_0 \frac{\sigma-\lambda}{z_0}$. But this creates the minor problem that the influence factor is not the same for all pools, a very desirable property in order to evaluate the threat of Sibyl-like attacks. The final touch in the reward function which resolves this issue is to make the influence of the stake of the pool leader on the factor $a_0'$ to disappear when the pool has the desired size of $z_0$. We therefore update the influence factor and the reward function to

$\Rightarrow \sigma' = z_0$

$$(1) \qquad r(\sigma,\lambda) \sim \sigma' + a_0'\lambda \text{ where } \sigma' = \min\{\sigma, z_0\} \text{ and } a_0' = a_0 \frac{\sigma' - \lambda\cdot(1-\sigma'/z_0)}{z_0} = a_0\left(\frac{z_0 - \cancel{\lambda\left(1-\frac{z_0}{z_0}\right)}}{z_0}\right) = a_0$$

For this reward function we define a class of joint strategies that we prove (Theorem 1) that they are *non-myopic Nash equilibria* and satisfy the following: (i) exactly $k$ pools are created, (ii) the players with the highest *potential profit* are those that form the pools, and in the case $a_0 = 0$, those are exactly the ones with the smallest cost. For the definition of non-myopic play we consider ranking the pools according to the rewards they are to distribute to their members, assuming they are saturated (a quantity we call pool *desirability*). Then, non-myopic utility assigns a payoff according to the strategy of a player calculating the rewards of the $k$ most desirable pools as if they are saturated. This type of play ensures that players "think ahead" and anticipate the moves of the other players.

A downside of the above is that non-myopic moves may increase the stake of a saturated pool above $z_0$, in case a pool stakeholder decreases its margin below the value in the Nash equilibrium. This motivates us to consider a more refined version of our game, where players act in two stages: in the first stage, they determine the margins and stake to be allocated in case they become pool leaders, while in the second stage, they determine the best possible allocation given the first stage strategy. We define an (approximate) Nash-equilibrium of the outer game to ensure that no player is worse off in any of the equilibria of the inner game. We validate our results for perfect strategies by providing an equivalent class of approximate Nash equilibria of the outer game (Theorem 3) that have the two desired properties (Theorem 2, Lemmas 4,5 ): (i) all the equilibria of the inner games determined by these joint strategies form $k$ pools of the same size $z_0 = 1/k$. (ii) the pool leaders of these pools are the players with the lowest cost when $a_0 = 0$ and the players with the most profitable combination of stake and cost $s \cdot a_0 \cdot \boxed{\frac{R}{1+a_0}}^{*} - c$, when $a_0 \neq 0$ where $R$ are the total rewards.

**Dynamics.** How the players are going to play such a relatively complicated game? We consider non-myopic dynamics best-respond to each other moves in succession. To do so, the players compute the desirability of each pool, which is the answer to the following question of the players: "if I allocate a small stake $x$ to pool $j$, how much do I expect to gain?". In other words, the desirability is the marginal reward of pool $j$ provided that it will become a successful pool and obtain stake $z_0$. A non-myopic player then assumes that each of the $k$ most desirable pools will increase in size to become

* Used to constrain the sum of all rewards to R

saturated and the remaining pools will end up with the stake of their pool leader, and allocates its stake accordingly. For pool leaders the situation is similar, but they have also to compute their margin. To do so, they calculate the maximum possible margin that still allows them to be one of the $k$ most desirable pools. The question then is whether these dynamics converge? how fast? and to which equilibrium? We provide experimental evidence that under reasonable assumptions of the stake distribution (for example, Pareto distribution) and of the cost distribution (for example, uniform distribution in an interval), the dynamics converge quickly to our Nash equilibrium that has $k$ saturated pools and the characteristic that all pools are formed by the players that are ranked best according to the most profitable combination of stake and cost $s \cdot a_0 \cdot \frac{R}{1+a_0} - c$ as predicted by the theoretical analysis.

**Equilibria and incentive compatibility.** The reward scheme has a Nash equilibrium in which the reward is distributed fairly among all stakeholders, except for pool leaders that get an additionally gain (Proposition 3). A nice property of this additional gain is that, all else being equal, it increases by at most $\delta x$ whenever the pool leader's cost decreases by $\delta x$. This means that the reward scheme is incentive compatible: no player will benefit by lying about its cost.

**Related work.** Our setting has certain similarities to cooperative game theory in which coalitions of players have a value. In our setting the players have weights (stake) and they are allowed to split it into various coalitions (pools). Our objective is to have a given number of equal-weight coalitions, which contrasts with the typical question in cooperative game theory on how the values of the coalitions are distributed (e.g., core or Shapley value) in such a way that the grand coalition is stable [18]. Actually, the games that we study are variants of congestion games with rewards on a network of parallel links, one for every potential pool. The reward on each link is determined by the reward function, which essentially determines an atomic splittable congestion game. But unlike simple atomic splittable congestion games [16], our games have different reward for pool leaders and for pool members. There are two main research directions for such games: whether they have unique equilibria and how to efficiently compute them [3]. Regarding the question of unique inner equilibria the most relevant paper to our inner game is [17] (but see also [19, 2]) which shows that under general continuity and convexity assumptions, games on parallel links have unique equilibria. However, the conditions on convexity do not meet our design objectives and they don't seem to be useful in our setting.

Our work is related to two aspects of delegation games, which are games that address the benefits and other strategic considerations for players delegating to someone else to play a game on their behalf, such as owners of firms hiring CEO's to run a company. The first aspect is somewhat superficially related to this work in pool formation the pool members delegate their power to pool leaders. The second aspect which is much more relevant to our approach is that delegation changes the utility of the players (for example, by considering "credible threats" [20, 21]) or creates a two-stage game [23, 9, 22]. A typical two-stage delegation game is non-myopic Cournot competition [10] in which in the outer game the firms (players) decide whether to be profit-maximizers or revenue-maximizers, while in the inner game they play a simple Cournot competition [15]. Unlike our case, the inner Cournot competition has a simple unique equilibrium which defines a simple two-stage game.

Another research area that is relevant to this work is mechanism design, because participants may have an incentive not to reveal their true parameters, e.g., the cost for running a pool [16, 24].

With respect to PoS blockchain systems, a different and notable approach to stake pools is to use the stake as voting power to elect a number of representatives, all of equal power, as in delegated PoS (DPoS) [14]; for example, the cryptocurrency EOS [12] has 21 representatives (called block producers). This type of scheme differs from ours in that (i) the incentives of voters are not taken into account thus issues of low voter participation are not addressed, (ii) elected representatives, despite getting equal power, are rewarded according to votes received; this inconsistency between representation and power may result in a relatively small fraction of stake controlling the system (e.g., currently EOS delegates representing just 2.2% of stakeholders are sufficient to halt the system,[2] which ideally could withstand a ratio less than 1/3), (iii) it may leave a large fraction of stakeholders without representation (e.g., in

EOS, currently, only 8% of total stake is represented by the 21 delegates[4]). Yet another alternative to stake pools is that of Casper [4], where players can propose themselves as "validators" committing some of their stake as collateral. The committed stake can be "slashed" in case of a proven protocol deviation. This type of scheme differs from ours in that (i) stakeholders wishing to abstain from protocol maintenance operations have no prescribed way of contributing to the mechanism (as in the case of voting in DPoS or joining a stake pool in our setting), (ii) a small fraction of stake may end up controlling the system while at the same time leaving a lot of stake without engaging in the protocol operation; this is because substantial barriers may be imposed in becoming a validator (e.g., in the EIP proposal for Casper[5] it is suggested that 1500 ETH will be the minimum deposit, which, at the time of writing is about $\$700K$); this can make it infeasible for many parties to engage directly; on the other hand reducing this threshold drastically may make the entry barrier too low and hence still allow a small amount of stake to control the system. As a separate point, it is worth noting that for both the above approaches there is no known game theoretic analysis that establishes a similar result to the one presented herein, i.e., that the mechanism can provably lead to a Nash equilibrium with desirable decentralisation characteristics that include a high number of protocol actors and Sybil attack resilience.

## 2 The model and objectives

**Notation - game**

- *$n$, number of players.*
- *$R$, total reward.*
- *$s_i$, stake of player $i$. It holds $\sum_{i=1}^{n} s_i = 1$.*
- *$c_i$, cost of player $i$ to form a pool $\pi_i$.*
- *$m_i$, reward margin of pool $\pi_i$.*
- *$\lambda_i$, stake that player $i$ will commit if he activates his own pool $\pi_i$.*
- *$\vec{a}_i = (a_{i,1}, ... a_{i,n})$, allocation of player's $i$ stake. $\sum_j a_{i,j} \leq s_i$ .*
- *$\sigma_j$, stake of pool $\pi_j$: $\sigma_j = \sum_{i=1}^{n} a_{i,j}$. We denote the vector of pool stakes by $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$. Pools can have zero stake.*
- *$r(\sigma, \lambda)$, reward of a pool with total stake $\sigma$ and allocated pool leader stake $\lambda$. It holds $\sum_j r(\sigma_j, a_{j,j}) \leq R$.*
- *Potential profit of a saturated pool with allocated pool leader stake $\lambda$ and cost $c$, $P(\lambda, c) = r(z_0, \lambda) - c$.*
- *We order the players according to $P(s_i, c_i)$. Player $i$ is the player with the $i_{th}$ highest $P(s_i, c_i)$.*
- *$\vec{S}^{(\vec{m},\vec{\lambda})} = (\vec{a}_i)_{i=1}^{n}$, joint strategy regarding allocation given $(\vec{m}, \vec{\lambda})$. $a_{i,i} \in \{0, \lambda_i\}$ and $S_i^{(\vec{m},\vec{\lambda})} = \vec{a}_i$.*
- *$u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$, utility that player $i$ gets from pool $\pi_j$.*
- *$u_i(\vec{S}^{(\vec{m},\vec{\lambda})})$, total utility of player $i$: $u_i(\vec{S}^{(\vec{m},\vec{\lambda})}) = \sum_{j=1}^{n} u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$.*

[4]Statistics extracted from `http://eos.dapptools.info/#/block-producers` on July 27th, 2018.
[5]See `https://eips.ethereum.org/EIPS/eip-1011`.

### 2.1 The general setting and objectives

There are $n$ stakeholders (aka players) with stakes $s = (s_1, \ldots, s_n)$ and costs $c = (c_1, \ldots, c_n)$. We assume that these parameters are fixed and known by everyone[6].

The players run a *collaborative project* (e.g., maintain a blockchain) and each player decides whether to participate directly or delegate his stake to other stakeholders. The total stake that is delegated to stakeholder $j$ forms a *pool*; we will call such a pool $\pi_j$, indexed by its *pool leader* $j$, and we will denote by $\sigma_j$ the total stake delegated to this pool by all players, including the pool-leader $j$.

The pools participate in the collaborative project through their leaders and we assume that this participation makes each participating pool leader $j$ to incur cost $c_j$. To incentivize the stakeholders and pool leaders to form pools and work for the collaborative project, we introduce a *reward scheme*. We assume that there is a *fixed reward* $R$ to be distributed among all pools. A *reward scheme* determines the way by which the reward $R$ is distributed to the pools and pool members, and *the central issue of this work is to determine reward schemes with desired properties.* It makes sense that the reward for pool leaders will be different than the reward of pool members to compensate them for the cost they incur by contributing to the collaborative project and perhaps incentivize them to take the initiative to form a pool.

We assume that the stakeholders want to maximize their own utility and that there are *no externalities*, i.e., outside factors that affect their behavior. This is an important point because in general there are other issues at play that affect the reward of each pool, such as mining games. Here we make the simplifying assumption that these issues play no role in the pool formation game.

*Our objective is to incentivize the stakeholders to form approximately $k$ pools, where $k$ is a given integer.* We further want no pool to have a disproportionally large size, so that no group can exert disproportionally large influence.

We summarize the model here. Formal definitions of the concepts appear later.

- We set up an appropriate reward scheme to incentivize the stakeholders to form approximately $k$ pools. Pools are formed with stakes $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, some of them empty.
- The reward scheme distributes a total fixed amount $R$ to the pools according to their stake $\sigma_i$ and the stake of their pool leader $a_{i,i}$. In particular pool $\pi_i$ gets reward $r(\sigma_i, a_{i,i})$ with $\sum_i r(\sigma_i, a_{i,i}) \leq R$. Note that we don't have to distribute the whole amount $R$.
- This defines a game for the stakeholders whose strategies consist of two parts: (i) their profit margin $m_i$ as potential pool leaders and the stake $\lambda_i$ that they commit to their potential pool, and (ii) their delegation of stake, which is an allocation of their stake to pools.
- The reward $r(\sigma_i, a_{i,i})$ of each pool $i$ is shared by its pool leader and its stakeholders. The pool leader first gets an amount $c_i^- = \min(c_i, r(\sigma_i, a_{i,i}))$. to cover the cost for running the pool, and a fraction $m_i$ of the remaining amount $(r(\sigma_i, a_{i,i}) - c_i^-)$, as compensation for running the pool. The rest $(1 - m_i) \cdot (r(\sigma_i, a_{i,i}) - c_i^-)$ is distributed to the stakeholders of the pool, including the pool leader, *proportionally to their contributed stake*. This leaves open interesting research questions regarding other approaches for sharing the reward within each pool, such as marginal contribution and bargaining value.

**Reward Schemes.** A reward scheme is in general a function $r : [0,1]^2 \to \mathbb{R}_{\geq 0}$ that takes the stake of a pool and the stake of the pool leader allocated to this pool and returns the payment for this pool so that: $\sum_i r(\sigma_i, a_{i,i}) \leq R$.

It should be clear that for appropriate values of the parameters, there is incentive for the stakeholders to form pools so that they can share the cost. Ideally, we want to find a reward function that, at equilibrium, it leads to the creation of the desired number of almost equal-stake pools independently of (i) number of players (ii) the distribution of stake and costs (iii) the degree of concurrency in selecting a strategy (iv) more forward-looking play, that is, non-myopic play as opposed to best-response.

[6] We revisit the issue of parameters in Appendix A.

For example, a player who considers creating a new pool can estimate how much stake he can attract, given the current distribution of stakes and pools.

This seems like an impossible task[7], so we have to settle for solutions that achieve the above goals approximately under some natural assumptions about the distribution of stake and costs and about the equilibria selection dynamics.

In simple terms, the reward function $r$ essentially has to satisfy two properties: it should be increasing fast for small values to incentivize players to join together in pools to share the cost, and it should be constant for large values to discourage the creation of large pools or equivalently to incentivize the breakup of large pools into smaller pools.

## 2.2 Definition of the game

We make the assumption that every player can be the leader of only one pool and each player has stake at most $z_0 = 1/k$. Players with stake more than $z_0$ or wishing to open more than one pool can be thought of as a set of distinct players, see also Section 5 on Sybil attacks. Below, we will use the notation: $(x)^+ = \max(0,x)$, and $[n] = \{1,\ldots,n\}$.

**Definition 1** (Strategy of a player)**.** The strategy of a player $i$ has two parts:

- $(m_i, \lambda_i)$, where $m_i \in [0,1]$ is the margin and $\lambda_i$ the stake that player $i$ will commit if he activates his own pool.

  (set of pools)
- $S_i^{(\vec{m},\vec{\lambda})} = \vec{a}_i^{(\vec{m},\vec{\lambda})}$ that is the allocation of player $i$' stake given $(\vec{m},\vec{\lambda})$. When the $(\vec{m},\vec{\lambda})$ can be inferred from the context we will use $\vec{a}_i$ for simplicity. $a_{i,j} \in [0,1]$ denotes the stake that player $i$ allocates to pool $\pi_j$ so that the total allocated stake is $\sum_{j=1}^n a_{i,j} \le s_i$. This allows for stake $s_i - \sum_{j=1}^n a_{i,j}$ of the player to remain unallocated. In addition $a_{i,i}^{(\vec{m},\vec{\lambda})} \in \{0, \lambda_i\}$.

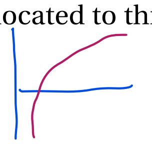
We describe the strategy of a player in two parts to be consistent also with our model of inner-outer game that we will describe later.

**Definition 2** (Pools)**.** Given a joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$, the stake allocated to a pool $\pi_j$ is denoted by $\sigma_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, or simply $\sigma_j$ for a less cluttered notation. A pool $\pi_j$ is called *active* when player $j$ allocates non-zero stake to it, that is, $a_{j,j} = \lambda_j > 0$. Note that only player $j$ can activate pool $\pi_j$. If a pool $\pi_j$ is active its stake is $\sigma_j = \sum_{i=1}^n a_{i,j}$, otherwise we assume that $\sigma_j = 0$. A pool is called *saturated* when its stake is at least $z_0$ .

The restriction that only player $j$ can activate pool $\pi_j$, by allocating non-zero sate to it, is necessary to prevent other players to force player $j$ pay the cost $c_j$ of operating the pool without his consent.

**Definition 3** (Pool Reward scheme)**.** We define a reward scheme $r$ as a function $r : [0,1]^2 \to \mathbb{R}_{\ge 0}$ that takes input the stake $\sigma_i$ of a pool $\pi_i$ and the stake $a_{i,i}$ of the pool leader allocated to this pool and returns the total reward for this pool, as well as it satisfies the following:

Note

- $\sum_{i=1}^n r(\sigma_i, a_{i,i}) \le R$, where $R$ the total rewards. $*$ $\frac{\sigma - c}{\sigma}$  $r(\sigma,\lambda) \sim \sigma$

- $r(0,0) = 0$. This means that an inactive pool will take zero rewards. Recall that we consider that the stake of an inactive pool is zero.

- $*$ $\frac{d[(r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma} > 0$, when $\sigma < z_0$. This means that the reward function is increasing fast for small values of total stake to incentivize players to join together in pools to share the cost. Recall that the reward of a pool member with stake $s$ of a pool with total stake $\sigma$, margin $m$ and pool leader's stake $\lambda$ is $(1-m)\cdot(r(\sigma,\lambda)-c)\cdot\frac{s}{\sigma}$. adds consts wrt derivative

[7]Actually, here is a simple reward scheme that achieves all goals: give no reward to the pools, unless there are exactly $k$ equal-stake pools, in which case each pool gets reward $R/k$. However, we are interested in reward schemes that can lead to a good Nash equilibrium starting at the state in which all players play no-participation and following natural symmetric, almost myopic dynamics, such as repeatedly having a random player playing best-response.

$\forall i \;\; r(\sigma_i, a_{ii}) \geqslant 0$

- $\forall \lambda \; r(\sigma, \lambda) = r(z_0, \lambda)$ when $\sigma > z_0$. This means that the reward function is constant for large values of total stake to discourage the creation of large pools.

### 2.3 Non myopic utility

Recall that the strategy of player $i$ is either to become a pool leader with reward margin $m_i$ by committing stake $\lambda_i$ and/or to delegate his stake to pools.

We first point out that the notion of Nash equilibrium for the game defined by the set of these strategies does not match our intuitive notion of stability. Indeed, suppose that we have reached a Nash equilibrium in this game, that is, a set of strategies from which no player has an incentive to deviate unilaterally. But there is an obvious problem with this definition of equilibrium: at a Nash equilibrium, all reward margins are 1. This is so, because by the definition of the Nash equilibrium the other players will keep their current strategy and the best response of a pool leader is to select the maximum possible reward margin. There are two problems here: first we definitely don't want the reward margins to be $1$, and second, the outcome is not expected to be a stable solution anyway. If all margins are $1$, a non-myopic player who is not a pool leader can start a new pool with smaller reward margin which will attract enough stake to make it profitable. We propose to study the non-myopic utility for the players that result from the following consideration. Players do not consider myopic best responses but non-myopic best responses. Specifically, *a player computes their utility using the estimated final size of their pool instead of the current size of the pool.* The estimated final size is either the stake that pool leader has allocated to this pool or the size of a saturated pool. The latter is used when the pool is currently ranked to be among the most desirable pools and the former when the pool is currently lower.

A non-myopic player that considers where to allocate their stake, wants to rank the pools with respect to the estimated reward at the Nash equilibrium. But this reward is not well-defined because the Nash equilibrium depends on the decisions of the other players. It makes sense then to use a crude ranking of the pools. Such a ranking can be based on the following thinking: "An unsaturated pool where I will place my stake will also be preferred by other like-minded players if it has relatively low margin and cost so the pool will become saturated. So, I will assume that the stake of the pool is actually $z_0$. On the other hand, if a pool has relatively high margin and cost will not grow and will lose also its members as other unsaturated pools offer better combination of margin and cost. This motivates the following ranking of pools:

**Definition 4** (Desirability and Potential Profit)**.** The potential profit of a saturated pool with allocated pool leader stake $\lambda$ and cost $c$ is $P(\lambda, c) = r(z_0, \lambda) - c$. Given a joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$, we define the desirability of a pool $\pi_j$

$$D_j(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} (1-m_j)P(\lambda_j, c_j) & \text{if } r(z_0, \lambda_j) - c_j \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{2}$$

The following definition abstracts the potential profit of a saturated pool.

Note that the desirability of a pool depends on its margin, the stake of the pool leader allocated to this pool and its cost. Not on $\sigma$

**Definition 5** (Ranking)**.** Given a joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$, $\text{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ of a pool $\pi_j$ is the ranking of the desirability $D_j(\vec{S}^{(\vec{m},\vec{\lambda})})$. The maximum desirability gets rank 1, the second maximum desirability gets rank 2, etc. Again to get a less cluttered notation, we will write $\text{rank}_j$ instead of $\text{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ whenever the joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ can be inferred from the context. Ties break according to the potential profit, specifically higher potential profit pools will be ranked higher; for convenience we assume that all potential profit values are distinct. The $k$ most desirable pools will be these ones with rank smaller or equal to $k$.

Given a ranking, we define the non-myopic stake of a pool to be either the stake allocated by the pool leader or the size of a saturated pool. The later is used when the pool is currently ranked to be among the most desirable pools and the former when the pool is currently lower.

**Definition 6** (Non-myopic stake ). Define the non-myopic stake of pool $\pi_j$ by

$$\sigma_j^{\mathrm{NM}}(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} \max(z_0, \sigma_j) & \text{if } \mathrm{rank}_j \le k \quad \text{saturated} \\ a_{j,j} & \text{otherwise.} \quad \text{just leader} \end{cases} \tag{3}$$

To simplify the notation we set $\sigma_j^{\mathrm{NM}}$ instead of $\sigma_j^{\mathrm{NM}}(\vec{S}^{(\vec{m},\vec{\lambda})})$, $\sigma_j$ instead of $\sigma_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, $\mathrm{rank}_j$ instead of $\mathrm{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ and $a_{j,j}$ instead of $a_{j,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$.

**Definition 7** (Utility). The utility $u_i(\vec{S}^{(\vec{m},\vec{\lambda})})$ of player $i$ from being a member of pool $\pi_j$ with non myopic stake $\sigma_j^{\mathrm{NM}}$ is

$$u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} 0 & \text{if } \pi_j \text{ is inactive } (a_{j,j} = 0) \\ (1 - m_j)\left(r(z_0, \lambda_j) - c_j\right)^+ \frac{a_{i,j}}{\sigma_j^{\mathrm{NM}}} & \text{else if } \mathrm{rank}_j \le k \quad \text{saturated} \\ (1 - m_j)\left(r(\lambda_j + a_{i,j}, \lambda_j) - c_j\right)^+ \frac{a_{i,j}}{\lambda_j + a_{i,j}} & \text{otherwise.} \quad \text{Just you \& leader} \end{cases} \tag{4}$$

The utility $u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ that the pool leader $j$ gets from pool $\pi_j$ is

$$u_{j,j}(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} 0 & \text{if } \pi_j \text{ is inactive} \\ r(\sigma_j^{\mathrm{NM}}, \lambda_j) - c_j & \text{else if } r(\sigma_j^{\mathrm{NM}}, \lambda_j) - c_j < 0 \quad \text{(run at loss)} \\ (r(\sigma_j^{\mathrm{NM}}, \lambda_j) - c_j)\left(m_j + \underbrace{(1 - m_j)\frac{\lambda_j}{\sigma_j^{\mathrm{NM}}}}_{\text{member of own pool}}\right) & \text{otherwise} \end{cases} \tag{5}$$

The utility of player $i$ is the sum of the utilities coming from all pools in which he participates as a pool leader or a pool member: $u_i(\vec{S}^{(\vec{m},\vec{\lambda})}) = \sum_{j=1}^{n} u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$.

## 3 The proposed reward scheme and its properties

Given our target parameter $k$, we define the reward function $r_k : [0,1]^2 \to \mathbb{R}_{\ge 0}$ of a pool $\pi$ with stake $\sigma$ and pool leader's allocated stake $\lambda$ as follows:

$$r_k(\sigma, \lambda) = \overbrace{\frac{R}{1 + a_0}}^{(1)} \cdot [\overbrace{\sigma'}^{(2)} + \lambda' \cdot a_0 \cdot \frac{\sigma' - \lambda' \cdot (1 - \sigma'/z_0)}{z_0}],$$

(1) Concrete ratio
(2) Not unbounded

where $\lambda' = \min\{\lambda, z_0\}$, $\sigma' = \min\{\sigma, z_0\}$ and $z_0, a_0$ are fixed parameters. A natural choice is $z_0 = 1/k$, where $k$ is the target of number of pools. For simplicity we will write $r$ instead of $r_k$.

We have: $a_0 \in [0, \infty)$, $(k \in \mathbb{N})$, $(k < n)$ and $R \in \mathbb{R}$. Note that the total rewards $R$ and $a_0$ should be selected such as it holds also $P(s_{k+1}, c_{k+1}) > 0$. (Incentive for new pool to join competition)

The next proposition shows that the proposed function is a suitable reward scheme.

**Proposition 1.** *The function $r(\cdot,\cdot)$ satisfies the properties of a pool reward scheme, cf. Definition 3.*

*Proof.* Follows easily as follows. (1) $\forall i \; r(\sigma_i, a_{i,i}) \ge 0$ , as $a'_{i,i} \le \sigma'_i$. (2) $\sum_{i=1}^{n} r(\sigma_i, a_{i,i}) \le R$, as $\frac{\sigma'_i - a'_{i,i} \cdot \frac{(z_0 - \sigma')}{z_0}}{z_0} \le 1$ and $\sum_{i=1}^{n} [\sigma_i + a_{i,i} \cdot a_0] = \sum_{i=1}^{n} \sigma_i + a_0 \cdot \sum_{i=1}^{n} a_{i,i} \le 1 + a_0$. (3) when $\sigma < z_0$ it holds: $\frac{d[r(\sigma,\lambda) - c) \cdot \frac{1}{\sigma}]}{d\sigma} > 0$. (4) $\forall \lambda \; r(\sigma, \lambda) = r(z_0, \lambda)$, when $\sigma > z_0$ because we have $\sigma' = \min\{\sigma, z_0\}$. (5) $r(0,0) = 0$, which means that an inactive pool $\pi_i$ takes zero reward. Recall that when $a_{i,i} = 0$ we consider also that $\sigma_i = 0$. □

### 3.1 Equilibria Analysis

The following lemma is very useful and its proof follows directly from the definition of the reward function.

**Lemma 1.** *The quantity* $(r(x,s_j)-c_j)/x$ *as a function of* $x$ *is increasing in* $[0,z_0)$ *and, if it is positive, decreasing in* $(z_0,\infty)$. *Its maximum is achieved at* $x=z_0$.

Since $x$ term continues to increase

The following lemma gives an upper bound on the utility of pool members. We will give an equilibrium that matches this upper bound.

**Lemma 2.** *In every joint strategy in which some player* $j$ *is not a pool leader, its utility is at most* $\max_l D_l\cdot(s_j/z_0)$, *where* $\max_l D_l$ *is the maximum desirability among all players.*

*Proof.* It suffices to show that player $j$ gets at most $D_l\frac{a_{j,l}}{z_0}$ from every pool $l$. The lemma follows directly from this by summing for all $l$: $\sum_l D_l\frac{a_{j,l}}{z_0}\le\max_l D_l\sum_l\frac{a_{j,l}}{z_0}=\max_l D_l\frac{s_j}{z_0}$.

The argument that for every pool $l$, player $j$ gets at most $D_l\frac{a_{j,l}}{z_0}$ follows directly from the definition of the utility of pool members when we consider the two cases depending on whether $\text{rank}_l$ is at most $k$ and more than $k$.

Specifically, when $\text{rank}_l\le k$, by the definition of the utility of pool members, the utility to player $j$ from pool $l$ is $D_l a_{j,l}/\sigma_l^{NM}\le D_l a_{j,l}/z_0$.

When $\text{rank}_l>k$, its utility is given by

$$(1-m_l)\left(r(\lambda_l+a_{j,l},\lambda_l)-c_l\right)^+\frac{a_{j,l}}{\lambda_l+a_{j,l}}\le(1-m_l)\left(r(z_0,\lambda_l)-c_l\right)^+\frac{a_{j,l}}{z_0}=D_l\frac{a_{j,l}}{z_0},$$

where the inequality comes from Lemma 1. □

The following proposition exposes an important property of the reward mechanism.

**Proposition 2.** *At every joint strategy in which the player with the highest desirability is a pool member, the player has a better strategy of opening a pool and committing all its stake at it. Consequently at every Nash equilibrium, the player with the highest desirability prefers to be a pool leader rather than a pool member.*

*Proof.* Fix a joint strategy and let player $j$ be the player with the highest desirability. Suppose that $j$ is not a pool leader. Then by Lemma 2, its utility is at most $D_j\frac{s_j}{z_0}$. When the player deviates and opens a pool in which it is the only member, the utility will not decrease. Indeed, its utility in this case will be $D_j\frac{s_j}{z_0}+m_jP(s_j,c_j)\ge D_j\frac{s_j}{z_0}$. Therefore being a pool leader is always a better response to being a pool member. Actually as we see, it is strictly preferable when $m_j>0$. □

To simplify the notation we assume that the players are ordered in terms of potential profit, e.g., player 1 is the player with the highest potential profit.

**Definition 8** (Perfect strategies)**.** We define a class of strategies, which we will call perfect. The margins are

$P(s_j, c_j)$ 3x better

$\Rightarrow$ users get $\frac{1}{3}$ reward

$$m^*=\begin{cases}1-\frac{P(s_{k+1},c_{k+1})}{P(s_j,c_j)} & \text{when } \text{rank}_j\le k\\ 0 & \text{otherwise,}\end{cases}$$

and the allocations are such that each of the first $k$ pools has stake $z_0$.

The following proposition gives the utilities at perfect strategies and it follows directly from the definitions.

Note: $k+1$ avoids losing tie for top $k$ desirable

Note: Belong to top $k$ regardless of what others set their margin as

$(1-m)P(s_i,c_i)\frac{s_i}{z_0}$

$\Downarrow$

$P(s_{k+1},c_{k+1})$ by def 8

**Proposition 3.** *In every perfect strategy, the utilities of the players are:*

if everybody uses perfect margin

$$u_i = P(s_{k+1},c_{k+1})\frac{s_i}{z_0} + \underbrace{(P(s_i,c_i) - P(s_{k+1},c_{k+1}))^+}_{\text{leader bonus}}. \tag{6}$$

*Furthermore the desirability of the first $k+1$ players is the same and equal to $P(s_{k+1},c_{k+1})$.*

Note that all the players get a fair reward, in the sense that it is a constant $P(s_{k+1},c_{k+1})/z_0$ times their stake, with the exception of each pool leader $i$, who gets an additional reward $P(s_i,c_i) - P(s_{k+1},c_{k+1})$. This additional reward can be viewed as a bonus for the efficiency and security that the pool leader brings to the system.

We will show that every perfect strategy is a Nash equilibrium of the game with the defined utilities.

**Theorem 1.** *Every perfect strategy is a Nash equilibrium.*

*Proof.* Consider first a player $j$ with rank at most $k$. This player is a pool leader of a pool of size $z_0$. We show that none of the possible responses improves its utility:

- Suppose that the player decreases its margin. This increases its desirability so that the new rank is still one of the first $k$ ranks. Since the non-myopic stake remains the same[8], this move will decrease the utility of the player.

- Suppose that the player increases its margin. compare to def. 8 Since before the change the first $k+1$ players have the same desirability, its desirability drops and its rank becomes larger than $k$. As a result the player will be in a pool by itself and its utility can only decrease (Lemma 1).

- Suppose that the player becomes a pool member of other pools. By Lemma 2, its utility can be at most $P(s_{k+1},c_{k+1})s_j/z_0$, which is lower that its current utility by $P(s_j,c_j) - P(s_{k+1},c_{k+1})$ (by Equation 6).

We now consider a player $j$ with rank higher than $k$. Again we show that none of the possible responses improves its utility. Notice first that by changing its allocation of stake, it can only hurt its utility since some of its stake ends up in pools with stake different than $z_0$, which can only lower its utility by Lemma 1. The other alternative is that the player becomes a pool leader. Since its rank is higher than $k$, the (non-myopic) stake of the pool contains only its own stake, which by Lemma 1 is again no better than the current utility. □

In the first case of the above proof, the pool leader of a pool with stake $z_0$ decreases its margin. By the definition of the non-myopic stake, the stake of its pool remains the same. But because this pool gets higher desirability, far-sighted pool members will prefer it and its size will increase beyond $z_0$. This raises the question whether perfect strategies are stable when the players play non-myopically beyond the level captured by our non-myopic utilities. To better understand the implications of far-sighted strategies, we consider a two-stage game in the next section.

## 4 Two stage game

**Definition of the game.** In order to also capture non-myopic moves in response to pool leaders changing margin or allocation, we define a two stage game, the "inner-outer game". Similar non-myopic play has already been considered in other games, most notably in *Cournot Equilibria*, as is discussed in the introduction and related work. In this section we reuse *non-myopic utility* and *desirability* as defined in previous sections, but when a pool has not been activated in the inner game, we define its desirability to be zero. This gives us a more realistic result, because in practice only pools that have already been created will be ranked.

[8] There is a possibility that non-myopic moves will increase the stake of the pool above $z_0$. This is the motivation for considering a two stage game.

We order players by $P(s_i, c_i)$, and $i$ will denote the player with the $i_{th}$ highest value according to this ordering. We break ties in ranking in arbitrary ways, our analysis will hold for all of them. In fact, we define two games here, the *inner* game, which focuses on the allocation of stake, and the *outer* game, which focuses on the reward margins and on the stake that potential pool leaders commit to their pools. In the *outer* game, player $i$ decide on his margin $m_i$ and on how much stake $\lambda_i$ to allocate to his pool, should he decide to activate it in the inner game. So a strategy of a player $i$ in the outer game is a tuple $(m_i, \lambda_i)$ of margin and allocated stake. $(\vec{m}, \vec{\lambda})$ is a joint strategy of the outer game.

In the *inner* game, the margins $\vec{m}$ and the stakes $\vec{\lambda}$, that potential pool leaders would allocate to their pools, are given, and the strategies of the players are their allocations. So in the inner game determined by $(\vec{m}, \vec{\lambda})$, a strategy of player $i$ is $S_i^{(\vec{m},\vec{\lambda})} = \vec{a}_i$, and a joint strategy is $\vec{S}^{(\vec{m},\vec{\lambda})}$. Note that if a player $i$ decides to activates his own pool, which means $a_{i,i} > 0$, then he is committed to allocate stake $\lambda_i$ to his pool, where $\lambda_i$ is part of his strategy of the outer game. So $a_{i,i} \in \{0, \lambda_i\}$. We assume that players decide their allocation trying to maximize their non-myopic utility. Recall that we have assumed that each player can create at most one pool and that the utility that an inactive pool gives to its members is zero. Note that each joint strategy of the outer game determines one inner game.

### 4.1 Definition of equilibria for inner and outer game

**Definition 9.** A joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ is a Nash equilibrium of the inner game defined by $(\vec{m}, \vec{\lambda})$ when for every player $j$

$$u_j(S_j'^{(\vec{m},\vec{\lambda})}, \underbrace{\vec{S}_{-j}^{(\vec{m},\vec{\lambda})}}_{\text{Joint strategy w/ strategy index } j \text{ removed}}) \le u_j(\vec{S}^{(\vec{m},\vec{\lambda})}), \qquad \text{for every } S_j'^{(\vec{m},\vec{\lambda})} \ne S_j^{(\vec{m},\vec{\lambda})}. \tag{7}$$

This is the standard Nash equilibrium notion when the players try to maximize their non-myopic utility.

To define the non-myopic equilibrium of the outer game, let us temporarily assume that there is a *unique Nash equilibrium in every inner game*. Then we define the utility of player $j$ in the outer game, where players have selected joint strategy $(\vec{m}, \vec{\lambda})$, as: $u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) = u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, where $\vec{S}^{(\vec{m},\vec{\lambda})}$ is the unique equilibrium of the inner game determined by $(\vec{m}, \vec{\lambda})$. So a joint strategy $(\vec{m}, \vec{\lambda})$ is an approximate $\epsilon$-non-myopic Nash equilibrium of the outer game when for every player $j$

$$u_j^{\text{outer}}(m_j', \vec{m}_{-j}, \lambda_j', \vec{\lambda}_{-j}) \le u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) + \epsilon, \qquad \text{for every } (m_j', \lambda_j') \ne (m_j, \lambda_j). \tag{8}$$

When there are multiple equilibria in the inner game, we define $u_j^{\text{outer}}(\vec{m}, \vec{\lambda})$ as *the set* of values $u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, where $\vec{S}^{(\vec{m},\vec{\lambda})}$ is a Nash equilibrium of the inner game determined by $(\vec{m}, \vec{\lambda})$.

Let

$$u_j^{\text{outer,up}}(\vec{m}, \vec{\lambda}) = \begin{cases} \sup u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) & \text{if } u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) \ne \emptyset, \\ -\infty & \text{elsewhere.} \end{cases} \tag{9}$$

sup = least upper bound

In the same way we define:

inf = greatest lower bound

$$u_j^{\text{outer,low}}(\vec{m}, \vec{\lambda}) = \begin{cases} \inf u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) & \text{if } u_j^{\text{outer}}(\vec{m}, \vec{\lambda}) \ne \emptyset, \\ -\infty & \text{elsewhere.} \end{cases} \tag{10}$$

Note that $u_j^{\text{outer}}(\vec{m}, \vec{\lambda})$ is a non-empty bounded subset of the reals and therefore always has both supremum and infimum: Upper- and lower bounds are given by $R$ and $(-\max\{c_1, \ldots, c_n\})$ respectively.

**Definition 10.** A joint strategy $(\vec{m}, \vec{\lambda})$ is an $\epsilon$-non-myopic Nash equilibrium when for every player $j$

$$u_j^{\text{outer,up}}(m_j', \vec{m}_{-j}, \lambda_j', \vec{\lambda}_{-j}) \le u_j^{\text{outer,low}}(\vec{m}, \vec{\lambda}) + \epsilon, \qquad \text{for every } (m_j', \lambda_j') \ne (m_j, \lambda_j). \tag{11}$$

### 4.2 Analysis

Now we will describe a set of joint strategies that (i) are approximate non-myopic Nash equilibria of the outer game and (ii) have the characteristic that in the inner games defined by these joint strategies, all the equilibria form $k$ saturated pools. Recall that a pool is *saturated* when its stake is at least $z_0$. The pool leaders of these pools in these equilibria of the inner game are again the players with the highest potential profit $P(s_i, c_i)$. Note that if all players activated a pool with the same margin and their whole stake, then the $k$ pools with the highest potential profit $P(s_i, c_i)$ would give the highest utility to their members.

The intuition for how the set of margins of these joint strategies is determined is the following: The $k$ players with the highest potential profit $P(s_i, c_i)$ set the maximum margin they can, so that their pools belong to the $k$ most desirable pools (the pools with the highest desirability), no matter which margins the other players set. Note that in this model, these players want to have strictly higher desirability than player $k+1$, because in a tie in ranking, they might otherwise lose. In more detail:

Let $G = \{1, \ldots, k\}$ be the set of those $k$ players with highest potential profit, $\epsilon = P(s_k, c_k) - P(s_{k+1}, c_{k+1})$ and $\epsilon_1^* = P(s_{k+1}, c_{k+1}) - P(s_{k+2}, c_{k+2})$. By assumption $\epsilon, \epsilon_1 > 0$.

For $\epsilon'$ such that $0 < \epsilon' < \min\{\underset{(1)}{\epsilon}, \underset{(2)}{P(s_{k+1}, c_{k+1})}\ \underset{(3)}{\epsilon_1}\}$ and $\alpha$ such that $\underbrace{\frac{s_{k+1}}{z_0} < \alpha < 1}_{\text{More than just stake of leader But not saturated}}$, we define $\vec{m}^*(\epsilon', \alpha)$ as follows:

Note: From (A) $-\epsilon'(1-\alpha) = \epsilon'\alpha - \epsilon'$ which is exactly $\epsilon'$ away from $\epsilon'\alpha$ in (B). An increase by $\epsilon'$ will make $(k+1)^{th}$ pool more desirable

$$m_i^*(\epsilon', \alpha) = \begin{cases} \frac{\overbrace{P(s_i, c_i) - P(s_{k+1}, c_{k+1})}^{\text{perfect strategy}} - \epsilon' \cdot (1-\alpha)}{P(s_i, c_i)} & \text{(A) when } i \in G, \\ \frac{0 + \epsilon' \cdot \alpha}{P(s_{k+1}, c_{k+1})} & \text{(B) when } i = k+1, \\ 0 \quad \text{(Not activated)} & \text{(C) elsewhere,} \end{cases}$$

Note: To see why we need $\epsilon'\alpha$ in (B), See * on page 25

(3) In case some player $i \in G$ disolves their pool

and let $\vec{\lambda}^*$ be the vectors with $\lambda_i^* = s_i$ for $i \in [n]$.

Note that margins are well defined because $0 \le m_i^* < 1$. We prove that: (i) For each $\epsilon'$ such that $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$, the joint strategies $(\vec{m}^*(\epsilon', \alpha), \vec{\lambda}^*)_{\frac{s_{k+1}}{z_0} < \alpha < 1}$ are $\epsilon'$-non-myopic Nash equilibria of the outer game. So we prove that for every $\epsilon'$ as defined above, there is a class of joint strategies that are $\epsilon'$-non-myopic Nash equilibria of the outer game (Theorem 3). (ii) For each $\epsilon'$ such that $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ and $\alpha$ such that $\frac{s_{k+1}}{z_0} < \alpha < 1$, all the equilibria of the inner game determined by joint strategy $(\vec{m}^*(\epsilon', \alpha), \vec{\lambda}^*)$ form $k$ saturated pools (Theorem 2). (iii) In the general case (for any $a_0$), the pool leaders of the $k$ saturated pools described above are the players of $G$, which are the players with the highest potential profit $P(s, c)$. If $a_0 = 0$, the players with highest $P(s, c)$ are the players with lowest cost, so this achieves an optimum in terms of social welfare in this case, since it minimizes the costs of running the system (Theorem 2).

The formal theorems and the proofs are given in Appendix B.

(1) would allow $(k+1)^{th}$ pool to give more utility than $k^{th}$
(2) Would allow some pool not in top $k$ to give more utility than one in top $k$

## 5 Sybil Resilience

As we have ntoed in the introduction, the objective of the $a_0$ parameter is to mitigate Sybil attacks. In general, the higher $a_0$ is, the more impact the allocated stake of the pool leader has on the rewards of the pool. Given that we have the equilibrium described before where the $k$ players with the highest potential profit have created a pool, let us consider an attacker who has stake $S < 1/2$ and creates multiple identities lying about its costs with the objective to obtain $k/2$ saturated pools with stake $S/(k/2)$ each one and control the system. Suppose the cost $r \cdot c_{k/2}$ is declared for all pools with $r \in [0, 1)$. This attacker would be deterred if it holds* $: S > \frac{k}{2} \cdot \left(\lambda_{k/2} - \frac{c_{k/2}}{R} \cdot (1-r) \cdot (1 + \frac{1}{a_0})\right)$. Recall $\lambda_{k/2}$ is the stake that player $k/2$ has committed to his own pool. The relation holds because the attacker should create $k/2$ pools with higher potential profit than the pool of $k/2$ player in order to succeed.

As a result, we can choose $a_0$ appropriately to lower bound the stake of such an attacker. Note that $a_0$ and total rewards $R$ should be such that it holds also $P(s_{k+1}, c_{k+1}) > 0$.

(1) call adversary A (2) Need $P(\sigma, \lambda)$ of every pool $\in A$ > $P_{k/2}(\sigma, \lambda)$ (3) Pools saturated $\Rightarrow \sigma = z_0$ & $a_0$ undampened

$$* \therefore \frac{R}{1+a_0}\left[z_0 + \lambda_A a_0\right] - c_A > \frac{R}{1+a_0}\left[z_0 + \lambda_{1/2} a_0\right] - c_{1/2}$$

$$\therefore \left[z_0 + \left(\frac{S}{k/2}\right)a_0\right] - \frac{(1-a_0)}{R} r c_{1/2} > \left[z_0 + \lambda_{1/2} a_0\right] - \frac{(1+a_0)}{R} c_{1/2}$$

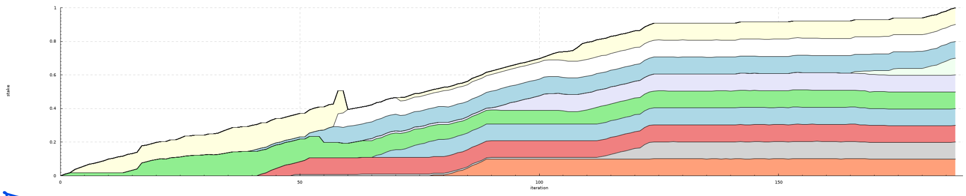


Figure 2: example dynamics ($c \in [0.001, 0.002]$, $a_0 = 0.02$)

# 6 Experiments

Complementary to our theoretical analysis, we have also simulated a slightly modified version of the game numerically[9]. Our goals were to verify the theory on the one hand and to get a feeling for the dynamics on the other hand, when players take turns and make “moves” to optimize their individual strategies.

A typical simulation result can be seen in figure 2: Time progresses from left to right, the colorful bands represent stake pools. We can see that starting from an initial state without any pools, pools are created and destroyed in a complex fashion, before we finally reach a stable configuration of $k = 10$ pools of equal size.

We simulated the game over a wide range of player costs and values for parameter $a_0$ and always reached a “nice” Nash-Equilibrium eventually (mostly even a *perfect* one in the sense of Definition 8), in full accordance with the theory.

# 7 Acknowledgements

The authors would like to thank Duncan Coutts for extensive discussions and many helpful suggestions. The second author was partially supported by H2020 project PRIVILEDGE # 780477. The third author was partially supported by the ERC Advanced Grant 321171 (ALGAME).

# References

[1] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. *CoRR*, abs/1406.5694, 2014.

[2] Umang Bhaskar, Lisa Fleischer, Darrel Hoy, and Cien-Chung Huang. Equilibria of Atomic Flow Games are not Unique. *SODA*, pages 748–757, 2009.

[3] Umang Bhaskar and Phani Raj Lolakapuri. Equilibrium computation in atomic splittable routing games with convex cost functions. *arXiv preprint arXiv:1804.10044*, 2018.

[4] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *CoRR*, abs/1710.09437, 2017.

[5] Robert Alan Dahl. *Democracy and its Critics*. Yale University Press, 1989.

[6] Phil Daian, Rafael Pass, and Elaine Shi. Snow white: Provably secure proofs of stake. Cryptology ePrint Archive, Report 2016/919, 2016. http://eprint.iacr.org/2016/919.

[7] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake protocol. Cryptology ePrint Archive, Report 2017/573, 2017. http://eprint.iacr.org/2017/573. To appear at EUROCRYPT 2018.

[9] For details and results, see Appendix C.

[8] John R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[9] C. Fershtman and K. L. Judd. Equilibrium incentives in oligopoly. *American Economic Review*, 77(5):pp. 927 – 940, 1987.

[10] Amos Fiat, Elias Koutsoupias, Katrina Ligett, Yishay Mansour, and Svetlana Olonetsky. Beyond myopic best response (in cournot competition). *Games and Economic Behavior*, 2013.

[11] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, pages 51–68, 2017.

[12] Ian Grigg. Eos, an introduction. https://eos.io/documents/EOS_An_Introduction.pdf, 2017.

[13] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. Cryptology ePrint Archive, Report 2016/889, 2016. http://eprint.iacr.org/2016/889.

[14] Dan Larimer. Delegated proof-of-stake consensus. https://bitshares.org/technology/delegated-proof-of-stake-consensus/, accessed 21.3.2018, 2018.

[15] A. Mas-Colell, J. Green, and M. D. Whinston. *Microeconomic Theory*. Oxford University Press, 1995.

[16] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[17] A Orda, R Rom, and N Shimkin. Competitive Routing in Multiuse Communication Networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.

[18] Guillermo Owen. Game theory academic press. *San Diego*, 1995.

[19] Oran Richman and Nahum Shimkin. Topological Uniqueness of the Nash Equilibrium for Selfish Routing with Atomic Users. *Mathematics of Operations Research*, 32(1):215–232, 2007.

[20] Thomas C Schelling. An essay on bargaining. *The American Economic Review*, 46(3):281–306, 1956.

[21] Thomas C Schelling. *The strategy of conflict*. Harvard university press, 1980.

[22] S. D. Sklivas. The strategic choice of managerial incentives. *RAND Journal of Economics*, 18(3):pp. 452 – 458, 1987.

[23] J. Vickers. Delegation and the theory of the firm. *Economic Journal*, 95 (Suppl.):pp. 138 – 147, 1985.

[24] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

# Appendices

## A Costs and incentive compatibility

We assumed that the costs are publicly known, but it may be that the costs for participating in the collaborative project are known only by the player. With privately known costs, the players may have a

(6) $u_i = P(s_{k+1}, c_{k+1})\frac{s_i}{z_0} + \underbrace{\left(P(s_i, c_i) - P(s_{k+1}, c_{k+1})\right)}_{\text{* Only term that changes}}$ | $P(s_i, c_i) = r(z_0, \lambda) - c$

reason to lie about their cost and the problem becomes a mechanism design problem. The objective now is to design an incentive compatible mechanism, i.e., a mechanism that gives incentives to players to declare their costs truthfully.

Let's consider the perfect Nash equilibrium of Section 3.1, in which the utilities are given by Equation 6). Suppose that a pool leader $j$ declared a different cost $\hat{c}_j$, but remained pool leader. Since $P(s_j, \hat{c}_j) - P(s_j, c_j) = c_j - \hat{c}_j$, the player will not get any benefit from lying. To see this, let $u_j(\hat{c}_j|c_j)$ denote the utility when the player declares cost $\hat{c}_j$ instead of the true cost $c_j$. Then by taking into account the cost, we have* $u_j(\hat{c}_j|\hat{c}_j) = u_j(\hat{c}_j|c_j) - c_j + \hat{c}_j$. Also from Equation 6, we see that $u_j(c_j|c_j) - u_j(\hat{c}_j|\hat{c}_j) = P(s_j, c_j) - P(s_j, \hat{c}_j)$. Putting them together we see that $u_j(\hat{c}_j|c_j) = u_j(c_j|c_j)$, thus the player has no reason to lie. With similar reasoning, a pool leader has no reason to lie by raising its cost so much that its rank increases above $k$.

Similar considerations, show that no pool member (i.e., with rank at least $k+1$) has an incentive to lie. This includes the special case of the player with rank $k+1$. As a conclusion, we see that under the assumption that the players end up at a perfect equilibrium, it is a dominant strategy to declare the true cost.

Alternatively, we could adapt the reward strategy to implement the Vickrey-Clarke-Groves (VCG) mechanism, which applies to all mechanism design problems. In this particular case, the VCG mechanism, would ask the players to declare their costs $\vec{c}$, but the reward scheme would *use a different vector of costs* $\vec{c}'$ *for the game*. The new costs $\vec{c}'$ will be such $P(s_j, c_j) = P(s_{k+1}, c_{k+1}) + \delta$, for all players with rank at most $k$ and for some tiny $\delta$.

# B Analysis of inner-outer game

Skip (very long proof by cases)

First, we will state a basic lemma that we will use in the proofs of the following lemmas and theorems. This lemma is a generalization of Lemma 2 and intuitively says that according to any joint strategy of any inner game, the utility that a player takes from allocating stake $s$ to other active pools is upper bounded by $D_{\max} \cdot s/z_0$, where $D_{\max}$ is the maximum desirability of all the other active pools. Formally:

**Lemma 3.** *For every joint strategy of the outer game* $(\vec{m}, \vec{\lambda})$ *and for every joint strategy* $\vec{S}^{(\vec{m},\vec{\lambda})}$ *of the inner game determined by* $(\vec{m}, \vec{\lambda})$*, it holds: For every player* $j$ *that has allocated stake* $s$ *to other active pools* $\sum_{i\in[n]\setminus\{j\}: i \text{ active}} u_{j,i}(\vec{S}^{(\vec{m},\vec{\lambda})}) \le D_{\max} \cdot \frac{s}{z_0}$*, where* $D_{\max}$ *is the maximum desirability according to* $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ *of all the other active pools.*

Its proof is similar to the proof of Lemma 2. Recall that inactive pools in this model have desirability zero and their members (if they exist) take utility zero from these pools.

## B.1 Equilibria of the inner game

**Theorem 2.** *For every* $\epsilon'$ *such that* $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ *and for every* $\alpha$ *such that* $\frac{s_{k+1}}{z_0} < \alpha < 1$*, it holds: A joint strategy* $\vec{S}^{(\vec{m}^*(\epsilon',\alpha),\vec{\lambda}^*)}$ *of the inner game determined by* $(\vec{m}^*(\epsilon', \alpha), \vec{\lambda}^*)$ *is a Nash equilibrium if and only if it forms* $k$ *active, saturated pools, whose pool leaders belong to* $G$.

*Proof.* This can be proved by the following two Lemmas 4, 5. □

**Lemma 4.** *For every* $\epsilon'$ *such that* $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}, \epsilon_1\}$ *and for every* $\alpha$ *such that* $\frac{s_{k+1}}{z_0} < \alpha < 1$*, it holds: In an inner game determined by* $(\vec{m}^*(\epsilon', \alpha), \vec{\lambda}^*)$*, joint strategies* $\vec{S}^{(\vec{m}^*(\epsilon',\alpha),\vec{\lambda}^*)}$ *that form* $k$ *active saturated pools, whose pool leaders belong to* $G$*, are Nash equilibria.*

*Proof.* We take an arbitrary $\epsilon'$ with $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ and an atribrary $\alpha$ with $\frac{s_{k+1}}{z_0} < \alpha < 1$ For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\epsilon', \alpha)$.

Let us take an arbitrary joint strategy $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ that forms $k$ saturated active pools, whose pool leaders belong to $G$. This means that players in $G$ have decided to activate their own pools with margins and the stakes determined by $(\vec{m}^*, \vec{\lambda}^*)$ in the outer game and that the other players have delegated all their stake to these pools, such that each pool has stake $z_0$. We will prove that this joint strategy is a Nash equilibrium of the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$, or in other words that no player can increase his non-myopic utility by changing his strategy, given that the other players do not change their strategies.

- Let us take a player $j$ that belongs to $G$. Recall that $\lambda_j^* = s_j$. The way in which he can change his strategy is to dissolve his pool (make it inactive) and to allocate his stake to one or more other pools whose pool leaders also belong to $G$ or to leave it unallocated. Note that he can also allocate his stake to an inactive pool whose pool leader decided not to allocate the stake (determined in the outer game) to it, but in this case his utility will become zero. So we can assume that the player in this case leaves his stake unallocated.

  1. Let us take a strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ of player $j$, according to which he dissolves his pool and allocates his stake to one or more other pools whose pool leaders also belong to $G$. Intuitively this strategy will decrease player $j$'s utility, because he will lose the rewards from the margin of his own pool and additionally cannot increase his profit as a pool member of the other pools, because no pool has higher desirability than his own pool.

     Note that a player in $G$ in the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$ cannot choose to activate his own pool and simultaneously be a member of another pool as his strategy, given that $\lambda_j^* = s_j$ and therefore $a_{j,j} \in \{0, s_j\}$. Formally: Regarding player $j$'s current non-myopic utility, we have:

$$u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \overset{\lambda_i^*=s_i}{=} u_{j,j}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) = (m_j^* + (1-m_j^*)\cdot\frac{s_j}{z_0})\cdot P(s_j,c_j)$$
$$\overset{P(s_j,c_j)>0}{>} (1-m_j^*)\cdot\frac{s_j}{z_0}\cdot P(s_j,c_j) = \frac{s_j}{z_0}\cdot(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha)) > 0.$$

     Note that $\text{rank}_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \le k$, because only $k$ pools are active and $\pi_j$ has positive desirability, given that $P(s_j,c_j) > P(s_{k+1},c_{k+1}) > 0$ by assumption.

     If he chooses a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ as described above, where he allocates some part of his stake $s_{j_1}$ to a pool $\pi_l$ and the remaining part $s_{j_2} = s_j - s_{j_1}$ to a pool $\pi_{l'}$, then we have :

$$u_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (1-m_l^*)\cdot P(s_l,c_l)\cdot\frac{s_{j_1}}{z_0+s_{j_1}} = (P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_{j_1}}{z_0+s_{j_1}}$$

     and

$$u_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_{j_2}}{z_0+s_{j_2}}.$$

     Note that by the description of $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, both $\pi_l$ and $\pi_{l'}$ have stake $z_0$, and their pool leaders belong to $G$. Furthermore, according to both joint strategies $(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$ and $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, at most $k$ pools are active and have positive desirability (in $(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$, player $j$ has dissolved his pool), thus the ranking of all active pools is smaller or equal to $k$ (they belong to the $k$ most desirable pools).

     So

$$u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot(\frac{s_{j_2}}{z_0+s_{j_2}} + \frac{s_{j_1}}{z_0+s_{j_1}}) < u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}),$$

     which means that player $j$ cannot increase his non-myopic utility by this change. Note that in the above proof, we did not assume anything specific about the stakes $s_{j_1}$ and $s_{j_2}$.

By induction on the number of the pools to which $j$ allocates stake, we can prove in the same way that $j$'s utility will not increase if he dissolves his pool.

Note that the number of pools to which $j$ allocates stake does not have an impact on desirability and ranking of the pools, because in all cases (i) there exist $k-1$ active pools with positive desirability ($j$ has dissolved his own pool) and (ii) desirability, which determines ranking, does not depend on pool size.

2. Let us consider the case where $j$ leaves his stake unallocated. Then his utility will drop to zero, which is smaller than his current utility $u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) > 0$.
3. Finally, let us look at the case where $j$ leaves some of his stake unallocated and allocates the rest to the other existing pools. Then by using the inequalities of the two previous cases, we can prove that his utility will decrease.

- Consider a player $j \notin G$. According to $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, this player has not activated his own pool. Then the ways in which he can change his strategy are (i) to activate his own pool by allocating the stake and margin specified in the outer game, (ii) to remove some parts of his stake from some pools and allocate them to other already saturated pools, or (iii) to remove some parts of his stake from some pools and leave them unallocated or allocate them to inactive pools. Recall that again $\lambda_j^* = s_j$, so player $j$ cannot activate his own pool and simultaneously allocate stake to other pools in the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$.

1. Consider the first case where $j$ changes his strategy to $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ by removing all his stake from the existing pools and activates his own pool as specified by the outer game.

Intuitively, his utility will decrease, because if he activates his own pool, this pool will not belong to the $k$ most desirable pools and thus will have non-myopic stake $s_j$. Additionally, the desirability of his own pool will be lower than the desirability of one of the active pools. Formally: His current utility is

$$u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) = \sum_{i\in[n]\sim j} u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) = \sum_{i\in[n]\sim j} [(1-m_i^*)\cdot \frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{z_0}\cdot P(s_i,c_i)]$$
$$= \sum_{i\in[n]\sim j} [\frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{z_0}(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))] = \frac{s_j}{z_0}\cdot(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha)).$$

If he activates his own pool with his whole stake as specified by the outer game ($\lambda_i^* = s_i$), then the ranking $\mathrm{rank}_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$ of this pool $\pi_j$ will be $k+1$. This holds because the desirability of $\pi_j$ will be strictly smaller than the desirability of the $k$ existing active pools, whose pool leaders belong to $G$.

In more detail, if $j = k+1$, we have for all $i \in G$:

$$D_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (1-\frac{\epsilon'\cdot\alpha}{P(s_{k+1},c_{k+1})})\cdot P(s_{k+1},c_{k+1})$$
$$= P(s_{k+1},c_{k+1}) - \epsilon'\cdot\alpha < P(s_{k+1},c_{k+1}) + \epsilon'\cdot(1-\alpha) = D_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}).$$

If $j \neq k+1$, we have for all $i \in G$:

$$D_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \le \big((1-0)\cdot P(s_j,c_j)\big)^+$$
$$< P(s_{k+1},c_{k+1}) + \epsilon'\cdot(1-\alpha) = D_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}).$$

As a result we have

$$
\begin{aligned}
u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) &\overset{\lambda_j^*=s_j}{=} u_{j,j}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\
&= \frac{R}{1+a_0} \cdot [m_j^* + (1-m_j^*) \cdot \frac{s_j}{s_j}] \cdot [s_j + s_j \cdot a_0 \cdot \frac{s_j \cdot (1 - \frac{z_0 - s_j}{z_0})}{z_0} - c_j \cdot \frac{1+a_0}{R}] \\
&= \frac{R}{1+a_0} \cdot [s_j + s_j \cdot a_0 \cdot \frac{s_j \cdot (1 - \frac{z_0 - s_j}{z_0})}{z_0} - c_j \cdot \frac{1+a_0}{R}] \\
&\overset{\frac{d[(r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma}>0,\text{when}\sigma<z_0\wedge j\notin G}{\leq} P(s_{k+1}, c_{k+1}) \cdot \frac{s_j}{z_0} \\
&< (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) \cdot \frac{s_j}{z_0} \\
&= u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}),
\end{aligned}
$$

which means that his utility will not increase by activating his own pool.

2. Now consider the case where $j$ changes his strategy to $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ by removing some parts of his stake from some pools and allocate them to two other, already saturated pools. We will prove that $j$'s utility will not increase in this case. Then we can use induction on the number of pools to which he allocates stake according to his new strategy, so that we prove the following statement: His utility will not increase by removing parts of his stake from some pools and allocating them to other already saturated pools.
   Intuitively, his utility will decrease, because the reward function has the property $r(\sigma, \lambda) = r(z_0, \lambda)$ for $\sigma > z_0$.
   Let $s_{j_i}$ be the stake that $j$ will remove from each pool $\pi_i$ and $a'_{j,l}$, $a'_{j,l'}$ the stake that he will allocate to pools $\pi_l$ and $\pi_{l'}$ respectively. Note that $j$'s stake, that exists in a pool $\pi_i$ after $j$ removes $s_{j_i}$, is $a_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_i}$ and that the stake that exists in the pools $\pi_l$ and $\pi_{l'}$ after he allocates the stake removed from the other pools to them is $a_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + a'_{j,l}$ and $a_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,l'}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + a'_{j,l'}$ respectively.
   His current utility is

$$
u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) = \sum_{i \in [n] \setminus j} u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) = \frac{s_j}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)).
$$

With this new strategy his utility will be

$$
\begin{aligned}
u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = &\sum_{i \in [n] \setminus \{l,l',j\}} [u_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})] \\
&+ u_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) + u_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\
&= \sum_{i \in [n] \setminus \{l,l',j\}} [(1-m_i^*) \cdot \frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_i}}{z_0} \cdot P(s_i, c_i)] \\
&+ (1-m_l^*) \cdot P(s_l, c_l) \cdot \frac{a'_{j,l} + a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{z_0 + a'_{j,l}} + \\
&(1-m_{l'}^*) \cdot P(s_{l'}, c_{l'}) \cdot \frac{a'_{j,l'} + a_{j,l'}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{z_0 + a'_{j,l'}} \\
&< u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}).
\end{aligned}
$$

So he cannot increase his utility by choosing this strategy. Note that the ranking of the pools also does not change in this case, because the desirability does not depend on pool size and the number of the active pools remains the same.

3. Consider the case where $j$ removes part of his stake from some pools and leaves it unallocated or allocates it to inactive pools: The utility in this case cannot increase, because as pool member, his utility is always greater than zero.
4. Finally, consider the case where $j$ does a combination of the two above strategies. Then we can prove that his utility will decrease using the inequalities from the previous cases.

$\square$

**Lemma 5.** *For every $\epsilon'$ such that $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ and for every $\alpha$ such that $\frac{s_{k+1}}{z_0} < \alpha < 1$ it holds: In an inner game determined by $(\vec{m}^*(\epsilon', \alpha), \vec{\lambda}^*)$, joint strategies $\vec{S}^{(\vec{m}^*(\epsilon',\alpha),\vec{\lambda}^*)}$ that do not form $k$ active saturated pools, whose pool leaders belong to $G$, are not a Nash equilibrium.*

*Proof.* We take arbitrary $\epsilon'$ such that $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}, \epsilon_1)\}$ and $\alpha$ such that $\frac{s_{k+1}}{z_0} < \alpha < 1$ to prove the above statement. For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\epsilon', \alpha)$.

First, we will prove that there is no Nash equilibrium joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ for which there exist one or more players in $G$ that have not chosen to activate their own pools.

Second, we will prove that there no Nash equilibrium joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ for which there exist one or more players $\notin G$ that have activated their pools.

Last, we will prove that there no Nash equilibrium joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ for which there exist one or more players who have allocated some of their stake to a pool with total stake more than $z_0$ or for which there exists some stake that is unallocated or has been allocated to an inactive pool.

- Consider a joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ for which there exists at least one player $j$ in $G$ that has not chosen to activate his own pool. We will prove that it cannot be a Nash equilibrium, because $j$ will increase his utility if he removes his stake from the other pools and activates his own pool.

  Note that if $j$ activates his own pool $\pi_j$, then $\pi_j$ will belong to the $k$ most desirable pools, because according to the margins that are specified by the outer game, no pool activated by player $\notin G$ has the same or higher desirability than $\pi_j$. This is proved in the second case of the proof of Lemma 4. In addition to this, all the other pools activated by players in $G$ have the same desirability as $\pi_j$, not higher.

  Regarding player $j$'s current utility, by Lemma 3 we have:

$$\begin{aligned} u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= \sum_{i \in [n] \setminus \{j\}} u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \\ &\leq \frac{s_j}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) \end{aligned}$$

  Note that $D_{\max} \leq P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)$ in this inner game, because even if all pools are activated, the pools of players in $G$ have the highest desirability, which is $P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)$, as we proved in the second case of proof of Lemma 4.

  If $j$ activates his own pool, then his utility will become:

$$\begin{aligned} u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) &\overset{\lambda_i^* = s_i}{=} u_{j,j}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\ &= (m_j^* + (1-m_j^*) \cdot \frac{s_j}{z_0}) \cdot P(s_j, c_j) \\ &\overset{P(s_j,c_j), m_j^* > 0}{>} (1-m_j^*) \cdot \frac{s_j}{z_0} \cdot P(s_j, c_j) \\ &= u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) > 0 \end{aligned}$$

- Consider a joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ that is Nash equilibrium. We will prove by contradiction that no player $\notin G$ has activated his own pool. Specifically we will prove that in such a case, $j$ will increase his utility if he dissolves his pool and allocates his stake to other pools. Therefore this joint strategy cannot be an equilibrium. By the previous case we know that in a joint strategy that is an equilibrium, all players in $G$ have activated their own pools and that these pools have strictly higher desirability than $\pi_j$. So $\pi_j$ does not belong to the $k$ most desirable pools. The utility of $j$ according to $S^{(\vec{m}^*,\vec{\lambda}^*)}$ will be

$$u_j(S^{(\vec{m}^*,\vec{\lambda}^*)}) \overset{\lambda_j^*=s_j}{=} u_{j,j}(S^{(\vec{m}^*,\vec{\lambda}^*)}) < \frac{s_j}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)).$$

This is computed in the same way as in the second case of the proof of Lemma 4.

If he chooses a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$, where he dissolves his own pool and allocates stake to pools whose pool leaders belong to $G$, so that no pool has stake has more than $z_0$, his utility will become:

$$\begin{aligned}
u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) &= \sum_{i\in[n]\smallsetminus j} u_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\
&= \sum_{i\in[n]\smallsetminus j} [(1-m_i^*) \cdot \frac{a_{j,i}(S^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})}{z_0} \cdot P(s_i, c_i)] \\
&= \frac{s_j}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) \\
&> u_j(S^{(\vec{m}^*,\vec{\lambda}^*)}).
\end{aligned}$$

- Let us suppose that according to joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)})$, there exists at least one pool $\pi_l$ with stake more than $z_0$. We will prove that in this case $S^{(\vec{m}^*,\vec{\lambda}^*)}$ is not a Nash equilibrium. In more detail, we will prove that a player $j$ that is pool member of this pool $\pi_l$ can increase his utility by choosing a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$. This strategy will be to remove some of his stake, let us say $s_{j_1} < \min\{z_0 - \sigma_i, \sigma_l - z_0\}$,

  and to allocate it to an unsaturated pool $\pi_i$, whose pool leader belongs to $G$. Firstly, $\pi_l$'s pool leader belongs to $G$, because in an equilibrium, as we proved in the previous two cases, only these players activate their own pools. Recall that these pools have positive desirability. Thus $\text{rank}_l(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \le k$.

  In addition, we know that there exists an unsaturated pool $\pi_i$, whose pool leader belongs to $G$, because $\pi_l$ has stake more than $z_0$ and there exist $k$ pools with pool leaders belonging to $G$. Moreover $\text{rank}_i(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \le k$ for the same reason as for $\pi_l$.

  Furthermore, desirability does not depend on pool size and thus:

  $\text{rank}_l(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \le k$ and $\text{rank}_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \le k$.

  Regarding the current utility of $j$ we have:

$$\begin{aligned}
u_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,l}(S^{(\vec{m}^*,\vec{\lambda}^*)})}{\sigma_l}, \\
u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,i}(S^{(\vec{m}^*,\vec{\lambda}^*)})}{z_0}.
\end{aligned}$$

If player $j$ chooses a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j)$, where he removes $s_{j_1}$ from $\pi_l$ and allocates to $\pi_i$, then only $u_{j,l}$ and $u_{j,i}$ will change. This happens because $j$'s allocations to pools other than $\pi_i$ and $\pi_l$ will remain unaffected, and thus the rewards of the other pools will remain unaffected as well. In addition to this, ranking is not affected as we again have $k$ active pools.

So

$$u_{j,l}(\vec{S'}^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (P(s_{k+1},c_{k+1}) + \epsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_1}}{\sigma_l - s_{j_1}}$$

and

$$u_{j,i}(\vec{S'}^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (P(s_{k+1},c_{k+1}) + \epsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + s_{j_1}}{z_0}.$$

So

$$u_{j,l}(\vec{S'}^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) + u_{j,i}(\vec{S'}^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) > u_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}).$$

As a result, joint strategy $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ as described above is not a Nash equilibrium.

- A joint strategy where a player has unallocated stake or stake allocated to an inactive pool is not a Nash equilibrium. This follows because the player can allocate his stake to a pool activated by a player in $G$ and obtain positive utility.

$\square$

### B.2 Equilibrium of the outer game

**Theorem 3.** *For every $\epsilon'$ with $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ and for every $\alpha$ with $\frac{s_{k+1}}{z_0} < \alpha < 1$ it holds: Joint strategy $(\vec{m}^*(\epsilon',\alpha), \vec{\lambda}^*)$ is an $\epsilon'$-non-myopic Nash equilibrium of the outer game.*

*Proof.* We take arbitrary $\epsilon'$ with $0 < \epsilon' < \min\{\epsilon, P(s_{k+1}, c_{k+1}), \epsilon_1\}$ and $\alpha$ with $\frac{s_{k+1}}{z_0} < \alpha < 1$ to prove the above statement. For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\epsilon',\alpha)$.

We will prove that $\forall i \in [n]$ and $\forall (m_i, \lambda_i) \neq (m_i^*, \lambda_i^*)$, it holds that

$$u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \epsilon'.$$

Specifically, we will examine the following cases for $i \in G$:

- $(m_i, \lambda_i)$, where $1 \geq m_i > m_i^*$ and $0 < \lambda_i \leq s_i = \lambda_i^*$.
- $(m_i, \lambda_i)$, where $0 \leq m_i < m_i^*$ and $0 < \lambda_i \leq s_i = \lambda_i^*$.
- $(m_i, \lambda_i)$, where $m_i = m_i^*$ and $0 < \lambda_i < s_i = \lambda_i^*$.
- $(m_i, \lambda_i) = (m_i, 0)$.

For $i \notin G$ we will examine the case $(m_i, \lambda_i) \neq (m_i^*, \lambda_i^*)$.

These are all the ways in which players can change the strategy described in the theorem. For each case we will prove that in the inner game that is determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, there is no equilibrium in which the utility of $i$ is higher than $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \epsilon'$.

The cases are described below:

- First we will prove that no player $i \in G$ has incentives to increase his margin more than $m_i^*$ in the outer game, regardless of $\lambda_i > 0$. By Lemmas 4,5 the only equilibria of the inner game determined by $(\vec{m}*, \vec{\lambda}*)$ are that ones where $k$ saturated, active pools have been activated by players in $G$. So for every $i \in G$ we have:

$$\begin{aligned} u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) &= u_i^{\text{outer,up}}(\vec{m}^*, \vec{\lambda}^*) \\ &= (m_i^* + (1-m_i^*) \cdot \frac{s_i}{z_0}) \cdot P(s_i, c_i) \\ &> \frac{s_i}{z_0} \cdot (1-m_i^*) \cdot P(s_i, c_i) \\ &\overset{i \in G}{=} \frac{s_i - \lambda_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)) + \frac{\lambda_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha)), \end{aligned}$$

where $0 < \lambda_i \leq s_i$.

1. If an $i \in G$ increases his margin by choosing $m_i$ with $\frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) + \epsilon' \cdot \alpha}{P(s_i,c_i)} \geq m_i > m_i^*$ and chooses arbitrary $\lambda_i$ with $0 < \lambda_i \leq s_i$, then:

$$\begin{aligned} &u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \\ &\leq \max\{(m_i + (1-m_i) \cdot \frac{\lambda_i}{z_0}) \cdot P(s_i, c_i) + \frac{s_i - \lambda_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) \\ &+ \epsilon' \cdot (1-\alpha)), \frac{s_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha))\} \\ &\leq m_i^* \cdot P(s_i, c_i) + \epsilon' + (1-m_i^*) \cdot \frac{s_i}{z_0} \cdot P(s_i, c_i) \\ &\leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \epsilon'. \end{aligned}$$

   Note that in the best case there is an equilibrium where (i) $i$ has not activated his own pool, and his utility is at most $\frac{s_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha))$ by Lemma 3 or (ii) $i$ has activated his own pool, that belongs to the $k$ most desirable pools, and has allocated the remaining stake to the active pools with the highest desirability, which are the pools of players in $G$. The utility that these pools will give him will be at most $\frac{s_i - \lambda_i}{z_0} \cdot (P(s_{k+1}, c_{k+1}) + \epsilon' \cdot (1-\alpha))$ by Lemma 3.

   If there is no equilibrium, recall that $u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) = -\infty$.

2. If an $i \in G$ increases his margin by choosing $m_i$ with $1 \geq m_i > \frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) + \epsilon' \cdot \alpha}{P(s_i,c_i)}$ and chooses arbitrary $\lambda_i > 0$, then we can prove that there is no equilibrium in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, where the non-myopic utility of $i$ will be higher than his current lower utility of the outer game denoted by $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

   This happens because in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$ we can prove that there is no an equilibrium where player $k+1$ and the other players of $G$ have not activated their own pools (note that the desirability of $\pi_{k+1}$ and of the pools whose pool leaders belong to $G$, when they are active, are strictly higher than the desirability of $\pi_i$, because $m_i > \frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) + \epsilon' \cdot \alpha}{P(s_i,c_i)}$).

   So in the best case, in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$ (i) there is an equilibrium where player $i$ has activated his own pool with rank worse than $k$ and has delegated his remaining stake $(s_i - \lambda_i)$ to pools whose pool leaders belong to $G$, or (ii) there is an equilibrium where $i$ has not activated his own pool and has delegated his whole stake to pools whose pool leaders belong to $G$, which have the highest desirability.

   We will prove that in both cases $i$'s non-myopic utility will not be higher than his current lower utility of the outer game denoted by $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

   As a result, $u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

   In more detail:

– In case (ii), by Lemma 3, his utility is at most

$$(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_i}{z_0} < u_i^{\text{outer,low}}(\vec{m}^*,\vec{\lambda}^*).$$

– In case (i), his utility is

$$\begin{aligned}
&[m_i+(1-m_i)\cdot\frac{\lambda_i}{\lambda_i}]\cdot(r(\lambda_i,\lambda_i)-c_i)\\
&+(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_i-\lambda_i}{z_0}\\
&\overset{\frac{d[(r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma}>0,\text{when}\sigma<z_0}{\le} P(\lambda_i,c_i)\cdot(0+(1-0)\cdot\frac{\lambda_i}{z_0})+\\
&(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_i-\lambda_i}{z_0}\\
&\overset{P(\lambda_i,c_i)\le P(s_i,c_i),m_i^*>0}{<} u_i^{\text{outer,low}}(\vec{m}^*,\vec{\lambda}^*).
\end{aligned}$$

We can prove that *there is no equilibrium in the inner game* determined by $(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})$, *where the players in G other than i have not activated their own pools*, in the same way as the first case of the proof of Lemma 5. Note that also in this inner game determined by $(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})$: (i) When these players activate their own pools, then these pools always have rank less or equal to $k$, regardless which other pools have been activated, and (ii) no other pool has strictly higher desirability and offers these players higher utility as members than they get by running their own pools.

Now we will prove by contradiction that *there is no equilibrium in the inner game defined by* $(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})$, *where the* $(k+1)$*-st player does not activate his own pool.*

Let us suppose that there is a joint strategy $\vec{S}^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}$ that is an equilibrium of the inner game and for which player $k+1$ has not activated his own pool. Then by Lemma 3 it holds:

$$u_{k+1}(\vec{S}^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}) \le (P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_{k+1}}{z_0}.$$

Then if he chooses a different strategy $S'^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}_{k+1}$ where he activates his own pool, his utility can be increased:

$$\begin{aligned}
u_{k+1}(S'^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}_{k+1},\vec{S}^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}_{-(k+1)}) &\overset{m^*_{k+1}=\frac{\epsilon'\cdot\alpha}{P(s_{k+1},c_{k+1})}}{=} P(s_{k+1},c_{k+1})\cdot\overset{1-m_i}{\boxed{(1-\frac{\epsilon'\cdot\alpha}{P(s_{k+1},c_{k+1})})}}\cdot\frac{s_{k+1}}{z_0}\\
&+\overset{m_i}{\boxed{\frac{\epsilon'\cdot\alpha}{P(s_{k+1},c_{k+1})}}}\cdot P(s_{k+1},c_{k+1})\\
&=(P(s_{k+1},c_{k+1})+\epsilon'\cdot(1-\alpha))\cdot\frac{s_{k+1}}{z_0}-\epsilon'\cdot\frac{s_{k+1}}{z_0}+\epsilon'\cdot\alpha\\
&=u_{k+1}(\vec{S}^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})})\underbrace{-\epsilon'\cdot\frac{s_{k+1}}{z_0}+\epsilon'\cdot\alpha}_{>0}\\
&\overset{\alpha>\frac{s_{k+1}}{z_0}}{>} u_{k+1}(\vec{S}^{(m_i,\vec{m}^*_{-i},\lambda_i,\vec{\lambda}^*_{-i})}).
\end{aligned}$$

*

Recall: For pool leader utility is

$$\left(m_i+(1-m_i)\frac{s_i}{z_0}\right)P(s_i,c_i)$$

↑ leader bonus

The intuition behind this is that $k+1$ can activate his own pool, that belongs to the $k$ most desirable pools, and in this way can increase his utility, because of the margin that he will take. Note that his desirability is worse than the desirability of players in $G$, but the difference is small, and thus being pool leader is more profitable for $k+1$ than being member of one of their pools.

Recall that in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, if player $k+1$ activates his own pool, this pool belongs to the $k$ most desirable pools, because $P(s_{k+1}, c_{k+1}) - \epsilon' \cdot \alpha > P(s_{k+2}, c_{k+2})$, where $P(s_{k+1}, c_{k+1}) - \epsilon' \cdot \alpha$ is its desirability. Note that the desirability of the other pools $\notin G$ is at most $P(s_{k+2}, c_{k+2})$ and that the desirability of $\pi_i$, if it is activated, is also lower than $P(s_{k+1}, c_{k+1}) - \epsilon' \cdot \alpha$. So there exist at most $k-1$ pools activated with higher desirability than that of the $(k+1)$-st player, which causes player $(k+1)$'s pool to belong to the $k$ most desirable pools when it is activated.

- No player $i \in G$ has an incentive to make his margin smaller than $m_i^*$, given that his pool already belongs to the best $k$ pools in all the equilibria of the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$.

  In more detail: If an $i \in G$ decreases his margin by choosing $m_i < m_i^*$ and chooses an arbitrary $\lambda_i \le s_i$, then in the best case there is an equilibrium of the inner game where $\pi_i$ will again belong to the $k$ most desirable pools, and as a result:

$$
\begin{aligned}
u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) &\le (m_i + (1-m_i)\cdot\frac{\lambda_i}{z_0})\cdot P(s_i, c_i) \\
&+ \frac{s_i - \lambda_i}{z_0}\cdot(P(s_{k+1}, c_{k+1}) + \epsilon'\cdot(1-\alpha)) \\
&\overset{m_i < m_i^*}{<} (m_i^* + (1-m_i^*)\cdot\frac{\lambda_i}{z_0})\cdot P(s_i, c_i) \\
&+ \frac{s_i - \lambda_i}{z_0}\cdot(P(s_{k+1}, c_{k+1}) + \epsilon'\cdot(1-\alpha)) \\
&\le u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).
\end{aligned}
$$

- No player $i \in G$ has incentives to commit less stake to his pool in the outer game, because the other existing pools have the same desirability and will not give it higher utility. In more detail: If a player $i \in G$ chooses margin equal to $m_i^*$, but $\lambda_i < s_i = \lambda_i^*$, then in the best case his pool will belong to the $k$ most desirable pools, and using Lemma 3, we will have:

$$
\begin{aligned}
u_i^{\text{outer,up}}(\vec{m}^*, \lambda_i, \vec{\lambda}^*_{-i}) &\le (m_i^* + (1-m_i^*)\cdot\frac{\lambda_i}{z_0})\cdot P(s_i, c_i) \\
&+ \frac{s_i - \lambda_i}{z_0}(P(s_{k+1}, c_{k+1}) + \epsilon'\cdot(1-\alpha)) \\
&\le u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).
\end{aligned}
$$

- Player $i \in G$ has no incentives to set $\lambda_i = 0$, because using Lemma 3, we can prove that in any equilibrium of the inner game determined by $(m_i, \vec{m}^*_{-i}, 0, \vec{\lambda}^*_{-i})$, where he has not activated his own pool, as $\lambda_i = 0$, his utility for being a pool member of other pools will be lower than $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

- Player $i \notin G$ has no incentives to choose margin $m_i \ne m_i^*$ or to commit stake $\lambda_i$ less than $s_i$, because we can prove in the same way as in the second case of the proof of Lemma 5 that in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, there is no equilibrium where he has activated his own pool, so margin and stake committed to his own pool do not have an impact on his utility, whichthat by Lemma 3 will be at most $\frac{s_i}{z_0}\cdot(P(s_{k+1}, c_{k+1}) + \epsilon'\cdot(1-\alpha))$. For this proof, it is important that players $\notin G$ cannot lower the desirability of the pools activated by players in $G$ by allocating stake to them strategically, because desirability does not depend on pool size. In addition to that, even if player $k+1$ sets margin zero, his desirability remains strictly lower than the desirability of all the players of $G$.

□

## C Experimental results

**Initialization** We simulate 100 players, and we use $k = 10$ for the desired number of pools. We assign stake to each player by sampling from a *Pareto distribution* with parameter $\alpha = 2$. Furthermore, we assign a *cost* to each player, uniformly sampled from $[c_{\min}, c_{\max}]$, where both $c_{\min}$ and $c_{\max}$ are configurable.

**Player strategies** The following *strategies* are available for players:

- A player can *lead a pool* with *margin* $m \in [0,1]$.
- Alternatively, a player can *delegate* his stake to existing pools. He can freely choose how much stake to delegate to each pool, and he does not have to delegate all of his stake.

Initially, there are no pools (so technically, all players play the second strategy and delegate to zero pools). When it is a player's turn to *move*, he can freely switch to another strategy:

- A pool leader can keep his pool, but change his margin, or close his pool and delegate to other pools.
- A player without pool can delegate differently or start a pool.

If a pool leader decides to close his pool, all stake delegated to that pool by other players automatically becomes un-delegated.

**Simulation step** In each step, we look for a player with a move that increases the player's utility by a non-trivial amount[10]. If a player with such a move is found, we apply that move and repeat. If not, we have reached an equilibrium.

We have to deal with the technical problem that for each player, there is an infinity of potential moves to consider. We solve this problem in an approximate manner as follows:

- For pool moves, instead of considering all margins in $[0,1]$, we restrict ourselves to one or two margins, namely 1 (to consider the case where the player plans running a one-man pool) and the highest margin $m < 1$ that has a chance[11] to attract members (calculated to a precision of $10^{-12}$ if such a margin exists).
- For delegation moves, we approximate the optimal delegation strategy using a local search heuristic ("beam search"). Furthermore, we restrict ourselves to a resolution of multiples of $10^{-8}$ of player stake.

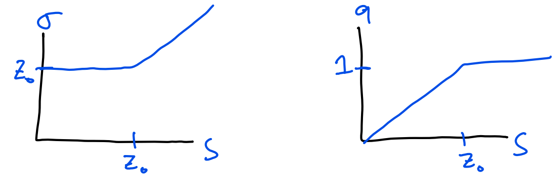
**Avoiding myopic moves** We have the problem of how to avoid "myopic" margin increases: It is always tempting for a pool leader to increase his margin (or for a delegating player to start a pool with a high margin), but such a move only makes sense if sufficiently few other players have incentive to create more desirable pools during the next steps.

To be more precise: If a player $A$ contemplates running a pool with margin $m < 1$, $A$ wants players to delegate to his pool and saturate it (if he wanted to run a one-man pool instead, the margin would be irrelevant and could be set to $1$).

Only pools with rank $\leq k$ attract delegations and have a chance of becoming saturated, so running a pool with margin $m$ only makes sense if the pool can reasonably be expected to end up with rank $\leq k$.

This means that when running such a pool, at most $k - 1$ other players should have incentive to run more desirable pools.

[10] Specifically, we consider utility $u_{\text{new}}$ non-trivially better than utility $u_{\text{old}}$ iff $u_{\text{new}} > u_{\text{old}} + 10^{-8}$.
[11] We make this precise in the next paragraph.




In order to determine whether $m$ satisfies this condition, we look at all other players. (1) For players who already run pools, we assume that they will continue running their pools and keep their margins.
(2) For each other player $B$, we check whether there exists a margin $m'$ such that by creating a pool with margin $m'$ and by assuming that that pool would have rank $m' \le k$, $B$ would increase its utility.

Let $B$ have stake $s$, costs $c$ and utility $u$. If $B$ manages to create a pool with rank $m' \le k$, then that pool's stake will be $\sigma := \max(s, z_0)$, and we can calculate its rewards $r$. Setting $q := \frac{s}{\sigma}$ and plugging in pool leader utility, we are looking for the minimal margin $m'$ satisfying

$$\underbrace{(r-c)\left[m' + (1-m')q\right]}_{(1)} > \underbrace{u}_{(2)}.$$

(1) Utility leader gets from pool
(2) Utility not running pool

We see that $r > c$ is a necessary condition. For $q = 1$ (i.e. $s \ge z_0$), $m' = 0$ is the obvious solution. For $q < 1$, we get

$$m' > \frac{u - (r-c)q}{(r-c)(1-q)},$$

and we pick $\frac{u-(r-c)q}{(r-c)(1-q)}$ as margin for player $B$.

We end up with a list of pools, one for each player, and we only allow $A$ to consider his pool move with margin $m$ if $A$'s pool would be amongst the $k$ most desirable pools in this list.

**Explaining the results** Outcome of each simulation is a diagram with various plots, visualizing the dynamics, and a table with data describing the reached equilibrium.

For the simulations reported here, we have always used the same stake distribution (sampled randomly from a Pareto distribution, as explained above) to make results more comparable (see figure 3).

**dynamics** The most interesting and most important plot, displaying the dynamic assignment of stake to pools. At the end of each simulation, once an equilibrium has been reached, we expect all stake to be assigned to ten pools of equal size.

**pools** The number of pools over time — this should end up at ten pools.

**margins** The evolution of pool margins. Instead of the margins themselves, their logarithms are plotted.

**ranks** The evolution of pool ranks over time.

**desirabilities** The evolution of pool desirabilities over time. For better visibility, the difference to $z_0$ is plotted instead of the desirabilities themselves.

In the tables describing the equilibrium, the meaning of the columns is as follows:

**player** The number of the player who leads the pool. Players are ordered descendingly by their *potential* $P(s, c)$ (cost $c$, stake $s$). Our expectation is to end up with ten pools, led by players 1–10.

**rk** The pool *rank*. We expect our final pools to have ranks 1–10.

**crk** The pool leader's *cost-rank*: The player with the lowest costs has cost-rank 1, the player with the second lowest costs has cost-rank 2 and so on. For low values of $a_0$, this should be close to the pool rank.

**srk** The pool leader's *stake-rank*: The player with the highest stake has stake-rank 1, the player with the second highest stake has stake-rank 2 and so on. For high values of $a_0$, this should be close to the pool rank.

**cost** The pool costs.

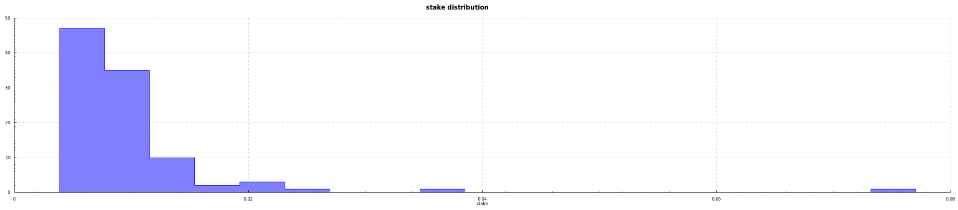



Figure 3: the stake-distribution used for all experiments

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 54 | 1 | 0.00156856 | 0.00898774 | 0.07704926 | 0.10000000 | 0.10154098 | 0.099073896587544 |
| 2 | 5 | 19 | 5 | 0.00121229 | 0.00125302 | 0.02052438 | 0.10000000 | 0.10041049 | 0.099073896587539 |
| 3 | 9 | 5 | 17 | 0.00108188 | 0.00088317 | 0.01216771 | 0.10000000 | 0.10024335 | 0.099073896587511 |
| 4 | 3 | 16 | 7 | 0.00120205 | 0.00063505 | 0.01694531 | 0.10000000 | 0.10033891 | 0.099073896587553 |
| 5 | 2 | 6 | 26 | 0.00108805 | 0.00053598 | 0.01075376 | 0.10000000 | 0.10021508 | 0.099073896587558 |
| 6 | 1 | 1 | 81 | 0.00100213 | 0.00047005 | 0.00613080 | 0.10000000 | 0.10012262 | 0.099073896587589 |
| 7 | 7 | 3 | 39 | 0.00105867 | 0.00047469 | 0.00898080 | 0.10000000 | 0.10017962 | 0.099073896587522 |
| 8 | 6 | 18 | 8 | 0.00121088 | 0.00042690 | 0.01635433 | 0.10000000 | 0.10032709 | 0.099073896587534 |
| 9 | 8 | 2 | 62 | 0.00103849 | 0.00026601 | 0.00693720 | 0.10000000 | 0.10013874 | 0.099073896587515 |
| 10 | 10 | 12 | 16 | 0.00115913 | 0.00011986 | 0.01224503 | 0.10000000 | 0.10024490 | 0.099073896587504 |

Table 1: low costs, low stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.02$)

**margin** The pool margin.

**player stake** The pool leader's stake.

**pool stake** The pool stake, including both leader and members.

**reward** The pool rewards (before distributing them amonst leader and members).

**desirability** The pool desirability.

We show the results of five examplary simulations with various costs and values for parameter $a_0$ (which gouverns the influence of pool leader stake on pool desirability):

- lows costs and low $a_0$, see figure 4 and table 1.
- lows costs and medium $a_0$, see figure 5 and table 2.
- lows costs and high $a_0$, see figure 6 and table 3.
- high costs and low $a_0$, see figure 7 and table 4.
- high costs and medium $a_0$, see figure 8 and table 5.
- high costs and high $a_0$, see figure 9 and table 6.

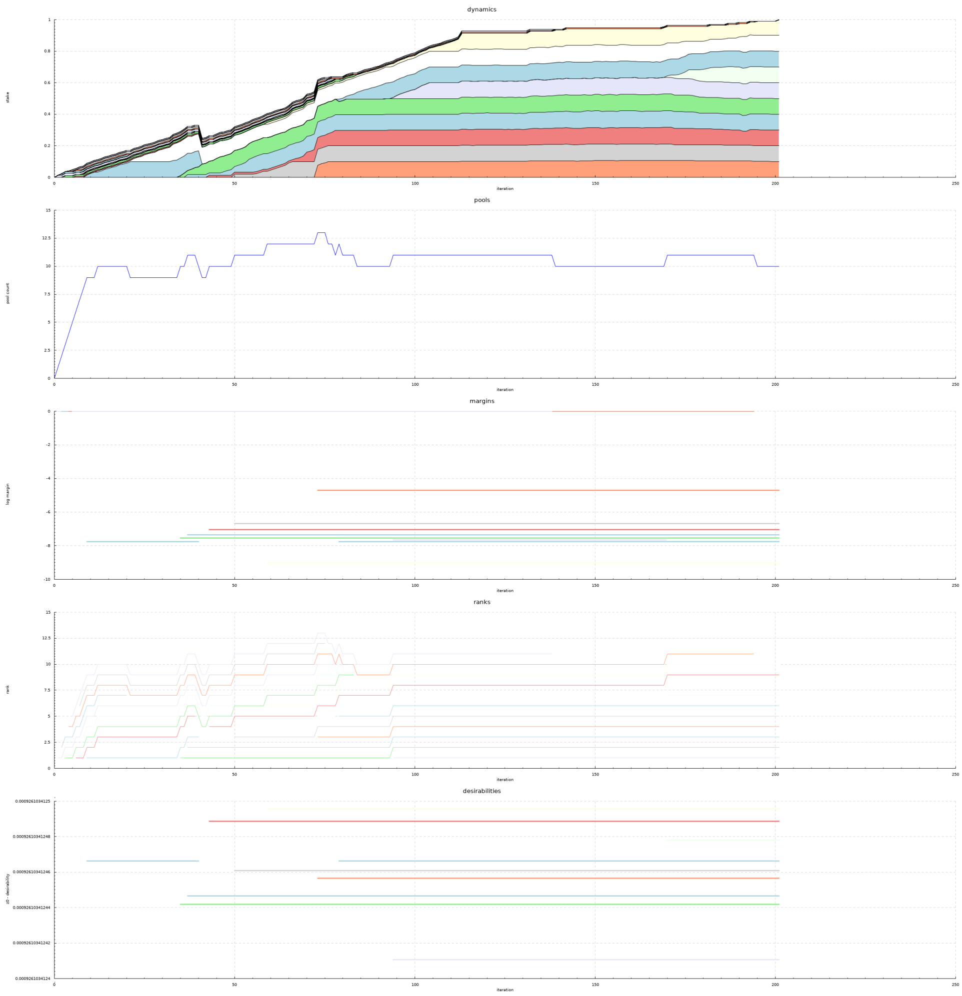


Figure 4: low costs, low stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.02$)

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 54 | 1 | 0.00156856 | 0.02828919 | 0.07704926 | 0.10000000 | 0.10385246 | 0.099390372362847 |
| 2 | 4 | 73 | 2 | 0.00173639 | 0.00742961 | 0.03741446 | 0.10000000 | 0.10187072 | 0.099390372362843 |
| 3 | 7 | 19 | 5 | 0.00121229 | 0.00424342 | 0.02052438 | 0.10000000 | 0.10102622 | 0.099390372362829 |
| 4 | 9 | 52 | 3 | 0.00156655 | 0.00297511 | 0.02507001 | 0.10000000 | 0.10125350 | 0.099390372362814 |
| 5 | 2 | 16 | 7 | 0.00120205 | 0.00255748 | 0.01694531 | 0.10000000 | 0.10084727 | 0.099390372362863 |
| 6 | 10 | 18 | 8 | 0.00121088 | 0.00217321 | 0.01635433 | 0.10000000 | 0.10081772 | 0.099390372362800 |
| 7 | 8 | 5 | 17 | 0.00108188 | 0.00136779 | 0.01216771 | 0.10000000 | 0.10060839 | 0.099390372362825 |
| 8 | 1 | 45 | 6 | 0.00152048 | 0.00090705 | 0.02002176 | 0.10000000 | 0.10100109 | 0.099390372362877 |
| 9 | 5 | 6 | 26 | 0.00108805 | 0.00059595 | 0.01075376 | 0.10000000 | 0.10053769 | 0.099390372362835 |
| 10 | 6 | 12 | 16 | 0.00115913 | 0.00063097 | 0.01224503 | 0.10000000 | 0.10061225 | 0.099390372362834 |

Table 2: low costs, medium stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.05$)

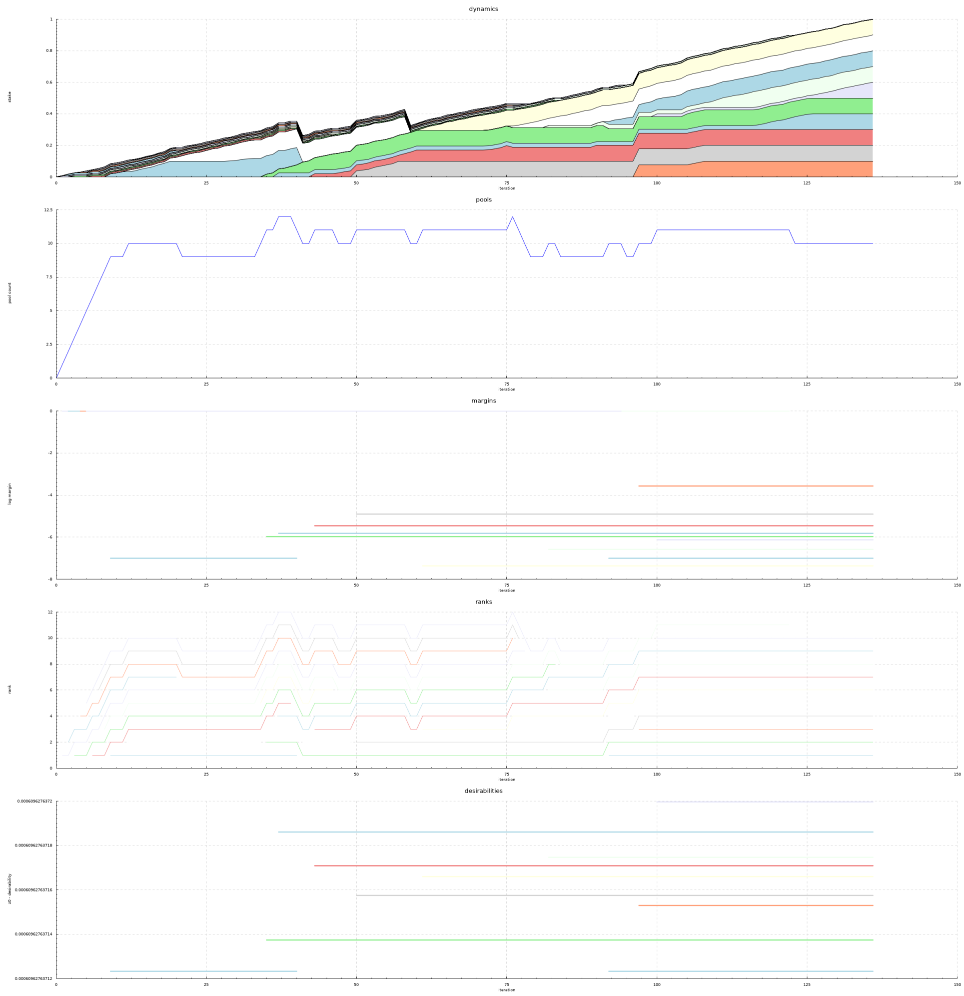



Figure 5: low costs, medium stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.05$)

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 1 | 8 | 54 | 1 | 0.00156856 | 0.23109808 | 0.07704926 | 0.10000000 | 0.13852463 | 0.105305784121818 |
| 2 | 1 | 73 | 2 | 0.00173639 | 0.09972617 | 0.03741446 | 0.10000000 | 0.11870723 | 0.105305784121865 |
| 3 | 3 | 52 | 3 | 0.00156655 | 0.05102956 | 0.02507001 | 0.10000000 | 0.11253500 | 0.105305784121842 |
| 4 | 2 | 19 | 5 | 0.00121229 | 0.03433392 | 0.02052438 | 0.10000000 | 0.11026219 | 0.105305784121863 |
| 5 | 7 | 80 | 4 | 0.00181011 | 0.03273015 | 0.02135839 | 0.10000000 | 0.11067919 | 0.105305784121826 |
| 6 | 6 | 45 | 6 | 0.00152048 | 0.02935389 | 0.02002176 | 0.10000000 | 0.11001088 | 0.105305784121833 |
| 7 | 10 | 16 | 7 | 0.00120205 | 0.01831648 | 0.01694531 | 0.10000000 | 0.10847266 | 0.105305784121808 |
| 8 | 4 | 18 | 8 | 0.00121088 | 0.01552360 | 0.01635433 | 0.10000000 | 0.10817716 | 0.105305784121835 |
| 9 | 5 | 64 | 10 | 0.00163171 | 0.00394146 | 0.01470840 | 0.10000000 | 0.10735420 | 0.105305784121835 |
| 10 | 9 | 82 | 9 | 0.00181572 | 0.00298747 | 0.01487410 | 0.10000000 | 0.10743705 | 0.105305784121813 |

Table 3: low costs, high stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.5$)

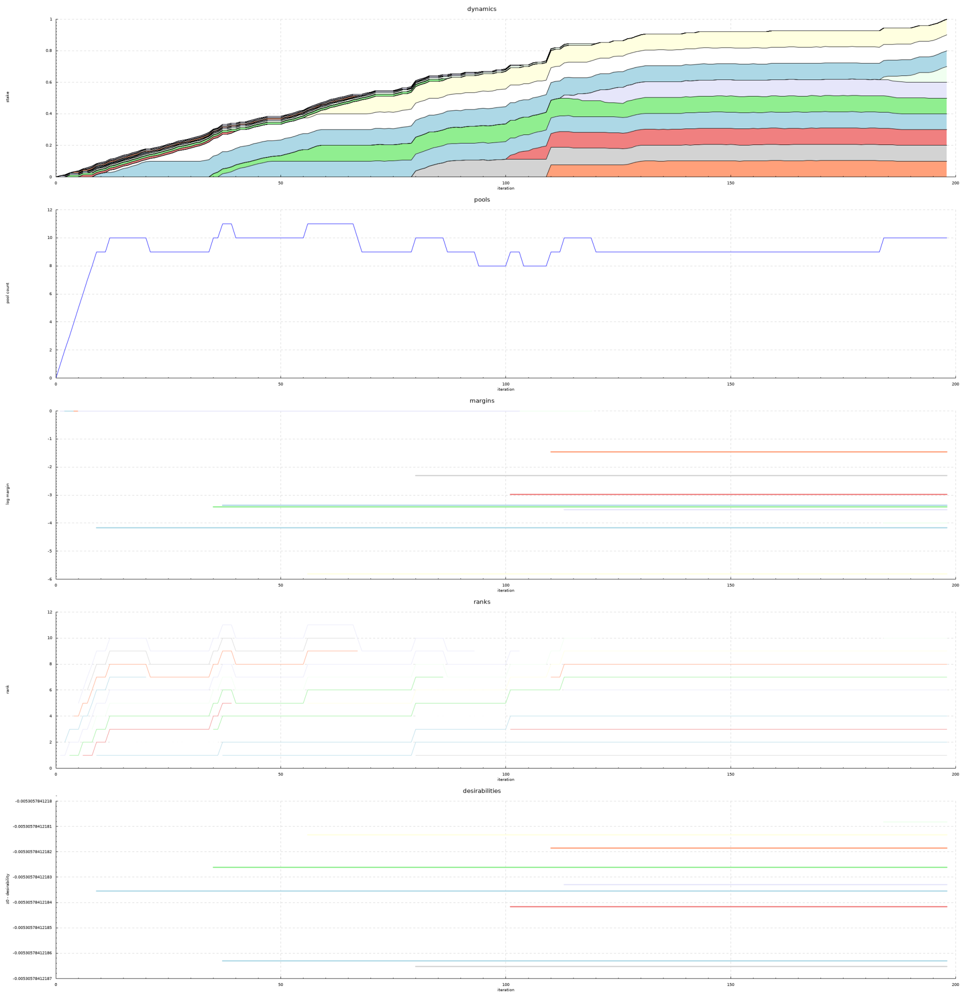



Figure 6: low costs, high stake influence ($c \in [0.001, 0.002]$, $a_0 = 0.5$)

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 1 | 4 | 1 | 81 | 0.05010638 | 0.14091746 | 0.00613080 | 0.10000000 | 0.10012262 | 0.042968072279390 |
| 2 | 8 | 2 | 62 | 0.05192430 | 0.10881324 | 0.00693720 | 0.10000000 | 0.10013874 | 0.042968072279376 |
| 3 | 3 | 3 | 39 | 0.05293337 | 0.09055053 | 0.00898080 | 0.10000000 | 0.10017962 | 0.042968072279392 |
| 4 | 10 | 4 | 65 | 0.05373632 | 0.07397050 | 0.00683225 | 0.10000000 | 0.10013665 | 0.042968072279365 |
| 5 | 1 | 5 | 17 | 0.05409406 | 0.06893326 | 0.01216771 | 0.10000000 | 0.10024335 | 0.042968072279396 |
| 6 | 9 | 6 | 26 | 0.05440243 | 0.06209143 | 0.01075376 | 0.10000000 | 0.10021508 | 0.042968072279369 |
| 7 | 7 | 7 | 70 | 0.05441660 | 0.06010553 | 0.00662237 | 0.10000000 | 0.10013245 | 0.042968072279379 |
| 8 | 5 | 8 | 47 | 0.05465632 | 0.05586892 | 0.00835115 | 0.10000000 | 0.10016702 | 0.042968072279383 |
| 9 | 6 | 9 | 96 | 0.05621197 | 0.02127128 | 0.00569447 | 0.10000000 | 0.10011389 | 0.042968072279383 |
| 10 | 2 | 10 | 90 | 0.05651394 | 0.01461751 | 0.00597063 | 0.10000000 | 0.10011941 | 0.042968072279394 |

Table 4: high costs, low stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.02$)

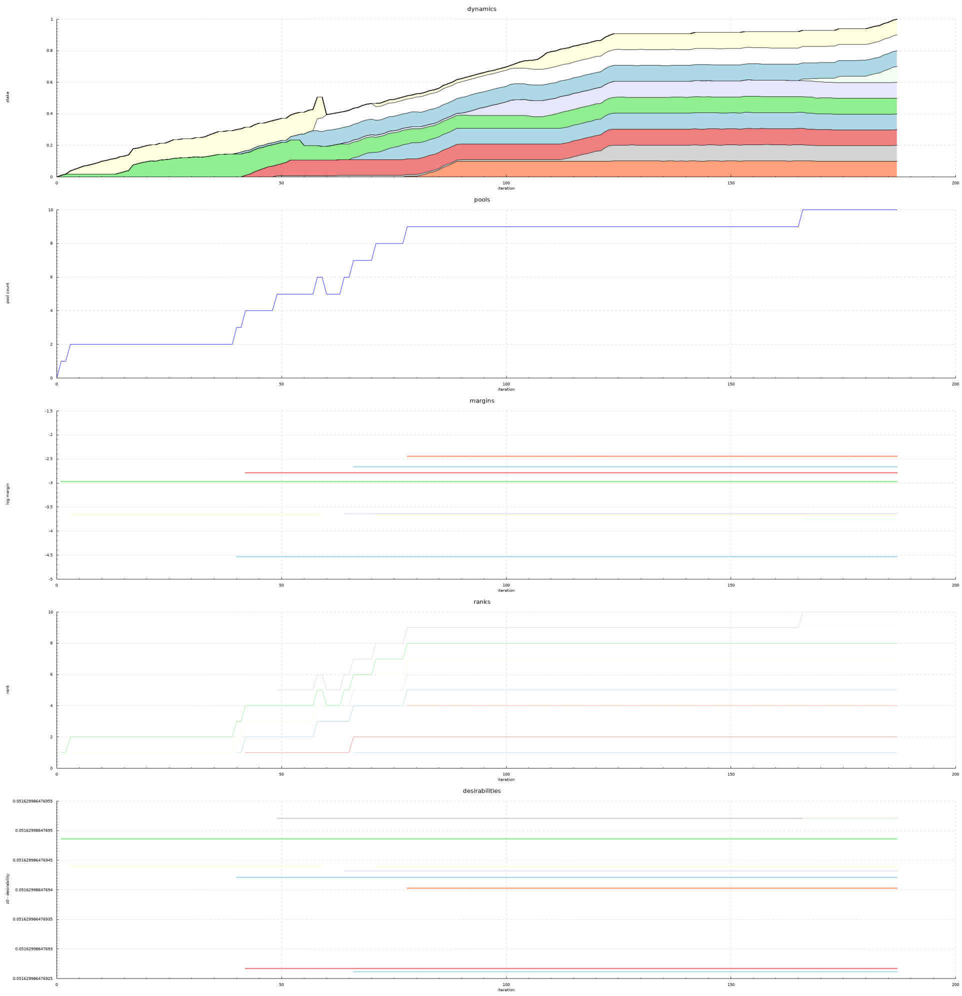



Figure 7: high costs, low stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.02$)

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 1 | 81 | 0.05010638 | 0.14071474 | 0.00613080 | 0.10000000 | 0.10030654 | 0.043136255076695 |
| 2 | 7 | 2 | 62 | 0.05192430 | 0.10917025 | 0.00693720 | 0.10000000 | 0.10034686 | 0.043136255076684 |
| 3 | 6 | 3 | 39 | 0.05293337 | 0.09216778 | 0.00898080 | 0.10000000 | 0.10044904 | 0.043136255076688 |
| 4 | 1 | 4 | 65 | 0.05373632 | 0.07443446 | 0.00683225 | 0.10000000 | 0.10034161 | 0.043136255076709 |
| 5 | 8 | 5 | 17 | 0.05409406 | 0.07262429 | 0.01216771 | 0.10000000 | 0.10060839 | 0.043136255076679 |
| 6 | 9 | 6 | 26 | 0.05440243 | 0.06500457 | 0.01075376 | 0.10000000 | 0.10053769 | 0.043136255076675 |
| 7 | 5 | 7 | 70 | 0.05441660 | 0.06050948 | 0.00662237 | 0.10000000 | 0.10033112 | 0.043136255076690 |
| 8 | 2 | 8 | 47 | 0.05465632 | 0.05736264 | 0.00835115 | 0.10000000 | 0.10041756 | 0.043136255076704 |
| 9 | 10 | 9 | 96 | 0.05621197 | 0.02124899 | 0.00569447 | 0.10000000 | 0.10028472 | 0.043136255076672 |
| 10 | 3 | 10 | 90 | 0.05651394 | 0.01480748 | 0.00597063 | 0.10000000 | 0.10029853 | 0.043136255076696 |

Table 5: high costs, medium stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.05$)

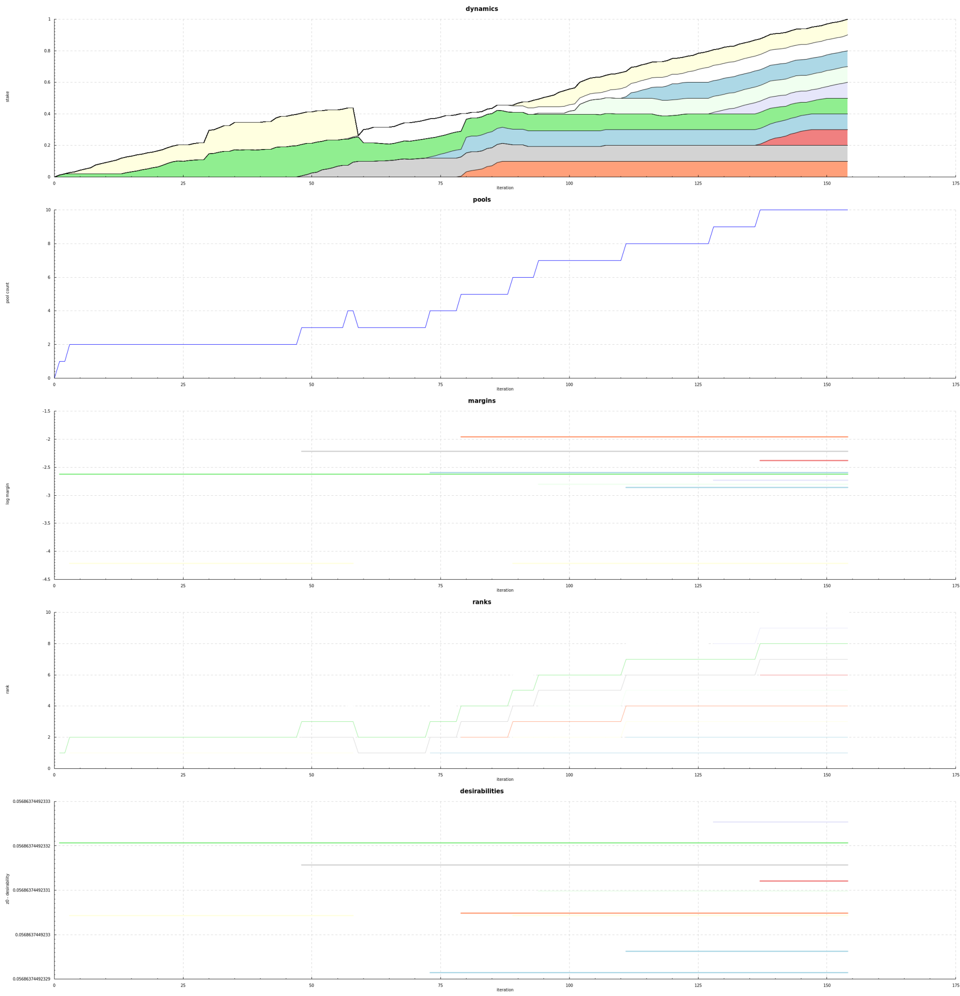


Figure 8: high costs, medium stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.05$)

| player | rk | crk | srk | cost | margin | player stake | pool stake | reward | desirability |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 4 | 1 | 81 | 0.05010638 | 0.08665199 | 0.00613080 | 0.10000000 | 0.10306540 | 0.048370013523060 |
| 2 | 10 | 2 | 62 | 0.05192430 | 0.06158357 | 0.00693720 | 0.10000000 | 0.10346860 | 0.048370013523048 |
| 3 | 2 | 3 | 39 | 0.05293337 | 0.06181530 | 0.00898080 | 0.10000000 | 0.10449040 | 0.048370013523073 |
| 4 | 1 | 5 | 17 | 0.05409406 | 0.06962481 | 0.01216771 | 0.10000000 | 0.10608385 | 0.048370013523074 |
| 5 | 8 | 6 | 26 | 0.05440243 | 0.05109295 | 0.01075376 | 0.10000000 | 0.10537688 | 0.048370013523051 |
| 6 | 6 | 4 | 65 | 0.05373632 | 0.02636475 | 0.00683225 | 0.10000000 | 0.10341613 | 0.048370013523057 |
| 7 | 9 | 8 | 47 | 0.05465632 | 0.02320800 | 0.00835115 | 0.10000000 | 0.10417557 | 0.048370013523048 |
| 8 | 5 | 7 | 70 | 0.05441660 | 0.01072862 | 0.00662237 | 0.10000000 | 0.10331119 | 0.048370013523058 |
| 9 | 3 | 54 | 1 | 0.07842812 | 0.19512770 | 0.07704926 | 0.10000000 | 0.13852463 | 0.048370013523062 |
| 10 | 7 | 19 | 5 | 0.06061469 | 0.02573105 | 0.02052438 | 0.10000000 | 0.11026219 | 0.048370013523056 |

Table 6: high costs, high stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.5$)

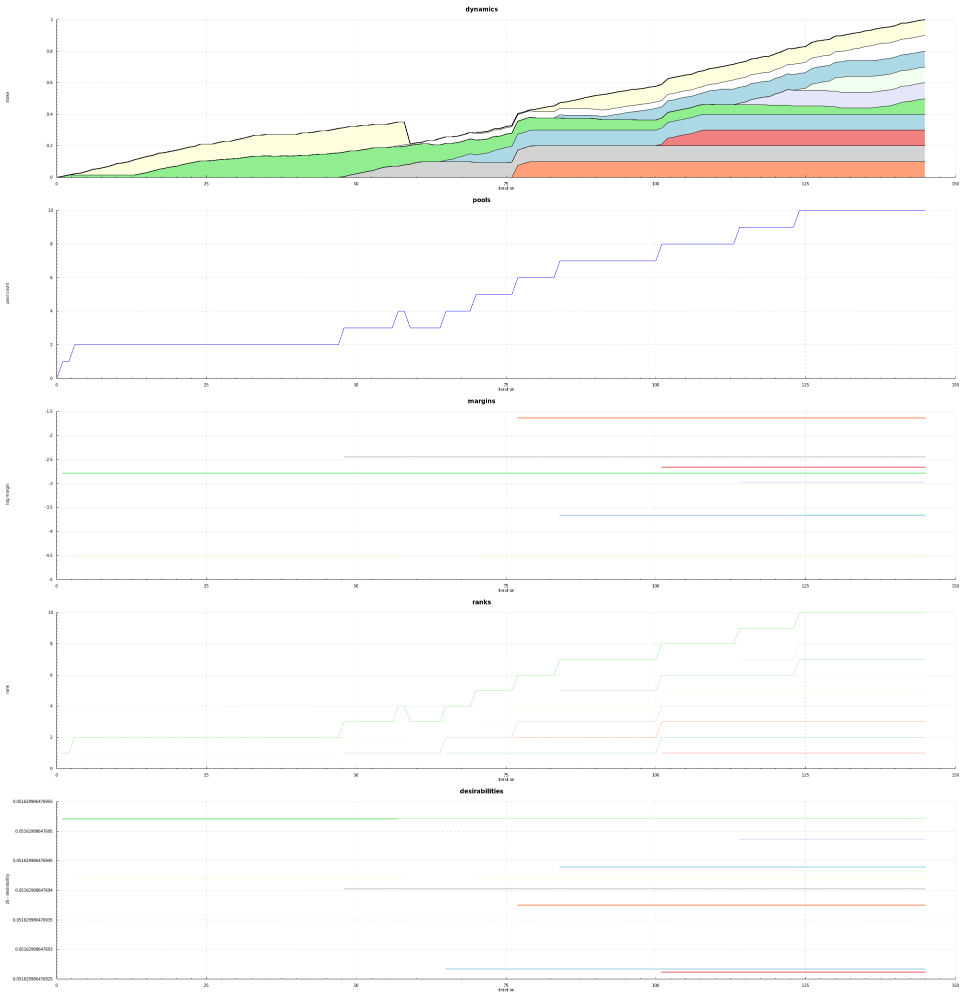



Figure 9: high costs, high stake influence ($c \in [0.05, 0.1]$, $a_0 = 0.5$)