PhD Open

Game-theoretic aspects in blockchain environments

Vangelis Markakis

Athens University of Economics and Business (AUEB)
Archimedes, Athena Research Center
Input Output Group (IOG)

Outline

- Reward sharing schemes and pool formation in Proof of Stake systems
- Governance in blockchains
- Transaction fee mechanisms

Why Game Theory?

- Blockchains are all about strategic interactions!
- Real-life games played by actual people
- Game theory can help a system designer in understanding the strategic behavior of the participants, and in designing rules to align his\her incentives with the players' incentives
- It also can help a player in understanding the behavior of his competitors and in selecting a good strategy for himself

Part 1: Reward sharing schemes and pool formation in Proof of Stake systems

Proof of Work vs Proof of Stake

- Proof of Work protocols rely on computational (hashing) power
  - Block producers need to solve some cryptographic puzzle
  - By brute force
- Very high energy consumption

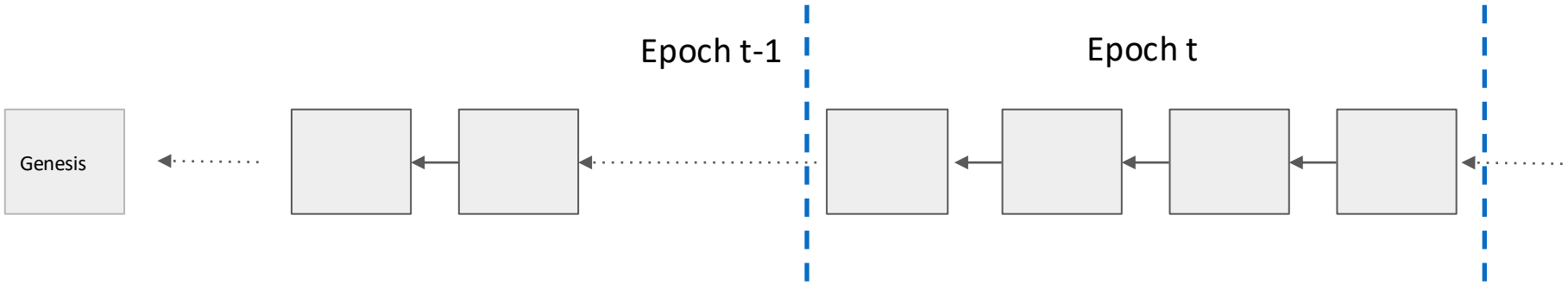
Proof of Work vs Proof of Stake

- Proof of Stake protocols
  - A different approach in designing blockchain protocols
- Computationally hard work replaced by randomization
  - Much more energy efficient!
- State of the system determined by the stake (amount of owned cryptocurrency) of each user
- The more stake you own, the higher the chance of producing blocks
- PoS protocols in operation since 2017
  - Ethereum switched to PoS in the Fall of 2022

Proof of Stake protocols

How do they work:

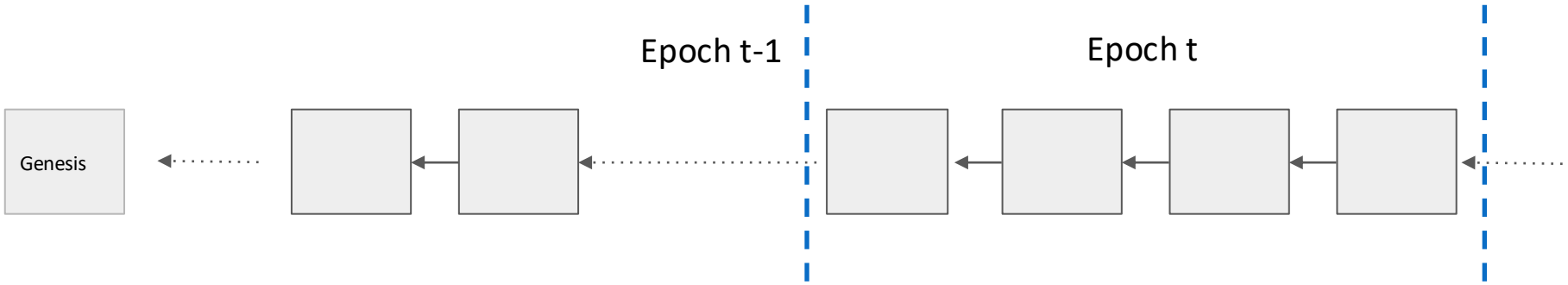
- Protocol runs on a per epoch basis
  - 6.5 mins in Ethereum, 5 days in Cardano
  - Each epoch is split into slots
- Registered validators: users who run a validating node and can participate in producing and validating blocks



Proof of Stake protocols

How do they work:

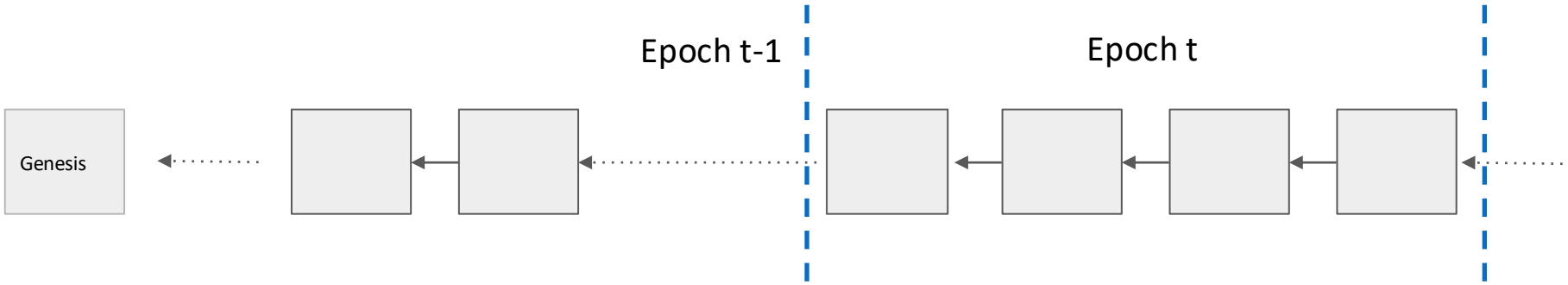
- Stake vector of validators at the beginning of a new epoch: $(s_1,\ldots, s_n)$
- Randomization used to select a committee of validators in each epoch
- Randomization also used to select a slot leader (or block proposer) among the set of registered validators for each time slot
  - ➢ the validator who is eligible to produce a block at a particular time slot
- Selection probability for validator i analogous to s_i (or $cap(s_i)$)



Proof of Stake protocols

How do they work:

- At a given slot
  - The slot leader proposes a new block of transactions to be added to the chain
  - The committee of validators vote on it
- In some slots we may have no new blocks
  - slot leader might be offline or out of sync or censored
- At the end of the epoch, rewards distributed to validators
  - block rewards + transaction fees



Proof of Stake protocols

How does someone become a validator?

- Needs only elementary technical knowledge
  - Installing and running the blockchain protocol
- Need to have the server on almost 24/7
  - some protocols penalize users for missing their chance of block production
- Need also some minimum amount of stake
  - In some cases explicitly enforced
  - In Ethereum, threshold = 32ETH ≈ 75000 USD (actually 127000 USD 1 year ago!)
  - If not enforced, any stakeholder can register but probability of “small” players producing blocks might be very small

Staking pools

- Stakeholders can operate as validators relying solely on their own stake
- But they can also join forces with other stakeholders to form a pool
  - With one pool member acting as the actual validator
  - For the non-validators, quite popular choice as a "passive" way to earn more money, without worrying about setting up a validating node
  - Blockchains may also offer tools to assist in choosing a validator to join

Staking pools

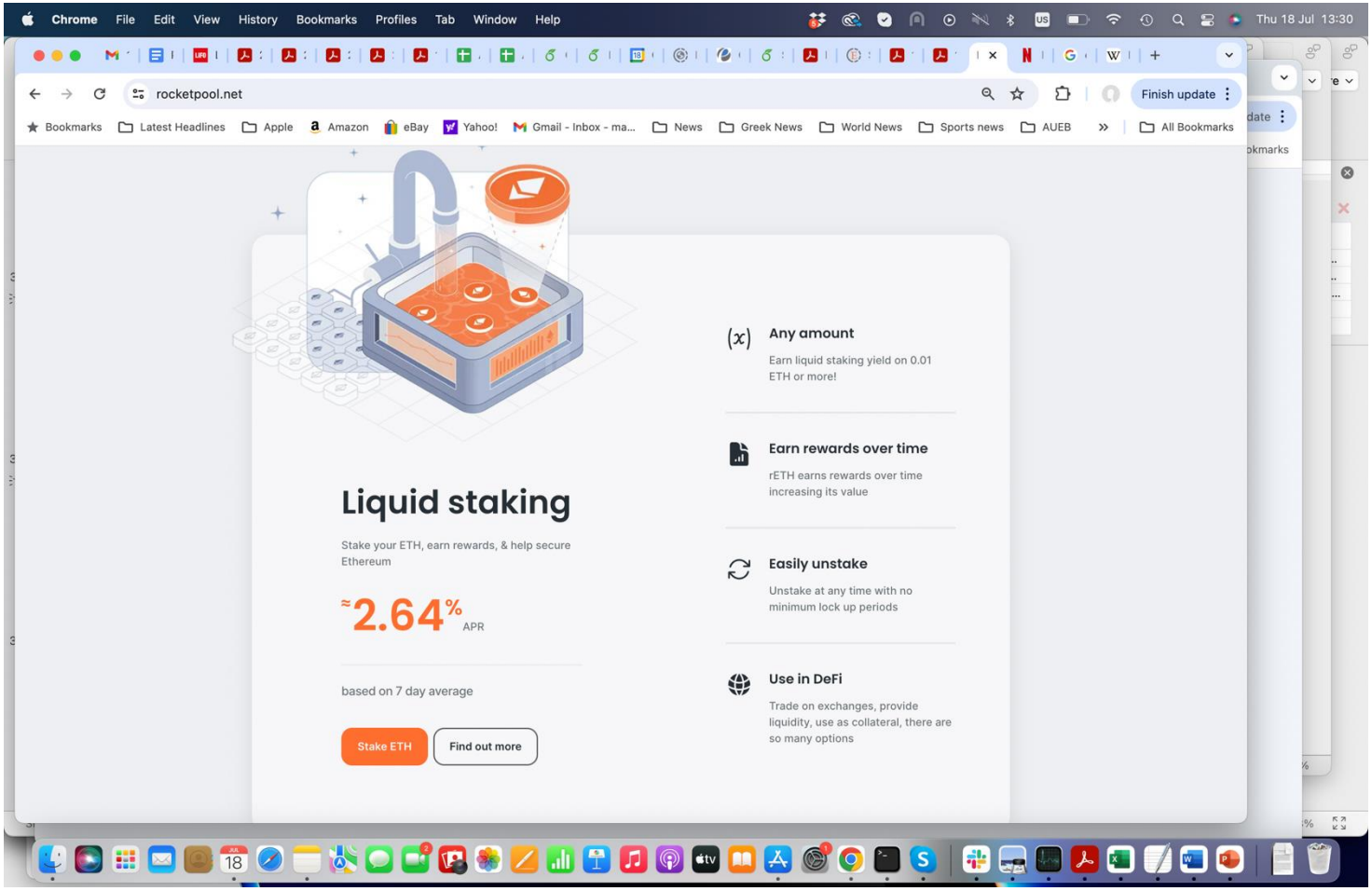
Options for pooling:

- can be supported by the protocol – onchain pooling (e.g. Cardano)
  - Each stakeholder can decide if she wants to be a validator (pool operator) or delegate her stake to a particular pool

- Or we can have offchain pooling:
  - can be supported by platforms that facilitate grouping into pools via smart contracts (e.g. Rocketpool for Ethereum)
  - Or simply by offchain agreements among individuals (applicable to all PoS systems)

Staking pools

A platform for pooling:



Onchain vs offchain pooling

In onchain pooling, the protocol needs to specify the entire reward mechanism:

1. total available reward per epoch, to be distributed to the pools
   - Coming from number of produced blocks and transaction fees
2. Reward sharing scheme at system level: how much reward is given to each pool in every epoch?
3. Reward sharing scheme at pool level: how is the pool reward split among the pool members?

In offchain pooling

1. The protocol specifies rewards to each individual validator
   - Coming from number of produced blocks and transaction fees
2. Reward sharing scheme at pool level:
   - Determined among the validator and the remaining pool members
   - Or determined by the mediator (if one exists)

Part 1a: A model of pool formation games for Ethereum-like protocols

Pool formation

A simple model

- Consider a set of players (stakeholders) N = {1, 2, …, n}
- Player i → stake s_i
  - Stake the player has decided to "lock" for pooling
- Threshold-based pool rewards
  - Assume that if a pool S⊆N forms, then it will obtain reward ρ(S):

$$\rho(S) = \begin{cases} 1, & \text{if } \sum_{i \in S} s_i \geq h \\ 0, & \text{otherwise} \end{cases}$$

  - Capturing Ethereum-like protocols
  - Pools are either winning (total stake at least h) or losing

Pool formation

- Q1: How should the members of a pool S split the reward?
- Q2: How can we evaluate a reward sharing rule?

Reward sharing among pool members

Q1: How should the members of a pool S split the reward?

- Most systems employ a proportional rule (according to stake owned)
  - Reward according to $s_i / \Sigma_j s_j$
  - After the validator claims her cost and profit margins
  - A quite natural rule to follow
- Is proportional sharing the best we can do?
  - We should first see what are the objectives we want to achieve

Relevant desiderata

- Promoting decentralization
  - creation of a high number of pools very desirable
  - Even beyond the number of pools: geographic diversity/robustness, presence of operators from all continents
- Sybil resistance
  - Discourage stakeholders to create multiple identities and split their stake into different pools
- Fairness
  - Creation of pools with roughly equal size
  - Fair allocation of rewards to its members
- Skin in the game
  - Operators contribute enough to their pool so that they are discouraged to take actions against the system

Reward sharing among pool members

- [Chen, Papadimitriou, Roughgarden '19]:
  - A formal justification for the use of proportional sharing
  - Axiomatic study but in a different context without pool formation
- What else could we try out?
- Almost anything from cooperative game theory!
  - Main objective in cooperative games: how to split rewards/costs so that formed coalitions are "stable"
  - But usually studied without the angle of decentralization or Sybil attacks

Solutions from cooperative games

The core [Gillies ’59]

- Most studied solution in cooperative games
- Informally: allocate rewards so that no subset of a pool has incentives to break away and form a pool on its own
  - A polyhedron of solutions (defined by linear inequalities)
- Main drawback: The core is empty in several games
  - Too strict to satisfy all core constraints
  - Core emptiness already discussed in the context of blockchains in [Lewenberg et al. ’15]

Solutions from cooperative games

The Shapley value [Shapley '53]

- Proposed as a way to reward more the most pivotal players
- Informally: what counts is the marginal contribution a player makes when added to a set; thus average out over all possible permutations of players
- Formally: Shapley value for player i in pool S =

$$\sum_{T \subseteq S \setminus \{i\}} \frac{|T|!(|S| - |T| - 1)!}{|S|!} (\rho(T \cup \{i\}) - \rho(T))$$

- Has already been proposed as a possible blockchain rewards sharing scheme [Chen et al. '20]
  - But studied only w.r.t. its computational aspects

Example for Shapley value computation

Consider a game where

- Threshold h=4
- 3 players, with stakes: $s_1 = 3$, $s_2 = 2$, $s_3 = 1$

Observation: For threshold-based binary rewards, it suffices to count in how many permutations each player is pivotal

Tutorial exercise: Compute each player's Shapley value and compare them to their proportional share

Pool formation

Back to our main questions

- Q1: How should the members of a pool S split the reward?
  - ➢ Notions from cooperative game theory: Proportional sharing, core, Shapley value,…
- Q2: How can we evaluate a reward sharing rule?
  - ➢ View them as noncooperative games and analyze their Nash equilibria

Detour: A short intro to noncooperative game theory

Game theory basics

What is a (noncooperative) game?

Any process where

- There are ≥ 2 interacting entities
- All entities need to take some decision
- The final outcome and the utility/profit of each player is determined by the choices of all players

Examples: board games, auctions, elections, ...

Game theory basics

Formal definition: A noncooperative game consists of

- A set of players $N = \{1, 2, \ldots, n\}$
- For every player i, a set of available strategies S^i
- For every player i, a utility function
  $u_i: S^1 \times \ldots \times S^n \rightarrow \mathbb{R}$

- Strategy profile: any vector in the form $(a_1, \ldots, a_n)$, with $a_i \in S^i$ (a_i is a valid action of player i)
  - Every strategy profile corresponds to an outcome of the game
  - The utility function describes the benefit/happiness that a player derives from the outcome of the game

Example 1: Bach or Stravinsky (BoS)



vs



- 2 players, a man and a woman
- 2 concerts of classical music, one dedicated to Bach, and one to Stravinsky
- The man prefers Bach, the woman prefers Stravinsky
- Both the man and the woman prefer to go somewhere together than go alone to a concert

Example 1: Bach or Stravinsky (BoS)





	B	S
B	2, 1	0, 0
S	0, 0	1, 2

- Matrix representation of the utility functions
- Any representation is acceptable as long as
  - $u_1(B, B) > u_1(S, S)$
  - $u_1(S, S) > u_1(S, B)$, $u_1(S, S) > u_1(B, S)$
  - Similarly for pl. 2
  - Not important that $u_1(B, S) = u_1(S, B)$, it suffices that both are less than $u_1(S, S)$
- The game is also known as "Battle of the Sexes"

Example 2: Rock-Paper-Scissors

- $S^1 = S^2 = \{R, P, S\}$
- A 0-sum game

	R	P	S
R	0, 0	-1, 1	1, -1
P	1, -1	0, 0	-1, 1
S	-1, 1	1, -1	0, 0

Nash equilibria



- Definition for 2 player games (Nash 1950): A strategy profile (a_1, a_2) is a Nash equilibrium, if no player has a unilateral incentive to deviate, given the other player's choice
- I.e., the following conditions should be satisfied:
  1. $u_1(a_1, a_2) \geq u_1(a', a_2)$ for every strategy $a' \in S^1$
  2. $u_2(a_1, a_2) \geq u_2(a_1, a')$ for every strategy $a' \in S^2$

Example 1: Bach or Stravinsky (BoS)





	B	S
B	2, 1	0, 0
S	0, 0	1, 2

2 Nash equilibria:

- (B, B) and (S, S)
- Both derive the same total utility (3 units)
- But each player has a preference for a different equilibrium

Pictorially:

a_1 \ a_2					
	(,)	(,)	(x_1,)	(,)	(,)
	(,)	(,)	(x_2,)	(,)	(,)
	(,)	(,)	(x_3,)	(,)	(,)
	(,y_1)	(,y_2)	(x, y)	(,y_4)	(,y_5)
	(,)	(,)	(x_5,)	(,)	(,)

In order for (a_1, a_2) to be a Nash equilibrium:

- x must be greater than or equal to any x_i in that column
- y must be greater than or equal to any y_j in that row

Games with more than 2 players

- The definition generalizes easily to any number of players
- The rationale is still that a player should not have an incentive to change her strategy

Some notation:

- Given a strategy profile $a = (a_1, \ldots, a_n)$, let a_{-i} the vector where we have removed the i-th coordinate:

$$a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$$

- E.g., if $a = (3, 5, 7, 8)$, then
  - $a_{-3} = (3, 5, 8)$
  - $a_{-1} = (5, 7, 8)$

Games with more than 2 players

- For any i, we can rewrite a strategy profile as $a = (a_i, a_{-i})$
- A profile $a = (a_1, \ldots, a_n)$ is a *Nash equilibrium*, if no player has a unilateral incentive to deviate, given the other players' choices
- I.e., for every player i, it should hold that:

$$u_i(a_i, a_{-i}) \geq u_i(a', a_{-i}) \text{ for every strategy } a' \in S^i$$

End of game theory detour

Shapley vs proportional sharing

Back to pool formation

- Basic (atomic) pool formation game
  - Set of players = {1, 2,…, n}
  - Each player chooses whether to open her own pool or delegate her stake to someone else's pool
  - Strategy space ≡ {1, 2,…,n}
    - If player i chooses herself: she is opening a pool
    - Otherwise: she joins someone else's pool
    - Valid strategy profiles: delegators choose a player who opens a pool
  - Possible outcomes of the game: every valid profile induces a partition into pools

Game-theoretic analysis

We focus on the case that $s_i < h$, for every player i:

- We are interested in players who need to collaborate with others in order to be part of a winning pool
- Much larger players would most likely split their stake and run their own pools of size ≈h

Definition: Given a reward scheme, a partition (C_1, C_2,…, C_m) is a Nash equilibrium if no player has a unilateral incentive to move her stake to another pool

Benchmark

Optimal decentralized formation:

OPT = maximum number of pools we can make with total stake at least h each

- Known as the Dual Bin Packing problem (or Bin Cover)
- Known to be NP-complete [Assmann et al. ’84]

What we are interested in:

- Are there good equilibria w.r.t. decentralization?
  - Counting number of winning pools formed at equilibrium
- Sybil resilience of equilibrium formations?

An example

Consider a game where

- Threshold h=4
- 6 players, with stakes: 3, 1, 1, 1, 1, 1

➢ Optimal partition: OPT = 2

- C_1 = the large player with a small one, C_2 = the remaining small ones together

➢ Under proportional sharing

- (C_1, C_2) is a Nash equilibrium

➢ Under the Shapley scheme:

- Not an equilibrium! (why?)



Price of Anarchy vs Price of Stability

- If a game has multiple equilibria, which one is more likely to occur?
  - No clear answer…
- Pessimistic view: look at the worst equilibrium w.r.t. underlying objective ⇒ Price of Anarchy (PoA) [Koutsoupias Papadimitriou '99]
- Optimistic view: look at the best equilibrium ⇒ Price of Stability (PoS) [Anshelevich et al. '04]

Price of Anarchy vs Price of Stability

- Consider a family of games $\mathcal{P}$
- For every game $G \in \mathcal{P}$:
  - Let $W_{min}(G)$ = minimum number of winning pools at an equilibrium of G
  - Respectively $W_{max}(G)$ = maximum number of winning pools
- Then, for the family $\mathcal{P}$

$$PoA = \sup_{G \in \mathcal{P}} OPT/W_{min}(G)$$

$$PoS = \sup_{G \in \mathcal{P}} OPT/W_{max}(G)$$

Ideally, we would like to have PoA = 1 or O(1)

Price of Anarchy vs Price of Stability

Fact: For pool formation games, the Price of Anarchy can be unbounded ($\Omega(OPT)$)

- The grand coalition is an equilibrium
- But a very unnatural one! Unlikely that this would be the outcome of selfish behavior

We instead focus on the Price of Stability

Shapley vs proportional sharing

[Kiayias, Koutsoupias, Markakis, Tsamopoulos AFT '25]

We have considered 2 types of games

- Atomic, as defined earlier
- Oceanic pool formation games
  - Oceanic games [Milnor, Shapley '78]
  - Volume of infinitesimally, non-atomic small players (the ocean) and finite number of bigger (atomic) players with finite stake each
  - Advocated for blockchain populations [Leonardos, Leonardos, Piliouras '19]

Shapley vs proportional sharing

Shapley value in the oceanic model [Milnor, Shapley '78]:

- Think of the non-atomic players in a pool as forming an interval [0, L]
- Consider random arrival times for the atomic players in [0, L]
- Let t_i be the arrival time of player i

Shapley value of player i =

Probability of being pivotal, for the pool that consists of

- The mass $[0, t_i]$
- The atomic players who arrived before t_i

Our main results

Theorem 1: For any oceanic or atomic game, the proportional sharing scheme always has an optimal equilibrium partition

- i.e, *Price of Stability = 1*
- Proof during the tutorial

Theorem 2: Consider oceanic games under the Shapley scheme, where

- for each large player, $s_i \geq h/4$
- there is a sufficiently large mass of small players

Then, there always exists an equilibrium where

number of pools ≥ ¾ OPT

- i.e., *Price of Stability ≤ 4/3*, and this is tight

Our main results

Theorem 3: Consider atomic games under the Shapley scheme, where for every player, either $s_i = 1$ or $s_i = a<h$. Then
Price of Stability ≤ 2 and this is tight

- Shapley value much harder to analyze for atomic games
- We conjecture PoS still remains O(1) for more general atomic games

Proof outline of Theorem 2

- For every game, we construct an equilibrium with the desired properties
- Proposed partition:
  - Each atomic player forms a pool only with a mass of non-atomic players
  - Remaining mass of non-atomic players also splits into pools
  - Total size of each pool ≈ 4/3h

(tight) example

Consider a game where

- Threshold h=3
- Every large player has stake a=2

➢ Optimal partition: each large player forms a pool with a mass of 1 unit of small players

➢ Equilibrium partition:

- Every large player forms a pool with 2 units of small players (pools of total stake = 4)
- All remaining small players form pools of size 4



Sybil resilience

- Blockchain systems are anonymous environments
- Feasible to split your stake into several wallets
- Controlling or participating in multiple pools can lead to gaining power (e.g. voting power) and to less decentralization
- Sybil attack: finding a way to split one's stake so as to gain more profit
- Generally hard to completely avoid Sybil attacks [Kwon et al. '19]

Sybil resilience

Q: What happens to equilibrium partitions when we allow Sybil attacks?

Theorem 3: The equilibria of the proportional scheme are not always Sybil resilient

- Can be illustrated with simple examples

On the other hand:
Theorem 4: For the Shapley scheme, the equilibria identified earlier are Sybil resilient

Sybil attacks under proportional sharing

Consider a game with

- 3 players with stake h/2 + 1
- h/2-1 players with 1 unit of stake

Optimal equilibrium:



Sybil attack: One of the 2 large players moves 1 unit of stake to the 2nd pool

$$h/2/(h+1) + 1/(h+1) > 1/2$$

Part 1b: The Cardano mechanism for on-chain pooling

Recall: onchain vs offchain pooling

In onchain pooling, the protocol needs to specify the entire reward mechanism:

1. total available reward per epoch, to be distributed to the pools
   - Coming from blocks and transaction fees
2. Reward sharing scheme at system level: how much reward is given to each pool in every epoch?
3. Reward sharing scheme at pool level: how is the pool reward split among the pool members?

In offchain pooling

1. The protocol specifies reward to each individual validator
   - Coming from blocks and transaction fees
2. Reward sharing scheme at pool level:
   - Determined among the validator and the remaining pool members
   - Or determined by the mediator (if one exists)

Onchain pooling

A format initiated by [Brunjes, Kiayias, Koutsoupias, Stouka '20]:

- Before the beginning of an epoch, operator *i* specifies
  - Her pledge λ_i = stake she allocates to the pool
    - may differ from the total stake in her possession
  - Her cost c_i per epoch (accounting for time, effort, electricity,...)
  - Her margin m_i = fraction of reward to be kept for herself

- A delegator needs to specify
  - The pool(s) to which she wants to delegate her stake
  - The amount of stake delegated to each pool

Reward sharing at pool level

- Suppose a pool S, receives a reward ρ(S) at an epoch
- Employ proportional sharing after the operator takes a cut
- The operator will keep for herself
  - her cost c_i
  - her margin from the remaining reward: m_i (ρ(S)-c_i)
- Left-overs: (1- m_i) (ρ(S)- c_i)
  - to be split among the delegators (incl. the operator) proportionately to the stake they contributed

Example



- Consider a pool with 3 members
  - Operator's pledge = s_1 = 300
  - s_2 = 150
  - s_3 = 50
- Suppose the operator submits c_1 = 3, m_1 = 10%
- Let ρ(S) = 100
- Payoff allocation
  - The operator gets: 3 + 0.1*97 + 3/5*87.3 = 65.08
  - Player 2 receives 26.19
  - Player 3 receives 8.73

Reward sharing at system level

- How shall we determine the reward to each pool?
- We could simply reward in proportion to the number of blocks produced by each pool
- BUT: also want to discourage Sybil attacks and provide “skin in the game”

Reward sharing at system level

- Some suggestions:
  - The reward of a pool should depend on
    - total stake of its pool members, capped at some threshold (for decentralization)
    - pledge of the pool operator (to deter Sybil attacks)
  - $\rho(\sigma, \lambda)$ = reward given to a pool with total stake σ and operator's pledge λ

A general class of reward sharing schemes

- $(s_1,\ldots, s_n)$: stake vector over all involved players
- Suppose $\Sigma_j s_j = 1$
- R = total reward to be distributed to the pools
- Consider a pool S, with pledge λ and total stake $\sigma = \Sigma_{j\in S} s_j$
- One could focus on formulas of the form:

$$\rho(\sigma, \lambda) = \frac{R}{\tau}(\sigma' + g(\sigma', \lambda'))$$

Where

- τ = normalization factor,
- σ' = min{σ, β}, with β = saturation threshold
- λ' = min{λ, β}

Rationale:

Linear reward (capped at β) + pledge-dependent bonus

- How about ρ analogous to σ' + αλ' ?

The reward formula of Cardano

Proposed in [Brunjes, Kiayias, Koutsoupias, Stouka '20]:

$$\rho(\sigma,\lambda) = \frac{R}{1+\alpha}\left(\sigma' + \alpha\cdot\lambda'\frac{\sigma' - \lambda'(1-\frac{\sigma'}{\beta})}{\beta}\right)$$

- α = Sybil parameter, currently set to 0.3

Note: For a saturated pool: $\rho(\sigma,\lambda) = [R/(1+\alpha)] \cdot (\beta+\alpha\lambda)$

Claim: This is a valid reward scheme
i.e., sum of rewards over all pools ≤ R

- we do not always allocate all of R

Reward sharing schemes

- In practice:
  - $\rho(\sigma, \lambda)$ is weighed by a performance factor
  - measuring how close the pool performed against the number of blocks it was "expected" to produce
  - For pool j, $f_j = p_j/e_j$
    - p_j = actual number of produced blocks
    - e_j = expected number of produced blocks for pool j, based on stake distribution
  - Final reward: $f_j \cdot \rho(\sigma, \lambda)$

- In theory we may assume performance factor = 1
- But in practice there may be a big variance

Reward sharing schemes

- Every choice for a formula $\rho(\sigma, \lambda)$ induces a game, which corresponds to a pool formation process
- We are interested in the same questions as in Part 1a
- Given a reward scheme via $\rho(\sigma, \lambda)$
  - Q1: does the corresponding game have equilibria?
  - Q2: if yes, do there exist "desirable" equilibria?

The reward formula of Cardano

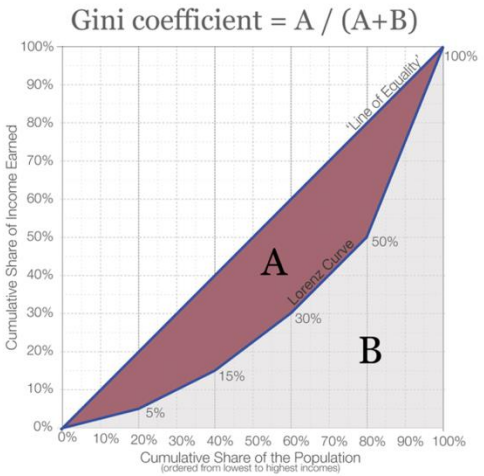
Main theoretical results from [Brunjes et al. '20]

- Study of games with non-myopic players
  - Utility estimated w.r.t. their future anticipation
  - i.e., if a pool is competitive against the other pools, it shall get saturated in the near future
- Price of Stability type of analysis:
  - There exists a Nash equilibrium with $k=1/\beta$ saturated pools
- Lower bounds on required stake to perform Sybil attack ($\Omega(k)$ stake to control $k/2$ pools at equilibrium)

Experimental evaluation

Decentralization metrics

- Nakamoto coefficient
  - Minimum number of distinct entities that control k/2+1 pools
  - Popular metric in blockchain systems
- Gini coefficient
  - Measuring the inequality in the distribution of wealth



Experimental evaluation

[Ovezik, Kiayias ’22]:

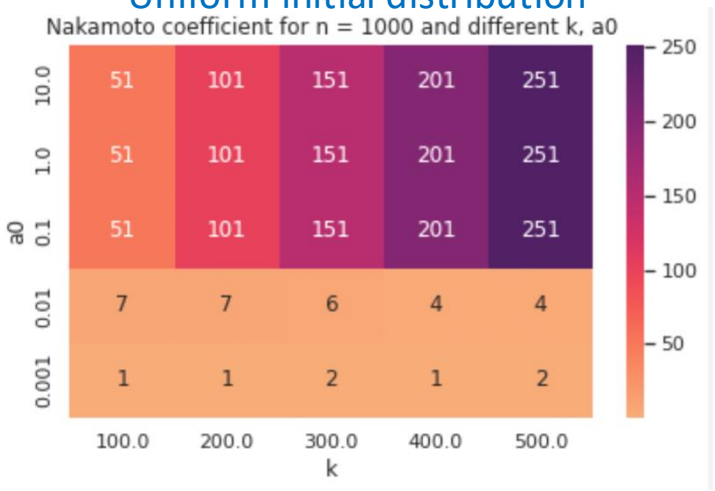
- A study of approximate best response dynamics
- Features:
  - Players take turns in an asynchronous manner
  - Multiple pools allowed: players can create more than 1 pool, if profitable to do so
- Almost always converges to an equilibrium with k pools (as the one identified in theory)

Experimental evaluation

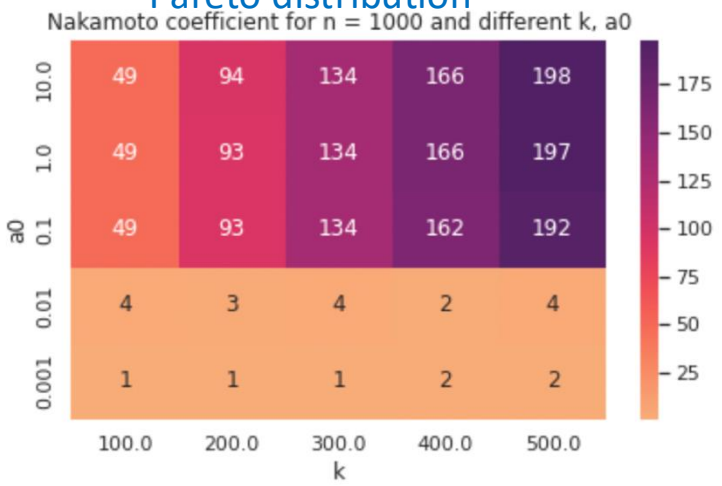
[Ovezik, Kiayias ’22]:

- Promising Nakamoto coefficient at the outcome of the simulations
  - Phase transition induced by the Sybil parameter
  - Ideal Nakamoto coefficient under uniform stake distribution
  - Not too far off under Pareto distribution

Uniform initial distribution



Pareto distribution



Conclusions and current research

- Blockchains introduce different angles for studying traditional reward sharing notions from cooperative games
  - Mainly due to the concerns for decentralization and Sybil attacks
- Are there better alternative reward sharing schemes?
  - Either at system-level or at pool-level?
- Should protocols support onchain pooling or not?
  - Or does the invisible hand of Adam Smith steer offchain pooling to good equilibria?
  - So far, Cardano seems more decentralized than Ethereum
- Better modelling of the heterogeneous population of users (small stakeholders, whales, cryptocurrency exchanges, …)
  - Game-theoretic models view players as (monetary) utility maximizers
  - But many operators care for popularity or for controlling more pools (leads to enhanced voting power too)