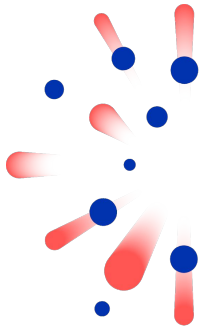


CARDANO

VIRTUAL SUMMIT 2020

SHELLEY EDITION

Delegation and stake pools 2

Incentives

WHY SIMULATIONS?

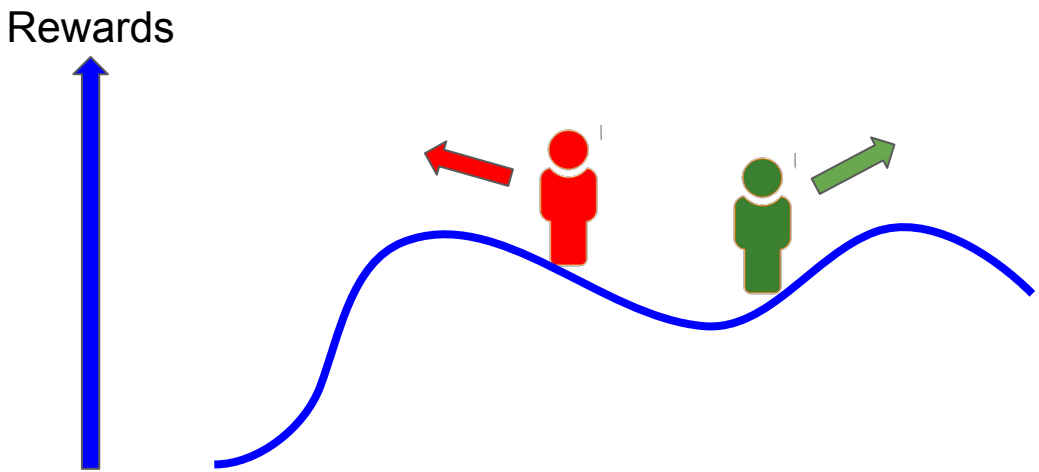
> *In theory, there is no difference between theory and practice. But in practice, there is.*

- By Benjamin Brewster

Convergence

- Understanding the Nash-Equilibria of a “game” is crucial when trying to understand the game.
- We know all Equilibria of the Cardano RSS are “nice”: **k pools of equal size**.
- But will the system reach an equilibrium?
- If so, how fast?
- This is the question of **convergence**.

What could go wrong?



- **Red** wants to go left to increase his rewards.
- **Green** pulls right instead.
- If they moved together, they could reach an equilibrium, but they keep dragging each other down.

Simulation

- We have studied the “staking game” by **simulating** it.
- We first generate a random population of players.
- Then we let the players move, each trying to get higher rewards.
- Hopefully, this will lead to an equilibrium.

SAMPLING

Generating random players

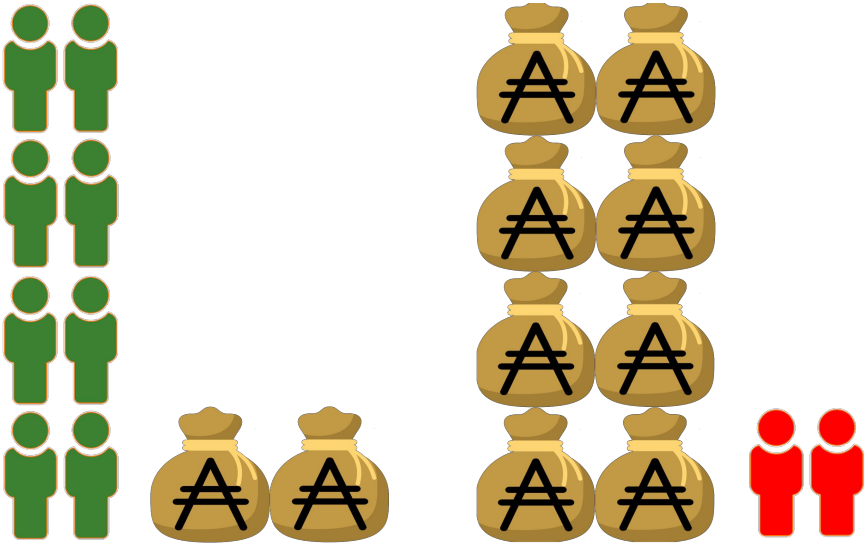
- Each player has random **stake** and random **costs**.
- We pick **costs** uniformly from an interval $[c_{min}, c_{max}]$.
- For **stake**, we try to be more realistic by using a **Pareto distribution**.
- This captures the empirical observation that wealth is not distributed equally, but rather follows an “80-20”-rule.

Pareto Distribution

80% of people

have

20% of stake.



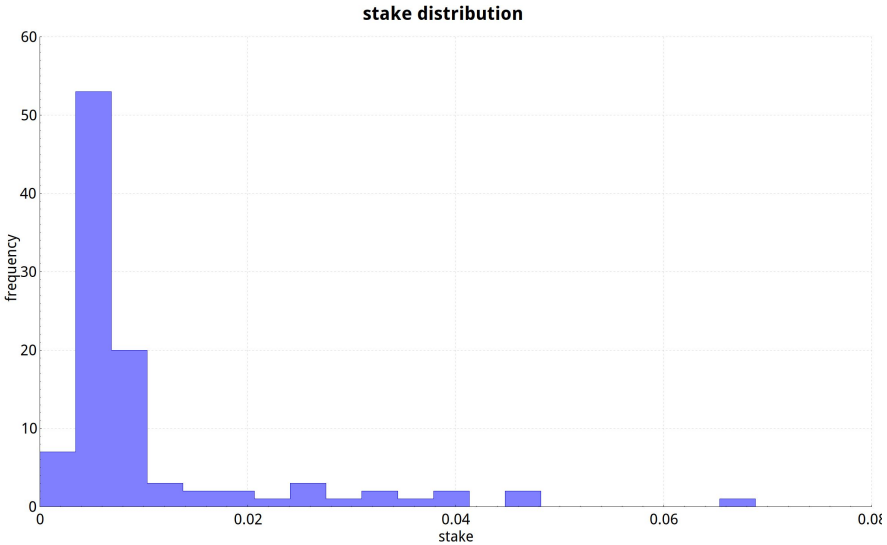
20% of people

have

80 % of stake.

Sample Distribution

A lot of people have little stake.



Few people have a lot of stake.

THE ALGORITHM

Initial strategy

- After players have been generated, they each pick an **initial strategy**.
- For most of our simulations, they initially simply don't do anything - neither run a pool nor delegate.
- Alternatively, they could all delegate to one big pool...
- ...or do something random…
- ...or even start nicely organized into k pools of equal size.

Simulation steps

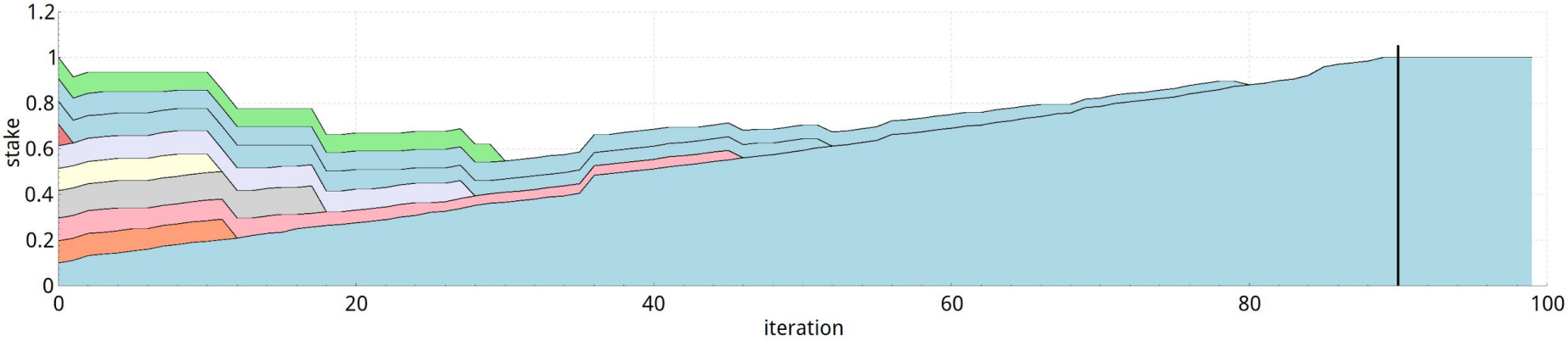
1. Pick a random player.
2. Check whether that player can increase his rewards by changing his strategy.
3. If yes, he changes the strategy, and we go back to step 1.
4. If not, we try another player.
5. If no player has a profitable move, we stop.

SIMULATING “NAIVE” RSS

Trying the “naive” RSS

- We simulated a “proportional” RSS, where pool rewards are simply proportional to pool stake.
- We tried it with different initial strategies.
- Even if we start with k pools of equal size, things go very wrong very quickly...

Simulated “naive” RSS



Result for the “naive” RSS

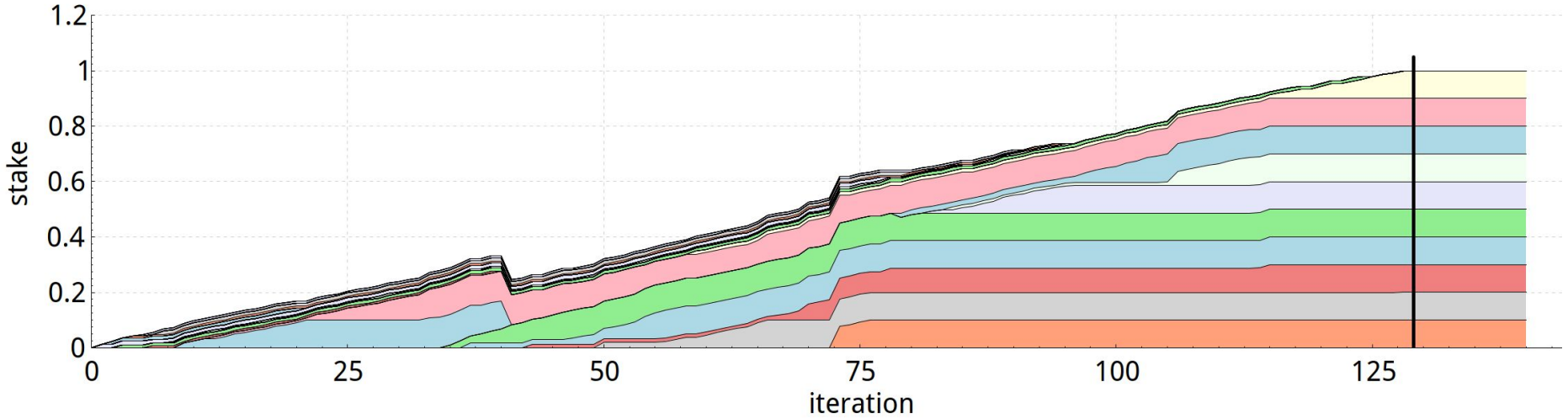
- The “naive” RSS converges quickly in less than 100 steps.
- We start with k pools of equal size.
- We quickly end up with **one big dictatorial pool**.

SIMULATING Cardano RSS

Trying the Cardano RSS

- We ran most simulations with all players doing nothing initially.
- Trying other initial strategies doesn't change the simulation result.
- We always reach a "nice" equilibrium with **k pools of equal size** quickly.

Simulated Cardano RSS



Not just pretty pictures...

Pools

player	rank	cost-rank	stake-rank	actual cost	declared cost	margin	player stake	pool stake	reward	desirability
1	4	54	1	0.00156856	0.00156856	0.00898774	0.07704926	0.10000000	0.10154099	0.099073896587544
2	5	19	5	0.00121229	0.00121229	0.00125302	0.02052438	0.10000000	0.10041049	0.099073896587539
3	9	5	17	0.00108188	0.00108188	0.00088317	0.01216771	0.10000000	0.10024335	0.099073896587511
4	3	16	7	0.00120205	0.00120205	0.00063505	0.01694531	0.10000000	0.10033891	0.099073896587553
5	2	6	26	0.00108805	0.00108805	0.00053598	0.01075376	0.10000000	0.10021508	0.099073896587558
6	1	1	81	0.00100213	0.00100213	0.00047005	0.00613080	0.10000000	0.10012262	0.099073896587589
7	7	3	39	0.00105867	0.00105867	0.00047469	0.00898080	0.10000000	0.10017962	0.099073896587522
8	6	18	8	0.00121088	0.00121088	0.00042690	0.01635433	0.10000000	0.10032709	0.099073896587534
9	8	2	62	0.00103849	0.00103849	0.00026601	0.00693720	0.10000000	0.10013874	0.099073896587515
10	10	12	16	0.00115913	0.00115913	0.00011986	0.01224503	0.10000000	0.10024490	0.099073896587504

We also get detailed information about the equilibrium...

...which enables us to study things like Sybil protection.

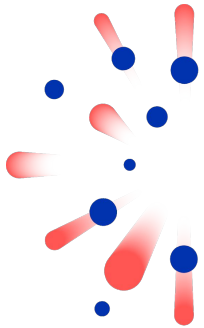
Pools

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

Result for the Cardano RSS

- We have run these simulations hundreds of times for a range of different settings.
- We tried various initial strategies, Pareto parameters and cost intervals.
- Details differ (it's random, after all!), but we always reach a nice equilibrium eventually.

THANK YOU FOR YOUR ATTENTION!



CARDANO
VIRTUAL SUMMIT 2020
SHELLEY EDITION