Trust Evolution Game in Blockchain

Hamda Al Breiki
College of Technological Innovation
Zayed University
Abu Dhabi, UAE
hamda.albreiki@zu.ac.ae

2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA) | 979-8-3503-1008-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/AICCSA56895.2022.10017651

***Abstract*—One of the main concepts behind Blockchain is trustlessness, where trust is shifted to a new paradigm through Cryptoeconomics. But how can people trust a trustless system? Blockchain protocols employ incentive structures predicated on game theory mechanisms in order to encourage the players (users and miners) in the system to act honestly. Users don't need to trust any single entity and there is no single point of failure that the system relies on(trustless system). In this paper, we developed an abstract game model for blockchain. The game model was for a repeated matrix game to be played by blockchain users. We studied how the game design enforces players to cooperate with each other to benefit the whole players in the network, which increase trust and make the system more trustworthy. Also, we applied two basic learning algorithms to see how the players of the game will evolve over time. We found that the players learn to cooperate in the game to get better payoffs.**

***Index Terms*—trust, blockchain, game theory, evolution game, trustless.**

I. Introduction

Trust is at the core of any economic system evolved and used by humankind. The form of trust changes over time. The recent technological and economic developments introduced a new paradigm shift for trust through Cryptoeconomics. Cryptoeconomics can be defined as a combination of cryptography, economics, and game theory incentive models incorporated into distributed blockchain protocols in order to create a secure, stable, and sustainable system [1]. Blockchain protocols employ incentive structures predicated on game theory mechanisms in order to encourage the players (users and miners) in the system to act honestly. Users don't need to trust any single entity and there is no single point of failure that the system relies on(trustless system). But how can people trust a trustless system? In this paper, we will have a first look into the trust evolution game in the blockchain system. We will study this game from an abstract level and try to apply part of what David Axelrod did in his famous book The Evolution of Cooperation, in the context of blockchain game [2]. According to David Axelrod: "Mutual cooperation can emerge in a world of egoists without central control by starting with a cluster of individuals who rely on reciprocity."

II. Problem Definition

In this paper, we applied multiagent and game theory concepts to the blockchain system. The main question we want to answer is: **how does trust among players evolve while playing in the trustless decentralized repeated game?** The players in blockchain are the users/miners. The users in this system do not trust each other, and they do not know each other. The objective for those players is to increase their chance of receiving the incentive of mining new blocks in a trustless decentralized system.

III. Related Literature

In 2008, Satoshi Nakamoto published a paper that discussed the need for an electronic payment system to solve the weakness of trust-based payment model that relies on a trusted third party and replace it with a cryptographic proof-based system. Nakamoto proposed a system for electronic transactions without relying on trust. The proposed system relies on a peer-to-peer network where payments are sent directly from one party to another using proof-of-work to record a public history of transactions. Nakamoto incorporated incentive models from game theory to encourage nodes to stay honest while using the system. The steps to run the network for the proposed system were summarized by [3] as follows:

1) New transactions are broadcast to all nodes
2) Each node collects new transactions into a block
3) Each node works on finding a difficult proof-of-work for its block
4) When a node finds a proof-of-work, it broadcasts the block to all nodes
5) Nodes accept the block only if all transactions are valid and not already spent
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash

Game theory is a core concept behind blockchain systems that is often overlooked. Incentive models from game theory are incorporated into distributed blockchain protocols in order to create a secure, stable, and sustainable system. In fact, blockchain did not eliminate the need for trust in the system, instead, it enforced trust via effective self-policing mechanisms from game theory. Different papers were published about different aspects of game theory related to blockchain. We will briefly discuss some of these papers.

Houy introduced and studied the Bitcoin mining game, in which miners make a decision regarding how many transactions they should include in the block they are mining. The paper focused on the mining incentives in the Bitcoin protocol[4]. Lewenberg et al. examined the dynamics of

pooled mining and the rewards that pools manage to collect. The authors used cooperative game theoretic tools to analyze how pool members may share these rewards. They showed that for some network parameters, especially under high transaction loads, it is difficult or even impossible to distribute rewards in a stable way: some participants are always incentivized to switch between pools[5].

Kiayias et al. studied the strategic considerations of miners participating in the bitcoin protocol. They considered two simplified forms of this game in which the miners have complete information. In the first game, miners release every mined block immediately but are strategic on which blocks to mine. In the second more complicated game, when a block is mined it is announced immediately, but it may not be released so that other miners cannot continue mining from it. In both games, the authors show that when the computational power of each miner is relatively small, their best response matches the expected behavior of the bitcoin designer. However, when the computational power of a miner is large, he deviates from the expected behavior, and other Nash equilibria arise[6].

Singh et al. considered a continuous time dynamic game model of bitcoin mining. The authors propose two types of solutions to the proposed model: cooperative (social optimum) mining strategy, and non-cooperative (Nash equilibrium) mining strategy. Authors found that it is always beneficial for the miners to consume or use the electricity jointly, in cooperation with the other miners. Cooperation gives the miner a higher total profit compared to a miner who mines selfishly. Moreover, if all the miners choose to mine according to the Nash equilibrium mining strategy, then the Electricity will deplete much faster than if they choose to mine according to the social optimum strategy. The authors also proposed a tax system in order to enforce social optimality in our bitcoin dynamic game model. This way, miners will be forced to behave or mine in a way that is best for the social welfare of the miners[7].

Liu et al. studied the dynamics of mining pool selection in a blockchain network, where mining pools may choose arbitrary block mining strategies. they identify the hash rate for puzzle-solving and the block propagation delay as two major factors determining the mining competition results. the authors modeled the strategy evolution of individual miners as an evolutionary game. They provided the theoretical analysis of evolutionary stability in the pool selection dynamics for a two-pool case[8].

Alzahrani and Bulusu proposed a decentralized consensus protocol that does not require PoW and randomly employs a different set of different sizes of validators on each block's proposal. The proposed protocol uses a game theoretical model to enforce the honest validators' behavior by rewarding honest validators and penalizing dishonest ones[9]. Dey proposed a methodology where we can use intelligent software agents to monitor the activity of stakeholders in the blockchain networks to detect anomalies such as collusion, using supervised machine learning algorithm and algorithmic game theory and stop the majority of attacks from taking place[10].

U(cooperate, cooperate)

- If both players are cooperating in the game at time t:
  - v increases → Market value ($v_t > v_{t-1}$)
  - d minimized → the propagation delay is at minimum
  - With p_i player i will receive r_t + tf, mining reward decrease overtime in a constant rate
  - Player i will consume c, c increases over time ($c_t > c_{t-1}$)

$$U(i)=p_i v_t(r_t+tf) - c_t$$

Fig. 1. Utility Function (Cooperate, Cooperate)

IV. Methodology

We started by defining an abstract model for the blockchain game and we defined all the parameters related to this model. After defining our model, we defined a utility function to calculate players' expected payoffs at each state of the game. Then we defined a general payoff matrix for the game, where we specify the properties for each state payoff for each player. After preparing our game model and the payoff matrix, we implemented two learning algorithms to be used for playing the game. These algorithms are fictitious play and satisficing learning. We conducted a round-robin tournament among fictitious play, satisficing learning, and always defect algorithm (which represents an attacker's behavior in the blockchain network). The results of the round-robin were used to run an evolutionary tournament.

V. Game Model

In this paper we define an abstract version of blockchain game as a repeated matrix game with the following parameters:

- n → miners/players
- A → actions set (Strategies):
  - cooperate: honest mining/validation for blocks
  - defect: dishonest mining/validation for blocks
- p → (p1, . . . , pn) the probabilities that miners succeed in solving the crypto-puzzle; these are proportional*to* their computational power and they sum up to 1
- c → computation cost
- r → mining reward
- tr → transaction fee
- v → market value
- d → propagation delay

Using the parameters defined for our game model, we generate utility functions to find players' expected payoff for each game state.

A. Payoffs Matrix

We will develop an abstract payoffs matrix for the game model we defined using the utility function for each player in each state. Table I shows the payoffs matrix for our blockchain game, where:

- i is the row player, j is the column player

U(cooperate, defect)

- If one player cooperate and the other player defect in the game at time t:
  - v decrease → Market value ($v_t < v_{t-1}$)
  - d increased → the propagation delay increased
  - With p_i player i (who cooperated) will receive r_t + tf, mining reward decrease overtime in a constant rate. Player j, who defected will not receive any incentive
  - Both players i and j will consume c, c increases over time ($c_t > c_{t-1}$)

$U(i)=p_i v_t(r_t+tf) - c_t$
$U(j)=-c_t$

Fig. 2. Utility Function (Cooperate, Defect)

U(defect, cooperate)

- If one player defect and the other player cooperate in the game at time t:
  - v decrease → Market value ($v_t < v_{t-1}$)
  - d increased → the propagation delay increased
  - Player i, who defected will not receive any incentive. With p_j player j (who cooperated) will receive r_t + tf, mining reward decrease overtime in a constant rate.
  - Both players i and j will consume c, c increases over time ($c_t > c_{t-1}$)

$U(i)=-c_t$
$U(j)=p_j v_t(r_t+tf) - c_t$

Fig. 3. Utility Function (Defect, Cooperate)

- r stands for reward
- lr stands for lower reward
- p stands for punishment
- lp stands for lower punishment
- r_i , lr_i , lp_i , p_i for row player i
- r_j , lr_j , lp_j , p_j for row player j

TABLE I
BLOCKCHAIN TRUST GAME

	cooperate	defect
cooperate	r_i, r_j	lr_i, lp_j
defect	lp_i, lr_j	p_i, p_j

VI. LEARNING ALGORITHMS

We implemented two learning algorithms which are: fictitious play and satisficing learning. Both algorithms require external parameters to run. In the fictitious play, we define prior beliefs for each action of the opponent player. In satisficing learning, we define initial aspiration level and learning rate.

U(defect, defect)

- If both players are defecting in the game at time t:
  - v decreases → Market value ($v_{t<} v_{t-1}$)
  - d maximize → the propagation delay is at maximum
  - Both players will not receive any incentive
  - Player i will consume c, c increases over time ($c_t > c_{t-1}$)

$U(i)= - c_t$

Fig. 4. Utility Function (Defect, Defect)

	Alyway Defect	Fictitious	Satisficing	TOTAL
Alyway Defect	-5.00	-1.0	-1.04	-7.04
Fictitious	4.00	10.0	10.00	24.00
Satisficing	3.91	10.0	8.65	22.56

Fig. 5. Round-robin Tournament (First Run)

VII. GAME IMPLEMENTATION

Let's assume our game payoff matrix will have the values shown in table II. We implemented the as a round-robin tournament among fictitious play, satisficing learning, and always defect algorithm (which represents an attacker behavior in the blockchain network). The results of the round-robin were used to implement an evolutionary tournament among the same three players. We run the game two times, with different prior beliefs for fictitious players, and aspiration levels and learning rates for satisficing players.

TABLE II
PAYOFFS MATRIX

	cooperate	defect
cooperate	10, 10	4, -1
defect	-1, 4	-5, -5

VIII. RESULTS AND DISCUSSION

A. First Run

Fictitious Player: Prior Believes (cooperate, defect) = (1.5, 3.5)
Satisficing Player: Aspiration level = 5, learning rate = 0.1

The Round-robin tournament was played repeatedly for 100 rounds, and the result for this tournament in the first run is given in figure 5.

After that, we run an evolutionary tournament for 1000 rounds using the results we got from the round-robin tournament. The performance of the different learning algorithms are illustrated in figure 6

B. Second Run

Fictitious Player: Prior Believes (cooperate, defect) = (2.5, 0.5)
Satisficing Player: Aspiration level = 4, learning rate = 0.7

The Round-robin tournament was played repeatedly for 100 rounds, and the result for this tournament in the first run is given in figure 7.
After that, we run an evolutionary tournament for 1000 rounds

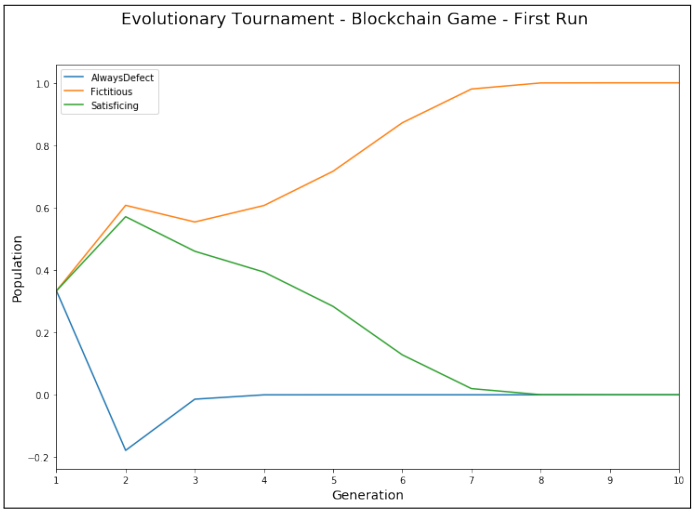

Fig. 6. Evolutionary Tournament (First Run)

	Alyway Defect	Fictitious	Satisficing	TOTAL
Alyway Defect	-5.00	-1.04	-1.0	-7.04
Fictitious	3.91	10.00	10.0	23.91
Satisficing	4.00	10.00	-5.0	9.00

Fig. 7. Round-robin Tournament (Second Run)

using the results we got from the round-robin tournament. The performance of the different learning algorithms are illustrated in figure 6

Changing the learning rate and aspiration level, affected the way the player learns in the game. With a lower learning rate the player was learning better and increased his payoffs.

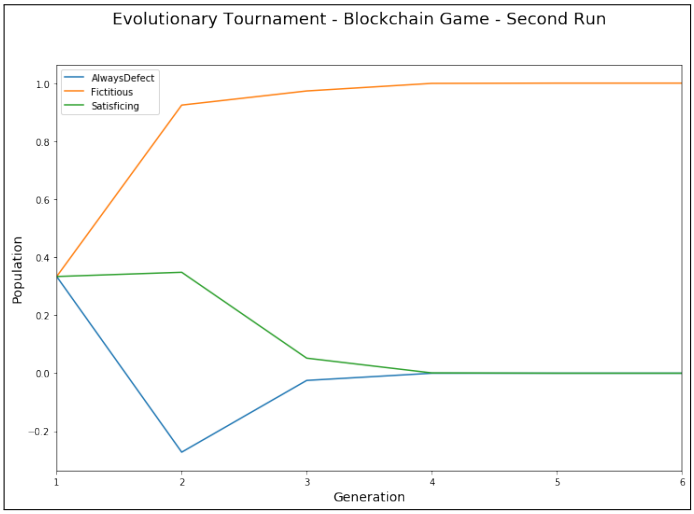

Fig. 8. Evolutionary Tournament (Second Run)

IX. Conclusion

In this paper, we developed an abstract game model for blockchain. The game model was for a repeated matrix game to be played by blockchain users. We studied how the game design enforces players to cooperate with each other to benefit the whole players in the network, which increase trust and make the system more trustworthy. Also, we applied two basic learning algorithms to see how the players of the game will evolve over time. We found that the players learn to cooperate in the game to get better payoffs. For satisfice players, changing the learning rate affected the final payoff the player received. With a lower learning rate, the player got better payoffs. The game model in this paper was very abstract, we did not use real values for the different parameters used in the game model for creating the payoffs matrix. The work can be extended to cover a real case study for an application on blockchain like Bitcoin. Also, more learning algorithms can be implemented to study players' behaviors in the blockchain and how trust is evolving among them.

References

[1] B. Curran. (2018) What is game theory? and how does it relate to cryptocurrency. [Online]. Available: https://blockonomi.com/game-theory/

[2] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *science*, vol. 211, no. 4489, pp. 1390–1396, 1981.

[3] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[4] N. Houy, "The bitcoin mining game," 2014.

[5] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein, "Bitcoin mining pools: A cooperative game theoretic analysis," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 919–927.

[6] A. Kiayias, E. Koutsoupias, M. Kyropoulou, and Y. Tselekounis, "Blockchain mining games," in *Proceedings of the 2016 ACM Conference on Economics and Computation*. ACM, 2016, pp. 365–382.

[7] R. Singh, A. D. Dwivedi, and G. Srivastava, "Bitcoin mining: A game theoretic analysis," 2018.

[8] X. Liu, W. Wang, D. Niyato, N. Zhao, and P. Wang, "Evolutionary game for mining pool selection in blockchain networks," *IEEE Wireless Communications Letters*, 2018.

[9] N. Alzahrani and N. Bulusu, "Towards true decentralization: A blockchain consensus protocol based on game theory and randomness," in *International Conference on Decision and Game Theory for Security*. Springer, 2018, pp. 465–485.

[10] S. Dey, "Securing majority-attack in blockchain using machine learning and algorithmic game theory: A proof of work," *arXiv preprint arXiv:1806.05477*, 2018.