Game of Coins

Alexander Spiegelman
Novi Reseacrh
USA

Idit Keidat
Technion
Israel

Moshe Tennenholtz
Technion
Israel

2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS) | 978-1-6654-4513-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDCS51616.2021.00095

***Abstract*—The cryptocurrency market is blooming. Tens of new coins emerge every year and their total market cap keeps growing. The research community is trying to keep up by proposing improved mining protocols and attacking existing ones. However, surprisingly as it may sound, most existing works overlook the real-life multi-coin market, by focusing on a system with a single coin.**

To the best of our knowledge, this paper was the first to consider a system with many coins and strategic miners that are free to choose where to mine. We first formalize the current practice of strategic mining in multi-coin markets as a singleton weighted congestion game and prove that any better-response dynamics in such a game converges to an equilibrium. Then, in our main result, we present a reward design attack that moves the system configuration from any initial equilibrium to a desired one. The attack is executed via temporary manipulation of coin rewards, which leads strategic miners to switch between coins. It applies to any better-response dynamics of the miners. To motivate our attack we show that in any equilibrium there is always at least one miner whose profit is higher in a different one, and thus may benefit from such an attack.

***Index Terms*—Cryptocurrency, Game Theory, Equilibrium Convergence**

I. Introduction

Cryptocurrencies are an arms race. Hundreds of digital coins have crept into the world market in the last decade [6], including more than a dozen with over a billion dollar market cap, e.g., [1], [3], [9], [10], [14]. The vast majority of cryptocurrencies are based on the notion of *proof of work (PoW)* [38]. As a result, the major strategic players in the cryptocurrency market are *miners* who devote their power to solving computational puzzles to find PoWs [14], [38].

The miners for a particular coin usually gain rewards that are proportional to the power they invest in the coin out of the total invested power (in the coin) by all miners. Each coin can be viewed as having some *weight* that reflects the reward it divides among its miners. In practice, a coin's weight (or reward) depends on its transaction rate, transaction fees, and its fiat exchange rate.

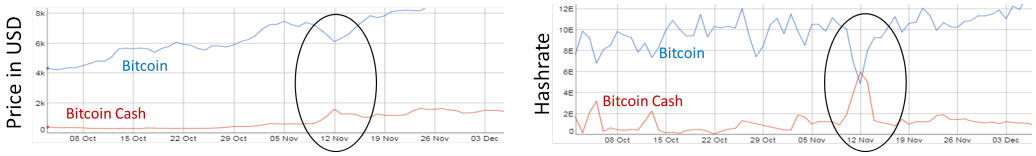
While the above description is not complete, it does capture the fundamental decision faced by a miner: where should I mine? One evidence for the prevalence of reward-based coin switching can be found online in websites like www.whattomine.com [11], where miners enter their mining parameters (technology, power, cost, et cetra) and get a list of coins they can mine for, ordered by their profitability. An interesting example happened on November 12 2017 [5], when a dramatic change in the Bitcoin to Bitcoin Cash [1] (a hard fork from Bitcoin) exchange rate led to a major inrush of miners to Bitcoin Cash (see Figure 1).

All in all, the structure of the cryptocurrency market suggests that we face here a game among miners, where each miner wishes to mine coins with high rewards while avoiding competition with other miners. In this paper we introduce for the first time the study of the cryptocurrency market as a game, consisting of a set of strategic players (miners) with possibly different mining powers and a set of coins with possibly different rewards (weights). The miners are free to choose to mine for any coin from the set, and we consider *general better-response* dynamics of the miners. That is, whenever any miner may benefit from deviating (i.e., changing the coin it mines for), some miner will take a step that improves his payoff; we allow an arbitrary sequence of such individual improvement steps (sometimes called *improving path* [35]). In our first result we prove that any such better response dynamics converges to a pure equilibrium (sometimes called the *finite improvement property* [35]) regardless of miner powers and coin rewards. This result is obtained by showing an *ordinal potential*, which according to [35], implies that an arbitrary better response dynamics converges to an equilibrium.

Having at hand the above result, we move to a discussion of strategic attacks and manipulations [40]. While many efforts have been invested in the study of single-coin attacks [17], [18], [23], [39], [41], we consider for the first time a multi-coin system and introduce an attack that manipulates the miners' optimization process. Given that a shift in the weight of a coin may influence miners to leave or join the coin, it is quite possible for an interested party to affect this weight, either by creating additional transactions with high fees [8] (sometimes called whale transactions [32]) or by manipulating the coin exchange rate [2], [4], [7], [21]. This way, a miner (or another interested party) can attempt to change the system equilibrium to a better one for them by moving players between coins. We show that under broad circumstances, for every equilibrium of such a game, there exists a miner and another equilibrium in which the miner's payoff is higher. The question is therefore: can one attack the system by temporarily increasing coin weights (rewards) in a way that will lead the system from a given equilibrium to a desired one, so that the system will remain in the desired equilibrium after reverting to the original weights?

The above reward design problem is challenging since miners might take *any* better response step, and may make their moves in any order. Nevertheless, having modified the weights, we

2575-8411/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDCS51616.2021.00095



(a) Bitcoin and Bitcoin Cash exchange rates over time. (b) Hashrate corresponds to the number of miners.

Fig. 1: Miners move from Bitcoin to Bitcoin Cash.

can use the previous result to claim that any better response dynamics will converge to an equilibrium. Notice that the latter may not be the desired one, but we can now modify the rewards again. We show that such a reward design (attack) for strategic players is feasible. Namely, we provide a (multi-step) algorithm for assigning rewards in equilibrium states that moves strategic players from any initial equilibrium to a desired one.

In summary, our contributions are as follows:

1) We formalize strategic mining in multi-coin systems as a game (Section II).
2) We prove that any better-response dynamics in such games, starting from an arbitrary configuration, converges to an equilibrium (Section III).
3) We show that, in many cases, for every equilibrium there is a miner and another equilibrium in which the miner's payoff is higher, motivating strategic attacks by manipulation. (Section IV).
4) We introduce the first attack on multi-coin systems by offering a reward design scheme that moves the system configuration from any initial equilibrium to any desired one for any better-response dynamics of the miners (Section V).

A. Related work

a) Game theory in cryptocurrencies: Several previous works have presented game theoretical analyses and attacks on cryptocurrencies [15], [17], [18], [23], [29], [30], [32], [39], [41], [42], [48]. A survey can be found in [33]. However, the vast majority of them deal (in one way or another) with miners' incentives to follow the coins' mining protocols. To the best of our knowledge, our work was the first to extend the study to a multi-coin system, establish fundamental game theoretical results, and present possible strategic attacks therein.

Since our work was first published in Arxiv on May 2018, several works have followed it [12], [23], [31], [34]. Mirkin et al. [34] exploit the reward mechanism of PoW cryptocurrencies to discourage miner participation in order to perform Denial-of-Service attacks. Goren and Spiegelman [23] presented an attack, called smart mining, that strictly dominates honest mining for rational miners. This attack exploits the vulnerability of the difficulty adjustment mechanism, which grows when new miners join the coin. Kwon et al. [31] consider a game with two coins and assume miners possess infinitesimal computational power. Under this assumptions they show that the game either converges to an equilibrium in which both coins are equally profitable or all non-loyal miners move to the more profitable one. We consider a more complex setting with an arbitrary number of coins and miners that cannot infinitesimally divide their power. In addition, we do not only prove that equilibrium always exists, but also prove that under a wide set of assumptions, there is more than one equilibrium and show how a powerful entity can manipulate the system to move among equilibria. Altman et al. [12] followed on our work by extending the model to allow edge computing service providers to support PoW offloading. In addition, they provide efficient algorithms for finding equilibria, which is complementary to our reward design schema that shows how to move the system among them.

b) Congestion games: Monderer and Shapley [35] proved that it is enough to show an ordinal potential in order to prove that every better response dynamics in a game convergences to a pure equilibrium. In addition, they proved in [35] that unweighted congestion games coincide with games that possess exact potentials. We show (in Section III) that our class of games does not possess an exact potential, and thus is not an unweighted congestion game. In fact, our game belongs to a broader family, called *weighted congestion games* [22], [25], where a player's payoff (or cost) depends on its weight and the weights of other players who share the resource, rather than only on the number of players that share the same resource as in unweighted congestion games. More specifically, our game is a *singleton* weighted congestion game, where each player is allowed to choose one resource.

Convergence to a pure Nash equilibrium in weighted congestion games was broadly studied before [13], [16], [19], [20], [22], [25]–[27]. Harks et al. [27] presented conditions on the cost functions in order for every better response dynamic to converge to an equilibrium in weighted congestion games. In particular, they showed that a cost function must be monotonic in order for every better response dynamic to converge to an equilibrium in a general weighted congestion game, and for the non-singleton case, they showed that affine (linear) or certain exponential cost functions are sufficient for convergence. The payoff function in our game is indeed monotonic (as necessary for convergence), but while their result on singleton weighted congestion games does not provide sufficient conditions for convergence, we show that in our game every better response dynamic indeed converges.

Anshelevich et al. [13] consider network design games in which the cost of each edge in the network is divided among all the players that share it, and players try to reduce the total cost on a path between two given nodes. Note that in this

setting, the weights of all players are equal, and the game is not a singleton weighted congestion game since each path may consist of several edges. In our game, in contrast, each miner may have a different mining power (weight) and is allowed to choose one coin (resource).

Even-Dar et al. [16] and Fotakis et al. [19] considered singleton weighted congestion games in the load balancing and selfish routing contexts, respectively. In both of these games, the cost function of a resource (machine or network link) is the the total sum of weights (jobs or traffic) of the players who share it divided by the resource weight (machine speed or link capacity). Both showed (among other results) that their games possess an ordinal potential and thus every better response dynamic converges therein. Our payoff function resembles theirs but is not identical to them. First, we consider payoff functions whereas they consider cost functions and the reduction between the two is usually not trivial. Second, two miners (players) who share the same coin (resource) in our game get different payoffs if they have different weights (mining power), whereas in their game, the cost of a resource is identical for all players who share it. We do, however, use a similar lexicographical technique to prove the existence of an ordinal potential in our game (our first result - Section III).

c) Reward design: Our main result is related to the literature on reward design [24], [44], [45] in machine learning, and to the best of our knowledge, is the first to introduce iterated reward design for strategic players in a multi-round multi-player setting. While seminal works in reward design assign/modify state rewards specifically in a reinforcement learning context [47], we design rewards for equilibrium states for *any* better response dynamics. On a more abstract game-theoretic level, our work complements work on *mediator design*, where a game is modified in order to lead to some desired behavior. The modification can be in the form of constraints on allowed behavior, as in the theory of social laws [43], conditional payment as a function of joint behavior [36], or even a conditional contract [37]. Our work is novel in introducing a multi-round dynamic intervention technique.

II. Model

A *system* in our model is a tuple $\langle \Pi, C \rangle$, where Π is a finite set of n miners (players) and C is a finite set of coins (resources). A miner $p \in \Pi$ has mining power $m_p \in \mathbb{R}_+$, which it can invest in one of the coins, i.e., the set of possible actions of p is C. We denote the *set of configurations* of a system $Q = \langle \Pi, C \rangle$ as $S_Q \triangleq C^n$ and denote by $s.p$ the action of miner $p \in \Pi$ in configuration $s \in S_Q$. When clear from the context, we omit the subscript indicating the system and simply write S. Given $s \in S$ and $c \in C$, we denote by $P_c(s) \subseteq \Pi$ the set of miners who mine for c in s, i.e., $P_c(s) \triangleq \{p \in \Pi \mid s.p = c\}$, and by $M_c(s)$ their total mining power, i.e., $M_c(s) \triangleq \Sigma_{p \in P_c(s)} m_p$. For $s \in S_Q,\ p \in \Pi,\ c \in C$ we denote by (s_{-p}, c) the configuration that is identical to s except that $s.p$ is replaced by c.

A *reward function* $F : C \to \mathbb{R}_+$ maps coins to rewards. A game $G_{\Pi,C,F}$ consists of a system $\langle \Pi, C \rangle$ and a reward function F. Every coin in a game $G_{\Pi,C,F}$ divides its reward among all the players who mine for it, and the miners' payoffs are defined as follows: For $s \in S$, the *revenue per unit (RPU)* of coin c in s is $RPU_c(G_{\Pi,C,F})(s) \triangleq \frac{F(c)}{M_c(s)}$. When clear from the context, we omit the parameter indicating the game. The *payoff function* of a miner $p \in \Pi$ is $u_p(s) \triangleq m_p \frac{F(s.p)}{M_{s.p}(s)} = m_p \cdot RPU_{s.p}(s)$. Table I summarizes the notations and definitions we use in this paper.

Π	set of miners
C	set of coins
$Q = \langle \Pi, C \rangle$	a system
m_p	mining power of miner p
S_Q	set of all conf. in system Q
$s.p$	the coin miner p mines for in conf. s
$P_c(s)$	set of miners that mine for coin c in conf. s
$M_c(s)$	total mining power invested in coin c in conf. s
$RPU_c(s)$	revenue per unit of coin c in conf. s
$u_p(s)$	payoff function of miner p in conf. s

TABLE I: Notations and definitions.

Given a game $G_{\Pi,C,F}$, a configuration $s \in S$, a miner $p \in \Pi$, and a coin $c \in C$, we say that p *moves* from $s.p$ to c in s if it changes its action from $s.p$ to c. A move from $s.p$ to c is a *better response step* for p if $u_p(s) < u_p((s_{-p}, c))$. We say that a miner $p \in \Pi$ is *stable* in a configuration s in game $G_{\Pi,C,F}$ if p has no better response steps in s. A configuration s is *stable* or a (pure) *equilibrium* if every miner $p \in \Pi$ is stable in s. A *better response dynamics* from s in $G_{\Pi,C,F}$ is a sequence of configurations resulting from a sequence of better response steps starting from s, which is either infinite or ends with a stable configuration. In case it is finite, we say that it *converges* to its final configuration.

A function $f : S \to \mathbb{R}$ is an *ordinal potential* for a game $G_{\Pi,C,F}$ if for any two configurations $s, s' \in S$ s.t. some better response step of a miner $p \in \Pi$ leads from s to s', it holds that $f(s) < f(s')$. If, in addition, $f(s') - f(s) = u_p(s') - u_p(s)$, then f is an *exact potential*. By [35], if a game $G_{\Pi,C,F}$ has an ordinal potential, then every better response dynamics converges.

a) Remark.: For simplicity of the presentation, we assume in this paper that coin rewards are fixed and do not depend on the total mining power invested in the coins. However, since the security of cryptocurrencies grows with the total mining power, it is reasonable to assume that the value of a coin – and thus also its reward– increases when the mining power increases and vice versa. As long as the reward moderately increases with the mining power, all our results hold. More specifically, for our results, it suffices to assume the following: Consider two configurations $s^1, s^2 \in S$ s.t. a better response step of some miner p leads from s^1 to s^2 by moving from coin c to c'. Then, $RPU_c(s) \geq RPU_c(s')$ and $RPU_{c'}(s) \leq RPU_{c'}(s')$.

III. Better Response Dynamics Convergence

In this section we prove that although a game $G_{\Pi,C,F}$ has no exact potential (Section III-A), every better-response dynamics of the miners in game $G_{\Pi,C,F}$ converges to a stable

configuration (pure equilibrium) regardless of the sets Π and C and the reward function F (Section III-B). To gain intuition, the reader is referred to the full paper [46], where we show how to construct a particular equilibrium in a game $G_{\Pi,C,F}$ for any Π, C and F, and give a simple ordinal potential function for the symmetric case in which F is a constant function i.e., $\forall c, c' \in C,\ F(c) = F(c')$, respectively.

A. No exact potential

Proposition 1. *The game $G_{\Pi,C,F}$ does not always have an exact potential.*

Proof. Let $G_{\Pi,C,F}$ be a game where $\Pi = \{p_1, p_2\}$, $m_{p_1} = 2, m_{p_2} = 1$, $C = \{c_1, c_2\}$, and $F(c_1) = F(c_2) = 1$. Assume by way of contradiction that $G_{\Pi,C,F}$ has an exact potential function H, and consider the following four configurations:

- $s^1 = \langle c_1, c_1 \rangle$. Payoffs: $u_{p_1}(s^1) = \frac{F(c_1)\cdot m_{p_1}}{m_{p_1}+m_{p_2}} = 2/3$, $u_{p_2}(s^1) = \frac{F(c_1)\cdot m_{p_2}}{m_{p_1}+m_{p_2}} = 1/3$.
- $s^2 = \langle c_1, c_2 \rangle$. Payoffs: $u_{p_1}(s^2) = \frac{F(c_1)\cdot m_{p_1}}{m_{p_1}} = 1$, $u_{p_2}(s^2) = \frac{F(c_2)\cdot m_{p_2}}{m_{p_2}} = 1$.
- $s^3 = \langle c_2, c_2 \rangle$. Payoffs: $u_{p_1}(s^3) = \frac{F(c_2)\cdot m_{p_1}}{m_{p_1}+m_{p_2}} = 2/3$, $u_{p_2}(s^3) = \frac{F(c_2)\cdot m_{p_2}}{m_{p_1}+m_{p_2}} = 1/3$.
- $s^4 = \langle c_2, c_1 \rangle$. Payoffs: $u_{p_1}(s^4) = \frac{F(c_2)\cdot m_{p_1}}{m_{p_1}} = 1$, $u_{p_2}(s^4) = \frac{F(c_1)\cdot m_{p_2}}{m_{p_2}} = 1$.

Note that $H(s_2) - H(s_1) + H(s_3) - H(s_2) + H(s_4) - H(s_3) + H(s_1) - H(s_4) = 0$. However, by definition of exact potential, we get that $(H(s_2) - H(s_1)) + (H(s_3) - H(s_2)) + (H(s_4) - H(s_3)) + (H(s_1) - H(s_4)) = (2/3) + (-1/3) + (2/3) + (-1/3) = 2/3 \neq 0$. A contradiction. □

B. Ordinal potential

To show an ordinal potential, we use the following definitions:

For a configuration $s \in S$ in a game $G_{\Pi,C,F}$, we define $list(s)$ to be the sequence of pairs in $\{\langle RPU_c(s), c\rangle \mid c \in C\}$ ordered lexicographically from smallest to largest. Denote by $v_i(s)$ the coin (second element of the pair) in the i^{th} entry in $list(s)$. Consider the ordered set $\langle L, \prec_L \rangle$, where $L \triangleq \{list(s) \mid s \in S\}$ is the set of all lists in $G_{\Pi,C,F}$, and $\prec_L$ is the lexicographical order. The *rank* of a list $list(s) \in L$, $rank(list(s))$, is the rank of $list(s)$ in $\prec_L$ from smallest to largest. An illustration of a $list(s)$ can be found in Figure 2.

Note that since Π and C are finite, we know that S and L are finite. The following two observations establish a connection between better response steps and the RPUs of the associated coins.

Observation 1. *Consider a game $G_{\Pi,C,F}$, $s \in S$, $v_i(s) \in C$, and $p \in \Pi$ s.t. $s.p = v_i(s)$. Then in every better response step of p that changes $s.p$ to a coin $v_j(s)$, it holds that $j > i$.*

Proof. By the definition of a better response step,

$$\frac{F(v_j(s))}{M_{v_j(s)}(s) + m_p} > \frac{F(v_i(s))}{M_{v_i(s)}(s)},$$

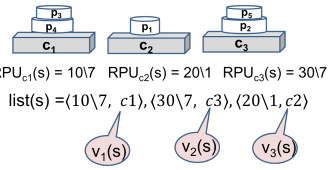


Fig. 2: Illustration of $list(s)$ in a system with three coins and five miners s.t. for $1 \leq i \leq 5$, $m_{p_i} = i$ and for $1 \leq i \leq 3$, the reward function $F(c_i) = 10i$.

and thus

$$RPU_{v_j(s)}(s) = \frac{F(v_j(s))}{M_{v_j(s)}(s)} > \frac{F(v_i(s))}{M_{v_i(s)}(s)} = RPU_{v_i(s)}(s).$$

By definition of $v(s)$, we get that $j > i$. □

Observation 2. *Consider a game $G_{\Pi,C,F}$. If some better response step from configuration s to configuration s' of a miner p changes $s.p = c$ to $s'.p = c'$, then $RPU_c(s) < min(RPU_c(s'), RPU_{c'}(s'))$.*

Proof. By definition of a better response step, $RPU_{c'}(s') > RPU_c(s)$. In addition, since $M_c(s') = M_c(s) - m_p$, we get that

$$RPU_c(s') = \frac{F(c)}{M_c(s')} = \frac{F(c)}{M_c(s) - m_p} > \frac{F(c)}{M_c(s)} = RPU_c(s).$$

□

We are now ready to prove that any game $G_{\Pi,C,F}$ has an ordinal potential function.

Theorem 1. *For any finite sets Π and C of miners and coins and reward function F, $H(s) \triangleq rank(list(s))$ is an ordinal potential in the game $G_{\Pi,C,F}$.*

Proof. Consider two configurations $s, s' \in S$ s.t. some better response step of a miner $p \in \Pi$ leads from configuration s to configuration s', and let $v_i(s) = s.p$ and $v_j(s) = s'.p$. We need to show that $H(s) < H(s')$. Since only the RPUs of $v_i(s)$ and $v_j(s)$ are affected we get that

$$\forall k \neq i, j\ RPU_{v_k(s)}(s) = RPU_{v_k(s)}(s'). \tag{1}$$

By Observation 1, we get that $j > i$, and thus $\forall k,\ 1 \leq k < i$, $RPU_{v_k(s)}(s) = RPU_{v_k(s)}(s')$. By Observation 2, we get that $min(RPU_{v_i(s)}(s'), RPU_{v_j(s)}(s')) > RPU_{v_i(s)}(s)$, and thus, together with the definition of v_i and Equation 1, we get that $\forall k,\ i \leq k \leq |C|$, $RPU_{v_k(s)}(s') \geq RPU_{v_i(s)}(s)$. Therefore, none of these coins "move down" to a position before i in $list(s')$ and so

$$\forall k,\ 1 \leq k < i, \langle RPU_{v_k(s)}(s), v_k(s)\rangle = \langle RPU_{v_k(s')}(s'), v_k(s')\rangle. \tag{2}$$

That is, the first $i-1$ elements of $list(s)$ are equal to the first $i-1$ elements of $list(s')$. It remains to show that the i^{th} element of $list(s')$ is lexicographically larger than the i^{th}

element of $list(s)$. Let $v_l(s) = v_i(s')$. From Equation 2, we know that $l \geq i$, so there are two possible cases:

- First, $l \in \{i, j\}$. The theorem follows from Observation 2.
- Second, $l > i,\ l \neq j$. In this case,

$$\begin{aligned}
&\langle RPU_{v_i(s)}(s), v_i(s)\rangle \\
&< \langle RPU_{v_l(s)}(s), v_l(s)\rangle && \text{(List order)} \\
&= \langle RPU_{v_l(s)}(s'), v_l(s)\rangle && \text{(Equation 1)} \\
&= \langle RPU_{v_i(s')}(s'), v_i(s')\rangle && \text{(Definition of } v_l(s)),
\end{aligned}$$

as needed.

□

Corollary 1. *For any finite sets Π and C of miners and coins and reward function F. For any configuration $s \in \langle \Pi, C\rangle$, any better response dynamic starting form s in the game $G_{\Pi,Q,F}$ converges to a stable configuration.*

IV. There is Often a Better Equilibrium

Before moving to our main result in which we describe an attack through dynamic reward design that transitions the system between equilibria, we establish a motivation for such an attack. In this section, we show that under broad circumstances, in every stable configuration there is at least one miner who has a higher payoff in another stable configuration. This means that such a miner will gain from moving the system there. Specifically, we prove this for games that satisfy the following assumptions (note that we use these assumptions only in this section):

Assumption 1 (Never alone)**.** *For a configuration $s \in S$ in a game $G_{\Pi,C,F}$, if there is a coin $c \in C$ s.t. $|P_c(s)| \leq 1$, then there is a miner $p \in \Pi$ s.t. changing $s.p$ to c is a better response step for p.*

Although this assumption cannot hold when $|\Pi| < 2|C|$, it often holds in practice since the number of miners must be much larger than the number of coins for the cryptocurrency to be secure (truly decentralized).

Assumption 2 (Generic game)**.** *For any two coins $c \neq c' \in C$ and two sets of players $P, P' \subseteq \Pi$ in a game $G_{\Pi,C,F}$,*

$$\frac{F(c)}{\Sigma_{p\in P} m_p} \neq \frac{F(c')}{\Sigma_{p\in P'} m_p}.$$

This assumption is common in game theory [28], and it makes sense in our game since mining power in practice is measured in billions of operations per hour and coin rewards are coupled with coin fiat exchange rates, so exact equality is unlikely.

The following observation follows from Assumption 1 and the fact that coins that are chosen by at least one miner always divide their entire reward. It stipulates that in every stable configuration, the sum of the payoffs the miners get is equal to the sum of the coins' rewards.

Observation 3 (All stable configurations are globally optimal)**.** *For every stable configuration $s \in S$ in a game $G_{\Pi,C,F}$ under Assumption 1, it holds that*

$$\sum\nolimits_{p\in\Pi} u_p(s) = \sum\nolimits_{c\in C} F(c).$$

From Observation 3 and Assumption 2 it is easy to show the following claim:

Claim 4. *Consider a game $G_{\Pi,C,F}$ under Assumptions 1 and 2. If the game has more than one stable configuration, then for every stable configuration s there exist a miner p and a stable configuration s' s.t. $u_p(s') > u_p(s)$.*

Proof. Consider a stable configuration s. By assumption, there exists another stable configuration $s' \neq s$. Therefore, there is a player p and coins $c \neq c'$ s.t. $p \in P_c(s)$ and $p \in P_{c'}(s')$. By Assumption 2, $\frac{F(c)}{M_c(s)} \neq \frac{F(c')}{M_{c'}(s')}$. Thus, $u_p(s') \neq u_p(s)$. If $u_p(s') > u_p(s)$, then we are done. Otherwise, by Observation 3, there is another player $p' \neq p$ s.t. $u_{p'}(s') > u_{p'}(s)$.

□

It remains to show that $G_{\Pi,C,F}$ has more than one stable configuration. We give here a sketch of the proof, which appears in Appendix A. Consider $\Pi = \{p_1, \ldots, p_n\}$ s.t. $m_{p_1} \geq m_{p_2} \geq \ldots \geq m_{p_n}$, and $\forall i, 1 \leq i \leq n$, let $\Pi_i = \{p_1, \ldots, p_i\}$. We first show (in Lemma 3 in Appendix A) that the game $G_{\Pi_2,C,F}$ has two different configurations in which miners p_1, p_2 do not share a coin and at most one of them is unstable. Then, we inductively construct two configurations in $G_{\Pi_i,C,F}$, $\forall i, 3 \leq i \leq n$, based on the two configurations in $G_{\Pi_{i-1},C,F}$, in which all miners in Π_{i-1} keep their locations and all miners except maybe the one that was unstable in $G_{\Pi_1,C,F}$ are stable. The construction step is captured by Claim 6 (Appendix A), where $p_{new} = p_i$ in the i^{th} step.

Finally, we show in Lemma 3 (Appendix A) that the two configurations we construct in $G_{\Pi,C,F}$ are stable: Let p_{ns} be the (possibly) unstable miner. By Assumption 1 (note that the assumption refers only to game $G_{\Pi,C,F}$), p_{ns} cannot be alone in a coin (otherwise there must be another unstable miner), and thus it shares the coin with a smaller (with less mining power) stable miner, which we show implies that p_{ns} is stable.

Our result is then captured by the following proposition, which follows from Claim 4 and Lemma 3 that are proven in Appendix A.

Proposition 2. *Consider a game $G_{\Pi,C,F}$ under Assumptions 1 and 2. Then for every stable configuration s in $G_{\Pi,C,F}$ there exist a miner p and a stable configuration $s' \neq s$ in which $u_p(s') > u_p(s)$.*

V. Reward Design attack: Moving Between Equilibria

Having established motivation for strategic attacks, we proceed to present our main result. We consider a system $Q = \langle \Pi, C\rangle$, where $\Pi = \{p_1, \ldots p_n\}$ s.t. $m_{p_1} > m_{p_2} > \ldots > m_{p_n}$, and a function F. For every two stable configurations $s_0, s_f \in S_Q$ in the game $G_{\Pi,C,F}$ we provide a mechanism to move the system from s_0 to the desired configuration s_f by temporarily increasing coin rewards. Note that once we lead the system to s_f, we can return to the original rewards (i.e., stop manipulating coin weights) because s_f is stable in $G_{\Pi,C,F}$.

a) Dynamic reward design attack.: Consider a system $\langle \Pi, C \rangle$ and a reward function F. A *dynamic-reward design attack* for game $G_{\Pi,C,F}$ is an algorithm that for any two stable configurations s_0, s_f in $G_{\Pi,C,F}$ moves the system from s_0 to s_f by following the protocol schema in Algorithm 1.

Algorithm 1 Protocol schema to move a system $\langle \Pi, C \rangle$ with reward function F from s_0 to s_f.

1: $s \leftarrow s_0$
2: **while** $s \neq s_f$ **do**
3: choose a reward function H s.t. $\forall c \in C,\ H(c) \geq F(c)$
4: allow better-response dynamics in $G_{\Pi,C,H}$, starting from s, to converge to some stable configuration s'
▷ convergence is guaranteed due to Corollary 1
5: $s \leftarrow s'$

The rest of the section is organized as follows: In Section V-A we describe the algorithm of our dynamic reward design attack and in Section V-B we present the algorithm's proof outline (some technical parts are defer to the Appendix).

A. Reward design attack

To describe the algorithm of a dynamic reward design attack we need to specify the reward function for every loop iteration in Algorithm 1. To this end, we define the notion of a *reward design function*, which maps system configurations to reward functions:

Definition 1 (reward design function)**.** *Consider a system Q. A* reward design function *F for system Q is a function mapping every configuration $s \in S_Q$ to a reward function, i.e., $F(s) : C \rightarrow \mathbb{R}_+$.*

As we shortly explain, we divide the algorithm into n stages, each of which may consists of many loop iterations, and provide one reward design function for each stage to define a specific reward function for each iteration.

Intuitively, we observe that miners with less mining power are easily moved between coins, meaning that we can increase a coin reward so that a *small miner* with little mining power will benefit from moving there, but *bigger miners* with more mining power prefer to stay in their current locations. Therefore, the idea is to evolve the current configuration to $s_f \in S$ in $n = |\Pi|$ stages, where in stage i, we move the $n-i+1$ miners with the smallest mining powers to the location (coin) of miner p_i in the final configuration s_f (i.e., $s_f.p_i$) while keeping the remaining miners in their (final) places. To this end, we define n intermediate configurations. For i, $1 \leq i \leq n$, we define s_i as:

$$s_i.p_k = \begin{cases} s_f.p_k & \forall 1 \leq k < i \\ s_f.p_i & \forall i \leq k \leq n \end{cases} \tag{3}$$

That is, in s_i, miners $p_1, \ldots, p_i$ are in their final locations and miners $p_i, \ldots, p_n$ are in the final location of miner p_i. Note that $s_n = s_f$. Figure 3a illustrates the stage transitions in the algorithm.

Notice that since we allow arbitrary better response dynamics (in every iteration), choosing a reward design function is a subtle task; miners can move according to any better response step, and we cannot control the order in which they move. One may attempt to design a reward function so that in the resulting game there is exactly one unstable miner with exactly one better response step in the current configuration. However, even given such a function, after that miner takes its step, other miners might become unstable, which can in turn lead to a better response dynamics process that depends on the order in which miners move and on the choices they make (in case they have more than one better response step). Hence, the main challenge is to be able to restrict the set of possible stable configurations reached by better response dynamics in each iteration.

In every loop iteration of stage $i > 1$ we pick a miner p_k that we want to move from $s_f.p_{i-1}$ to $s_f.p_i$ (as explained shortly) and choose the reward function carefully so that (1) p_k's only better response step is $s_f.p_i$, (2) all other miners are stable, and (3) in every stable configuration reached by better response dynamics after p_k's step, p_k is in $s_f.p_i$, all miners $p_{k+1}, \ldots, p_n$ are in either $s_f.p_{i-1}$ or $s_f.p_i$, and all the other miners (who have more mining power) remain in their (final) locations.

Moreover, we prove by induction below that our reward design function of stage i (defined below) guarantees that the set of possible configurations reached by better response dynamics in a stage $i > 1$ is

$$T_i \triangleq \left\{ s \in S \;\middle|\; \begin{array}{l} \forall k,\ 1 \leq k < i :\ s.p_k = s_f.p_k \\ \forall k, i \leq k \leq n :\ s.p_k \in \{s_f.pi, s_f.p_{i-1}\} \end{array} \right\}$$

Notice that the stage starts at s_{i-1}, which is in T_i. We now explain how we choose the reward design function for stage i. First, for every configuration $s \in T_i \setminus \{s_i\}$, we choose the *mover* – the miner to move from $s_f.p_{i-1}$ to $s_f.p_i$ – to be the miner with the following index:

$$\mu_i(s) \triangleq min\{j | \forall l,\ j < l \leq n : s.p_l = s_f.p_i\}.$$

Let $\alpha_i(s) = \mu_i(s) - 1$. Intuitively, we use $p_{\alpha_i(s)}$ as an *anchor* in configuration s; we choose a reward function that increases the reward of coin $s_f.p_i$ as much as possible without making the anchor unstable. As a result, all the miners in $P_{s_f.p_{i-1}}(s)$ (who are bigger than or equal to the anchor) remain stable, whereas miner $p_{\mu_i(s)}$ has a unique better response step to move to $s_f.p_i$. Note that, by definition, $s.p_{\mu_i(s)} \neq s_f.p_i$. Figure 3b illustrates $\mu_i(s)$ and $\alpha_i(s)$ for some configuration $s \in T_i$.

In order to make sure that miners not in $P_{s_f.p_{i-1}}(s) \cup P_{s_f.p_i}(s)$ also remain stable, and in order to guarantee that that every better response dynamics after $p_{\mu_i(s)}$'s step converges to a configuration in T_i, we choose a reward function that evens out the RPUs of all coins other than $s_f.p_i$. For $s \in S$, let

(a) Configuration s_i.

(b) Iteration starting from $s \in T_i$ and moving $p_{\mu_i(s)}$ to $s_f.p_i$. $p_{\alpha_i(s)}$ is the anchor.

Fig. 3: Reward design algorithm: (a) stages; and (b) iteration within stage i. Boxes represent coins, discs represent miners. The unlabeled bottom discs represent possible bigger miners who are in their final locations.

$R(s) \triangleq max\{RPU_c(s) \mid c \in C\}$. The reward design function H_i for stage $i > 1$ is:

$$\text{for } n \geq i > 1,\ s \in T_i,\ c \in C,$$
$$H_i(s)(c) = \begin{cases} R(s) \cdot (M_c(s) + m_{p_{\alpha_i(s)}}) & \text{for } c = s_f.p_i \\ R(s) \cdot M_c(s) & \text{otherwise} \end{cases} \quad (4)$$

That is, we add the weight of the anchor $\alpha_i(s)$ to the reward of the coin we wish to have the mover $\mu_i(s)$ move to. Note that the RPUs of all coins except $s_f.p_i$ in the game $G_{\Pi,C,H_i(s)}$ are equal to $R(s)$. In addition, note that if a miner bigger than or equal to $p_{\alpha_i(s)}$ moves to $s_f.p_i$, then $s_f.p_i$'s RPU becomes no bigger than $R(s)$. However, since $m_{p_{\mu_i(s)}} < m_{p_{\alpha_i(s)}}$, $p_{\mu_i(s)}$ has a unique better response step to move to $s_f.p_i$. Therefore, we get that our reward design function allows us to control the first step of the better response dynamics process. In the next section we give more intuition on how it also restricts the stable configuration at the end of any better response dynamics process at stage i to the set T_i.

As for the first stage, note that we need to move all miners to coin $s_f.p_1$, so intuitively we only need to increase its reward high enough. We therefore choose:

$$\text{for } s \in S,\ c \in C,$$
$$H_1(s)(c) = \begin{cases} max\{F(c') \mid c' \in C\} \cdot \Sigma_{p \in \Pi} m_p & \text{for } c = s_f.p_1 \\ F(c) & \text{otherwise} \end{cases} \quad (5)$$

In Algorithm 2 we present the algorithm of our reward design attack, and in the next sections we prove that every stage in the algorithm eventually completes.

Algorithm 2 The algorithm of the reward design attack.

```
s ← s_0
for i=1 ...n do
    while s ≠ s_i do          ▷ s_i is defined in Equation 3
        define H_i as in Equations 4 and 5
        allow better-response dynamics in G_{Π,C,H_i(s)}, starting
        from s, to converge to some stable configuration s'
        s ← s'
```

B. Proof outline

The proof for stage 1 termination is straightforward so we skip it. To prove that every stage $i > 1$ eventually completes we use the following key lemma on stable configurations in the stage:

Lemma 2. *Consider a configuration $s \in T_i \setminus \{s_i\}$. Then every better response dynamics in the game $G_{\Pi,C,H_i(s)}$ that starts at s converges to a configuration $s' \in T_i$ such that:*

1) $\forall k,\ 1 \leq k < \mu_i(s),\ s'.p_k = s.p_k$.
2) $s'.p_{\mu_i(s)} = s_f.p_i$.

The formal proof of the lemma is technical and for better readability is given in Appendix B. Here we give an intuitive justification for it. As part of the proof, we show that within stage i, all the reached configurations (both stable and unstable) are in T_i. Let $c = s_f.p_{i-1}$ and $c' = s_f.p_i$. After $p_{\mu_i(s)}$ moves to c' according to his only better response step, in the resulting configuration s', the RPUs of all coins not in $\{c, c'\}$ remain $R(s)$. Moreover,

$$RPU_c(s') = \frac{H_i(s)(c)}{M_c(s')} = \frac{R(s) \cdot M_c(s)}{M_c(s) - m_{p_{\mu_i(s)}}},$$

and

$$RPU_{c'}(s') = \frac{H_i(s)(c')}{M_{c'}(s')} = \frac{R(s) \cdot (M_c(s) + m_{p_{\alpha_i(s)}})}{M_c(s) + m_{p_{\mu_i(s)}}}.$$

Note that, although $RPU_c(s') > RPU_c(s)$, it is still not high enough to drive miners not in $P_{c'}(s')$, which by definition have more mining power than $p_{\mu_i(s)}$, to move to c. In addition, since $RPU_{c'}(s') < RPU_{c'}(s)$, no miner want to move to c', and thus the only miners that possibly have better response steps at s' are miners in $P_{c'}(s')$ who wish to move to c. Moreover, the total mining power of the miners who actually move to c is smaller than $p_{\mu_i(s)}$, otherwise, c's RPU would go below $R(s)$. In the proof we use the above intuition to formulate an invariant who captures the lemma statement and prove it by induction on better response steps. The lemma then follows from Theorem 1 (every better response dynamics converges to a stable configuration).

We next use Lemma 2 to prove that every stage $i > 1$ completes in a finite number of loop iterations. To this end, we associate with every configuration $s \in T_i$ a binary vector $vec_i(s)$ indicating, for each $j \geq i$, whether p_j is in $P_{s_f.p_i}(s)$,

where it needs to be at the end of the stage. Formally, consider the ordered set $\langle V, \prec_v \rangle$, where $V \triangleq \{0,1\}^n$ is the set of all binary vectors of length n, and $\prec_v$ is the lexicographical order. For a configuration $s \in T_i$, we define $vec_i(s)$ to be a vector in V such that:

$$\forall j, 1 \leq j < i,\ vec_i(s)[j] = 0$$
$$\forall j, i \leq j \leq n,\ vec_i(s)[j] = \begin{cases} 1 & \text{if } p_j \in P_{s_f.p_i}(s) \\ 0 & \text{otherwise} \end{cases}$$

and the function $\Phi_i : T_i \to \{1, .., |V|\}$ to be the rank of $vec(s)$ in V. For example, by the definition, the vector $vec_i(s_i)$ contains zeros in the first $i-1$ entries and ones in the rest.

Theorem 2. *Every stage $i > 1$ of Algorithm 2 completes in a finite number of loop iterations.*

Proof. By definitions of stage i and set T_i, the first configuration of stage i is $s_{i-1} \in T_i$. By definition of $\mu_i(s)$, for all $s \in T_i \setminus \{s_i\}$, $\mu_i(s) \notin P_{s_f.p_i}(s)$. By inductively applying Lemma 2, we get that every loop iteration in stage i ends in a configuration in T_i. Therefore, consider a loop iteration of stage i that starts in configuration $s \neq s_i$ and ends in configuration s', we get by Lemma 2 that $p_{\mu_i(s)} \in P_{s_f.p_i}(s')$ and $\forall k,\ i \leq k < \mu_i(s)$, $s.p_k = s'.p_k$. In addition, by definition of $\mu_i(s)$, $p_{\mu_i(s)} \neq P_{s_f.p_i}(s)$. Meaning that vectors $vec_i(s)$ and $vec_i(s')$ are equals up to the index $\mu_i(s) - 1$, and in $vec_i(s)[\mu_i(s)] = 0$ whereas $vec_i(s')[\mu_i(s)] = 1$. Therefore $\Phi(s') > \Phi(s)$. Now since the set T_i is finite, we get that after a finite number of iterations we reach configuration s_i.

□

VI. Discussion

Our work studies and challenges the cryptocurrency market from a novel angle – the strategic miners' behavior in a system with multiple coins. Although our model is theoretical, it does capture the nature of the real-life cryptocurrency market and our results emphase that cryptocurrency protocol designers must take into account the whole market environment when reasoning about their coin's security.

There are several central followups one may consider. First, our reward design attack is effective for arbitrary better-response dynamics, but one may wonder about its speed of convergence under specific markets. Second, we consider convergence to an equilibrium, and one may consider also convergence to a bad (possibly unstable) configuration in which, for example, a particular miner will have a dominant position in a coin, killing (at least for a while) the basic guarantee of security for that coin and allowing him to get a bigger portion of the reward. In addition, one may wonder about the asymmetric case, where some coins can be mined only by a subset of the miners. Finally, we believe that the reward design attack is an interesting technique on it own and might be applicable to other domains like load balancing and selfish routing.

References

[1] Bitcoin cash. https://www.bitcoincash.org/, Accessed: 2019-07-01.

[2] Bitcoin price manipulation: Economists warn just one person may have caused value surge. https://www.express.co.uk/finance/city/905222/bitcoin-price-news-latest-manipulation-cryptocurrency-surge-plummet-stock-exchange, Accessed: 2019-07-01.

[3] Cardano. https://www.cardano.org/en/home/, Accessed: 2019-07-01.

[4] Crypto whales and how they manipulate the price. https://steemit.com/cryptocurrency/@endpoint/crypto-whales-and-how-they-manipulate-the-price, Accessed: 2019-07-01.

[5] Cryptocurrency info charts. https://bitinfocharts.com/comparison/hashrate-btc-bch.html, Accessed: 2019-07-01.

[6] Cryptocurrency market state visualization. https://coin360.io/, Accessed: 2019-07-01.

[7] Cryptocurrency price manipulation is unavoidable. https://www.cnbc.com/2018/02/13/cryptocurrency-price-manipulation-is-unavoidable-nem-president-says.html, Accessed: 2019-07-01.

[8] How the winner got fomo3d prize a detailed explanation. https://medium.com/coinmonks/how-the-winner-got-fomo3d-prize-a-detailed-explanation-b30a69b7813f, Accessed: 2019-07-01.

[9] Litecoin. https://litecoin.com/, Accessed: 2019-07-01.

[10] Neo. https://neo.org/, Accessed: 2019-07-01.

[11] What to mine. https://whattomine.com/, Accessed: 2019-07-01.

[12] Eitan Altman, Alexandre Reiffers, Daniel S Menasche, Mandar Datar, Swapnil Dhamal, and Corinne Touati. Mining competition in a multi-cryptocurrency ecosystem at the network edge: a congestion game approach. *ACM SIGMETRICS Performance Evaluation Review*, 46(3):114–117, 2019.

[13] E. Anshelevich, A. Dasgupta, J. Kleinberg, É. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.

[14] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform, 2014.

[15] Miles Carlsten, Harry Kalodner, S Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 154–167. ACM, 2016.

[16] Eyal Even-Dar, Alex Kesselman, and Yishay Mansour. Convergence time to nash equilibria. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming*, pages 502–513, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[17] Ittay Eyal. The miner's dilemma. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 89–103. IEEE, 2015.

[18] Ittay Eyal and Emin Gun Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*, pages 436–454. Springer, 2014.

[19] Dimitris Fotakis, Spyros Kontogiannis, Elias Koutsoupias, Marios Mavronicolas, and Paul Spirakis. The structure and complexity of nash equilibria for a selfish routing game. In Peter Widmayer, Stephan Eidenbenz, Francisco Triguero, Rafael Morales, Ricardo Conejo, and Matthew Hennessy, editors, *Automata, Languages and Programming*, pages 123–134, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[20] Dimitris Fotakis, Spyros Kontogiannis, and Paul Spirakis. Selfish unsplittable flows. In Josep Díaz, Juhani Karhumäki, Arto Lepistö, and Donald Sannella, editors, *Automata, Languages and Programming*, pages 593–605, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[21] Neil Gandal, JT Hamrick, Tyler Moore, and Tali Oberman. Price manipulation in the bitcoin ecosystem. *Journal of Monetary Economics*, 2018.

[22] M. Goemans, Vahab Mirrokni, and A. Vetta. Sink equilibria and convergence. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 142–151, Oct 2005.

[23] Guy Goren and Alexander Spiegelman. Mind the mining. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 475–487, 2019.

[24] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett,

editors, *Advances in Neural Information Processing Systems 30*, pages 6765–6774. Curran Associates, Inc., 2017.
[25] Tobias Harks. *Theoretical and Computational Aspects of Resource Allocation Games*. PhD thesis, Technical University of Berlin, 2012.
[26] Tobias Harks and Max Klimm. On the existence of pure nash equilibria in weighted congestion games. *Mathematics of Operations Research*, 37(3):419–436, 2012.
[27] Tobias Harks, Max Klimm, and Rolf H. Mohring. Characterizing the existence of potential functions in weighted congestion games. *Theory of Computing Systems*, 49, 2011.
[28] Ron Holzman and Nissan Law-yone. Network structure and strong equilibrium in route selection games. *Mathematical Social Sciences*, 46(2):193–205, 2003.
[29] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. Game-theoretic analysis of ddos attacks against bitcoin mining pools. In *International Conference on Financial Cryptography and Data Security*, pages 72–86. Springer, 2014.
[30] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 365–382. ACM, 2016.
[31] Yujin Kwon, Hyoungshick Kim, Jinwoo Shin, and Yongdae Kim. Bitcoin vs. bitcoin cash: Coexistence or downfall of bitcoin cash? In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 935–951. IEEE, 2019.
[32] Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In *International Conference on Financial Cryptography and Data Security*, pages 264–279. Springer, 2017.
[33] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. A survey on blockchain: a game theoretical perspective. *IEEE Access*, 7:47615–47643, 2019.
[34] Michael Mirkin, Yan Ji, Jonathan Pang, Ariah Klages-Mundt, Ittay Eyal, and Ari Jules. Bdos: Blockchain denial of service. *arXiv preprint arXiv:1912.07497*, 2019.
[35] D. Monderer and L.S. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
[36] Dov Monderer and Moshe Tennenholtz. k-implementation. *Journal of Artificial Intelligence Research*, 21:37–62, 2004.
[37] Dov Monderer and Moshe Tennenholtz. Strong mediated equilibrium. *Artif. Intell.*, 173(1):180–195, 2009.
[38] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.
[39] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 305–320. IEEE, 2016.
[40] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007.
[41] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 515–532. Springer, 2016.
[42] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security*, pages 477–498. Springer, 2016.
[43] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial intelligence*, 73(1-2):231–252, 1995.
[44] Jonathan Sorg, Richard L Lewis, and Satinder P. Singh. Reward design via online gradient ascent. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2190–2198. Curran Associates, Inc., 2010.
[45] Jonathan Sorg, Satinder P. Singh, and Richard L. Lewis. Internal rewards mitigate agent boundedness. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1007–1014, 2010.
[46] Alexander Spiegelman, Idit Keidar, and Moshe Tennenholtz. Game of coins. *arXiv preprint arXiv:1805.08979*, 2018.
[47] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998.
[48] Itay Tsabary and Ittay Eyal. The gap game. In *Proceedings of the 2018 ACM SIGSAC conference on Computer and Communications Security*, pages 713–728, 2018.

APPENDIX

A. Formal Proofs From Section IV

Claim 5. *Consider a game $G_{\Pi,C,F}$, a configuration $s \in S$, a coin $c \in C$, and two miners $p, p' \in P_c(s)$ s.t. $m_p \leq m_{p'}$. If p is stable in s, then p' is stable in s as well.*

Proof. Since p is stable in s, we get that

$$F(c)\frac{m_p}{M_c(s)} \geq F(c')\frac{m_p}{M_{c'}(s)+m_p} \text{ for every } c' \in C$$

Thus,

$$F(c)\frac{1}{M_c(s)} \geq F(c')\frac{1}{M_{c'}(s)+m_p} \text{ for every } c' \in C.$$

Since, $m_p \leq m_{p'}$, it follows that

$$F(c)\frac{1}{M_c(s)} \geq F(c')\frac{1}{M_{c'}(s)+m_{p'}} \text{ for every } c' \in C.$$

Thus,

$$F(c)\frac{m_{p'}}{M_c(s)} \geq F(c')\frac{m_{p'}}{M_{c'}(s)+m_{p'}} \text{ for every } c' \in C,$$

and p' is stable in s.

□

Claim 6. *Let F be a reward function. Consider a system $Q = \langle \Pi, C \rangle$, and a configuration $s \in S_Q$. Now consider another system $Q' = \langle \Pi', C \rangle$ s.t. $\Pi' = \Pi \cup \{p_{new}\}$, $p_{new} \notin \Pi$, and $m_{p_{new}} \leq min\{m_p | p \in \Pi\}$. Let*

$$c = \underset{c' \in C}{\text{argmax}}\ F(c')\frac{m_{p_{new}}}{M_{c'}(s)+m_{p_{new}}}$$

and consider a configuration $s' \in S_{Q'}$ s.t. for all $p \in \Pi$ $s'.p = s.p$ and $s'.p_{new} = c$. Then p_{new} is stable in s' in game $G_{\Pi',C,F}$, and every player $p \in \Pi$ that is stable in s in $G_{\Pi,C,F}$ is also stable in s' in $G_{\Pi',C,F}$.

Proof. By construction of configuration s', $M_c(s') = M_c(s) + m_{p_{new}}$ and $\forall c' \neq c\ M_{c'}(s') = M_{c'}(s)$. Therefore, by the way we pick c, we get that

$$F(c)\frac{m_{p_{new}}}{M_c(s')} \geq F(c')\frac{m_{p_{new}}}{M_{c'}(s')} \text{ for all } c' \in C,$$

and thus p_{new} is stable in s'. Now consider a player $p \in \Pi$ that is stable in s, we show that p is stable also in s'. Consider two cases:

- First, $p \in P_{c'}(s')$ s.t. $c \neq c'$. By construction, $p \in P_{c'}(s)$, and since p is stable in s we know that

  $$F(c')\frac{m_p}{M_{c'}(s)} \geq F(c'')\frac{m_p}{M_{c''}(s)+m_p}$$

  for every $c'' \in C$. Now since $M_c(s') > M_c(s)$ and for all $c'' \neq c$, $M_{c''}(s') = M_{c''}(s)$, we get

  $$F(c')\frac{m_p}{M_{c'}(s')} \geq F(c'')\frac{m_p}{M_{c''}(s')+m_p}$$

  for every $c'' \in C$. Meaning that p is stable in s'.
- Second, $p \in P_c(s')$. Since p_{new} is stable in s' and $m_p \geq m_{p_{new}}$, we get by Claim 5 that p is stable in s'.

□

Lemma 3. *Any game $G_{\Pi,C,F}$ under Assumptions 1 and 2 has at least two different stable configurations.*

Proof. Let $p_1, \ldots, p_n$ be the miners in Π in decreasing mining power, i.e., $m_{p_1} \geq m_{p_2} \geq \ldots \geq m_{p_n}$ and let $c_1, \ldots, c_l$ be the coins in C sorted by decreasing coin rewards, i.e., $F(c_1) \geq F(c_2) \geq \ldots \geq F(c_l)$. Note that through the proof we construct several games, but we assume Assumptions 1 and 2 only in the original game $G_{\Pi,C,F}$. Let $\Pi_1, \ldots, \Pi_n$ be the following sets of miners s.t. $\Pi_k \triangleq \{p_1, \ldots, p_k\}$, $1 \leq k \leq n$. Next consider two configurations s_1^2, s_2^2 in game $G_{\Pi_2,C,F}$: $s_1^2 = \langle c_1, c_2 \rangle$ (i.e., $s_1^2.p_1 = c_1$ and $s_1^2.p_2 = c_2$) and $s_2^2 = \langle c_2, c_1 \rangle$. Note that $s_1^2 \neq s_2^2$, and since $F(c_1) \geq F(c_2)$, p_1 is stable in s_1^2 and p_2 is stable in s_2^2.

We now use s_1^2, s_2^2 to inductively construct a sequence of pairs of configurations $\langle s_1^2, s_2^2 \rangle, \langle s_1^3, s_2^3 \rangle, \ldots, \langle s_1^n, s_2^n \rangle$, where $\forall\ 2 \leq k \leq n$, s_1^k, s_2^k are two configurations in game $G_{\Pi_k,C,F}$. For $3 \leq k \leq n$, let

$$x_1 = \underset{c \in C}{\operatorname{argmax}}\ F(c) \frac{m_{p_k}}{M_c(s_1^{k-1}) + m_{p_k}}$$

and

$$x_2 = \underset{c \in C}{\operatorname{argmax}}\ F(c) \frac{m_{p_k}}{M_c(s_2^{k-1}) + m_{p_k}},$$

we construct $s_1^k = s_1^{k-1} \times x_1$ (i.e., $\forall 1 \leq i < k$, $s_1^k.p_i = s_1^{k-1}.p_i$ and $s_1^k.p_k = x_1$) and $s_2^k = s_2^{k-1} \times x_2$.

Note that since $s_1^2 \neq s_2^2$, we get by construction that $s_1^n \neq s_2^n$. It remains to show that configurations s_1^n and s_2^n are stable. Assume by way of contradiction that it is not the case, and assume w.l.o.g. that s_1^n is not stable. By inductively applying Claim 6, and since p_1 is stable in s_1^2 we get that players p_1 and $p_3, \ldots, p_n$ are stable in s_1^n. Thus p_2 is not stable. Recall that $s_1^2.p_2 = c_2$, and thus by construction $s_1^n.p_2 = c_2$. Recall also that we assume Assumptions 1 and 2, and consider two cases:

- First, $|P_{c_2}(s_1^n)| = 1$. By Assumption 1, there is a miner $p \in \Pi$ s.t. changing $s_1^n.p$ to c_2 is a better response step for p. Thus, p is not stable in s_1^n. In addition, by definition of better response step, we know that $p_i \neq p_2$. – A contradiction to p_2 being the only unstable miner in s_1^n.
- Second, $|P_{c_2}(s_1^n)| > 1$. Since $p_1 \in P_{c_1}(s_1^2)$, we get that there is a stable miner $p \in P_{c_2}(s_1^2)$ s.t. $p \in \{p_3, \ldots, p_n\}$, and thus, $m_2 \geq m_p$. Therefore, by Claim 5, we get that p_2 is stable in s_1^n. A contradiction.

□

B. Proof of The Key Lemma 2 From Section V

Lemma 2 (restated). *Consider a configuration $s \in T_i \setminus \{s_i\}$. Then every better response dynamics in the game $G_{\Pi,C,H_i(s)}$ that starts at s converges to a configuration $s' \in T_i$ such that:*

1) $\forall k,\ 1 \leq k < \mu_i(s)$, $s'.p_k = s.p_k$.
2) $s'.p_{\mu_i(s)} = s_f.p_i$.

Proof. Let $c = s_f.p_{i-1}$ and $c' = s_f.p_i$. By definition of $\mu_i(s)$, and since $s \neq s_i$ we know that $p_{\mu_i(s)} \in P_c(s)$. We first show that the only better response step in configuration s in game $G_{\Pi,C,H_i(s)}$ is that $p_{\mu_i(s)}$ moves to c'. This follows from the following three observations.

1) Since $\forall c'' \neq c'$, $RPU_{c''}(s) = R(s)$ and $RPU_{c'}(s) > R(s)$, we get that no miner has a better response step to move to any coin $c'' \neq c'$.
2) Since $m_{p_{\mu_i(s)}} < m_{p_{\alpha_i(s)}}$, we get that

$$\begin{aligned} RPU_{c'}((s_{-p_{\mu_i(s)}}, c')) &= \frac{R(s) \cdot (M_{c'}(s) + m_{p_{\alpha_i(s)}})}{M_{c'}(s) + m_{p_{\mu_i(s)}}} \\ &> R(s) = RPU_c(s), \end{aligned}$$

   and thus moving to c' is a better response step for $p_{\mu_i(s)}$ in configuration s.
3) Now consider a miner $p_k \notin P_{c'(s)}$ s.t. $k \neq \mu_i(s)$. By definition of $\mu_i(s)$, $k < \mu_i(s)$, thus $m_{p_k} > m_{p_{\mu_i(s)}}$, and thus,

$$\frac{R(s) \cdot (M_{c'}(s) + m_{p_{\alpha_i(s)}})}{M_{c'}(s) + m_{p_k}} \leq R(s).$$

All in all, we get that the only better response step in configuration s is that $p_{\mu_i(s)}$ moves to c'.

Let $s^0 = (s_{-p_{\mu_i(s)}}, c')$ be the configuration reached after $p_{\mu_i(s)}$ takes its step. We next prove by induction that every configuration s' that is reached by better response steps starting from s^0 satisfies the following properties:

Ψ_1. $\forall k,\ 1 \leq k < \mu_i(s)$: $s'.p_k = s.p_k$. The heavy miners up to the anchor remain stable.
Ψ_2. $s'.p_{\mu_i(s)} = c'$. The mover remains stable.
Ψ_3. $\forall k,\ \mu_i(s) < k \leq n$: $s'.p_k \in \{c, c'\}$. The light miners have two options.
Ψ_4. $M_c(s^0) \leq M_c(s') \leq M_c(s)$. The power of miners in coin c does not fluctuate.
Ψ_5. $M_{c'}(s) \leq M_{c'}(s') \leq M_{c'}(s^0)$. The power of miners in coin c' does not fluctuate.

Note that since $s \in T_i$, $\Psi_1 - \Psi_3$ imply that $s' \in T_i$.

Base. We show that the properties $\Psi_1 - \Psi_5$ are satisfied for configuration s^0.

Ψ_1 – Ψ_2. Follow by definitions of $\mu_i(s)$ and T_i, and since $s^0 = (s_{-p_{\mu_i(s)}}, c')$.
Ψ_3. Trivially follows.
Ψ_4 – Ψ_5.

Induction step. Consider two configurations $s^1, s^2 \in S$ s.t. a better response step of some miner leads from s^1 to s^2, and s^1 satisfies properties $\Psi_1 - \Psi_5$. We show that s_2 satisfies properties $\Psi_1 - \Psi_5$ as well.

a) Useful equations for configuration s^1.: We start by proving two useful equations on the RPUs of coins c and c' in configuration s^1 using Ψ_4 and Ψ_5:

By Ψ_4, $M_c(s^1) \leq M_c(s)$. In addition, since $H_i(s)(c) = R(s) \cdot M_c(s)$, we get that

$$RPU_c(s^1) = \frac{H_i(s)(c)}{M_c(s^1)} = \frac{R(s) \cdot M_c(s)}{M_c(s^1)} \geq \frac{R(s) \cdot M_c(s)}{M_c(s)} = R(s). \tag{6}$$

By Ψ_5, $M_{c'}(s^1) \leq M_{c'}(s^0)$. In addition, since (1) $m_{p_{\mu_i(s)}} < m_{p_{\alpha_i(s)}}$, (2) $M_{c'}(s) = M_{c'}(s^0) - m_{p_{\mu_i(s)}}$, and (3) $H_i(s)(c') = R(s) \cdot (M_{c'}(s) + m_{p_{\alpha_i(s)}})$, we get that

$$\begin{aligned} RPU_{c'}(s^1) &= \frac{H_i(s)(c')}{M_{c'}(s^1)} = \frac{R(s) \cdot (M_{c'}(s) + m_{p_{\alpha_i(s)}})}{M_{c'}(s^1)} \\ &= \frac{R(s) \cdot (M_{c'}(s^0) - m_{p_{\mu_i(s)}} + m_{p_{\alpha_i(s)}})}{M_{c'}(s^1)} \\ &\geq \frac{R(s) \cdot M_{c'}(s^0)}{M_{c'}(s^0)} = R(s). \end{aligned} \tag{7}$$

b) Useful equations for configurations s^1 and s^2.: We now prove two useful equations on the RPUs of coins not in $\{c, c'\}$ in configurations that satisfy $\Psi_1 - \Psi_3$.

By definition of H_i, $\forall c'' \notin \{c, c'\}$, $H_i(s)(c'') = R(s) \cdot M_{c''}(s)$. Since $s \in T_i$, $\forall k, i \leq k \leq n : \ s.p_k \in \{c, c'\}$, and by definition of $\mu_i(s)$, we know that $\mu_i(s) \geq i$. Therefore, by $\Psi_1 - \Psi_3$, we get that

$$\forall c'' \notin \{c, c'\}, \ M_{c''}(s^1) = M_{c''}(s), \tag{8}$$

and thus

$$\forall c'' \notin \{c, c'\}, \ RPU_{c''}(s^1) = \frac{H_i(s)(c'')}{M_{c''}(s^1)} = \frac{R(s) \cdot M_{c''}(s)}{M_{c''}(s)} = R(s) \tag{9}$$

c) Induction step proof.: We are now ready to prove that s_2 satisfies properties $\Psi_1 - \Psi_5$.

$\Psi_1 - \Psi_2$. Since Ψ_1 and Ψ_2 hold in s^1, we only need to show that $p_1, \ldots, p_{\mu_i(s)}$ are stable in s^1. Let $k \in \{1, \ldots, \mu_i(s)\}$ and note that $m_{p_k} \geq m_{p_{\mu_i(s)}}$. By Equation 9, for all $c'' \notin \{c, c'\}$, $RPU_{c''}(s^1) = R(s)$, and by Equations 6 and 7, we know that $RPU_c(s^1) \geq R(s)$ and $RPU_{c'}(s^1) \geq R(s)$, respectively. Therefore, it remains to show that p_k does not have a better response step to c or c':

- By Ψ_4, $M_c(s^1) \geq M_c(s^0)$, and thus we get that

$$\begin{aligned} \frac{H_i(c)}{M_c(s^1) + m_{p_k}} &\leq \frac{H_i(c)}{M_c(s^0) + m_{p_{\mu_i(s)}}} \\ &= \frac{H_i(c)}{M_c(s)} = \frac{M_c(s) \cdot R(s)}{M_c(s)} = R(s). \end{aligned}$$

  Therefore, p_k does not have a better response step to c.

- By Ψ_2, $p_{\mu_i(s)} \in P_{c'}(s^1)$. Therefore, if $p_k = p_{\mu_i(s)}$, then we are done. Otherwise, $k < \mu_i(s) = \alpha_i(s) + 1$, so $m_{p_k} \geq m_{p_{\alpha_i(s)}}$. By Ψ_5, $M_{c'}(s^1) \geq M_{c'}(s)$, and thus

$$\begin{aligned} \frac{H_i(c')}{M_{c'}(s^1) + m_{p_k}} &\leq \frac{H_i(c')}{M_{c'}(s) + m_{p_{\alpha_i(s)}}} \\ &= \frac{R(s) \cdot (M_{c'}(s) + m_{p_{\alpha_i(s)}})}{M_{c'}(s) + m_{p_{\alpha_i(s)}}} = R(s). \end{aligned}$$

  Therefore, p_k does not have a better response step to c'.

Ψ_3. By Equations 6 and 7, $RPU_c(s^1) \geq R(s)$ and $RPU_{c'}(s^1) \geq R(s)$. By Equation 9, for all $c'' \notin \{c, c'\}$, $RPU_{c''}(s^1) = R(s)$. Therefore, since by the inductive assumption (Ψ_3), miners $p_{\mu_i(s)+1}, \ldots, p_n$ are in $P_c(s^1) \cup P_{c'}(s^1)$, we get that none of them has a better response step to move to a coin $c'' \notin \{c, c'\}$, and thus Ψ_3 holds in s^2 as well.

Ψ_4. By definitions of s^0, $\mu_i(s)$, and T_i, we know that $P_c(s^0) \cap \{p_{\mu_i(s)}, \ldots, p_n\} = \emptyset$. Since Ψ_1 holds in s^0 and in s^1, we get that $\forall k, \ 1 \leq k < \mu_i(s)$: $s^1.p_k = s^0.p_k$. Therefore, we get that $M_c(s^0) \leq M_c(s^2)$. It remains to show that $M_c(s^2) \leq M_c(s)$. By Equation 9, for all $c'' \notin \{c, c'\}$, $RPU_{c''}(s^1) = R(s)$. By Equations 6 and 7, $RPU_c(s^1) \geq R(s)$ and $RPU_{c'}(s^1) \geq R(s)$. Now assume by way of contradiction that $M_c(s^2) > M_c(s)$. Thus,

$$RPU_c(s^2) = \frac{H_i(c)}{M_c(s^2)} < \frac{H_i(c)}{M_c(s)} = \frac{R(s) \cdot M_c(s)}{M_c(s)} = R(s).$$

Now since $RPU_c(s^2) < R(s) \leq RPU_c(s^1)$, we get that $s^2 = (s^1_{-p}, c)$ for some $p \in \Pi$. A contradiction to Observation 2.

Ψ_5. Since we already showed that s^2 satisfies $\Psi_1 - \Psi_3$, by Equation 8, we get that $\forall c'' \notin \{c, c'\}$, $M_{c''}(s^2) = M_{c''}(s)$. In addition, since $s^0 = (s_{-p_{\mu_i(s)}}, c')$, $\forall c'' \notin \{c, c'\}$, $M_{c''}(s^0) = M_{c''}(s)$. Therefore, we get that $\forall c'' \notin \{c, c'\}$, $M_{c''}(s) = M_{c''}(s^0) = M_{c''}(s^2)$, and thus $M_c(s^2) + M_{c'}(s^2) = M_c(s^0) + M_{c'}(s^0) = M_c(s) + M_{c'}(s)$. Hence, Ψ_5 follows from Ψ_4.

Now together with Theorem 1, we know that every better response dynamics in the game $G_{\Pi, C, H_i(s)}$ that starts at s converges to some configuration s' that satisfies $\Psi_1 - \Psi_5$. Since $s \in T_i$, we get that

$$\begin{aligned} &\forall k, \ 1 \leq k \leq i-1 \ : s.p_k = s_f.p_k \\ &\forall k, i \leq k \leq n : \ s.p_k \in \{s_f.p_i, s_f.p_{i-1}\} \end{aligned}$$

And since $s \neq s_i$, we get that $i \leq \mu_i(s)$. Thus, by Ψ_1, $\forall k, \ 1 \leq k \leq i-1 : s'.p_k = s_f.p_k$. In addition, by Ψ_3, we get that $\forall k, \ i \leq k \leq n, \ s.p_k \in \{s_f.p_i, s_f.p_{i-1}\}$. Therefore, $s' \in T_i$, and the lemma follows from Ψ_1 and Ψ_2.

□