1 A Game-theoretic Characterization of Bitcoin Mining

The mining game is played by a set $\mathbf{P} = \{1, \ldots, m\}$ of players, with $m \geq 2$. In this game, each player has some reward depending on the number of blocks she owns. We denote by $\mathbf{B}$ the set of all possible blocks, and we assume that there is a special block $\varepsilon \in \mathbf{B}$ that is called the genesis block. Moreover, we associate the following functions to these blocks:

- owner : $(\mathbf{B} \setminus \{\varepsilon\}) \to \mathbf{P}$: This function assigns an owner to each block, except for the genesis block that is assumed not to have an owner.

- succ : $\mathbf{B} \times \mathbf{P} \to \mathbf{B}$: This function tells us which block will a player p use to extend the current blockchain when mining on top of a block b. We require this function to be injective and if $\text{succ}(b_1, p) = b_2$, then $\text{owner}(b_2) = p$.

Note that in Bitcoin there are several different blocks that a player p can use to extend the blockchain when mining upon a block b (depending e.g on the ordering of transactions, or the nonce being used to announce the block). Since we are interested primarily in miners' behaviour, we just focus on the owner of the block following b, and do not consider the possibility of two different block belonging to p being added on top of b. Alternatively, if we consider the Bitcoin protocol, we could say that all the different blocks that p can put on top of b are considered equivalent, since they give p the same reward.

To give a game-theoretic characterization of bitcoin mining, we need to formalize the knowledge that each player has. More precisely, given a subset q of $\mathbf{B}$, define $\mathcal{G}(q) = (N, E)$ as a graph satisfying that:

$$\begin{aligned} N &= q \\ E &= \{(b_1, b_2) \in q^2 \mid \exists p \text{ s.t. } \text{succ}(b_1, p) = b_2\} \end{aligned}$$

Then q is said to be a body of knowledge (of a player) if $\mathcal{G}(q)$ is a tree rooted at ε (in particular, $\varepsilon \in q$). If q is a body of knowledge, then we use notation $\mathcal{T}(q)$ instead of $\mathcal{G}(q)$ to make explicit the fact that $\mathcal{G}(q)$ is a tree.

Given a body of knowledge q, we say that the blockchain of q is the longest path in $\mathcal{T}(q)$, if such a path is unique, in which case we denote it by $\text{bc}(q)$. If two or more different paths are tied for the longest, then we say that the blockchain in q does not exists, and we assume that $\text{bc}(q)$ is not defined (so that $\text{bc}(\cdot)$ is a partial function).

On each step, miners looking to maximise their rewards choose a block in the current body of knowledge, and attempt to mine from this block. Thus, in each turn, each of the players race to place the next block in the body of knowledge, and only one of them succeeds. The probability of succeeding is directly related to the comparative amount of hash power available to this player, the more hash power the likely it is that she will mine the next block before the rest of the players. Once a player places a block, this block is added to the current state, obtaining a different body of knowledge, and the game continues from this new state.

Let Q be the set of all possible bodies of knowledge, and let $\mathbf{Q} = Q^m$. Each tuple $\mathbf{q} = (q_1, \ldots, q_m)$ in $\mathbf{Q}$ is a state of the mining game, where each component q_p of $\mathbf{q}$ is a body of knowledge that represents the knowledge of player p.

Given a player $p \in \mathbf{P}$, a block $b \in \mathbf{B}$ and bodies of knowledge $q, q' \in Q$, we denote by $\text{mine}(p, b, q, q')$ the action played in the mining game, in which player p mines block b, places this block in her current body of knowledge q and decides to disclosure a portion q' of q. Thus, action $\text{mine}(p, b, q, q')$ is valid if:

- $b \in q$,

- $\text{succ}(b, p) \notin q$

- $q' \subseteq q \cup \{\text{succ}(b, p)\}$.

Notice that in the previous definition q' is assumed to be a body of knowledge, so that the condition $q' \subseteq q \cup \{\text{succ}(b, p)\}$ is equivalent to the condition that $\mathcal{T}(q')$ is a subtree of $\mathcal{T}(q \cup \{\text{succ}(b, p)\})$ rooted at ε. Moreover, action mine(p, b, q, q') is said to be valid in a state $\mathbf{q} = (q_1, \ldots, q_m)$ in $\mathbf{Q}$ if mine(p, b, q, q') is a valid action and $q = q_p$.

Let $p \in \mathbf{P}$ be a player, $\mathbf{x} = (x_1, \ldots, x_m)$ a state in $\mathbf{Q}$ and mine(p, b, x_p, q) a valid action of p in state $\mathbf{x}$. Then the result of applying mine(p, b, x_p, q) to $\mathbf{x}$ is a state $\mathbf{y} = (y_1, \ldots, y_m)$ in $\mathbf{Q}$ such that:

- $y_p = x_p \cup \{\text{succ}(b, p)\}$, and
- for every $i \in \{1, \ldots, m\}$ such that $i \neq p$, it holds that $y_i = x_i \cup q$.

In a full disclosure scenario all the allowed actions are of the form mine$(p, b, q, q \cup \{\text{succ}(b, p)\})$, that is, all the information of player p after placing b in its body of knowledge is sent to the other players. In this case, we simplify the notation and use mine(p, b, q) instead of mine$(p, b, q, q \cup \{\text{succ}(b, p)\})$.

Given a player $p \in \mathbf{P}$, the set of actions for player p is defined as:

$$\mathbf{A}_p \quad = \quad \{\text{mine}(p, b, q, q') \mid \text{mine}(p, b, q, q') \text{ is a valid action}\}.$$

Given an action $a \in \mathbf{A}_p$ and a state $\mathbf{q} \in \mathbf{Q}$ such that a is a valid action in $\mathbf{q}$, we use $a(\mathbf{q})$ to denote the state resulting of applying a to $\mathbf{q}$. Moreover, we denote by $\mathbf{A}$ the set of all possible actions, that is, $\mathbf{A} = \mathbf{A}_1 \cup \cdots \cup \mathbf{A}_m$.

Given a player $p \in \mathbf{P}$ and a state $\mathbf{q} \in \mathbf{Q}$, the pay-off of player p in $\mathbf{q}$ is denoted by $r_p(\mathbf{q})$. Moreover, assuming that there is a function r_p for each player $p \in \mathbf{P}$, define $\mathbf{R} = (r_1, \ldots, r_m)$ as the pay-off function of the game.

Finally, we assume that the hash power of each player p is given by a number $h_p \in (0, 1)$ that represents the probability that player p succeeds in placing the next block. Thus, we assume that:

$$\sum_{p=1}^{m} h_p \quad = \quad 1,$$

and we define $\mathbf{H} = (h_1, \ldots, h_m)$ as the hash power distribution.

Summing up, from now on we consider an infinite stochastic game $\Gamma = (\mathbf{P}, \mathbf{A}, \mathbf{Q}, \mathbf{R}, \mathbf{H})$ where:

- $\mathbf{P}$ is the set of player.
- $\mathbf{A}$ is the set of possible actions.
- $\mathbf{Q}$ is the set of states.
- $\mathbf{R}$ is the pay-off function.
- $\mathbf{H}$ is the hash power distribution.

1.1 Stationary equilibrium

A strategy for a player $p \in \mathbf{P}$ is a function $s : \mathbf{Q} \to \mathbf{A}_p$. We define $\mathbf{S}_p$ as the set of all strategies for player p, and $\mathbf{S} = \mathbf{S}_1 \times \mathbf{S}_2 \times \cdots \times \mathbf{S}_m$ as the set of combined strategies for the game (recall that we are assuming that $\mathbf{P} = \{1, \ldots, m\}$ is the set of players).

Next we define a notion of how likely is reaching a state using a particular strategy, when starting at some specific state. Formally, given an initial state $\mathbf{q}_0 \in \mathbf{Q}$ and a strategy $\mathbf{s} =$

$(s_1, \ldots, s_m)$ in $\mathbf{S}$, the probability of reaching state $\mathbf{q} \in \mathbf{Q}$ is recursively defined as follows:

$$\mathbf{Pr}^{\mathbf{s}}(\mathbf{q} \mid \mathbf{q}_0) \ = \ \begin{cases} 0 & \text{if } |\mathbf{q}| < |\mathbf{q}_0|, \\ & \text{or if } |\mathbf{q}| = |\mathbf{q}_0| \text{ and } \mathbf{q} \neq \mathbf{q}_0 \\ 1 & \text{if } \mathbf{q} = \mathbf{q}_0 \\ \displaystyle\sum_{\substack{\mathbf{q}' \in \mathbf{Q}\,: \\ |\mathbf{q}'| = k-1}} \mathbf{Pr}^{\mathbf{s}}(\mathbf{q}' \mid \mathbf{q}_0) \cdot \Bigg(\sum_{\substack{p \in \{1,\ldots,m\}\,: \\ s_p(\mathbf{q}') = a \text{ and } a(\mathbf{q}') = \mathbf{q}}} h_p \Bigg) & \text{if } |\mathbf{q}| > |\mathbf{q}_0| \text{ and } |\mathbf{q}| = k \end{cases}$$

We finally have all the necessary ingredients to define the pay-off of a player in a mining game given a particular strategy.

Definition 1. *Let $p \in \mathbf{P}$, $\mathbf{q}_0 \in \mathbf{Q}$, $\mathbf{s} \in \mathbf{S}$, $\beta \in [0, 1]$ and $n \geq 0$. Then the β discounted utility of player p for the strategy $\mathbf{s}$ from the state $\mathbf{q}_0$ in a mining game with n steps, denoted by $u_p^n(\mathbf{s} \mid \mathbf{q}_0)$, is defined as:*

$$u_p^n(\mathbf{s} \mid \mathbf{q}_0) \ = \ \sum_{i=0}^{n} \beta^i \cdot \Bigg(\sum_{\mathbf{q} \in \mathbf{Q}\,:\,|\mathbf{q}| = i} r_p(\mathbf{q}) \cdot \mathbf{Pr}^{\mathbf{s}}(\mathbf{q} \mid \mathbf{q}_0) \Bigg)$$

Moreover, the β discounted utility of player p for the strategy $\mathbf{s}$ from the state $\mathbf{q}_0$ in an infinite mining game, denoted by $u_p(\mathbf{s} \mid \mathbf{q}_0)$, is defined as:

$$u_p(\mathbf{s} \mid \mathbf{q}_0) \ = \ \sum_{i=0}^{\infty} \beta^i \cdot \Bigg(\sum_{\mathbf{q} \in \mathbf{Q}\,:\,|\mathbf{q}| = i} r_p(\mathbf{q}) \cdot \mathbf{Pr}^{\mathbf{s}}(\mathbf{q} \mid \mathbf{q}_0) \Bigg)$$

Given a player $p \in \mathbf{P}$, a combined strategy $\mathbf{s} \in \mathbf{S}$, with $\mathbf{s} = (s_1, \ldots, s_m)$, and a strategy s for player p ($s \in \mathbf{S}_p$), we denote by $(\mathbf{s}_{-p}, s)$ the strategy $(s_1, \ldots s_{p-1}, s, s_{p+1}, \ldots, s_m)$.

Definition 2. *Let $\mathbf{q}_0 \in \mathbf{Q}$, $\mathbf{s} \in \mathbf{S}$, $\beta \in [0, 1]$ and $n \geq 0$. Then $\mathbf{s}$ is a β discounted stationary equilibrium in a mining game with n steps if for every player $p \in \mathbf{P}$ and every strategy s for player p ($s \in \mathbf{S}_p$), it holds that:*

$$u_p^n(\mathbf{s} \mid \mathbf{q}_0) \ \geq \ u_p^n((\mathbf{s}_{-p}, s) \mid \mathbf{q}_0).$$

Moreover, $\mathbf{s}$ is a β discounted stationary equilibrium in the infinite mining game if for every player $p \in \mathbf{P}$ and every strategy s for player p ($s \in \mathbf{S}_p$), it holds that:

$$u_p(\mathbf{s} \mid \mathbf{q}_0) \ \geq \ u_p((\mathbf{s}_{-p}, s) \mid \mathbf{q}_0).$$

2 Full Disclosure Scenario and Two Players

In this section, we consider $\mathbf{P} = \{1, 2\}$, and we assume full disclosure of information between players. For every state $\mathbf{q} \in \mathbf{Q}$ with $\mathbf{q} = (q_1, \ldots, q_m)$, in this case it holds that $q_i = q_j$ for every $i, j \in \{1, \ldots, m\}$. Thus, in this section we simplify the notation and use a body of knowledge q to represent a state of the game, instead of a tuple $\mathbf{q}$ of the form $(q, \ldots, q)$.

For $p \in \{1, 2\}$, we define the indicator function $\chi_p : \mathbf{B} \to \{0, 1\}$ as follows:

$$\chi_p(b) \ = \ \begin{cases} 1 & \text{owner}(b) = p \\ 0 & \text{otherwise} \end{cases}$$

Given a body of knowledge q such that $\text{bc}(q)$ is defined, the length of path $\text{bc}(q)$ is defined as the number of edges in $\text{bc}(q)$, and it is denoted by $|\text{bc}(q)|$. Moreover, given $i \in \{0, \ldots, |\text{bc}(q)|\}$, we use $\text{bc}(q, i)$ to denote the i-th block in the path $\text{bc}(q)$.

Next step is to describe different strategies that the players can use in the game. The first strategy we describe will be called *default*, and it will reflect the desired behaviour of the miners participating in the Bitcoin network. Intuitively, in a state q, a player following this strategy will try to mine upon the final block that appears in the blockchain of q. If the blockchain in state q does not exist, meaning that there are two longest paths from the genesis block, the player will mine on the final block of the path that contains the highest number of her blocks. We call this strategy default, and we define it formally as follows:

$$\text{default}_p(q) = \begin{cases} mine(p, last(\text{bc}(q)), q), & \text{if bc}(q) \text{ exists} \\ mine(p, best(q), q), & \text{if bc}(q) \text{ does not exist} \end{cases}$$

Here $last(\text{bc}(q))$ returns the last block in $\text{bc}(q)$, and $best(q)$ returns the last block of the path that is of maximal length in q, and on which the player p has the highest number of blocks compared to all maximal paths in q. If there is more than one such path, $best(q)$ is the one that is smallest lexicographically. Intuitively, $best(q)$ is the block on which a benevolent player will mine upon when it is not clear what the blockchain is.

2.1 Constant Reward

For a player $p \in \mathbf{P}$, its reward $r_p(q)$ is defined as:

$$r_p(q) \quad = \quad \begin{cases} 0 & \text{if bc}(q) \text{ is not defined} \\ c \cdot \displaystyle\sum_{i=0}^{|\text{bc}(q)|} \chi_p(\text{bc}(q,i)) & \text{otherwise} \end{cases}$$