

APPENDIX A DESCRIPTION OF THE MINING STRATEGY

We introduce some commonly used terms and describe the mining strategy as follows. Say, given an interval I_k , the interval's *absolute change* of hashes during a time period of length Δt is commonly defined as follows,

$$\int_{t_{k-1}+\Delta t}^{t_k+\Delta t} h_i(t)dt - \int_{t_{k-1}}^{t_k} h_i(t)dt \quad (1)$$

The *percent change* is the absolute change with respect to the current interval's hashes $\int_{t_{k-1}}^{t_k} h_i(t)dt$. It is commonly defined as follows,

$$\frac{\int_{t_{k-1}+\Delta t}^{t_k+\Delta t} h_i(t)dt - \int_{t_{k-1}}^{t_k} h_i(t)dt}{\int_{t_{k-1}}^{t_k} h_i(t)dt} \quad (2)$$

The *mining strategy* is the percent change with respect to the time period of length Δt , which is used to describe the rate of the percent change. It is commonly defined as follows,

$$\frac{\int_{t_{k-1}+\Delta t}^{t_k+\Delta t} h_i(t)dt - \int_{t_{k-1}}^{t_k} h_i(t)dt}{\Delta t \int_{t_{k-1}}^{t_k} h_i(t)dt} \quad (3)$$

where the mining strategy has two forms: the *continuous-time mining strategy* and the *discrete-time mining strategy*. Besides, a relationship between the continuous-time mining strategy and the discrete-time mining strategy is as follows: We can use the mathematical property of the continuous-time mining strategy to calculate the hash rate function $\int_{t_{k-1}}^{t_k} h_i(t)dt$, from which the discrete-time mining strategy with $\Delta t = |t_{k+1} - t_k|$ is simply available.

APPENDIX B ESTIMATE DAILY COMPUTING POWER

We introduce the method of estimating computing power. People commonly use the concept of Bitcoin *mining difficulty* to estimate the hash rate of the Bitcoin network.

Let denote the mining difficulty of the Bitcoin network at time t by $\xi(t)$. Bitcoin uses $\xi(t)$ to roughly guarantee that the duration of each block takes 600 seconds on average. Vividly, Bitcoin uses the historical block times to predict future block times. Say, $\xi(t)$ changes every 2,016 blocks time. For each change, its value for the future 2,016 blocks is set, such that the rate for the historical 2,016 blocks equals 600 seconds per block. Hence, the relationship between $\xi(t)$ and the hash rate of the Bitcoin network at time t can be roughly predicted as follows,

$$\sum_{i=1}^N h_i(t) \approx \frac{2^{32}\xi(t)}{600 \text{ secs}} \quad (4)$$

where $t \in I_k$ and the length of I_k equals one day. However, mining difficulty has a *property* that after each change, $\xi(t)$ will remain a constant in a period of 2,016 blocks (i.e., around $2,016 \times 600$ seconds) until the next change. Such a property leads to a problem in Eq. (4), where it predicts that the hash rate of the Bitcoin network should remain a specific value in the next period of $2,016 \times 600$ seconds. Nevertheless, in real-world situations, the hash rate of the Bitcoin network may be different from that specific value.

Therefore, we need to measure the real-world hash-rate related data to adjust the Eq. (4). Based on the Bitcoin mining mechanism, we measure the number of blocks found by mining pools, and have a better estimation on the hash rate of the Bitcoin network at time t as follows,

$$\sum_{i=1}^N h_i(t) \approx \frac{\#_{\text{actual bitcoin}}}{\#_{\text{expected bitcoin}}} \times \frac{2^{32}\xi(t)}{600 \text{ secs}} \quad (5)$$

where $t \in I_k$ and the length of I_k equals one day. The symbol $\#_{\text{actual bitcoin}}$ means the actual number of blocks found on the Bitcoin network in the interval. And, the symbol $\#_{\text{expected bitcoin}}$ means the expected number of blocks to be found on the Bitcoin network in this interval (i.e., $\#_{\text{expected bitcoin}} = \left\lfloor \frac{t_k - t_{k-1}}{600 \text{ secs}} \right\rfloor$ blocks), where each block is mathematically expected to take exactly 600 seconds.

Similarly, the number of hashes of the Bitcoin network in the interval is as follows,

$$\sum_{i=1}^N \int_{t_{k-1}}^{t_k} h_i(t)dt \approx \frac{\#_{\text{actual bitcoin}}}{\#_{\text{expected bitcoin}}} \times \int_{t_{k-1}}^{t_k} \frac{2^{32}\xi(t)}{600 \text{ secs}} dt \quad (6)$$

From Eq. (5), the hash rate of mining pool i at time t can be estimated as follows,

$$h_i(t) \approx \frac{\#_{\text{actual pool } i}}{\#_{\text{expected bitcoin}}} \times \frac{2^{32}\xi(t)}{600 \text{ secs}} \quad (7)$$

where $t \in I_k$ and the length of I_k equals one day. The symbol $\#_{\text{actual pool } i}$ means the actual number of blocks found by mining pool i in the interval.

Similarly, the number of hashes of mining pool i in the interval is as follows,

$$\int_{t_{k-1}}^{t_k} h_i(t)dt \approx \frac{\#_{\text{actual pool } i}}{\#_{\text{expected bitcoin}}} \times \int_{t_{k-1}}^{t_k} \frac{2^{32}\xi(t)}{600 \text{ secs}} dt \quad (8)$$

where the definite integral $\int_{t_{k-1}}^{t_k} \xi(t)dt$ is simple since the value of $\xi(t)$ changes once every 20,160 minutes. And the mining difficulty is a field of Bitcoin block header that is permanently stored in the Bitcoin blockchain. By default, the related parameters about daily computing power are estimated with a reference to Eqs. (5-8).

APPENDIX C PROOFS

C.1. Proof of Proposition 4.5

Proof: Let the number of hashes of new mining pools in the interval I_k be $H_j(I_k)$, where $j \notin [1, N]$ and $H_j(I_k) = 0$. In the next interval I_{k+1} , it will add $H_j(I_{k+1})$ hashes to the Bitcoin network, where $H_j(I_{k+1}) > 0$ since mining activities are profitable. Also we have $R(I_{k+1}) = R(I_k)$ and $\sum_{i=1}^N H_i(I_{k+1}) = \sum_{i=1}^N H_i(I_k)$ since $R(I_k)$ and $\sum_{i=1}^N H_i(I_k)$ are stable. Therefore, the problem of adding $H_j(I_{k+1})$ hashes to the Bitcoin network for maximizing the expected payoff of new mining pools in the next interval I_{k+1} can be formulated as the optimization problem (9).

$$\begin{aligned} & \underset{H_j(I_{k+1})}{\text{maximize}} && \frac{H_j(I_{k+1})R(I_k)}{H_j(I_{k+1}) + \sum_{i=1}^N H_i(I_k)} - cH_j(I_{k+1}) \\ & \text{subject to} && \frac{R(I_k)}{\sum_{i=1}^N H_i(I_k)} > c \end{aligned} \quad (9)$$

It implies that the solutions of either increasing too much computing power or too little, to the Bitcoin network are not optimal. Therefore, we are going to find the optimal one. Specifically, we let the first derivative $\frac{\partial}{\partial H_j(I_{k+1})} \left[\frac{H_j(I_{k+1})R(I_k)}{H_j(I_{k+1}) + \sum_{i=1}^N H_i(I_k)} - cH_j(I_{k+1}) \right] = 0$, and get two saddle points: $H_j(I_{k+1}) = -\sqrt{\frac{R(I_k) \sum_{i=1}^N H_i(I_k)}{c}} - \sum_{i=1}^N H_i(I_k)$, $H_j(I_{k+1}) = \sqrt{\frac{R(I_k) \sum_{i=1}^N H_i(I_k)}{c}} - \sum_{i=1}^N H_i(I_k)$, respectively. The first saddle point can be removed since it does not satisfy the condition of $H_j(I_{k+1}) > 0$. The second saddle point can be proven to be the optimal one since the second order derivative $\frac{\partial^2}{\partial H_j(I_{k+1}) \partial H_j(I_{k+1})} \left[\frac{H_j(I_{k+1})R(I_k)}{H_j(I_{k+1}) + \sum_{i=1}^N H_i(I_k)} - cH_j(I_{k+1}) \right] = -2\sqrt{cR(I_k) \sum_{i=1}^N H_i(I_k)} < 0$. ■

C.2. Proof of Proposition 4.6

Proof: Let the number of hashes of an existing mining pool i in the interval I_k be $H_i(I_k)$, where $i \in [1, N]$, $H_i(I_k) > 0$. Mining pool i will increase/decrease $H_{i'}(I_{k+1})$ hashes to/from the Bitcoin network in the next interval I_{k+1} . Also, $R(I_{k+1}) = R(I_k)$, $\sum_{i=1}^N H_i(I_{k+1}) = \sum_{i=1}^N H_i(I_k)$ since $R(I_k)$ and $\sum_{i=1}^N H_i(I_k)$ are stable. Moreover, there are no new mining pools. Therefore, the problem of increasing/decreasing $H_{i'}(I_{k+1})$ hashes to/from the Bitcoin network for maximizing the expected payoff of mining pool i in the next interval can be formulated as the optimization problem (10).

$$\begin{aligned} & \underset{H_{i'}(I_{k+1})}{\text{maximize}} && [H_{i'}(I_{k+1}) + H_i(I_k)] \times \\ & && \left[\frac{R(I_k)}{H_{i'}(I_{k+1}) + \sum_{i=1}^N H_i(I_k)} - c \right] \\ & \text{subject to} && \frac{R(I_k)}{\sum_{i=1}^N H_i(I_k)} > c \end{aligned} \quad (10)$$

We are going to find the optimal solution. And similar to the previous optimization problem (9), the optimal

solution is $\sqrt{\frac{R(I_k)(-H_i(I_k) + \sum_{i=1}^N H_i(I_k))}{c}} - \sum_{i=1}^N H_i(I_k)$. It means that for an existing mining pool i whose hashes satisfy $H_i(I_k) < \left(1 - \frac{c \sum_{i=1}^N H_i(I_k)}{R(I_k)}\right) \sum_{i=1}^N H_i(I_k)$, the optimal strategy is to add $\sqrt{\frac{R(I_k)(-H_i(I_k) + \sum_{i=1}^N H_i(I_k))}{c}} - \sum_{i=1}^N H_i(I_k)$ hashes to the Bitcoin network in the next interval. Otherwise, the optimal solution is to decrease $\sum_{i=1}^N H_i(I_k) - \sqrt{\frac{R(I_k)(-H_i(I_k) + \sum_{i=1}^N H_i(I_k))}{c}}$ hashes from the Bitcoin network in the next interval. ■

C.3. Proof of Proposition 4.8

Proof: Let the number of hashes of an existing mining pool i in the interval I_k be $H_i(I_k)$, where $i \in [1, N]$, $H_i(I_k) > 0$. Mining pool i will decrease $H_{i'}(I_{k+1})$ hashes from the Bitcoin network in the next interval I_{k+1} where $H_{i'}(I_{k+1}) > 0$ since mining activities are non-profitable. Also, $R(I_{k+1}) = R(I_k)$ and $\sum_{i=1}^N H_i(I_{k+1}) = \sum_{i=1}^N H_i(I_k)$ since $R(I_k)$, $\sum_{i=1}^N H_i(I_k)$ are stable. Therefore, the problem of decreasing $H_{i'}(I_{k+1})$ hashes from the Bitcoin network for maximizing the expected payoff of mining pool i in the next interval can be formulated as the optimization problem (11).

$$\begin{aligned} & \underset{H_{i'}(I_{k+1})}{\text{maximize}} && [-H_{i'}(I_{k+1}) + H_i(I_k)] \times \\ & && \left[\frac{R(I_k)}{-H_{i'}(I_{k+1}) + \sum_{i=1}^N H_i(I_k)} - c \right] \\ & \text{subject to} && \frac{R(I_k)}{\sum_{i=1}^N H_i(I_k)} \leq c \end{aligned} \quad (11)$$

We are going to find the optimal solution. There are two cases.

Case 1. If an existing mining pool i whose hashes satisfy $\frac{R(I_k)}{-H_i(I_k) + \sum_{i=1}^N H_i(I_k)} > c$, then the optimization problem (11) can be represented the same form as the optimization problem (9). Therefore, after applying the proposition 4.5, we get the optimal solution $H_{i'}(I_{k+1}) = \sum_{i=1}^N H_i(I_k) - \sqrt{\frac{R(I_k)(-H_i(I_k) + \sum_{i=1}^N H_i(I_k))}{c}}$ where $H_{i'}(I_{k+1}) > 0$.

Case 2. If an existing mining pool i whose hashes satisfy $\frac{R(I_k)}{-H_i(I_k) + \sum_{i=1}^N H_i(I_k)} \leq c$, then intuitively the optimal solution is to decrease all hashes $H_{i'}(I_{k+1})$ where $H_{i'}(I_{k+1}) = H_i(I_k)$ from the Bitcoin network in the next interval. ■

C.4. Proof of Theorem 4.9

Proof: Given a stable mining revenue R . The number of hashes of the Bitcoin network is $\sum_{i=1}^N H_i(I_k)$ hashes in interval I_k . We are going to prove that $\lim_{n \rightarrow \infty} \sum_{i=1}^N H_i(I_{k+n})$ will converge to a Nash equilibrium point among mining pools. There are two cases.

Case 1. If mining activities are profitable, then we apply the proposition 4.5, and know that extra hashes $\sqrt{\frac{R \sum_{i=1}^N H_i(I_k)}{c}} - \sum_{i=1}^N H_i(I_k)$ will be added to the Bitcoin network by new mining pools. It means that the number of hashes of the Bitcoin network in the next interval will be $\sum_{i=1}^N H_i(I_{k+1}) = \sum_{i=1}^N H_i(I_k) +$

$\left(\sqrt{\frac{R \sum_{i=1}^N H_i(I_k)}{c}} - \sum_{i=1}^N H_i(I_k)\right) = \sqrt{\frac{R \sum_{i=1}^N H_i(I_k)}{c}}$ hashes. So on and so forth, after n intervals, we have $\lim_{n \rightarrow \infty} \sum_{i=1}^N H_i(I_{k+n}) = \frac{R}{c}$.

Case 2. If mining activities are non-profitable, then we repeatedly apply the proposition 4.8, and know that the optimal solution for a major mining pool i to maximize its expected payoff is to decrease $\sum_{i=1}^N H_i(I_k) - \sqrt{\frac{(-H_i(I_k) + \sum_{i=1}^N H_i(I_k))R}{c}} > 0$ hashes from the Bitcoin network in the next interval, and finally go to *Case 1* of this proof. ■

APPENDIX D CASE STUDY

D.1. Mining Revenue Increases Dramatically

We will analyze the best-response strategy if the mining revenue increases dramatically. We make the following assumptions:

Assumption 1.1: Two existing mining pools X , Y , and one new mining pool Z with corresponding hashes of $H_x(I_k)$, $H_y(I_k)$, $H_z(I_k)$ in the interval I_k , where,

$$H_x(I_k) > 0; H_y(I_k) > 0; H_z(I_k) = 0 \quad (12)$$

$$\sum_{i=1}^N H_i(I_k) = H_x(I_k) + H_y(I_k) + H_z(I_k) \quad (13)$$

Assumption 1.2: The mining activities are profitable and satisfy the following conditions,

$$\begin{aligned} H_x(I_k) &< \left(1 - \frac{c \sum_{i=1}^N H_i(I_k)}{R(I_k)}\right) \sum_{i=1}^N H_i(I_k) \\ H_y(I_k) &< \left(1 - \frac{c \sum_{i=1}^N H_i(I_k)}{R(I_k)}\right) \sum_{i=1}^N H_i(I_k) \end{aligned} \quad (14)$$

Assumption 1.3: The mining revenue is highly profitable, where,

$$R(I_{k+1}) = R(I_k) \quad (15)$$

Case 1 $\{Increase, Increase, Increase\}$. The expected payoff of mining pool X , Y , Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_x(I_{k+1})} \quad (16)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_y(I_{k+1})} \quad (17)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (18)$$

Case 2 $\{Increase, Unchange, Increase\}$. The expected payoff of mining pool X , Y , Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_x(I_{k+1})} \quad (19)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_y(I_k)} \quad (20)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (21)$$

Case 3 $\{Unchange, Increase, Increase\}$. The expected payoff of mining pool X , Y , Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_x(I_k)} \quad (22)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_y(I_{k+1})} \quad (23)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (24)$$

Case 4 $\{Unchange, Unchange, Increase\}$. The expected payoff of mining pool X , Y , Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_x(I_k)} \quad (25)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_y(I_k)} \quad (26)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (27)$$

Case 5 $\{Increase, Increase, Unchange\}$. The expected payoff of mining pool X , Y , Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_x(I_{k+1})} \quad (28)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_y(I_{k+1})} \quad (29)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_z(I_k)} \quad (30)$$

Case 6 {Increase, Unchange, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_x(I_{k+1})} \quad (31)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_y(I_k)} \quad (32)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_z(I_k)} \quad (33)$$

Case 7 {Unchange, Increase, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_x(I_k)} \quad (34)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_y(I_{k+1})} \quad (35)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_z(I_k)} \quad (36)$$

Case 8 {Unchange, Unchange, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_x(I_k)} \quad (37)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_y(I_k)} \quad (38)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_z(I_k)} \quad (39)$$

D.2. Mining Revenue Decreases Dramatically

We will analyze the best-response strategy if the mining revenue decreases dramatically. We make the following assumptions:

Assumption 2.1: Two existing mining pools X, Y , and one new mining pool Z with corresponding hashes of $H_x(I_k), H_y(I_k), H_z(I_k)$ in the interval I_k , where,

$$H_x(I_k) > 0; H_y(I_k) > 0; H_z(I_k) = 0 \quad (40)$$

$$\sum_{i=1}^N H_i(I_k) = H_x(I_k) + H_y(I_k) + H_z(I_k) \quad (41)$$

Assumption 2.2: The mining activities are profitable and satisfy the following conditions,

$$\begin{aligned} H_x(I_k) &\geq \left(1 - \frac{c \sum_{i=1}^N H_i(I_k)}{R(I_k)}\right) \sum_{i=1}^N H_i(I_k) \\ H_y(I_k) &\geq \left(1 - \frac{c \sum_{i=1}^N H_i(I_k)}{R(I_k)}\right) \sum_{i=1}^N H_i(I_k) \end{aligned} \quad (42)$$

Assumption 2.3: The mining revenue is slightly profitable, where,

$$R(I_{k+1}) = R(I_k) \quad (43)$$

Case 1 {Decrease, Decrease, Increase}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_x(I_{k+1})} \quad (44)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_y(I_{k+1})} \quad (45)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (46)$$

Case 2 {Decrease, Unchange, Increase}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_x(I_{k+1})} \quad (47)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_y(I_k)} \quad (48)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (49)$$

Case 3 {Unchange, Decrease, Increase}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_x(I_k)} \quad (50)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_y(I_{k+1})} \quad (51)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (52)$$

Case 4 {Unchange, Unchange, Increase}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_x(I_k)} \quad (53)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_y(I_k)} \quad (54)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_{k+1})}{H_x(I_k) + H_y(I_k) + H_z(I_{k+1}) - cH_z(I_{k+1})} \quad (55)$$

Case 5 {Decrease, Decrease, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_x(I_{k+1})} \quad (56)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_y(I_{k+1})} \quad (57)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_{k+1}) + H_y(I_{k+1}) + H_z(I_k) - cH_z(I_k)} \quad (58)$$

Case 6 {Decrease, Unchange, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_{k+1})}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_x(I_{k+1})} \quad (59)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_y(I_k)} \quad (60)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_{k+1}) + H_y(I_k) + H_z(I_k) - cH_z(I_k)} \quad (61)$$

Case 7 {Unchange, Decrease, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_x(I_k)} \quad (62)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_{k+1})}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_y(I_{k+1})} \quad (63)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_k) + H_y(I_{k+1}) + H_z(I_k) - cH_z(I_k)} \quad (64)$$

Case 8 {Unchange, Unchange, Unchange}. The expected payoff of mining pool X, Y, Z in the next interval I_{k+1} will be,

$$\mathbb{E}[f_x(I_{k+1})] = \frac{R(I_{k+1})H_x(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_x(I_k)} \quad (65)$$

$$\mathbb{E}[f_y(I_{k+1})] = \frac{R(I_{k+1})H_y(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_y(I_k)} \quad (66)$$

$$\mathbb{E}[f_z(I_{k+1})] = \frac{R(I_{k+1})H_z(I_k)}{H_x(I_k) + H_y(I_k) + H_z(I_k) - cH_z(I_k)} \quad (67)$$

APPENDIX E

TRANSACTION SIZE AFTER VS. BEFORE SEGWIT

E.1. Description of the Transaction Size

Segregated Witness (SegWit) is a soft fork of the Bitcoin protocol that separates the witness data (i.e., digital signatures) from the total transaction data and allows more base transaction data (i.e., without the witness data) to be stored in a single block. SegWit introduces a new unit named *weight unit* (WU) to measure the size of transaction data and block data, and transitions to a new block limit of 4 million WU. The existing 1 MB block limit still holds for these non-SegWit nodes. The relationship between *weight unit* and *byte* is as follows,

$$\text{weight unit} = \text{base size in byte} \times 3 + \text{total size in byte} \quad (68)$$

where the base size does not include the witness data while the total size includes the witness data. The equation (68) holds for both transaction and block. That is, a transaction's weight unit equals three times of the base transaction size in byte plus the total transaction size in byte. And a block's weight unit equals three times of the base block size in byte plus the total block size in byte. In practice, both weight unit and total size are directly available via querying the Bitcoin full node. For example, we can query the weight unit and the total size of a transaction using the '*bitcoin-cli getrawtransaction true*' command. Similarly, we can query the weight unit and the total size of a block using the '*bitcoin-cli getblock*' command.

Bitcoin also uses *virtual size* in *vbyte* to measure the size of transaction and block, where the relationship between *vbyte* and *weight unit* is as follows,

$$\begin{aligned} \text{virtual size} &= \text{weight unit} / 4 \\ &= \text{base size in byte} + \text{witness data in byte} / 4 \end{aligned} \quad (69)$$

where the equation (69) also holds for both transaction and block. It implies that the new block limit of 4 million WU measured in virtual size is 1 million vbytes. In practice, we can query the virtual size of a transaction using the '*bitcoin-cli getrawtransaction true*' command.

E.2. Comparison of the Transaction Size

We will use both the virtual size and the total size to study the transaction size after versus before SegWit activation. Before SegWit activation, there is no SegWit transaction; hence, all transactions are non-SegWit. After SegWit activation, transactions can be grouped into non-SegWit transactions and SegWit transactions.

TABLE I compares the transaction size in byte after versus before SegWit activation on a SegWit-compatible Bitcoin full node from Feb 26, 2016, to Nov 25, 2020. First, we can see that the average transaction size in byte (i.e., overall) of top mining pools after SegWit activation is larger than its size (i.e., overall) before SegWit activation since the transaction from Bitcoin end-user is getting larger in byte. For example, the average transaction size of AntPool after SegWit is 534.19 bytes, which is larger than its size before SegWit (i.e., 509.66 bytes).

TABLE I

TRANSACTION SIZE IN BYTE AFTER VS. BEFORE SEGWIT ACTIVATION ON A SEGWIT-COMPATIBLE BITCOIN FULL NODE FROM FEB 26, 2016 TO NOV 25, 2020

Mining Pool		Before SegWit (Overall)	After SegWit		
			Non-SegWit	SegWit	Overall
AntPool	Min	127	157	190	157
	Max	99,877	99,946	224,432	224,432
	Mean	509.66	477.04	642.24	534.19
	Median	226	226	283	248
	Std	2095.71	2059.78	3000.18	2427.97
F2Pool	Min	69	104	96	96
	Max	188,477	240,181	224,850	240,181
	Mean	509.79	463.18	637.98	535.77
	Median	226	226	283	249
	Std	2254.78	2202.60	3100.71	2614.74
BTC.com	Min	108	85	190	85
	Max	99,887	99,925	224,447	224,447
	Mean	506.94	479.98	634.51	536.55
	Median	226	226	283	249
	Std	2057.84	2043.16	3086.50	2477.78
ViaBTC	Min	97	85	189	85
	Max	99,905	99,999	188,537	188,537
	Mean	514.57	486.20	659.04	546.17
	Median	226	226	283	249
	Std	2094.55	2118.90	3255.24	2572.04

TABLE II

TRANSACTION SIZE IN VBYTE AFTER VS. BEFORE SEGWIT ACTIVATION ON A SEGWIT-COMPATIBLE BITCOIN FULL NODE FROM FEB 26, 2016 TO NOV 25, 2020

Mining Pool		Before SegWit (Overall)	After SegWit		
			Non-SegWit	SegWit	Overall
AntPool	Min	127	157	109	109
	Max	99,877	99,946	99,826	99,946
	Mean	509.66	477.04	395.39	448.80
	Median	226	226	198	225
	Std	2095.71	2059.78	1788.12	1970.42
F2Pool	Min	69	104	86	86
	Max	188,477	240,181	99,704	240,181
	Mean	509.79	463.18	391.54	433.43
	Median	226	226	200	225
	Std	2254.78	2202.60	1775.51	2036.45
BTC.com	Min	108	85	100	85
	Max	99,887	99,925	99,581	99,925
	Mean	506.94	479.98	391.67	447.65
	Median	226	226	190	225
	Std	2057.84	2043.16	1844.19	1973.10
ViaBTC	Min	97	85	100	85
	Max	99,905	99,999	99,919	99,999
	Mean	514.57	486.20	405.84	458.32
	Median	226	226	198	225
	Std	2094.55	2118.90	1921.27	2052.84

Second, we can see that the average size in byte of the SegWit transaction is larger than the non-SegWit transaction after SegWit activation. For example, the average size of the SegWit transaction of AntPool is 642.24 bytes, which is larger than the average size of the non-SegWit transaction after SegWit activation (i.e., 477.04 bytes). It implies that larger transaction (in byte) prefers SegWit than non-SegWit.

TABLE II compares the transaction size in vbyte after versus before SegWit activation. First, we can see that the average transaction size in vbyte (i.e., overall) of top mining pools after SegWit activation is smaller than its size before SegWit activation, because the witness data in vbyte is only counted as one quarter of its size in byte (see Eq. (69)). For example, the average transaction size in vbyte of AntPool after SegWit is 448.80 vbytes, which is smaller than its size before SegWit (i.e., 509.66 vbytes). Second, we can see

that the average size in vbyte of the SegWit transaction is smaller than the non-SegWit transaction after SegWit activation. For example, the average size of the SegWit transaction of AntPool is 395.39 vbytes, which is smaller than the average size of the non-SegWit transaction after SegWit activation (i.e., 477.04 vbytes).

In summary, first, we find that the average transaction size in byte after SegWit activation is larger than its size before SegWit activation, since transaction from Bitcoin end-user is getting larger in byte; however, the average transaction size in vbyte after SegWit activation is smaller than its size before SegWit activation, because the witness data in vbyte is only counted as one quarter of its size in byte (see Eq. (69)). Second, we find that larger transaction in byte prefers SegWit than non-SegWit since the virtual size of witness data is counted smaller in vbyte under the new block limit of 1 million vbytes (see Eq. (69)).

APPENDIX F

MEASURING DYNAMIC OF DAILY BTC PRICE

We will measure the dynamic of the daily BTC price, and evaluate the percent error between the daily closing price and the daily BTC price.

A. Measurement of Dynamic

We will measure the dynamic of the daily BTC price and see how does the daily BTC price fluctuate within a day.

Fig. 1 shows the dynamic of the daily BTC price from Feb 26, 2016, to Nov 25, 2020, where the solid blue line represents the daily high price, the dashed black line represents the daily low price, and the red area bounded by the daily high and low BTC prices, represents the dynamic of daily BTC price. Vividly, we can see that the red area of a given day is quite narrow, meaning that the daily BTC price does not fluctuate much within a day, and the dynamic of the daily BTC price (within a day) is negligible on most days.

B. Evaluation of Daily Closing Price

Based on the above measurement of the negligible dynamic of the daily BTC price (within a day), our solution uses the daily closing price to assess the value of the BTCs mined the day. We will evaluate the impact of the dynamic of the daily BTC price on the solution.

Specifically, we use a commonly used term named *percent error* to evaluate the percentage error between the daily closing price and the daily BTC price. Note that the daily BTC price is a *random variable* with a certain dynamic, while the daily closing price is a *constant* value with no dynamic. Commonly, the percent error of the daily closing price with respect to the daily BTC price is defined as follows,

$$\text{percent error} = \max \left\{ \frac{| \text{daily closing price} - \text{daily high price} |}{\text{daily closing price}}, \frac{| \text{daily closing price} - \text{daily low price} |}{\text{daily closing price}} \right\} \quad (70)$$

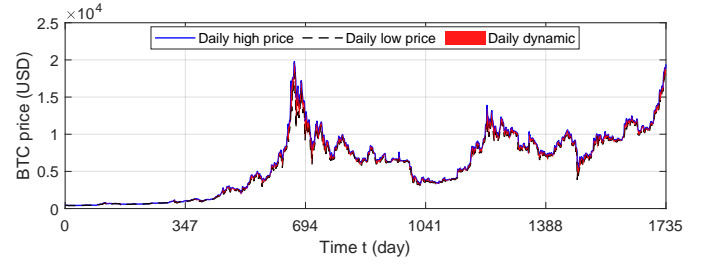


Fig. 1. The dynamic of the daily BTC price from Feb 26, 2016, to Nov 25, 2020, where the solid blue line represents the daily high price, the dashed black line represents the daily low price, and the red area between the daily high and low prices represents the dynamic of daily BTC price (within a day).

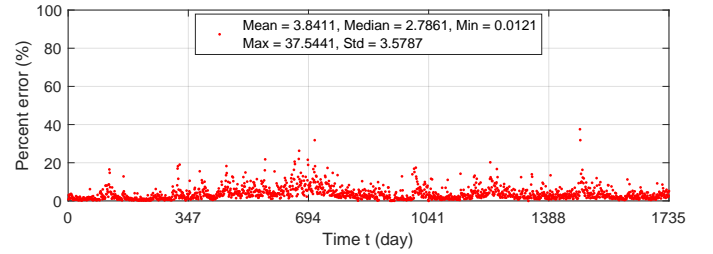


Fig. 2. The percent error of the daily closing price with respect to the daily BTC price from Feb 26, 2016, to Nov 25, 2020 (within a day).

where the symbol $\frac{|\text{daily closing price} - \text{daily high price}|}{\text{daily closing price}}$ means the percent error of the daily closing price with respect to the daily high price. And the symbol $\frac{|\text{daily closing price} - \text{daily low price}|}{\text{daily closing price}}$ means the percent error of the daily closing price with respect to the daily low price.

Fig. 2 shows the percent error of the daily closing price with respect to the daily BTC price from Feb 26, 2016, to Nov 25, 2020. We can see that it is good to use daily closing price to price the BTCs mined the day due to the negligible dynamic of the daily BTC price. And the average percent error of the daily closing price with respect to the daily BTC price is negligible (i.e., 3.8411%).

In summary, first, we find that the daily BTC price does not fluctuate much within a day, and the dynamic of the daily BTC price is negligible on most days. Second, we find that it is good to use the daily closing price to assess the value of the BTCs mined the day since the average percent error between the daily closing price and the daily BTC price is negligible (i.e., 3.8411%).

APPENDIX G

THE LOW-PROFITABLE RATIO OF DAILY MINING REVENUE TO DAILY COMPUTING POWER

We will study the low-profitable ratio of daily mining revenue to daily computing power.

As shown in Fig. 5 in the manuscript, the Bitcoin network moves towards a stable and relatively low-profitable state, where the average ratio of daily mining revenue to daily computing power is around 1.2 USD/(EH/Day). The low-profitable state mainly includes two costs: the electricity cost and the machine cost.

A. The Electricity Cost

The world's average electricity cost is 0.136 USD per kWh as of December 2021¹. To make it simple, we take the latest model² of Antminer S19 Pro as an example. The power efficiency of Antminer S19 Pro is 29.5 J/TH³ where 1 watt equals 1 joule per second. Hence, we have the power efficiency as follows,

$$\begin{aligned} 29.5J/TH &= 29.5W/((1k \times 3600s/h)/TH) \\ &= 0.000008194kWh/TH \end{aligned} \quad (71)$$

The maximum throughput of Antminer S19 Pro is 110 TH/s². Therefore, the daily electricity cost of hosting an Antminer S19 Pro is as follows,

$$\begin{aligned} 0.136USD/kWh \times 0.000008194kWh/TH \times \\ 110TH/s \times 24h \times 3600s/h &= 10.591USD/Day \end{aligned} \quad (72)$$

and hence, the daily electricity cost per EH per day is as follows,

$$\begin{aligned} 10.591USD/Day &= \\ 10.591USD/((110TH/s \times 24h \times 3600s/h) \times 10^{-6}EH) \\ &= 1.1143USD/(EH/Day) \end{aligned} \quad (73)$$

B. The Machine Cost

The machine cost of Antminer S19 Pro is 4400USD³. We consider a machine lifetime of 10 years. Therefore, the daily machine cost is as follows,

$$4400USD/(10y \times 365days/year) = 1.2054USD/day \quad (74)$$

and hence, the daily machine cost per EH per day is as follows,

$$\begin{aligned} 1.2054USD/day &= \\ 1.2054USD/((110TH/s \times 24h \times 3600s/h) \times 10^{-6}EH) \\ &= 0.1268USD/(EH/Day) \end{aligned} \quad (75)$$

Therefore, the cost of the mining activities is as follows,

$$\begin{aligned} 1.1143USD/(EH/Day) + 0.1268USD/(EH/Day) \\ = 1.2411USD/(EH/Day) \end{aligned} \quad (76)$$

In summary, first, we find that the cost of the mining activities is 1.2411USD/(EH/Day), which well validates the experiment result in Fig. 5 (i.e., with a percent error of 3.425%). Second, we find that the electricity cost dominates the cost of the mining activities.

C. Events of Hash Rate Dropping

Based on the data⁴, we analyzed the events of hash rate dropping of mining pools up to Aug 31, 2022, and organized these events into Table III as shown below, where the source code of a concrete sample in Table III is available to the public⁵.

The results show that most events of hash rate dropping (e.g., May 18, 2020, Jan 10, 2020) are due to low mining revenue per computing power. That is, most of the trace data validate our model (see Theorem 4.9 in the manuscript) and result (see Fig. 5 in the manuscript). Nevertheless, we still found two events of hash rate dropping (i.e., Apr 16, 2021, and Jun 29, 2021) in which mining activities were profitable but our current model cannot explain. Specifically, for the former event of the hash rate dropping on Apr 16, 2021, we believe it is related to the unexpected event of Xinjiang Grid Blackouts⁶. And this unexpected event resulted in a significant dropping in hash rate of several major Bitcoin mining pools such as AntPool and BTC.com⁷. For the latter event of the hash rate dropping on Jun 29, 2021, we believe it is related to a government ban in China. As reported by mainstream media, the government and regulators of China moved to effectively shut down all crypto mining operations across the country since May 2021 for the purpose of mitigating financial risks⁸,⁹. And this government-mandated ban in China resulted in a 38% drop in the global hash rate by July 2021¹⁰.

In summary, our model well validates most events of hash rate dropping of mining pools. Though certain non-economic behaviors (e.g., unexpected Grid Blackouts, government regulation) exist, the effect of rational strategies could be evident in the long term, e.g., the ratio of a pool's daily mining revenue to the pool's daily computing power in Fig. 5.

⁴<https://www.theblock.co/data/on-chain-metrics/bitcoin/bitcoins-daily-real-time-hashrate-by-pool>

⁵<https://github.com/Canhui/AppendixBTC/blob/main/codeTB1.m>

⁶<https://news.bitcoin.com/bitcoin-hashrate-drops-xinjiang-blackouts-blamed-btc-price-slides/>

⁷<https://www.nasdaq.com/articles/bitcoin-mining-hash-rate-drops-as-blackouts-instituted-in-china-2021-04-16>

⁸https://www.voanews.com/a/east-asia-pacific_voa-news-china_why-china-cracking-down-bitcoin/6207912.html

⁹<https://www.jbs.cam.ac.uk/insight/2021/geographic-shift/>

¹⁰<https://www.reuters.com/world/china/china-central-bank-vows-crackdown-cryptocurrency-trading-2021-09-24/>

¹https://www.globalpetrolprices.com/electricity_prices/

²<https://www.asicminervalue.com/miners/bitmain/antminer-s19-pro-110th>

³<https://shop.bitmain.com/release/AntminerS19Pro/overview>

TABLE III

EVENTS OF HASH RATE DROPPING OF MINING POOLS UP TO AUG 31, 2022 (REMARK: $c=1.2 \text{ USD}/(\text{EH}/\text{DAY})=0.1037 \text{ USD}/(\text{TH}/\text{s})$, ALSO SEE FIG. 5 IN THE MANUSCRIPT)

Day	Mining Pool	Hash Rate (EH/s)	Hash Rate Drops Significantly?	Mining Revenue over Computing Power (USD/(TH/s))	Profitability	Pure Conditional Strategy
Mar 18, 2020	AntPool	10.07	Y	$0.08 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	F2Pool	16.68	Y	$0.08 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	BTC.com	12.57	Y	$0.08 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	ViaBTC	7.09	Y	$0.08 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
May 20, 2020	AntPool	12.06	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	F2Pool	16.08	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	BTC.com	12.65	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	ViaBTC	6.01	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
Jun 10, 2020	AntPool	12.19	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	F2Pool	18.42	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	BTC.com	14.33	N	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	ViaBTC	6.11	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
Oct 29, 2020	AntPool	12.89	Y	$0.1 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	F2Pool	17.77	Y	$0.1 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	BTC.com	14.28	Y	$0.1 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	ViaBTC	9.71	Y	$0.1 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
Apr 16, 2021	AntPool	19.45	Y	$0.4 (> c)$	Profitable	-
	F2Pool	24.07	Y	$0.4 (> c)$	Profitable	-
	BTC.com	16.14	Y	$0.4 (> c)$	Profitable	-
	ViaBTC	14.66	Y	$0.4 (> c)$	Profitable	-
Jun 29, 2021	AntPool	10.87	Y	$0.23 (> c)$	Profitable	-
	F2Pool	10.32	Y	$0.23 (> c)$	Profitable	-
	BTC.com	9.93	Y	$0.23 (> c)$	Profitable	-
	ViaBTC	19.64	Y	$0.23 (> c)$	Profitable	-
Jun 23, 2022	AntPool	30.39	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	F2Pool	10.74	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	BTC.com	10.17	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	ViaBTC	19.56	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
Jul 11, 2022	AntPool	31.76	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	F2Pool	29.8	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 1)
	BTC.com	8.57	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)
	ViaBTC	19.3	Y	$0.09 (\leq c)$	Non-Profitable	Proposition 4.8 (Case 2)