An Economic Analysis of Difficulty Adjustment Algorithms in Proof-of-Work Blockchain Systems*

Shunya Noda† Kyohei Okumura‡ Yoshinori Hashimoto§

First Draft: June 26, 2019; Current Draft: February 10, 2020

Abstract

The design of the *difficulty adjustment algorithm* (DAA) of the Bitcoin system is vulnerable as it dismisses miners' strategic responses to policy changes. We develop an economic model of the Proof-of-Work based blockchain system. Our model allows miners to pause operation when the expected reward is below the shutdown point. Hence, the supply of aggregate hash power can be elastic in the cryptocurrency price and the difficulty target of the mining puzzle. We prove that, when the hash supply is elastic, the Bitcoin DAA fails to adjust the block arrival rate to the targeted level. In contrast, the DAA of another blockchain system, Bitcoin Cash, is shown to be stable even when the cryptocurrency price is volatile and the supply of hash power is highly elastic. We also provide empirical evidence and simulation results supporting the model's prediction. Our results indicate that the current Bitcoin system might collapse if a sharp price reduction lowers the reward for mining denominated in fiat money. However, this crisis can be prevented through the upgrading of DAA.

JEL Codes: C61, D47, G20, L86

Keywords: Blockchain, Proof-of-Work, Difficulty Adjustment, Mining, Market Design

*This paper was previously titled as "A Lucas Critique to the Difficulty Adjustment Algorithm of the Bitcoin System." We are grateful to Shumpei Goke, Shota Ichihashi, Fuhito Kojima, Hitoshi Matsushima, Daisuke Oyama, Takuo Sugaya, and all the seminar participants in the University of British Columbia, the University of Tokyo, Waseda University, the Scaling Bitcoin Workshop 2019 at Tel Aviv, and Otaru University of Commerce for helpful comments. All remaining errors are our own.

†Contact Author. Vancouver School of Economics, University of British Columbia, 6000 Iona Dr, Vancouver, BC V6T 1L4 Canada. E-mail: shunya.noda@gmail.com Noda has been supported by the Funai Overseas Scholarship and the E. K. Potter fellowship.

‡Graduate School of Economics, University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. E-mail: kyohei.okumura@gmail.com

§BUIDL, Ltd., Neutrino, 12-10, Sakuragaoka, Shibuya-ku, Tokyo-to, 150-0031, Japan. E-mail: yoshi.h@buidl.jp The opinions expressed in this article are Hashimoto's own and do not reflect the view of BUIDL, Ltd.

1 Introduction

Bitcoin is the oldest and largest blockchain platform intended for a peer-to-peer electronic cash system, which is often referred to as *cryptocurrency*. In contrast to traditional payment systems, in which a trusted party manages the history of transactions in a centralized way, the Bitcoin system (and blockchain in general) does not rely on trust in any individual party. Instead, the Bitcoin system is developed as a protocol from which no agent has an incentive to deviate. The Bitcoin system has attracted many economists' interest because it might improve the users' welfare through reduction of the transaction cost.[1]

We study the properties of Bitcoin as an electronic payment system. Users constantly submit transaction requests, and the Bitcoin system is demanded to process them rapidly. However, because the Bitcoin system is decentralized, block generation that occurs too rapidly would increase the risk of synchronization failures. To balance these two conflicting goals, Bitcoin set a policy target to append a new block (a set of new transactions validated by the system) every 10 minutes. Achieving this policy target is not trivial because no institution controls the timing of generating a new block. Although the Bitcoin system incorporates a mechanism to solve this problem (the *difficulty adjustment algorithm*, described in detail later), its behavior at the equilibrium has not been well examined. The goal of this study is to characterize the condition under which the Bitcoin system can successfully stabilize the time interval of block generation. Using this insight, we also make a policy recommendation that will improve the design of the Bitcoin system.

We will explain the mechanics of the Bitcoin system to describe our research question and contribution in more detail. The Bitcoin system allows anyone to work as a record-keeper, called a *miner*. However, if a single miner could create a new block too often, then he would have the power to take advantage of the system.[2] To avoid this, the Bitcoin system

[1] For example, Malinova and Park (2017) Tinn (2018), Aoyagi and Adachi (2019), and Cong and He (2019) study the effect of the mitigation of information friction. Huberman, Leshno, and Moallemi (2019a,b) contrast the Bitcoin system with traditional systems in terms of the existence of the monopolist (trusted central institution).

[2] For example, the miner can successfully make a double-spending attack, in such a case. For the detail,

incorporates a consensus mechanism called *Proof-of-Work* (PoW) (Dwork and Naor, 1992; Back, 2002). The PoW works by selecting a block creator randomly from the set of active miners. Each miner's probability of being selected is proportional to the computational cost expended. This feature makes it costly for each miner to create new blocks frequent enough to attack the system successfully.

More specifically, to create a new block, a miner must find a number called *nonce* satisfying the following property: when the block data (the information about the previous block and newly validated transactions) and the nonce are jointly input into a cryptographic hash function, SHA-256, the returned hash value is numerically smaller than a parameter, called "*difficulty target*." The difficulty target is a number specified by the system. For each input, SHA-256 returns a (virtually) ex ante unpredictable value;[3] thus, each *hash attempt* (compute a hash value with one nonce) is equivalent to drawing a lottery. Miners try many different nonces until they win, i.e., the returned hash value is smaller than the difficulty target. When a miner "wins the prize" and becomes a block creator, he is rewarded with Bitcoins by the system and users. Miners can increase the probability of winning the prize by spending more computational resources to increase the number of trials.

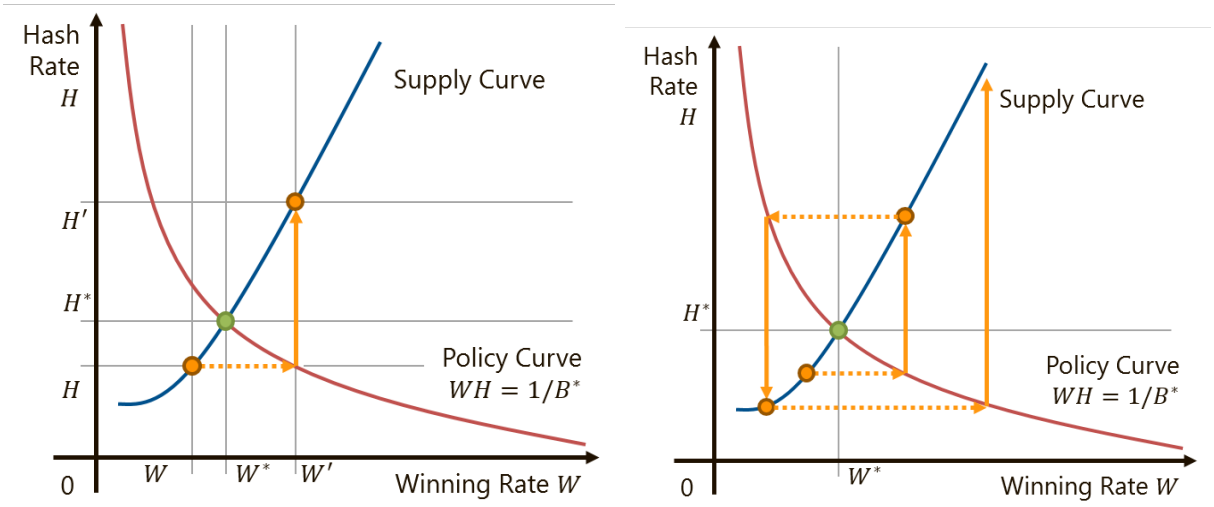
Let W be the *winning rate* (the winning probability per each hash attempt), and H be the *hash rate* (total number of hash attempts made in the world per unit time). Since W is small,[4] the number of blocks generated in a time interval approximately follows a Poisson process with arrival rate WH. The Bitcoin system aims at generating a new block every 10 minutes ($=: B^*$) on average. To achieve this, the system should choose the winning rate W to satisfy $WH = 1/B^*$.

The hash rate is not constant over time because (i) technology development improves computational efficiency, and (ii) miners are allowed to enter/exit the market anytime. To

see Antonopoulos (2014) or Narayanan, Bonneau, Felten, Miller, and Goldfeder (2016).

[3]By the property of SHA-256 (or cryptographic hash functions in general), it is infeasible to infer the nonce from the hash value except by trying many nonces. Hence, before inputting the nonce, a miner has no information about the hash value.

[4]In April 2019, the per attempt winning probability was 3.65×10^{-23}.



(a) One Time (b) Long Run

Figure 1: Difficulty adjustment executed by the Bitcoin system (**DAA-1**). The difficulty adjustment (W to W') influences the hash supply (H to H'), but this indirect effect is not taken into account. Hence, the adjustment always overshoots the ideal level, W^* and H^*.

adjust the winning rate to the current situation, Bitcoin (and other PoW-based blockchain systems) incorporates a *difficulty adjustment algorithm* (DAA). The Bitcoin DAA adjusts the difficulty target every 2016 blocks. Although the hash rate is not observable, it can be estimated from the realized block times. Our analysis articulates that the Bitcoin DAA can be interpreted as a sample analog of the policy target equation, $WH = 1/B^*$. The system estimate the realized hash rate $\hat{H}$ from the historical data and decides the winning rate of the next 2016 blocks so that the targeted block arrival rate is achieved along with the estimated hash rate; i.e., specify the new winning rate W' to satisfy $W'\hat{H} = 1/B^*$ (this is illustrated as dotted arrows of Figure 1).

The main contribution of this study is to show that the Bitcoin DAA may fail to stabilize the block arrival rate. The Bitcoin DAA is subject to the Lucas critique (Lucas, 1976): it dismisses miners' systematic response against the change in the winning rate. The behavior of the winning rate and hash rate under the Bitcoin system are isomorphic to that of the price and quantity in the cobweb model (Kaldor 1934). If the system increases the winning rate from W to W', miners can earn a larger expected reward for each attempt. It encourages

miners to spend more computation power on mining (i.e., turn on relatively inefficient mining facilities they already own). Thus, the hash rate supplied under the new winning rate W', denoted by H', is larger than H (the solid arrows of Figure 1). Because W' is chosen to satisfy $1/B^* = W'\hat{H}$ ($\approx W'H$), the block arrival rate in the next period, $W'H'$, is systematically larger than $1/B^*$. Furthermore, when the hash supply is elastic (i.e., miners sensitively respond to the change in the expected reward), the updated block arrival rate $W'H'$ can be *even more distant* from the policy target $1/B^*$, compared with the previous one WH (Figure 1a). In such a case, just like classical cobweb dynamics, the block arrival rate never converges to the policy target even in a long run (Figure 1b). Instead, it oscillates and diverges (Theorem 1).

The hash supply is indeed elastic when the expected reward is low. Energy consumption accounts for a large fraction of mining costs. When the reward becomes lower than the electricity cost, miners naturally power off their mining facilities. Hence, when the reward stays near the miners' shutdown points, the hash supply becomes sensitive to the reward. In addition, since the Bitcoin price has been volatile, it is possible that the reward reaches this level due to a price drop. Our empirical analysis suggests that the reward was nearing this critical level in November 2018. (Section 4)

We also show that the DAA used in another blockchain platform, Bitcoin Cash, successfully stabilizes the block time. The DAA of Bitcoin Cash adjusts the difficulty target every period, using the moving average of the block time as an estimator of the hash rate. Unlike the DAA of Bitcoin, the hash rate is estimated from the block data generated with diverse difficulty targets. Thanks to this feature, the DAA of Bitcoin Cash makes a better estimate of the ideal level of the hash rate, H^*; thus, it is stable even when the hash supply is highly elastic (Theorem 2). In addition, we show that other DAAs proposed to existing blockchain system are also stable (Section 6).

Finally, we provide counterfactual simulation results that replicate the situation in late 2018. Consistent with our theoretical prediction, the Bitcoin DAA fails to create new blocks

at a constant rate. In contrast, the DAA of Bitcoin Cash successfully generates new blocks in even intervals. This result indicates that the DAA of Bitcoin Cash outperforms the Bitcoin DAA. Further, we argue that miners may not agree to change the DAA because miners with high marginal costs benefit from oscillation. (Section 5)

The remainder of the paper is organized as follows. Section 2 describes our model of the blockchain economy. In Section 3, we analyze the behavior of the economy around the steady state. Section 4 presents empirical results, and Section 5 details simulation results. Section 6 studies alternative DAA design. Section 7 reviews the literature. We make concluding remarks in Section 8.[5]

2 Model

We consider an environment of a single PoW-based cryptocurrency system. We refer to the currency unit of the cryptocurrency as the *coin*, and refer to the fiat money as the *dollar*. Although we follow the terminology used by the Bitcoin system for explanation, our analysis is applicable to a broad class of blockchain systems that require the use of computer processing power for miners to add new blocks.

2.1 Block Generation

Time is discrete and indexed by $t = 0, 1, 2, \ldots$. Each t represents the height (index) of the block. We assume that all the miners use their hash power on the single chain (of blocks).[6] We move to the next period (block) if and only if a miner successfully creates a new block.

As explained in the introduction, the task required for miners before creating a new block is mathematically isomorphic to win a prize in the lottery. The probability that a miner

[5]The codes and notebooks for our simulation is available in the following repository. https://github.com/okuchap/DAA_Lucas

[6]Biais, Bisiere, Bouvard, and Casamatta (2019) show that mining the longest chain without forking is a Markov perfect equilibrium. They also show that there is another equilibrium that generates forks. We ignore the multiplicity of equilibria and assume that all of the miners are working on a single chain. Even if all the miners mine the longest chain, the current Bitcoin system will be proven to be vulnerable.

successfully creates a new block t with one hash attempt is denoted by $W(t)$. We call $W(t)$ the *winning rate* in period t. The system controls the winning rate by choosing the target, and the winning rate of each block is recorded in the blockchain.

Since the winning rate is small, the total number of attempts needed for creating a new block approximately follows $\text{Exp}(W(t))$, where $\text{Exp}(\lambda)$ represents the exponential distribution with intensity λ. Let $H(t)$ be the total number of hash attempts per unit time. We call $H(t)$ the *hash rate*. Let $B(t)$ be the physical time (minutes) needed for creating the t-th block. We call $B(t)$ the *block time*. By the above arguments, we have

$$B(t) = \frac{\delta(t)}{W(t)H(t)}, \tag{1}$$

$$\delta(t) \sim \text{Exp}(1), \text{i.i.d.} \tag{2}$$

This implies that $B(t) \sim \text{Exp}(W(t)H(t))$, and therefore,

$$\mathbb{E}[B(t)] = \frac{1}{W(t)H(t)}.$$

In the Bitcoin system, a miner is required to record the timestamp of block creation. By taking the difference of two adjacent blocks, we can calculate the block time $B(t)$. Hence, $B(t)$ is observable.

Remark 1 (Winning Rate vs Difficulty). The information about the winning rate is stored as the *difficulty* in the blockchain. Hence, many previous works study the behavior of the difficulty, rather than the winning rate. The winning rate is proportional to the reciprocal of the difficulty. Given the difficulty, the Bitcoin system chooses the target according to the following rule:

$$\begin{aligned}\text{target of block } t &= \frac{\text{target of the genesis (initial) block}}{\text{difficulty of block } t} \\ &= 2^{256-32}/(\text{difficulty of block } t).\end{aligned}$$

For each nonce, SHA-256 returns a 256-bit number uniformly at random, and a block is created if and only if the returned hash value is smaller than the target. Hence,

$$W(t) = (\text{target of block t})/2^{256} = 1/[2^{32} \cdot (\text{difficulty of block } t)].$$

We study the behavior of the winning rate rather than the difficulty because the winning rate is mathematically more tractable. However, be cognizant that these two are one-to-one.

Remark 2 (Accuracy of Timestamps). The timestamps are written by the block creator, and not necessarily accurate. However, if a miner writes a timestamp that is very inconsistent with the current situation, the Bitcoin system does not accept the created block. Accordingly, timestamps are not too inaccurate and can be assumed correct when we take a moving average of a sufficient number of block times.

2.2 Difficulty Adjustment Algorithm

Let B^* be the targeted average block time. Recall that the winning rate $W(t)$ is controlled by the system through determination of the difficulty target. Ideally, we would like to set

$$W(t) = \frac{1}{B^* H(t)} \tag{3}$$

so as to satisfy $\mathbb{E}[B(t)] = B^*$. However, the hash rate $H(t)$ is not directly observable because the number of unsuccessful attempts is not reported. In addition, $H(t)$ changes over time due to changes in economic incentives and technological developments. Accordingly, (3) is not implementable.

Alternatively, cryptocurrency systems have *difficulty adjustment algorithms* (DAA) that estimate the current hash rate $H(t)$ by taking a moving average of the past block times, and using the estimate to choose the winning rate $W(t)$.

This study intensively analyzes the performance of two DAAs, **DAA-1** and **DAA-2**.

DAA-1 has been used in Bitcoin since its launch. **DAA-2** was proposed by Séchet (2017) and has been used in Bitcoin Cash since November 2017 (Bitcoin ABC, 2018).

2.2.1 DAA-1 (Bitcoin)

In **DAA-1**(T), the winning rate is adjusted every T blocks. We refer to a set of T blocks that share the same winning rate as a *section*. Sections are indexed by $n = 0, 1, 2, \ldots$. By definition, block t belongs to section n if and only if $t \in \{Tn, Tn+1, \ldots, T(n+1)-1\}$. Denote the winning rate of section n by $w(n)$ ($W(t) = w(n)$ whenever block t belongs to section n). **DAA-1**(T) updates $w(n)$ according to the following rule:

$$w(n+1) = \frac{\sum_{t=Tn}^{T(n+1)-1} B(t)}{T \cdot B^*} \cdot w(n). \tag{4}$$

Note that the maximum likelihood estimator of the intensity of exponentially distributed variables is the reciprocal of their sample mean. Accordingly, when we assume the hash rate to be constant for previous T periods,

$$\hat{H}(t) = \left[\frac{\sum_{s=t-T+1}^{t} W(s)B(s)}{T} \right]^{-1} \tag{5}$$

would be an efficient estimator of the historical hash rate H (see Appendix B). Let us denote the maximum likelihood estimator of the historical hash rate in section n by $\hat{H}(n)$. Then, (4) is rewritten as

$$w(n+1) = \frac{1}{B^* \hat{H}(n)}.$$

In this sense, (4) is can be interpreted as a sample analog of (3). However, the hash rate is estimated from the historical data, and therefore, it is lagged: (5) is an estimator of the hash rate in section n, rather than that in section $n+1$.

Remark 3 (Design Intention). It is ambiguous whether our sample-analog interpretation is the original design intention of the Bitcoin developers. The white paper of Bitcoin

(Nakamoto, 2008) does not involve a concrete formula for the difficulty adjustment.[7] To our knowledge, the official documentation provided by Bitcoin.org does not explain the theoretical justification for the Bitcoin DAA either.[8]

2.2.2 DAA-2 (Bitcoin Cash)

Unlike **DAA-1**, **DAA-2** updates the winning rate every period. Every period, **DAA-2**(T) takes a moving average of the past T block times and use it to adjust the winning rate. The recurrence relation is represented as follows:

$$W(t+1) = \frac{\sum_{s=t-T+1}^{t} B(s)}{B^* \cdot \sum_{s=t-T+1}^{t} \frac{1}{W(s)}}. \tag{6}$$

Recall that $1/W(t)$ is the expected number of hash attempts required to produce block t. Based on this fact, Bitcoin Cash regards $\sum_{s=t=T+1}^{T} 1/W(s)$ as a proxy of the hash attempts exerted in previous T periods. Assuming the hash rate is constant and equal to H in previous T periods, the total hash attempts is equal to $H \sum_{s=t-T+1}^{t} B(s)$. In this sense, **DAA-2** is also a sample analog of (3). Similar to **DAA-1**, the estimator is constructed from the historical data; thus, it is also lagged.

Remark 4 (Bounded Adjustment). Currently implemented DAAs are slightly more complex than (4) and (6) due to practical issues. For example, in the Bitcoin system, if the correction factor $(\sum_{t=Tn}^{T(n+1)-1} B(t)/TB^*)$ is larger than 4 or smaller than 1/4, then 4 or 1/4 are used instead. The goal of this rule is to prevent the difficulty change from being too abrupt. It is

[7] Nakamoto (2008) only states *"To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases"* in p-3.

[8] For example, Blockchain Guide (https://bitcoin.org/en/blockchain-guide) only states that *"If it took fewer than two weeks to generate the 2,016 blocks, the expected difficulty value is increased proportionally (by as much as 300%) so that the next 2,016 blocks should take exactly two weeks to generate if hashes are checked at the same rate. If it took more than two weeks to generate the blocks, the expected difficulty value is decreased proportionally (by as much as 75%) for the same reason."* (accessed on 02/09/2020)

not binding in the neighborhood of the policy target and is irrelevant to the local behavior of DAA around the steady state (Section 3). In addition, this rule turned out not to make a significant influence on our simulation results (Section 5).

2.3 Hash Supply

We assume that there is a continuum of (latent) miners, who can supply hash power by paying some variable cost (e.g., electricity cost). We assume that miners are risk-neutral[9] and price takers, and no switching cost is required to shut down or restart mining facilities.

When a miner successfully creates block t, he is rewarded with $M(t)$ coins. In many cryptocurrency systems, $M(t)$ comprises a seigniorage reward provided by the system and transaction fees paid by users. Both seigniorage reward and transaction fees are recorded in the blockchain.

Hypothetically, we can also design the rule for reward provision. However, many cryptocurrency systems treat $M(t)$ as fixed because (i) a change in seigniorage reward may cause inflation, and (ii) to change the rule for transaction fees, we need to consider the complex incentives of users. In this study, we assume that $M(t)$ is exogenous.[10]

Although the reward is paid in coins, the resources for running mining facilities are procured in dollar-denominated markets. Accordingly, the cost for providing hash power should be paid in dollars. Let $S(t)$ be the *cryptocurrency price* (or the currency exchange rate of dollar/coin) in period t. For simplicity, we assume that $S(t)$ is constant within period t. This assumption is innocuous as long as $B(t)$ is not too large. We assume that $S(t)$ follows a geometric Brownian motion:

$$S(t+1) - S(t) = \mu S(t)B(t) + \sigma S(t)\sqrt{B(t)}\epsilon(t), \tag{7}$$

[9] Instead of risk-neutrality, we can alternatively assume that miners can form a mining pool to share the risk.

[10] Huberman et al. (2019a,b) discuss that when the block time is unstable, transactions are congested, and therefore, users tend to pay a higher reward. We also ignore this feature because the seigniorage reward is currently dominant.

where

$$\epsilon(t) \sim N(0,1), \text{ i.i.d..} \tag{8}$$

We call μ the *drift rate* and σ the *volatility*.

Remark 5 (Exogeneity). Here, we assume that the cryptocurrency price $S(t)$ is exogenous in that it is not affected by the performance of the DAA (i.e., the behavior of the blockchain system). This assumption might be unrealistic. If the system fails to stabilize the block time, then users of the blockchain system feel disutility. If users leave from the system, then the cryptocurrency price $S(t)$ may endogenously decline. Clearly, this endogenous effect could only increase the importance of the block time stabilization. To provide a conservative prediction, we assume that $S(t)$ is independent from the DAA.

For each hash attempt, a miner earns an *expected dollar reward* of

$$R(t) \coloneqq W(t)M(t)S(t). \tag{9}$$

Let $C : \mathbb{R}_+ \to \mathbb{R}$ be the *aggregate variable cost function of hash rate*. $C(H)$ represents the total variable cost (in dollars) required to supply hash rate H. We assume that C is twice differentiable, increasing, and strictly convex. A representative miner's optimization problem is

$$\max_{H \in \mathbb{R}_+} \left\{ R(t)H - C(H) \right\}.$$

Let $MC \coloneqq dC/dH$ be the *marginal cost function*. At the optimum, miners will supply hash rate until its marginal cost coincides the expected reward: the equilibrium $H(t)$ satisfies

$$MC(H(t)) = R(t). \tag{10}$$

Let $\Phi \coloneqq (MC)^{-1}$ be the inverse function of the marginal cost. Then, (10) is rewritten as

$$H(t) = \Phi(R(t)). \tag{11}$$

We call Φ the *hash supply function*. By assumption, Φ is increasing and differentiable. The shape of the hash supply function is not observable from the system.

Note that we assume that the variable cost function is exogenously endowed. In this sense, the incentives for long-term investments (e.g., purchasing new mining facilities) are not explicitly modeled here, and our analysis is applicable even if miners have not chosen the optimal level of long-term investments.

The equation system, (1), (2), (7), (8), (9), (11), and either (4) (if **DAA-1**(T) is used) or (6) (if **DAA-2**(T) is used), represents the dynamics of this blockchain economy.

3 Elasticity and Local (In)Stability

This section studies the condition under which each DAA stabilizes the average block time. For analytical tractability, this section excludes randomness from the equation system: we assume

1. Block time is deterministic: $\delta(t) = \mathbb{E}[\delta(t)] = 1$ holds deterministically.
2. The cryptocurrency price is fixed: $\mu = \sigma = 0$. Hence, $S(t) = S$ for all t.
3. The mining reward is constant: $M(t) = M$ for all t.

Under the above assumptions, (1) and (11) are simplified to

$$B(t) = \frac{1}{W(t)H(t)}, \tag{12}$$

$$H(t) = \Phi(W(t)MS). \tag{13}$$

Accordingly, when the targeted block time B^* is achieved, the winning rate W^* must solve

$$\frac{1}{B^*} = W^*\Phi(W^*MS). \tag{14}$$

This is the policy target. It is also the steady state of the economy.

This section studies the local behavior of the economic system around the steady state, where the linear approximation is justifiable:

$$\log f(X(t)) \approx \log f(X^*) + \frac{f'(X^*)X^*}{f(X^*)}\tilde{X}(t).$$

Here, $\tilde{X}(t)$ denotes the percentage deviation of $X(t)$ from its steady-state value X^*, i.e.,

$$\tilde{X}(t) = \frac{X(t) - X^*}{X^*}.$$

We convert the original non-linear equation system into percentage deviations from the policy target. We say that a DAA is *stable* if for any initial value $\tilde{W}(0)$, the percentage deviation of the winning rate, $\tilde{W}(t)$, converges to 0 as $t \to \infty$ under the DAA. When this is achieved, the block time and hash rate also converges to the targeted level. Our goal in this section is to characterize the condition under which DAAs are stable.

3.1 (In)Stability of DAA-1

When block generation is deterministic and the price is fixed, the block time and hash rate is constant within each section. Slightly abusing the notation, we denote the block time and hash rate in section n by $B(n)$ and $H(n)$, respectively. Then, the evolution of B, w, H is

$$\begin{aligned} B(n) &= \frac{1}{w(n)H(n)}, \\ H(n) &= \Phi(w(n)MS), \\ w(n+1) &= \frac{B(n)}{B^*}w(n). \end{aligned}$$

Performing log-linearization around the policy target, we have

$$\tilde{B}(n) = -\tilde{w}(n) - \tilde{H}(n), \tag{15}$$

$$\tilde{H}(n) = \varepsilon \tilde{w}(n), \tag{16}$$

$$\tilde{w}(n+1) = \tilde{w}(n) + \tilde{B}(n), \tag{17}$$

where ε is the *elasticity* of the hash supply function at the steady state:

$$\varepsilon := \frac{\Phi'(W^*MS)W^*MS}{\Phi(W^*MS)}.$$

The elasticity ε measures the responsiveness of the hash supply with respect to the expected reward, i.e., how much the hash supply changes when the expected reward changes fractionally. Since Φ is increasing, we have $\varepsilon \geq 0$.

Combining (15), (16), and (17), we have

$$\tilde{w}(n+1) = -\varepsilon \tilde{w}(n).$$

It is clear that $\tilde{w}(n) \to 0$ (i.e., $W(t) \to W^*$) if and only if $\varepsilon < 1$.

Theorem 1. *For any T, **DAA-1**(T) is stable if and only if $\varepsilon < 1$.*

As illustrated in Figure 1a, if the elasticity of the hash supply function is larger than one, **DAA-1** fails to adjust the winning rate. This is because its design dismisses the fact that the adjustment of the winning rate also influences the hash rate.

Practically, we cannot always expect that $\varepsilon < 1$. Bitcoin miners use similar mining facilities, hence, their shutdown points are also similar. Once the reward drops down and reaches the shutdown point, miners would not hesitate to turn off their facilities. Hence, when the price and the expected reward remain at a low level, the hash supply becomes highly elastic. See Section 4 for more detailed discussion.

3.2 Stability of DAA-2

The evolution of B, W, and H under **DAA-2**(T) is characterized by (6), (12), and (13). Performing log-linearization around the steady state, we have

$$\tilde{B}(t) = -\tilde{W}(t) - \tilde{H}(t) \tag{18}$$

$$\tilde{H}(t) = \varepsilon \tilde{W}(t) \tag{19}$$

$$\tilde{W}(t+1) = \frac{1}{T} \sum_{s=t-T+1}^{t} \tilde{W}(s) + \frac{1}{T} \sum_{s=t-T+1}^{t} \tilde{B}(s). \tag{20}$$

Combining (18), (19), and (20), we have

$$\tilde{W}(t+1) = -\frac{\varepsilon}{T} \sum_{s=t-T+1}^{t} \tilde{W}(s). \tag{21}$$

The following theorem shows that $\tilde{W}(t)$ converges to zero if and only if $\varepsilon < T$.

Theorem 2. ***DAA-2***(T) *is stable if and only if* $\varepsilon < T$.

Proof Sketch. $\tilde{W}(t)$ converges asymptotically to the steady-state value 0 if and only if all the roots of the following characteristic equation are smaller than one in absolute value.

$$\phi(\lambda) := \lambda^T + \frac{\varepsilon}{T}\lambda^{T-1} + \cdots + \frac{\varepsilon}{T}\lambda + \frac{\varepsilon}{T} = 0.$$

We prove that this is the case if and only if $\epsilon < T$. For the full proof, see Appendix A. □

In contrast to Theorem 1, Theorem 2 is a positive result. To mitigate the randomness from block generation (which we do not model explicitly in this section), the system needs to select relatively large T. In reality, Bitcoin Cash has employed $T = 144$. We assess that $\varepsilon \geq 144$ cannot happen practically. In this sense, the DAA of Bitcoin Cash (**DAA-2**) is much more robust than the DAA of Bitcoin (**DAA-1**).

Remark 6 (With Randomness). If we take into account the randomness in block generation and prices, then our recurrence relation involves white-noise terms. Under **DAA-1**(T), $\tilde{w}(n)$ follows an autoregressive model of order 1 ($AR(1)$), and under **DAA-2**(T), $\tilde{W}(t)$ follows an autoregressive-moving-average model of order T ($ARMA(T)$). In both cases, the process is covariance stationary if and only if all the roots of characteristic equations are smaller than one in absolute values. Hence, the conditions required for "stability" are unchanged.

3.3 Intuition

If the estimator of the hash rate were to return the steady-state hash rate $H^* \coloneqq \Phi(W^*MS)$, then both DAAs would adjust the winning rate to the steady-state level immediately. Accordingly, DAAs should be designed to estimate the steady-state hash rate H^*, rather than the past historical hash rate.

DAA-1 fails to estimate H^*. **DAA-1** uses the block time data generated with an identical winning rate. This feature makes the estimator systematically "biased" from the real target: if the past winning rate was too small, $W < W^*$, the historical hash rate should have been also small $H < H^*$. **DAA-1** estimates H, rather than H^*. Hence, **DAA-1**'s estimate is inconsistent to H^*.

In contrast, **DAA-2** adjusts the winning rate every period, and therefore, the data is generated with various winning rates. Some data are generated under $W < W^*$, and others are generated under $W > W^*$. Accordingly, the historical hash rate also varies around H^*. The estimated hash rate aims at the average of these varied hash rates; thus, the estimate becomes closer to H^*. Thanks to this feature, **DAA-2** makes a better estimate of H^*; hence, **DAA-2** performs better.

3.4 Impulse Response

We show how **DAA-1** and **DAA-2** absorb the effect of one-time persistent price shock. We assume that the price has been constant until period 0; i.e., $S(t) = S'$ for $t = 0, -1, \ldots$.

The winning rate has also stayed at the steady-state level given S'; i.e., $W(t) = W'$ for $t = 0, -1, \ldots$, where W' solves

$$\frac{1}{B^*} = W'\Phi(W'MS').$$

In period 1, the price suddenly and permanently drops to $S < S'$: $S(t) = S$ for $t = 1, 2, \ldots$. Under the new price level, W' is lower than the policy target (and the steady state) W^*, defined by (14). We define W' by

$$\frac{W' - W^*}{W^*} = -0.5.$$

This is equivalent to set $\tilde{W}(t) = -0.5$ for all $t \leq 0$ exogenously.

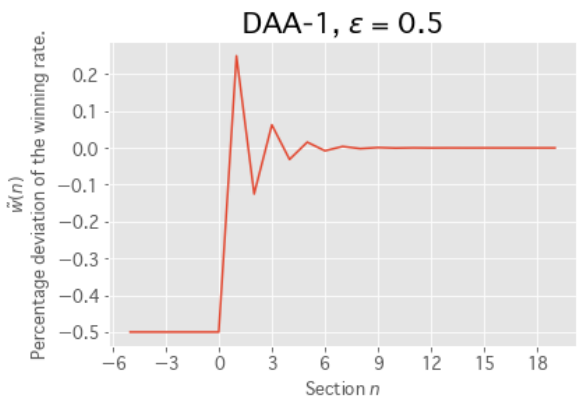
Figure 2 visualizes the evolution of $\tilde{W}(t)$ from period (block) 1. With relatively low elasticity ($\varepsilon = 0.5$), **DAA-1** successfully adjusts the winning rate to the policy target in a few sections (Figure 2a). However, if the elasticity is above one, **DAA-1** oscillates and diverges (Figure 2b). In contrast, Figure 2c illustrates that, with the same elasticity, **DAA-2**(144) makes the winning rate quickly converge to the policy target (recall that 1 section is equivalent to 2016 blocks in the Bitcoin system). Furthermore, even if the elasticity is much higher ($\varepsilon = 10$), **DAA-2**(144) successfully absorbs the price shock (Figure 2d).

Although Figure 2 articulates a characteristic of **DAA-1** and **DAA-2**, it is not a "fair comparison" in that it ignores the volatility of the cryptocurrency price and the randomness of block generation. In Section 5, we propose a simulation result that fully considers the randomness.

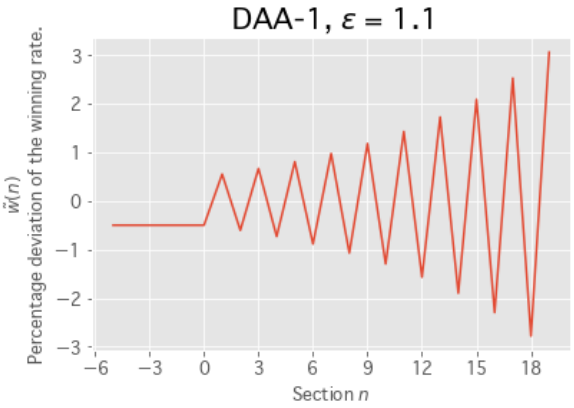
4 Empirical Analysis

4.1 Overview

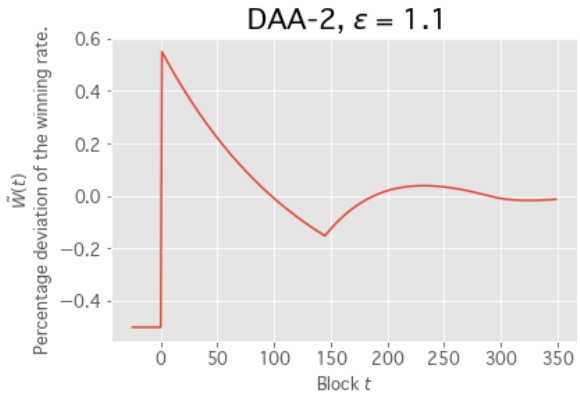
The aim of this section is to articulate how the supply of hash rate empirically responds to the change in per hash rewards.



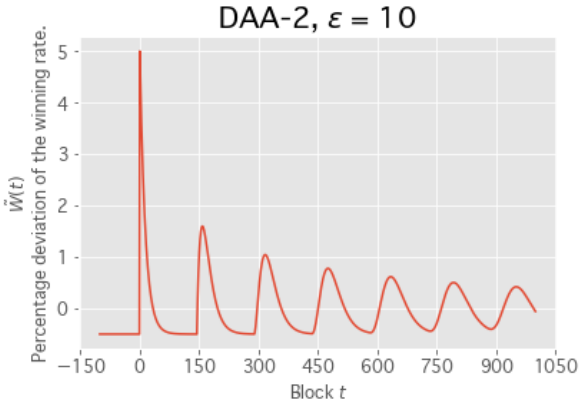
(a) **DAA-1**, elasticity = 0.5



(b) **DAA-1**, elasticity = 1.1



(c) **DAA-2**(144), elasticity = 1.1



(d) **DAA-2**(144), elasticity = 10

Figure 2: How **DAA-1** and **DAA-2** absorb the effect of one-time persistent price shock. The vertical axis expresses the percentage deviation of the winning rate from the post price shock steady state. The horizontal axis expresses sections (2a, 2b) and blocks (2c, 2d).

Since blockchain is a decentralized ledger system, the information about past blocks (timestamps for calculating block time, difficulty for calculating winning rate, seigniorage, total transaction fees in the block, etc.) are publicly disclosed and stored everywhere. We downloaded the full blockchain database of Bitcoin by setting up a full node (i.e., declaring to work as a miner).

The block time $B(t)$ is calculated by subtracting the timestamp of block $t-1$ from the timestamp of block t. Since the timestamps are reported by block creators and not always accurate, it may take a negative value. Among 120,000 blocks (from block 450000 created on December 25th, 2017 to block 569999 created on April 3rd, 2019), we observe that 608 ($\approx 0.51\%$) of the blocks had an earlier timestamp than their parent blocks. We regard them as errors, and perform linear interpolation. $M(t)$ is calculated as the sum of the seigniorage reward (fixed to 12.5 BTC during the observed period) and transaction fees in block t.[11] The winning rate $W(t)$ is computed from the difficulty target of block t.

We merge the blockchain data with the data of USD/BTC exchange rates (the price) at a cryptocurrency exchange, Gemini. The data was downloaded from CryptoDataDownload.[12] We obtained the hourly USD/BTC exchange data, and used the latest available exchange rate before the creation of t-th block as a proxy for $S(t)$.

Figure 3 plots the estimated historical hash rate (according to (5), where we use $T = 144$), in units of Ehash/second (10^{18} attempts per second) and the expected reward ($R(t) := W(t)M(t)S(t)$) in units of USD/Ehash.

The aggregate hash rate had been increasing until October 2018. This is natural because (i) due to technology development, mining facilities had become computationally more efficient, and (ii) miners had accumulated long-term investments since rewards had been high. Recall that our model assumes that the aggregate variable cost function is fixed because our

[11] We observed four outliers in the historical reward. The block reward of block 501726 and block 526591 were smaller than 12.5 BTC. This is allowed by the protocol, but a profit-maximizing miner never chooses this option. Block 469057 and block 470189 had block rewards that were too large (43.22 BTC and 92.16 BTC), due to the transaction fees paid by some (seemingly) irrational users. The block rewards for these blocks are replaced by that of their child blocks.

[12] https://www.cryptodatadownload.com/.

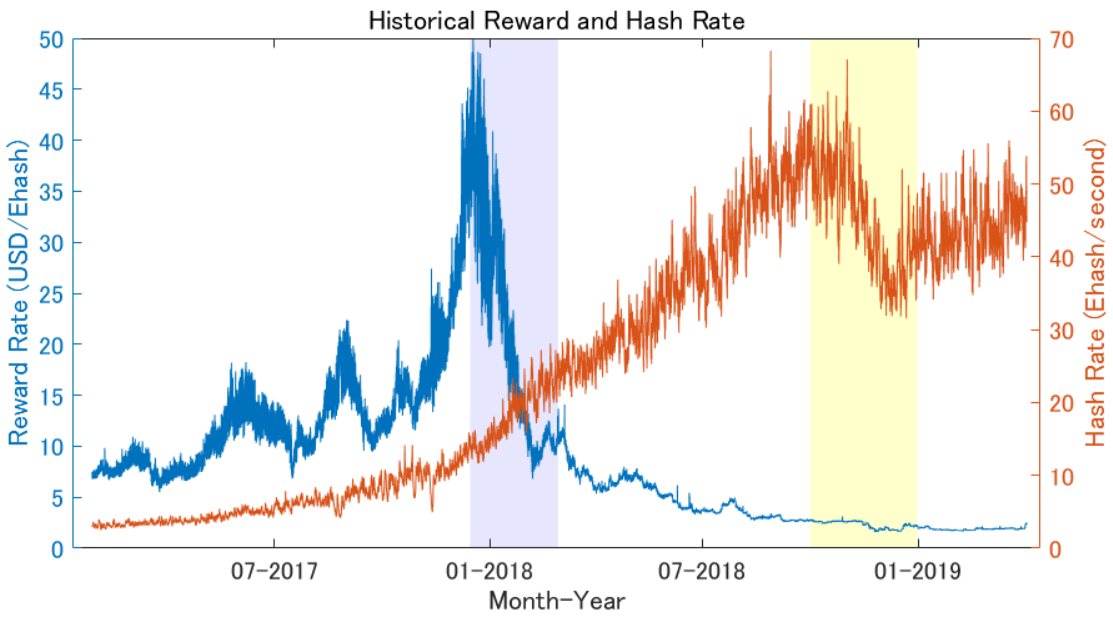


Figure 3: The historical rate of expected reward (USD/Ehash) and estimated hash rate (Ehash/hour). The yellow interval (October 2018 to December 2018 will discussed further in later sections.

aim is to study the short-term behavior of the blockchain economy. When the technology development or the long-term investment improves the power efficiency, the hash supply curve shifts. Consequently, more hash power is provided at the same reward.

We cannot infer from Figure 3 that the hash rate is responsive to the reward. The hash rate did not respond to the largest price drop that occurred from the end of 2017 to the beginning of 2018 (blue area of Figure 3). However, this is also consistent with our theory because miners operate their mining facilities as long as the price is above the shutdown point. As discussed in Subsection 4.2, the reward level after the largest price drop (5 USD/Ehash) is still large enough for most of industrial miners to continue operation.

The hash rate suddenly dropped in November 2018 (yellow area of Figure 3). At that time, the expected reward was at a very low level (about 1.6 USD/Ehash). We suspect that a significant portion of miners idled mining facilities because the reward went below the shutdown point. From this point on, we focus on the data from October 2018 to November 2018 and present supporting evidence.

4.2 Shutdown Points of Mining Facilities

This subsection provides a rough estimate of the shutdown point of each mining facility from its spec. As of late 2018, Bitcoin mining could be profitable only when a miner uses an *application-specific-integrated-circuit* (ASIC) machine that specializes in computing the value of SHA-256 (Taylor 2017). These machines are developed and sold by specialized vendors. Although miners' detailed profit structure is not observable, we can infer the shutdown point from the catalog specs of mining ASIC machines. Mining facilities specialize in computing hash values exclusively and cannot be repurposed. Accordingly, the threshold at which the per hash reward exceeds per hash (marginal) electricity cost should be close to the shutdown point.

We obtained the data of the advertised specs of per unit hash power and power (electricity) usage of mining facilities that specialize in computing SHA-256 from ASIC Miner Value.[13] From these data, we compute power efficiency (the power needed to execute a unit hash attempt). Combining it with the electricity price, we estimate the USD-denominated variable cost to execute a unit attempt. We use this as an estimate of the shutdown point of each individual facility.

Table 1 lists the advertised specs of mining facilities (mining ASIC machines) that were released between March 2018 and September 2018 (a half year right before the focused interval). We can assume that these machines are state-of-the-art in late 2018. Hence, miners who use older and less efficient mining facilities also exist in the mining market.

Although we do not have data about the precise electricity price that each miner faces, we observe that most of the hash supplies are from Chinese mining pools.[14] We employ 0.060 USD/kWh, which is in line with electricity price in China for industrial pool miners.[15]

[13] https://www.asicminervalue.com/

[14] Block creators can leave short comments (up to 100 bytes) in a special field of the block, called *coinbase*. Most miners voluntarily declare their identity here. According to BTC.com, in November 2018, 66.77% of blocks are generated by 8 largest Chinese mining pools (BTC.com, AntPool, ViaBTC, F2Pool, BTC.TOP, Poolin, Huobi.pool, and Dpool). In addition, 9.85% of blocks were created by other small miners, and 13.02% did not disclose their identity. Hence, the share of Chinese miners could be even larger.

[15] In an article of LongHash, Mao Shixing, one of the co-founders of F2Pool, says *"The electricity costs in*

Multiplying the electricity cost by power efficiency, we obtain the marginal cost of each mining facility.

Assuming this electricity price is correct, the estimate of marginal cost gives a lower bound of the shutdown point of the mining facility: there may exist some additional variable cost (e.g., labor cost) and better outside options (e.g., to mine other cryptocurrencies[16]), but a miner never operates the facility when its electricity cost is larger than the reward.[17]

We cannot observe the precise share of each mining facility, however, Table 1 allows us to make an assessment for the shape of the hash supply to the unobserved reward level. It seems reasonable to assume that, as of late 2018, some miners started to suspend their facilities when the expected reward went below 2.5 USD/Ehash, and the hash supply would have declined sharply if the reward had gone below 1.5 USD/Ehash.

4.3 Observed Shutdown in late 2018

Here, we analyze the last quarter of 2018, October 2018 to December 2018. The analysis in Subsection 4.2 suggests that this is low enough for some miners to stop providing hash power.

We assume that the hash supply is linear in reward, i.e.,

$$H(t) = \alpha + \beta R(t),$$

and estimate the intercept and coefficient (α, β) by the maximum likelihood estimation. We obtain the following estimators and their estimated variance-covariance matrix (derivation

China are between 0.26 RMB (.04 USD) and 0.38 RMB (.06 USD) per kilowatt. This is only the cost of electricity itself, not including other mining costs like labor." (LongHash, 2019)

[16]Mining facilities specialize in computing a particular cryptographic hash functions. Accordingly, Bitcoin mining facilities can be diverted only to other cryptocurrencies that use SHA-256 as the mining puzzle.

[17]One may think that a miner may believe that the Bitcoin price recover soon, and then he may want to keep operating mining facilities to acquire BTC. However, if a miner has such a belief and wants to long some BTC, it is cheaper to buy BTC at an exchange.

Table 1: The efficiency of mining ASIC machines released from March 2018 to September 2018. These are the specs of the state-of-the-art mining facilities as of October 2018. The electricity price is assumed to be 0.060 USD/kWh.

Model	Release	Power Efficiency (kWh/Ehash)	Shutdown Point (Marginal Cost) (USD/Ehash)
ASICminer 8 Nano Pro	2018/05	14.6199	0.8772
MicroBT Whatsminer M10S	2018/09	17.6768	1.0606
MicroBT Whatsminer M10	2018/09	18.0556	1.0833
Innosilicon T2 Turbo+ 32T	2018/09	19.0972	1.1458
Innosilicon T2 Turbo	2018/08	22.9167	1.3750
Canaan AvalonMiner 921	2018/09	23.6111	1.4167
Bitfily Snow Panther B1+	2018/08	23.8095	1.4286
Bitfily Snow Panther B1	2018/07	23.9583	1.4375
Aladdin Miner 16Th/s Bitcoin	2018/07	24.3056	1.4583
Innosilicon T2 Terminator	2018/05	25.3553	1.5213
Halong Mining DragonMint T1	2018/04	25.6944	1.5417
Bitmain Antminer S9j (14.5Th)	2018/08	25.8621	1.5517
Bitmain Antminer S9i (14Th)	2018/05	26.1905	1.5714
Canaan AvalonMiner 841	2018/04	26.3480	1.5809
Bitmain Antminer S9 Hydro (18Th)	2018/08	26.6667	1.6000
Bitmain Antminer S9i (13.5Th)	2018/05	26.9547	1.6173
Bitmain Antminer S9i (13Th)	2018/05	27.3504	1.6410
Ebang Ebit E9i	2018/07	29.2181	1.7531
Ebang Ebit E9.2	2018/05	30.5556	1.8333
Ebang Ebit E9.3	2018/05	30.5556	1.8333
MicroBT Whatsminer M3X	2018/03	45.5556	2.7333
Bitmain Antminer V9 (4Th)	2018/03	71.3194	4.2792

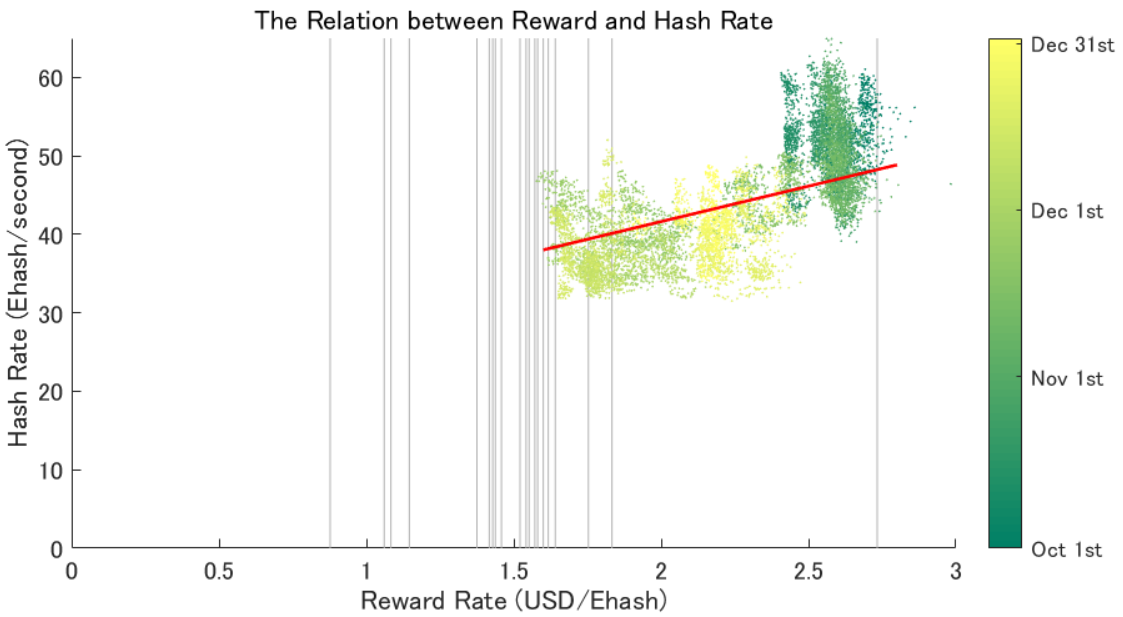


Figure 4: The estimated hash supply for October 2018 to December 2018. Pointwise estimates are scattered as green dots, and the linear fit ($\hat{H} = 23.60 + 9.02R$) is drawn as a red line. The shutdown points (or marginal costs) of mining facilities (shown in Table 1) is drawn as gray vertical lines.

is described in Appendix C):

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} 23.60 \\ 9.02 \end{pmatrix}, \quad \widehat{\mathrm{Var}} \begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \begin{pmatrix} 6.40 & -2.78 \\ -2.78 & 1.24 \end{pmatrix}.$$

Here, the hash rate $H(t)$ is expressed in Ehash/second, and the reward rate $R(t)$ is expressed in USD/Ehash.

Figure 4 visualizes the estimated hash supply. The reward is assigned to the horizontal axis, and hash rate is assigned to the vertical axis. Pointwise estimates (a reciprocal of 144-period moving average of $W(t)B(t)$) are scattered as green dots. The linear fit, $\hat{\alpha} + \hat{\beta}R(t)$, is plotted as the red line. The gray vertical lines represent the shutdown points of mining facilities (shown in Table 1).

As we expected, the estimated hash supply is responsive to the change in reward, when the reward is sufficiently low. However, the observed reward rate was not low enough to shut down most of the state-of-the-art mining facilities at that time (the minimum reward

in this quarter was 1.58 USD/Ehash). Although it cannot be estimated from the empirical data, we expect that if the reward had gone down further and below 1.5 USD/Ehash, then the hash rate should have dropped more drastically. Furthermore, since the Bitcoin price is volatile, this was a realistic scenario.

5 Simulation

5.1 Setup

This section evaluates the performance of DAAs by simulation. We choose the parameters to mimic the environment from October 2018 to December 2018, and compare paths generated along with **DAA-1** and **DAA-2**. Our aim is to evaluate the extent to which (counterfactual) adoption of **DAA-2** could have improved the stability of block generation.

As we discussed in Section 4, we only have limited information about the hash supply function. Based on the observations in Section 4, we attempt to choose a function that satisfies the following three criteria:

1. The supremum of the hash supply function is close to the historical maximum (about 55 Ehash/second) at that time.

2. The hash supply function matches data when the reward is in the historically observed range (1.58 USD/Ehash or larger).

3. To capture shutdown points estimated in Subsection 4.2, the slope of the hash supply function should become steep when the reward is around 1.5 USD/Ehash.

As a function that satisfies these criteria, we use a (scaled) sigmoid function:

$$\Phi(R) = \frac{55}{1 + \mathrm{e}^{-3\cdot(R-1.5)}} \quad \text{(Ehash/second)}$$

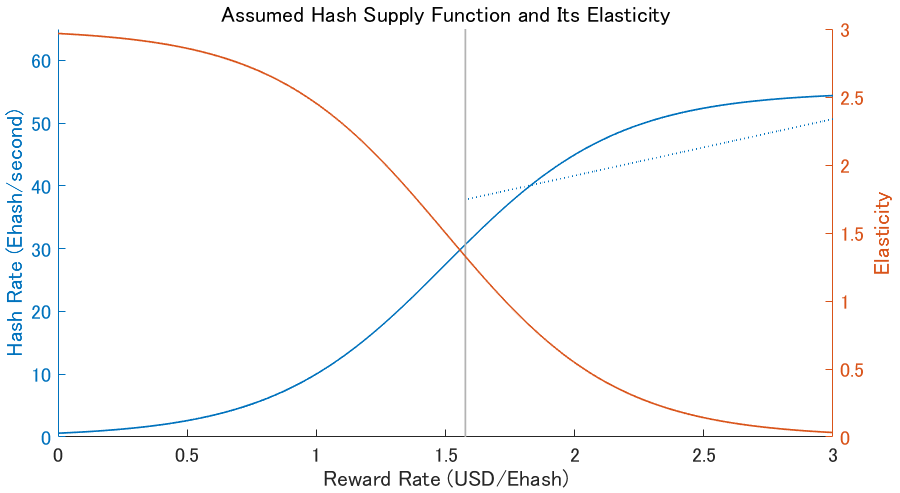


Figure 5: The shape of the hash supply function assumed in Section 5 (blue solid line), and its elasticity (red solid line). The gray horizontal line represents the minimum observed reward (1.58 USD/Ehash). The blue dotted line is a linear fit derived in Subsection 4.3.

The shape of our hash supply function and its elasticity is depicted in Figure 5.[18]

We calculate the historical drift rate (return) and volatility from the hourly Bitcoin price data of the focused quarter. The historical drift rate was -0.55%/day. To provide a conservative prediction that does not rely on the negativity of the drift rate, we instead set the drift rate to zero: $\mu = 0$. The historical volatility was 4.24%/day. We use this value as the (true) volatility of our model.

During this quarter, transaction fees had been low: 94% of generated blocks had a total transaction fee smaller than 0.5 BTC (4% of the seigniorage reward). We ignore transaction fees and fix $M(t)$ to 12.5 BTC (which is equal to the per block seigniorage reward provided by the system) throughout the simulation.

We use the history until block 551442 (generated on November 25th, 2018) as the initial state of the simulation. At this block, the reward attained the minimum level of this quarter (1.58 USD/Ehash). We regard block 551442 as period 0, and initiate the simulation from

[18]The elasticity is given by

$$\varepsilon(R) := \frac{R \cdot \Phi'(R)}{\Phi(R)} = \frac{3 \cdot \mathrm{e}^{-3\cdot(R-1.5)}}{1 + \mathrm{e}^{-3\cdot(R-1.5)}}.$$

period 1. The initial price $S(0)$ is 3604.04 USD/BTC, and the initial winning rate $W(0)$ is 0.000035/Ehash. Accordingly, the initial reward $R(0)$ is 1.58 USD/Ehash.

5.2 A Sample Path

We have two different factors of randomness: price shock ($\epsilon(t)$, which determines the price $S(t)$) and block shock ($\delta(t)$, which determines the block time $B(t)$ together with the arrival rate $W(t)H(t)$). Figure 6 shows a sample path of $S(t)$ and $\delta(t)$. In order to articulate the performance of different DAAs, for each simulation, we fix the sequence of price shocks and block shocks and compare the path of $(W(t), H(t), B(t))$ along with the same realization of shocks.

The red lines of Figure 7 depicts the behavior of the winning rate $W(t)$, reward $R(t)$, hash rate $H(t) = \Phi(R(t))$, and block time $B(t) = \delta(t)/(W(t)H(t))$, along with the shock depicted in Figure 6. The left column shows the behavior of the economy under **DAA-1**(2016) (close to the Bitcoin DAA), and the right column shows the behavior under **DAA-2**(144) (close to the Bitcoin Cash DAA). The horizontal axis represents the real time, rather than block numbers. However, graph areas are colored differently for every 2016 blocks (= one section in **DAA-1**(2016)); thus, we can also infer block numbers. The simulation was run for 84 days (= 12 weeks). If blocks are generated every 10 minutes, then 12096 blocks (= 6 sections in **DAA-1**(2016)) are produced in total.

Given the hash supply function Φ, the seigniorage reward $M(t)$, and the price $S(t)$, we can also compute the first-best winning rate $W^*(t)$ by (numerically) solving the following equation:

$$\frac{1}{B^*} = W^*(t)\Phi(W^*(t)M(t)S(t)).$$

If the winning rate is set to $W^*(t)$ in block t, the block arrival rate becomes exactly equal to $1/B^*$. The behavior of the first-best winning rate $W^*(t)$, first-best reward $R^*(t) := W^*(t)MS(t)$, first-best hash rate $H^*(t) := \Phi(R^*(t))$, and block time under the first-best

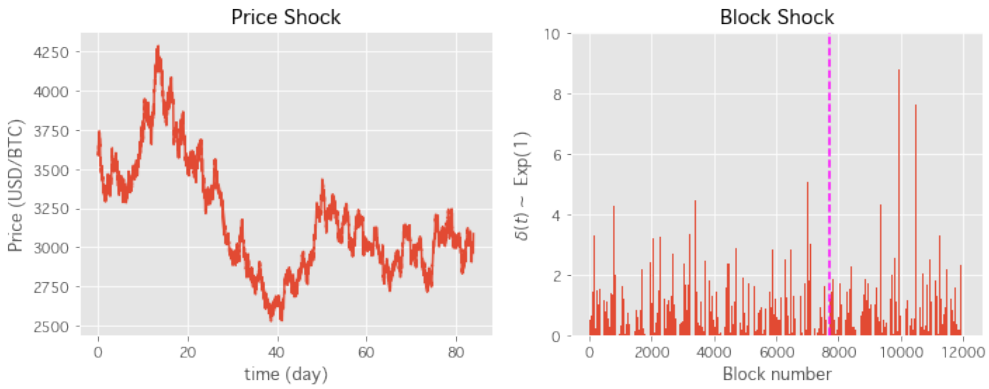


Figure 6: A sample path of prices ($S(t)$) and block shocks ($\delta(t)$). This realization is used for the sample path shown in Figure 7. The purple vertical line of the block shock graph indicates the last block generated by **DAA-1**(2016) ($t = 7704$).

Figure 7: A sample path of the simulation result. The left column shows **DAA-1**(2016) (close to the Bitcoin DAA), and the right column shows **DAA-2**(144) (close to the Bitcoin Cash DAA). The blue line depicts the first-best. Shocks are depicted in Figure 6. Graph areas are colored differently for every 2016 blocks.

Table 2: Summary statistics of block times generated in simulations. We generate 5000, 84-day paths to compute the statistics. DAAs are evaluated with the identical realizations of the price shocks and block shocks. Over X measures the fraction of paths that experience at least one block time larger than X minutes.

	DAA-1(2016)	**DAA-1**(2016) w/ bound	**DAA-2**(144)	**DAA-2**(144) w/ bound
mean (min.)	14.0443	14.0427	11.2424	11.2424
std (min.)	18.1396	18.1130	10.9201	10.9201
over 120 min.	0.8504	0.8504	0.1838	0.1838
over 180 min.	0.5337	0.5337	0.0024	0.0024

arrival rate $B^*(t) = \delta(t)B^*$ are drawn as blue lines in Figure 7.

Clearly, **DAA-1**(2016) fails to stabilize block generation. As predicted in Section 3, **DAA-1**(2016) does not make the winning rate closer to the first-best level. Instead, the winning rate alternates between too high and too low. On average, block generation is too slow: at the end of the 84th day, **DAA-1**(2016) created only 7704 blocks, which is 63.7% of the policy aim (12096 blocks). What is worse, this situation does not tend to get better even after the period of observation.

In contrast, **DAA-2**(144) successfully generates new blocks at a constant rate. The winning rate slightly oscillates, but roughly traces the first-best level. **DAA-2**(144) steadily generates 2016 blocks every 14 days, and therefore, light-gray and yellow areas come at even intervals. At the end of the 84th day, it created 12049 blocks, which is 99.6% of the ideal amount. Generated block times are also close to the one generated by the first-best difficulty adjustment.

5.3 Statistics of Block Times

Table 2 shows the statistics of block times generated by simulations. We generate 5000 different 84-day paths to compute the statistics. DAAs are evaluated with identical realizations

of the price shocks and block shocks.

As mentioned in Remark 4, Bitcoin and Bitcoin Cash incorporate the rules to prevent the change of the winning rate from being too abrupt. Specifically, in the Bitcoin system, if the updated winning rate $w(n+1)$ specified by (4) is larger than $4w(n)$ (or smaller than $0.25w(n)$, then $w(n+1) = 4w(n)$ ($w(n+1) = 0.25w(n)$) is used instead. Similarly, in the Bitcoin Cash system, if the updated winning rate $W(t+1)$ specified by (6) is larger than $2W(t)$ (or smaller than $0.5W(t)$), then $W(t+1) = 2W(t)$ ($W(t+1) = 0.5W(t)$) is used instead. These DAAs are shown as "**DAA-1**(2016) w/ bound" and "**DAA-2**(144) w/ bound" in Table 2. We find that the bound does not have a large influence on statistics in our 84-day simulations.

Compared with **DAA-2**(144), **DAA-1**(2016) has a larger mean block time and mean standard deviation. Furthermore, **DAA-1**(2016) experiences extremely long block times (over 120 minutes and 180 minutes) more frequently. They indicate the oscillation of the winning rate. This is consistent with our observation in the previous subsection.

To summarize, consistent with our theoretical analysis, **DAA-1** fails to adjust the winning rate to an appropriate level. Consequently, blocks are not generated in even intervals. In contrast, **DAA-2** roughly traces the first-best winning rate, and therefore, stably generates new blocks.

5.4 Miners' Profit and Its Implication to the Policy Adoption

Our theoretical and simulation results indicate that **DAA-2** outperforms **DAA-1** in the sense that **DAA-2** stabilizes the block arrival rate more efficiently. To improve users' convenience and decrease the risk of synchronization failures, the Bitcoin system should adopt **DAA-2** (or one of other alternative DAAs introduced in Section 6). In this subsection, we discuss the practical possibility that the Bitcoin community adopts the policy change.

In the Bitcoin system (and any PoW-based blockchain systems), miners intrinsically have the right to select a consensus rule. If miners do not agree with the proposed policy change,

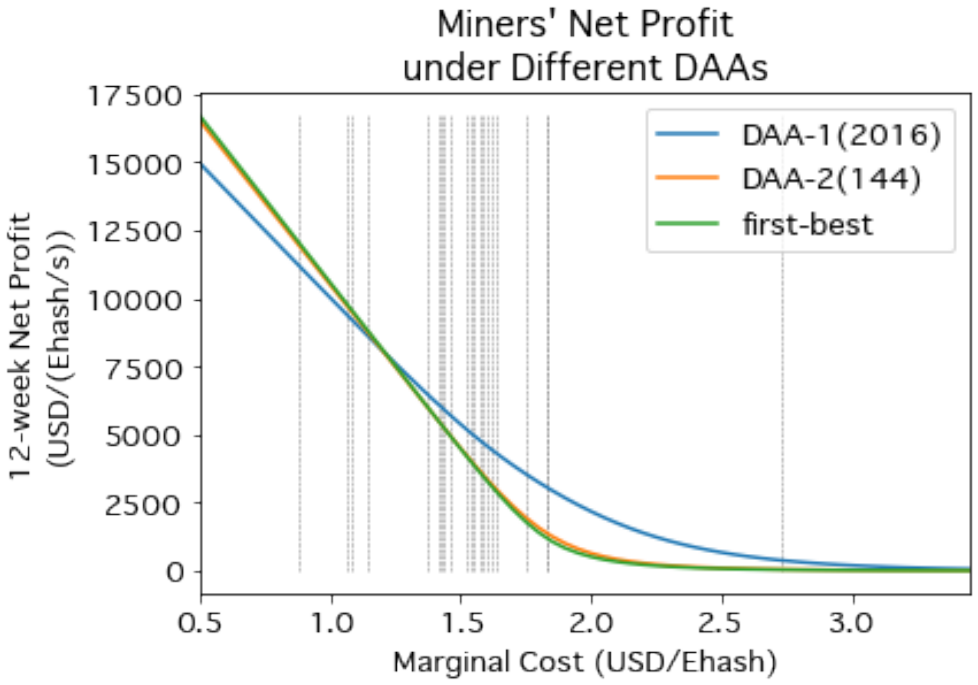


Figure 8: The relationship between miners' net profit and the marginal (per hash) cost of mining under different DAAs. The gray vertical lines represent the estimated marginal costs of mining facilities (shown in Table 1).

they can continue using the old rule by extending a blockchain that is invalid with respect to the new rule. This fact was stated in the original Bitcoin white paper.[19] Hence, any proposal is adopted only when miners agree. However, our simulation result indicates that miners may not want to switch to **DAA-2**, as it is less profitable for a large fraction of miners.

We created 5000 different 84-day sample paths and calculated the return from owning one unit of mining facility with various marginal cost. Figure 8 illustrates the relationship between the marginal cost and net profit (12-week total rewards - total electricity costs). The gray vertical lines represent the estimated marginal costs of mining facilities (see Table 1).

Profits are decreasing in the marginal cost, but the profit curve is nonlinear. This is because mining facilities have some option value. When the reward is below the marginal cost, miners can turn off their facilities. Accordingly, even if the reward goes to a very low

[19] *"They [Miners] vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them."* (Nakamoto 2008, p.8)

level, miners will not incur a loss (if we ignore the fixed cost).[20]

DAA-1 favors miners whose marginal costs are large. This is because the value of an option becomes larger when the price of underlying assets become more volatile. Under **DAA-1**, the winning rate oscillates, and sections that are either too easy or too difficult arrive alternately. Inefficient miners prefer this feature because they can earn a positive profit by operating only in easy sections. In contrast, under **DAA-2**, the winning rate does not oscillate by much. Accordingly, the mining puzzle is consistently too difficult for inefficient miners, and they cannot earn a positive profit. This is the reason why the profit curve of **DAA-1** is flatter than that of **DAA-2** and the first-best difficulty adjustment (choose $W^*(t)$ every period).

Note that a high marginal cost does not always mean the miner uses outdated mining facilities. Our analysis does not take account of the fixed cost of purchasing mining facilities. Technology development not only improves power efficiency but also decreases the fixed cost for owning a unit hash power. Because of this, the most profitable mining facility may not be the one with the lowest marginal cost, and we cannot conclude that mining facilities with higher marginal costs will be weeded out due to competition.

With our simulation setup, a large fraction of the state-of-the-art mining facilities are categorized as "inefficient facilities" in the sense that they can yield a larger profit from **DAA-1**. If the Bitcoin system adopts **DAA-2**, the value of these mining facilities is depreciated. For this reason, the Bitcoin community may be reluctant to upgrade of the DAA. In this sense, the Bitcoin system has a fundamental incentive conflict between users and miners.

[20] Hashimoto and Noda (2019) evaluate mining facilities as a financial asset and find that the profit from mining facilities can be replicated by a portfolio of call options.

6 Other Difficulty Adjustment Algorithms

6.1 DAA-3 (Abandoned Proposal of Bitcoin Cash)

When Bitcoin Cash upgraded its DAA in November 2017, in addition to the accepted algorithm (**DAA-2** with $T = 144$, coded by Séchet, 2017), two other powerful candidates, **DAA-3** and **DAA-4**, were proposed by Booth (2017) and Harding (2017), respectively. According to Bitcoin ABC (2018), while their constructions are different, these three algorithms produced similar results in testing.

DAA-3 keeps monitoring the past block time and immediately fixes the winning rate when an "emergency" (i.e., block time becomes far from the targeted level) is observed.

DAA-3 is parameterized by the following five parameters: (i) increment u, (ii) decrement d, (iii) lower threshold $\underline{\theta}$, (iv) upper threshold $\overline{\theta}$, and (v) sample size T. If the sample mean of block time is below $\underline{\theta}$, it indicates that the current mining puzzle is too easy (the winning rate is too high). Then, **DAA-3** decrements the winning rate by d percent. Conversely, if the sample mean is above $\overline{\theta}$, **DAA-3** increments the winning rate by u percent. To make block time $B(t)$ closer to the targeted level B^*, the thresholds should surround the target: $B^* \in (\underline{\theta}, \overline{\theta})$ should be satisfied. Furthermore, to achieve a gradual adjustment, the designer intended to select small u and d.

$$W(t+1) = \begin{cases} (1-d) \cdot W(t) & \text{if } \dfrac{1}{T} \displaystyle\sum_{s=t-T+1}^{t} B(s) < \underline{\theta} \\ (1+u) \cdot W(t) & \text{if } \dfrac{1}{T} \displaystyle\sum_{s=t-T+1}^{t} B(s) > \overline{\theta} \\ W(t) & \text{otherwise} \end{cases}$$

We assess that **DAA-3** is robust even if the hash supply function is elastic. This is because **DAA-3** does not overshoot the winning rate, in contrast to **DAA-1** (and **DAA-2**). Figure 9 illustrates the difficulty adjustment by **DAA-3**. Imagine that the current winning

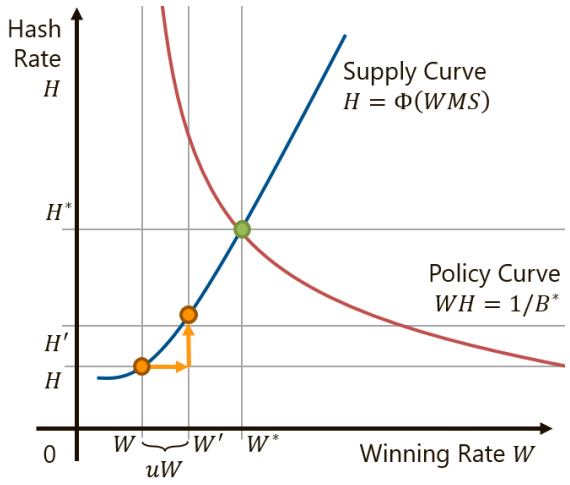


Figure 9: Difficulty adjustment by **DAA-3**. Since the increment of the winning rate is fixed and small ($W'H = (1+u)WH$), an adjustment does not overshoot the policy target, in contrast to **DAA-1** (see also Figure 1).

rate W is too low ($WH < 1/B^*$). **DAA-1** makes the winning rate W jump up to W' such that $W'H = 1/B^*$. As illustrated in Figure 1, this adjustment always overshoots the policy target W^* in that $W'H' > W'H = W^*\Phi(W^*MS) = 1/B^*$. By contrast, **DAA-3** only increases the winning rate by $u \cdot W$, which is typically small compared with the distance from the steady state, $|W^* - W|$. Under W', more hash power is supplied because the expected reward is also increased ($H' > H$). However, the updated winning rate W' is typically still below the ideal level W^* because the increments of the winning rate (u and d) are small. Accordingly, the oscillation and divergence we observed for **DAA-1** (and **DAA-2**, when the elasticity of the hash supply is extreme) cannot be caused by **DAA-3**, as long as sufficiently small u and d are chosen. The degree of smallness required for the increment depends on the elasticity of the hash supply.

6.2 Ethereum

Ethereum is the second largest cryptocurrency (in market capitalization) after Bitcoin. Although the protocol of Ethereum is significantly different from Bitcoin, it is also PoW-based

as of June 2019.[21] The structure of the current Ethereum DAA is similar to **DAA-3**. The target block time of Ethereum is between 10 to 19 seconds. The Ethereum system decreases the winning rate by $1/2048 \approx 0.05\%$ when the block time is less than 10 seconds. If the block time is more than 10 seconds, it increases the winning rate by (block time - 10 seconds)/2048, while the upper bound is set to 99/2048. Similar to **DAA-3**, the DAA of Ethereum is stable even if the hash supply is elastic.

6.3 DAA-4 (Abandoned Proposal of Bitcoin Cash)

DAA-4, proposed by Harding (2017), estimates the hash rate by taking the moving average of $W(t)B(t)$ every period. In addition, unlike any other DAAs discussed in this paper, **DAA-4** takes a weighted moving average that puts heavier weight on recent blocks.

Mathematically, **DAA-4** is expressed as the following recurrence relation:

$$W(t+1) = \frac{2 \cdot \sum_{s=t-T+1}^{t} (s-t+T) W(s) B(s)}{T(T+1)B^*}. \tag{22}$$

Recall that $\sum_{s=t-T+1}^{t}(s-t+T) = T(T+1)/2$. Accordingly,

$$\frac{2 \cdot \sum_{s=t-T+1}^{t} (s-t+T) W(s) B(s)}{T(T+1)}$$

is a weighted average of $W(t)B(t)$. Hence, it is a consistent estimator of $1/H(t)$, assuming that the hash rate has been constant for previous T periods. Hence, **DAA-4** is also a sample analog of (3).

As we did in Section 3, let us ignore the randomness caused by the price shock ϵ and block shock δ. Linearizing (22), we have

$$\tilde{W}(t+1) = \frac{2}{T(T+1)} \sum_{s=t-T+1}^{t} (s-t+T) \left(\tilde{W}(s) + \tilde{B}(s) \right). \tag{23}$$

[21] Ethereum is planned to replace the consensus mechanism to the Proof-of-Stake system by the end of 2021.

	DAA-2(144) (revisited)	**DAA-4**(144)	**DAA-2'**(144)
mean (min.)	11.2424	11.2906	11.2261
std (min.)	10.9201	11.0013	10.9018
over 120 min.	0.1838	0.2122	0.1790
over 180 min.	0.0024	0.0022	0.0020

Table 3: Summary statistics of block times generated in simulations. The way to generate statistics the same as that of Table 2. **DAA-2**(144), **DAA-4**(144), and **DAA-2'**(144) exhibit a very similar performance.

The linearized economy under **DAA-4** is represented by (18), (19), and (23). Combining the above equations, we have

$$\tilde{W}(t+1) = -\frac{2\varepsilon}{T(T+1)} \sum_{s=t-T+1}^{t} (s-t+T)\tilde{W}(s).$$

Similar to **DAA-2**, with a large T, **DAA-4**(T) is stable.

Theorem 3. ***DAA-4***(T) *is stable if* $\varepsilon < (T+1)/2$.

Proof. See Appendix A. □

Unlike Theorems 1 and 2, Theorem 3 proposes a loose sufficient condition. Indeed, $\varepsilon < (T+1)/2$ is not necessary. For example, when $T = 2$, (23) reduces to

$$\tilde{W}(t+1) = -\frac{2\varepsilon}{3}\tilde{W}(t) - \frac{\varepsilon}{3}\tilde{W}(t-1),$$

and this recurrence relation is stable if and only if $\varepsilon < 3$, while $(T+1)/2 = 3/2$. Although we do not provide a tight necessary and sufficient condition in this study, we believe that $\varepsilon < (T+1)/2$ is not a practically binding condition with a large enough T (e.g., $T = 144$).

Table 3 shows the summary statistics of block times generated by **DAA-4**(144). The performance of **DAA-4**(144) in this simulation is quite similar to **DAA-2**(144), and much better than **DAA-1**(2016).

It should be noted that, compared with **DAA-1** and **DAA-2**, **DAA-4** needs more detailed information about timestamps. Recall that block times are calculated by taking differences of timestamps. As for **DAA-1**(T) and **DAA-2**(T), to calculate $W(t+1)$, we only need the information about "the total time spent on creating previous T blocks" (i.e., $\sum_{t-T+1}^{t} B(s)$). To compute it, we only need to know the timestamp of block t and $t-T$. However, since **DAA-4**(T) computes the weighted average of $B(s)$'s, to calculate $W(t+1)$, we need the information about $B(s)$ for each $s = T-t+1, \ldots, t$. They depend on all the past $T+1$ timestamps. As we mentioned in Remark 2, timestamps are not always accurate and can possibly be manipulated by block creators. In this sense, **DAA-4** is less robust.

6.4 DAA-2'

Many DAAs reduce to an identical linear recurrence relation once we log-linearize the economy. For example, if we get rid of the recency weight from **DAA-4**, its local behavior becomes equivalent to **DAA-2**.

The following DAA is also locally identical to **DAA-2**.

$$W(t+1) = \frac{\sum_{s=t-T+1}^{t} W(s)}{T} \cdot \frac{\sum_{s=t-T+1}^{t} B(s)}{T \cdot B^*}. \tag{24}$$

We define the DAA characterized by (24) as **DAA-2'**(T). Since its local behavior is identical to **DAA-2**(T), **DAA-2'**(T) is stable if and only if $\varepsilon < T$.

While **DAA-2** and **DAA-2'** have the same recurrence relation in the linearized economy (21), the original adjustment rules ((6) and (24)) are different. However, their performances in our simulation are surprisingly similar (Table 3) although **DAA-2'** marginally outperforms **DAA-2** in all of the four statistics. This result supports the validity of our theoretical analysis focusing on the local behavior of DAAs.

DAA-2' has a simpler (and more natural) form than **DAA-2** and records a slightly better simulation result. Furthermore, similar to **DAA-1**(T) and **DAA-2**(T), this rule only

depends on the total block times of previous T periods. Accordingly, **DAA-2'** could also be a good candidate when the Bitcoin system considers upgrading its own DAA.

7 Related Literature

Previous literature has analyzed the long-term growth of the difficulty level. Bowden, Keeler, Krzesinski, and Taylor (2018) analyzes how the difficulty level evolves over time. Kraft (2016) models mining as a Poisson process with time-dependent intensity, and proposes an alternative DAA that performs better than the current Bitcoin DAA in such an environment. Our focus is different from theirs in that we fully take account of short-term entry/exit of miners.

The design of the Bitcoin system has also attracted market/mechanism designers' interest. Previous works have suggested a number of alternative mechanisms for transaction fees (Chiu and Koeppl, 2017; Basu, Easley, O'Hara, and Sirer, 2019; Easley, O'Hara, and Basu, 2019; Huberman et al., 2019a; 2019b). In contrast, this study articulates the importance of the design of the DAA. To the best of our knowledge, no previous work has developed an economic model that takes account of endogenous shutdown to study the design of DAA.

The long-term sustainability of the Bitcoin system has also been considered (Abadi and Brunnermeier, 2018; Budish, 2018, Chiu and Koeppl, 2019). Even after miners optimally choose their long-term investment decision (i.e., purchase mining facilities), they can still choose the quantity of supply (i.e., run or idle their mining facilities). Accordingly, our problem does not disappear even in a long run.

A number of previous works have studied Bitcoin stability. They have considered whether an attacker can profitably take advantage of the system (e.g., Nakamoto, 2008; Rosenfeld, 2009; 2011; Decker and Wattenhofer, 2013). These works assume that the hash supply is constant. As this study shows, the hash supply may respond to the reward, and this feature may become an additional instability factor. Note also that stabilization of block

generation is important for user's convenience and the sustainability of the system, even if attackers were absent.

8 Concluding Remarks

We proved that the current Bitcoin DAA (**DAA-1**) is unstable if the hash supply is elastic. It does not take into account endogenous entry/exit of miners, and therefore, the winning rate and the block arrival rate may oscillate and diverge. We have not historically observed such an outcome just because the cryptocurrency price stayed high and the reward has not dropped to a critical level. However, there is no guarantee that we will never experience the oscillation and divergence of the block time in the future. The history of cryptocurrency is too short, and the situation surrounding cryptocurrencies is changing dramatically. Hence, we cannot obtain a concrete prediction by pure induction. Furthermore, because anyone can work as a miner, it is expected that miners will accumulate long-term investments until mining becomes unprofitable. Even if the Bitcoin price stays high, the long-term investments eventually saturate, and then, a small change in the Bitcoin price will significantly affect miners' operation decision. In this scenario, we would face a crisis similar to that nearly occurred in November 2018.

We also proved that, in contrast to **DAA-1**, **DAA-2** stabilizes the block time even when the hash supply is highly elastic. Furthermore, **DAA-2** is already implemented in another blockchain platform, Bitcoin Cash, and seems to have empirically performed well. To summarize, the Bitcoin system needs an upgrade in its DAA to prevent the future crisis. Although there may exist an even better algorithm, the current DAA of Bitcoin Cash seems a satisfactory solution.

Appendix

A Proofs

A.1 Proof of Theorem 2

Define

$$\begin{aligned} f(\lambda) &\coloneqq (\lambda - 1) \cdot \phi(\lambda) \\ &= \lambda^{T+1} - \left(1 - \frac{\varepsilon}{T}\right) \lambda^T - \frac{\varepsilon}{T}. \end{aligned} \tag{25}$$

Sufficiency Suppose that $\varepsilon < T$. Then, since $(1 - \varepsilon/T) > 0$, for $\lambda \in \mathbb{R}_{>0}$ the first term of (25) is increasing and the second and third terms are decreasing. Accordingly, $f(\lambda) = 0$ has at most one solution in $\mathbb{R}_{>0}$, and if such a solution λ^* exists, we have $f(\lambda) > 0$ for all $\lambda > \lambda^*$. Indeed, by construction of f, we have $f(1) = 0$. Accordingly, $f(\lambda) > 0$ for all (real) $\lambda > 1$.

For all (possibly non-real) λ, we have

$$\begin{aligned} |f(\lambda)| &= \left|\lambda^{T+1} - \left(1 - \frac{\varepsilon}{T}\right) \lambda^T - \frac{\varepsilon}{T}\right| && (26) \\ &\geq \left|\lambda^{T+1}\right| - \left|\left(1 - \frac{\varepsilon}{T}\right) \lambda^T + \frac{\varepsilon}{T}\right| && (27) \\ &\geq \left|\lambda^{T+1}\right| - \left|\left(1 - \frac{\varepsilon}{T}\right) \lambda^T\right| - \left|\frac{\varepsilon}{T}\right| && (28) \\ &= \left|\lambda^{T+1}\right| - \left(1 - \frac{\varepsilon}{T}\right) \left|\lambda^T\right| - \frac{\varepsilon}{T} \\ &= f(|\lambda|). \end{aligned}$$

We already showed that whenever $|\lambda| > 1$, we have $f(|\lambda|) > 0$. Accordingly, if $|\lambda| > 1$, then λ cannot be a solution of $f(\lambda) = 0$.

Consider a solution λ such that $|\lambda| = 1$. Such a solution must satisfy both triangular

inequalities, "(26) $\geq$ (27)" and "(27) $\geq$ (28)," with equalities. This happens if and only if there exist non-negative real numbers η_1, η_2 such that

$$\lambda^{T+1} - \left(1 - \frac{\varepsilon}{T}\right)\lambda^T - \frac{\varepsilon}{T} = \eta_1 \cdot \left[\left(1 - \frac{\varepsilon}{T}\right)\lambda^T + \frac{\varepsilon}{T}\right], \tag{29}$$

$$\left(1 - \frac{\varepsilon}{T}\right)\lambda^T = \eta_2 \cdot \frac{\varepsilon}{T}. \tag{30}$$

Clearly, $\eta_2 = 0$ cannot hold when λ is a solution. Combining (29) and (30), we have

$$\lambda = \frac{1}{\eta_2}(\eta_1 + 1)(\eta_2 + 1)\left(1 - \frac{\varepsilon}{T}\right).$$

Since the right hand side of the above equation is a positive real number, λ must also be a positive real number. Thus, $\lambda = 1$ must be the case.

Hence, $f(\lambda) = 0$ does not have a solution λ such that $\lambda \neq 1$ and $|\lambda| \geq 1$. Furthermore, since $\phi(1) > 0$, $\lambda = 1$ is not a solution of $\phi(\lambda) = 0$. Therefore, the characteristic equation does not have any root that is larger than one in absolute value. Accordingly, $\tilde{W}(t) \to 0$ as $t \to \infty$.

Necessity Suppose that $\varepsilon = T$. Then, $f(\lambda) = \lambda^{T+1} - 1$. The solution of $f(\lambda) = 0$ is

$$\lambda = \cos\left(\frac{2\pi k}{T+1}\right) + i \sin\left(\frac{2\pi k}{T+1}\right) \text{ for } k = 0, 1, 2, \ldots, T.$$

When $k = 0$, we have $\lambda = 1$, and it is the solution of $x - 1 = 0$. Accordingly, the rest of the solutions ($k = 1, 2, \ldots, T$) are the solutions of $\phi(\lambda) = 0$. Since all of these solutions satisfy $|\lambda| = 1$, $\tilde{W}(t)$ does not converge.

Suppose that $\varepsilon > T$. Then, we have

$$\left|\tilde{W}(t+1)\right| = \left|-\frac{\varepsilon}{T}\sum_{s=t-T+1}^{t}\tilde{W}(s)\right| = \frac{\varepsilon}{T}\left|\sum_{s=t-T+1}^{t}\tilde{W}(s)\right| > \left|\sum_{s=t-T+1}^{t}\tilde{W}(s)\right|.$$

Recall that $\tilde{W}(t)$ does not converge to zero when $\varepsilon/T = 1$. Accordingly, it does not converge when $\varepsilon/T > 1$ either.

A.2 Proof of Theorem 3

Suppose that $\varepsilon < (T+1)/2$. $\tilde{W}(t)$ converges to zero if and only if all the roots of the following characteristic equation are smaller than one in absolute value.

$$\phi(\lambda) \coloneqq \lambda^T + \frac{2\varepsilon}{T(T+1)} \sum_{t=0}^{T-1} (t+1)\lambda^t = 0.$$

Define

$$\begin{aligned} f(\lambda) &\coloneqq (\lambda - 1) \cdot \phi(\lambda) \\ &= \lambda^{T+1} - \left(1 - \frac{2\varepsilon}{T+1}\right) \lambda^T - \frac{2\varepsilon}{T(T+1)} \sum_{t=0}^{T-1} \lambda^t. \end{aligned} \tag{31}$$

By assumption, $1 - 2\varepsilon/(T+1) > 0$. Accordingly, for $\lambda \in \mathbb{R}_{>0}$, the first term of (31) is increasing and all the other terms are decreasing. Accordingly, $f(\lambda) = 0$ has at most one solution in $\mathbb{R}_{>0}$, and if such a solution λ^* exists, we have $f(\lambda) > 0$ for all $\lambda > \lambda^*$. By construction, we have $f(1) = 0$. Hence, $f(\lambda) > 0$ for all (real) $\lambda > 1$.

For all (possibly non-real) λ, we have

$$
\begin{aligned}
|f(\lambda)| &= \left|\lambda^{T+1} - \left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T - \frac{2\varepsilon}{T(T+1)}\sum_{t=0}^{T-1}\lambda^t\right| && (32)\\
&\geq \left|\lambda^{T+1}\right| - \left|\left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T + \frac{2\varepsilon}{T(T+1)}\sum_{t=0}^{T-1}\lambda^t\right| && (33)\\
&\geq \left|\lambda^{T+1}\right| - \left|\left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T\right| - \sum_{t=0}^{T-1}\left|\frac{2\varepsilon}{T(T+1)}\lambda^t\right| && (34)\\
&= \left|\lambda^{T+1}\right| - \left(1 - \frac{2\varepsilon}{T+1}\right)\left|\lambda^T\right| - \frac{2\varepsilon}{T(T+1)}\sum_{t=1}^{T-1}\left|\lambda^t\right|\\
&= f(|\lambda|).
\end{aligned}
$$

We already showed that whenever $|\lambda| > 1$, we have $f(|\lambda|) > 0$. Accordingly, if $|\lambda| > 1$, then λ cannot be a solution of $f(\lambda) = 0$.

Consider a solution λ such that $|\lambda| = 1$. Such a solution must satisfy both triangular inequalities, "(32) $\geq$ (33)" and "(33) $\geq$ (34)," with equalities. This happens if and only if there exist non-negative real numbers $\eta_1, \eta_2, \ldots, \eta_{T+1}$ such that

$$
\begin{aligned}
&\lambda^{T+1} - \left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T - \frac{2\varepsilon}{T(T+1)}\sum_{t=0}^{T-1}\lambda^t\\
=&\eta_{T+1}\left[\left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T + \frac{2\varepsilon}{T(T+1)}\sum_{t=0}^{T-1}\lambda^t\right],
\end{aligned}
$$

$$
\left(1 - \frac{2\varepsilon}{T+1}\right)\lambda^T = \frac{2\eta_T\varepsilon}{T(T+1)}\sum_{t=0}^{T-1}\lambda^t,
$$

and

$$
\frac{2\varepsilon}{T(T+1)}\sum_{t=0}^{k}\lambda^t = \frac{2\eta_k\varepsilon}{T(T+1)}\sum_{t=0}^{k-1}\lambda^t
$$

for $k = 1, \ldots, T-1$. These conditions are satisfied only when all of the arguments of $1, \lambda, \lambda^2, \ldots, \lambda^{T+1}$ are aligned. This is possible only if λ is a positive real number. Thus, $\lambda = 1$ must be the case.

Hence, $f(\lambda) = 0$ does not have a solution λ such that $\lambda \neq 1$ and $|\lambda| > 1$. Furthermore, since $\phi(1) > 0$, $\lambda = 1$ is not a solution of $\phi(\lambda) = 0$. Therefore, all of the roots of $\phi(\lambda) = 0$ is smaller than one in absolute value. Accordingly, $\tilde{W}(t) \to 0$ as $t \to \infty$.

B Efficient Estimate of the Historical Hash Rate

We prove the (asymptotic) efficiency of (5) by showing that it is a maximum likelihood estimator. We assume that the hash rate is constant in previous T period; i.e., $H(s) = H(t)$ for $s = t, t-1, \ldots, t-T+1$. Then, since $B(s)$ follows $\mathrm{Exp}(W(s)H(t))$ i.i.d., the likelihood function is

$$\mathcal{L}(H(t)) = \prod_{s=t-T+1}^{t} W(s)H(t) \cdot \mathrm{e}^{-W(s)H(t)B(s)}.$$

Accordingly, the log-likelihood is

$$\begin{aligned}\log \mathcal{L}(H) &= \sum_{s=t-T+1}^{t} \{\log W(s) + \log H(t) - W(s)H(t)B(s)\} \\ &\propto \sum_{s=t-T+1}^{t} \{\log H(t) - W(s)H(t)B(s)\}\end{aligned}$$

The maximum likelihood estimator $\hat{H}(t)$ maximizes the above log-likelihood. Taking the first order condition, we have (5).

C Derivation of the Maximum Likelihood Estimator

We assume that the hash rate is linear in the reward; i.e., $H(t) = \alpha + \beta R(t)$. Hence, the following is our likelihood function.

$$\mathcal{L}(\alpha, \beta) = \prod_{t} W(t)(\alpha + \beta R(t)) \cdot \mathrm{e}^{-W(t)(\alpha+\beta R(t))B(t)}.$$

Accordingly, the log-likelihood is

$$\begin{aligned}\log \mathcal{L}(\alpha, \beta) &= \sum_t \{\log W(t) + \log(\alpha + \beta R(t)) - W(t)(\alpha + \beta R(t))B(t)\} \\ &\propto \sum_t \{\log(\alpha + \beta R(t)) - W(t)(\alpha + \beta R(t))B(t)\}\end{aligned}$$

We obtain the maximum likelihood estimators $(\hat{\alpha}, \hat{\beta})$ by numerically maximizing the log-likelihood. The value of our estimator is $\hat{\alpha} = 23.60$, $\hat{\beta} = 9.02$.

We will obtain the estimated variance of $(\hat{\alpha}, \hat{\beta})$ by computing the Fisher information matrix. By the standard property of the maximum likelihood estimator, we have

$$\mathrm{Var}\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = (I(\alpha_0, \beta_0))^{-1}$$

where $I(\alpha_0, \beta_0)$ is the Fisher information matrix derived as follows:

$$\begin{aligned} I(\alpha_0, \beta_0) &\coloneqq \mathbb{E}_{\{B(t)\}}\left[- \begin{pmatrix} \dfrac{\partial^2 \log \mathcal{L}(\alpha, \beta)}{\partial \alpha^2} & \dfrac{\partial^2 \log \mathcal{L}(\alpha, \beta)}{\partial \alpha \partial \beta} \\ \dfrac{\partial^2 \log \mathcal{L}(\alpha, \beta)}{\partial \alpha \partial \beta} & \dfrac{\partial^2 \log \mathcal{L}(\alpha, \beta)}{\partial \beta^2} \end{pmatrix} \Bigg|_{(\alpha,\beta)=(\alpha_0,\beta_0)} \right] \\ &= \mathbb{E}_{\{B(t)\}}\begin{bmatrix} \displaystyle\sum_t \frac{1}{(\alpha_0 + \beta_0 R(t))^2} & \displaystyle\sum_t \frac{R(t)}{(\alpha_0 + \beta_0 R(t))^2} \\ \displaystyle\sum_t \frac{R(t)}{(\alpha_0 + \beta_0 R(t))^2} & \displaystyle\sum_t \frac{R(t)^2}{(\alpha_0 + \beta_0 R(t))^2} \end{bmatrix} \\ &= \begin{pmatrix} \displaystyle\sum_t \frac{1}{(\alpha_0 + \beta_0 R(t))^2} & \displaystyle\sum_t \frac{R(t)}{(\alpha_0 + \beta_0 R(t))^2} \\ \displaystyle\sum_t \frac{R(t)}{(\alpha_0 + \beta_0 R(t))^2} & \displaystyle\sum_t \frac{R(t)^2}{(\alpha_0 + \beta_0 R(t))^2} \end{pmatrix}. \end{aligned}$$

We obtain an estimate of $I(\alpha_0, \beta_0)$ by substituting $(\hat{\alpha}, \hat{\beta})$ into I:

$$\widehat{I(\alpha_0, \beta_0)} = I(\hat{\alpha}, \hat{\beta}).$$

Hence, the estimated variance-covariance matrix of $(\hat{\alpha}, \hat{\beta})$ is

$$\widehat{\text{Var}}\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = (I(\hat{\alpha}, \hat{\beta}))^{-1} = \begin{pmatrix} 6.40 & -2.78 \\ -2.78 & 1.24 \end{pmatrix}.$$

References

Abadi, J. and M. Brunnermeier (2018): "Blockchain Economics," Working Paper.

Antonopoulos, A. M. (2014): *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*, O'Reilly Media, Inc.

Aoyagi, J. and D. Adachi (2019): "Economic Implications of Blockchain Platforms," Working paper.

Back, A. (2002): "Hashcash - A Denial of Service Counter-Measure," Tech. rep.

Basu, S., D. Easley, M. O'Hara, and E. G. Sirer (2019): "Towards a Functional Fee Market for Cryptocurrencies," Working Paper, arXiv:1901.06830.

Biais, B., C. Bisiere, M. Bouvard, and C. Casamatta (2019): "The Blockchain Folk Theorem," *The Review of Financial Studies*, 32, 1662–1715.

Bitcoin ABC (2018): "Difficulty Adjustment Algorithm Update," https://www.bitcoinabc.org/2017-11-01-DAA/, last visited 05/25/2019.

Booth, N. (2017): "Proposed New Algo for BCC Difficulty Adjustments," D578, Abandoned Proposal for Bitcoin Cash. https://reviews.bitcoinabc.org/D578, last visited 05/28/2019.

Bowden, R., H. P. Keeler, A. E. Krzesinski, and P. G. Taylor (2018): "Block arrivals in the Bitcoin blockchain," Working Paper, arXiv:1801.07447.

Budish, E. (2018): "The Economic Limits of Bitcoin and the Blockchain," Working Paper.

Chiu, J. and T. V. Koeppl (2017): "The Economics of Cryptocurrencies–Bitcoin and Beyond," Working Paper.

——— (2019): "Blockchain-Based Settlement for Asset Trading," *The Review of Financial Studies*, 32, 1716–1753.

Cong, L. W. and Z. He (2019): "Blockchain Disruption and Smart Contracts," *The Review of Financial Studies*, 32, 1754–1797.

Decker, C. and R. Wattenhofer (2013): "Information Propagation in the Bitcoin Network," in *IEEE P2P 2013 Proceedings*, IEEE, 1–10.

Dwork, C. and M. Naor (1992): "Pricing via Processing or Combatting Junk Mail," in *Annual International Cryptology Conference – CRYPTO' 92*, Springer, 139–147.

Easley, D., M. O'Hara, and S. Basu (2019): "From Mining to Markets: The Evolution of Bitcoin Transaction Fees," *Journal of Financial Economics*, forthcoming.

Harding, T. (2017): "Implement Weighted-Time Difficulty Adjustment Algorithm," D622, Abandoned Proposal for Bitcoin Cash. https://reviews.bitcoinabc.org/D622, last visited 05/28/2019.

Hashimoto, Y. and S. Noda (2019): "Pricing of Mining ASIC and Its Implication to the High Volatility of Cryptocurrency Prices," Working Paper.

Huberman, G., J. D. Leshno, and C. Moallemi (2019a): "An Economic Analysis of the Bitcoin Payment System," Working Paper.

——— (2019b): "An Economist's Perspective on the Bitcoin Payment System," in *AEA Papers and Proceedings*, vol. 109, 93–96.

Kaldor, N. (1934): "A Classificatory Note on the Determinateness of Equilibrium," *The Review of Economic Studies*, 1, 122–136.

Kraft, D. (2016): "Difficulty Control for Blockchain-Based Consensus Systems," *Peer-to-Peer Networking and Applications*, 9, 397–413.

LongHash (2019): "F2Pool Exclusive: Chinese Electricity for Mining Costs Between .04 to .06 USD per Kilowatt," https://www.longhash.com/news/f2pool-exclusive-chinese-electricity-for-mining-costs-between-04-to-06-usd-per-kilowatt, last visited 06/03/2019.

Lucas, R. E. (1976): "Econometric Policy Evaluation: A Critique," *Carnegie-Rochester Conference Series on Public Policy*, 1, 19 – 46.

Malinova, K. and A. Park (2017): "Market Design with Blockchain Technology," Working Paper.

Nakamoto, S. (2008): "Bitcoin: A Peer-to-Peer Electronic Cash System," The Bitcoin White Paper.

Narayanan, A., J. Bonneau, E. Felten, A. Miller, and S. Goldfeder (2016): *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press.

Rosenfeld, M. (2009): "Analysis of Hashrate-Based Double Spending," Working Paper, arXiv:1402.2009.

——— (2011): "Analysis of Bitcoin Pooled Mining Reward Systems," Working Paper, arXiv:1112.4980.

Séchet, A. (2017): "Implement Simple Moving Average Over Work Difficulty Adjustment Algorithm," D601, Accepted Proposal for Bitcoin Cash. https://reviews.bitcoinabc.org/D601, last visited 05/28/2019.

Taylor, M. B. (2017): "The Evolution of Bitcoin Hardware," *Computer*, 50, 58–66.

Tinn, K. (2018): "'Smart' Contracts and External Financing," Working Paper.