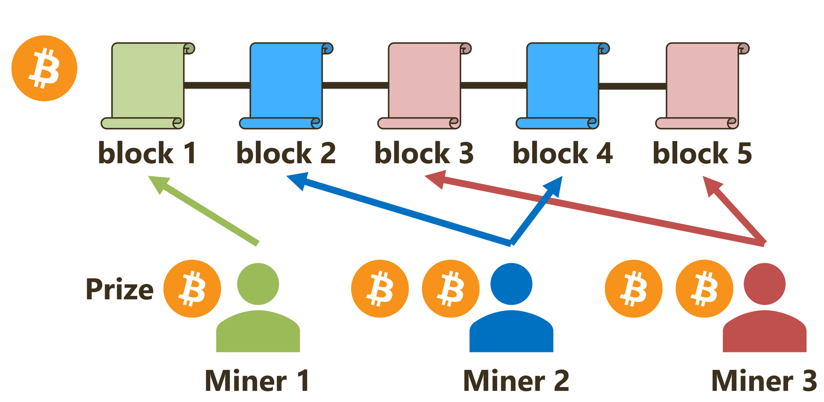
Miners' Reward elasticity and Stability of Competing Proof-of-Work Cryptocurrencies

Kohei Kawaguchi (HKUST, kkawaguchi@ust.hk)
Shunya Noda (The University of Tokyo, shunya.noda@e.u-Tokyo.ac.jp)

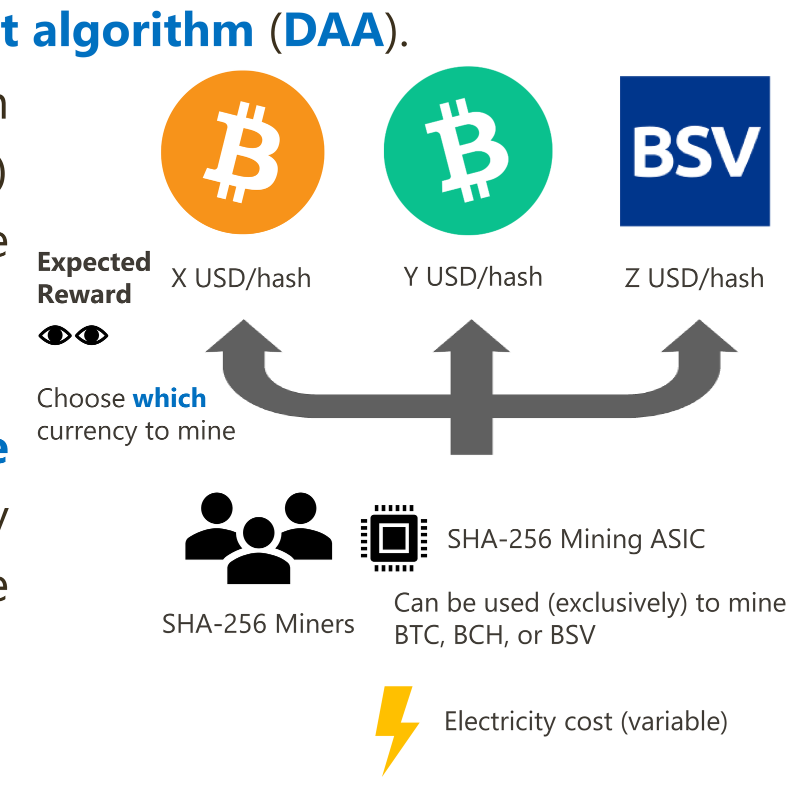
Summary

- **Proof-of-Work cryptocurrencies** (e.g., Bitcoin) hire **miners** to maintain the system by algorithmically setting the reward.
- Miners are freelance contributors and have strong discretion as to which cryptocurrency to contribute and how much they work for. Thus, the nature of miners' **hash supply** (≈ labor supply) is essential for the cryptocurrency's stability.
- Indeed, this paper (and our previous work, Noda, Okumura, and Hashimoto (2020)) shows that the combination of the **difficulty adjustment algorithm** (**DAA**, which controls miners' reward) and the value of the **reward-elasticity of the hash supply** is crucial for the cryptocurrency's stability.
- We develop a short-run supply-side model of the **multicurrency mining market** and **estimate** the hash supply elasticity of **Bitcoin** (**BTC**), **Bitcoin Cash** (**BCH**), and **Bitcoin SV** (**BSV**) by exploiting the discontinuity created by an event called **halving**.
- Bitcoin's DAA can stabilize the cryptocurrency only if the elasticity is low. The stability of Bitcoin hinges on external factors lowering the hash supply elasticity, such as the **interaction with competing currencies** (Bitcoin Cash, Bitcoin SV).
- By contrast, BCH and BSV are stable despite having a very **elastic hash supply** because they adopt **efficient difficulty adjustment algorithms**.
- By upgrading the difficulty adjustment algorithm, Bitcoin can **prevent possible future crises** before they happen.

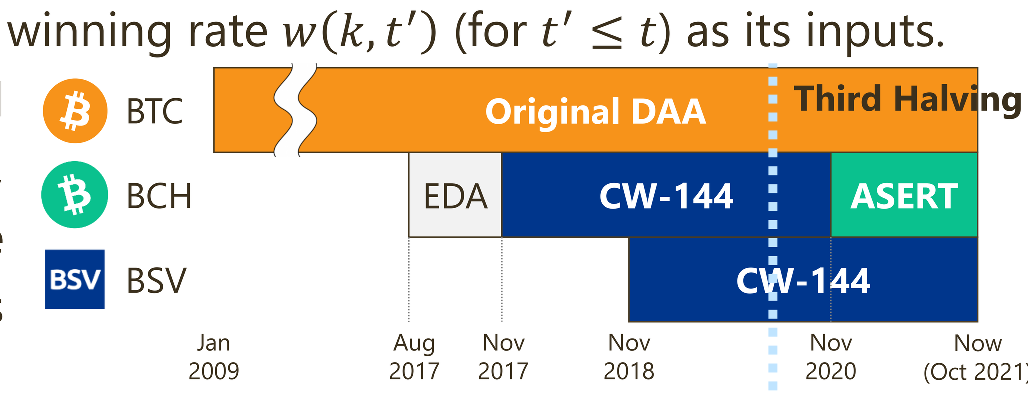
Technical Background + Model



- **Cryptocurrency** manages its transaction history using a ledger called blockchain.
- **Blockchain** is literally chain of blocks.
  - We consider a multi-currency market. Each (crypto)**currency** is indexed by k.
  - Block **height** is indexed by t.
- **Miners** produce new blocks and append them to the blockchain.
- Many miners work on this task, and upon producing t-th block of currency k, the block creator receives a **prize** $m(k,t)$ coins.
  - The value of $m(k,t)$ for each t is prescheduled. It is halved every 210,000 blocks (this event is called **halving**). The last halving (third halving) occurred in 2020, and it reduces m from 12.5 to 6.25.
  - **BCH** (April 8, 2020) → **BSV** (April 10, 2020) → **BTC** (May 11, 2020)
- To prevent a miner from **monopolizing** the ledger, the system wants to **randomly** choose the next block creator.
- To this end, Proof-of-Work cryptocurrency requires miners to **draw lotteries**.
  - Draw a lottery = Computing a **hash function** once (counted as 1 hash).
  - **Hash rate** $h(k,t)$ (hash/second) = **labor** input in a unit time.
  - The hash rate is not observable.
- The **winning rate** $w(k,t)$ (= the probability of success per each lottery draw) is a policy variable, using a **difficulty adjustment algorithm** (**DAA**).



- Miner's **expected reward** from a unit hash computation: $r(k,t) = w(k,t)m(k,t)e(k,t)$ (USD/hash), where $e(k,t)$ is the exchange rate between the cryptocurrency k and USD.
  - $r(k,t)$ is publicly observable.
  - We consider a **short-run supply-side model** of miners. Miners' capital (facility for mining) is fixed, and miners decide how to **operate** dynamically.
  - $h(k,t)$ is a function of $\left(r(k,t)\right)_{k\in K}$.
- **Block time** $b(k,t)$ (= time needed for producing one block) approximately follows an **exponential distribution** with mean $\mathbb{E}[b(k,t)] = 1/w(k,t)h(k,t)$ (second). BTC, BCH, and BSV aim at achieving $\mathbb{E}[b(k,t)] = 600$ seconds.
- **Difficulty Adjustment Algorithm** (**DAA**) selects a new winning rate $w(k,t+1)$ using past block time $b(k,t')$ and winning rate $w(k,t')$ (for $t' \le t$) as its inputs.
- Multiple DAAs have been used and implemented. **Original DAA**, **CW-144**, and **ASERT** are the names of three different DAAs that are studied in this paper.



Theoretical Prediction

- Preliminary results by Noda et al. (2020): In a single-currency model (i.e., we ignore BCH and BSV and focus only on BTC), the **original DAA**, **CW-144**, and **ASERT** asymptotically achieve the targeted average block time if and only if the (**own**) **reward-elasticity of hash supply** is smaller than **1**, **144**, and **575**, respectively.
- If the elasticity is larger than one, the original DAA causes the **overshoot** of the winning rate. **Too easy blocks** (the winning rate is too low) and **too difficult blocks** (too high) arrive alternately, and the block time **oscillates** and **diverges**.
  - Difficulty adjustment has **two effects**. (i) Change the speed of producing a block by directly changing the difficulty of it. (ii) Change the **hash rate** by changing the **reward** provided for miners. The **original DAA** completely dismisses (ii), and it does not work if (ii) is strong (i.e., hash supply is elastic).
- **CW-144** and **ASERT** are convergent in virtually any environment.
- In a multi-currency environment, the **cross elasticity** also matters.
  - **Direct effect:** BTC's winning rate ↑ ⇒ BTC's hash rate ↑
  - **Indirect effect:** BTC's winning rate ↑ ⇒ BCH's and BSV's hash rate ↓ ⇒ BCH and BSV increase their winning rate to maintain the block time = BCH's and BSV's reward ↑ ⇒ BTC's hash rate ↓
  - The indirect effect **attenuate** the total elasticity
    → The existence of **BCH and BSV stabilizes BTC**'s block time.

Estimation + Simulation

- We approximate the hash supply function by a log-log linear function.

$$h(k,t;\alpha,\beta) = \bar{h}(t)\cdot\exp\left(\alpha_k + \sum_{k'\in[K]}\beta_{k',k}\log r(k',t)\right)$$

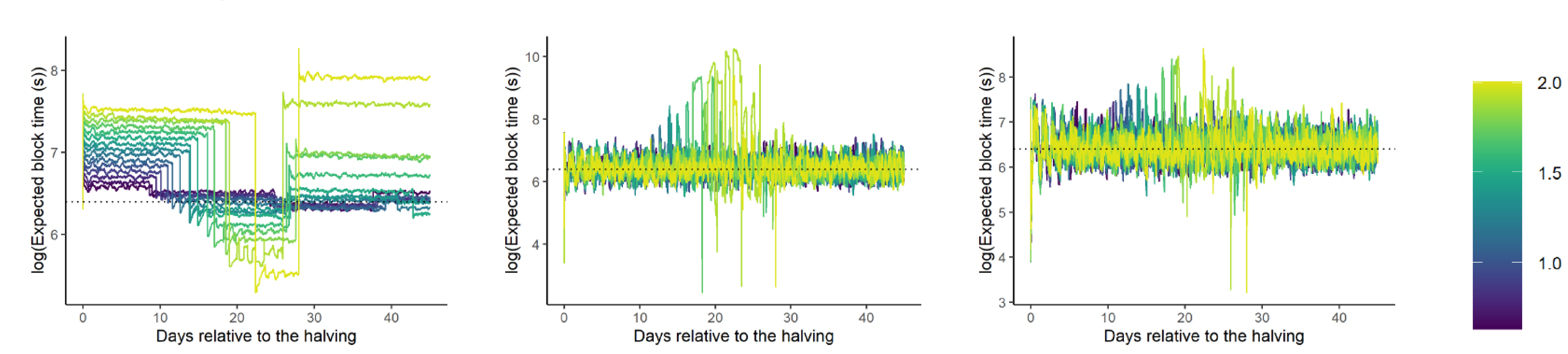
  - $\beta_{a,b}$ is currency a's reward elasticity of currency b's hash supply.
- We use MLE to estimate the parameters of (α,β).
- We identify β by looking at the data **before and after the third halving**.

Reward (From)	Constant (α)	Hash Supply (To): BTC	Hash Supply (To): BCH	Hash Supply (To): BSV
BTC	52.879***	0.626***	-3.981***	-3.186***
	(1.973)	(0.103)	(0.113)	(0.106)
BCH	49.851***	-0.240*	5.386***	-1.540***
	(1.995)	(0.095)	(0.127)	(0.093)
BSV	47.764***	-0.223*	-1.219***	4.869***
	(1.973)	(0.098)	(0.076)	(0.118)

Constant (α) — Own- and Cross-Elasticity (β)

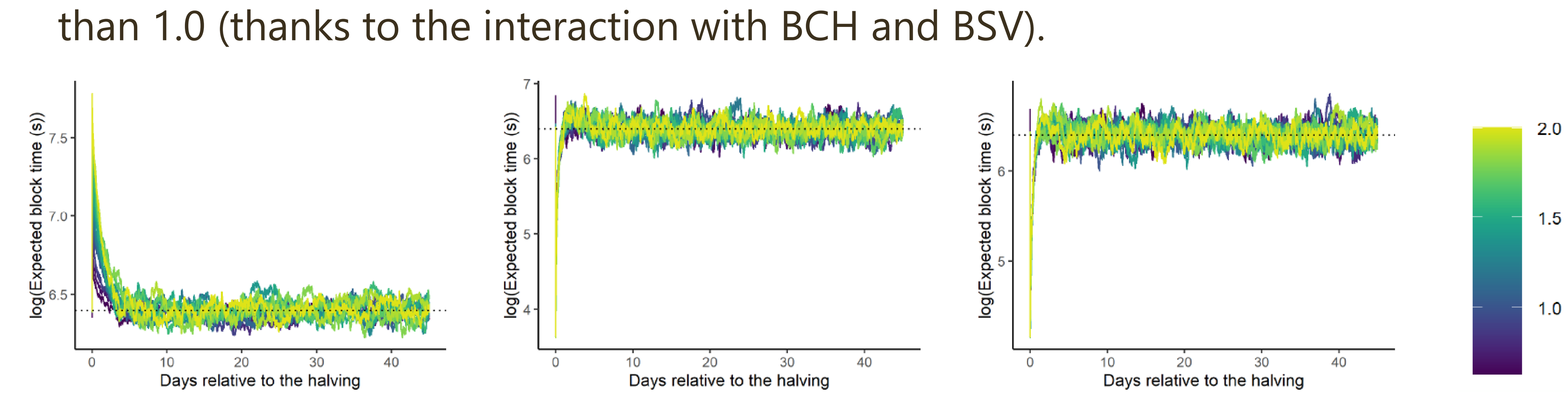
* $p < 0.05$
** $p < 0.01$
*** $p < 0.001$

- The hash supply is increasing in its own reward (diagonal elements) and decreasing in its rivals' reward (off-diagonal elements).
- **BTC**'s hash supply is very **inelastic** (own elasticity < 1). Therefore, BTC has survived despite it has used the inefficient original DAA.
- **BCH**'s and **BSV**'s own elasticities are much larger than 1. They were not to survive if the original DAA were maintained.



(a) BTC: actual (b) BCH: actual (c) BSV: actual

- The simulation shows that if BTC faces a larger $\beta_{\mathrm{BTC,BTC}}$, BTC's block time starts to **oscillate** and **diverge**. The threshold is around 1.5, which is substantially larger than 1.0 (thanks to the interaction with BCH and BSV).



(g) BTC: all ASERT (h) BCH: all ASERT (i) BSV: all ASERT

- By upgrading BTC's DAA to the state-of-the-art DAA, **ASERT**, we can stabilize BTC's block arrival for virtually any own elasticity.
- Upgrading also stabilizes the hash rate, which contributes to the improvement of the security-cost efficiency (i.e., cryptocurrency's security per energy consumption).