The Economics of Cryptocurrencies

Jonathan Chiu\textsuperscript{1}  Thorsten Koeppl\textsuperscript{2}

\textsuperscript{1}Bank of Canada
\textsuperscript{2}Queen’s U

Economics of Payments IX
November 2018

Disclaimer: The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Canada.
What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.

   ▶ A ledger of **digital** balances updated in a **decentralized** fashion.

2) We show that cryptocurrencies cannot achieve immediate and final settlement.

   ▶ Why? Need to avoid a double spending problem.

3) We evaluate the efficiency of a cryptocurrency system.

   ▶ Positive inflation is optimal while transaction fees should be minimized.

   ▶ Currently, welfare loss in BITCOIN of 1.4% of consumption, but potentially as low as 0.08%.
What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.
   ▶ A ledger of digital balances updated in a decentralized fashion.

2) We show that cryptocurrencies cannot achieve immediate and final settlement.
   ▶ Why? Need to avoid a double spending problem.
What We Do

1) We formally model a cryptocurrency system according to the Bitcoin protocol.
   - A ledger of digital balances updated in a decentralized fashion.

2) We show that cryptocurrencies cannot achieve immediate and final settlement.
   - Why? Need to avoid a double spending problem.

3) We evaluate the efficiency of a cryptocurrency system.
   - Positive inflation is optimal while transaction fees should be minimized.
   - Currently, welfare loss in BITCOIN of 1.4% of consumption, but potentially as low as 0.08%.
Why Cryptocurrency is Special?
Why Cryptocurrency is Special?
Why Cryptocurrency is Special?

Physical Tokens

Immediate and Final Settlement
Why Cryptocurrency is Special?

Digital Tokens

Subject to double-spending problems
Digital Currency with Trusted Third Party

B sends $1 to S
Digital Currency without Trusted Third Party

B sends $1 to S

No central authority to keep record
How Cryptocurrency works

1. Consensus Protocol

2. Reward Scheme

3. Confirmation Lags
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)

2. Reward Scheme

3. Confirmation Lags
Blockchain

A book containing the ledger of all past transactions

$B_{T-1}$
Blockchain

Transactions broadcasted to the miners for validation
The miner who first completes the proof-of-work can update the blockchain.
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner

2. Reward Scheme

3. Confirmation Lags
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner

2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees

3. Confirmation Lags
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner

2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees

3. Confirmation Lags
   - double spending is discouraged by confirmation delay
No confirmation lag (N=0)

Time $T$

Goods delivered on the spot

Seller gets payment after confirmation

$b$

$B_{T-1}$

$B_T$

Miners

Goods delivered on the spot

Seller gets payment after confirmation
No confirmation lag (N=0)

Time $T$

Goods delivered on the spot

Seller gets payment after confirmation

Buyer perform secret mining to undo the payment

$B_{T-1}$

$B$ $S$

miners

$B_T$

$B$ $S$

miners

$b$ $a$
No confirmation lag (N=0)

Buyer perform secret mining to undo the payment. If successful, buyer gets goods without paying the seller.

If successful, buyer gets goods without paying the seller.

Buyer perform secret mining to undo the payment.
One confirmation lag (N=1)

- After time $T$, miners are involved.
- Goods delivered after one confirmation.
- Seller gets payment after one confirmation.

$B_{T-1}$

$B_T$

Goods delivered after one confirmation

Seller gets payment after one confirmation
One confirmation lag (N=1)

Secret mining for one period is not sufficient
One confirmation lag (N=1)

Need to secretly mine for N+1=2 period
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner

2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees

3. Confirmation Lags
   - double spending is discouraged by confirmation delay
   - if goods are delivered after $N$ validations are observed, then the buyer needs to win the mining game $N + 1$ times to revoke the transaction
How Cryptocurrency works

1. Consensus Protocol
   - competition in the form of mining: “miners” compete to update the public ledger (i.e. Blockchain)
   - the prob. of winning is proportional to the fraction of computational power owned by a miner

2. Reward Scheme
   - reward winning miners by seigniorage and transaction fees

3. Confirmation Lags
   - double spending is discouraged by confirmation delay
   - if goods are delivered after $N$ validations are observed, then the buyer needs to win the mining game $N + 1$ times to revoke the transaction
Questions

Take as given the design of the cryptocurrency system:

1. How well does it function as a payment system?

2. How to optimally set policy parameters?  
   e.g. currency growth, transaction fees

3. How best to use it for different types of transactions?  
   e.g. retail vs large value
Model
Environment

Based on Lagos and Wright (2005)

Time is discrete: \( t = 0, 1, 2, \ldots \)

Three types of agents.
- \( B \) buyers
- \( \sigma B \) sellers
- \( M \) miners

Buyers and seller use balances recorded in a ledger to finance bilateral trade.

Balances in the ledger grow at rate \( \mu \) and there are transaction fees \( \tau \).
Proof-of-Work

$M$ miners compete to update the ledger by solving a costly computational task with a random success rate.

Miner $i$ chooses computer power $q_i$ to maximize profits

$$\rho(q_i) R - q_i \alpha$$

where

- $R$ mining reward in real terms
- $\alpha$ price of computer power
- $\rho$ probability of winning given by

$$\rho(q_i) = \frac{q_i}{\sum_{m=1}^{M} q_m}$$

**Results:**

1) Higher $R$ induces higher mining activities $\sum_{m=1}^{M} q_m = MQ$.

2) As $M \to \infty$, all rents $R$ are dissipated.
Trading

- Preferences
  - Buyer: \( \varepsilon u(x_t) - h_t \), where \( \varepsilon \sim F \)
  - Seller: \( -c(x_t) + h_t \)

- Trading
  - Day: buyer sells \( h \) to acquire balances \( z \)
  - Night: spends \( d \leq z \) to buy \( x \) from a seller
  - Next day: the seller uses \( d \) to buy \( h \)
Night Trading

In session 0, a buyer meets with a seller and makes a take-it-or-leave-it-offer \((x, d, N)\)

- immediate payment \(d\) in real balances
- \(x\) goods to be delivered after confirmations of the payment in \(N\) consecutive blocks

After trade, the buyer can attempt to double spend
Incentives to Double Spend
Transactions in Lagos-Wright

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \bar{N} \]

- \( n = 0 \): start with \((z_b, z_s)\)
- buyer offers \((x, d)\)
- seller rejects 
- buyer accepts \((z_b - d, z_s + d)\)

Diagram:
- Buyer offers \((x, d)\)
- Seller rejects
- Buyer accepts \((z_b - d, z_s + d)\)
Double-Spending Problem

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \tilde{N} \]

- Start with \((z_b, z_s)\)
- Buyer offers \((x, d, N)\) to seller

Double-Spending Problem
Double-Spending Problem

\[
\begin{align*}
n &= 0 & n &= 1 & n &= 2 & \ldots & n &= N & \ldots & n &= \bar{N} \\
\text{start with} \quad & (z_b, z_s) \\
\text{buyer offers} \quad & (x, d, N) \\
\text{seller} \quad & \text{rejects} \\
\text{accepts} \\
\text{buyer chooses} \quad & q_0 \\
\end{align*}
\]
Double-Spending Problem

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \bar{N} \]

- Start with \((x, d, N)\)
- Buyer offers \((z_b, z_s)\)
- Seller
  - Rejects \((z_b, z_s)\)
  - Accepts
    - Buyer chooses \(q_0\)
      - Secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
    - Succeeds
      - Buyer chooses \(q_1\)

Double Spending Problem
Double-Spending Problem

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \bar{N} \]

start with \((z_b, z_s)\)

buyer offers \((x, d, N)\)

seller

rejects

\((z_b, z_s)\)

accepts

buyer chooses \(q_0\)

secret mining fails

\((z_b - d, z_s + d(1 - \tau))\)

succeeds

buyer chooses \(q_2\)

\[ \text{Double Spending Problem} \]

Chiu & Koeppl – Cryptocurrencies 32
Double-Spending Problem

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \bar{N} \]

- Start with \((z_b, z_s)\)
- Buyer offers \((x, d, N)\)
- Seller rejects \((z_b, z_s)\)
- Buyer accepts \((z_b, z_s)\)
- Buyer chooses \(q_0\)
- Secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
- Buyer chooses \(q_1\)
- Secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
- Buyer chooses \(q_2\)
- Secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
- Buyer chooses \(q_N\)
- Succeeds

Double Spending Problem
Double-Spending Problem

\[ n = 0 \quad n = 1 \quad n = 2 \quad \ldots \quad n = N \quad \ldots \quad n = \bar{N} \]

- **Start with** \((z_b, z_s)\)
- **Buyer offers** \((x, d, N)\)
- **Seller rejects** \((z_b, z_s)\)
- **Accepts**
  - Buyer chooses \(q_0\) secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
  - Succeeds
  - Buyer chooses \(q_1\) secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
  - Succeeds
  - Buyer chooses \(q_2\) secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
  - Succeeds
  - Buyer chooses \(q_N\) secret mining fails \((z_b - d, z_s + d(1 - \tau))\)
  - Succeeds

\(\downarrow\) Double Spending Problem

\(\uparrow\) Double Spending Problem
No Double Spending Constraint

For any contract \((x, d, N)\), the expected payoff from a DS attempt is

\[
D_0(d, N) = \max_{\{q_n\}_{n=0}^{N}} P^{\beta} \frac{[d + R(1 + N)]}{\mu} - \sum_{n=0}^{N} \left( \prod_{t=0}^{n-1} \frac{q_n}{QM + q_n} \right) \alpha q_n
\]

where

\[
P = \prod_{n=0}^{N} \left( \frac{q_n}{QM + q_n} \right) \text{ is the prob. of success}
\]

\[
R = \frac{Z(\mu - 1) + D\tau}{N + 1} \text{ are the rewards from mining}
\]

**Lemma**

If \(D_0(d, N) = 0\), then the contract \((x, d, N)\) is double-spending proof.
Double-Spending Proof Contracts

Proposition

Suppose $M \to \infty$. A contract $(x,d,N)$ is double-spending proof (i.e. settlement is final) if

$$d < R(N + 1)N.$$ 

Otherwise, the settlement is final only with probability

$$1 - P(d, N) = \frac{N + 1}{\sqrt{\frac{d}{R} + (N + 1)}}.$$ 

Results:

- Settlement cannot be both immediate ($N = 0$) and final ($P = 0$).
- Rewards help discourage double spending and improve finality.
- There is a trade-off between trade size $d$, settlement lag $N$ and finality $1 - P$. 
Key Trade-off

Figure: Trade Size vs. Settlement Lag vs. Finality
Cryptocurrency Equilibrium

**Definition**

A DS-proof cryptocurrency equilibrium with \((\mu, \tau)\) and \(M \to \infty\) is given by contracts \((x(\varepsilon), d(\varepsilon), N(\varepsilon))\), money demand \(z(\varepsilon)\) and a mining choice \(q\) such that

1. the contracts satisfy the No-DS-constraint,
2. the money demand and the offer maximizes a buyer’s utility,
3. the mining choice maximizes a miner’s utility
4. and markets clear.

**Theorem**

A DS-proof cryptocurrency equilibrium exists for \(B\) sufficiently large.
Optimal Reward Scheme

Define social welfare as

$$\mathcal{W} = B \int \left[ \sigma \delta^{N(\varepsilon)} \varepsilon u(x(\varepsilon)) - x(\varepsilon) \right] dF_\varepsilon(\varepsilon) - \frac{\beta}{\mu} R(\bar{N} + 1)$$

- trade surplus
- mining costs

Proposition

The optimal reward structure sets transaction fees to zero and only relies on seignorage: $\tau = 0$ and $\mu > 1$. 
Optimal Reward Scheme

Define social welfare as

$$\mathcal{W} = B \int \left[ \sigma \delta^{N(\varepsilon)} \varepsilon u(x(\varepsilon)) - x(\varepsilon) \right] dF_\varepsilon(\varepsilon) - \frac{\beta}{\mu} R(\bar{N} + 1)$$

- trade surplus
- mining costs

Proposition

The optimal reward structure sets transaction fees to zero and only relies on seignorage: $$\tau = 0$$ and $$\mu > 1$$.

- The reason is that the inflation tax is shared by all buyers while transaction fees are paid only by the active ones who have a high valuation of money.
- ... levying reward costs upfront in terms of inflation allows distortions to be smoothed out across all buyers.
**Optimal Reward Scheme**

Define social welfare as

\[ \mathcal{W} = B \int \left[ \sigma \delta^{N(\varepsilon)} \varepsilon u(x(\varepsilon)) - x(\varepsilon) \right] dF_{\varepsilon}(\varepsilon) - \frac{\beta}{\mu} R(\bar{N} + 1) \]

- **Trade surplus**
- **Mining costs**

**Proposition**

The optimal reward structure sets transaction fees to zero and only relies on seignorage: \( \tau = 0 \) and \( \mu > 1 \).

- The reason is that the inflation tax is shared by all buyers while transaction fees are paid only by the active ones who have a high valuation of money.
- ... levying reward costs upfront in terms of inflation allows distortions to be smoothed out across all buyers
- Implication: long-run zero currency growth is suboptimal
Quantitative Assessment
Calibration – Basic Parameters

<table>
<thead>
<tr>
<th></th>
<th>values</th>
<th>targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>0.999916</td>
<td>period length = 1 day</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.999999</td>
<td>block time = 10 min</td>
</tr>
<tr>
<td>$\mu$</td>
<td>1.00025</td>
<td>money growth (9.6% p.a.)</td>
</tr>
<tr>
<td>$\tau$</td>
<td>0.000088</td>
<td>total fees/vol per block</td>
</tr>
<tr>
<td>$B$</td>
<td>6873428</td>
<td>max. # of average-sized transactions</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>0.0178</td>
<td>vol per day/total BTC</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1</td>
<td>normalized</td>
</tr>
</tbody>
</table>

Source: 2015 data from Blockchain.info

▶ We use log utility.
▶ We use data on the distribution of transactions.
▶ Confirmation lags cannot be observed directly.
1. Welfare Comparison

<table>
<thead>
<tr>
<th>Regime</th>
<th>Welfare Cost as % of consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cash (Friedman Rule)</td>
<td>0%</td>
</tr>
<tr>
<td>Cash (2% inflation)</td>
<td>0.003%</td>
</tr>
<tr>
<td>Bitcoin (benchmark)</td>
<td>1.410%</td>
</tr>
<tr>
<td>$\mu - 1 = 9.5%, \tau = 0.0088%$</td>
<td>mining cost: $359.98 millions</td>
</tr>
<tr>
<td>Bitcoin (optimal policy)</td>
<td>0.080%</td>
</tr>
<tr>
<td>$\mu - 1 = 0.17%, \tau = 0%$</td>
<td>mining cost: $6.9 millions</td>
</tr>
</tbody>
</table>

- Welfare loss is currently very large mainly due to the mining cost.
- ... can be reduced substantially by lowering money growth and setting transaction fees to zero.
- Long-run BTC design will bring money growth to 0 and is, thus, inefficient.
## 2. Best Usage of Cryptocurrency Technology

<table>
<thead>
<tr>
<th></th>
<th>Retail Payments (US Debit cards)</th>
<th>Large Value Payments (Fedwire)</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg transaction size</td>
<td>$38.29</td>
<td>$6552236</td>
</tr>
<tr>
<td>annual volume</td>
<td>59539 millions</td>
<td>135 millions</td>
</tr>
<tr>
<td>optimal $\mu$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>optimal $\tau$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>confirmation lag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>welfare loss</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mining cost (per year)</td>
<td>$4.33 millions</td>
<td>$22.10 billions</td>
</tr>
</tbody>
</table>

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume

Chiu & Koeppel – Cryptocurrencies 41
2. Best Usage of Cryptocurrency Technology

<table>
<thead>
<tr>
<th></th>
<th>Retail Payments (US Debit cards)</th>
<th>Large Value Payments (Fedwire)</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg transaction size</td>
<td>$38.29</td>
<td>$6552236</td>
</tr>
<tr>
<td>annual volume</td>
<td>59539 millions</td>
<td>135 millions</td>
</tr>
<tr>
<td>optimal $\mu$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>optimal $\tau$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>confirmation lag</td>
<td></td>
<td></td>
</tr>
<tr>
<td>welfare loss</td>
<td>0.00052%</td>
<td>0.0060%</td>
</tr>
<tr>
<td>mining cost (per year)</td>
<td>$4.33 millions</td>
<td>$22.10 billions</td>
</tr>
</tbody>
</table>

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume
- retail system incurs a lower welfare loss and mining costs

Chiu & Koeppi – Cryptocurrencies
2. Best Usage of Cryptocurrency Technology

<table>
<thead>
<tr>
<th></th>
<th>Retail Payments (US Debit cards)</th>
<th>Large Value Payments (Fedwire)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>avg transaction size</strong></td>
<td><strong>$38.29</strong></td>
<td><strong>$6552236</strong></td>
</tr>
<tr>
<td><strong>annual volume</strong></td>
<td><strong>59539 millions</strong></td>
<td><strong>135 millions</strong></td>
</tr>
<tr>
<td><strong>optimal $\mu$</strong></td>
<td><strong>0.038%</strong></td>
<td><strong>0.53%</strong></td>
</tr>
<tr>
<td><strong>optimal $\tau$</strong></td>
<td><strong>0%</strong></td>
<td><strong>0%</strong></td>
</tr>
<tr>
<td><strong>confirmation lag</strong></td>
<td><strong>2 mins</strong></td>
<td><strong>12 mins</strong></td>
</tr>
<tr>
<td><strong>welfare loss</strong></td>
<td><strong>0.00052%</strong></td>
<td><strong>0.0060%</strong></td>
</tr>
<tr>
<td><strong>mining cost (per year)</strong></td>
<td><strong>$4.33 millions</strong></td>
<td><strong>$22.10 billions</strong></td>
</tr>
</tbody>
</table>

- DS-proof iff $d < R \cdot N(1 + N)$
  - retail: small trade size, high volume
  - interbank: large trade size, low volume
- retail system incurs a lower welfare loss and mining costs
- ... requires smaller rewards
- ... induces shorter confirmation lags
What to Take Away

1) Owing to its digital nature, a cryptocurrency is fundamentally different from cash.

2) One can understand the economics of such a system well by looking at the incentives to double-spend.

3) BITCOIN is not only really expensive in terms of mining costs, but also inefficient in its long-run design.

4) It provides a more efficient payment system when the volume of transactions is large relative to the individual transaction size.

On-going project: Blockchain for security settlement, cross-border payments, ...
Thanks!
Appendix
Microfoundations for Mining

Investing computing power $q_m$ allows a miner to solve the PoW problem with probability

$$F(t) = 1 - e^{-\mu_m \cdot t}$$

within a time interval $t$, where $1/\mu_m = D/q(m)$ is the expected time to solve the problem.

Hence, $D$ is the difficulty parameter for the PoW problem.

The first solution among miners, min($\tau_1, \ldots, \tau_M$), is thus also exponentially distributed and the probability for any miner to solve it first is given by

$$\rho_n(q_n) = \frac{q_n}{\sum_{m=1}^{M} q_m}.$$
Oligopolistic Mining Equilibrium

Maximizing profits by miner $j$ yields as a FOC

$$\left( \frac{\sum_{i=1}^{N} q_i - q_j}{\left( \sum_{i=1}^{N} q_i \right)^2} \right) \frac{\beta}{\mu} R = \alpha$$

Imposing symmetry, we obtain for the total mining cost

$$C = \alpha M Q = \frac{M - 1}{M} \frac{\beta}{\mu} R.$$ 

For $M \to \infty$ all rents are dissipated and we obtain

$$C = \frac{\beta}{\mu} R.$$
Trading

\[ \text{Day} \quad \text{Night} \quad \text{Day} \]

\[ t \quad t + 1 \]

\begin{align*}
\text{buyer:} & \quad \text{sell } h & \text{buy } x \\
\text{seller:} & \quad \text{sell } x & \text{buy } h
\end{align*}

Two markets

- centralized market in day
- decentralized market at night

Preferences

- Buyer: \( \varepsilon u(x_t) - h_t \), where \( \varepsilon \sim F \)
- Seller: \( -x_t + h_t \)

Trading

- Day: buyer sells \( h \) to acquire real balances \( z \)
- Night: spends \( d \leq z \) to buy \( x \) from a seller
- Next day: the seller uses \( d \) to buy \( h \)
Day Market

The value of a buyer who draws $\varepsilon$ is

$$\max_{z', h} -h + V(z'; \varepsilon)$$

subject to

$$h + z \geq z' \geq 0$$

where $z'$ are the real balances carried to the night market.

**Assumption:**
Transactions can be perfectly monitored and there is full liability so that double spending is not a problem.
Night Market

The night market is divided into $\tilde{N} + 1$ trading sessions.

- In session 0, a buyer meets with a seller w.p. $\sigma$ and makes a take-it-or-leave-it-offer $(x, d, N)$.
- There is immediate payment $d$ in real balances.
- $x$ goods are to be delivered after confirmation of the payment in $N$ consecutive blocks.

The offer $(x, d, N)$ determines whether the buyer has an incentive to double spend or not.
Optimal DS Proof Contracts

At the start of the night market, the buyer with \( z \) makes a take-it-or-leave-it offer \((x, d, N)\) to a seller.

The buyer will never carry more real balances than necessary so that \( z = d \) and the offer is given by \((x(d), N(d))\).

Requiring the offer to be double spending proof the buyer solves

\[
\max_{(x,d,N)} -d + V(d; \varepsilon)
\]

subject to

\[
V(d; \varepsilon) = \sigma \delta^N \varepsilon u(x) + (1 - \sigma) \frac{\beta}{\mu} d
\]

\[
x \leq \frac{\beta}{\mu} d(1 - \tau)
\]

\[
d \leq R(N + 1)N
\]
Sufficient Condition for DS proof

The optimal contract is DS proof if

\[ \sigma [\delta \varepsilon_{\text{max}} u'(\bar{x})(1 - \tau)E(x) - 1] < i \]

where

\[ \bar{x} = (1 - \tau)2R \] is the maximum trade size with \( N = 1 \)
\[ E(x) \leq \frac{3}{4} \] is the elasticity of \( x \) w.r.t. \( d \) at \( N = 1 \)

The reason is that the tightest constraint to avoid DS is a confirmation lag of \( N = 1 \).

This condition is satisfied when

- the opp. cost of carrying balances is high (\( i \) is high)
- the matching friction is high (\( \sigma \) is low)
- the marginal utility is low (\( \varepsilon \) is low)
Existence Proof

We use Kakutani’s Fixed Point Theorem.

Fix \((\mu, \tau)\). The reward \(R\) determines the aggregate money supply \(S(R)\) which in turn determines total rewards \(R'\). Hence, we need to find a fixed point for \(R\) given aggregate money demand for a correspondence

\[
T(R) = \left(\frac{(\mu - 1) + \sigma\tau}{N + 1}\right) S(R).
\]

Aggregate money demand can be shown to be u.h.c, convex in \(R\) which pins down the aggregate transaction fees and, hence, \(R'\).

Furthermore, given \(B\) sufficiently large, we can find a lower bound on \(R_{\text{min}} > 0\) such that \(R > R_{\text{min}}\).

Hence, we can restrict the correspondence to a compact set and show that the correspondence has a closed graph.
Optimal Contracts

We use data on transactions to recover the implied distribution of $\varepsilon$.

**Figure**: Implied Distribution of Shocks

**Figure**: Optimal Delay
Higher inflation implies distortions and higher mining costs ...

.. but positive inflation is optimal due to lower confirmation lags.
Optimal Design II – Effects of Transaction Fees

- Same trade-off ...
- ... but zero transaction costs seem to be optimal given $\mu > 0$. 