On profitability of block withholding strategies

Ricardo Pérez-Marco (CNRS, IMJ-PRG, Paris 7)
(joint work with C. Grunspan)

IFIP WG Performance 2018 SOCCA

Toulouse, France

December 6, 2018

"On profitability of Selfish Mining", ArXiv:1805.08281, 5/2018.
"On profitability of Stubborn Mining", ArXiv:1808.01041, 8/2018.
"On profitability of Trailing Mining", ArXiv:1811.09322, 11/2018.

On profitability of Selfish Mining

On profitability of Selfish Mining

Selfish Mining

Selfish Mining

History

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.
- L. Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power" ArXiv:1312.7013, 12/2013.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.
- L. Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power" ArXiv:1312.7013, 12/2013.
- Further papers and textbooks.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.
- L. Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power" ArXiv:1312.7013, 12/2013.
- Further papers and textbooks.

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.
- L. Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power" ArXiv:1312.7013, 12/2013.
- Further papers and textbooks.

Bitcoin Protocol rule "Bitcoin miners release blocks as soon as they are validated".

Selfish Mining

History

- RHornings's Bitcointalk thread "Mining cartel attack", 12/2010.
- M. Rosenfeld "Analysis of Bitcoin pooled mining reward systems" ArXiv:1112.4980, 12/2011.
- I. Eyal, E.M. Sirer "Majority is not Enough: Bitcoin Mining is Vulnerable" ArXiv:1311.0243, 11/2013.
- L. Bahack, "Theoretical Bitcoin Attacks with less than Half of the Computational Power" ArXiv:1312.7013, 12/2013.
- Further papers and textbooks.

Bitcoin Protocol rule "Bitcoin miners release blocks as soon as they are validated".

Bitcoin Stability Conjecture Protocol rules are aligned with self-interest of the network actors.

Description

Description

General block withholding strategies

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.
- Relies on a good connection so that a share of $0 < \gamma \leq 1$ miners adopt the selfish block in case of competition.

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.
- Relies on a good connection so that a share of $0 < \gamma \leq 1$ miners adopt the selfish block in case of competition.

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.
- Relies on a good connection so that a share of $0 < \gamma \leq 1$ miners adopt the selfish block in case of competition.

Consequences

- Slows the network, hence it reduces the total "Profit and Loss" (PnL) per unit if time of the network.

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.
- Relies on a good connection so that a share of $0 < \gamma \leq 1$ miners adopt the selfish block in case of competition.

Consequences

- Slows the network, hence it reduces the total "Profit and Loss" (PnL) per unit if time of the network.
- Creates a large amount of orphan blocks (hence, the attack is noticeable)

Description

General block withholding strategies

- Witheld blocks trying to build an advantage with a relative hashing power $0 < q < 1/2$.
- Timely release blocks to invalidate blocks validated by honest miners.
- Relies on a good connection so that a share of $0 < \gamma \leq 1$ miners adopt the selfish block in case of competition.

Consequences

- Slows the network, hence it reduces the total "Profit and Loss" (PnL) per unit if time of the network.
- Creates a large amount of orphan blocks (hence, the attack is noticeable)
- All other things being equal, after 2016 blocks, the diffficulty adjusts down.

Selfish mining algorithm

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

- If $\Delta = 0$ the SM mines normally.

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

- If $\Delta = 0$ the SM mines normally.
- If $\Delta = 1$ then the SM broadcasts his block. A competition follows.

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

- If $\Delta = 0$ the SM mines normally.
- If $\Delta = 1$ then the SM broadcasts his block. A competition follows.
- If $\Delta = 2$ then the SM broadcasts his secret fork.

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

- If $\Delta = 0$ the SM mines normally.
- If $\Delta = 1$ then the SM broadcasts his block. A competition follows.
- If $\Delta = 2$ then the SM broadcasts his secret fork.
- If $\Delta \geq 3$ then the SM broadcasts blocks from his secret fork to match the length of the public blockchain.

Selfish mining algorithm

Let $\Delta \geq 0$ be the advance of the secret fork over the public blockchain. When the honest miners validate a block then (Selfish Mining (SM) algorithm):

- If $\Delta = 0$ the SM mines normally.
- If $\Delta = 1$ then the SM broadcasts his block. A competition follows.
- If $\Delta = 2$ then the SM broadcasts his secret fork.
- If $\Delta \geq 3$ then the SM broadcasts blocks from his secret fork to match the length of the public blockchain.
- Except in the first two cases, the SM keeps working on top of his secret fork.

Stubborn Mining algorithms

Stubborn Mining algorithms

- Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

Stubborn Mining algorithms

- Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

Stubborn Mining algorithms

- Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

- Lead-Stubborn Mining (LSM) When $\Delta \geq 2$ as in SM with $\Delta \geq 3$, and for $\Delta = 1$ releases all the secret fork and mines normally on top of it.

Stubborn Mining algorithms

- Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

- Lead-Stubborn Mining (LSM) When $\Delta \geq 2$ as in SM with $\Delta \geq 3$, and for $\Delta = 1$ releases all the secret fork and mines normally on top of it.
- Equal Fork Stubborn Mining (EFSM) As in the previous case for $\Delta = 1$, but if the deviant miner finds a new block he does not reveal it.

Stubborn Mining algorithms

• Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

- Lead-Stubborn Mining (LSM) When $\Delta \geq 2$ as in SM with $\Delta \geq 3$, and for $\Delta = 1$ releases all the secret fork and mines normally on top of it.
- Equal Fork Stubborn Mining (EFSM) As in the previous case for $\Delta = 1$, but if the deviant miner finds a new block he does not reveal it.
- A-Trailing Stubborn Mining (TSM_A) The deviant miner keeps mining on top of his fork even when $\Delta \leq 0$ and only gives up when $\Delta = -A$ for $A = 1, 2, \ldots$.

Stubborn Mining algorithms

• Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

- Lead-Stubborn Mining (LSM) When $\Delta \geq 2$ as in SM with $\Delta \geq 3$, and for $\Delta = 1$ releases all the secret fork and mines normally on top of it.
- Equal Fork Stubborn Mining (EFSM) As in the previous case for $\Delta = 1$, but if the deviant miner finds a new block he does not reveal it.
- A-Trailing Stubborn Mining (TSM_A) The deviant miner keeps mining on top of his fork even when $\Delta \leq 0$ and only gives up when $\Delta = -A$ for $A = 1, 2, \ldots$.

Stubborn Mining algorithms

• Nayak-Kumar-Miller-Shi, *"Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack"*, IEEE European Symp. Security and Privacy, 2016.

These are variations of SM algorithm.

- Lead-Stubborn Mining (LSM) When $\Delta \geq 2$ as in SM with $\Delta \geq 3$, and for $\Delta = 1$ releases all the secret fork and mines normally on top of it.
- Equal Fork Stubborn Mining (EFSM) As in the previous case for $\Delta = 1$, but if the deviant miner finds a new block he does not reveal it.
- A-Trailing Stubborn Mining (TSM_A) The deviant miner keeps mining on top of his fork even when $\Delta \leq 0$ and only gives up when $\Delta = -A$ for $A = 1, 2, \ldots$.

Markov chain model

Markov chain model

Markov chain or “state machine” models.

Markov chain model

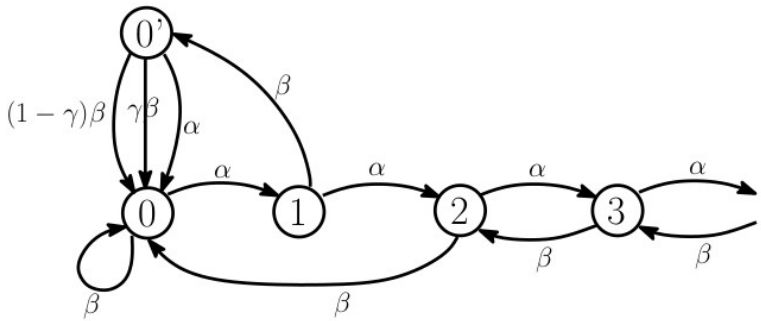
Markov chain or “state machine” models.

For SM

Markov chain model

Markov chain or “state machine” models.

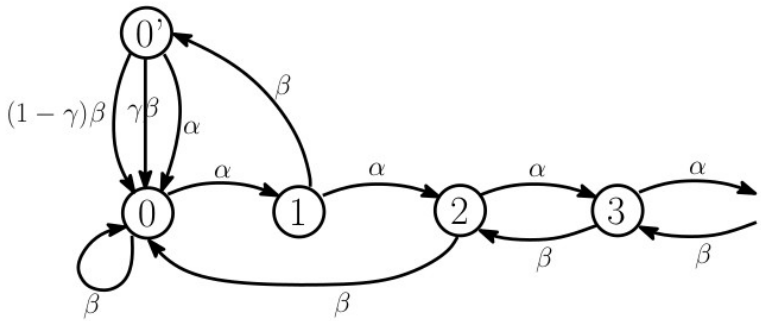
For SM



Markov chain model

Markov chain or “state machine” models.

For SM

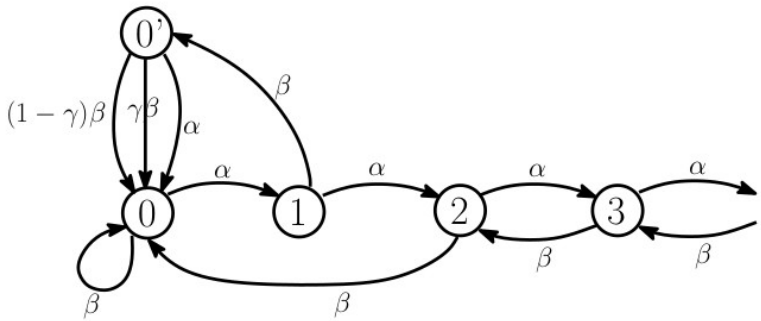


For LSM

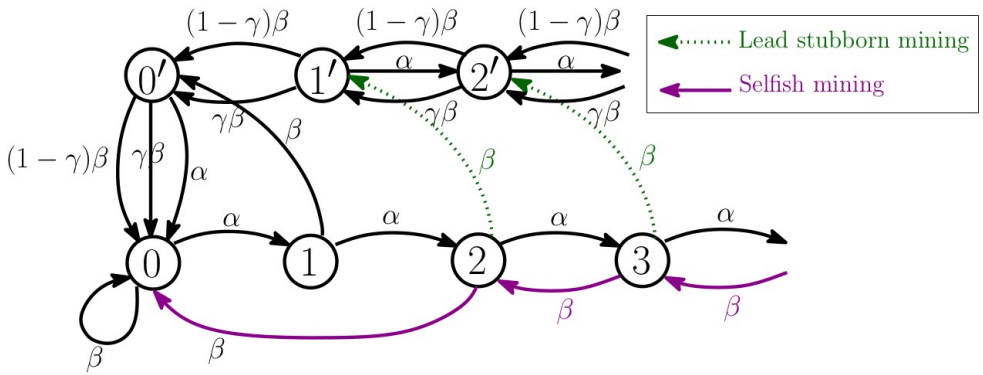
Markov chain model

Markov chain or “state machine” models.

For SM



For LSM



Markov model limitations

Markov model limitations

- The profitability analysis depends in a fundamental way on the duration of the attack cycles.

Markov model limitations

- The profitability analysis depends in a fundamental way on the duration of the attack cycles.
- The stationnary probability of the Markov model computes the probability of being in a given state in a steady regime.

Markov model limitations

- The profitability analysis depends in a fundamental way on the duration of the attack cycles.
- The stationnary probability of the Markov model computes the probability of being in a given state in a steady regime.
- The Markov model offers no insight of the duration of the attack cycles nor on the time to reach a steady regime.

Markov model limitations

- The profitability analysis depends in a fundamental way on the duration of the attack cycles.
- The stationnary probability of the Markov model computes the probability of being in a given state in a steady regime.
- The Markov model offers no insight of the duration of the attack cycles nor on the time to reach a steady regime.
- There is no proper analysis of profitability in the literature.

Profit and Loss

Profit and Loss

- Profit and Loss (PnL) of a business

$$PnL = R - C = \text{Profit} - \text{Cost}$$

Profit and Loss

- Profit and Loss (PnL) of a business

$$PnL = R - C = \text{Profit} - \text{Cost}$$

- What counts is Profit and Loss per unit time (PnLt)

$$PnLt = R_t - C_t = (\text{Profit per unit time}) - (\text{Cost per unit time})$$

Profit and Loss

- Profit and Loss (PnL) of a business

$$PnL = R - C = \text{Profit} - \text{Cost}$$

- What counts is Profit and Loss per unit time (PnLt)

$$PnLt = R_t - C_t = (\text{Profit per unit time}) - (\text{Cost per unit time})$$

- Key observation: C_t for a non-stopping mining operation is the same for honest mining or a deviant strategy.

Profit and Loss

- Profit and Loss (PnL) of a business

$$PnL = R - C = \text{Profit} - \text{Cost}$$

- What counts is Profit and Loss per unit time (PnLt)

$$PnLt = R_t - C_t = (\text{Profit per unit time}) - (\text{Cost per unit time})$$

- Key observation: C_t for a non-stopping mining operation is the same for honest mining or a deviant strategy.

Repetition games

Repetition games

Definition (Repetition games)

A repetition game follows an strategy of repeated cycles.

Repetition games

Definition (Repetition games)

A repetition game follows an strategy of repeated cycles.

Repetition games

Definition (Repetition games)

A repetition game follows an strategy of repeated cycles.
Let R, C and T be random variables resp. of revenue, cost and duration over a cycle.

Repetition games

Definition (Repetition games)

A repetition game follows an strategy of repeated cycles. Let R, C and T be random variables resp. of revenue, cost and duration over a cycle. The game is integrable when

$$\mathbb{E}[T] < +\infty \ .$$

Repetition games

Definition (Repetition games)

A repetition game follows an strategy of repeated cycles. Let R, C and T be random variables resp. of revenue, cost and duration over a cycle. The game is integrable when

$$\mathbb{E}[T] < +\infty \ .$$

Theorem (Profitability of integrable games)

$$\mathbb{E}[PnLt] = \frac{\mathbb{E}[R] - \mathbb{E}[C]}{\mathbb{E}[T]} \ .$$

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle.

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle.

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle. The (R_i) (resp. (C_i), (T_i)) are integrable i.i.d. random variables.

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle. The (R_i) (resp. (C_i), (T_i)) are integrable i.i.d. random variables. Let $PnLt_n$ be the $PnLt$ after n cycles:

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle. The (R_i) (resp. (C_i), (T_i)) are integrable i.i.d. random variables. Let $PnLt_n$ be the $PnLt$ after n cycles:

$$PnLt_n = \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} T_i} = \frac{\frac{1}{n}\sum_{i=1}^{n} R_i - \frac{1}{n}\sum_{i=1}^{n} C_i}{\frac{1}{n}\sum_{i=1}^{n} T_i}.$$

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle. The (R_i) (resp. (C_i), (T_i)) are integrable i.i.d. random variables. Let $PnLt_n$ be the $PnLt$ after n cycles:

$$PnLt_n = \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} T_i} = \frac{\frac{1}{n}\sum_{i=1}^{n} R_i - \frac{1}{n}\sum_{i=1}^{n} C_i}{\frac{1}{n}\sum_{i=1}^{n} T_i}.$$

By the Strong Law of Large Numbers we have that almost surely

Proof of the Profitability Theorem

Proof.

R_i, C_i and T_i values for the i-cycle. The (R_i) (resp. (C_i), (T_i)) are integrable i.i.d. random variables. Let $PnLt_n$ be the $PnLt$ after n cycles:

$$PnLt_n = \frac{\sum_{i=1}^{n} R_i - \sum_{i=1}^{n} C_i}{\sum_{i=1}^{n} T_i} = \frac{\frac{1}{n}\sum_{i=1}^{n} R_i - \frac{1}{n}\sum_{i=1}^{n} C_i}{\frac{1}{n}\sum_{i=1}^{n} T_i} .$$

By the Strong Law of Large Numbers we have that almost surely

$$\mathbb{E}[PnLt] = \lim_{n\to+\infty} PnL_n = \frac{\mathbb{E}[R] - \mathbb{E}[C]}{\mathbb{E}[T]} .$$

□

Revenue ratio

Revenue ratio

To compare integrable games with repetition and equal cost the benchmark is the Revenue Ratio

Revenue ratio

To compare integrable games with repetition and equal cost the benchmark is the Revenue Ratio

Definition (Revenue Ratio)

The revenue ratio of a game with repetition is

$$\Gamma = \frac{\mathbb{E}[R]}{\mathbb{E}[T]}$$

Revenue ratio

To compare integrable games with repetition and equal cost the benchmark is the Revenue Ratio

Definition (Revenue Ratio)

The revenue ratio of a game with repetition is

$$\Gamma = \frac{\mathbb{E}[R]}{\mathbb{E}[T]}$$

Corollary

Let S_1 and S_2 be integrable non-stopping mining strategies. Strategy S_1 is more profitable than strategy S_2 if and only if $\Gamma(S_1) \geq \Gamma(S_2)$.

Notations

Notations

- Two group of miners with relative hashrates

$$0 < q < 1/2 < p < 1 \ , \quad p + q = 1$$

Notations

- Two group of miners with relative hashrates

$$0 < q < 1/2 < p < 1 \ , \quad p + q = 1$$

- The block validation times T' and T are exponentially distributed random variables with resp. parameters α' and α.

Notations

- Two group of miners with relative hashrates

$$0 < q < 1/2 < p < 1 \,, \quad p + q = 1$$

- The block validation times T' and T are exponentially distributed random variables with resp. parameters α' and α.
- The probabilities of success of each group are

$$\mathbb{P}[T < T'] = p \,, \quad \mathbb{P}[T' < T] = q \,.$$

Notations

- Two group of miners with relative hashrates

$$0 < q < 1/2 < p < 1 \,, \quad p + q = 1$$

- The block validation times T' and T are exponentially distributed random variables with resp. parameters α' and α.
- The probabilities of success of each group are

$$\mathbb{P}[T < T'] = p \,, \quad \mathbb{P}[T' < T] = q \,.$$

- $N'(t)$ and $N(t)$ numbers of validated blocks at time t are Poisson processes with resp. parameters α' and α.

Notations

- Two group of miners with relative hashrates

$$0 < q < 1/2 < p < 1 \ , \quad p + q = 1$$

- The block validation times T' and T are exponentially distributed random variables with resp. parameters α' and α.
- The probabilities of success of each group are

$$\mathbb{P}[T < T'] = p \ , \quad \mathbb{P}[T' < T] = q \ .$$

- $N'(t)$ and $N(t)$ numbers of validated blocks at time t are Poisson processes with resp. parameters α' and α.

$$\mathbb{P}[N(t) = n] = \frac{(\alpha t)^n}{n!} e^{-\alpha t} \ , \quad \mathbb{P}[N'(t) = n] = \frac{(\alpha' t)^n}{n!} e^{-\alpha' t}$$

Simple example: Honest strategy

Simple example: Honest strategy

- Duration of the cycle for the honest strategyis the stopping time:

$$\tau_H = T' \wedge T$$

Simple example: Honest strategy

- Duration of the cycle for the honest strategyis the stopping time:

$$\tau_H = T' \wedge T$$

- We compute $\mathbb{E}[\tau_H] = \tau_0 = \frac{1}{\alpha+\alpha'}$.

Simple example: Honest strategy

- Duration of the cycle for the honest strategyis the stopping time:

$$\tau_H = T' \wedge T$$

- We compute $\mathbb{E}[\tau_H] = \tau_0 = \frac{1}{\alpha+\alpha'}$.
- Therefore, if $b > 0$ is the block reward, $\mathbb{E}[R] = p.0 + q.b = qb$

Simple example: Honest strategy

- Duration of the cycle for the honest strategyis the stopping time:

$$\tau_H = T' \wedge T$$

- We compute $\mathbb{E}[\tau_H] = \tau_0 = \frac{1}{\alpha+\alpha'}$.
- Therefore, if $b > 0$ is the block reward, $\mathbb{E}[R] = p.0 + q.b = qb$
- The revenue ratio of the honest strategy is

$$\Gamma(H) = \frac{qb}{\tau_0}$$

Selfish mining

Selfish mining

- The combinatorics of each cycle c of attack gives a revenue $R(c) = N(c)b$ and has a duration $T(c)$.

Selfish mining

- The combinatorics of each cycle c of attack gives a revenue $R(c) = N(c)b$ and has a duration $T(c)$.
- If the probability of each cycle is $p(c)$ then

$$\mathbb{E}[R] = \sum_c p(c)R(c)$$

$$\mathbb{E}[T] = \sum_c p(c)T(c)$$

Selfish mining

- The combinatorics of each cycle c of attack gives a revenue $R(c) = N(c)b$ and has a duration $T(c)$.
- If the probability of each cycle is $p(c)$ then

$$\mathbb{E}[R] = \sum_c p(c)R(c)$$

$$\mathbb{E}[T] = \sum_c p(c)T(c)$$

- The combinatorics is involved and the computation of each $T(c)$ involves conditional probabilities and iterated integrals... too complex! We need new tools and ideas...

Martingale Technique

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s) \ .$$

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s) \ .$$

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s) \ .$$

A stopping time τ is a random variable taking values in $\mathbb{R}_+$ only depending on $(N(t))_{t\leq\tau}$.

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s) \ .$$

A stopping time τ is a random variable taking values in $\mathbb{R}_+$ only depending on $(N(t))_{t\leq\tau}$.

Theorem (Doob's Stopping Time Theorem)

Let $(N(t))_{t\in\mathbb{R}_+}$ be a martingale and τ be a bounded stopping time.

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s) \ .$$

A stopping time τ is a random variable taking values in $\mathbb{R}_+$ only depending on $(N(t))_{t\leq\tau}$.

Theorem (Doob's Stopping Time Theorem)

Let $(N(t))_{t\in\mathbb{R}_+}$ be a martingale and τ be a bounded stopping time.

Martingale Technique

Definition (Martingale and Stopping Time)

A martingale is a stochastic process $(N(t))_{t\in\mathbb{R}_+}$ with an adapted decreasing filtration $(\Sigma_t)_{t\in\mathbb{R}_+}$ such that $N(t)$ is Σ_t-measurable, and for $t > s$

$$\mathbb{E}[N(t)|\Sigma_s] = N(s)\ .$$

A stopping time τ is a random variable taking values in $\mathbb{R}_+$ only depending on $(N(t))_{t\leq\tau}$.

Theorem (Doob's Stopping Time Theorem)

Let $(N(t))_{t\in\mathbb{R}_+}$ be a martingale and τ be a bounded stopping time.Then we have $\mathbb{E}[N(\tau)] = N(0)$.

Poisson Games

Poisson Games

Theorem (Poisson Races)

N and N' two independent Poisson processes with parameters α' and α with $\alpha' < \alpha$ and $N(0) = N'(0) = 0$.

Poisson Games

Theorem (Poisson Races)

N and N' two independent Poisson processes with parameters α' and α with $\alpha' < \alpha$ and $N(0) = N'(0) = 0$.

Poisson Games

Theorem (Poisson Races)

N and N' two independent Poisson processes with parameters α' and α with $\alpha' < \alpha$ and $N(0) = N'(0) = 0$. Then, the stopping time

$$\tau = \inf\{t > 0; N(t) = N'(t) + 1\}$$

is finite a.s. and integrable.

Poisson Games

Theorem (Poisson Races)

N and N' two independent Poisson processes with parameters α' and α with $\alpha' < \alpha$ and $N(0) = N'(0) = 0$. Then, the stopping time

$$\tau = \inf\{t > 0; N(t) = N'(t) + 1\}$$

is finite a.s. and integrable. Moreover, we have

$$\mathbb{E}[\tau] = \frac{1}{\alpha - \alpha'}, \quad \mathbb{E}[N'(\tau)] = \frac{\alpha'}{\alpha - \alpha'}, \quad \mathbb{E}[N(\tau)] = \frac{\alpha}{\alpha - \alpha'}.$$

Proof

Proof.

Assume τ bounded

Proof

Proof.

Assume τ bounded

Proof

Proof.

Assume τ bounded (otherwise truncate $\tau \wedge t_0$ and make $t_0 \to +\infty$).

Proof

Proof.

Assume τ bounded (otherwise truncate $\tau \wedge t_0$ and make $t_0 \to +\infty$). The compensated Poisson processes $N(t) - \alpha t$ and $N'(t) - \alpha' t$ are martingales.

Proof

Proof.

Assume τ bounded (otherwise truncate $\tau \wedge t_0$ and make $t_0 \to +\infty$). The compensated Poisson processes $N(t) - \alpha t$ and $N'(t) - \alpha' t$ are martingales. Doob's Stopping Time Theorem gives

Proof

Proof.

Assume τ bounded (otherwise truncate $\tau \wedge t_0$ and make $t_0 \to +\infty$). The compensated Poisson processes $N(t) - \alpha t$ and $N'(t) - \alpha' t$ are martingales. Doob's Stopping Time Theorem gives

$$\alpha \mathbb{E}[\tau] = \mathbb{E}[N(\tau)] = \mathbb{E}[N'(\tau)] + 1 = \alpha' \mathbb{E}[\tau] + 1$$

Proof

Proof.

Assume τ bounded (otherwise truncate $\tau \wedge t_0$ and make $t_0 \to +\infty$). The compensated Poisson processes $N(t) - \alpha t$ and $N'(t) - \alpha' t$ are martingales. Doob's Stopping Time Theorem gives

$$\alpha \mathbb{E}[\tau] = \mathbb{E}[N(\tau)] = \mathbb{E}[N'(\tau)] + 1 = \alpha' \mathbb{E}[\tau] + 1$$

from where we get

$$\mathbb{E}[\tau] = \frac{1}{\alpha - \alpha'}$$

and the two other formulas. □

Selfish Mining Stopping Time

Selfish Mining Stopping Time

- Let $T_1, T_2, \ldots$ and $T'_1, T'_2, \ldots$ interblock validation times.

Selfish Mining Stopping Time

- Let $T_1, T_2, \ldots$ and $T'_1, T'_2, \ldots$ interblock validation times.
- $S_n = T_1 + \ldots + T_n$, $S'_n = T'_1 + \ldots + T'_n$.

Selfish Mining Stopping Time

- Let $T_1, T_2, \ldots$ and $T'_1, T'_2, \ldots$ interblock validation times.
- $S_n = T_1 + \ldots + T_n$, $S'_n = T'_1 + \ldots + T'_n$.

Lemma (Duration of attack cycles)

The duration of attack cycles for selfish mining is given by the stopping time

$$\tau_{SM} = \inf\{t \geq T_1; N(t) = N'(t) - 1 + 2 \cdot \mathbf{1}_{T_1 < T'_1} + 2 \cdot \mathbf{1}_{T'_1 < T_1 < S_2 < S'_2}\}$$

Selfish Mining Revenue Ratio

Selfish Mining Revenue Ratio

Theorem (SM Revenue Ratio)

τ_{SM} and $R(\tau_{SM,\gamma})$ are integrable and

Selfish Mining Revenue Ratio

Theorem (SM Revenue Ratio)

τ_{SM} and $R(\tau_{SM,\gamma})$ are integrable and

Selfish Mining Revenue Ratio

Theorem (SM Revenue Ratio)

τ_{SM} and $R(\tau_{SM,\gamma})$ are integrable and

$$\mathbb{E}[R(\tau_{SM})] = \frac{(1+pq)(p-q)+pq}{p-q}\, qb - (1-\gamma)p^2 q\, b$$

$$\mathbb{E}[\tau_{SM}] = \frac{(1+pq)(p-q)+pq}{p-q}\, \tau_0$$

Selfish Mining Revenue Ratio

Theorem (SM Revenue Ratio)

τ_{SM} and $R(\tau_{SM,\gamma})$ are integrable and

$$\mathbb{E}[R(\tau_{SM})] = \frac{(1+pq)(p-q)+pq}{p-q}\,qb - (1-\gamma)p^2 q\,b$$

$$\mathbb{E}[\tau_{SM}] = \frac{(1+pq)(p-q)+pq}{p-q}\,\tau_0$$

and

$$\Gamma(SM) = \frac{qb}{\tau_0} - (1-\gamma)\frac{p^2 q(p-q)b}{((1+pq)(p-q)+pq)\,\tau_0} \leq P(H)$$

The Theorem from beyond

The Theorem from beyond

The following theorem shows that in a stable regime without difficulty adjustments the Bitcoin protocol is stable with respect to block withholding strategies.

The Theorem from beyond

The following theorem shows that in a stable regime without difficulty adjustments the Bitcoin protocol is stable with respect to block withholding strategies.

Theorem (Optimality of Honest Mining)

For any block withholding strategy S we have

$$\Gamma(S) \leq \Gamma(H) = \frac{qb}{\tau_0}$$

Lead-Stubborn Mining Stopping Time

Lead-Stubborn Mining Stopping Time

Lemma (Duration of attack cycles)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{LSM}

$$\xi_{LSM} = \tau + (T_{N(\tau)+1} \wedge T'_{N(\tau)+1}) \cdot \mathbf{1}_{T'_1 \leq T_1}$$

Lead-Stubborn Mining Stopping Time

Lemma (Duration of attack cycles)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{LSM}

$$\xi_{LSM} = \tau + (T_{N(\tau)+1} \wedge T'_{N(\tau)+1}) \cdot \mathbf{1}_{T'_1 \leq T_1}$$

Lead-Stubborn Mining Stopping Time

Lemma (Duration of attack cycles)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{LSM}

$$\xi_{LSM} = \tau + (T_{N(\tau)+1} \wedge T'_{N(\tau)+1}) \cdot \mathbf{1}_{T'_1 \leq T_1}$$

with

$$\tau = \inf\{t \geq T_1; N(t) = N'(t) + \mathbf{1}_{T_1 < T'_1}\}$$

Lead-Stubborn Mining Revenue Ratio

Lead-Stubborn Mining Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{LSM} and $R(\xi_{LSM})$ are integrable and

Lead-Stubborn Mining Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{LSM} and $R(\xi_{LSM})$ are integrable and

Lead-Stubborn Mining Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{LSM} and $R(\xi_{LSM})$ are integrable and

$$\mathbb{E}[R(\xi_{LSM})] = \left(\frac{p + pq - q^2}{p - q}\right) qb - pq\, f\, b$$

$$\mathbb{E}[\xi_{LSM}] = \frac{p + pq - q^2}{p - q}\, \tau_0$$

Lead-Stubborn Mining Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{LSM} and $R(\xi_{LSM})$ are integrable and

$$\mathbb{E}[R(\xi_{LSM})] = \left(\frac{p + pq - q^2}{p - q}\right) qb - pq\, f\, b$$

$$\mathbb{E}[\xi_{LSM}] = \frac{p + pq - q^2}{p - q}\, \tau_0$$

with $f = \frac{1-\gamma}{\gamma} \cdot \left(1 - \frac{1}{2q}(1 - \sqrt{1 - 4(1-\gamma)pq})\right)$

Lead-Stubborn Mining Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{LSM} *and* $R(\xi_{LSM})$ *are integrable and*

$$\mathbb{E}[R(\xi_{LSM})] = \left(\frac{p + pq - q^2}{p - q}\right) qb - pq\, f\, b$$

$$\mathbb{E}[\xi_{LSM}] = \frac{p + pq - q^2}{p - q}\, \tau_0$$

with $f = \frac{1-\gamma}{\gamma} \cdot \left(1 - \frac{1}{2q}(1 - \sqrt{1 - 4(1-\gamma)pq})\right)$ *and*

$$\Gamma(\xi_{LSM}) = \frac{qb}{\tau_0} - \frac{(p-q)pq\, f}{p + q(p-q)} \frac{b}{\tau_0}$$

Equal Fork Stubborn Stopping Time

Equal Fork Stubborn Stopping Time

Lemma (Equal Fork Stubborn Stopping Time)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{EFSM}

$$\xi_{EFSM} = \inf\{t \geq 0; N(t) = N'(t) + 1\}$$

Equal Fork Stubborn Stopping Time

Lemma (Equal Fork Stubborn Stopping Time)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{EFSM}

$$\xi_{EFSM} = \inf\{t \geq 0; N(t) = N'(t) + 1\}$$

Equal Fork Stubborn Stopping Time

Lemma (Equal Fork Stubborn Stopping Time)

The duration of attack cycles for Lead-Stubborn mining is given by the stopping time ξ_{EFSM}

$$\xi_{EFSM} = \inf\{t \geq 0; N(t) = N'(t) + 1\}$$

Note: Same stopping time that for Poisson Games.

Equal Fork Stubborn Revenue Ratio

Equal Fork Stubborn Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{EFSM} and $R(\xi_{EFSM})$ are integrable and

Equal Fork Stubborn Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{EFSM} and $R(\xi_{EFSM})$ are integrable and

Equal Fork Stubborn Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{EFSM} and $R(\xi_{EFSM})$ are integrable and

$$\mathbb{E}[R(\xi_{EFSM})] = \frac{q}{p-q}b - gb$$

$$\mathbb{E}[\xi_{EFSM}] = \frac{\tau_0}{p-q}$$

Equal Fork Stubborn Revenue Ratio

Theorem (LSM Revenue Ratio)

ξ_{EFSM} *and* $R(\xi_{EFSM})$ *are integrable and*

$$\mathbb{E}[R(\xi_{EFSM})] = \frac{q}{p-q}b - gb$$

$$\mathbb{E}[\xi_{EFSM}] = \frac{\tau_0}{p-q}$$

with $g = \frac{1-\gamma}{\gamma}\left(1 - \frac{1}{2(1-\gamma)q}\left(1 - \sqrt{1-4(1-\gamma)pq}\right)\right)$ *and*

$$\Gamma(\xi_{EFSM}) = \frac{qb}{\tau_0} - (p-q)g\frac{b}{\tau_0}$$

A-Trailing Mining Revenue Ratio

A-Trailing Mining Revenue Ratio

Theorem (TSM_A Revenue Ratio)

$$\Gamma(\xi_{TSM_A}) = \frac{b}{\tau_0}\cdot$$

$$\frac{q + \frac{(1-\gamma)pq(p-q)}{(p+pq-q^2)[A+1]}\left(\left([A-1] + \frac{1}{p}\frac{P_A(\lambda)}{[A+1]}\right)\lambda^2 - \frac{2}{\sqrt{1-4(1-\gamma)pq}+p-q}\right)}{1 + \frac{(1-\gamma)pq}{p+pq-q^2}(A+1)\left(\frac{[2]}{[A+1]} - \frac{2}{A+1}\right)}$$

$$\textit{where } [n] = \frac{1-\lambda^n}{1-\lambda}, \textit{ and } P_A(\lambda) = \frac{1-A\lambda^{A-1}+A\lambda^{A+1}-\lambda^{2A}}{(1-\lambda)^3}$$

Profitability after a difficulty adjustment

Profitability after a difficulty adjustment

- A block withholding attack that aims to orphan honest mined blocks slows down the network.

Profitability after a difficulty adjustment

- A block withholding attack that aims to orphan honest mined blocks slows down the network.
- After $n_0 = 2016$ (sic, 2015) blocks we have the difficulty divided by a factor $\delta > 1$. We need to re-evaluate the profitability.

Profitability after a difficulty adjustment

- A block withholding attack that aims to orphan honest mined blocks slows down the network.
- After $n_0 = 2016$ (sic, 2015) blocks we have the difficulty divided by a factor $\delta > 1$. We need to re-evaluate the profitability.
- For an attack cycle, $\mathbb{E}[R]$ is unchanged but $\mathbb{E}[T]$ is changed to $\delta^{-1}\mathbb{E}[T]$ and the Revenue Ratio is multiplied by δ.

Profitability after a difficulty adjustment

- A block withholding attack that aims to orphan honest mined blocks slows down the network.
- After $n_0 = 2016$ (sic, 2015) blocks we have the difficulty divided by a factor $\delta > 1$. We need to re-evaluate the profitability.
- For an attack cycle, $\mathbb{E}[R]$ is unchanged but $\mathbb{E}[T]$ is changed to $\delta^{-1}\mathbb{E}[T]$ and the Revenue Ratio is multiplied by δ.
- The apparent hashrate of the attackers becomes $\tilde{q} > q$.

Profitability after a difficulty adjustment

- A block withholding attack that aims to orphan honest mined blocks slows down the network.
- After $n_0 = 2016$ (sic, 2015) blocks we have the difficulty divided by a factor $\delta > 1$. We need to re-evaluate the profitability.
- For an attack cycle, $\mathbb{E}[R]$ is unchanged but $\mathbb{E}[T]$ is changed to $\delta^{-1}\mathbb{E}[T]$ and the Revenue Ratio is multiplied by δ.
- The apparent hashrate of the attackers becomes $\tilde{q} > q$.
- Depending on q and γ selfish mining strategies can become profitable.

BIP coutermeasure against block withholding

BIP coutermeasure against block withholding

- The problem comes from the difficulty adjustment formula that ignores orphan blocks and therefore sub-estimates the total hashrate of the network.

BIP coutermeasure against block withholding

- The problem comes from the difficulty adjustment formula that ignores orphan blocks and therefore sub-estimates the total hashrate of the network.
- It would be enough to include in honest validated blocks "proof of orphans".

BIP coutermeasure against block withholding

- The problem comes from the difficulty adjustment formula that ignores orphan blocks and therefore sub-estimates the total hashrate of the network.
- It would be enough to include in honest validated blocks "proof of orphans".
- This could be cone by propagating orphan headers through the network and include the data in validated blocks.

BIP coutermeasure against block withholding

- The problem comes from the difficulty adjustment formula that ignores orphan blocks and therefore sub-estimates the total hashrate of the network.
- It would be enough to include in honest validated blocks "proof of orphans".
- This could be cone by propagating orphan headers through the network and include the data in validated blocks.
- Then the new adjustment formula will take the ratio of the difference of first and last timestamp in a n_0 period and divide it by $n_0 + n'$ where n' is the number of orphan blocks.

Apparent hashrates

Apparent hashrates

Theorem (Apparent hashrates)

$$\tilde{q}(SM) = \frac{((1+pq)(p-q)+pq)q-(1-\gamma)p^2q(p-q)}{p^2q+p-q}$$

$$\tilde{q}(LSM) = q \cdot \frac{p+pq-q^2}{p+pq-q} - \frac{pq(p-q)f(\gamma)}{p+pq-q}$$

$$\tilde{q}(EFSM) = \frac{q}{p} - \left(1-\frac{q}{p}\right) f(\gamma)$$

$$\tilde{q}(TSM_A) = \frac{q + \frac{(1-\gamma)pq(p-q)}{(p+pq-q^2)[A+1]} \left(\left([A-1] + \frac{1}{p}\frac{P_A(\lambda)}{[A+1]} \right) \lambda^2 - \frac{2}{\sqrt{1-4(1-\gamma)pq-}} \right.}{\frac{p+pq-q}{p+pq-q^2} + \frac{(1-\gamma)pq}{p+pq-q^2}(A+\lambda)\left(\frac{1}{[A+1]} - \frac{1}{A+\lambda}\right)}$$

Expected difficulty adjustments

Expected difficulty adjustments

Theorem (Expected difficulty adjustments)

$$\mathbb{E}[\delta(SM)] = \frac{p - q + pq(p-q) + pq}{p^2 q + p - q}$$

$$\mathbb{E}[\delta(LSM)] = \frac{p + pq - q^2}{p + pq - q}$$

$$\mathbb{E}[\delta(EFSM)] = \frac{1}{p}$$

$$\mathbb{E}[\delta(TSM_A)] = \frac{\frac{p+pq-q^2}{p-q} + (A+1) \cdot \frac{(1-\gamma)pq}{p-q} \cdot \left(\frac{1-\lambda^2}{1-\lambda^{A+1}} - \frac{2}{A+1} \right)}{\frac{pq+p-q}{p-q} + \frac{(1-\gamma)pq}{p-q}((Ap+q)\frac{1-\lambda^2}{1-\lambda^{A+1}} - 1)}$$

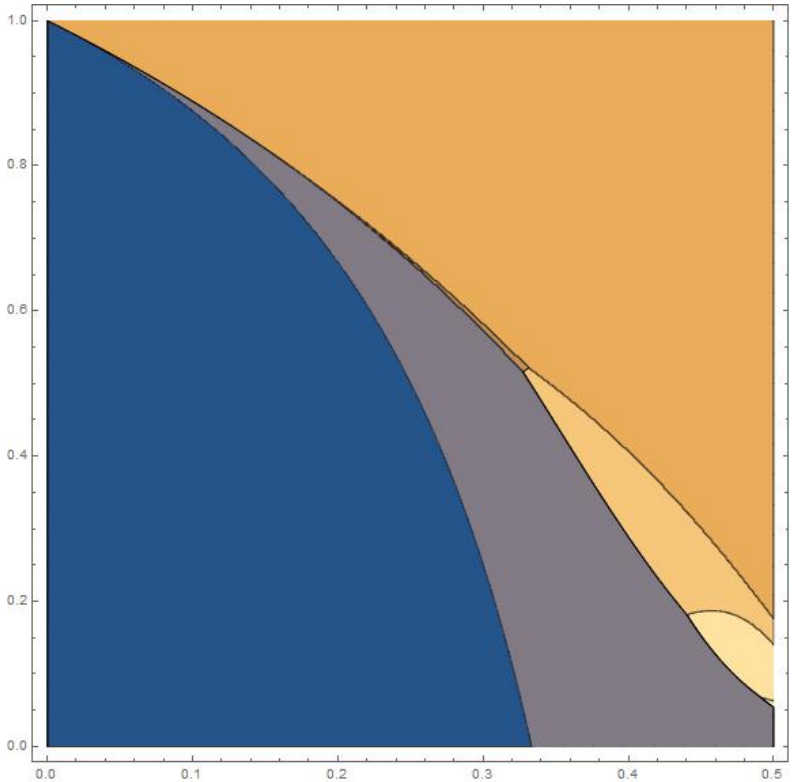
Comparisons after a difficulty adjustment

Comparisons after a difficulty adjustment

For different values of q and γ we can compare the different strategies after a difficulty adjustment:

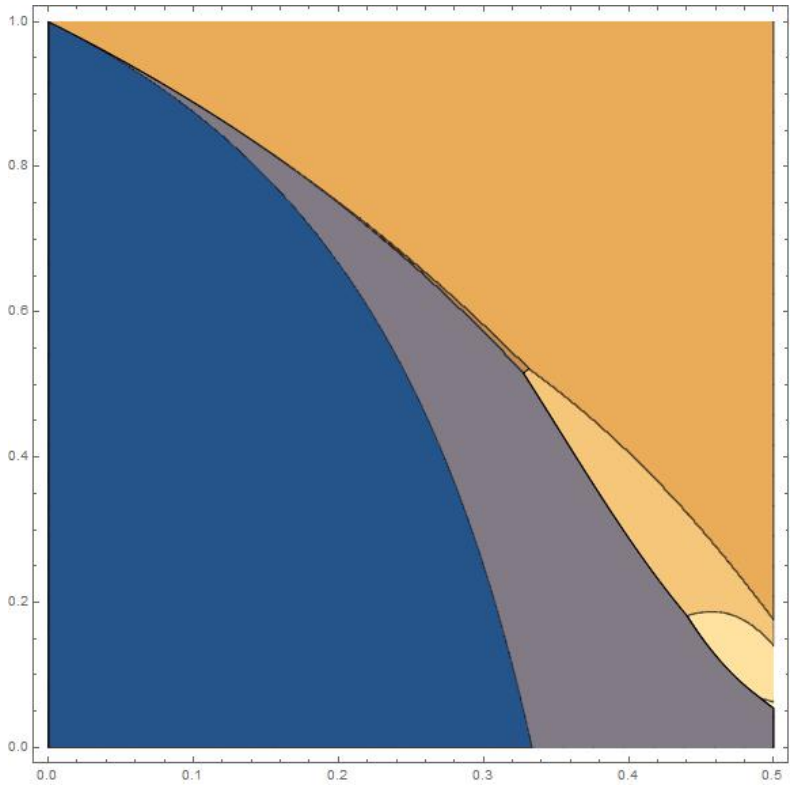
Comparisons after a difficulty adjustment

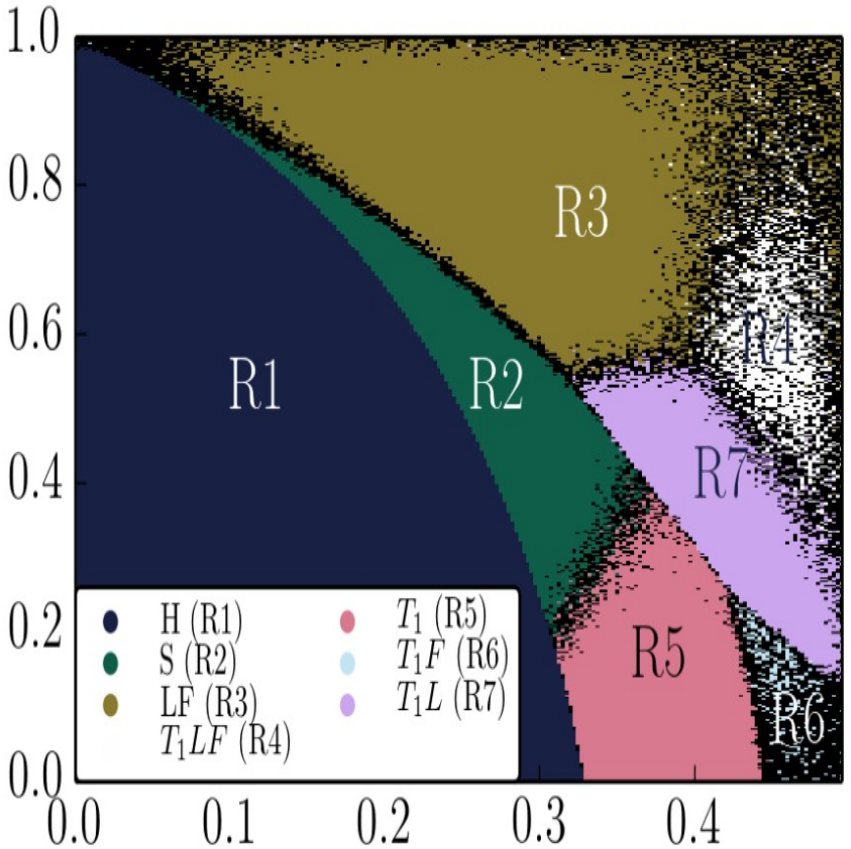
For different values of q and γ we can compare the different strategies after a difficulty adjustment:



Comparisons with NKMS2016

Comparisons with NKMS2016





Thank you for your attention!