# Jeu du minage, et sécurité du protocole Bitcoin

par Cyril Grunspan

*07/12/2022*

## Remerciements.

Institut des Actuaires

**Remerciements.**

Institut des Actuaires

Samuel Cywie

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

Sandrine Lemery

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

Sandrine Lemery

Florence Picard

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

Sandrine Lemery

Florence Picard

Olivier Lopez

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

Sandrine Lemery

Florence Picard

Olivier Lopez

Alexis Collomb

**Remerciements.**

Institut des Actuaires

Samuel Cywie

Michel Fromenteau

Sandrine Lemery

Florence Picard

Olivier Lopez

Alexis Collomb

Ricardo Pérez-Marco

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

B3i « the blockchain insurance industry initiative » à la suite de R3 CEV

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\,N_t - c\,t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

B3i « the blockchain insurance industry initiative » à la suite de R3 CEV

Investissements de CNP Assurance (Stratumn), AXA venture (Blockstream)

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

B3i « the blockchain insurance industry initiative » à la suite de R3 CEV

Investissements de CNP Assurance (Stratumn), AXA venture (Blockstream)

Lloyds, assureur de certaines activités de Ledger

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

B3i « the blockchain insurance industry initiative » à la suite de R3 CEV

Investissements de CNP Assurance (Stratumn), AXA venture (Blockstream)

Lloyds, assureur de certaines activités de Ledger

Nexus Mutual, assureur de la DEFI

*On the profitability of selfish blockchain mining under consideration of ruin*, Hansjörg Albrecher, Pierre-Olivier Goffard, Operations Research, 2021

Processus $R_t = u + b\, N_t - c\, t$. Calcul de $\mathbb{P}[\tau_u \leqslant t]$, $\tau_u = \mathrm{Inf}\,\{t \geqslant 0\,;\, R_t = 0\}$. Problème dual d'un problème classique de ruine d'un assureur

Questions de minage de Bitcoin du domaine de l'actuariat

B3i « the blockchain insurance industry initiative » à la suite de R3 CEV

Investissements de CNP Assurance (Stratumn), AXA venture (Blockstream)

Lloyds, assureur de certaines activités de Ledger

Nexus Mutual, assureur de la DEFI

« Contrats discrets » (DLC) de Tadge Dryja, plateformes assurance paramétrique

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Traitement probabiliste et combinatoire

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Traitement probabiliste et combinatoire

Stratégies de minage déviantes rentables sur le long terme

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Traitement probabiliste et combinatoire

Stratégies de minage déviantes rentables sur le long terme

Non prise en compte de la production blocs orphelins dans la « DAA » actuelle

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Traitement probabiliste et combinatoire

Stratégies de minage déviantes rentables sur le long terme

Non prise en compte de la production blocs orphelins dans la « DAA » actuelle

**Prouver qu'un Bitcoin modifié serait dépourvu de cette anomalie**

Articles avec Ricardo Pérez-Marco, *Double Spend Races* IJTAF 2018

*On Profitability of Selfish Mining*, arxiv 2018

Notion de cycle d'attaque de stratégie de minage

Traitement probabiliste et combinatoire

Stratégies de minage déviantes rentables sur le long terme

Non prise en compte de la production blocs orphelins dans la « DAA » actuelle

**Prouver qu'un Bitcoin modifié serait dépourvu de cette anomalie**

**Trouver une méthode simple donnant le seuil en terme de puissance de hachage relative au-delà un mineur n'a plus intérêt à miner honnêtement dans Bitcoin actuel**

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Première « cryptomonnaie » décentralisée

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Première « cryptomonnaie » décentralisée

Décentralisation maximale : n'importe qui peut devenir validateur de blocs sans besoin de se faire connaître ni de demander de permission

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Première « cryptomonnaie » décentralisée

Décentralisation maximale : n'importe qui peut devenir validateur de blocs sans besoin de se faire connaître ni de demander de permission

Théorèmes prouvant qu'il n'y a pas de double-dépense possible

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Première « cryptomonnaie » décentralisée

Décentralisation maximale : n'importe qui peut devenir validateur de blocs sans besoin de se faire connaître ni de demander de permission

Théorèmes prouvant qu'il n'y a pas de double-dépense possible

D'autres cryptomonnaies existent basées sur le PoS

Deux découvertes majeures dans Bitcoin, technologiques et théoriques

- Première monnaie programmable
- Première solution au problème des généraux byzantins dans un **système ouvert**

Première « cryptomonnaie » décentralisée

Décentralisation maximale : n'importe qui peut devenir validateur de blocs sans besoin de se faire connaître ni de demander de permission

Théorèmes prouvant qu'il n'y a pas de double-dépense possible

D'autres cryptomonnaies existent basées sur le PoS

Moins étudiées, en pratique plus centralisées, potentiellement sensibles à l'attaque **Long Range Attack**.

1. **Réseau ouvert**
    - Adresse Bitcoin

1. **Réseau ouvert**
    - Adresse Bitcoin
2. Trois bases de données
    - UTXO-set

1. **Réseau ouvert**
    - Adresse Bitcoin
2. Trois bases de données
    - UTXO-set
    - Blockchain

1. **Réseau ouvert**
    - Adresse Bitcoin
2. Trois bases de données
    - UTXO-set
    - Blockchain
    - Mempool

1. **Réseau ouvert**
    - Adresse Bitcoin
2. Trois bases de données
    - UTXO-set
    - Blockchain
    - Mempool
3. Noeuds lourds, légers
    - Mineurs

1. **Réseau ouvert**
    - Adresse Bitcoin
2. Trois bases de données
    - UTXO-set
    - Blockchain
    - Mempool
3. Noeuds lourds, légers
    - Mineurs
    - Wallets

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max
4. Mise à jour tous les $n_0 = 2016$ blocs de $\Delta$ : $\Delta_i = \frac{n_0 \cdot \tau_0}{T} \times \Delta_{i-1}$, évaluation de $T$ grace aux timestamps

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max
4. Mise à jour tous les $n_0 = 2016$ blocs de $\Delta$ : $\Delta_i = \frac{n_0 \cdot \tau_0}{T} \times \Delta_{i-1}$, évaluation de $T$ grace aux timestamps
5. But : un bloc découvert toutes les $\tau_0 = 10$ minutes.

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max
4. Mise à jour tous les $n_0 = 2016$ blocs de $\Delta$ : $\Delta_i = \frac{n_0 \cdot \tau_0}{T} \times \Delta_{i-1}$, évaluation de $T$ grace aux timestamps
5. But : un bloc découvert toutes les $\tau_0 = 10$ minutes.
6. Chaque bloc rapporte 1 coinbase (6.25฿) + frais de tx

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \mathrm{Target}$ et $\mathrm{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max
4. Mise à jour tous les $n_0 = 2016$ blocs de $\Delta$ : $\Delta_i = \frac{n_0 \cdot \tau_0}{T} \times \Delta_{i-1}$, évaluation de $T$ grace aux timestamps
5. But : un bloc découvert toutes les $\tau_0 = 10$ minutes.
6. Chaque bloc rapporte 1 coinbase (6.25฿) + frais de tx
7. Probabilité de réussite de double-dépense faible

1. En-tête d'un bloc : timestamp, version, MerkleRoot, version, PreviousBlock-Hash, Target, Nonce
2. Bloc $\mathcal{B}$ valide si $h(\mathcal{B}) < \text{Target}$ et $\text{Target} = \frac{1}{\Delta}$. $\Delta =$ difficulté de minage
3. Blockchain officielle : $(\mathcal{B}_i)_{0 \leqslant i \leqslant n}$ tq $\sum_{i \geqslant 0} \Delta_i$ max
4. Mise à jour tous les $n_0 = 2016$ blocs de $\Delta$ : $\Delta_i = \frac{n_0 \cdot \tau_0}{T} \times \Delta_{i-1}$, évaluation de $T$ grace aux timestamps
5. But : un bloc découvert toutes les $\tau_0 = 10$ minutes.
6. Chaque bloc rapporte 1 coinbase (6.25฿) + frais de tx
7. Probabilité de réussite de double-dépense faible
8. Stratégie honnête : toujours miner sur le dernier bloc de la blockchain officielle

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Processus de comptage : $(\boldsymbol{N}'(t), \boldsymbol{N}(t))$ des blocs minés par (Alice,Bob).

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Processus de comptage : $(\boldsymbol{N}'(t), \boldsymbol{N}(t))$ des blocs minés par (Alice,Bob).

# hashs calculés en une seconde par (Alice, Bob) : $(h', h)$

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Processus de comptage : $(\boldsymbol{N}'(t), \boldsymbol{N}(t))$ des blocs minés par (Alice,Bob).

# hashs calculés en une seconde par (Alice, Bob) : $(h', h)$

Difficulté $\Delta$ : # hashs nécessaires pour trouver une preuve de travail (moyenne)

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Processus de comptage : $(\boldsymbol{N}'(t), \boldsymbol{N}(t))$ des blocs minés par (Alice,Bob).

# hashs calculés en une seconde par (Alice, Bob) : $(h', h)$

Difficulté $\Delta$ : # hashs nécessaires pour trouver une preuve de travail (moyenne)

Durée $(\boldsymbol{T}', \boldsymbol{T})$ mise par (Alice, Bob) avant de trouver une solution

Un groupe d'attaquants (Alice) et des honnêtes mineurs (Bob)

Processus de comptage : $(\boldsymbol{N}'(t), \boldsymbol{N}(t))$ des blocs minés par (Alice,Bob).

# hashs calculés en une seconde par (Alice, Bob) : $(h', h)$

Difficulté $\Delta$ : # hashs nécessaires pour trouver une preuve de travail (moyenne)

Durée $(\boldsymbol{T}', \boldsymbol{T})$ mise par (Alice, Bob) avant de trouver une solution

Variables $\boldsymbol{T}, \boldsymbol{T}'$ suivent des lois exponentielles de paramètres $\alpha, \alpha'$

$$\alpha = \frac{h}{\Delta}$$
$$\alpha' = \frac{h'}{\Delta}$$

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha + \alpha'} = \frac{h}{h + h'}$

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha+\alpha'} = \frac{h}{h+h'}$

Puissance de hachage relative de (Alice, Bob) : $\left(\frac{\alpha}{\alpha+\alpha'}, \frac{\alpha'}{\alpha+\alpha'}\right) =: (p, q)$

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha+\alpha'} = \frac{h}{h+h'}$

Puissance de hachage relative de (Alice, Bob) : $\left(\frac{\alpha}{\alpha+\alpha'}, \frac{\alpha'}{\alpha+\alpha'}\right) =: (p, q)$

Hypothèse : $(p, q)$ reste constant et $q < p$.

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha+\alpha'} = \frac{h}{h+h'}$

Puissance de hachage relative de (Alice, Bob) : $\left(\frac{\alpha}{\alpha+\alpha'}, \frac{\alpha'}{\alpha+\alpha'}\right) =: (p, q)$

Hypothèse : $(p, q)$ reste constant et $q < p$.

Variable aléatoire $\mathrm{Inf}\,(\boldsymbol{T}, \boldsymbol{T}')$ = loi exponentielle paramètre $\alpha + \alpha'$

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha+\alpha'} = \frac{h}{h+h'}$

Puissance de hachage relative de (Alice, Bob) : $\left(\frac{\alpha}{\alpha+\alpha'}, \frac{\alpha'}{\alpha+\alpha'}\right) =: (p, q)$

Hypothèse : $(p, q)$ reste constant et $q < p$.

Variable aléatoire $\mathrm{Inf}\,(\boldsymbol{T}, \boldsymbol{T}')$ = loi exponentielle paramètre $\alpha + \alpha'$

Processus de comptage $(\boldsymbol{N}, \boldsymbol{N}')$ = processus de Poisson paramètre $(\alpha, \alpha')$

On a $\mathbb{P}[\boldsymbol{T} < \mathbf{T}'] = \frac{\alpha}{\alpha+\alpha'} = \frac{h}{h+h'}$

Puissance de hachage relative de (Alice, Bob) : $\left(\frac{\alpha}{\alpha+\alpha'}, \frac{\alpha'}{\alpha+\alpha'}\right) =: (p, q)$

Hypothèse : $(p, q)$ reste constant et $q < p$.

Variable aléatoire $\mathrm{Inf}\,(\boldsymbol{T}, \boldsymbol{T}')$ = loi exponentielle paramètre $\alpha + \alpha'$

Processus de comptage $(\boldsymbol{N}, \boldsymbol{N}')$ = processus de Poisson paramètre $(\alpha, \alpha')$

On a

$$\begin{aligned} \alpha &= \frac{p}{\mathbb{E}[\mathrm{Inf}(\boldsymbol{T}, \boldsymbol{T}')]} \\ \alpha' &= \frac{q}{\mathbb{E}[\mathrm{Inf}(\boldsymbol{T}, \boldsymbol{T}')]} \end{aligned}$$

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\tau$.

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\tau$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $G_i, T_i, H_i$

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\tau$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $G_i, T_i, H_i$
   - coûts par unité de temps indépendants de la stratégie

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\tau$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $G_i, T_i, H_i$
   - coûts par unité de temps indépendants de la stratégie
   - Taux de rendement long-terme : $\frac{G_1 + \ldots + G_n}{T_1 + \ldots + T_n} \longrightarrow \frac{\mathbb{E}[G]}{\mathbb{E}[T]}$

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\boldsymbol{\tau}$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $\boldsymbol{G_i, T_i, H_i}$
   - coûts par unité de temps indépendants de la stratégie
   - Taux de rendement long-terme : $\frac{\boldsymbol{G_1 + ... + G_n}}{\boldsymbol{T_1 + ... + T_n}} \longrightarrow \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{T}]}$
   - Taux de rendement honnête $\sim$ puissance de hachage relative $q$

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\tau$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $G_i, T_i, H_i$
   - coûts par unité de temps indépendants de la stratégie
   - Taux de rendement long-terme : $\frac{G_1 + \ldots + G_n}{T_1 + \ldots + T_n} \longrightarrow \frac{\mathbb{E}[G]}{\mathbb{E}[T]}$
   - Taux de rendement honnête $\sim$ puissance de hachage relative $q$
   - Long terme : $\mathbb{E}[T] \sim \mathbb{E}[H]$ et $\frac{\mathbb{E}[G]}{\mathbb{E}[T]} \sim \frac{\mathbb{E}[G]}{\mathbb{E}[H]}$

1. Stratégie : suite de cycles ; cycle = temps d'arrêt $\boldsymbol{\tau}$.
2. Rentabilité d'une stratégie de minage
   - attaques répétitives, Gains, durées, avancée hauteur : $\boldsymbol{G_i, T_i, H_i}$
   - coûts par unité de temps indépendants de la stratégie
   - Taux de rendement long-terme : $\frac{\boldsymbol{G_1 + \ldots + G_n}}{\boldsymbol{T_1 + \ldots + T_n}} \longrightarrow \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{T}]}$
   - Taux de rendement honnête $\sim$ puissance de hachage relative $q$
   - Long terme : $\mathbb{E}[\boldsymbol{T}] \sim \mathbb{E}[\boldsymbol{H}]$ et $\frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{T}]} \sim \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{H}]}$
3. Stratégie déviante rentable s'il existe $\tau$ tq $\mathbb{E}[\boldsymbol{G} - q\,\boldsymbol{H}] > 0$

1. Alice (attaquant) contre Bob (honnêtes mineurs)

1. Alice (attaquant) contre Bob (honnêtes mineurs)
2. Stratégie 1+2. Si Alice découvre un bloc en premier, elle attend encore que deux blocs soient découverts avant éventuellement de publier ses blocs secrets. Sinon, elle abandonne et revient miner sur le dernier bloc de la blockchain officielle.

1. Alice (attaquant) contre Bob (honnêtes mineurs)
2. Stratégie 1+2. Si Alice découvre un bloc en premier, elle attend encore que deux blocs soient découverts avant éventuellement de publier ses blocs secrets. Sinon, elle abandonne et revient miner sur le dernier bloc de la blockchain officielle.
3. Univers des résultats possibles : $\Omega = \{B, \mathrm{AAA}, \mathrm{AAB}, \mathrm{ABA}, \mathrm{ABB}\}$
   - $\mathbb{P}[B] = p, \mathbb{P}[\mathrm{AAA}] = q^3, \mathbb{P}[\mathrm{AAB}] = \mathbb{P}[\mathrm{ABA}] = p\,q^2, \mathbb{P}[\mathrm{ABB}] = p^2\,q$

1. Alice (attaquant) contre Bob (honnêtes mineurs)
2. Stratégie 1+2. Si Alice découvre un bloc en premier, elle attend encore que deux blocs soient découverts avant éventuellement de publier ses blocs secrets. Sinon, elle abandonne et revient miner sur le dernier bloc de la blockchain officielle.
3. Univers des résultats possibles : $\Omega = \{B, \mathrm{AAA}, \mathrm{AAB}, \mathrm{ABA}, \mathrm{ABB}\}$
   - $\mathbb{P}[B] = p, \mathbb{P}[\mathrm{AAA}] = q^3, \mathbb{P}[\mathrm{AAB}] = \mathbb{P}[\mathrm{ABA}] = p\,q^2, \mathbb{P}[\mathrm{ABB}] = p^2\,q$
   - $\boldsymbol{G}(B) = \boldsymbol{G}(\mathrm{ABB}) = 0, \boldsymbol{G}(\mathrm{AAA}) = 3, \boldsymbol{G}(\mathrm{AAB}) = \boldsymbol{G}(\mathrm{ABA}) = 2$

1. Alice (attaquant) contre Bob (honnêtes mineurs)
2. Stratégie 1+2. Si Alice découvre un bloc en premier, elle attend encore que deux blocs soient découverts avant éventuellement de publier ses blocs secrets. Sinon, elle abandonne et revient miner sur le dernier bloc de la blockchain officielle.
3. Univers des résultats possibles : $\Omega = \{B, \mathrm{AAA}, \mathrm{AAB}, \mathrm{ABA}, \mathrm{ABB}\}$
   - $\mathbb{P}[B] = p, \mathbb{P}[\mathrm{AAA}] = q^3, \mathbb{P}[\mathrm{AAB}] = \mathbb{P}[\mathrm{ABA}] = p\,q^2, \mathbb{P}[\mathrm{ABB}] = p^2\,q$
   - $\boldsymbol{G}(B) = \boldsymbol{G}(\mathrm{ABB}) = 0, \boldsymbol{G}(\mathrm{AAA}) = 3, \boldsymbol{G}(\mathrm{AAB}) = \boldsymbol{G}(\mathrm{ABA}) = 2$
   - $\boldsymbol{H}(B) = 1, \boldsymbol{H}(\mathrm{ABB}) = \boldsymbol{H}(\mathrm{AAB}) = \boldsymbol{H}(\mathrm{ABA}) = 2, \boldsymbol{H}(\mathrm{AAA}) = 3$

1. Alice (attaquant) contre Bob (honnêtes mineurs)
2. Stratégie 1+2. Si Alice découvre un bloc en premier, elle attend encore que deux blocs soient découverts avant éventuellement de publier ses blocs secrets. Sinon, elle abandonne et revient miner sur le dernier bloc de la blockchain officielle.
3. Univers des résultats possibles : $\Omega = \{B, \mathrm{AAA}, \mathrm{AAB}, \mathrm{ABA}, \mathrm{ABB}\}$
   - $\mathbb{P}[B] = p, \mathbb{P}[\mathrm{AAA}] = q^3, \mathbb{P}[\mathrm{AAB}] = \mathbb{P}[\mathrm{ABA}] = p\,q^2, \mathbb{P}[\mathrm{ABB}] = p^2\,q$
   - $\boldsymbol{G}(B) = \boldsymbol{G}(\mathrm{ABB}) = 0, \boldsymbol{G}(\mathrm{AAA}) = 3, \boldsymbol{G}(\mathrm{AAB}) = \boldsymbol{G}(\mathrm{ABA}) = 2$
   - $\boldsymbol{H}(B) = 1, \boldsymbol{H}(\mathrm{ABB}) = \boldsymbol{H}(\mathrm{AAB}) = \boldsymbol{H}(\mathrm{ABA}) = 2, \boldsymbol{H}(\mathrm{AAA}) = 3$
4. Taux de rendement $\Gamma = \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{H}]} = \frac{q^2 \cdot (4-q)}{1+q+q^3}$ et $\Gamma > q$ ssi $q > \sqrt{2} - 1$.

**Supposons réseau Bitcoin sans ajustement de difficulté. Alors, la meilleure stratégie est toujours la stratégie honnête.**

**Supposons réseau Bitcoin sans ajustement de difficulté. Alors, la meilleure stratégie est toujours la stratégie honnête.**

**Preuve.** Posons $\tau = \mathbb{E}[\mathrm{Inf}(\boldsymbol{T}, \boldsymbol{T}')]$. A priori $\tau \neq 10$ minutes. On a $(\boldsymbol{N}, \boldsymbol{N}')$ processus de Poisson de paramètre $(\alpha, \alpha') = \left(\frac{p}{\tau}, \frac{q}{\tau}\right)$. Soit une stratégie de minage $\boldsymbol{\tau}$ avec $\mathbb{E}[\boldsymbol{\tau}] < \infty$. On a $\boldsymbol{G} \leqslant \boldsymbol{N}'(\boldsymbol{\tau})$ et par hypothèse. Le processus $\boldsymbol{N}'(t) - \alpha' t$ est une martingale. Donc, d'après le théorème de Doob, $\mathbb{E}[\boldsymbol{N}'(\boldsymbol{\tau})] = \alpha' \, \mathbb{E}[\boldsymbol{\tau}]$. D'où $\Gamma = \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{\tau}]} \leqslant \alpha' = \frac{q}{\tau}$. Par ailleurs, la stratégie honnête donne en moyenne $q$ en un temps $\tau$, soit $\Gamma(\mathrm{HM}) = \frac{q}{\tau}$.

**Supposons réseau Bitcoin sans ajustement de difficulté. Alors, la meilleure stratégie est toujours la stratégie honnête.**

**Preuve.** Posons $\tau = \mathbb{E}[\mathrm{Inf}(\boldsymbol{T}, \boldsymbol{T}')]$. A priori $\tau \neq 10$ minutes. On a $(\boldsymbol{N}, \boldsymbol{N}')$ processus de Poisson de paramètre $(\alpha, \alpha') = \left(\frac{p}{\tau}, \frac{q}{\tau}\right)$. Soit une stratégie de minage $\boldsymbol{\tau}$ avec $\mathbb{E}[\boldsymbol{\tau}] < \infty$. On a $\boldsymbol{G} \leqslant \boldsymbol{N}'(\boldsymbol{\tau})$ et par hypothèse. Le processus $\boldsymbol{N}'(t) - \alpha' t$ est une martingale. Donc, d'après le théorème de Doob, $\mathbb{E}[\boldsymbol{N}'(\boldsymbol{\tau})] = \alpha' \, \mathbb{E}[\boldsymbol{\tau}]$. D'où $\Gamma = \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{\tau}]} \leqslant \alpha' = \frac{q}{\tau}$. Par ailleurs, la stratégie honnête donne en moyenne $q$ en un temps $\tau$, soit $\Gamma(\mathrm{HM}) = \frac{q}{\tau}$.

Par suite, le problème est du à l'ajustement de difficulté dans Bitcoin...

**Supposons réseau Bitcoin sans ajustement de difficulté. Alors, la meilleure stratégie est toujours la stratégie honnête.**

**Preuve.** Posons $\tau = \mathbb{E}[\mathrm{Inf}(\boldsymbol{T}, \boldsymbol{T}')]$. A priori $\tau \neq 10$ minutes. On a $(\boldsymbol{N}, \boldsymbol{N}')$ processus de Poisson de paramètre $(\alpha, \alpha') = \left(\frac{p}{\tau}, \frac{q}{\tau}\right)$. Soit une stratégie de minage $\boldsymbol{\tau}$ avec $\mathbb{E}[\boldsymbol{\tau}] < \infty$. On a $\boldsymbol{G} \leqslant \boldsymbol{N}'(\boldsymbol{\tau})$ et par hypothèse. Le processus $\boldsymbol{N}'(t) - \alpha' t$ est une martingale. Donc, d'après le théorème de Doob, $\mathbb{E}[\boldsymbol{N}'(\boldsymbol{\tau})] = \alpha' \, \mathbb{E}[\boldsymbol{\tau}]$. D'où $\Gamma = \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{\tau}]} \leqslant \alpha' = \frac{q}{\tau}$. Par ailleurs, la stratégie honnête donne en moyenne $q$ en un temps $\tau$, soit $\Gamma(\mathrm{HM}) = \frac{q}{\tau}$.

Par suite, le problème est du à l'ajustement de difficulté dans Bitcoin...

Peut-on rétablir Bitcoin comme l'imaginait Satoshi Nakamoto ?

En présence d'un attaquant, on constate la présence de blocs orphelins

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

Or, il ignore les efforts ayant conduit à créer tous les blocs orphelins

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

Or, il ignore les efforts ayant conduit à créer tous les blocs orphelins

Cela conduit à un ajustement de difficulté à la baisse alors qu'il y a le même nombre d'acteurs sur le réseau...

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

Or, il ignore les efforts ayant conduit à créer tous les blocs orphelins

Cela conduit à un ajustement de difficulté à la baisse alors qu'il y a le même nombre d'acteurs sur le réseau...

Idée : prise en compte des blocs orphelins dans la formule d'ajustement de difficulté

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

Or, il ignore les efforts ayant conduit à créer tous les blocs orphelins

Cela conduit à un ajustement de difficulté à la baisse alors qu'il y a le même nombre d'acteurs sur le réseau...

Idée : prise en compte des blocs orphelins dans la formule d'ajustement de difficulté

Nouvelle formule : $\Delta_i = \frac{(n_0 + n_1) \cdot \tau_0}{T} \times \Delta_{i-1}$ où $n_1 =$ nombre de blocs « orphelins » détectés (i.e., blocs n'appartenant pas à la blockchain officielle).

En présence d'un attaquant, on constate la présence de blocs orphelins

Le paramètre de difficulté $\Delta$ devrait refléter la réelle puissance de calcul déployée sur tout le réseau.

Or, il ignore les efforts ayant conduit à créer tous les blocs orphelins

Cela conduit à un ajustement de difficulté à la baisse alors qu'il y a le même nombre d'acteurs sur le réseau...

Idée : prise en compte des blocs orphelins dans la formule d'ajustement de difficulté

Nouvelle formule : $\Delta_i = \frac{(n_0+n_1)\cdot\tau_0}{T} \times \Delta_{i-1}$ où $n_1 =$ nombre de blocs « orphelins » détectés (i.e., blocs n'appartenant pas à la blockchain officielle).

Dans ce cas, $\Gamma = \frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{D}]}$ où $\boldsymbol{D} = \boldsymbol{H} +$ #blocs orphelins.

Soit $\boldsymbol{\tau}$ un cycle d'attaque avec $\mathbb{E}[\boldsymbol{\tau}] < \infty$. Pour $X = A, H$, on note $\mathrm{Off}_X$ (resp. $\mathrm{Orph}_X$) = #blocs officiels (resp. orphelins) minés par $X$ durant le cycle. On pose :

$$\begin{aligned} \boldsymbol{N}(\boldsymbol{\tau}) &= \mathrm{Off}_H + \mathrm{Orph}_H \\ \boldsymbol{N}'(\boldsymbol{\tau}) &= \mathrm{Off}_A + \mathrm{Orph}_A \end{aligned}$$

Le gain d'Alice est $\boldsymbol{G}(\boldsymbol{\tau}) = \mathrm{Off}_A$.

On a

$$\mathrm{Off}_A + \mathrm{Off}_H + \mathrm{Orph}_H \leqslant \boldsymbol{D}(\boldsymbol{\tau})$$

Les processus $\boldsymbol{N}$ et $\boldsymbol{N}'$ sont des processus de Poisson de paramètres $\lambda \cdot p$ et $\lambda \cdot q$ où $\lambda$ est un paramètre du à l'ajustement de difficulté. La condition $\mathbb{E}[\boldsymbol{\tau}] < \infty$ entraîne $\mathbb{E}[\boldsymbol{N}(\boldsymbol{\tau})] = \lambda\, p\, \mathbb{E}[\boldsymbol{\tau}]$ et $\mathbb{E}[\boldsymbol{N}'(\boldsymbol{\tau})] = \lambda\, q\, \mathbb{E}[\boldsymbol{\tau}]$.

Donc,

$$p\,\mathbb{E}[\mathrm{Off}_A] \leqslant p\mathbb{E}[\boldsymbol{N}'(\tau)] = p\lambda q\,\mathbb{E}[\boldsymbol{\tau}] = q\lambda p\,\mathbb{E}[\boldsymbol{\tau}] = q\,\mathbb{E}[\boldsymbol{N}(\boldsymbol{\tau})] = q\,\mathbb{E}[\mathrm{Off}_H] + q\,\mathbb{E}[\mathrm{Orph}_H]$$

ce qui entraîne :

$$\begin{aligned}\mathbb{E}[\boldsymbol{G}(\boldsymbol{\tau})] &= \mathbb{E}[\mathrm{Off}_A] \\ &= p\,\mathbb{E}[\mathrm{Off}_A] + q\,\mathbb{E}[\mathrm{Off}_A] \\ &\leqslant q\,\mathbb{E}[\mathrm{Off}_H] + q\,\mathbb{E}[\mathrm{Orph}_H] + q\,\mathbb{E}[\mathrm{Off}_A] \\ &\leqslant q\cdot\mathbb{E}[\boldsymbol{D}(\boldsymbol{\tau})]\end{aligned}$$

D'où le résultat.

On peut être plus précis. On a

$$\begin{aligned} p\,\mathbb{E}[\mathrm{Off}_A] + p\,\mathbb{E}[\mathrm{Orph}_A] &= p\mathbb{E}[\boldsymbol{N}'(\boldsymbol{\tau})] = p\lambda q\,\mathbb{E}[\boldsymbol{\tau}] = q\lambda p\,\mathbb{E}[\boldsymbol{\tau}] \\ &= q\,\mathbb{E}[\boldsymbol{N}(\boldsymbol{\tau})] \\ &= q\,\mathbb{E}[\mathrm{Off}_H] + q\,\mathbb{E}[\mathrm{Orph}_H] \end{aligned} \tag{1}$$

Soit $\mathrm{Orph}'_A$ les blocs orphelins d'Alice détectés par le réseau. On a d'une part

$$\mathbb{E}[\boldsymbol{D}(\boldsymbol{\tau})] = \mathbb{E}[\mathrm{Off}_H] + \mathbb{E}[\mathrm{Orph}_H] + \mathbb{E}[\mathrm{Off}_A] + \mathbb{E}[\mathrm{Orph}'_A]$$

et d'autre part d'après (1),

$$\begin{aligned} \mathbb{E}[\mathrm{Off}_A] + \mathbb{E}[\mathrm{Orph}'_A] &= p\,\mathbb{E}[\mathrm{Off}_A] + p\,\mathbb{E}[\mathrm{Orph}'_A] + q\,\mathbb{E}[\mathrm{Off}_A] + q\,\mathbb{E}[\mathrm{Orph}'_A] \\ &\leqslant p\,\mathbb{E}[\mathrm{Off}_A] + p\,\mathbb{E}[\mathrm{Orph}_A] + q\,\mathbb{E}[\mathrm{Off}_A] + q\,\mathbb{E}[\mathrm{Orph}'_A] \end{aligned}$$

$$
\begin{aligned}
&\leqslant q\,\mathbb{E}[\mathrm{Off}_H] + q\,\mathbb{E}[\mathrm{Orph}_H] + q\,\mathbb{E}[\mathrm{Off}_A] + q\,\mathbb{E}[\mathrm{Orph}'_A] \\
&\leqslant q \cdot \mathbb{E}[\boldsymbol{D}(\boldsymbol{\tau})]
\end{aligned}
$$

Supposons que le protocole accorde une récompense $x \leqslant 1$ à tout créateur de bloc orphelin détecté par le réseau. Alors,

$$\boldsymbol{G}(\boldsymbol{\tau}) = \mathrm{Off}_A + x \cdot \mathrm{Orph}'_A \leqslant \mathrm{Off}_A + \mathrm{Orph}'_A$$

Donc,

$$\mathbb{E}[\boldsymbol{G}(\boldsymbol{\tau})] \leqslant \mathbb{E}[\mathrm{Off}_A] + \mathbb{E}[\mathrm{Orph}'_A] \leqslant q \cdot \mathbb{E}[\boldsymbol{D}(\boldsymbol{\tau})] \tag{2}$$

D'où

**Théorème 1.** *Soit un réseau Bitcoin modifié avec formule d'ajustement de difficulté qui prend en compte la production de blocs orphelins et accorde une récompense $x \leqslant 1$ coinbase à tout créateur de bloc orphelin. Alors, la stratégie honnête est toujours la meilleure.*

Cela découle de (2) et du fait que $\frac{\mathbb{E}[\boldsymbol{G}]}{\mathbb{E}[\boldsymbol{D}]} = q$ pour la stratégie honnête.

1. **Jeu du minage**

   - Blocs = jetons

1. **Jeu du minage**
   - Blocs = jetons
   - Actions Lancer, Ecraser, Abandon

1. **Jeu du minage**
   - Blocs = jetons
   - Actions Lancer, Ecraser, Abandon
2. **Lancer** : un croupier lance une pièce de monnaie. Effet :
   - Pile : coût 0, le joueur gagne un jeton

1. **Jeu du minage**
   - Blocs = jetons
   - Actions Lancer, Ecraser, Abandon
2. **Lancer** : un croupier lance une pièce de monnaie. Effet :
   - Pile : coût 0, le joueur gagne un jeton
   - Face : coût $q$€, la banque gagne un jeton

1. **Jeu du minage**
   - Blocs = jetons
   - Actions Lancer, Ecraser, Abandon
2. **Lancer** : un croupier lance une pièce de monnaie. Effet :
   - Pile : coût 0, le joueur gagne un jeton
   - Face : coût $q$€, la banque gagne un jeton
3. **Ecraser** : possible si le joueur (resp. la banque) possède $a$ (resp. $h$) jetons avec $a > h$. Effet : La banque perd tous ses jetons, le joueur perd $h+1$ jetons mais il gagne $h+1-q$ €.

1. **Jeu du minage**
   - Blocs = jetons
   - Actions Lancer, Ecraser, Abandon
2. **Lancer** : un croupier lance une pièce de monnaie. Effet :
   - Pile : coût 0, le joueur gagne un jeton
   - Face : coût $q$€, la banque gagne un jeton
3. **Ecraser** : possible si le joueur (resp. la banque) possède $a$ (resp. $h$) jetons avec $a > h$. Effet : La banque perd tous ses jetons, le joueur perd $h+1$ jetons mais il gagne $h+1-q$ €.
4. **Abandon** : le joueur et la banque perdent tous leurs jetons. Coût 0.

1. Discrétisation sur le nombre d'actions possibles $n$

1. Discrétisation sur le nombre d'actions possibles $n$
2. Espérance de gain maximale $E$ avec $n$ actions au plus partant de $a$ jetons contre la banque ayant $h$ jetons.

1. Discrétisation sur le nombre d'actions possibles $n$
2. Espérance de gain maximale $E$ avec $n$ actions au plus partant de $a$ jetons contre la banque ayant $h$ jetons.
3. Code Mathematica 11.3

1. Discrétisation sur le nombre d'actions possibles $n$
2. Espérance de gain maximale $E$ avec $n$ actions au plus partant de $a$ jetons contre la banque ayant $h$ jetons.
3. Code Mathematica 11.3

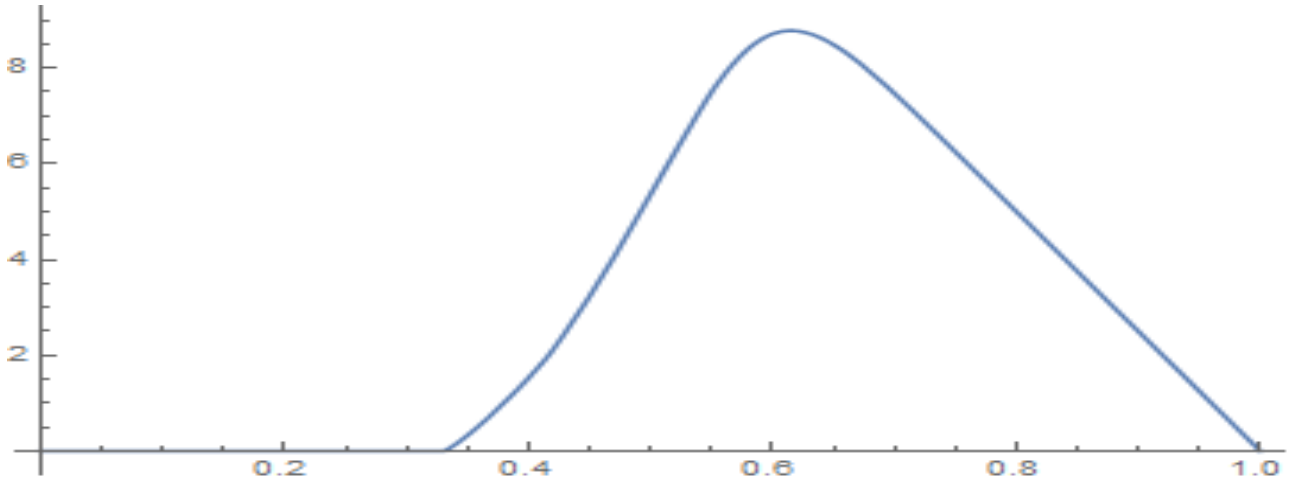
```
E[a_, h_, n_, q_, c_] := E[a, h, n, q, c] =
  Piecewise[{(* case a>h ==> override, wait *)
    {Max[(h + 1) - c + E[a - h - 1, 0, n - 1, q, c], q*E[a + 1, h, n - 1, q, c] + (1 - q) * (E[a, h + 1, n - 1, q, c] - c)], (a > h)},
    (* case a<=h ==> abandon, flip coin *)
    {Max[E[0, 0, n - 1, q, c], q*E[a + 1, h, n - 1, q, c] + (1 - q) * (E[a, h + 1, n - 1, q, c] - c)], (a <= h)}}];
(* Initial conditions*)
E[a_, h_, 0, q_, c_] := 0;
```

1. Discrétisation sur le nombre d'actions possibles $n$
2. Espérance de gain maximale $E$ avec $n$ actions au plus partant de $a$ jetons contre la banque ayant $h$ jetons.
3. Code Mathematica 11.3

```
E[a_, h_, n_, q_, c_] := E[a, h, n, q, c] =
  Piecewise[{(* case a>h ==> override, wait *)
    {Max[(h + 1) - c + E[a - h - 1, 0, n - 1, q, c], q * E[a + 1, h, n - 1, q, c] + (1 - q) * (E[a, h + 1, n - 1, q, c] - c)], (a > h)},
    (* case a<=h ==> abandon, flip coin *)
    {Max[E[0, 0, n - 1, q, c], q * E[a + 1, h, n - 1, q, c] + (1 - q) * (E[a, h + 1, n - 1, q, c] - c)], (a <= h)}}];
(* Initial conditions*)
E[a_, h_, 0, q_, c_] := 0;
```

4. Si $q = 0.3294$, $E[0, 0, 100, q, q] > 0$. Si $q = 0.3293$ $E[0, 0, 200, q, q] = 0$

On trace le graphe de $E[0,0,n,q,q]$ pour $n=50$ en fonction de $q \in [0,1]$.




A partir d'un certain seuil proche de 32,94%, la stratégie honnête n'est plus la meilleure.

Mais si $q \longrightarrow 1$, la meilleure stratégie se rapproche de la stratégie honnête.

(Cependant seul le cas $q < \frac{1}{2}$ est réellement intéressant.)

1. Discrétisation sur le nombre d'actions **Lancer** possibles $n$ et fin si retour à la situation $a = h = 0$

1. Discrétisation sur le nombre d'actions **Lancer** possibles $n$ et fin si retour à la situation $a = h = 0$
2. Calcul formel : `Assuming[(q > 0), Simplify[Φ[0, 0, 3, q, q]]]`. On obtient $\Phi(0, 0, 3, q, q) = 0$ pour $q < \sqrt{2} - 1$ et $\Phi(0, 0, 3, q, q) = q \cdot (3q - 1 - q^2 - q^3)$ pour $\sqrt{2} - 1 < q$

1. Discrétisation sur le nombre d'actions **Lancer** possibles $n$ et fin si retour à la situation $a = h = 0$
2. Calcul formel : `Assuming[(q > 0), Simplify[Φ[0, 0, 3, q, q]]]`. On obtient $\Phi(0,0,3,q,q) = 0$ pour $q < \sqrt{2} - 1$ et $\Phi(0,0,3,q,q) = q \cdot (3q - 1 - q^2 - q^3)$ pour $\sqrt{2} - 1 < q$
3. Code Mathematica 11.3

1. Discrétisation sur le nombre d'actions **Lancer** possibles $n$ et fin si retour à la situation $a = h = 0$

2. Calcul formel : `Assuming[(q > 0), Simplify[Φ[0, 0, 3, q, q]]]`. On obtient $\Phi(0,0,3,q,q) = 0$ pour $q < \sqrt{2} - 1$ et $\Phi(0,0,3,q,q) = q \cdot (3q - 1 - q^2 - q^3)$ pour $\sqrt{2} - 1 < q$

3. Code Mathematica 11.3

```
Φ[a_, h_, n_, q_, c_] := Φ[a, h, n, q, c] =
  Piecewise[{(* case a>h+1 ==> override, flip coin *)
    {Max[(h + 1) - c + Φ[a - h - 1, 0, n, q, c], q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a > h + 1)},
    (* case a=h+1 ==> override, flip coin *)
    {Max[(h + 1) - c, q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a == h + 1)},
    (* case a<=h ==> abandon, flip coin *)
    {Max[0, q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a <= h)}}];
(* Initial conditions*)
Φ[a_, h_, 0, q_, c_] := If[a > h, a - c * (a - h), 0];
```

1. Discrétisation sur le nombre d'actions **Lancer** possibles $n$ et fin si retour à la situation $a = h = 0$

2. Calcul formel : `Assuming[(q > 0), Simplify[Φ[0, 0, 3, q, q]]]`. On obtient $\Phi(0,0,3,q,q) = 0$ pour $q < \sqrt{2} - 1$ et $\Phi(0,0,3,q,q) = q \cdot (3q - 1 - q^2 - q^3)$ pour $\sqrt{2} - 1 < q$

3. Code Mathematica 11.3

```
Φ[a_, h_, n_, q_, c_] := Φ[a, h, n, q, c] =
  Piecewise[{(* case a>h+1 ==> override, flip coin *)
    {Max[(h + 1) - c + Φ[a - h - 1, 0, n, q, c], q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a > h + 1)},
    (* case a=h+1 ==> override, flip coin *)
    {Max[(h + 1) - c, q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a == h + 1)},
    (* case a<=h ==> abandon, flip coin *)
    {Max[0, q * Φ[a + 1, h, n - 1, q, c] + (1 - q) * (Φ[a, h + 1, n - 1, q, c] - c)], (a <= h)}}];
(* Initial conditions*)
Φ[a_, h_, 0, q_, c_] := If[a > h, a - c * (a - h), 0];
```

4. La stratégie 1+2 est la meilleure si par cycle $H \leqslant 3$.

- Variation du jeu de Pile ou Face classique avec des jetons

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions
- Jeu biaisé en faveur du joueur

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions
- Jeu biaisé en faveur du joueur
- Jeu du Minage : le joueur ne paye que les actions **Lancer** si le résultat est Face !

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions
- Jeu biaisé en faveur du joueur
- Jeu du Minage : le joueur ne paye que les actions **Lancer** si le résultat est Face !
- Bitcoin : la difficulté ne reflète pas la vraie puissance de calcul déployée sur le réseau

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions
- Jeu biaisé en faveur du joueur
- Jeu du Minage : le joueur ne paye que les actions **Lancer** si le résultat est Face !
- Bitcoin : la difficulté ne reflète pas la vraie puissance de calcul déployée sur le réseau
- Une stratégie rétention de blocs ralentit la progression du réseau

- Variation du jeu de Pile ou Face classique avec des jetons
- Le joueur dispose plus ou moins des mêmes actions
- Jeu biaisé en faveur du joueur
- Jeu du Minage : le joueur ne paye que les actions **Lancer** si le résultat est Face !
- Bitcoin : la difficulté ne reflète pas la vraie puissance de calcul déployée sur le réseau
- Une stratégie rétention de blocs ralentit la progression du réseau
- Baisse inconsidérée du paramètre de difficulté de minage

- Accepter de perdre de l'argent pendant 15 jours minimum

- Accepter de perdre de l'argent pendant 15 jours minimum
- Attendre ajustement de difficulté

- Accepter de perdre de l'argent pendant 15 jours minimum
- Attendre ajustement de difficulté
- Supposer qu'il n'y aura pas afflux de nouveau mineurs attirés par une difficulté réduite

- Accepter de perdre de l'argent pendant 15 jours minimum
- Attendre ajustement de difficulté
- Supposer qu'il n'y aura pas afflux de nouveau mineurs attirés par une difficulté réduite
- Supposer cohésion des honnêtes mineurs

- Accepter de perdre de l'argent pendant 15 jours minimum
- Attendre ajustement de difficulté
- Supposer qu'il n'y aura pas afflux de nouveau mineurs attirés par une difficulté réduite
- Supposer cohésion des honnêtes mineurs
- Bitcoin « protégé » par ses paramètres en dur dans le code : 10 minutes et 2016

- Accepter de perdre de l'argent pendant 15 jours minimum
- Attendre ajustement de difficulté
- Supposer qu'il n'y aura pas afflux de nouveau mineurs attirés par une difficulté réduite
- Supposer cohésion des honnêtes mineurs
- Bitcoin « protégé » par ses paramètres en dur dans le code : 10 minutes et 2016
- Ancien protocole Ethereum plus fragile : ajustement en continu, rémunération de certains blocs orphelins (oncles), connectivité a priori >0 mais meilleure formule ajustement de difficulté

- Idée : le paramètre de difficulté $\Delta$ doit refléter la réelle puissance de hachage

- Idée : le paramètre de difficulté $\Delta$ doit refléter la réelle puissance de hachage
- Compter les orphelins

- Idée : le paramètre de difficulté $\Delta$ doit refléter la réelle puissance de hachage
- Compter les orphelins
- Blockchain officielle : celle qui maximise $\sum \Delta_i$ mais en cas d'égalité, celle qui signale le plus d'orphelins

- Idée : le paramètre de difficulté $\Delta$ doit refléter la réelle puissance de hachage
- Compter les orphelins
- Blockchain officielle : celle qui maximise $\sum \Delta_i$ mais en cas d'égalité, celle qui signale le plus d'orphelins
- Ajustement de difficulté : $\Delta' = \Delta \cdot \frac{\Delta D}{\Delta T}$ où $D$ est une fonction de difficulté ($D = H$, hauteur de la Blockchain dans Bitcoin aujourd'hui) qui augmente de 1 pour chaque bloc officiel ou orphelin détecté.

- Variation du jeu du Minage

- Variation du jeu du Minage
- Le joueur dispose des mêmes actions :
  - **Lancer** : un croupier lance une pièce de monnaie. Effet :
    - Pile : coût 0, le joueur gagne un jeton

- Variation du jeu du Minage
- Le joueur dispose des mêmes actions :
  - **Lancer** : un croupier lance une pièce de monnaie. Effet :
    - Pile : coût 0, le joueur gagne un jeton
    - Face : coût $q$, la banque gagne un jeton

- Variation du jeu du Minage
- Le joueur dispose des mêmes actions :
  - **Lancer** : un croupier lance une pièce de monnaie. Effet :
    - Pile : coût 0, le joueur gagne un jeton
    - Face : coût $q$, la banque gagne un jeton
  - L'action **Ecraser** coûte plus cher : le joueur paye en plus pour chaque jeton remplacé ! Possible si le joueur (resp. la banque) possède $a$ (resp. $h$) jetons avec $a > h$. Effet : La banque perd tous ses jetons, le joueur perd $h+1$ jetons mais il gagne $h+1-q\cdot(h+1)$ €.

- Variation du jeu du Minage
- Le joueur dispose des mêmes actions :
  - **Lancer** : un croupier lance une pièce de monnaie. Effet :
    - Pile : coût 0, le joueur gagne un jeton
    - Face : coût $q$, la banque gagne un jeton
  - L'action **Ecraser** coûte plus cher : le joueur paye en plus pour chaque jeton remplacé ! Possible si le joueur (resp. la banque) possède $a$ (resp. $h$) jetons avec $a > h$. Effet : La banque perd tous ses jetons, le joueur perd $h+1$ jetons mais il gagne $h+1-q\cdot(h+1)$ €.
  - Action **Abandon** identique : tout le monde perd tous ses jetons. Coût 0.

Espérance de gain maximale $\Omega$ avec $n$ actions au plus, partant de $a$ jetons contre la banque qui possède $h$ jetons

**Théorème 2.** *Pour tous entiers $a, h$,n, on a $\Omega(a,h,n) \leqslant p \cdot a$ avec $p = 1-q$.*

**Corollaire 3.** *pour tout entier $n$ on a $\Omega(0,0,n) = 0$.*

**Théorème 4.** *Quelle que soit la stratégie de minage choisie avec $\mathbb{E}[\tau] < \infty$ et quelle que soit la connectivité de l'attaquant, on a*

$$\mathbb{E}[\boldsymbol{G}(\tau)] \;\leqslant\; q\,\mathbb{E}[\boldsymbol{D}(\tau)]$$

La stratégie honnête est toujours la meilleure !

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.
- **Dès qu'un mineur possède 32.94% de puissance de hachage du réseau, quelle que soit sa connectivité, il n'a plus intérêt à miner honnêtement.**

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.
- **Dès qu'un mineur possède 32.94% de puissance de hachage du réseau, quelle que soit sa connectivité, il n'a plus intérêt à miner honnêtement.**
- **La plus simple de ces stratégies est la stratégie 1+2.**

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.
- **Dès qu'un mineur possède 32.94% de puissance de hachage du réseau, quelle que soit sa connectivité, il n'a plus intérêt à miner honnêtement.**
- **La plus simple de ces stratégies est la stratégie 1+2.**
- De telles attaques sont néanmoins difficiles à mettre en oeuvre en pratique sur Bitcoin qui est « protégé » par ses paramètres en dur « 2016 » et « 600 ».

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.
- **Dès qu'un mineur possède 32.94% de puissance de hachage du réseau, quelle que soit sa connectivité, il n'a plus intérêt à miner honnêtement.**
- **La plus simple de ces stratégies est la stratégie 1+2.**
- De telles attaques sont néanmoins difficiles à mettre en oeuvre en pratique sur Bitcoin qui est « protégé » par ses paramètres en dur « 2016 » et « 600 ».
- **En prenant en compte la production de blocs orphelins et en l'intégrant dans la formule d'ajustement de difficulté, Bitcoin modifié est « parfait » au sens où pouvait l'imaginer son créateur : la stratégie de minage honnête est optimale.**

- La formule d'ajustement de difficulté de Bitcoin a un défaut qui peut être attaquée par des attaques de rétention de blocs.
- **Dès qu'un mineur possède 32.94% de puissance de hachage du réseau, quelle que soit sa connectivité, il n'a plus intérêt à miner honnêtement.**
- **La plus simple de ces stratégies est la stratégie 1+2.**
- De telles attaques sont néanmoins difficiles à mettre en oeuvre en pratique sur Bitcoin qui est « protégé » par ses paramètres en dur « 2016 » et « 600 ».
- **En prenant en compte la production de blocs orphelins et en l'intégrant dans la formule d'ajustement de difficulté, Bitcoin modifié est « parfait » au sens où pouvait l'imaginer son créateur : la stratégie de minage honnête est optimale.**
- **On a donné deux preuves de ce résultat.**