

**Mémoire présenté devant le CNAM pour l'obtention du Master
Droit Economie Gestion, mention Actuariat
et l'admission à l'Institut des Actuaires**

le 20 janvier 2022

Par : Cyril GRUNSPAN

Titre: Jeu du minage, et sécurité du protocole Bitcoin

Confidentialité : NON OUI (Durée : 1 an 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Présidente du Jury :
Mme Sandrine LEMERY

signatures

Entreprise :

Nom :

Membres présents du jury de
l'Institut des Actuaires :

pe *peints par Teaus*

M. Olivier CAYOT

Mme Brigitte DUBUS

M. Jean-Marie NESSI

Mme Florence PICARD

SLJ
SN

Directeur de mémoire en entreprise :

Nom :

Signature :

Membres présents du jury du
Cnam :

pe *peints par Teaus*

M. Nathanaël ABECERA

M. Alexis COLLOMB

M. David FAURE

SN


**Autorisation de publication et de
mise en ligne sur un site de diffusion
de documents actuariels (après
expiration de l'éventuel délai de
confidentialité)**

Signature du responsable entreprise

Secrétariat :

Bibliothèque :

Signature du candidat



Jeu du minage, et sécurité du protocole Bitcoin

PAR CYRIL GRUNSPAN

26 janvier 2022

Résumé

Dans ce mémoire, nous considérons la sécurité et la stabilité de Bitcoin. Nous donnons un algorithme simple donnant le seuil minimal en terme de puissance de hachage relative au-delà duquel un mineur n'est plus incité à se comporter de manière honnête. Nous donnons une stratégie simple expliquant cette anomalie due à une faille dans la formule d'ajustement du paramètre de difficulté de minage dans le protocole Bitcoin. Nous donnons aussi une reformulation en terme de variation du jeu classique de pile ou face. Nous démontrons qu'une légère modification de Bitcoin qui prendrait en compte la production de blocs orphelins l'immuniserait contre d'éventuelles attaques de rétention de blocs. De telles analyses sont nécessaires si un assureur souhaite prendre part à l'économie des cryptomonnaies comme par exemple, couvrir les risques inhérents au fonctionnement d'une plateforme d'achat et de vente. Nous proposons plusieurs pistes à un assureur qui souhaiterait se positionner dans cet univers. Le nouveau monde de la finance décentralisée et l'incertitude naturelle liée aux contrats automatiques pas toujours exempts de bugs informatiques semblent un territoire plein d'opportunités. Enfin, nous expliquons le principe des log-contracts discrets avancés par Tadge Dryja. Il pourrait à l'avenir donner vie à une autre finance décentralisée sur Bitcoin en étendant les possibilités du « Lightning Network », une surcouche du réseau Bitcoin avec des performances inégalées en terme de vitesse et de débit des transactions. Les assurances paramétriques pourraient naturellement y trouver une place.

Mots-clefs: Bitcoin, Blockchain, Preuve de travail, Processus de décision markovien, Jeu de Pile ou Face, Consensus distribué

Abstract

In this paper, we consider the security and stability of Bitcoin. We give a simple algorithm giving the minimum threshold in terms of relative hash power beyond which a miner has no incentive to behave honestly. We give a simple strategy explaining this anomaly due to a flaw in the mining difficulty parameter adjustment formula in the Bitcoin protocol. We also give a reformulation in terms of a variation of the classical coin flip game. We show that a slight modification of Bitcoin that takes into account the production of orphan blocks would make it immune to possible block withholding attacks. Such analyses are necessary if an insurer wishes to take part in the cryptocurrency economy, such as covering risks inherent to the operation of an exchange platform. We propose several avenues for an insurer who would like to position itself in this universe. The new world of decentralized finance and the natural uncertainty linked to automatic contracts that are not always computer bug free seem to be a territory full of opportunities. Finally, we explain the principle of discrete log-contracts advanced by Tadge Dryja. In the future, it could give life to another decentralized finance on Bitcoin by extending the possibilities of the "Lightning Network", an overlay of the Bitcoin network with outstanding performance in terms of speed and transaction throughput. Parametric insurance could naturally find a place in this.

Keywords: Bitcoin, Blockchain, Proof of Work, Markov Decision Process, Coin tossing, Distributed Consensus

1 Introduction : les assureurs et les cryptomonnaies

La première apparition de bitcoin remonte au 18 août 2008. On a la preuve que ce jour-là une personne du nom de Satoshi Nakamoto a enregistré le nom de domaine bitcoin.org. Plus tard, sur le forum de discussion bitcointalk.org, son créateur confiera avoir travaillé sur ce projet depuis le printemps 2007. Tout laisse penser que des essais de fonctionnement de bitcoin ont été réalisés entre 2007 et 2008 avant que Satoshi Nakamoto ne rédige son papier fondateur et en fasse part au monde entier sur Metzdowd, une liste de diffusion pour cryptographes avertis, le 31 octobre 2008. Le code informatique a été ensuite rendu public en janvier 2009 et il est remarquable que mis à

part certains beugues mineurs inévitables corrigés pour certains avec l'aide d'Hal Finney, un célèbre cryptographe américain, le code de Bitcoin ait été opérationnel assez rapidement. Sur le fond, la plupart des techniques existaient déjà avant lui. Les preuves de travail remontent aux travaux de Cynthia Dwork et Moni Naor en 1992. L'idée de contrat automatique (« smart contract ») remonte aux écrits de Nick Szabo en 1994 et celle de blockchain (on parlait alors de « Timestamp server ») remonte aux travaux de Stuart Haber et W. Scott Stornetta en 1995. Mais la plupart du temps, ces idées n'étaient que des projets ou des brillants articles de recherche, rarement mis en pratique. Le génie de Satoshi Nakamoto a été de mettre toutes ces idées ensemble et de donner vie concrètement à une « cryptomonnaie » (j'utilise ce terme même si le G20 s'y refuse et préfère parler de « crypto-actif »). Le créateur du Bitcoin est avant tout un informaticien professionnel de grand talent. A cela, il faut ajouter une idée vraiment nouvelle : un algorithme de consensus (généralement appelé aujourd'hui consensus de Nakamoto) dans un univers dit « permissionless » où personne n'a besoin de demander d'autorisation à personne pour rentrer dans le réseau. Cette algorithme basé sur l'utilisation répétée de preuves de travail est la pierre fondatrice du Bitcoin. Elle permet la création d'une monnaie vraiment décentralisée. Plusieurs cryptomonnaies existaient avant bitcoin mais aucune n'était vraiment décentralisés et pour cette raison étaient vulnérables. Il est remarquable que toutes les tentatives de cryptomonnaies décentralisées avaient échoué avant bitcoin, ce qui a pu expliquer du reste le scepticisme au début de certains cryptographes sur Metzdowd.

Mais rapidement, Bitcoin a gagné en popularité. Hal Finney a vite été rejoint par plusieurs informaticiens de talent comme Gavin Andresen, Mike Hearn ou Peter Todd devenus chacun développeurs de Bitcoin Core - certains ont bien sûr quitté Bitcoin Core depuis. Les passionnés se retrouvaient sur bitcointalk un forum crée par Satoshi Nakamoto dès 2009. Là ont commencé à s'échanger réellement les premiers bitcoins qui étaient même parfois offerts au début ! C'est aussi là que certains utilisateurs ont commencé à s'interroger sur la sécurité réelle du protocole Bitcoin, voire à en proposer d'autres. En 2010, Bitcoin étaient déjà connu du grand public. En France, le premier article du journal « Le Monde » remonte semble-t-il à 2011. Puis, des premières plateforme d'échange de cryptomonnaies sont apparues. Bitcoin-Central, l'ancêtre de Paymium a par exemple été crée en 2011. Coinbase qui a aujourd'hui une taille comparable à celles des GAFAM a été crée un an après en 2012. Il est faux de croire que les Français soient passés à côté de l'avènement du Bitcoin et des cryptomonnaies. Certains étaient des pionniers mais faute de soutien, ou plutôt à cause de batons dans les roues, ils n'ont pas pu prospérer comme d'autres acteurs américains. En 2013, une plateforme, Mt Gox a fait faillite. En 2015, Vitalik Buterin, un passionné de bitcoin et l'un des premiers journalistes du magazine en ligne Bitcoin-Magazine a crée Ethereum, une autre cryptomonnaie s'appuyant sur une légère variation du consensus de Nakamoto. Les premiers ethers ont été préminés et proposés à l'achat. Aujourd'hui, la capitalisation de toutes les cryptomonnaies est supérieur à 2 000 milliards de dollars (en hausse) avec une prédominance de bitcoin d'environ 40%.

Dans un contexte où les taux directeurs sont toujours bas, certains assureurs imaginent investir dans les cryptomonnaies pour diversifier leur portefeuille.

1.1 L'avènement des blockchains décentralisées

1.1.1 L'adieu aux projets "blockchain" fantaisistes

Il était déjà question de « blockchain », on parle maintenant surtout de finance décentralisée et de cryptomonnaies. A bien des égards, cela peut sembler une régression puisque la « blockchain » présentée comme une technologie sous-jacente au bitcoin semblait pouvoir être utilisée pour résoudre toute sorte de problèmes y compris des problèmes sans rapport de près ou de loin avec la finance. On était alors en 2015. En une, *The Economist* parlait de machine à créer de la confiance (« The Trust Machine »). Blythe Masters, ancienne employée expérimentée de JP Morgan que l'on disait être à l'origine des CDS avait crée une start-up, « Digital Assets Holdings », qui avait réussie à lever 100 millions de dollars pour porter la nouvelle technologie aux banques d'investissements. Celles-ci s'étaient elles-mêmes regroupées dans une nouvelle association R3-CEV pour réfléchir ensemble au principe d'une nouvelle blockchain. De son côté, IBM mettait en avant Hyperledger, une solution « blockchain » à destination des entreprises qui pourraient créer ainsi facilement des "blockchains privées" dont elles garderaient le contrôle.

Que reste-t-il de tous ces projets ? Peu de choses. Blythe Masters ne dirige plus l'entreprise qu'elle a créée ; Goldman Sachs, Santander et JP Morgan ont quitté R3-CEV et IBM a cessé de développer Hyperledger mettant son équipe « blockchain » au chômage. Conformément au célèbre graphe du changement et des innovations, après l'immense attente de la « révolution blockchain » est venue immanquablement la terrible désillusion.

Certes, certains jusqu'au-boutistes » soutiennent encore le point de vue « blockchain oui, Bitcoin non » et songent même parfois à des bases de données plus générales fonctionnant grâce à des preuves de travail comme Bitcoin ou Ethereum. Difficile de leur donner tort sur le papier puisque même Stuart Haber et Scott Stornetta, les créateurs du concept de « blockchain » privée, l'avaient imaginé [7]. L'expérience récente a montré cependant que la plupart de ces projets n'ont pas tenu la route.

Et du reste, si l'on copie réellement Bitcoin, on ne comprend pas pourquoi des mineurs devraient dépenser de l'énergie pour sécuriser et enrichir une base de données s'ils ne sont pas rémunérés pour ce travail. Ce mécanisme semble mieux adapté à la gestion d'un registre de comptes distribué comme une cryptomonnaie. Par ailleurs, des systèmes distribués existent déjà depuis longtemps et existeront encore sans l'utilisation des preuves de travail. Internet notamment n'a pas attendu Bitcoin pour fonctionner.

1.1.2 Horodatage et Tokenization

Bien que le prix Nobel d'économie français Jean Tirole et ses partisans pour qui « le bitcoin est une bulle » puissent le regretter, il n'y a de fait pas ou plus de projet « blockchain » d'envergure qui ne s'appuie sur des registres publics comme Bitcoin ou Ethereum. La perte d'intérêt pour d'éventuelles « blockchains privées » ne semble pas non plus signifier exactement que la « blockchain » ne puisse être utilisée que pour effectuer des transactions financières. C'est véritablement le mirage de la blockchain privée qui semble passée de mode - de même qu'à moment donné, les entreprises ont cessé de vouloir absolument créer des réseaux de communication intranet privés. Puisque Bitcoin et Ethereum ont fait leur preuve, pourquoi ne pas simplement tirer profit de leurs capacités ? Ainsi Bitcoin et Ethereum sont utilisés naturellement pour horodater des documents. Sur Bitcoin, cela est possible depuis l'introduction du « Opcode » (commande informatique) « OP_RETURN ». Il permet en quelque sorte de laisser des traces non financières dans le registre du bitcoin, ce qui peut s'interpréter (toute proportion gardée) comme du vandalisme, un peu à la manière de certains touristes romains de l'Antiquité en visite en Egypte qui pouvaient paraît-il laisser des traces écrites sur les pyramides, encore visibles aujourd'hui. Le protocole Opentimestamp inventé par le développeur Peter Todd exploite cette possibilité de laisser des métadonnées dans des transactions bitcoin. Certaines entreprises se sont positionnées sur ce terrain de l'horodatage. Il ne s'agit ici que de délivrer des preuves d'existence. Sur Ethereum, on peut faire un peu plus : associer naturellement un jeton numérique unique à un document numérique (par exemple une photo numérique) qui lui-même peut être échangé, à la manière dont un objet peut changer de propriétaire. On utilise alors généralement le standard ERC 721 pour créer des jetons non-fongibles (NFT pour « Non Fungible Token »). C'est grâce à ce standard que des start-ups cherchent à construire une plateforme pour l'industrie du luxe. D'une certaine façon, l'existence de ces jetons ou « tokens » permet de réconcilier propriété intellectuelle et internet. Ainsi, des oeuvres d'art numériques ont pu être valorisées à plusieurs dizaines de millions de dollars. On rentre alors dans le monde de la DeFi sur Ethereum. L'apparition encore récente des sidechains sur Bitcoin comme Liquid ou RSK permet en théorie de développer une activité semblable dans l'univers « Bitcoin » mais ce n'est pas le cas aujourd'hui. Mais quelle que soit la blockchain utilisée, cette économie se construit sur une blockchain publique où circule naturellement une cryptomonnaie décentralisée (ou qui cherche à le devenir).

1.1.3 Un nouvel eldorado

En vérité, le discours radical de ces grands économistes paraît avoir beaucoup vieilli. Même les dirigeants officiels - américains surtout - semblent plus mesurés. Le boom des cryptomonnaies en 2021 n'y est pas étranger. Le prix du bitcoin a très fortement augmenté passant d'environ 10 000€ à 50 000€ en quelques semaines, une progression fulgurante. Des grandes entreprises, attirés par le nouvel eldorado des cryptomonnaies ou tout simplement pour diversifier leur trésorerie ne tournent

plus autour du pot. Pour ne donner qu'un exemple, Tesla a déclaré avoir acquis l'équivalent de 1.5 milliards de dollars en bitcoins... Les fonds d'investissement, toujours à l'affût de possibles opportunités de gain et sous la pression de certains épargnants n'hésitent pas non plus à prendre des positions sur le marché des cryptomonnaies. C'est à confirmer mais il semblerait que Blackrock, le plus gros gestionnaire d'actifs au monde ait aussi pris une position en futures sur Bitcoin pour plusieurs centaines de millions de dollars en bitcoin. Ce n'est rien à l'échelle de ce géant mais la prise de position est significative de cette nouvelle tendance. Même JP Morgan autrefois si négative envers Bitcoin souhaite pouvoir proposer un accès à ce nouveau marché pour ses clients. Dans le même sens, Goldman Sachs vient également de déposer une demande de création d'ETF à la SEC. Il semblerait d'ailleurs qu'il existe déjà trois ETF Bitcoin au Canada. Les banques d'investissement ne sont pas loin de créer des desks de trading de cryptomonnaies...

1.2 Le rôle des assureurs

Dans ce contexte, que peuvent faire les assureurs et quel est leur intérêt ?

1.2.1 Les assureurs cherchent à s'appropriier la technologie

D'abord, la « blockchain » n'est pas une nouveauté pour eux. Dans le sillage des banquiers, de grands assureurs et réassureurs tels AIG, Allianz, Scor, Swiss Re ou Generali ont créé l'association B3I (« the blockchain insurance industry initiative) qui s'est transformé en 2018 en véritable entreprise. Elle a créé récemment un produit d'assurance expérimental basé sur « Corda » la blockchain inventée par R3-CEV. Jusqu'à présent, c'est la transparence et l'automatisation des paiements qui retiennent l'attention des assureurs.

A travers des fonds de capital-risque (venture capital) dont ils font partie, les assureurs ont aussi participé à des levées de fonds de start-ups de l'écosystème blockchain. Ainsi CNP Assurance a investi plusieurs millions d'euros dans Stratum, une jeune pousse française. De manière plus significative, en 2016, AXA Strategic Ventures a investi près de 55 millions d'euros dans Blockstream, la plus prestigieuse entreprise de l'écosystème bitcoin (et aussi la plus prolifique) et qui est aujourd'hui valorisée à plusieurs milliards de dollars. Mais dans les deux cas, nous n'avons pas d'information sur la nature de l'investissement. S'agit-il de parts de la société acquises par les assureurs ? Ou bien ont-ils simplement acheté sur plusieurs années la R&D de ces sociétés pour s'approprier les nouvelles technologies ? Difficile de le savoir.

1.2.2 Investir dans les cryptomonnaies ?

Certains assureurs sont allés plus loin. Aux Etats-Unis, la vénérable *Massachusetts Mutual Life Insurance Co.* fondée au dix-neuvième siècle a acquis pour l'équivalent de 100 millions de dollars en bitcoins. Ses dirigeants assurent qu'ils ne s'agit que d'une « première étape » et se tiennent prêts à « explorer d'autres opportunités dans le futur ». Les taux directeurs des principales banques centrales étant extrêmement bas, voire négatifs, tous les autres taux d'intérêt le sont aussi. Il devient alors de plus en plus compliqué pour un assureur de tenir la promesse faite à ses assurés. D'où la tentation de se lancer dans de nouveaux marchés plus porteurs. L'un des fonds d'épargne retraite de NZ Fund en Nouvelle-Zélande a paraît-il acquis pour cette raison 200 millions d'euros en bitcoin.

En France, la nouvelle loi PACTE permet en théorie de suivre le même chemin. L'article 21 qui modifie l'article L113 du code des Assurances et l'article 26 qui modifie le Code Monétaire et Financier permettent maintenant aux assureurs de proposer à des fonds d'alternatifs des contrats utilisant bitcoin comme unité de compte et, en conséquence, les autorise à investir sur le marché des cryptomonnaies pour provisionner ces possibles contrats. Il convient tout de même de préciser que suivant Solvabilité II, les Assureurs restent naturellement soumis au principe de la « personne prudente » concernant leurs investissements.

Ajoutons que l'Autorité des Marchés Financiers vient de donner son feu vert (août 2021) pour qu'un gestionnaire d'actifs puisse créer un ETF lié au Bitcoin.

En Suisse, certains assureurs permettent aussi à leurs assurés de payer leurs cotisations en bitcoin.

1.3 Assurer la nouvelle finance

1.3.1 Assurer le cyber-risque des plateformes

Mais plus qu'un simple investissement dans le marché des cryptomonnaies, les assureurs ont d'autres opportunités comme celle d'assurer les actifs détenus par des grandes plateformes d'échange. Il est probable que toutes les grandes plateformes le font déjà même si l'information est difficile à obtenir. Nous savons que la nouvelle banque crypto.com qui permet à ses clients d'acquérir facilement certaines cryptomonnaies est assurée (au moins) à hauteur de 360 millions de dollars par *Arch Underwriting*, une division de la Lloyd's. La solution Ledger Vault de la société française Ledger a aussi passé un contrat avec l'assureur anglais pour une couverture d'actifs d'environ 150 millions de dollars. Ceci peut sembler paradoxal. Le Bitcoin est devenu populaire pour la confiance qu'il instaure sur son réseau. Il est notamment prouvé qu'en prenant quelques précautions élémentaires, aucune escroquerie n'est possible. Mais cela n'écarte pas cependant le risque de se faire voler ses clés secrètes. Les plateformes d'échange qui laissent en ligne leurs actifs sont de ce fait particulièrement vulnérables à ce risque de piratage. Même si seulement 1% de leur chiffre d'affaire est exposé en ligne, cela représente plusieurs centaines de millions de dollars. Les assureurs qui se positionnent depuis plusieurs années sur le cyber-risque sont naturellement en mesure d'assurer des grandes plateformes d'échange de cryptomonnaies.

1.3.2 Assurance paramétrique des « smart contracts »

Coinbase, un des nouveaux géants de cette nouvelle finance et désormais coté en bourse pourrait lui-même se transformer en assureur pour ses clients. Mais plus encore que l'assurance contre le piratage de certains portefeuilles en ligne, il nous semble que la finance décentralisée ou « Defi » définie comme un ensemble d'applications financières décentralisées construites principalement dans l'univers d'Ethereum offre des opportunités sans précédent pour les assureurs. Il ne s'agit pas ici d'assurer le vol de clés secrètes mais le risque d'erreurs (« bug ») dans des « smart-contracts ». La « Defi » constitue un écosystème remarquable et offre de nombreuses opportunités. En particulier, au lieu de garder son argent dans des portefeuilles « à froid », on peut le placer *à priori* sans risque dans des « pools » de liquidité qui permettent de faire fonctionner des teneurs de marché automatiques (« automated market makers »). L'argent placé est ainsi simplement rémunéré et permet en plus d'acquérir des jetons de gouvernance qui ont une valeur de marché. On peut aussi mettre en jeu (« staker ») son argent pour sécuriser une cryptomonnaie. Il existe ainsi de multiples raisons d'investir dans la « Defi ». Les rendements peuvent être très importants, ce qui explique son succès grandissant. Aujourd'hui, près de 80 milliards de dollars est investi dans des smart-contracts sur Ethereum. Un chiffre totalement spectaculaire : l'an dernier à peine 500 millions de dollars y étaient investis... Chaque semaine, on apprend de nouveaux piratages mais cela ne refroidit pas pour autant les investisseurs. C'est dans ce contexte qu'a émergé récemment la startup Nexus Mutual qui assure certains smart-contracts sur Ethereum. Il semble que ce soit la plus importante des start-ups proposant des produits d'assurance dans la finance décentralisée : 1% des actifs de la Defi serait assuré par Nexus Mutual. Les produits d'assurance sont de « simples » contrats d'assurances paramétriques qui se déclenchent suivant les variations d'un « oracle ». En soi, cette technique n'est pas étrangère aux assureurs. Plusieurs risques de catastrophes naturels utilisent déjà ce modèle. Enfin, le succès des NFT (« non fungible tokens ») qui permettent d'une certaine façon de concilier propriété intellectuelle et internet devrait aussi attirer les assureurs. Rien ne les empêche d'assurer un NFT de même qu'ils assurent des pièces de musée qui voyagent à l'occasion d'une exposition. Ces NFT peuvent aussi représenter certains pouvoirs spéciaux dans des jeux vidéos qui une richesse dont on peut souhaiter vouloir s'assurer contre le risque de perte. Avec les NFT, les assureurs ont probablement un avenir dans le monde des jeux vidéos.

Il nous paraît donc que les assureurs auraient intérêt à revoir leur position sur la « blockchain ». L'échec de l'assureur Fizzy qui promettait d'assurer des billets d'avion contre des retards de vol et de verser des dédommagements sur une blockchain peut nous éclairer. L'idée était certainement bonne mais malheureusement trop peu de clients a souscrit à un tel contrat.

L'avenir de l'assurance semble davantage prendre corps dans la finance décentralisée et les cryptomonnaies plutôt que dans des applications blockchains inexistantes.

Au fond, les deux mondes de l'assurance et des cryptomonnaies ne sont pas si différents l'un de l'autre. D'un point de vue technique, le problème du minage de cryptomonnaies peut du reste se voir comme un problème dual de la ruine en assurance. Au lieu de modéliser le flux positif d'arrivée de nouveaux contrats par un paramètre constant comme c'est souvent fait et les annonces négatives de nouveaux sinistres par un processus de Poisson, on peut modéliser les annonces positives de créations de nouveaux blocs par un processus de Poisson et le coût de minage par un paramètre constant. C'est ce point de vue que nous allons développer maintenant.

2 Assurer ?

Il semble que l'univers des cryptomonnaies soit plein d'opportunités pour les assureurs. Néanmoins, le milieu est-il vraiment sain ? Assurer des vendeurs qui vendent de la camelote à des clients n'a pas de sens, en plus d'être illégal. Fondamentalement, on doit savoir si on peut faire confiance à Bitcoin et aux autres cryptomonnaies. Il s'agit ici d'étudier la sécurité des protocoles décentralisés. Si un assureur doit assurer une plateforme d'échange qui vend des bitcoins ou une cryptomonnaie lambda, il doit au moins chercher à savoir si bitcoin ou lambda est de près ou de loin un actif ayant les propriétés d'une monnaie ou bien s'il ne s'agit que d'une arnaque, une « bulle ». L'assureur est ainsi amené à regarder de plus près la sécurité du protocole bitcoin ou lambda, à comprendre en profondeur le fonctionnement des cryptomonnaies.

La principale attaque sur un réseau de cryptomonnaies est l'attaque à la double dépense. Un utilisateur fut-il un mineur ne doit pas être en mesure de dépenser deux fois une même somme. L'analyse sur Bitcoin a été faite par Satoshi lui-même dans son papier fondateur. Il ressort qu'en prenant des précautions élémentaires, une transaction bitcoin est sûre ou quasiment sûre : on ne peut revenir en arrière ; ce qui a été ordonné ne peut être effacé. Plus exactement, la probabilité de réussite peut être rendu très faible. De plus, à moins de répéter sans arrêt des escroqueries sur des sommes gigantesques, un mineur n'a pas intérêt à se lancer dans une telle activité déviante. Il lui faudrait mobiliser énormément d'argent, ce qui est impossible en pratique. La résolution du problème de la double dépense par l'utilisation des preuves de travail est la principale prouesse de Bitcoin.

Mais il faut aussi savoir si le protocole est stable, c'est-à-dire si le registre des transactions (la "blockchain") progresse régulièrement d'un bloc toutes les dix minutes en moyenne pour Bitcoin avec des acteurs qui vont tous dans la même direction et cherchent à valider des blocs le plus rapidement possible et à les publier. Est-ce vraiment une conséquence des règles du protocole ? Pour un acteur quelconque, suivre celui-ci représente-t-il toujours la meilleure chose à faire ?

3 Résumé du réseau Bitcoin

Voici, sans rentrer dans les détails, une brève description du réseau Bitcoin [13]. Il est formé par des milliers de "noeuds" reliés en partie les uns aux autres, de sorte que le tout forme un graphe irréductible. Aucun acteur n'a un rôle supérieur aux autres. Tous jouent ou peuvent jouer le même rôle et réaliser les mêmes opérations. Chaque noeud possède trois bases de données locales :

- la blockchain,
- l'ensemble des UTXO,
- la mempool.

La blockchain est le registre de toutes les transactions confirmées. L'ensemble des UTXO est l'ensemble de toutes les "pièces" de monnaie en circulation. La mempool est l'ensemble des transactions passées mais non encore confirmées.

Chaque fois qu'un noeud reçoit une transaction, il examine si celle-ci est bien légale et cherche à déplacer de l'argent qui fait partie de l'ensemble des UTXO. Il vérifie aussi que la transaction ne rentre pas en conflit avec une autre transaction déjà présente dans sa mempool et qu'elle est de plus rédigée suivant les formes requises par le protocole. Si tel est le cas, la transaction est ajoutée à sa mempool et transmise aux noeuds voisins.

Parmi les noeuds du réseau, il en existe des “lourds” qui mènent en plus une activité de minage.

Un mineur est un noeud particulier qui cherche en plus à constituer un bloc. Par définition, un bloc \mathcal{B} est un ensemble de données dont la taille est d’environ 2 Méga Bytes (pour Bitcoin). Il est formé d’une référence à un ancien bloc, d’un ensemble de transactions piochées dans la mempool du mineur, d’une date de création t , de la difficulté de minage Δ et d’un paramètre appelé “nonce”. Dans les présentations du Bitcoin, on raconte souvent que le nonce est là pour créer de l’aléa afin que la relation suivante soit vérifiée :

$$h(\mathcal{B}) < \frac{1}{\Delta} \quad (1)$$

où h est la fonction de hachage $\text{SHA}_{256} \circ \text{SHA}_{256}$ (SHA_{256} est une fonction de hachage ou fonction dite à sens unique recommandée par l’Agence Nationale de Sécurité Américaine (NSA)). La relation (1) est le critère adopté par tous les noeuds du réseau qui permet d’affirmer que le bloc \mathcal{B} est valide. Vu la difficulté du problème, l’existence de \mathcal{B} est en soi une « preuve de travail ». Avant de trouver \mathcal{B} , le mineur a du effectivement dépenser beaucoup de temps et d’énergie à tester des tas de solutions possibles.

Ce qui est écrit ci-dessus n’est pas faux mais en vérité, le nonce qui avait peut-être un intérêt au début du Bitcoin ne sert plus à rien. Il est trop petit (4 octets seulement) et on peut suffisamment créer de l’aléa autrement, par exemple en permutant toutes les transactions rangées dans les feuilles d’un arbre de Merkle ou bien en faisant varier un “extra-nonce” caché dans la syntaxe du script associé à une transaction particulière dénommée “coinbase” et qui représente une création monétaire accordée au mineur pour avoir réussi à découvrir un nouveau bloc.

La difficulté Δ est ajustée tous les 2016 blocs (pour Bitcoin) de sorte qu’en moyenne, le réseau mette 10 minutes pour valider un bloc. A chaque ajustement de difficulté, le réseau calcule le temps T mis pour valider la dernière série de 2016 blocs grâce à l’horodatage des blocs. D’où l’intérêt (c’est le seul) d’inscrire la date de création d’un bloc dans son en-tête (le paramètre t évoqué plus haut). Si ce temps T est supérieur à 14 jours = 2016×10 minutes, la difficulté diminue. Dans le cas contraire, elle augmente. Concrètement, le nouveau paramètre de difficulté Δ' est :

$$\Delta' = \Delta \times \frac{2016 \times 10}{T} \quad (2)$$

où T est ici calculé en minutes (voir ci-dessous pour plus d’explications sur l’impact de cette formule).

La suite des blocs forme une chaîne de blocs ou “blockchain” suivant le terme employé la première fois par Hal Finney, un cryptographe américain connu notamment pour avoir été le premier bénéficiaire d’une transaction Bitcoin. De manière technique, il s’agit d’une chaîne listée de pointeurs de hashes.

Lorsqu’il a trouvé un nouveau bloc, le mineur transmet sa découverte aux noeuds avec qui il est connecté afin qu’ils le propagent. Un bloc met souvent plus d’une minute pour atteindre 90% du réseau.

Un noeud qui reçoit un bloc vérifie qu’il est légal (il vérifie notamment que la relation (1) est vérifié. Le cas échéant, il met à jour sa base de données des UTXO et vide une partie de sa mempool.

Enfin, et non le moindre, un noeud qui reçoit deux chaînes de blocs distinctes considérera comme officielle celle qui maximise $\sum \Delta_i$, autrement dit, la blockchain la plus solide, celle qui a été la plus compliquée à construire. La blockchain ayant cette particularité est appelée la blockchain officielle. En pratique, il s’agit de la blockchain la plus longue car la suite des difficulté (Δ_i) est localement constante. En outre, entre deux blocs de même hauteur (la hauteur d’un bloc \mathcal{B} est le nombre de blocs qui sépare \mathcal{B} du bloc initial), un noeud privilégiera toujours le premier bloc reçu.

Tel est plus ou moins le protocole Bitcoin décrit dans le papier fondateur de Satoshi Nakamoto [8].

Dans la dernière section de cet article, on trouve une preuve mathématique du fait que la probabilité de réussite d’une double-dépense est très faible pourvu qu’on prenne des précautions élémentaires. Ce résultat est spectaculaire. Jamais avant Satoshi on n’avait réussi à surmonter le problème de la double-dépense dans un réseau distribué décentralisé et ouvert (en libre accès i.e., « permissionless »). Au lieu de trouver un réseau qui assure a priori qu’il n’y a pas de double dépense possible, Satoshi fonde la sécurité du réseau sur la théorie des probabilités.

Autre prouesse réalisée par Bitcoin : l'avènement des contrats automatiques ou "smart-contracts" qui avaient été imaginés par le cryptographe américain Nick Szabo [12]. Dans l'univers Bitcoin, un exemple de "smart-contract" est une simple transaction bitcoin. Pour comprendre, on définit un output (TXO) comme la donnée d'un montant (en bitcoin) et d'une condition pour pouvoir dépenser cet argent. De manière classique, cela peut-être : prouver qu'on possède la clé secrète dont le hash de la clé publique (appelé aussi adresse bitcoin) est connu. C'est de cette façon que l'on peut prétendre posséder des bitcoins. Un output a deux états possibles : dépensé ou non-dépensé. Un input est au contraire une référence à un output et la preuve que la condition associée est satisfaite. L'input est formé d'un script libérateur (il permet de "libérer" de l'argent bloqué). L'output a au contraire la forme d'un script bloquant (il donne les conditions pour qu'une certaine somme d'argent puisse être dépensée). Une transaction est formellement un ensemble d'inputs et d'outputs avec la condition que l'argent dépensé (la somme des montants des inputs) est supérieur à l'argent envoyé (la somme des outputs). La différence entre les deux ira au mineur qui inscrira cette transaction dans un bloc de la blockchain officielle. Les inputs et outputs ont la forme de code informatique. Le langage de programmation Bitcoin est volontairement sommaire. C'est un langage dit "à pile". La partie output remplit la pile (les bitcoins qui vont être envoyés). Plus la transaction est compliquée, plus la pile est haute. Au contraire, la partie output mise à côté, vide la pile. L'oeil exercé peut parfois deviner la nature d'une transaction Bitcoin en examinant le code informatique. En particulier, le langage de programmation utilisé n'est pas Turing-complet (il ne permet pas de réaliser des boucles), ce qui apporte une grande sécurité au réseau monétaire : bien que toujours possible, les bugs sont souvent facilement détectables. Signalons les mises à jour récentes du Bitcoin qui apportent plus d'anonymat aux transactions en rendant indiscernables des contrats compliqués (nécessitant plusieurs signatures) par rapport aux simples transactions (qui ne requièrent qu'une seule signature).

Note 1. Le fonctionnement d'Ethereum est assez semblable à celui de Bitcoin : il s'appuie sur l'utilisation répétée de preuves de travail. Il y a néanmoins plusieurs différences notables. Ethereum impose un temps de minage interblocs très réduits de l'ordre de 10-15 secondes, ce qui conduit inévitablement à des problèmes de synchronisation et des mineurs qui risquent de miner des blocs ignorés par la majorité (de tels blocs sont dits orphelins). Pour contrer ce risque et inciter tous les mineurs à prendre quand même part au réseau, Ethereum rémunère même les créateurs de blocs orphelins (sous certaines conditions facilement vérifiées en pratique et dans une moindre proportion que s'il s'agissait d'un bloc normal). La blockchain officielle elle-même se définit en tenant compte de tous ces éventuels blocs orphelins. Autre différence : le langage de programmation des smart-contracts (Solidity) est un langage moderne permettant de réaliser des boucles qui fonctionnent grâce à un compilateur. Dans ce cas, auditer un smart-contract sur Ethereum est potentiellement beaucoup plus compliqué que sur Bitcoin. Les smart-contracts sur Ethereum permettent de créer l'univers de la « Defi » (finance décentralisée) sur Ethereum mais expliquent aussi la présence persistante de beugues découverts chaque semaine.

4 Sécurité et stabilité du protocole

4.1 Double dépense et répétitions

L'analyse menée par Satoshi est correcte (son calcul dans sa dernière section est approximatif mais peut-être facilement corrigé [4]). Cependant, la probabilité de réussite d'une escroquerie à la double dépense n'est qu'un élément de la sécurité du réseau. En effet, dans son étude, Satoshi considère la possibilité de dépenser deux fois une certaine transaction donnée TX. L'attaque commence une fois que l'attaquant a envoyé TX (en vérité, pour être cohérent avec les calculs, on doit supposer que l'attaquant a déjà secrètement préminé un bloc avant de commencer son attaque). L'attaque est réussie si le mineur parvient à produire une chaîne de blocs plus longue que la blockchain officielle ne contenant pas TX et après que TX ait reçu 6 ou 7 confirmations. Ceci signifie qu'un bloc de la blockchain officielle contient TX et que 6 ou 7 blocs ont été ensuite ajoutés à la blockchain officielle depuis la parution de ce bloc. Satoshi montre que sous des hypothèses raisonnables en terme de puissance de hachage relative du mineur, la probabilité de réussite de cet événement est faible. Cependant, l'attaquant pourrait prendre de l'avance et ne commencer son attaque qu'après avoir

déjà préminé 6 ou 7 blocs d'avance sur la blockchain officielle. La réussite de son attaque serait alors certaine. Un tel scénario est en principe possible. En effet, s'il ne publie pas ses blocs découverts, l'avance éventuelle que peut posséder un mineur sur la blockchain officielle peut se modéliser par une chaîne de Markov simple sur \mathbb{N} . L'état $\{n\}$ du mineur ($n \in \mathbb{N}$) correspond à une avance de n blocs sur la blockchain officielle et la probabilité d'augmenter cette avance est $q < \frac{1}{2}$. La chaîne de Markov est naturellement absorbante en 0. Son « avance » ne peut être négative ; s'il a du retard, le mineur revient miner sur le dernier bloc de la blockchain officielle. Cette chaîne de Markov est irréductible : chaque état est atteignable à partir d'un autre. Il apparaît donc clairement que la possibilité d'atteindre un niveau n quelconque est certain. Tôt ou tard, le mineur parviendra à une avance de 6 ou 7 blocs et, s'il commence son attaque à ce moment-là, celle-ci sera toujours réussie. Donc, en théorie, en attendant suffisamment longtemps, l'attaquant pourra toujours réussir une attaque à la double-dépense avec probabilité 1 ! Ainsi, la probabilité de réussite d'une double-dépense calculée par Satoshi n'est qu'un élément parmi d'autre à prendre en compte pour évaluer la sécurité du réseau Bitcoin. Il faut aussi considérer le rendement de l'attaque en considérant qu'elle a lieu régulièrement. Si ce rendement est supérieur au rendement que le mineur pourrait gagner en minant honnêtement, alors il y a un problème : le mineur n'est pas incité à être honnête. Concernant l'attaque à la double dépense, on peut montrer que sous certaines hypothèses raisonnables, ce n'est effectivement le cas : un mineur n'a pas d'intérêt à répéter des attaques à la double-dépense [6].

4.2 Sur l'incitation des noeuds à être honnête

De manière générale, puisque rien ne peut être imposé par la force, tout ne doit être qu'une question d'incitation. Les simples noeuds qui n'ont pas d'activité de minage sont en réalité les seuls à n'avoir pas d'intérêt financier à prendre part au réseau. Cela leur coûte de l'argent en terme de connexion, de mémoire et de bande passante. Ils le font par conviction, pour faire vivre le réseau. Leur importance qui n'avait peut-être pas été perçue par Satoshi à l'origine est loin d'être négligeable. Dans certains cas, ils peuvent voter pour activer une éventuelle mise à jour du protocole proposée par les développeurs (« User Activated Soft Fork »). Ces milliers de noeuds sont en quelque sorte les étendards de la communauté Bitcoin.

Pour étudier la sécurité d'un protocole, il faut passer en revue chacune des règles prescrites et s'assurer qu'elles correspondent bien pour un acteur quelconque à maximiser son profit : tout comportement déviant ne pourrait que le conduire à perdre de l'argent.

Il y a notamment un point sur lequel la communauté Bitcoin s'est rapidement interrogée : pourquoi les mineurs devraient-ils diffuser immédiatement un nouveau bloc découvert ? C'est sous-entendu dans l'article fondateur du Bitcoin. Néanmoins, le fait qu'un mineur découvre un bloc ne pourrait-il pas lui donner la possibilité de creuser encore plus l'écart en minant secrètement sur ce bloc découvert ? Rapidement des passionnés du Bitcoin se sont posés la question. Il y a un fil sur le forum historique bitcointalk.org initié dès 2010 par le pseudonyme « RHorning » [9]. Le problème a été partiellement traité par Meni Rosenfeld [10]. Puis en 2013, deux articles indépendants ont conclu que les règles du protocole Bitcoin n'étaient pas en conformité avec les intérêts des utilisateurs. Plus exactement, elles pouvaient être en conflit. Et ce point pouvait faire l'objet d'attaques dites de rétention de blocs. Ci-dessous, nous présentons une attaque très simple qui permet de comprendre ce phénomène.

4.3 Performance d'une stratégie de minage

4.3.1 Revenu et coût par unité de temps

Sur la durée, une stratégie de minage est toujours répétitive. Quels que soient les choix du mineur, à moment donné, il reviendra toujours à son point de départ à miner tout comme les honnêtes mineurs sur le dernier bloc de la blockchain officielle. On dit alors que le mineur a réalisé un cycle d'attaque [5]. On note τ la durée aléatoire d'un tel cycle et G le gain de mineur accumulé au cours de ce cycle (G est par exemple mesuré en terme de bitcoins ou de nombre de blocs minés par le mineur et ajoutés à la blockchain officielle). Si un mineur répète n fois sa stratégie, avec des notations évidentes, il va gagner par unité de temps

$$\frac{G_1 + \dots + G_n}{\tau_1 + \dots + \tau_n} = \frac{\frac{G_1 + \dots + G_n}{n}}{\frac{\tau_1 + \dots + \tau_n}{n}}$$

qui converge vers $\frac{\mathbb{E}[G]}{\mathbb{E}[\tau]}$ (en supposant $\mathbb{E}[\tau] < \infty$). De la même façon, le coût par unité de temps du mineur sur le long terme est $\frac{\mathbb{E}[C]}{\mathbb{E}[\tau]}$ où C est le coût par cycle de son activité de minage et le revenu net du mineur par unité de temps est $\frac{\mathbb{E}[G]}{\mathbb{E}[\tau]} - \frac{\mathbb{E}[C]}{\mathbb{E}[\tau]}$. Or, le point est que le coût de son activité de minage par unité de temps ne dépend pas du fait que le mineur garde secret ou non des blocs sur un ordinateur (il dépend du coût de l'électricité, du prix de son matériel, des salaires versés aux employés, etc.). Qu'il rende ou non public des blocs n'a pas d'impact sur ce coût par unité de temps. Par suite, entre deux stratégies de minage (ayant même coût de fonctionnement en moyenne par unité de temps), un mineur rationnel choisira la stratégie qui maximise son revenu par unité de temps sur le long terme $\Gamma = \frac{\mathbb{E}[G]}{\mathbb{E}[\tau]}$.

4.3.2 Effet de l'ajustement de difficulté sur le revenu par unité de temps

Reprenons la situation ci-dessus où un mineur répète une même stratégie face au reste du réseau composé exclusivement d'honnêtes mineurs. On suppose que la puissance de hachage totale reste la même. Le paramètre de difficulté de minage imposé par le protocole va être modifié par l'attitude du mineur. Au bout d'une période de minage de 2016 blocs officiels, en présence d'un mineur adoptant une stratégie de rétention de blocs qui ralentit la progression naturelle de la blockchain officielle, le réseau va en général mettre plus longtemps que les 14 jours prévus, ce qui va donner lieu à un ajustement de difficulté à la baisse. Puis, au cours du temps, ce paramètre va se stabiliser. Reprenons la formule donnant le l'ajustement de difficulté donné plus haut

$$\Delta' = \Delta \cdot \frac{2016 \times 10}{T} \quad (3)$$

où T est le temps mis pour valider les 2016 blocs ajoutés à la blockchain officielle. Au cours du temps, le paramètre de difficulté va se stabiliser (Δ et Δ' ont la même distribution), ce qui entraîne qu'en moyenne, la formule impose une durée de minage entre la validation de deux blocs officiels égale à 10 minutes et ceci, quelle que soit la stratégie de minage choisie.

Notation 2. *En fait, il y a un léger biais du au fait que $\frac{1}{T}$ soit une loi Gamma-Inverse et donc que $\mathbb{E}[\frac{1}{T}] \neq \frac{1}{\mathbb{E}[T]}$. De plus, T mesure en réalité le temps mis pour valider 2015 blocs et non 2016 (il s'agit d'un beugue mineur dans le code Bitcoin). Cela entraîne un temps de minage interblocs cible d'un peu plus de 10 secondes. Le calcul exact donne 10 minutes et 1.19 secondes.*

La durée de minage entre validation de deux blocs officiels étant de 10 minutes, la durée d'un cycle d'attaque est donc proportionnelle en moyenne à la progression de la hauteur de la blockchain officielle durant ce cycle : $\mathbb{E}[\tau] = \mathbb{E}[H] \times 10$ où H désigne la progression de la hauteur durant ce cycle d'attaque. C'est une conséquence du théorème de Wald. Il en résulte que le revenu par unité de temps sur le long terme du mineur est proportionnel à $\frac{\mathbb{E}[G]}{\mathbb{E}[H]}$. C'est donc la quantité importante qui sert à départager deux stratégies de minage ; le mineur rationnel choisira la stratégie qui maximise $\Gamma = \frac{\mathbb{E}[G]}{\mathbb{E}[H]}$.

Notons que la stratégie honnête correspond à un cycle qui prend fin dès qu'un bloc est découvert. Dans ce cas, on a $\mathbb{E}[G] = q$ et $\mathbb{E}[H] = 1$ où q est la puissance de hachage relative du mineur. Donc, $\Gamma = q$ pour la stratégie honnête et un mineur rationnel a intérêt à adopter une stratégie déviante si et seulement si $\Gamma > q$.

4.3.3 Effet d'une autre formule d'ajustement de difficulté sur un réseau Bitcoin modifié.

Imaginons une autre formule d'ajustement de difficulté le réseau Bitcoin qui obéirait régulièrement à cette nouvelle formule :

$$\Delta' = \Delta \cdot \frac{D \times 10}{T} \quad (4)$$

où D désigne la progression d'une certaine quantité appelée fonction difficulté, au cours d'une période de validation de 2016 blocs officiels. Dans le cas de Bitcoin utilisé en pratique, on a $D = 2016$ comme on l'a vu. Autrement dit, cette « fonction de difficulté » pour Bitcoin « normal » augmente de 1 à chaque enregistrement d'un bloc sur la blockchain officielle. Autrement dit, cette fonction de difficulté n'est autre que la hauteur de la blockchain elle-même dans le cas Bitcoin « classique ».

Mais on imagine ici que D désigne une fonction de difficulté quelconque qui, comme la hauteur, ne puisse qu'augmenter au cours du temps. On peut par exemple considérer que le réseau prenne en compte la production de blocs orphelins et qu'entre deux événements sur le réseau, la fonction de difficulté progresse d'autant de blocs découverts détectés (blocs officiels ou blocs orphelins). C'est pratiquement déjà le cas sur Ethereum qui, en réalité, ne prend pas en compte tous les blocs orphelins détectés mais seulement la production de blocs appelés « oncle » qui ont un parent direct dans la blockchain officielle. Avec cette nouvelle formule, le temps de minage d'un cycle devient maintenant proportionnel à la progression de la fonction difficulté.

Le même raisonnement que précédemment montre que pour Bitcoin modifié, le revenu par unité de temps sur le long terme devient $\Gamma = \frac{\mathbb{E}[G]}{\mathbb{E}[D]}$ où D désigne la progression de la fonction difficulté au cours d'un cycle d'attaque (au lieu de $\Gamma = \frac{\mathbb{E}[G]}{\mathbb{E}[H]}$).

4.4 Connectivité du mineur

La connectivité, généralement notée γ mesure la capacité du mineur à pervertir les mineurs honnêtes. Elle quantifie l'emprise qu'il a sur eux. Concrètement, il s'agit d'une probabilité : en cas de compétition entre deux blocs de même hauteur (l'un étant miné par l'attaquant et l'autre par un mineur honnête), la probabilité qu'un mineur honnête cherche à miner un bloc sur le bloc de l'attaquant est γ . Il convient de préciser que dans le cas du bloc de l'attaquant, il s'agit d'un bloc qu'il avait déjà miné et qu'il gardait secret mais qu'il propage dès qu'il découvre que d'autres mineurs ont trouvé un bloc de même hauteur.

Normalement, sur Bitcoin, une telle compétition ne devrait pas avoir lieu puisqu'un mineur doit toujours miner sur le dernier bloc reçu. Mais il peut y avoir des problèmes de connexion et un bloc peut mettre près d'une minute à parcourir le réseau. Donc, si l'attaquant a déjà miné un bloc tenu secret A et qu'il observe la parution d'un bloc B de même hauteur que A, il peut profiter d'une bonne connectivité pour diffuser rapidement A et faire croire à une partie du réseau qu'il vient de trouver ce bloc A alors qu'il l'avait déjà découvert plusieurs minutes auparavant... Le réseau sera alors divisé : une partie croira que le dernier bloc de la blockchain officielle est A et l'autre que c'est B. Le conflit prendra fin à la découverte d'un nouveau bloc qui s'imposera à tous à moins que de nouveau, l'attaquant ne réagisse en publiant rapidement un bloc secret de même hauteur...

En particulier, si la connectivité du mineur est 1, il est libre de publier ou non un bloc qu'il vient de découvrir puisqu'il sera toujours sûr de l'enregistrer dans la blockchain officielle. S'il détecte la présence d'un autre bloc de même hauteur qu'un de ses blocs tenus secrets de même hauteur, il lui suffit de réagir en publiant rapidement son bloc secret qui finira toujours par s'imposer dans la blockchain officielle.

Il en résulte que si la connectivité du mineur est non-nulle, il est illusoire d'espérer que la meilleure stratégie soit la stratégie honnête pour Bitcoin tel qu'il existe aujourd'hui. En particulier, si $\gamma = 1$, l'attaquant peut se contenter de ne faire que réagir aux parutions des honnêtes mineurs.

Pour toute valeur de γ , il existe un seuil en terme de puissance de hachage relative au-delà duquel un mineur de connectivité γ a intérêt d'adopter une stratégie déviante. Clairement la fonction $\gamma \mapsto q_\gamma$ est décroissante. Donc, pour étudier la cohérence d'un protocole donné (en l'occurrence le protocole Bitcoin ici), on peut supposer naturellement que $\gamma = 0$.

Cette hypothèse $\gamma = 0$ correspond au pire scénario pour un attaquant. Si la meilleure stratégie de minage n'est pas la stratégie honnête dans le cas où $\gamma = 0$ alors les règles du protocole ont un problème qu'il faut chercher à modifier. Notons par ailleurs que le réseau évolue constamment au cours du temps avec des noeuds qui se déconnectent ou se déconnectent de sorte qu'en réalité il est illusoire d'imaginer un paramètre γ constant.

Nous faisons maintenant l'hypothèse $\gamma = 0$, pire scénario pour l'attaquant.

4.5 Un exemple simple : la stratégie « 1+2 »

Voici un exemple très simple permettant de comprendre pourquoi la stratégie honnête n'est pas toujours la plus rentable même lorsque la connectivité du mineur est égale à 0.

Supposons que le réseau soit composé d'un attaquant noté Alice et d'un ensemble de mineurs honnêtes qui suivent les règles édictées par Satoshi Nakamoto. L'attaque est définie de la façon suivante. Alice commence à miner. Si les honnêtes mineurs trouvent un bloc avant Alice, l'attaque prend fin. Si par contre Alice réussit à miner un bloc \mathcal{B} avant les honnêtes mineurs, alors Alice garde secret \mathcal{B} et continue à miner secrètement par dessus ce bloc. Ensuite, quelle que soit l'identité des mineurs ayant validé les blocs suivants, dès que deux blocs ont été découverts après la découverte de \mathcal{B} , l'attaque prend fin. Si elle a l'avantage, c'est-à-dire si elle a miné plus de blocs que les honnêtes mineurs, Alice révèle ses blocs secrets et impose son « fork » (sa suite de blocs) à la blockchain officielle qui procède alors à une petite réorganisation. Si ce n'est pas le cas, inutile pour elle de diffuser quoique ce soit puisqu'aucun de ses blocs ne deviendra officiel. En notant par A un bloc découvert par Alice et par B un bloc découvert par les honnêtes mineurs, l'attaque a donc la forme d'un mot formé avec les lettres A et B . La stratégie « 1+2 » est précisément cette attaque répétée un grand nombre de fois. Le « 1 » dans le nom de la stratégie s'explique par le fait qu'on attend de découvrir un bloc. Puis, lorsque c'est le cas, on attend ensuite que deux blocs soient découverts. D'où le « +2 ».

Exemple 3. Alice mine un bloc. Puis les honnêtes mineurs en mine un aussi et enfin Alice en mine un autre. La suite de blocs associée est ABA . Dans ce cas, Alice a réussi son attaque et propage ses deux blocs gardés secrets (les deux « A »). La blockchain officielle procède à une légère réorganisation : son désormais avant-dernier bloc (le bloc « B ») a été remplacé (il est désormais orphelin) et la hauteur de la blockchain a progressé d'un bloc au moment où Alice publie ses deux blocs. Au final, Alice gagne la récompense contenue dans deux blocs.

L'univers Ω de tous les résultats possibles à l'issue d'un cycle d'attaque est :

$$\Omega = \{B, AAA, AAB, ABA, ABB\}$$

Notons q la puissance de hachage relative d'Alice. C'est la probabilité qu'elle a de découvrir un bloc avant les honnêtes mineurs. De même, on note $p = 1 - q$ la puissance de hachage relative des honnêtes mineurs. Notons par G le nombre de blocs validés par Alice durant son attaque et qui finissent par rejoindre la blockchain officielle. C'est le revenu d'Alice suite à son attaque. Soit H le nombre de blocs officiels ajoutés à la blockchain officielle à la suite de l'attaque. C'est la progression de la hauteur de la blockchain officielle entre le début et la fin de l'attaque. On a

$$\mathbb{P}[B] = p, \mathbb{P}[AAA] = q^3, \mathbb{P}[AAB] = \mathbb{P}[ABA] = p q^2, \mathbb{P}[ABB] = p^2 q$$

avec

$$G(B) = G(ABB) = 0, G(AAA) = 3, G(AAB) = G(ABA) = 2$$

et

$$H(B) = 1, H(ABB) = H(AAB) = H(ABA) = 2, H(AAA) = 3$$

Donc,

$$\begin{aligned} \mathbb{E}[G] &= p \cdot 0 + q^3 \cdot 3 + p q^2 \cdot 2 + p q^2 \cdot 2 + p^2 q \cdot 0 \\ &= 3q^3 + 4p q^2 \\ &= q^2 \cdot (3q + 4p) \\ &= q^2 \cdot (3 + p) \\ &= q^2 \cdot (4 - q) \end{aligned}$$

et de même,

$$\begin{aligned} \mathbb{E}[H] &= p \cdot 1 + q^3 \cdot 3 + p q^2 \cdot 2 + p q^2 \cdot 2 + p^2 q \cdot 2 \\ &= p + 3q^3 + 4p q^2 + 2p^2 q \\ &= p + q \cdot (3q^2 + 4p q + 2p^2) \\ &= p + q \cdot (q^2 + 2 \cdot (p + q)^2) \\ &= p + q \cdot (q^2 + 2) \\ &= p + q + q + q^3 \\ &= 1 + q + q^3 \end{aligned}$$

Donc, le rendement de la stratégie « 1+2 » est $\frac{q^2 \cdot (4-q)}{1+q+q^3}$.

Par suite, la stratégie « 1+2 » est plus rentable que la stratégie honnête si et seulement si $\frac{q^2 \cdot (4-q)}{1+q+q^3} > q$. Après calcul, cette relation équivaut à

$$q > \sqrt{2} - 1 \quad (5)$$

Ainsi, si un mineur dispose d'un peu plus de 41% de puissance de hachage, il n'a pas intérêt à suivre le protocole. On peut du reste montrer que cette stratégie est la meilleure possible si l'on impose que le cycle d'attaque du mineur prend fin au plus immédiatement après la découverte de trois blocs (voir Section 6.3).

On voit donc avec cette stratégie déviante très simple que les règles du protocole peuvent aller à l'encontre des intérêts des mineurs. Il y a un seuil en terme de puissance de hachage au-delà duquel le mineur n'a pas intérêt à suivre le protocole. On vient de voir que ce seuil est inférieur à $\sqrt{2} - 1 \approx 41,4\%$.

5 Un algorithme pour calculer le taux de rendement maximal d'un mineur dans le réseau Bitcoin

Note 4. Tout ce chapitre peut être sauté en première lecture.

Le fait que la production de blocs orphelins ne soit pas pris en compte dans la formule d'ajustement de difficulté de Bitcoin - et donc que le taux de hachage réel de tout le réseau soit de fait sous-estimé - entraîne naturellement un ajustement à la baisse de la difficulté de minage. En présence d'un attaquant disposant de suffisamment de puissance de hachage relative que l'on notera q dans toute la suite (sa connectivité est généralement notée γ), cette baisse peut-être significative. Il peut carrément se retirer du réseau et miner ailleurs (par exemple sur Bitcoin Cash). Il peut aussi se lancer dans une attaque dite de « rétention de blocs » qui, à terme, et dans certains cas, peut devenir plus rentable que la stratégie de minage dite « honnête » consistant à toujours miner sur le dernier bloc découvert par le réseau. Suivant q et γ , le taux de hachage apparent sur le long terme de la stratégie déviante (définie comme étant la proportion de blocs validés par l'attaquant dans la blockchain officielle) peut devenir supérieure à q qui représente le taux de hachage apparent normal de la stratégie « honnête ». Le but de cette section est d'établir un algorithme efficace permettant de calculer le taux de hachage apparent - long terme - maximal dans le cas où $\gamma = 0$.

5.1 Un processus de décision markovien

Pour simplifier, nous considérerons que la connectivité du mineur est nulle : $\gamma = 0$. L'état du système est décrit par un couple $(a, h) \in \mathbb{N}^2$ où a désigne le nombre de blocs secrets miné(s) par l'attaquant depuis le dernier bloc officiel et reconnu comme tel par l'attaquant, c'est-à-dire qu'il ne cherchera pas à le remplacer. De même h désigne le nombre de blocs minés par les honnêtes mineurs depuis ce dernier bloc commun à la blockchain officielle et au fork de l'attaquant.

Supposons que le système soit dans l'état (a, h) avec $a > h$. Alors, l'attaquant a le choix entre miner secrètement ou publier $h + 1$ blocs tenus secrets, ce qui a pour effet d'écraser les h derniers blocs de la blockchain officielle.

- S'il décide de miner secrètement, alors le système évolue vers l'état $(a + 1, h)$ avec probabilité q ou $(a, h + 1)$ avec probabilité $p = 1 - q$.
- S'il décide de publier $h + 1$ blocs tenus secret, alors l'état du système devient égale à $(a - h - 1, 0)$ avec probabilité 1.

De même, si $a \leq h$, l'attaquant a le choix entre miner secrètement ou abandonner, ce qui a pour effet de ramener le système à son état initial.

- S'il décide de miner secrètement, alors le système évolue vers l'état $(a + 1, h)$ avec probabilité q ou $(a, h + 1)$ avec probabilité $p = 1 - q$.
- S'il décide d'abandonner, alors l'état du système revient à $(0, 0)$ avec probabilité 1.

Autrement dit, l'état du système évolue de manière aléatoire suivant une loi de probabilité qui dépend du choix du mineur. Par exemple, s'il abandonne (avec $a \leq h$) alors la probabilité de se retrouver dans l'état $(0, 0)$ est 1. S'il décide de miner secrètement, alors l'état du système évolue suivant le résultat d'une pièce de monnaie truquée (p en faveur des honnêtes mineurs et q en faveur de l'attaquant). Notons que si l'attaquant a l'avantage, on considère qu'il n'a pas la possibilité d'abandonner. On écarte a priori les comportements suicidaires qui ne sont pas optimaux.

Si le choix du mineur ne se base que sur l'état du système au moment considéré (cas markovien), alors un choix d'actions s'identifie à une simple fonction $\mathbf{a}: \mathbb{N}^2 \rightarrow \{0, 1\}$ avec $\mathbf{a}(a, h) = 0$ si le système est dans l'état (a, h) et que le mineur décide d'abandonner (possible si $a \leq h$) ou de rendre public $h + 1$ blocs (possible si $a > h$). Le cas $\mathbf{a}(a, h) = 1$ correspond au cas où le mineur décide de miner secrètement. Au départ, le mineur cherche à miner un bloc au dessus du dernier bloc de la blockchain officielle. Donc, on a toujours $\mathbf{a}(0, 0) = 1$.

Définition 5. Une stratégie est un choix d'actions markovien $\mathbf{a}: \mathbb{N}^2 \rightarrow \{0, 1\}$ tel que $\mathbf{a}(0, 0) = 1$.

Par ailleurs, suivant la transition du système, la hauteur de la blockchain avance et l'attaquant gagne une récompense. Cette récompense est faite des blocs minés par l'attaquant qu'il rend public et qui remporte l'adhésion des honnêtes mineurs. Concrètement, si $a > h$ et que le mineur décide de publier $h + 1$ blocs, alors il gagne la récompense contenue dans $h + 1$ blocs. Dans tous les autres cas, il ne gagne rien.

Toute stratégie \mathbf{a} détermine une marche aléatoire \mathbf{X} sur une chaîne de Markov $(\mathbb{N}^2, \mathbb{P}_{\mathbf{a}})$ dont l'état initial est $\mathbf{X}_0 = (0, 0)$.

Définition 6. Soit $\mathbf{a} \in \mathbb{N}^2$, on note $\nu_{\mathbf{a}}$ l'instant de premier retour en $(0, 0)$ pour la chaîne de Markov $(\mathbb{N}^2, \mathbb{P}_{\mathbf{a}})$ i.e., $\nu_{\mathbf{a}} = \text{Inf} \{n \in \mathbb{N}^* / \mathbf{X}_n = (0, 0)\}$

On ne s'intéresse qu'aux choix d'actions conduisant périodiquement à $(0, 0)$. En particulier, on ne considérera pas le cas où $\mathbf{a}(a, h) = 1$ pour tout $(a, h) \in \mathbb{N}^2$ qui consisterait à miner secrètement sans arrêt sans rien faire d'autre.

Définition 7. Soit \mathfrak{A} l'ensemble des stratégies telles que $(\mathbb{N}^2, \mathbb{P}_{\mathbf{a}})$ soit irréductible, récurrente et transitive.

Donc, si $\mathbf{a} \in \mathfrak{A}$, alors $\mathbb{E}[\nu_{\mathbf{a}}] < +\infty$. La durée $\nu_{\mathbf{a}}$ correspond au nombre de transitions sur la chaîne de Markov durant un cycle d'attaque. Le mineur ne fait que répéter des cycles d'attaque.

Chaque transition a une durée aléatoire. Par exemple, si le mineur est dans l'état (a, h) et décide d'écraser les h derniers blocs de la blockchain (possible si $a > h$) ou bien d'abandonner (possible si $a \leq h$), alors la transition est immédiate. Si par contre, il décide de miner secrètement, alors la transition met une durée t où t suit une loi exponentielle de paramètre τ_0 avec $\tau_0 = 10$ minutes ou $\tau_1 = \frac{\tau_0}{d}$ où d est un facteur d'ajustement de difficulté après validation de 2016 blocs (officiels).

Le temps mis pour revenir à l'état $(0, 0)$ est alors $\tau_{\mathbf{a}} = \sum_{n=1}^{\nu_{\mathbf{a}}} t_n \mathbb{1}_{\mathbf{a}(\mathbf{X}_n)=1}$. D'après la formule de Wald, ce temps d'arrêt est intégrable : $\mathbb{E}[\tau_{\mathbf{a}}] < +\infty$. Par ailleurs, le mineur cherche au moins à miner un bloc au début de son attaque car $\mathbf{a}(0, 0) = 1$. Donc, $0 < \mathbb{E}[\tau_{\mathbf{a}}]$ et à la fin d'un cycle d'attaque, lorsque la chaîne de Markov revient en $(0, 0)$, la blockchain officielle a toujours progressé d'au moins un bloc. Le temps d'arrêt $\tau_{\mathbf{a}}$ représente un cycle d'attaque à l'issue duquel l'attaquant revient dans son état initial et reconnaît l'ensemble de la blockchain officielle comme un mineur normal. Ce cycle d'attaque est répété indéfiniment.

5.2 Fonction objectif

Il s'agit de maximiser le revenu du mineur par unité de temps. En raisonnant comme dans la Section 4.5, on comprend qu'il s'agit de trouver la ou les stratégies $\mathbf{a} \in \mathfrak{A}$ qui maximisent le taux de rendement $\frac{\mathbb{E}[R(\tau_{\mathbf{a}})]}{\mathbb{E}[\tau_{\mathbf{a}}]}$ où $R(\tau_{\mathbf{a}})$ est le revenu du mineur à l'issue d'un cycle d'attaque i.e., nombre de blocs minés par l'attaquant et ajoutés à la blockchain officielle à l'issue d'un cycle d'attaque et $\tau_{\mathbf{a}}$ est la durée de ce cycle d'attaque.

Théorème 8. *Soit une stratégie $\mathbf{a} \in \mathfrak{A}$. Alors, après ajustement de la difficulté, on a $\mathbb{E}[\tau_{\mathbf{a}}] = \mathbb{E}[H(\tau_{\mathbf{a}})] \cdot \tau_0$ où $H(\tau_{\mathbf{a}})$ est le nombre de blocs officiels ajoutés à la blockchain officielle après un cycle d'attaque $\tau_{\mathbf{a}}$ et $\tau_0 = 10$ minutes.*

Démonstration. Un cycle d'attaque prend toujours fin au moment où un bloc officiel est révélé. En moyenne après ajustement de difficulté, le temps d'attente entre deux blocs officiels est $\tau_0 = 10$ minutes. Donc, en moyenne, la durée d'un cycle d'attaque est $\mathbb{E}[\tau_{\mathbf{a}}] = \mathbb{E}[H(\tau_{\mathbf{a}})] \cdot \tau_0$. \square

Corollaire 9. *La fonction objectif est $\frac{\mathbb{E}[R(\tau_{\mathbf{a}})]}{\mathbb{E}[H(\tau_{\mathbf{a}})]}$.*

La stratégie « honnête » correspond à $\mathbf{a}(a, h) = 0$ pour tout $(a, h) \neq (0, 0)$. Dans ce cas, le cycle prend fin dès qu'un bloc est découvert et il y a une probabilité q que ce soit un bloc découvert par le mineur. Donc, dans ce cas, $\frac{\mathbb{E}[R(\tau_{\mathbf{a}})]}{\mathbb{E}[H(\tau_{\mathbf{a}})]} = q$. D'où la définition suivante.

Définition 10. *On dit qu'une stratégie \mathbf{a} est gagnante si son rendement est plus rentable que la stratégie honnête, autrement dit si $\frac{\mathbb{E}[R(\tau_{\mathbf{a}})]}{\mathbb{E}[H(\tau_{\mathbf{a}})]} > q$.*

Notons que dans les cas des stratégies de minage égoïste, on a $H(\tau) = N(\tau) \vee N'(\tau)$ (en notant τ le temps d'arrêt mettant fin au cycle d'attaque) [3] où $N(t)$ (resp. $N'(t)$) est le processus de Poisson comptant les découvertes de blocs par les honnêtes mineurs (resp. l'attaquant). Mais dans le cas général, cette égalité n'est pas nécessairement vraie car l'attaquant a la possibilité de réaliser des publications partielles sans mettre fin au cycle d'attaque de la stratégie.

Le but est d'identifier la meilleure stratégie \mathbf{a} maximisant la fonction objectif $\frac{\mathbb{E}[R(\tau_{\mathbf{a}})]}{\mathbb{E}[H(\tau_{\mathbf{a}})]}$ et de trouver le seuil minimal en terme de puissance de hachage relative q au-delà duquel cette stratégie n'est pas la stratégie honnête.

5.3 Résolution en utilisant un solveur

Chaque transition sur la chaîne de Markov octroie un revenu R_a au mineur et R_h aux honnêtes mineurs. En vertu de la loi des grands nombres, suivant le calcul mené dans la Section 4.5, le pourcentage de blocs minés par l'attaquant dans la blockchain officielle est aussi bien $\frac{\mathbb{E}[R(\nu_{\mathbf{a}})]}{\mathbb{E}[H(\nu_{\mathbf{a}})]}$ que $\frac{\mathbb{E}[R_a]}{\mathbb{E}[R_a] + \mathbb{E}[R_h]}$. Ces deux quantités sont égales. S'il existe une stratégie gagnante, alors $\frac{\mathbb{E}[R_a]}{\mathbb{E}[R_a] + \mathbb{E}[R_h]} > q$. Donc,

$$\mathbb{E}[\tilde{R}] > 0$$

avec $\tilde{R}_{\mathbf{a}} = R_a - q \cdot (R_a + R_h)$. Cette quantité peut s'interpréter comme un revenu net que gagne le mineur à chaque transition. Le second facteur $q \cdot (R_a + R_h)$ s'interprète comme un coût de minage ; celui-ci est proportionnel aux nombres de blocs officiels minés et le facteur de proportionnalité est q . L'existence d'une stratégie gagnante revient à résoudre un problème de décision markovienne : on recherche une stratégie \mathbf{a} telle que la chaîne de Markov associée maximise la quantité $\mathbb{E}[\tilde{R}_{\mathbf{a}}]$ où $\tilde{R}_{\mathbf{a}}$ est une récompense (positive ou négative) accordée au mineur à chaque transition. Si pour la stratégie optimale, on a $\mathbb{E}[\tilde{R}_{\mathbf{a}}] > 0$, alors il existe une stratégie gagnante. Pour le calcul du rendement effectif, on est conduit à étudier le même problème mais avec une récompense de la forme $R_a - \lambda \cdot (R_a + R_h)$ pour $\lambda \in \mathbb{R}_+$. La plus grande valeur de λ telle que l'on ait $\mathbb{E}[\tilde{R}_{\mathbf{a}}] > 0$ pour au moins une stratégie \mathbf{a} correspond au taux de hachage effectif. En pratique, la solution s'obtient en utilisant un solveur de décision Markovienne. C'est la voie choisie dans [11]. Mais a priori, le nombre d'états de la chaîne de Markov est infini alors que les solveurs connus ne traitent que des cas où le nombre d'états est fini, ce qui conduit à des difficultés techniques. Nous proposons ci-dessous (Section 5.4) une autre méthode qui permet d'éviter ce problème.

Note 11. Concrètement, dans le cas général où la connectivité est non-nulle, le problème a été traité et résolu numériquement dans [11]. Dans cet article, les auteurs, modélisent le problème formellement à l'aide d'une démarche du type « machine à états » semblable à celle exposée ici mais avec un paramètre supplémentaire tenant compte de la connectivité [11]. L'état du système est représenté par un triplet (a, h, f) où $f \in \{i, r, a\}$ représente la possibilité de provoquer un fork de la blockchain officielle en présentant un certain nombre de blocs en compétition avec ceux minés par les honnêtes mineurs. Ceci n'est possible que si les trois conditions suivantes sont réunies : (1) $a \geq h$, (2) les honnêtes mineurs viennent juste de découvrir un nouveau bloc et (3) la connectivité de l'attaquant est suffisante pour pouvoir convaincre une fraction $\gamma \neq 0$ des honnêtes mineurs de miner sur le fork (de même hauteur que la blockchain officielle) créée par l'attaquant et rendu public. Le paramètre $f = i$ (pour « irrelevant ») signifie que ces conditions ne sont pas toutes remplies, $f = r$ (pour « relevant ») signifie qu'au contraire, cela est possible et $f = a$ (pour « active ») signifie qu'un « fork » est en cours de traitement par le réseau. A chaque état du système, le mineur a la possibilité de réaliser l'une des actions suivantes : abandonner et revenir sur la blockchain officielle, attendre et miner secrètement sur son fork, effacer les derniers blocs de la blockchain officielle en rendant public $h + 1$ blocs (dans le cas où $a > h$) ou provoquer un fork en rendant public h blocs (lorsque $f = r$).

5.4 Un algorithme

5.4.1 Notations

Les processus représentant les découvertes des blocs du réseau et de l'attaquant sont représentés par deux processus de Poisson indépendants N et N' de conditions initiales $N(0) = N'(0) = 0$. Pour $n \in \mathbb{N}$, on pose $\tilde{S}_n = \text{Inf} \{t \in \mathbb{R}_+ / N(t) + N'(t) \geq n\}$. C'est l'instant où un n -ème bloc est découvert, par les honnêtes mineurs ou par l'attaquant.

Par convention, le bloc officiel numéroté 0 est le dernier bloc reconnu à la fois par l'attaquant et les honnêtes mineurs.

La position du mineur à l'instant $t \in \mathbb{R}$ est noté $X(t) = (a, h) \in \mathbb{N}^2$. Le premier (resp. second) indice a (resp. h) de $X(t)$ représente le nombre de blocs tenus secrets minés par l'attaquant (resp. le nombre de blocs minés par le reste du réseau et non-reconnus par l'attaquant). Remarquons que d'après la définition de \mathfrak{a} , on a exclu la possibilité que le mineur abandonne sans rien publier alors qu'il a de l'avance par rapport à la blockchain officielle ; on a toujours : $\mathbb{E}[H(\tau) | X(0) = (0, 0)] > 0$

Définition 12. On note $\mathcal{T}_{i,j}$ l'ensemble des temps d'arrêt $\tau = \tau_{\mathfrak{a}} \in L^1$ avec $\mathfrak{a} \in \mathfrak{A}$ tels que $\mathbb{E}[H(\tau) | X(0) = (i, j)] > 0$ et on pose $\mathcal{T} = \mathcal{T}_{0,0}$.

On suppose que l'attaquant adopte une stratégie \mathfrak{a} .

Définition 13. On note $R(t)$ le nombre de blocs minés par l'attaquant entre 0 et t et présents dans la blockchain officielle en t et $H(t)$ la hauteur de la blockchain officielle en t .

Note 14. Etant donné que les processus $R(t)$ et $H(t)$ sont discontinus, on considère toujours que $R(t)$ et $H(t)$ représentent en réalité $R(t^+)$ et $H(t^+)$.

5.4.2 Linéarisation du problème

Le temps de minage interbloc suit une loi exponentielle. Donc, \tilde{S}_n suit une loi Gamma (dont le paramètre dépend de la difficulté de minage). Donc, si τ est un temps d'arrêt intégrable, $\tau \wedge \tilde{S}_n \xrightarrow{L^1} \tau$ et on a :

$$\begin{aligned} \mathbb{E}[R(\tau \wedge \tilde{S}_n)] &\longrightarrow \mathbb{E}[R(\tau)] \\ \mathbb{E}[H(\tau \wedge \tilde{S}_n)] &\longrightarrow \mathbb{E}[H(\tau)] \end{aligned}$$

Par suite, s'il existe une stratégie \mathfrak{a} et un temps d'arrêt $\tau_{\mathfrak{a}}$ tel que $\frac{\mathbb{E}[R(\tau_{\mathfrak{a}})]}{\mathbb{E}[H(\tau_{\mathfrak{a}})]} > q$, alors il existe aussi un cycle d'attaque τ tel que le nombre de blocs découverts dans chaque cycle d'attaque ne dépasse pas n et tel que $\frac{\mathbb{E}[R(\tau)]}{\mathbb{E}[H(\tau)]} > q$. Cette inégalité s'exprime sous la forme

$$\mathbb{E}[R(\tau) - qH(\tau)] > 0$$

et se traduit en disant que le revenu net du mineur est positif à l'issue de τ en considérant que le coût du minage est proportionnel au nombre de blocs découverts durant le cycle d'attaque τ avec un coefficient de proportionnalité q .

On peut renforcer la condition initiale. Le mineur part de l'état $(0, 0)$ et revient à l'état $(0, 0)$. D'après le Théorème 8, le problème revient à rechercher le rendement optimal $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0)=(0,0)]}{\mathbb{E}[H(\tau)|X(0)=(0,0)]}$. Celui-ci peut se linéariser de la façon suivante

Définition 15. Pour $(i, j) \in \mathbb{N}^2$, on pose

$$\mathcal{C}(i, j) = \left\{ c \in \mathbb{R} / \sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] > 0 \right\}$$

et $\mathcal{C} = \mathcal{C}(0, 0)$.

Notons que $\mathcal{C}(i, j) \neq \emptyset$ car $0 \in \mathcal{C}(i, j)$.

Théorème 16. On a $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0)=(0,0)]}{\mathbb{E}[H(\tau)|X(0)=(0,0)]} = \sup \mathcal{C}$.

La démonstration de ce résultat se trouve aussi dans [11] pour le cas général.

Démonstration. Soit τ un temps d'arrêt intégrable. Alors, $N(\tau)$ et $N'(\tau)$ sont intégrables (il suffit de considérer le temps d'arrêt $\tau \wedge n$ avec $n \in \mathbb{N}$ puis d'appliquer le théorème d'arrêt de Doob et de faire tendre n vers l'infini. Voir la preuve de la Proposition 4.3 de [5]). On a $R(\tau) \leq N'(\tau)$ et $H(\tau) \leq N(\tau) + N'(\tau)$. Donc, $R(\tau)$ et $H(\tau)$ sont intégrables. Soit $c \in \mathcal{C}$. Alors, il existe τ un temps d'arrêt intégrable tel que $\mathbb{E}[R(\tau)|X(0) = (0, 0)] > c \mathbb{E}[H(\tau)|X(0) = (0, 0)]$. Donc, nécessairement $\mathbb{E}[H(\tau)|X(0) = (0, 0)] > 0$ car $\mathbb{E}[H(\tau)|X(0) = (0, 0)] \geq \mathbb{E}[R(\tau)|X(0) = (0, 0)] > c \mathbb{E}[H(\tau)|X(0) = (0, 0)] \geq 0$ et par suite, $\frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} > c$. Donc, $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} > c$. Ceci étant vrai pour tout $c \in \mathcal{C}$. On en déduit que $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} \geq \sup \mathcal{C}$.

Réciproquement, soit $c < \sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]}$. Alors, il existe τ un temps d'arrêt tel que $c < \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]}$. Donc, $\mathbb{E}[R(\tau) - cH(\tau)|X(0) = (0, 0)] > 0$. Donc, $c \in \mathcal{C}$. Donc, $c \leq \sup \mathcal{C}$. Ceci étant vrai pour tout $c < \sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]}$, on en déduit que l'autre inégalité i.e., $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} \leq \sup \mathcal{C}$. Par suite, $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} = \sup \mathcal{C}$. \square

Le théorème permet d'en déduire en particulier que $\sup \mathcal{C} \leq 1$ car $R(\tau) \leq H(\tau)$ pour tout temps d'arrêt τ . De plus, la stratégie honnête correspondant au temps d'arrêt $\tau = \tilde{S}_1$ vérifie $\frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} = q$. Donc,

$$0 < q \leq \sup \mathcal{C} \leq 1 \quad (6)$$

La preuve du Théorème 16 s'adapte au cas où $X(0) = (i, j) \in \mathbb{N}^2$: $\sup_{\tau \in \mathcal{T}_{i,j}} \frac{\mathbb{E}[R(\tau)|X(0) = (i, j)]}{\mathbb{E}[H(\tau)|X(0) = (i, j)]} = \sup \mathcal{C}_{i,j}$

$$0 \leq \sup_{\tau \in \mathcal{T}_{i,j}} \frac{\mathbb{E}[R(\tau)|X(0) = (i, j)]}{\mathbb{E}[H(\tau)|X(0) = (i, j)]} = \sup \mathcal{C}_{i,j} \leq 1 \quad (7)$$

pour $(i, j) \in \mathbb{N}^2$.

Une fois linéarisée, les auteurs de [11] font ensuite appel à un solveur de problème de décision markovienne pour trouver $\sup \mathcal{C}$ dans le cas général. La difficulté à laquelle ils font face est due au fait que ce solveur ne s'applique que lorsque le nombre d'états est fini. Nous ne rencontrons pas ce problème si on le discrétise.

5.4.3 Discrétisation du problème

Définition 17. Pour $(i, j, n, c) \in \mathbb{N}^3 \times \mathbb{R}$, on pose :

$$\Phi_c(i, j, n) = \sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau \wedge \tilde{S}_n) - cH(\tau \wedge \tilde{S}_n)|X(0) = (i, j)] \in \bar{\mathbb{R}}$$

Lorsque $c = q$, on pose $\Phi(i, j, n) = \Phi_q(i, j, n)$.

Notons que $\forall n \in \mathbb{N}^*$, $\Phi(0, 0, n) \geq 0$ car si $\tau = \tilde{S}_1$, alors $\mathbb{E}[R(\tau \wedge \tilde{S}_n) - qH(\tau \wedge \tilde{S}_n)|X(0) = (i, j)] = 0$.

Exemple 18. On a $\Phi_c(i, j, 0) = j \mathbb{1}_{j > i} - c \cdot (i \vee j)$.

Théorème 19. On a $\sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] = \lim_{n \rightarrow \infty} \Phi_c(i, j, n)$

Démonstration. Soit $(i, j) \in \mathbb{N}$ et $c \in \mathbb{R}$. D'après la Définition 17, on a

$$\Phi_c(i, j, n) \leq \sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] \quad (8)$$

pour tout $n \in \mathbb{N}$. De plus la suite $n \mapsto \Phi_c(i, j, n)$ est clairement croissante. Donc, elle est convergente dans $\overline{\mathbb{R}}$ et

$$\lim_{n \rightarrow \infty} \Phi_c(i, j, n) \leq \sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] \leq +\infty \quad (9)$$

Réciproquement, soit $l = \sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)]$ et supposons que $l \in \mathbb{R}$. Pour tout $0 < \varepsilon < l$, il existe $\tau \in L^1$ tel que

$$0 < l - \varepsilon < \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] \quad (10)$$

On a $R(\tau) \leq N'(\tau)$ et $H(\tau) \leq N(\tau) + N'(\tau)$. Donc, $R(\tau)$ et $H(\tau)$ sont intégrables. De plus, pour tout $n \in \mathbb{N}$, on a $R(\tau \wedge \tilde{S}_n) \leq R(\tau)$ et $H(\tau \wedge \tilde{S}_n) \leq H(\tau)$. Donc, $R(\tau \wedge \tilde{S}_n)$ et $H(\tau \wedge \tilde{S}_n)$ sont intégrables et d'après le théorème de convergence dominée, $R(\tau \wedge \tilde{S}_n)$ (resp. $H(\tau \wedge \tilde{S}_n)$) converge vers $R(\tau)$ (resp. $H(\tau)$) dans L^1 . Par suite, pour tout $\varepsilon > 0$, il existe $n \in \mathbb{N}$ tel que

$$0 < l - \varepsilon < \mathbb{E}[R(\tau \wedge \tilde{S}_n) - cH(\tau \wedge \tilde{S}_n)|X(0) = (i, j)] \quad (11)$$

L'inégalité entraîne $l - \varepsilon < \Phi_c(i, j, n) \leq \lim_{n \rightarrow \infty} \Phi_c(i, j, n)$. En faisant tendre $\varepsilon \rightarrow 0$, on en déduit que

$$\mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] \leq \lim_{n \rightarrow \infty} \Phi_c(i, j, n)$$

et par suite

$$\sup_{\tau \in \mathcal{T}_{i,j}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (i, j)] \leq \lim_{n \rightarrow \infty} \Phi_c(i, j, n) \quad (12)$$

En raisonnant de la même façon, on parvient au même résultat dans le cas où $l = +\infty$. \square

Corollaire 20. On a $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} = \sup \{c \in \mathbb{R} / \exists n \in \mathbb{N}, \Phi_c(0, 0, n) > 0\}$

Démonstration. Cela découle du Théorème 16 et du Théorème 19. \square

Le théorème suivant donne une condition pour que le mineur soit incité à miner honnêtement sur la blockchain officielle.

Théorème 21. On a $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} = q$ si et seulement si $\Phi(0, 0, n) = 0$ pour tout $n \in \mathbb{N}$.

Démonstration. Supposons que $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} > q$. Alors, $\exists c > q \exists \tau / \frac{\mathbb{E}[R(\tau)]}{\mathbb{E}[H(\tau)]} > c$. Donc, $\sup_{\tau \in \mathcal{T}} \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (0, 0)] > 0$. Donc, d'après le Théorème 19, $\exists n \in \mathbb{N}^* / \Phi(0, 0, n) > 0$. Réciproquement, si $\exists n \in \mathbb{N}^* / \Phi(0, 0, n) > 0$ alors $\exists \tau / \mathbb{E}[R(\tau) - qH(\tau)|X(0) = (0, 0)] > 0$. Donc, par continuité, $\exists c > q, \mathbb{E}[R(\tau) - cH(\tau)|X(0) = (0, 0)] > 0$. Donc, $\sup_{\tau \in \mathcal{T}} \frac{\mathbb{E}[R(\tau)|X(0) = (0, 0)]}{\mathbb{E}[H(\tau)|X(0) = (0, 0)]} \geq c > q$. \square

Corollaire 22. Un mineur n'est pas incité à être honnête si et seulement si il existe $n \in \mathbb{N}$ tel que $\Phi(0, 0, n) > 0$.

5.4.4 Un algorithme pour calculer $\Phi_c(i, j, n)$

On a le théorème fondamental suivant :

Théorème 23. Soit $n \in \mathbb{N}^*$. Si $i < j$, on a :

$$\Phi_c(i, j, n) = \text{Max}(\Phi_c(0, j - i - 1, n) + (1 - c)(i + 1), p\Phi_c(i + 1, j, n - 1) + q\Phi_c(i, j + 1, n - 1))$$

et si $j \leq i$, on a :

$$\Phi_c(i, j, c, n) = \text{Max}(-ci, p\Phi_c(i + 1, j, n - 1) + q\Phi_c(i, j + 1, n - 1))$$

Démonstration. En effet, si $i < j$, le mineur a la possibilité de réaliser l'action « écraser » qui consiste à publier $i + 1$ blocs (avec $i < j$) ou l'action « attendre » i.e., continuer à miner en secret. S'il décide de publier les $i + 1$ premiers blocs qu'il a minés en secret, ces blocs deviennent officiels et l'attaquant les reconnaît naturellement. Si $j \leq i$, le mineur peut miner en secret ou abandonner. Dans ce cas, le mineur ne gagne rien et reconnaît simplement les i derniers blocs de la blockchain officielle. \square

Les nombres $\Phi_c(i, j, n)$ s'obtiennent donc par récurrence à l'aide du Théorème 23 et de la condition initiale (cas $n = 0$) suivante : $\Phi_c(i, j, 0) = (1 - c)j$ si $i < j$ et $\Phi_c(i, j, 0) = -ci$ si $j \leq i$. Ceci permet de calculer $\Phi_c(0, 0, n)$ pour tout $c \in \mathbb{R}$ et $n \in \mathbb{N}$. L'algorithme est efficace en utilisant le principe de mémoïzation. Ci-dessous un programme simple en R.

Exemple 24. Un programme pour calculer $\Phi_c(i, j, n)$ en R en utilisant le package « memoise ».

```
library(memoise)
Phi <- function(q,c,i,j,n){
  if (n==0){
    if(i<j){return((1-c)*j)}
    else{return(-c*i)}
  }
  else{
    if (i<j){
      return(max(Phi(q,c,0,j-i-1,n)+(1-c)*(i+1),(1-q)*Phi(q,c,i+1,j,n-1)+q*Phi(q,c,i,j+1,n-1)))}
    else{return(max(-c*i,(1-q)*Phi(q,c,i+1,j,n-1)+q*Phi(q,c,i,j+1,n-1)))}
  }
}
Phi <- memoise(Phi)
```

Exemple 25. Par récurrence, en utilisant, le Théorème 23, on montre facilement les égalités suivantes : $\Phi(0, 0, 1) = \Phi(0, 0, 2) = 0$ et $\Phi(0, 0, 3) = pq(q^2 + 2q - 1)_+$.

Le calcul précédent entraîne facilement l'équivalence $\Phi(0, 0, 3) > 0 \iff q > \sqrt{2} - 1$. Donc, en vertu du Corollaire 22, il s'ensuit qu'un mineur quelconque n'est pas incité à être honnête dès que sa puissance de hachage relative est supérieure à $\sqrt{2} - 1$. On retrouve le seuil obtenu en étudiant la stratégie « 1+2 ». Nous pouvons maintenant améliorer ce résultat.

Théorème 26. Si $q \geq 32,94\%$, alors la stratégie honnête n'est pas optimale.

Démonstration. Avec $q = 32,94\%$, le calcul montre que $\Phi(0, 0, 92) = 4 \cdot 10^{-7} > 0$. Donc, d'après le Théorème 21, on en déduit que la stratégie honnête n'est pas optimale pour $q \geq 32,94\%$. \square

Il nous est impossible d'améliorer ce nombre avec notre ordinateur. Le seuil semble être environ de 32,94%.

5.5 Algorithme simplifié pour calculer le taux de hachage maximum

Notons que la fonction $c \mapsto \Phi_c(i, j, n)$ est décroissante. On en déduit l'algorithme suivant pour calculer le taux de hachage apparent maximal.

On fait varier c à partir de $c = q + \varepsilon$ (avec $\varepsilon > 0$ fixé) puis n à partir de $n = 0$ et on s'arrête si on trouve n tel que $\Phi_c(0, 0, n) > 0$. Dans ce cas, c est remplacé par $c + \varepsilon$ et on recherche de nouveau n tel que $\Phi_c(0, 0, n) > 0$. S'il est impossible de trouver un entier n tel que $\Phi_c(0, 0, n) > 0$ et $n < N$ (avec N fixe), l'algorithme s'arrête et le rendement maximum recherché est $c - \varepsilon$.

5.6 Recherche de la stratégie optimale

L'algorithme précédent du Théorème 23 donne le taux de hachage maximal apparent mais ne donne pas précisément la politique de minage que doit suivre l'attaquant. Si l'on veut savoir ce que doit faire l'attaquant lorsque l'état du système est (i_0, j_0) , on reprend le même algorithme mais en imposant que lorsque l'état du système est (i_0, j_0) , le mineur choisit par exemple obligatoirement l'action « attendre » (on peut aussi choisir une autre action). On doit donc considérer l'algorithme modifié suivant :

Pour $n \in \mathbb{N}^*$,

- cas $(i, j) = (i_0, j_0)$:

$$\Phi_c(i_0, j_0, n) = p \Phi_c(i_0 + 1, j_0, n - 1) + q \Phi_c(i_0, j_0 + 1, n - 1)$$

- cas $(i, j) \neq (i_0, j_0)$

- cas $i < j$:

$$\Phi_c(i, j, n) = \Phi_c(0, j - i - 1, n) + (1 - c)(i + 1, p \Phi_c(i + 1, j, n - 1) + q \Phi_c(i, j + 1, n - 1))$$

- cas $j \leq i$:

$$\Phi_c(i, j, c, n) = \text{Max}(-c i, p \Phi_c(i + 1, j, n - 1) + q \Phi_c(i, j + 1, n - 1))$$

et avec toujours les mêmes conditions initiales : $\Phi_c(i, j, 0) = (1 - c) j$ si $i < j$ et $\Phi_c(i, j, 0) = -c i$ si $j \leq i$.

Si cet algorithme retourne la même valeur que l'algorithme de la Section 5.4.4, cela signifie que l'action « attendre » conduit à un taux de hachage apparent maximum.

6 Le Jeu du Minage

Nous proposons une autre formulation du problème de minage en terme de jeu simple où un joueur joue contre une banque. On repart du critère $\mathbb{E}[G(\tau) - q H(\tau)] > 0$ qui permet de savoir si une stratégie de minage est plus performante que la stratégie honnête. On interprète la quantité $R(t) = G(t) - H(t)$ comme un revenu net. Tout se passe comme si le mineur devait payer un coût de minage constant par morceau au cours du temps et qui augmente de q (en prenant pour unité, la valeur moyenne contenue dans un bloc de la blockchain officielle, soit un peu plus qu'une « coinbase ») à chaque fois que la hauteur de la blockchain officielle augmente de 1.

6.1 Une variation du jeu de Pile ou Face classique

On peut comparer les actions du mineur à celles d'un joueur jouant contre une banque au jeu noté JM (pour jeu du minage) suivant. Au cours de ce jeu, le joueur accumule régulièrement des jetons qu'il peut sous certaines contraintes convertir en argent sonnante et trébuchant (en euros mettons). A chaque instant, le joueur dispose d'au plus trois actions possibles :

Lancer. Un croupier lance une pièce de monnaie truquée en faveur de la banque. La probabilité d'obtenir « Pile » est q .

- Si le résultat est « Pile », le joueur gagne un jeton et ne paye rien.
- Si le résultat est « Face », la banque gagne un jeton et le joueur verse q au croupier.

Ecraser. Cette action n'est possible que si le joueur (resp. la banque) possède a (resp. h) jetons avec $a > h$. Dans ce cas, la banque perd tous ses jetons, le joueur perd $h + 1$ jetons mais gagne $h + 1$ € et donne aussi q € au croupier. Son résultat net est donc $h + 1 - q$.

Abandon. La banque et le joueur perdent tous leurs jetons. Cette action ne coûte rien au joueur.

Plus exactement, on note $JM(i, j)$ le jeu décrit ci-dessus mais où le joueur part d'une situation où il possède i jetons contre j pour la banque, $(i, j) \in \mathbb{N}^2$.

On peut faire quelques commentaires.

- Une pièce lancée par le croupier revient à la découverte d'un bloc par le mineur ou les honnêtes mineurs.
- L'action **Lancer** est équivalente pour le mineur au choix de miner secrètement et d'attendre la découverte d'un bloc. Si le résultat est « Face », alors la blockchain officielle progresse d'un bloc. La fonction hauteur augmente donc de 1. D'où un coût q versé par le joueur dans ce cas. Si par contre, le résultat est « Pile », le bloc découvert par le mineur est gardé secret. La hauteur de la blockchain officielle n'augmente donc pas. D'où un coût nul.
- L'action **Ecraser** revient pour le mineur à remplacer les h derniers blocs de la blockchain officielle par les siens. Pour que cette action soit possible, il faut que le mineur révèle un autre bloc en plus, ce qui fait progresser la hauteur de la blockchain officielle de 1. Le mineur gagne alors la récompense contenue dans $h + 1$ blocs et dans le même temps, la hauteur de la blockchain officielle augmente de 1. D'où un gain net de $h + 1 - q$.
- L'action **Abandon** revient pour le mineur à laisser tomber son « fork » tenu secret et à revenir miner sur le dernier bloc de la blockchain officielle. Il ne gagne rien ni ne perd rien avec cette action car la hauteur de la blockchain officielle est inchangée.

Il s'agit d'une variation du jeu élémentaire de Pile ou Face où normalement le joueur doit toujours payer q € au croupier et gagne 1€ directement si le résultat est « Pile ». L'espérance de gain de ce jeu est évidemment nulle. Une variation classique de ce jeu de pile ou face mais avec des jetons est la suivante. A la demande d'un joueur qui doit pour cela payer q €, un croupier peut lancer une pièce de monnaie. Quel que soit le résultat, le joueur doit payer q € s'il choisit cette action. Si l'on obtient « Pile », le joueur gagne un jeton. Sinon, c'est la banque qui gagne un jeton. La pièce de monnaie est truquée en faveur de la banque : la probabilité d'obtenir Pile est $q < \frac{1}{2}$. Si, au cours du jeu, le joueur a un nombre de jetons supérieur au nombre de jetons h de la banque, il peut décider de faire perdre tous ses jetons à la banque. Cette action lui fait aussi perdre $h + 1$ jetons mais elle lui rapporte $(h + 1)$ €. A tout moment, le joueur peut abandonner et revenir au début du jeu. Là encore, on peut montrer que l'espérance de gain du joueur est nulle.

On reconnaît en $JM(0, 0)$ une variation tordue du jeu de Pile ou Face où le joueur ne payerait que les fois où le croupier obtient « Face ». On comprend donc que ce jeu peut être à l'avantage du joueur même si l'équivalent pour le joueur de la stratégie « honnête » pour un mineur offre une perspective de revenu nul. En effet, la stratégie dite « honnête » est celle qui force le joueur à utiliser l'action **Ecraser** dès que $a = 1$ et $h = 0$ et à abandonner si $a = 0$ et $h = 1$. La stratégie prend donc fin après deux actions dont une action **Lancer** au début. A l'issue de cette stratégie, le joueur gagne $1 - q$ avec probabilité q et paye q avec probabilité $1 - q$. Donc, l'espérance de revenu de cette stratégie est nulle.

Définition 27. Soit un jeu où un joueur joue contre banque. On dit que le jeu est biaisé en faveur du joueur s'il existe une stratégie prenant fin à un instant aléatoire d'espérance finie τ telle que $\mathbb{E}[R(\tau)] > 0$ où $R(\tau)$ est le revenu net accumulé par le joueur entre 0 et τ .

Si tel est le cas, en choisissant par exemple l'action **Abandon** après τ (ou si possible une action **Ecraser**), alors le joueur se retrouve dans le même état qu'au début du jeu. Il peut alors répéter sa stratégie et obtient un gain potentiellement infini.

Définition 28. On appelle cycle de jeu le premier instant hormis l'instant initial où le joueur se retrouve sans aucun jeton ainsi que la banque.

6.2 Espérance de revenu maximal et seuil de tolérance

On cherche à connaître le seuil sur q à partir duquel le jeu $JM(0, 0)$ est biaisé en faveur du joueur. Pour cela on peut considérer le jeu tronqué $JM(i, j, n)$ qui ne laisse au joueur de $JM(i, j)$ que n actions possibles dans tout le jeu avec $n \in \mathbb{N}$. On peut montrer que le jeu $JM(0, 0)$ est biaisé en faveur du joueur si le jeu $JM(0, 0, n)$ l'est pour au moins un entier n . Il en résulte que le jeu du minage $JM(0, 0)$ est biaisé en faveur du joueur s'il existe $n \in \mathbb{N}$ tel que $E(0, 0, n) > 0$ où $E(a, h, n)$ désigne le revenu net moyen maximal du joueur qui joue à $JM(a, h, n)$ ($(a, h, n) \in \mathbb{N}^3$). Or, on peut calculer par récurrence $E(a, h, n)$. En effet, de trois choses l'une :

- Si $n > 0$ et $a > h$, le joueur dispose d'encore au moins une action. Il peut soit utiliser l'action **Ecraser** soit utiliser l'action **Lancer**. Donc,

$$E(a, h, n) = \text{Max}\{h + 1 - q + E(a - h - 1, 0, n - 1), qE(a + 1, h, n - 1) + (1 - q) \cdot (E(a, h + 1, n - 1) - q)\}$$

- Si $n > 0$ et $a \leq h$, le mineur peut soit abandonner, soit utiliser l'action **Lancer**. Donc,

$$E(a, h, n) = \text{Max}\{E(0, 0, n - 1), qE(a + 1, h, n - 1) + (1 - q) \cdot (E(a, h + 1, n - 1) - q)\}$$

- Si $n = 0$ alors $E(a, h, 0) = 0$. En effet, si le joueur ne dispose même pas d'une action, il ne peut rien faire ni rien gagner.

Voici un code possible en 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 => adopt, wait *)
{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;
```

Numériquement, on observe que $E(0, 0, 100) = 1,47699 \times 10^{-6} > 0$ pour $q = 32,94\%$. Par contre, on est incapable de trouver $n \in \mathbb{N}$ tel que $E(0, 0, n) > 0$ pour $q \leq 32,93\%$. D'où le résultat.

Proposition 29. *Avec une connectivité nulle, le seuil de tolérance en terme de puissance de hachage relative au delà duquel un mineur n'a plus intérêt à rester honnête est environ 32,94%.*

6.3 Retour sur la stratégie 1+2

Pour calculer le seuil minimal dans la section précédente, on a considéré le cas d'un joueur qui serait forcé d'abandonner dès qu'il a utilisé n actions. On pourrait aussi considérer un joueur qui ne pourrait pas utiliser l'action **Lancer** plus de n fois (autrement dit, le croupier rentrerait chez lui dès qu'il a lancé n pièces de monnaie !) et dont la stratégie prendrait nécessairement fin dès que le joueur perd tous ses jetons ainsi que la banque (autrement dit, si l'on aboutit à $a = h = 0$, ce qui peut survenir après une action **Abandon** ou après une action **Ecraser** avec $a = h + 1$). Ceci correspond à un cycle de jeu. Etant donné que chaque action **Lancer** correspond à un bloc découvert, le cycle de jeu envisagé correspond à un cycle d'attaque pour lequel il n'y aurait pas plus de n blocs découverts en tout (officiels ou non). Notons $\Phi(a, h, n)$ le revenu net maximal moyen que peut espérer un joueur partant d'une situation où le joueur (resp. la banque) dispose de a (resp. h) jetons. Supposons $n > 0$. De trois choses l'une :

- Si $a > h + 1$, le joueur peut utiliser l'action **Ecraser** ou l'action **Lancer**. Dans tous les cas, le jeu continue.
- Si $a = h + 1$, le joueur peut utiliser l'action **Ecraser** auquel cas la stratégie prend fin ou l'action **Lancer**.
- Si $a \leq h$, le joueur peut utiliser l'action **Abandon** auquel cas la stratégie prend fin ou l'action **Lancer**.

Par ailleurs, si $n = 0$ et $a > h$, alors le joueur peut utiliser $a - h$ fois l'action **Ecraser** jusqu'à ce que $a = h = 0$. Il gagne alors a et doit payer $(a - h) \cdot q$. Si par contre, $n = 0$ et $a \leq h$ alors il ne peut utiliser que l'action **Abandon** qui ne lui donne rien.

D'où le calcul de $\Phi(a, h, n)$ par récurrence sur n .

- Si $n > 0$ et $a > h+1$, alors

$$\Phi(a, h, n) = \text{Max}\{a - h - 1 + \Phi(a - h - 1, 0, n - 1), q \Phi(a + 1, h, n - 1) + (1 - q) \cdot (\Phi(a, h + 1, n - 1) - q)\}$$

- Si $n > 0$ et $a = h+1$, alors

$$\Phi(a, h, n) = \text{Max}\{a - h - 1, q \Phi(a + 1, h, n - 1) + (1 - q) \cdot (\Phi(a, h + 1, n - 1) - q)\}$$

- Si $n > 0$ et $a \leq h$, alors

$$\Phi(a, h, n) = \text{Max}\{0, q \Phi(a + 1, h, n - 1) + (1 - q) \cdot (\Phi(a, h + 1, n - 1) - q)\}$$

- Si $n = 0$ et $a > h$, alors $\Phi(a, h, 0) = a - (a - h) \cdot q$

- Si $n = 0$ et $a \leq h$, alors $\Phi(a, h, 0) = 0$.

D'où un code possible en Mathematica 11.3.

```

ϕ[a_, h_, n_, q_, c_] := ϕ[a, h, n, q, c] =
Piecewise[
  (* case a>h+1 => override, wait *)
  {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, wait *)
  {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 => adopt, wait *)
  {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];

```

Le calcul montre alors que si $q \leq \sqrt{2} - 1$, alors $\Phi(0, 0, 3) = 0$ et si $q > \sqrt{2} - 1$, alors

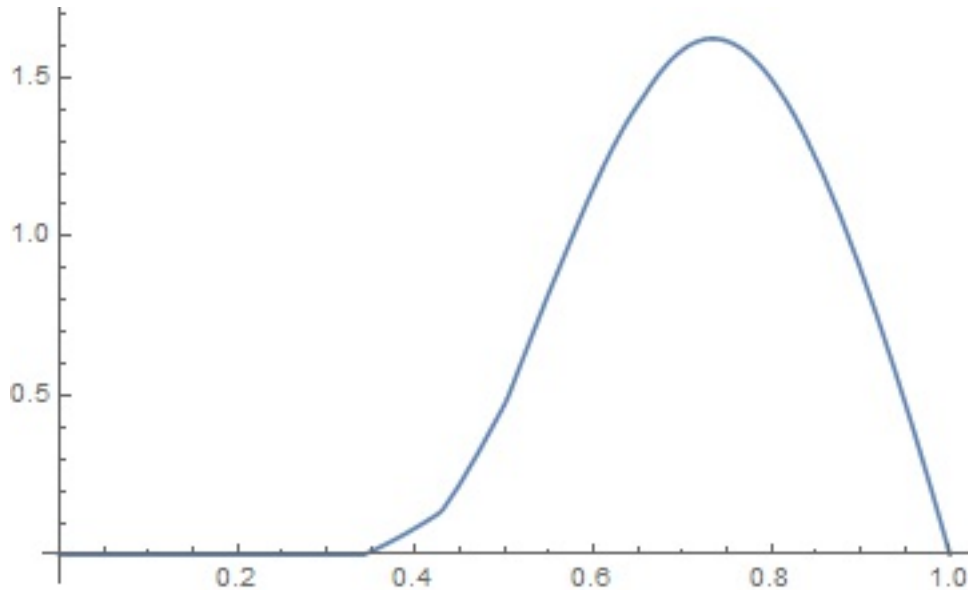
$$\Phi(0, 0, 3) = q \cdot (3q - 1 - q^2 - q^3)$$

On obtient par exemple ce résultat en utilisant la commande

```
Assuming[q > 0, Simplify[ϕ[0, 0, 3, q, q]]]
```

On retrouve le seuil de $\sqrt{2} - 1$ comme dans la stratégie 1+2 et on peut montrer qu'il s'agit bien de la stratégie optimale sous la contrainte d'un cycle comportant au plus trois blocs et $q > \sqrt{2} - 1$.

On s'est amusé ci-dessous à considérer la valeur de $\Phi(0, 0, 10)$ (en ordonnée) pour différentes valeurs de q (en abscisse), même pour $q > \frac{1}{2}$. Voici ce qui est observé.



Le fait que le graphe soit décroissant à partir d'une certaine valeur et devient carrément nul si $q = 1$ s'interprète facilement : plus on contrôle le réseau, moins on a besoin de tricher pour augmenter son profit. A la limite, si on contrôle tout le réseau, il suffit de miner normalement. Par contre, pour tricher (sans connectivité) il faut un minimum de puissance de hachage.

7 Pourquoi Bitcoin existe-t-il encore ?

Les mineurs ne sont pas incités à rester honnêtes et pourtant le réseau fonctionne sans problème majeur depuis sa création. Comment expliquer cela ? D'abord, supposons que la puissance de hachage totale reste la même au cours du temps et imaginons Bitcoin tel qu'il est aujourd'hui mais sans ajustement de difficulté tous les 2016 blocs. Le nombre R de blocs officiels validés par un mineur qui mènerait une attaque quelconque de durée T est nécessairement inférieur au nombre total de blocs découverts par le mineur durant cette période. Donc, $R \leq N'(T)$ où $(N'(t))_{t \geq 0}$ est un processus de Poisson de paramètre $\frac{q}{\tau_0}$ (le mineur met en moyenne $\frac{\tau_0}{q}$ avant de découvrir un bloc) avec comme précédemment $\tau_0 = 10$ minutes et q désigne la puissance de hachage relative du mineur. Sous l'hypothèse $\mathbb{E}[T] < +\infty$, il est facile d'en déduire que $\mathbb{E}[R] \leq \mathbb{E}[N'(T)]b \leq \mathbb{E}[N'(\mathbb{E}[T])]b = q \cdot \frac{\mathbb{E}[T]}{\tau_0} b$ où b est la récompense moyenne contenue dans un bloc. Il ne s'agit pas là d'une démonstration mais cette suite d'inégalités peut se justifier sans peine. Par suite, $\frac{\mathbb{E}[R]}{\mathbb{E}[T]} \leq q \cdot \frac{b}{\tau_0}$. Autrement dit, le rendement de n'importe quelle stratégie ne pourrait être qu'inférieur à q qui est le rendement de la stratégie honnête. C'est probablement ce résultat que Satoshi avait en tête même s'il ne l'a pas écrit. Autrement dit, la meilleure stratégie est toujours la stratégie honnête s'il n'y a pas d'ajustement de difficulté. Le problème vient donc de cet ajustement de difficulté. Le fait est qu'en présence d'un attaquant qui suit une stratégie déviante de rétention de blocs, il y a création de nombreux blocs orphelins. Ces blocs ne sont pas pris en compte par la formule d'ajustement de difficulté. Le protocole « croit » que le réseau a peiné pour valider les blocs et ajuste la difficulté à la baisse. Or pourtant, il n'y a pas plus ni moins d'acteurs qu'avant. Le nombre d'acteurs est resté le même ; la puissance de hachage totale dépensée pour valider des blocs n'a pas changé. Donc, le paramètre de difficulté qui reflète cette puissance de hachage totale ne devrait pas baisser non plus. Néanmoins, ce paramètre diminue car le protocole ne retient que l'existence de blocs officiels. Pour « réparer » Bitcoin, il faudrait donc modifier légèrement le protocole : les noeuds devraient relayer tous les blocs légaux mais aussi les preuves d'existence de blocs orphelins. Un mineur serait incité à relayer cette information dans les blocs qu'il mine car en cas d'égalité entre deux blocs de même hauteur, un noeud considérerait comme officiel le bloc signalant le plus de blocs orphelins. Autrement dit, la blockchain officielle serait toujours la plus solide mais en incluant les blocs orphelins en cas d'égalité. C'est à peu près ce que fait Ethereum qui ne retient à vrai dire que les blocs orphelins qui ne sont pas trop éloignés de la blockchain officielle (les « oncles »). Tous les 2016 blocs, le protocole serait en mesure de détecter le nombre de blocs orphelins durant la dernière période de minage. Si on note n_1 ce nombre, la nouvelle formule d'ajustement de difficulté serait simplement :

$$\Delta' = \Delta \times \frac{(2016 + n_1) \times 10}{T} \quad (13)$$

Dans ces conditions, on peut montrer alors que le rendement du mineur serait toujours inférieur à q (voir ci-dessous, Section 8). Autrement dit, la meilleure stratégie serait toujours la stratégie honnête. Le raisonnement que nous avons fait nous permet aussi de comprendre pourquoi le réseau fonctionne sans incident majeur depuis sa création. Pour tirer profit de cette faille, il faut mener une attaque de rétention de blocs et accepter dans une première partie de perdre de l'argent jusqu'à ce que la difficulté baisse puis progressivement commencer à tirer profit de la stratégie déviante jusqu'à ce que le revenu net du mineur repasse positif à moment donné. Or, il faut au moins 2 semaines avant que la difficulté ne baisse. Et il faut aussi attendre de récupérer tout l'argent perdu dans la première partie de l'attaque. Par suite, une telle attaque est nécessairement laborieuse. Il faut au minimum que le mineur accepte de perdre de l'argent pendant 15 jours tout en misant sur le fait qu'aucun nouveau mineur ne viendra sur le réseau Bitcoin, ce qui est peu probable si la difficulté baisse franchement. Un tel événement devrait au contraire attirer des mineurs. Autrement dit, le protocole comporte une faille mais cette faille est difficilement exploitable. Il semble qu'aucun mineur possédant suffisamment de puissance de hachage n'accepte de perdre de l'argent pendant au moins quinze jours tout en pariant sur une hypothèse difficilement concevable.

Bitcoin est donc essentiellement protégé par le fait que les ajustements de difficulté n'interviennent que toutes les deux semaines. Ce sont les paramètres « en dur » 2016 et 10 minutes qui protègent Bitcoin contre les attaques dites de rétention de blocs.

Note 30. La sécurité d'Ethereum s'étudie de la même façon. Sa formule d'ajustement de difficulté prend en compte la production de blocs orphelins. Elle est donc beaucoup plus robuste que la formule correspondante pour Bitcoin. Il s'ensuit que les seuils de sécurité sont comparables. Néanmoins, les ajustements de difficulté se font en continu sur Ethereum. Donc, un mineur malveillant qui aurait plus de puissance de hachage que le seuil requis pourrait quasi-immédiatement tirer profit d'une stratégie déviante. De ce point de vue, Ethereum est beaucoup plus vulnérable que Bitcoin. Ajoutons par ailleurs qu'entre deux blocs reçus de même hauteur, un mineur honnête sur Ethereum ne mine pas sur le dernier bloc reçu mais aléatoirement sur l'un des deux, ce qui entraîne que la connectivité de l'attaquant est toujours d'au moins $\frac{1}{2}$.

8 Bitcoin modifié

Imaginons un Bitcoin un peu différent mais où les honnêtes mineurs signaleraient les blocs orphelins et que leur existence serait ensuite enregistrée dans la blockchain officielle. Il est tout à fait possible d'inciter les mineurs à enregistrer ces existences de blocs orphelins en modifiant légèrement la règle qui définit la blockchain officielle. Ce serait la blockchain la plus solide (celle qui maximise la difficulté) i.e., en pratique la plus longue. Mais en cas d'égalité entre deux blockchains de même hauteur, la blockchain officielle serait celle qui signale le plus de blocs orphelins. Par ailleurs, imaginons qu'à l'issue d'une période de validation de 2016 blocs officiels, il y ait comme aujourd'hui un ajustement de difficulté mais avec une formule d'ajustement légèrement modifiée qui serait

$$\Delta' = \Delta \cdot \frac{(2016 + n_1) \times 10}{T}$$

où n_1 désigne le nombre de blocs orphelins signalés durant la dernière période de minage (de 2016 blocs). Autrement dit, on aurait une fonction de difficulté D qui ne serait pas exactement la fonction hauteur de la blockchain mais une fonction qui augmenterait de 1 chaque fois qu'un nouveau bloc est enregistré dans la blockchain officiel (que ce bloc soit un bloc officiel ou un bloc orphelin détecté par le réseau). En raisonnant comme précédemment, s'il existe une stratégie plus rentable que la stratégie honnête alors il existe une stratégie de durée τ telle que $\mathbb{E}[R(\tau)] > 0$ avec $R(\tau) = G(\tau) - qD(\tau)$.

Notons que tout bloc orphelin détecté par le réseau est nécessairement un bloc miné par les honnêtes mineurs et remplacé par un bloc de l'attaquant suite à une action **Ecraser**.

Ceci conduit à l'étude du jeu de minage modifié noté JMM que nous décrivons ci-dessous.

Au cours de ce jeu, le joueur accumule régulièrement des jetons qu'il peut sous certaines contraintes convertir en argent sonnant et trébuchant (en euros mettons). A chaque instant, le joueur dispose d'au plus trois actions possibles :

Lancer. Un croupier lance une pièce de monnaie truquée en faveur de la banque. La probabilité d'obtenir « Pile » est q .

- Si le résultat est « Pile », le joueur gagne un jeton et ne paye rien.
- Si le résultat est « Face », la banque gagne un jeton et le joueur verse q au croupier.

Ecraser. Cette action n'est possible que si le joueur (resp. la banque) possède a (resp. h) jetons avec $a > h$. Dans ce cas, la banque perd h jetons, le joueur perd $h + 1$ jetons mais gagne $h + 1$ € et donne aussi $q \cdot (h + 1)$ € au croupier. Son résultat net est donc $(h + 1) \cdot (1 - q)$ €.

Abandon. La banque et le joueur perdent tous leurs jetons. Cette action ne coûte rien au joueur.

On note $JMM(i, j)$ le jeu décrit ci-dessus et où le joueur part d'une situation où il possède i jetons contre j pour la banque, $(i, j) \in \mathbb{N}^2$.

La seule différence entre JM et JMM est la conséquence de l'action **Ecraser**. Pour JMM, le joueur doit certes payer pour l'avancée de la blockchain officielle (progression de 1) mais il doit aussi payer pour la création de h blocs orphelins (les h blocs des honnêtes mineurs qui ont été remplacés et visibles par tous). D'où un coût égal à $q \cdot (h + 1)$ à la suite de cette action.

Définition 31. Notons $JMM(a, h, n)$ le jeu modifié à partir de $JMM(a, h)$ qui force le joueur à arrêter sa stratégie après n actions et soit $\Omega(a, h, n)$ le revenu net maximal du joueur qui joue à $JMM(a, h, n)$.

Il est clair que $JMM(0, 0)$ est biaisé au profit du joueur si et seulement s'il existe $n \in \mathbb{N}$ tel que $JMM(0, 0, n)$ le soit aussi, ce qui se traduit par $\Omega(0, 0, n) > 0$. Or, on a le résultat suivant.

Théorème 32. Pour tous entiers a, h, n , on a $\Omega(a, h, n) \leq p \cdot a$ avec $p = 1 - q$.

Démonstration. Le résultat est vrai si $n = 0$. Supposons le vrai au rang $n - 1 \geq 0$. Alors,

$$q\Omega(a + 1, h, n - 1) + p \cdot (\Omega(a, h + 1, n - 1) - q) \leq q \cdot p \cdot (a + 1) + p \cdot (p \cdot a - q) = p \cdot a$$

Donc, si $a > h$,

$$\begin{aligned} \Omega(a, h, n) &= \text{Max}\{(h + 1) \cdot (1 - q) + \Omega(a - h - 1, 0, n - 1), q\Omega(a + 1, h, n - 1) + (1 - q) \cdot (\Omega(a, h + 1, \\ &\quad n - 1) - q)\} \\ &\leq \text{Max}\{p(h + 1) + p(a - h - 1), pa\} = pa \end{aligned}$$

et si $a \leq h$,

$$\begin{aligned} \Omega(a, h, n) &= \text{Max}\{\Omega(0, 0, n - 1), q\Omega(a + 1, h, n - 1) + (1 - q) \cdot (\Omega(a, h + 1, n - 1) - q)\} \\ &\leq \text{Max}\{0, pa\} = pa \end{aligned}$$

D'où le résultat. \square

On obtient enfin le corollaire suivant.

Corollaire 33. Pour tout entier n , $\Omega(0, 0, n) = 0$.

Ainsi, le jeu du minage modifié $JMM(0, 0)$ n'est pas biaisé, ce qui montre que si la connectivité est nulle ($\gamma = 0$) alors la meilleure stratégie est la stratégie honnête pour le mineur qui prend part au réseau Bitcoin modifié.

Chose remarquable, on peut montrer que ce résultat subsiste même si la connectivité du mineur est non nulle.

Théorème 34. Quelle que soit la stratégie de minage choisie avec $\mathbb{E}[\tau] < \infty$ et quelle que soit la connectivité de l'attaquant, on a

$$\mathbb{E}[G(\tau)] \leq q \mathbb{E}[D(\tau)]$$

Démonstration. Au cours d'un cycle d'attaque, on distingue parmi les $N'(\tau)$ blocs minés par l'attaquant le nombre Orph_A de blocs orphelins et le nombre Off_A de blocs officiels. On note de même $N(\tau)$ les blocs minés par les honnêtes mineurs et parmi eux Off_H et Orph_H les nombres de blocs officiels et orphelins. En particulier, on a :

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

et de plus, $G(\tau) = \text{Off}_A$. Tous les blocs orphelins des honnêtes mineurs sont publics et seront enregistrés tôt ou tard dans la blockchain officielle. Seuls les blocs orphelins de l'attaquant peuvent rester secrets. Donc,

$$\text{Off}_A + \text{Off}_H + \text{Orph}_H \leq D(\tau)$$

Les deux processus N et N' sont des processus de Poisson de paramètres $\lambda \cdot p$ et $\lambda \cdot q$ où λ est un paramètre du à l'ajustement de difficulté ($\lambda = \frac{d}{\tau_0}$ avec $\tau_0 = 10$ minutes et d est un facteur d'ajustement de difficulté). En temps normal, si sur le réseau tous les mineurs sont honnêtes, alors $\lambda = \frac{1}{\tau_0}$ avec $\tau_0 = 10$ minutes. Donc, la condition $\mathbb{E}[\tau] < \infty$ entraîne $\mathbb{E}[N(\tau)] = \lambda p \mathbb{E}[\tau]$ et $\mathbb{E}[N'(\tau)] = \lambda q \mathbb{E}[\tau]$. Cela découle du fait que si M est un processus de Poisson de paramètre α alors le processus compensé $M(t) - \alpha t$ est une martingale. Donc,

$$p \mathbb{E}[\text{Off}_A] \leq p \mathbb{E}[N'(\tau)] = p \lambda q \mathbb{E}[\tau] = q \lambda p \mathbb{E}[\tau] = q \mathbb{E}[N(\tau)] = q \mathbb{E}[\text{Off}_H] + q \mathbb{E}[\text{Orph}_H]$$

ce qui entraîne :

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

D'où le résultat. □

Corollaire 35. *Dans le cadre d'un Bitcoin modifié, la stratégie la plus rentable est toujours la stratégie honnête, quelque que soit la connectivité du mineur.*

9 Log-contrats discrets - d'après Tadge Dryja

Nous avons évoqué en introduction le cas des assurances paramétriques qui se déclenchent automatiquement si une certaine condition est vérifiée. Ce type d'assurance est utilisé principalement pour couvrir des risques liés à des catastrophes naturelles. Des agriculteurs par exemple peuvent avoir intérêt à couvrir un tel risque. Les contrats sont proposés par une compagnie d'assurance qui accepte le risque contre rémunération. De manière générale, un contrat type repose sur un oracle qui à une certaine date (fixe ou non) délivre un message et peut entraîner un versement d'argent. C'est aussi le principe par exemple des contrats forward en finance qui dépend d'un indice et d'une maturité. Plus généralement, c'est celui des produits dérivés. La plupart du temps, ces contrats qui mettent en jeu un oracle ne sont pas décentralisés. Il y a une partie tierce qui se porte garante des transactions. Dans le cas des « futures », c'est organisé à travers une chambre de compensation (et la contrepartie est du reste inconnue) ; chaque variation de l'indice donne lieu à des appels de marge. La finance de marché traditionnelle fonctionne grâce à des intermédiaires. L'intermédiation bancaire par exemple dont le rôle consiste à mettre en relation des prêteurs et des emprunteurs en est un exemple significatif.

L'exemple récent de la crise du libor a aussi montré que l'oracle pouvait parfois être corrompu par les différents acteurs. L'enjeu était bien sûr considérable puisque le libor intervenait (intervient toujours) dans des contrats (essentiellement des swaps) dont la somme totale atteignait paraît-il des centaines de milliers de milliards de dollars. Au final, nombreux sont les acteurs qui en ont fait les frais...

9.1 Scalabilité et anonymat

Voici un exemple concret. Imaginons qu'Alice et Bob parient sur un indice XYZ. Si à une date T, XYZ est en dessous d'une certaine valeur v_{ref} , alors Alice s'engage à verser à Bob en bitcoins autant de points d'indice inférieur à v . Sinon, si l'indice est entre v et une valeur V_{max} , Bob s'engage à verser à Alice en bitcoin chaque point d'indice au-dessus de v . Au total, il n'y a qu'un nombre fini de résultats possibles.

De tels contrats dépendant d'un oracle existent naturellement sur des blockchains comme Ethereum même s'il y en a peu sur Bitcoin car un oracle n'a pas naturellement sa place sur ce réseau. Cependant, la plupart du temps, ces contrats sont enregistrés dans la blockchain, ce qui pose des problèmes d'anonymat et d'échelle (« scalabilité »). L'oracle qui voit naturellement le contrat enregistré dans la blockchain pourrait être tenté de biaiser son message pour favoriser un participant. De plus, le nombre de transactions par bloc est limité et le réseau pourrait être saturé en cas de succès avec un grand nombre de contrats dérivés enregistrés en plus des transactions naturelles qui y circulent déjà. La proposition de log-contrat discret (nous traduisons ainsi « discreet log contract ») répond à ces problèmes dans la mesure où le contrat lui-même n'est pas divulgué [1]. Il est tenu « hors chaîne » par ses participants. D'où le terme « discret ». Au final, le monde extérieur ne voit que des transactions normales.

Le mécanisme repose sur une astuce cryptographique semblable à celle utilisée pour créer des canaux de paiement bidirectionnel hors chaîne. L'avantage de ces canaux est que deux contreparties peuvent s'échanger régulièrement de l'argent sans avoir besoin de se faire confiance et sans la nécessité d'enregistrer régulièrement ces transactions dans la blockchain. Ces canaux de paiement sont l'un des piliers du « Lightning Network », une surcouche du réseau Bitcoin que nous avons déjà évoquée.

Dans le cas des canaux de paiement, c'est un échange de « secrets » qui permet de mettre à jour ces canaux. Pour les log-contrats discrets, c'est un message signé par l'oracle. De manière technique, il faut que celui-ci possède une paire de clé basée sur un schéma cryptographique de Schnorr.

Voici l'idée. Appelons Olivia l'oracle. Au début du contrat, Olivia exhibe une clé publique qu'elle va utiliser à maturité pour signer le résultat. Par exemple, à cette date, elle signe un message donnant la valeur de l'indice XYZ. La signature s de ce message (qui bien sûr dépend du résultat obtenu) permet au vainqueur X du contrat (Alice ou Bob) de déplacer des fonds vers une adresse $\text{Pub}_X + s \cdot G$ qu'il est le seul à contrôler (G est un générateur de groupe). Le fait est que les différentes signatures s possibles sont inconnues mais les $s \cdot G$ le sont dès le début. Il s'ensuit que l'ensemble des $\text{Pub}_X + s \cdot G$ pour chaque possible message m est connu au début du contrat mais pas $\text{Priv}_X + s$. La signature s à maturité peut s'interpréter comme le « logarithme » du contrat. D'où le nom « log-contrat ». Le mécanisme complet est basé sur cette idée. Au début du contrat, Alice et Bob rédigent des tas de transactions suivant tous les résultats possibles. Il y a donc autant de transactions que de résultats possibles. Il faut aussi ajouter la possibilité d'un retour des fonds si aucune action n'est prise. Ces propositions de transaction ne sont pas diffusées sur la blockchain mais signées en partie et transmises à l'autre. Ce sont les analogues des « commit transactions » dans la création d'un canal de paiement. Une fois rédigées, Alice et Bob peuvent maintenant envoyer chacun des fonds vers un contrat multisig. C'est l'analogue de la transaction d'ouverture dans la création d'un canal de paiement sur le « Lightning Network ». Cette transaction est inscrite dans la blockchain mais n'est rien d'autre qu'une transaction multisig normale. Impossible de détecter la présence d'un contrat à terme mettant en jeu un oracle.

A la fin du contrat, seule une transaction sera valide. La divulgation du résultat et du message s signé par Olivia va permettre sans ambiguïté de déplacer l'argent contenu dans le contrat multisig vers le vainqueur du contrat qui est le seul à posséder la clé privée $\text{Priv}_X + s$ car s est révélé à la maturité du contrat. Le bénéficiaire peut simplement récupérer ses fonds en enregistrant cette transaction gagnante. Alice et Bob peuvent aussi convenir de recommencer un autre log-contrat avec le même ou un autre oracle. Inutile alors d'enregistrer quoi que ce soit dans la blockchain. Le canal ouvert entre Alice et Bob sera mis à jour. Ils gardent leur contrat hors chaîne avec la possibilité d'y revenir à n'importe quel moment. De cette façon, on peut imaginer une véritable finance décentralisée sur Bitcoin avec des chambres de compensation totalement décentralisées. C'est en tout cas ce que rend possible cette proposition de Tadge Dryja. Elle rencontrera peut-être le même succès que son autre invention, le « Lightning Network » [2].

10 Conclusion

Nous avons cherché à savoir quel rôle pouvaient jouer les assureurs dans le nouveau monde des cryptomonnaies. Ils peuvent simplement assurer des plateformes d'échanges mais pour cela, ils doivent faire l'effort de comprendre les produits proposés, ce qui demande de passer en revue la sécurité de différents protocoles. Nous avons brièvement étudié celui de Bitcoin dans ce mémoire. Nous avons mis en évidence une faille dans la formule d'ajustement de difficulté.

Nous avons montré que l'étude de la sécurité des protocoles comme celui du Bitcoin, basés sur l'utilisation répétée de preuves de travail peut s'étudier en considérant une variante biaisée au profit du joueur du jeu classique de Pile ou Face.

Nous avons donné un algorithme simple qui permet d'évaluer le seuil au-delà duquel un mineur n'a plus intérêt à se comporter de manière honnête sur le réseau Bitcoin. A ma connaissance, seuls des solveurs de problèmes de décisions markoviennes avaient jusqu'ici été utilisés pour résoudre ce problème.

Nos calculs numériques montrent que le seuil vaut environ 32.94% lorsque le mineur a une connectivité nulle.

Bitcoin ne représente certes pas tout le monde des cryptomonnaies mais son protocole basé sur les preuves de travail est pour l'heure majoritaire. D'autres cryptomonnaies comme NXT, Tezos ou bientôt Ethereum 2.0 fonctionnent (ou promettent de fonctionner) de manière radicalement différente. Elles présentent l'avantage de se passer de débauche énergétique nécessaire pour sécuriser le registre des transactions. Il n'y a plus de « mineur » obligé de gaspiller de l'énergie pour éventuellement trouver le prochain bloc qu'il pourra ajouter ensuite à la blockchain officielle. Dans le cadre de ces protocoles, chacun peut proposer un nouveau bloc et la probabilité d'être accepté est proportionnelle à la richesse que l'on possède. On n'a pas à faire tourner des machines jusqu'à obtenir une preuve de travail. L'heureux élu est sélectionné en fonction de sa participation dans le réseau et concrètement de la richesse qu'il possède. De tels protocoles sont ainsi basés sur des preuves de participation. On parle aussi parfois preuve d'enjeu (« Proof of Stake » en anglais).

Pendant, il existe notamment une attaque dévastatrice sur de tels réseaux, très simple à mener. Elle consiste à réécrire entièrement la blockchain depuis le début et à en produire une autre de taille plus importante. Ces attaques portent le nom de « Long-range attack » en anglais. Elles sont très faciles à réaliser étant donné qu'il est très simple de construire des blocs. Contrairement à Bitcoin, cela ne coûte rien. C'est pourquoi, en général, de tels protocoles possèdent en plus de la première phase de sélection basée sur des preuves de participation, une seconde phase dite de validation. Dans cette phase, un certain nombre de validateurs bien identifiés votent pour accepter ou non le bloc. Il s'agit pour ces noeuds particuliers de se mettre d'accord sur un bloc. On retrouve un problème de recherche de consensus bien connu en informatique car ici les validateurs à chaque date de validation sont connus. C'est le problème classique des généraux byzantins pour un système distribué fermé. Plusieurs solutions ont été proposées depuis les travaux initiaux de Leslie Lamport dans les années 80. On parle d'algorithmes « Byzantine Fault Tolerant ». Mais tous ces consensus qui existaient avant Bitcoin, (certains sont sophistiqués) sont des consensus dits « à autorisation » (permissionned en anglais). Le problème est : étant donné un nombre *déterminé* de noeuds (comme un ensemble de généraux byzantins), comment parvenir à un consensus ? Le problème de Bitcoin n'est pas celui-là puisque le nombre de mineurs comme de validateurs est à chaque instant aléatoire. On n'a pas besoin sur Bitcoin de demander l'accord de validateurs pour proposer un nouveau bloc. Chacun peut rentrer ou sortir du réseau comme dans un moulin. Bitcoin réalise un consensus dans un univers où personne n'a besoin de demander d'autorisation. L'algorithme de consensus est dit « permissionless » en anglais. Cela crée une différence notable entre tous les projets. Certes, on peut imaginer des projets « verts » sans débauche énergétique mais c'est au détriment de la décentralisation du réseau, avec une phase de validation. A notre connaissance, aucun des projets concurrents à Bitcoin n'atteint le même niveau de décentralisation. C'est pourquoi, bitcoin est en train de devenir ce qu'était l'or autrefois mais dans un univers numérique. C'est aussi pourquoi tôt ou tard, des banques centrales seront amenés à conserver des stocks de bitcoin comme autrefois des stocks d'or. Certains acteurs économiques dont aussi des assureurs l'ont compris et ont déjà placé une partie de leur trésorerie en bitcoin, malgré son cours très volatile.

Nous avons vu que les assureurs aussi peuvent jouer un rôle dans la nouvelle finance décentralisée. Cela peut être assurer des smart-contracts compliqués contre un risque de bug mais aussi assurer des NFT et autres produits pouvant par exemple intervenir dans le monde des jeux vidéos. Enfin, nous avons expliqué comment la proposition de log-contrat discret de Tadge Dryja pouvait potentiellement conduire à une véritable finance décentralisée « hors chaîne » sur le modèle du « Lightning Network » sur Bitcoin . On peut ainsi imaginer une véritable plateforme d'achat et de suivi pour les contrats d'assurance paramétriques.

Remerciements. Je remercie vivement la direction du CNAM, Alexis Collomb et Sandrine Lemery pour m'avoir donné l'occasion d'écrire un mémoire sur les cryptomonnaies. Les résultats présentés ici sont connexes à des études menées avec mon collaborateur et ami Ricardo Pérez-Marco. Je dédie naturellement ce travail à Michel Fromenteau.

Bibliographie

- [1] T. Dryja. Discreet log contracts. *Adiabat.github.io/dlc.pdf*, 2017.

- [2] T. Dryja et J. Poon. The bitcoin lightning network: scalable off-chain instant payment. *Light-ning.network/docs/*, 2016.
- [3] I. Eyal et E. Sirer. Majority is not enough: bitcoin mining is vulnerable. *Financial Cryptography and Data Security*, pages 436–454, 2014.
- [4] C. Grunspan et R. Pérez-Marco. Double spend races. *Int. Journal Theoretical and Applied Finance*, 21(08), 2018.
- [5] C. Grunspan et R. Pérez-Marco. On profitability of selfish mining. *Arxiv.org/abs/1805.08281v3*, 2018.
- [6] C. Grunspan et R. Pérez-Marco. On profitability of nakamoto double spend. *Probability in the Engineering and Informational Science*, pages 1–15, 2021.
- [7] S. Haber et W. Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.
- [8] S. Nakamoto. Bitcoin: a peer-to-peer electronic cash system. *www.bitcoin.org/bitcoin.pdf*, released on November 1st 2008 on the USENET Cryptography Mailing List "Bitcoin P2P e-cash paper".
- [9] RHorning. Mining cartel attack. *Bitcointalk.org/index.php?topic=2227.0*, 2010.
- [10] M. Rosenfeld. Analysis of bitcoin pooled mining reward systems. *ArXiv:1112.4980*, 2011.
- [11] A. Sapirstein, Y. Sompolinsky, et Zohar A. Optimal selfish mining strategies in bitcoin. *International Conference on Financial Cryptography*, 2016.
- [12] N. Szabo. Smart contracts. *Szabo.best.vuh.net/smart.contracts.html*, 1994.
- [13] R. Wattenhofer. *Blockchain Science*. Inverted Forest Publishing, 2019.