A-024

グラフクリーク探索問題を利用した仮想通貨採掘時間の分散解析
Analysis about variance of cryptocurrency mining time using graph creek search problem

池辺 慶† 櫻井 幸一†
Kei Ikebe Kouichi Sakurai

1. はじめに

2009 年にサトシナカモトが開発したビットコイン[8]は取引の正当性を保障するためにプルーフオブワークという合意形成手段を用いている.これは、計算困難な問題を解き、その解を求めた者が報酬を得る合意形成アルゴリズムのことである.ビットコインではナンスの探索という計算困難な問題をプルーフオブワークに用いており、その探索をマイニングと呼ぶ.プルーフオブワークによる合意形成は第三者による取引の正当性の保障が必要ないため、円やドルなどの法定通貨と比べると取引の際に必要となる手数料が非常に安くて済むという利点がある.また、プルーフオブワークによる合意形成で取引の偽装を行うためにはマイニングを行っている計算機の計算能力の過半数を占める必要があり、現実的ではないことからセキュリティ上の安全性も保障されている.

ビットコインのマイニングはナンスの探索によって行われるが、その探索時間の期待値は事前に設定した難易度によって決められている.期待値よりも非常に早い時間でマイニングを完了させることが出来てしまうと、攻撃者によってブロックの分岐などを引き起こされる恐れがあることから、マイニングにかかる時間の分散は小さいことが望まれる.しかし、ビットコインではその性質上マイニングにかかる時間の分散が大きくなってしまうという問題点[10]が存在する.

本研究では既存研究で提案されたハッシュ関数の連結による小分散化[1]とクリーク探索を利用したマイニングによる小分散化[7]の二つの手法を組み合わせることでさらなる小分散化を実現した.

2. 背景

2.1 暗号学的ハッシュ関数

ハッシュ関数[11]とは入力された値に対して全く異なる一定長の出力を返す関数である.特に、暗号学的ハッシュ関数とは情報セキュリティの用途に適するような暗号数理的性質を持つハッシュ関数のことを表す.暗号学的ハッシュ関数は以下の性質を有する.

1.入力される値が少しでも異なる時、全く異なる値を返す.$\left(hash(a) \neq hash(a')\right)$
2.出力から入力を推測するのが困難である.(一方向性)
3.ハッシュ値$hash(a) = A$が分かっているとき、出力がAとなるa以外の入力bを求めるのが困難である.(原像計算困難性、弱衝突耐性)
4.$hash(a) = hash(b)$のようなハッシュ値が一致する異なる2つの入力を見つけるのが困難である.(強衝突耐性)

†九州大学 Kyushu University

2.2 ブロックチェーン

ブロックチェーンは 2008 年にサトシナカモトによってビットコインの公開取引台帳として用いるために開発された、ブロックと呼ばれるデータ群を鎖のようにつなげて管理する仕組みである.各ブロック中にはヘッダーハッシュ(前のブロックのハッシュ値)、タイムスタンプやトランザクションデータなどのデータ、ナンス等が記録される.

ブロックチェーンの特徴としてデータの改ざんに強い事が挙げられる.ブロック中には前のブロックのハッシュ値が記録されるため、仮に過去のブロックが改ざんされたらそのブロックのハッシュ値が変わってしまう.そのため、芋づる式に現在のブロックまで影響を及ぼしてしまう.トランザクションデータはハッシュ関数を用いて複数の取引データの管理をしているため、取引データを改ざんしようとした場合、ブロックと同様にハッシュ値が変わってしまうため不正を行うのは難しくなっている.

2.3 マイニング

マイニングとはブロックチェーンにおいて次のブロックを作成するために条件を満たすようなナンスを探索する作業である.具体的には、事前に設定した難易度Dをもとに、以下の不等式を満たすようなナンスを探索する.i番目のブロックのヘッダーハッシュh^iを、取引などのデータをT^i、ナンスを$nonce^i$と記し、ビット列aとbの連結を$a \,||\, b$と記す.

$$hash(h^i \,||\, T^i \,||\, nonce^i) < D$$

ハッシュ値はハッシュ関数の特性から、ナンスの値が少しでも異なると全く異なるハッシュ値が出力される.また、出力される値を元に入力を逆算することも難しいため、条件を満たすナンスを探索するためにはしらみつぶしにナンスを入力し、条件を満たすハッシュ値が出力されるまで探索を行う必要がある.

2.4 マイニング時間の分散

マイニング時間の分散について考える.条件を満たすナンスの探索という試行は各試行間に相関関係はなく、独立な試行である.つまり、ナンスの探索は解を発見するか否かのベールヌイ試行である.よって、探索が終了するまでに行う試行回数の確率分布は幾何分布である.幾何分布は離散的な確率分布であるが、マイニングにおける試行回数は十分大きく、また、成功確率は十分小さい.このことから探索が終わってから次の探索が終了するまでの時間は指数分布に近似できると考えられる[2].よってビットコインのマイニング時間の分散は期待値の 2 乗となる.これはナンスの探索が一瞬で終わる場合や、逆に期待値の倍近い時間かかる場合が存在することを意味する.分散が大きいと幸運な攻撃者に

よるブロックチェーンの分岐などが発生する可能性や、実用上での不便さなどが発生する[3].

3. 既存研究

3.1 ハッシュ関数に基づく計算問題に対するマイニング時間の小分散化～直列連結及び並列連結～

2020 年に発表されたこの論文[1]は、ビットコインで行われるハッシュ関数を用いたナンスの探索によるマイニングを複数連結する事でマイニング時間の小分散化を試みる.連結の方法は直列、並列の 2 種類存在する.本論文では今回用いる直列連結について説明する.

直列連結ではブロックチェーンにおける n個のマイニングを直列連結し、新たな 1 つのマイニングとして行う.このマイニングで取り扱う取引のデータは T^iのみである.このマイニングで求めるのは n 個のナンスの組である.また、この時解くべき計算困難な問題は以下の連立不等式である.

$$\begin{cases} h_1^i = hash(h_1^{i-1} \,||\, T^i \,||\, 1 \,||\, nonce_1^i) < D \; and \\ h_2^i = hash(h_1^i \,||\, T^i \,||\, 2 \,||\, nonce_2^i) < D \; and \\ \cdots \\ h_n^i = hash(h_{n-1}^i \,||\, T^i \,||\, 1 \,||\, nonce_n^i) < D \end{cases}$$

次にこのマイニング方法におけるマイニング時間の分散について考える.$X_1, X_2, \ldots, X_n$をブロックチェーンにおける n 個の連結したブロックそれぞれのマイニングにかかる時間を表す確率変数とする.X_iは通常のビットコインと同じ確率分布であるため、指数分布に従うと考えられる.Y を n 個のマイニングを連結した際にかかるマイニングの時間を表す確率変数とすると、連立不等式を満たすようなナンスの探索を行うためには $nonce_1^i$から$nonce_n^i$まで順番に探索を行う必要がある.つまり、探索にかかる時間 Y は$X_1, X_2, \ldots, X_n$の和となる.つまり、

$$Y = X_1 + X_2 + \ldots + X_n$$

この時次の定理が成り立つ

定理 ビットコインのマイニングにかかる時間の期待値をμ_0、分散をσ_0^2とする.直列連結によるマイニングにかかる時間の期待値をμ分散をσ^2とする.$\mu = \mu_0$のとき、

$$\sigma^2 = \sigma_0^2/n$$

証明 直列連結によるマイニングでn個連結されている各ブロックのマイニングにかかる時間はそれぞれ $X_1, X_2, \ldots, X_n$であり、X_i、$i = 1, 2, \ldots, n$ は同一の指数分布に従う.また、それぞれの試行は独立である.それらの期待値をμ'、分散をσ'^2とすると、$n \to \infty$のとき、和の確率変数 Y は期待値$n\mu'$、分散$n\sigma'^2$の正規分布 $N(n\mu', n\sigma'^2)$に近似的に従う.これを中心極限定理[9]と呼ぶ.よって、

$$\begin{aligned} \sigma^2 &= \mathrm{n}\sigma'^2 \\ &= \mathrm{n}\mu'^2 \\ &= (n\mu')^2/n \\ &= \mu^2/n \\ &= \mu_0^2/n \\ &= \sigma_0^2/n \end{aligned}$$ (fin)

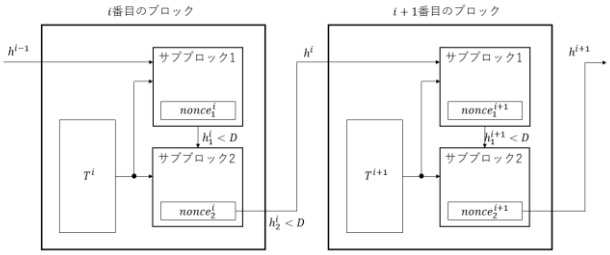

図 1 マイニングの直列連結

3.2 グラフクリーク探索問題による仮想通貨の評価

グラフ理論において無向グラフ$G = (V, E)$のクリークとは、頂点の部分集合$C \subseteq V$のうち、Cに属する任意の 2 頂点を結ぶ辺が存在するような頂点の集合のことを呼ぶ.クリークに属する頂点数をそのクリークの大きさと呼び、k クリークとは大きさkのクリークのことである.与えられたグラフ中に指定された大きさのクリークがあるかどうかを求めることをクリーク探索問題と呼び、特にグラフ中で最も大きいクリークを探す問題を最大クリーク問題と呼ぶ.

クリークを用いたマイニングでは、取引などのトランザクションデータをシード値のよう

に扱うことでグラフを作成する.作成したグラフ中に事前に設定した大きさのクリークがあるか探索し、そのクリークを求めることをビットコインのマイニングで行われるナンスの探索の代わりに行う.

作成するグラフの頂点数をv、ブロックに格納されるデータをT、求めるクリークの大きさをk、頂点間が辺を持つ難易度をD とするとクリーク探索によるマイニングは以下のようにして行われる.

1. 頂点をV_n、$n = 1, 2, \ldots, v$ と記す時、$V_1 = (T||1), V_2 = (T||2), \ldots, V_v = (T||v)$ として頂点を v 個作成する.
2. 2 頂点V_a, V_bにおいて以下の不等式を満たすとき、その頂点間に辺を作成する.
$$hash(V_a \,||\, V_b) < D$$
3. 1、2 で作成したグラフから大きさkのクリークを探索する.

既存研究[6]の実験から、クリーク探索によるマイニングはビットコインで用いられているナンスを用いたマイニングよりも分散が小さくなることが確認されている.

4. クリーク探索によるマイニングの直列連結の実装

4.1 アルゴリズム

基本的な考え方は直列連結と同じで、簡単な探索を複数回行い、その探索時間の和をマイニングにかかった時間とする.マイニングで取り扱う取引のデータをT^i、頂点数をv_n、連結度をn、難易度をD、求めるクリークの大きさをk、ヘッダーハッシュをh とすると以下のようにして探索を行う.

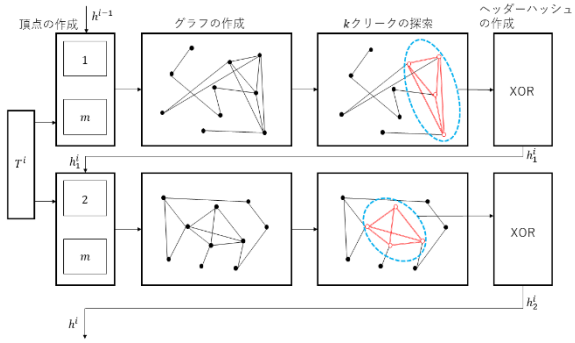

図 2 クリーク探索によるマイニングの直列連結 (頂点数 10、連結度 2、k = 4 の場合)

1. 頂点をV_m、$m = 1,2,\ldots,v_n$、連結した時の段数をyと記す時、$V_m = (T^i \,||\, h \,||\, y \,||\, m)$として頂点を$v_n$個作成する.
2. 2 頂点V_a, V_bにおいて以下の不等式を満たすとき、その頂点間に辺を作成する.
$$hash(V_a \,||\, V_b) < D$$
3. 1、2 で作成したグラフから大きさkのクリークを探索し、その頂点群を$V_1', V_2', \ldots, V_k'$とする.
4. 次のヘッダーハッシュは
$$h = hash(XOR(V_1', V_2', \ldots, V_k'))$$
とする.
5. ヘッダーハッシュを用いてマイニングを直列化させ、次の探索を行う.

次に、頂点数と難易度の設定について説明する.グラフの作成にかかる時間は頂点数の 2 乗に比例するため、連結度nの時グラフ作成にかかる時間は$\alpha n v_n^2$である(αは比例定数).$v_n = rv_1$とし、$\alpha n v_n^2 = \alpha v_1^2$となるとき、$r = 1/\sqrt{n}$となる.次に難易度の設定で

あるが、連結度nの時に作成されるグラフ中に期待されるクリークの数をnum_nとすると、こちらは$num_n = num_1$となるようなDを設定する必要がある.連結度nの時の難易度D_nとし、これによって決まる辺を作成する確率をp_nとする.また、$p_n = gp_1$とするとき、比例定数gの導出は以下のようにして行う.

導出 連結度 1 の時のグラフ中に存在するkクリークの期待される数num_1は$num_1 = {}_{v1}C_k p_1^{k(k-1)/2}$となる($p_1$は難易度$D$によって定まる頂点間が辺を持つ確率).同様にして$num_n$を求めることができる.この時の頂点間が辺を持つ確率をp_nとし、num_nの式を展開すると

$$num_n = {}_{vn}C_k p_n^{k(k-1)/2}$$
$$= {}_{v1/\sqrt{n}}C_k p_n^{k(k-1)/2}$$
$$= {}_{v1/\sqrt{n}}C_k (gp_1)^{k(k-1)/n}$$

となる.よって求めるgは

$${}_{v1/\sqrt{n}}C_k (gp_1)^{k(k-1)/n} = {}_{v1}C_k p_1^{k(k-1)/2}$$
$$(gp_1)^{k(k-1)/2} = ({}_{v_1}C_k / {}_{v1/\sqrt{n}}C_k) p_1^{k(k-1)/2}$$
$$g = ({}_{v_1}C_k / {}_{v1/\sqrt{n}}C_k)^{2/k(k-1)}$$

となる.

表 1 実装環境

メモリ	16.0GB
CPU	Intel Core i7-10750H CPU @ 2.60GHz 2.60GHz
使用言語	Pyhton3

4.2 実験

本実験では連結度 1 の時、頂点数$v = 2^{14}$、求めるクリークの大きさ$k = 4$、難易度$D = 2^{248}$という条件で行う.また、実験の環境は(表 1)である.本実験では 100 回探索を行い、その探索時間を記録し、また、平均時間と分散を記録した.結果を(表 2) に示す.

結果から連結度が大きくなるにつれて分散が小さくなることが分かった.一方で、平均時間は連結度 1 と 8 で 100 秒以上の差があった.以上のことからマイニングにかかる時間を大きく変化させることなく分散を小さくすることには成功したが、厳密に連結させることは出来なかったと考えられる.

4.3 考察

平均時間が一定にならず、連結度を大きくするごとに小さくなってしまった原因を考察する.原因は複数考えられるが、まず、頂点数または難易度の設定が適正ではなかった可能性があげられる.クリーク探索によるマイニングにかかる時間はグラフ作成にかかる時間とグラフの探索にかかる時間の和となる.このマイニングの連結を行う際の頂点数の設定はグラフ作成にかかる時間を均一化するよう設定し、クリーク探索にかかる時間については考慮しなかった.理由としては頂点作成の時間は完全に頂点数に依存するのに対して、クリーク探索にかかる時間の期待値に関しては頂点数は関係が薄いと考えたためである.

線形探索において頂点数は最悪の計算時間には影響を及ぼすが、その期待値には大きく影響を与えない.なぜなら、クリークを発見し次第探索を終了するためグラフのサイズは関係無いからである.次に難易度であるが、頂点数に関わらずグラフ中に一定数のk クリークが存在するよう設定した.実験ではグラフのサイズに関わらずグラフ中に 10 個のkクリークの存在が期待できるようにした.しかし、本来は連結度が大きくなるごとに期待されるkクリークの数は減らすべきである.ただ、このkクリークの期待される数は大きすぎても小さすぎても意味がなく、また、多少のブレが存在するため、連結度を大きくした際に期待されるkクリークの数を小さくしてしまうとグラフ中にk クリークが存在しなくなってしまう可能性がある.そのため意図的にグラフ中に存在するクリークの数を頂点数に関わらず一定にした.結果としてグラフのサイズに対してkクリークの数が大きくなってしまい探索時間が短くなってしまったと考えられる.以上のことから難易度の設定を正しく設定出来てなかった事が原因と考えられる.

表 2 クリーク探索によるマイニングの直列連結

連結度	1	2	3	4	5	6	7	8
平均時間(s)	428.2298871	344.0079191	331.8718539	327.8646506	318.2387817	313.6074561	318.9917718	313.2920291
分散(s^2)	20360.021	1113.614495	368.0165289	181.6365718	67.44465602	55.73393739	36.99443384	26.65008763

対策としては連結度 1 のときのグラフのサイズを大きくし、同時に、グラフ中に存在が期待されるkクリークの数も大きくすることで多少連結してグラフ中のkクリークの期待される数を小さくしても問題をなくすことで解決されると考えられる.一方で、やはりkクリークの期待される数が大きくなりすぎても小さくなりすぎても探索の意味をなさないため、連結には限度が存在すると考えられる.

5. 結論

2009 年にサトシナカモトが考案されたビットコインは現在でも利用されており、また、ビットコインの考案以降に実用化された仮想通貨にも大きな影響を与えている.ビットコインにおける取引はプルーフオブワークによって合意形成が行われており、その際にマイニングが行われる.ビットコインにおけるマイニングとは、取引等のトランザクションデータや前のブロックのハッシュ、ナンス等をハッシュ関数に入力し、出力されるハッシュ値が一定の値以下になるようなナンスを探索することを言う.つまり、ナンスの探索という計算困難な問題の解を求めることがビットコインにおけるマイニングである.マイニングの難易度は求めるハッシュ値の大きさの上限によって決定され、その難易度によってマイニングにかかる時間の期待値は決定される.例えば、ビットコインの場合はマイニングにかかる時間の期待値は 10 分である.また、難易度により期待値のほかに分散も決定される.ビットコインの探索時間は、ハッシュ関数の特性より、その分布は指数分布に従う.したがって、ビットコインの分散は期待値の 2 乗という大きな値になる.分散が大きいとセキュリティ上の問題や実用上の不都合が生じる.

ビットコインのマイニング時間の分散を小さくするため、2 つの既存研究では異なるアプローチで小分散化を行っていた.1 つ目はマイニングを複数連結し分散を小さくする方法である.この方法は既存のマイニングよりも少し簡単なマイニングを複数回行うことで、そのマイニングにかかる時間のブレを小さくするという方法論である.2 つ目は、マイニングをナンスの探索ではなく、クリークの探索によって行う方法である.こちらは従来の物とは異なる計算困難な問題を解くことでマイニングを行うという方法論である.

本研究ではクリーク探索によるマイニングの直列連結を行った.実験の結果は完璧な直列化にはならなかったが、目的に近いものは実装できた.結果としてクリーク探索によるマイニングからさらに分散を小さくすることに成功したが、同時に探索にかかる時間の期待値も小さくなってしまい、純粋に直列化とは言えない結果になった.

本研究では全て 1 台の PC を用いて実験を行ったが、実際のマイニングは複数台の PC を用いて、並列に計算を行う.その場合での小分散化の効率については要検証である.また、本実験で行った全ての方法で小分散化の効果は確認されたが、一方で、セキュリティ上での問題点には考慮しなかった.実際にシステムとして運用する場合、安全性も保障する必要がある.クリーク探索によるマイニングの直列連結は難易度の設定を正しく設定出来なかったため、厳密に設定した場合の小分散化の効果の確認とクリーク探索によるマイニングの並列化についても今後の課題である.

謝辞

本研究を進めるにあたり、ご多忙にも関わらず、熱心にご指導いただきました櫻井幸一教授に心から感謝致します.また、研究にご協力いただきました櫻井研究室の皆様に心から感謝いたします. 本研究の一部は JSPS 科研費基盤(B) JP18H03240 の支援を受けています.

参考文献

[1] 穴田啓晃, 櫻井幸一, “ハッシュ関数に基づく計算問題に対するマイニング時間の小分散化～直列連結及び並列連結～”, 信学技報, vol. 120, no. 28, ISEC2020-9, pp. 33-40, 2020.
[2] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, “Block arrivals in the bitcoin blockchain.” CoRR, abs/1801.07447, 2018.
[3] G. Bissias and B. N. Levine. Bobtail, “A proof-ofwork target that minimizesblockchain mining variance (draft).” CoRR, abs/1709.08750, 2017.
[4] bitcoinwiki. Confirmation. https://en.bitcoin.it/wiki/Confirmation (accessed 2021/02/01)
[5] Hiroaki Anada, Tomohiro Matsushima, Chunhua Su, Weizhi Meng, Junpei Kawamoto, Samiran Bag, Kouichi Sakurai,“Analysis of Variance of Graph-Clique Mining for Scalable Proof of Work. ”Inscrypt 2018: 101-114
[6] 松島智洋, 穴田啓晃, 川本淳平, Bag Samiran, 櫻井幸一, “グラフクリーク探索問題に対するビットコイン・マイニングの評価”, 火の国情報シンポジウム.2016.4B-2, 2016.
[7] 松島智洋, “グラフクリーク探索問題による仮想通貨の評価”, 九州大学理学部物理学科情報理学コース卒業論文, 2016
[8] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” https://bitcoin.org/bitcoin.pdf, 2008.
[9] S. M. Ross. “Introduction to Probability Models.” Academic Press, 11th edition, 2014.
[10] Rosenfeld, Meni,“Analysis of bitcoin pooled mining reward systems,”arXiv preprint arXiv:1112.4980, 2011.
[11] Bruce Schneier,“ Applied Cryptography,”John Wiley & Sons, 1996.