複数連結マイニング及びクリーク探索問題による仮想通貨の採掘時間の分散解析

池辺　慶[1]

概要：仮想通貨は取引を行う際にマイニングという計算困難な問題の解の探索を行っている．この探索時間は性質上ブレが生じてしまう．これはセキュリティの観点上好ましくない．本研究では既存研究で提案されたマイニングを実装し，小分散化の効果の確認と評価を行った．また，既存研究の提案方法を組み合わせる事が可能か評価を行った．その結果、既存研究の提案方法のマイニングは確かに小分散化の効果があったが，一方で理論値とは異なる結果となった．また，既存研究の二つの手法を組み合わせることでさらに小分散化の効率を高めることができた．

1. 緒論

1.1 研究背景

2009 年にサトシナカモトが開発したビットコイン [N08] 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 [RM11] が存在する．

既存の研究では，複数のマイニングを連結することで分散を小さくする手法 [AS20] と，ナンスの探索の代わりにグラフ中の k クリーク探索でマイニングを行う手法 [MAK+16] の 2 つが挙げられる．これらの手法は分散の値は従来のマイニング方法よりも小さい事が期待されるが，理論評価式を得られておらず，また，実機上での小分散化の効果についての確認もされていない．

よって本研究では既存研究で提案されたマイニング手法を Python 上で実装し，分散や期待値，確率分布の評価及び実機上での小分散化の効果の確認を行う．また，既存研究の二つの手法を組み合わせる事による小分散化についての効果を確認し，Python 上での実装を行う．

2. 背景

2.1 ハッシュ関数

2.1.1 暗号学的ハッシュ関数

ハッシュ関数 [S96] 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．($hash(a) \neq hash(a')$)
(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)

[1] 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を h^i，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 との連結を $a||b$ と記す．

$$hash(h^i \,||\, T^i \,||\, nonce^i) \;<\; D \tag{1}$$

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．

2.3.1 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 [BKKT18]．単位時間あたりの探索を終了する回数を λ とし，次に探索が終了するまでの時間 X が指数分布に従う時，$X = x$ となる確率密度関数は以下のようになる．

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & (x \geq 0) \\ 0 & (x < 0) \end{cases} \tag{2}$$

マイニングにかかる時間の期待値は事前に設定した難易度を元に決定される．この期待値 $E(X)$ の求め方は，マイニングの難易度を D，出力されるハッシュ値のデータサイズを s ビット，ナンスを発見する確率を p，単位時間あたりの探索試行回数を m 回とすると以下のようになる．

$$p \;=\; \frac{D}{2^m} \tag{3}$$

$$E(X) \;=\; \frac{1}{pm} \tag{4}$$

確率変数 X が指数分布に従う時，その期待値 $E(X)$ と分散 $V(X)$ は以下のようになる．

$$E(X) \;=\; \frac{1}{\lambda} \tag{5}$$

$$V(X) \;=\; \frac{1}{\lambda^2} \tag{6}$$

式 (5),(6) からビットコインのマイニング時間の分散は期待値の 2 乗となる事が分かる．これが意味するのは，ナンスの探索が一瞬で終わってしまう場合もあれば，逆に期待値の倍近い時間がかかってしまう可能性があるということである．このように分散が大きいと幸運な攻撃者によってブロックチェーンの分岐などの攻撃が発生してしまう恐れがある．また，取引にかかる時間にブレが生じてしまい，実用上での不便さなどが発生してしまう [BLB17].

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 [AS20] 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．各連結方法の詳細及び小分散化の効果は以降の章で説明する．

この研究では，連結を行った際の小分散化の効果についての理論評価は行っているが，実際に機械上で動かした際の小分散化の効果については確認されていない．

3.1.1 直列連結

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 $(nonce_1^i, nonce_2^i, ..., nonce_n^i)$ 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．

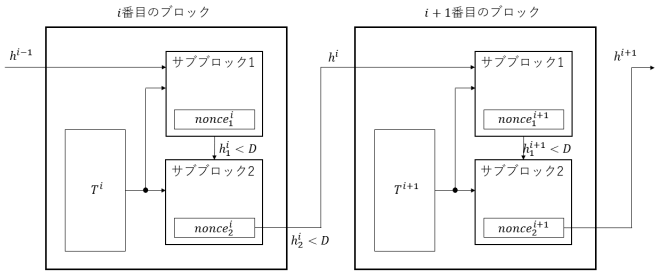

図 1 マイニングの直列連結 (連結度 2 の場合)

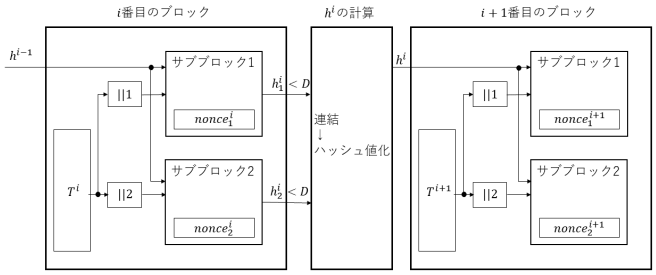

図 2 マイニングの並列連結 (連結度 2 の場合)

$$\begin{cases} h_1^i \ = \ hash(h_n^{i-1} \ || \ T^i \ || \ 1 \ || \ nonce_1^i) \ < \ D \ and \\ h_2^i \ = \ hash(h_1^i \ || \ T^i \ || \ 2 \ || \ nonce_2^i) \ < \ D \ and \\ ... \\ h_n^i \ = \ hash(h_n^i \ || \ T^i \ || \ n \ || \ nonce_n^i) \ < \ D \end{cases} \tag{7}$$

また，このアルゴリズムを図で表すと次のようになる．(図 1)

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．$X_1, X_2, ..., X_n$ 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 (7) 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 $X_1, X_2, ..., X_n$ の和となる．つまり，

$$Y \ = \ X_1 \ + \ X_2 \ + \ ... \ + \ X_n \tag{8}$$

この時次の定理が成り立つ．

定理 1 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，

$$\sigma^2 \ = \ \frac{\sigma_0^2}{n} \tag{9}$$

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 $X_1, X_2, ..., X_n$ であり，X_i，$i \ = \ 1, 2, ..., n$ 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 [R14] と呼ぶ．よって，

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

(fin)

マイニング時間の期待値は事前に難易度を調整することで条件に合わせることが可能である．

3.1.2 並列連結

並列連結も直列連結と同様に，ブロックチェーンにおける n 個のマイニングを並列連結し，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 $(nonce_1^i, nonce_2^i, ..., nonce_n^i)$ 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．

$$\begin{cases} h_1^i \ = \ hash(h^{i-1} \ || \ T^i \ || \ 1 \ || \ nonce_1^i) \ < \ D \ and \\ h_2^i \ = \ hash(h^{i-1} \ || \ T^i \ || \ 2 \ || \ nonce_2^i) \ < \ D \ and \\ ... \\ h_n^i \ = \ hash(h^{i-1} \ || \ T^i \ || \ n \ || \ nonce_n^i) \ < \ D \end{cases} \tag{10}$$

並列連結によるマイニングで用いられるヘッダーハッシュ h^i は以下のようになる．

$$h^i \ = \ hash(h_1^i \ || \ h_2^i \ || \ ... \ || \ h_n^i) \tag{11}$$

また，このアルゴリズムを図で表すと次のようになる．(図 2)

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．$X_1, X_2, ..., X_n$ 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 は直列の場合と同様で，

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，Y は以下のようになる．

$$Y = \max_{1\leq i\leq n}(X_i) \tag{12}$$

並列連結によるマイニングのマイニング時間の分散についての定理を述べるために以下の補題を証明する．

補題 1 X_i，$i = 1,2,...,n$ を期待値 μ 分散 σ^2 の指数分布に従う独立同一分布とする．$Y_n = \max_{1\leq i\leq n}(X_i)$，$Z_n = \min_{1\leq i\leq n}(X_i)$ とおく．このとき，

$$E[Z_n] = \frac{1}{n}\mu \tag{13}$$

$$V[Z_n] = (\frac{1}{n})^2\sigma^2 \tag{14}$$

$$E[Y_n] = (\sum_{i=1}^{n}\frac{1}{i})\mu \tag{15}$$

$$V[Y_n] = (\sum_{i=1}^{n}(\frac{1}{i})^2)\sigma^2 \tag{16}$$

証明 まず Z_n の確率密度関数 $fz_n(x)$ を求める．X_i，$i = 1,2,...,n$ は独立より，恒等式 $\int_0^x fz_n(t)dt = 1 - (\int_x^\infty f(t)dt)^n$ が成り立つ．両辺を x で微分すると

$$fz_n(x) = (n\lambda)e^{-n\lambda x} \tag{17}$$

これは Z_n の確率分布が期待値 $1/(n\lambda)$ の指数分布であることを意味する．よって

$$E[Z_n] = \frac{1}{n\lambda} = \frac{1}{n}\mu \tag{18}$$

$$V[Z_n] = S\ (E[Z_n])^2 = \frac{\mu^2}{n^2} = (\frac{1}{n})^2\sigma^2 \tag{19}$$

Z_k，$k = 1,2,...n$ はそれぞれ相関関係のない独立な試行より，次の等式を得られる．

$$Y_n = Z_n + Z_{n-1} + ... + Z_1 = \sum_{i=1}^{n} Z_i \tag{20}$$

よって $E[Y_n]$，$V[Y_n]$ は

$$E[Y_n] = E[\sum_{i=1}^{n} Z_i] = \sum_{i=1}^{n} E[Z_i] = (\sum_{i=1}^{n}\frac{1}{i})\mu \tag{21}$$

$$V[Y_n] = V[\sum_{i=1}^{n} Z_i] = \sum_{i=1}^{n} V[Z_i] = (\sum_{i=1}^{n}(\frac{1}{i})^2)\sigma^2 \tag{22}$$

(fin)

よって補題 1 から次の定理が成立する．

定理 2 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，

$$\sigma^2 = \frac{\sum_{i=1}^{n}(\frac{1}{i})^2}{(\sum_{i=1}^{n}\frac{1}{i})^2}\sigma_0^2 \tag{23}$$

証明 X_i，$i = 1,2,...,n$ は独立同一分布である．それらの期待値を μ'，分散を σ'^2 とすると，補題 1 から

$$\mu = (\sum_{i=1}^{n}\frac{1}{i})\mu' \tag{24}$$

$$\sigma^2 = (\sum_{i=1}^{n}(\frac{1}{i})^2)\sigma'^2 \tag{25}$$

(fin)

$\mu = \mu_0$，$\mu^2 = \sigma^2$，$\mu'2 = \sigma'^2$ より式 (24)(25) から求めたい式 (23) を得ることができる．

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 $V_n, n = 1,2,...,v$ と記す時, $V_1 = (T||1), V_2 = (T||2), ..., V_v = (T||v)$ 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．

$$hash(V_a \mid\mid V_b) < D \tag{26}$$

(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 [MAK+16] 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が，マイニングにかかる時間の期待値や分散についての理論評価値は得られていない．

4. 既存研究のマイニングアルゴリズムの実装

4.1 マイニングの直列，並列連結の実装

4.1.1 直列連結の実験

本実験で用いるアルゴリズムは 3.1.1 で説明したものを

表 1　実装環境

メモリ	16.0GB
CPU	Intel Core i7-10750H
プログラミング言語	Python3

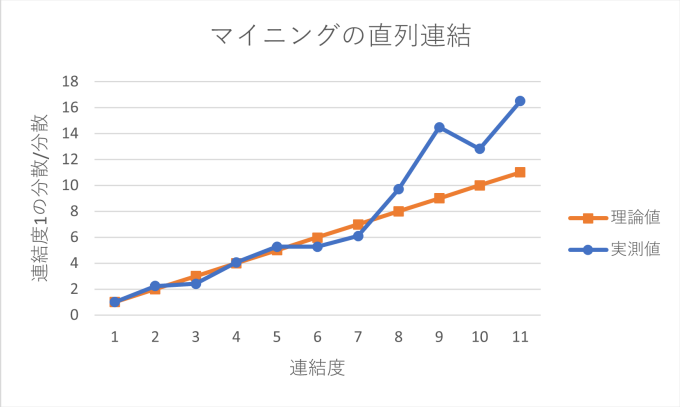

図 3　マイニングの直列連結による小分散化の効率

用いる．また，ブロック内の取引データは乱数で代用する．難易度は連結度 1 の時の難易度 D_1 とした時，連結度 n のときの難易度 D_n は nD_1 とすることでマイニングにかかる時間の期待値を全ての連結度で等しくする．連結度 n の時，難易度を nD_1 とすることの妥当性については以下で証明する．

証明 式 (3), (4) より探索にかかる時間の期待値 $E(X)$ は $E(X) = \frac{2^m}{Dm}$ となる．マイニングの直列連結を行う際，その探索時間は各探索にかかった時間の和となる．また，探索を n 回行う時，その探索にかかる時間の期待値 $E_n(X)$ は各探索は独立な試行なため，$E_n(X) = nE(X) = \frac{n2^m}{Dm}$ となる．よって，D を n 倍することで $E_n(X) = E(X)$ となる．

(fin)

本実験は試行回数 500 回で実験を行い，その平均と分散を測定した．実装の環境は表 1 で示す．また，ハッシュ関数は SHA-256，難易度 $D_1 = 10 * 2^{236}$ で行った．結果を表 2 に示す．

また，実機上でのマイニングの直列連結による小分散化の効率についてグラフで表すと図 3 のようになった．

グラフより，連結度 7 程度まではおよそ理論値と同じ程度の小分散化の効率であるが，連結度 8 以降で小分散化の効率が向上していることが分かる．

4.1.2　並列連結の実験

本実験で用いるアルゴリズムは 3.1.2 で説明したものを用いる．また，直列連結同様，ブロック内の取引データは乱数で代用する．難易度は連結度 1 の時の難易度 D_1 とした時，連結度 n のときの難易度 D_n は $(\sum_{k=1}^{n} \frac{1}{k})D_1$ とすることでマイニングにかかる時間の期待値を全ての連結度で等しくする．連結度 n の時，難易度を $(\sum_{k=1}^{n} \frac{1}{k})D_1$ とすることの妥当性については以下で証明する．

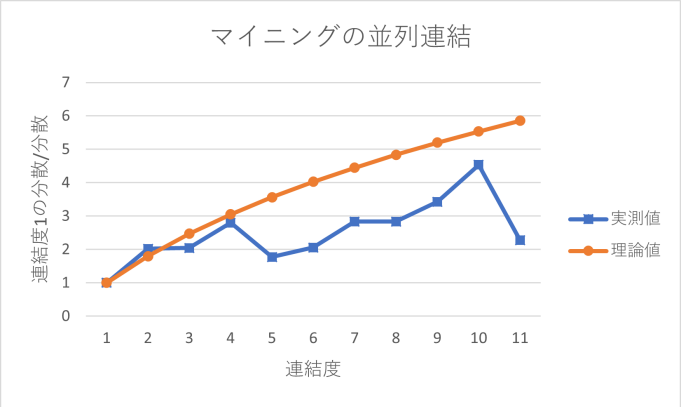

図 4　マイニングの並列連結による小分散化の効率

証明 並列連結したマイニングは，その探索時間は最もマイニングに時間がかかったものに依存する．式 (21) より，期待値 μ, 分散 σ^2 の指数分布に従う独立同一分布 X_i, $i = 1, 2, ..., n$ の $E[Y_n]$,$Y_n = \max_{1 \le i \le n}(X_i)$ は $E[Y_n] = (\sum_{i=1}^{n} \frac{1}{i})\mu$ となる．期待値 μ は式 (3), (4) より難易度 D に反比例することが分かる．よって $E[Y_n] = E[X]$ とするには D を $(\sum_{k=1}^{n} \frac{1}{k})$ 倍すればよい．

(fin)

本実験では直列連結同様，試行回数 500 回で平均と分散を測定した．また，実際に並列化を行うのは困難であるため，連結度 n の時，n 回探索を行い，それぞれの探索時間を記録し，最も時間がかかった物をそのマイニングの探索時間とすることで疑似的に並列化を実装した．

本実験に用いるハッシュ関数は SHA-256 である．また，連結度 1 の時の難易度 D_1 は $D_1 = 2^{240}$ とし，実験の環境は直列の時と同様である (表 1)．結果は表 3 のようになった．

実機上における並列連結による小分散化の効率をグラフで表すと図 4 のようになった．

表から，平均の探索時間が連結度に関わらずほぼ一定の値を示していることが分かり，疑似的ではあるがマイニングの並列化が実装できていることが分かる．一方で，グラフより小分散化の効率は理論値よりも低いことが分かる．

4.1.3　考察

直列連結，並列連結どちらも実機上での小分散化の効果が確認できた一方で，理論値とは少し異なる値となった．

直列連結において連結度 9 以降で分散が理論値よりも小さくなる理由についての考察を行う．連結度が大きくなるにつれてマイニングの難易度は低くなる．それにより，マイニングが易しくなりすぎて，結果として少ない回数の探索でマイニングが終わってしまう場合が多発してしまう．その結果，全体が探索時間が短い方に偏ってしまい分散が小さくなったと考えられる．分散に限らず，平均の探索時間も小さくなっていることから，最良の探索が多発してい

表 2 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]	22.20376848	20.83580554	21.23023177	20.23397924	19.88626502	20.23328228	20.54022365	18.41951885
分散 [s^2]	510.6362899	228.6171143	212.1368706	125.4509227	96.57119759	96.43822076	83.77330386	52.541870884
連結度 1 の分散/分散	1	2.23358733	2.407107678	4.070406807	5.287666537	5.2949576	6.095453639	9.7186545
連結度	9	10	11					
平均時間 [s]	17.76063147	18.57334545	18.30792985					
分散 [s^2]	35.23645263	39.84353996	30.9332094					
連結度 1 の分散/分散	14.49170537	12.81603719	16.5077048					

表 3 マイニングの並列連結

連結度	1	2	3	4	5	6	7	8
平均時間 [s]	14.38655972	11.73894467	14.72289616	13.80627773	15.36605849	13.73798362	14.1367311	14.1367311
分散 [s^2]	190.2746711	94.32656433	93.04171728	67.97824664	107.3733709	92.6486381	67.15131056	67.15131056
連結度 1 の分散/分散	1	2.017190729	2.045046853	2.799052351	1.772084359	2.053723347	2.833521334	2.833521334
理論値	1	1.8	2.469387755	3.048780488	3.562156007	4.024771838	4.446964726	4.836087995
連結度	9	10	11					
平均時間 [s]	14.82399097	14.73880934	14.106849821					
分散 [s^2]	55.48090025	41.97190433	83.46465812					
連結度 1 の分散/分散	3.429552697	4.533381893	2.279703474					
理論値	5.197577024	5.535574693	5.85331883					

ることが分かる．

次に並列連結について考察する．理論値よりも連結度 5 以降で小分散化の効率が悪い原因は，マイニングにかかる時間の期待値が小さいため，そのぶん多少の計算時間のブレが大きく分散に反映されてしまい，結果として分散が大きくなってしまったことが原因と考えられる．また，この実験は疑似的に実装した並列化で，実際には直列的に計算を行っている．そのため，段数が増えるほどに測定にかかる時間は長くなってしまう．それにより排熱等の原因で実験中の計算機の能力が変化し，分散が大きくなってしまったと考えられる．

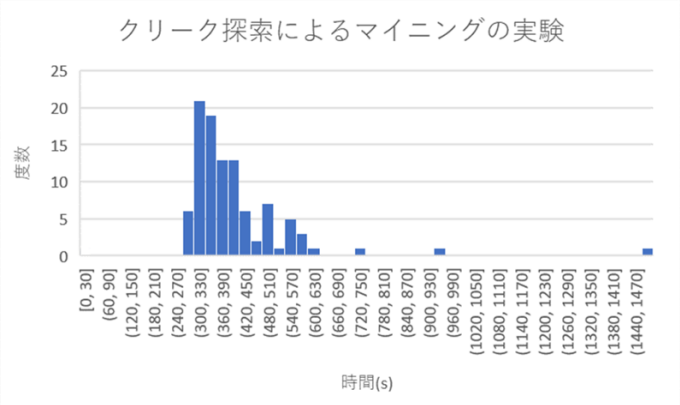

図 5 クリーク探索によるマイニングにかかった時間の頻度分布

4.2 クリーク探索によるマイニングの実装

4.2.1 実験

本実験で用いるマイニングのアルゴリズムは 3.2 のものを用いる．また，探索の方法は線形探索で行う．マイニングの条件は頂点数 $v = 2^{14}$，求めるクリークの大きさ $k = 4$，難易度 $D = 2^{248}$ で行う．また，実験の環境は 4.1.1 の実験と同様である (表 1)．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．探索時間についての度数分布を図 5 で示す．横軸がマイニングにかかった時間，縦軸がそのマイニングにかかった時間帯の個数を示す．

結果は平均 $\mu = 408.4487986(s)$，分散 $\sigma^2 = 22312.48285(s^2)$ となった．通常のマイニングの場合，分散は期待値 (=探索時間の平均) の 2 乗となるため，クリーク探索によるマイニングは通常のマイニングよりも分散が小さいことが確認できる．

4.2.2 考察

既存研究の課題であった，探索にかかる時間について考察する．クリーク探索によるマイニングはグラフの作成とグラフの探索という二つのパートに分けられる．グラフの作成にかかる時間は頂点数 v にのみ依存する．具体的には辺の作成に必要な計算量は v^2 に比例する．よってグラフ作成にかかる時間は $\alpha v^2(\alpha$ は比例定数) である．

次にグラフの探索にかかる時間について考える．頂点数 v のグラフから大きさ k のクリークを線形探索する際の必要な計算量は $O(v^k)$ である．また，難易度 D によってグラフ中に存在が期待される k クリークの数 num が決定される．頂点数が k となる，グラフ G の部分グラフ S は ${}_vC_k$ 個存在する．D によって決定される，頂点間が辺を持つ確率を p とすると，頂点数 k のグラフが完全グラフとなる確率は $p^{\frac{k(k-1)}{2}}$ より num は ${}_vC_k p^{\frac{k(k-1)}{2}}$ となる．num が小さすぎると探索が失敗してしまい，大きすぎると探索がすぐに終了してしまい，探索の意味をなさなくなってしまう．そのため，D は適切な値に設定する必要がある．グラフ G 中 k クリークが num 個存在するとき，探索に必要な計算量は $O(\frac{v^k}{num})$ である．D が適切な値の場合 num は小さい値のため探索に必要な計算量は $O(v^k)$ となる．

よってクリーク探索によるマイニングに必要な計算量は $\alpha v^2 + O(v^k)$ となる．

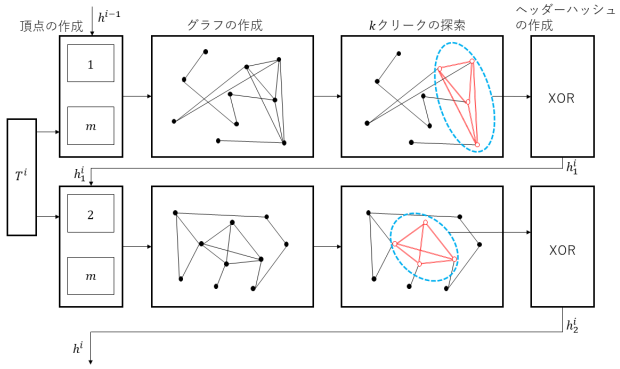

図 6 クリーク探索によるマイニングの直列連結 (頂点数 10，連結度 2，$k = 4$ の場合)

4.3 クリーク探索によるマイニングの直列接続の実装

4.3.1 アルゴリズム

基本的な考え方は 4.1.1 の直列連結と同じで，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．また，探索のアルゴリズムを図 6 に示す．

(1) 頂点を V_m，$m = 1, 2, ..., v_n$，連結した時の段数を y と記す時，$V_m = (T^i \mid\mid h \mid\mid y \mid\mid m)$ 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．

$$hash(V_a \mid\mid V_b) < D \tag{27}$$

(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 $V'_1, V'_2, ...V'_k$ とする．

(4) 次のヘッダーハッシュは $h = hash(XOR(V'_1, V'_2, ...V'_k))$ 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．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．$v_n = rv_1$ とし $\alpha n v_n^2 = \alpha v_1^2$ となるとき，$r = \frac{1}{\sqrt{n}}$ 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 の時の難易度を 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 4.2.2 より $num_1 = {}_{v_1}C_k p_1^{\frac{k(k-1)}{2}}$ である．同様にして num_n も同様にして導出される．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

$$\begin{aligned} num_n &= {}_{v_n}C_k p_n^{\frac{k(k-1)}{2}} \\ &= {}_{\frac{v_1}{\sqrt{n}}}C_k p_n^{\frac{k(k-1)}{2}} \\ &= {}_{\frac{v_1}{\sqrt{n}}}C_k (gp_1)^{\frac{k(k-1)}{2}} \end{aligned}$$

よって，求める g は

$$\begin{aligned} {}_{\frac{v_1}{\sqrt{n}}}C_k (gp_1)^{\frac{k(k-1)}{2}} &= {}_{v_1}C_k p_1^{\frac{k(k-1)}{2}} \\ (gp_1)^{\frac{k(k-1)}{2}} &= ({}_{v_1}C_k / {}_{\frac{v_1}{\sqrt{n}}}C_k) p_1^{\frac{k(k-1)}{2}} \\ g &= ({}_{v_1}C_k / {}_{\frac{v_1}{\sqrt{n}}}C_k)^{\frac{2}{k(k-1)}} \end{aligned} \tag{28}$$

となる．

4.3.2 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 4.1.1 の実験と同様である (表 1)．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 4 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．

4.3.3 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

表 4　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

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．

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，k クリークの期待される数が過度に多いもしくは少ないと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

本研究では 2 つの既存研究で提案されたマイニングを実装し，評価を行った．また，既存研究で提案された方法を組み合わせることが可能か評価を行った．

複数のマイニングの連結による小分散化は，連結の方法によって直列と並列の 2 種類があるが，どちらも実験により，実機上での小分散化の効果は確認できた．一方で理論評価とは異なる値となった．直列連結の実験では連結度 9 以降で理論値よりも分散が小さくなったが，これは難易度が小さくなりすぎてしまい，結果として最良の探索が数多く発生し，結果として分散が小さくなってしまったと考えられる．並列連結は連結度 5 以降で分散が理論値よりも大きくなったが，これは探索時間の期待値が小さかったため，少しの探索時間のブレが大きく反映されてしまい，結果として分散が大きくなってしまったのではないかと考えられる．また，実験中の CPU 性能変化によって分散が大きくなってしまったのではないかと考えられる．

クリーク探索によるマイニングの実験では，従来のナンスによる探索よりも分散小さいことが確認できた．また，探索に必要な計算量は $\alpha v^2 + O(v^k)$ であることを導出した．

最後に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．

参考文献

[AS20] 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.

[BKKT18] R. Bowden, H. P. Keeler, A. E. Krzesinski, and P. G. Taylor, “Block arrivals in the bitcoin blockchain.” CoRR, abs/1801.07447, 2018.

[BLB17] G. Bissias and B. N. Levine. Bobtail, “A proof-ofwork target that minimizes blockchain mining variance (draft).” CoRR, abs/1709.08750, 2017.

[BS21] ビットコイン相場 in 日本, http://ビットコイン相場.com/ (accessed 2021/02/01)

[BW21] bitcoinwiki. Confirmation. https://en.bitcoin.it/wiki/Confirmation (accessed 2021/02/01)

[HTC+16] 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

[MAK+16] 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.

[M16] 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

[N08] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” https://bitcoin.org/bitcoin.pdf, 2008.

[N21] NIST, “Secure hashing”, http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html (accessed 2021/01/01)

[R14] S. M. Ross. “Introduction to Probability Models.” Academic Press, 11th edition, 2014.

[RM11] Rosenfeld, Meni, “Analysis of bitcoin pooled mining reward systems,” arXiv preprint arXiv:1112.4980, 2011.

[S96] Bruce Schneier, “ Applied Cryptography,” John Wiley & Sons, 1996.