Vol.2019-MBL-93 No.25
Vol.2019-ITS-79 No.25
2019/11/22

ブロックチェーンのProof-of-workの計算資源を利用して最適化問題の解探索を行うプロトコル

柴田 直樹[1,a)]

概要：ビットコインなどのブロックチェーンの維持のために莫大な電力および計算資源が使われている．本稿では，この浪費されている計算資源および電力を任意のユーザ (クライアント) の登録した最適化問題の近似解の探索に利用できるようにする方法を提案する．クライアントは解候補の評価プログラムおよび料金をシステムにジョブとして登録する．提案手法を用いたブロックチェーンの維持のために，多数のノードがこのプログラムを実行する．最も良い解を見つけたノードに対し，クライアントがジョブとして登録した料金が支払われる．提案手法は，ビットコインにおける Proof-of-work に似た動作原理を採用しており，完全な分散処理が可能である．

1. はじめに

ビットコイン [9] の登場以来，ブロックチェーンを利用した暗号通貨が多数開発されてきた．ブロックチェーンとはリスト構造のデータであり，データを次々と新たに追加することができ，過去に登録されたデータの修正が困難となるように設計されている．暗号通貨における全ての取引の履歴はこのブロックチェーンに保存される．ビットコインでは，Proof-of-work(PoW) と呼ばれる仕組みにより正しい取引の結果を決めており，コインの二重使用等を防いでいる．PoW においては，ネットワークに参加するノードがある計算を行うことで，ノードの持つ計算量に応じた投票権を行使できるような多数決がとられる．なりすましの容易な計算機ネットワークにおいて，PoW は非常にロバストに動作する一方，PoW のために浪費される計算資源および電力が社会問題となっている．2018 年においては，このための電力はアイルランド全体で消費される電力 (3.1 ギガワット) に及んだ [12].

暗号通貨の維持のために浪費される多大な電力の問題を解決するため，PoW の代替として利用できる様々なプロトコルが考案されてきた．これらにより消費される電力を抑えられる一方，完全分散ではなかったり，プロトコル特有の問題を抱えているものが多い．PoW のセキュリティの高さと頑健性は他の代替プロトコルより優れており，このため PoW に基づく暗号通貨は依然として最もよく使われている．

1 奈良先端科学技術大学院大学
a) n-sibata@is.naist.jp

本稿では PoW を代替するブロックチェーンの仕組みを提案する．提案手法では，ブロックチェーンにおいて多数決をとるために必要な計算量を，任意の最適化問題のインスタンスの近似解を探索するために利用できる．提案手法によりブロックチェーンを最適化問題を解くためのバッチ処理システムとして利用できる．提案手法は，ジョブの登録，実行，見つかった最適解のクライアントへの提供などの仕組みなどを提供し，任意のユーザ (クライアント) が最適化問題のインスタンスをジョブとして登録することができる．

2. 関連研究

2.1 ビットコインと Proof-of-Work

ビットコイン [9] は，ロバストでセキュアな完全分散型の暗号通貨であり，取引の時間的順序を P2P 型分散タイムスタンプサーバに記録する．この時間的順序に対して，PoW による計算量的な証明が与えられる．ネットワークにおいては多数の IP アドレスを確保するのが容易であり，各 IP アドレスに対して投票権の与えられる多数決はうまく機能しない．PoW は，このような環境において計算量の多寡に応じて投票権の与えられる多数決を実現するのに

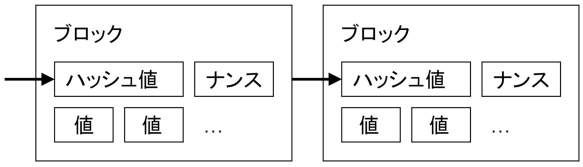

図 1: ビットコインにおける分散タイムスタンプサーバ [9]

ⓒ 2019 Information Processing Society of Japan

使用される．

ビットコインの分散タイムスタンプサーバ (図 1) は，データアイテムのブロックにタイムスタンプを刻印し，多数のブロックをリンクトリストとして保存する．このリンクトリストはブロックチェーンと呼ばれる．各ブロックは，そのブロックに含まれるデータアイテムと，一つ前のブロックのハッシュ値を含む．新しいブロックがチェーンに追加される毎に，新しいブロックのハッシュ値が計算され，ネットワーク上にブロードキャストされる．ビットコインの PoW では，このハッシュ値が決められた数のゼロビットで始まる数値を見つける．各ブロックには，ナンスと呼ばれる整数値を格納するエントリが用意されており，ブロック全体のハッシュ値が決められた数のゼロビットで始まるナンスを持つブロックのみが，有効なブロックとして受理される．ネットワークを維持するために，新しいブロックを追加することに成功したノードにインセンティブとして新しいコインが授与される．ブロックを追加しようとするノードのことをマイナーと呼ぶ．善意のマイナーは，そのノードの知る限り最も長いチェーンにブロックを追加しようとする．大半の CPU 資源が，善意のマイナーによりブロックの追加に使用される限り，正しい取引を記録したチェーンが最も速く伸びる．計算量に応じて投票権の与えられる多数決はこのようにして実現される．各ブロックが追加されるのにかかる時間をブロックタイムと呼ぶ．ハッシュ値のゼロビットの数は，ブロックタイムの期待値が 10 分になるように自動的に調整される．ビットコインは，下記の 3 つの性質を満たす．

- 完全分散であり，自立的に調整される
- ノードや，外部の組織，人などに一切依存しない
- ブロックに記録されたデータを変更することが困難
- なりすましに耐性がある
- 各マイナーが新たなブロックの追加に成功する確率は，そのノードの CPU 資源の量に比例する
- ブロックの正当性は，任意のノードが任意のタイミングで検証できる
- 任意のノードが事前登録無しにいつでも参加できる

しかしながら，ビットコインではマイナーが CPU 資源を PoW のために費やす必要があり，これはハッシュ値を繰り返し計算することである．これは，CPU 資源の浪費である．

2.2 Proof-of-Work の代替手法

ビットコインにより多大な電力が浪費される問題に対処するため，数々の Proof-of-work を代替するための手法が提案されてきた．

Proof-of-stake と Proof-of-burn [10] は，最近になって開発された暗号通貨に利用されている手法である．

Proof-of-stake は，最初に Peercoin [11] において実装された．この手法では，所持する通貨の量や，ノードの古さ，ランダム選択などを組み合わせて次のブロックを追加するノードを選ぶ．所持する通貨の量が多いノードほど頻繁にブロックを追加することができ，従って多くの新しいコインを得ることができる．このプロトコルでは，プロトコルにより選ばれた新しいブロックを追加するノードを信用する必要がある．実質的に，金持ちのノードによりネットワークがコントロールされる．Proof-of-activity [3] は，PoW と Proof-of-stake の組み合わせである．この手法では，マイナーは PoW と同様の方法で，チェーンに空のブロックヘッダーを追加する．このヘッダーには，Proof-of-stake と同様の方法で複数のノードが指名されており，これらのノード群が新しいブロックに署名する．新しいブロックが全てのノードにより署名されると，このブロックはブロックチェーンの一部として受理される．この手法の利点として，ネットワークをコントロールするために CPU 資源と所持する通貨の両方が必要になることが挙げられる．

Proof-of-burn [10] では，ある特別な宛先に送られたコインが回収不能になるような宛先を用意しておき，ノードが所持しているコインをこの宛先に送った場合に投票権が得られるようにする手法である．この手法では，投票権を持つノードを信用する必要がある．Proof-of-burn を利用して，コインをある暗号通貨から別の暗号通貨に移すことができる．これには，まずコインを宛先の暗号通貨に対応する宛先に送り，このコインをもとの暗号通貨において回収不能にする．次に宛先の暗号通貨において新たな取引を作り，もとの暗号通貨における取引を参照しつつ，相当する量のコインを発生させる．

Proof-of-useful-work [2]，Gridcoin [6]，Permacoin [8] では，多数決における CPU 資源をより意味のある用途に利用できる．

Proof-of-useful-work [2] では，多数決における CPU 資源を Orthogonal Vectors problems を解くために利用できる．Primecoin [7] では，CPU 資源を新たな素数を探すために利用できる．しかしながら，これらの問題を解く社会的な需要がどれほどあるのかは明らかではない．

Gridcoin [6] は，Proof-of-research と呼ばれる手法を利用する．この手法では，マイナーが Berkeley Open Infrastructure for Network Computing(BOINC) [1] で計算を行うことにより，マイナーに対しインセンティブを授与する．この手法では，多数決を取るための CPU 資源を非常に意味のある用途に利用することができるが，暗号通貨は BOINC システムに依存する必要がある．したがって，BOINC システムがダウンすると，Gridcoin は動作しなくなる．

Permacoin [8] においては，多数決をとるための計算機資

源を分散データストレージとして利用できる．新しいブロックを追加するために，ローカルのストレージ上のデータにランダムアクセスできる必要がある．マイニングのために，このデータが損傷していないことを Proof-of-retrievability と呼ばれる手法により暗号学的に証明する．

Proof-of-space [4] は，証明者と検証者の間のプロトコルであり，証明者がある大きなデータをローカルストレージに保存しておく．検証者は，証明者にデータを送るよう指示し，これにより証明者がデータを保存しているか確認する．このプロトコルは，確認に必要な計算量および通信量が小さくなるように設計されている．Proof-of-space を完全分散のブロックチェーンにおいて利用するためには，新たなブロックを追加するのに成功したノードを決めるための方法と，各ノードがノードの追加に成功する確率を知る方法が必要になる．ノードの追加に成功する確率は，各ノードが保存しているデータ量に比例する必要がある．これらの実用上の問題については，文献 [5] で考察されている．論文中で述べられているとおり，Proof-of-space およびそれを利用したブロックチェーンには，固有の弱点がある．まず，マイナーが同一のストレージ領域を利用して複数のチェーンを同時にマイニングできてしまう．また，同一のストレージ領域を利用してわずかに異なったブロックの追加を試みることができる．

3. 提案手法

本稿では，ブロックチェーンで利用できる完全分散型多数決プロトコルを提案する．提案プロトコルは，2.1 節で説明した長所を全て備えており，PoW のために浪費される計算資源を最適化問題の近似解の探索に利用できる．また，任意のノードが解候補の評価プログラムであるエバリュエータを含むジョブを登録することができる．提案プロトコルにおいては，整数値の代わりに，解候補とその評価値を連結したものをナンスとして用いる．有効なナンスを生成するためには，マイナーはエバリュエータを実行し，何らかの解候補を評価する必要がある．多数決を取る過程において多数のナンスが生成され，従って多数の解候補が評価される．ノード間の共謀を防止するため，提案手法では二つの異なった方法でマイニングノードにインセンティブを提供する．新しいブロックを追加することに成功したマイナーに対しては，PoW と同様に新たなコインを授与する．各ジョブに対し最も良い近似解を見つけたノードに対しては，ジョブを登録したノードが料金を支払う．クライアントはマイニングをすることなくジョブを登録することができ，マイナーはジョブを登録する必要はない．もし，ジョブが全く登録されていなければ，自動的に空のジョブが登録され，これにより提案手法は PoW と同等の働きをする．

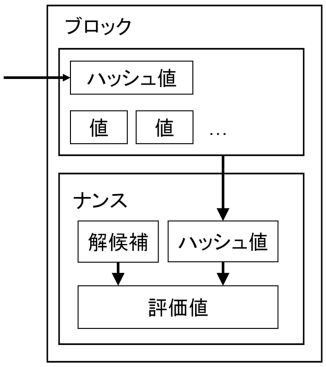

図 2: 提案手法の最小限の機能を利用するためのブロックチェーンのデータ構造

本稿では，解とは近似解のことを指し，最適化問題の解を得るとは，近似解を得ることを意味する．

3.1 キーとなるアイデア

まずエバリュエータ，ジョブ，クライアントについて説明する．エバリュエータは解探索を行う最適化問題のインスタンスを含み，与えられた解候補の評価値を決定的に計算するプログラムである．実行される環境によらず入力が同じであれば全く同じ出力を与える．ジョブは解探索の実行に必要なデータ全てであり，エバリュエータもジョブに含まれる．任意のノード (クライアント) は，ジョブをブロックチェーンに登録することで最適化問題の解探索をリクエストする．例えば，あるクライアントが巡回セールスマン問題のあるインスタンスの解探索を行いたい場合には，そのためのエバリュエータを実装し，これを含むジョブをブロックチェーンに登録する．この場合，都市の巡回順がエバリュエータの入力であり，経路長が出力となる．

提案手法においては，解候補とその評価値を連結したものをナンスとする．提案手法を利用したブロックチェーンが，登録された複数のジョブに含まれたエバリュエータから次のブロックを追加するためのナンスを生成するためのエバリュエータを選択する．PoW と同様に，次のブロックを追加するために，マイナーはナンスを含むブロック全体のハッシュ値が指定された個数のゼロビットで始まるようなナンスを探す．提案手法では，PoW と異なり，全てのナンスが有効ではない．提案手法においては，有効なナンスを生成するために指定されたエバリュエータを使用して解候補を評価する必要がある．この解候補と正しい評価値の組み合わせが有効なナンスである．次のブロックを追加するためには，マイナーは多数のナンスを生成する必要があり，そのために多数の解候補を評価する必要がある．マイナーが次のブロックを追加するためのナンスを見つけたということは，マイナーが多数の解候補を評価したことの確率的な証明となる．

マイナーが解候補の評価値を計算することの目的は二つ

ある．一つはナンスを含むブロック全体のハッシュ値が指定された個数のゼロビットで始まるような有効なナンスを探し，次のブロックを追加することであり，これによりマイナーは PoW と同様にコインを得ることができる．もう一つの目的は，良い評価値を持つ解を探すことである．提案手法においては，よい解を見つけるために多数の解候補を評価する必要があると仮定しており，全ノード中で最も良い解を見つけたノードに，クライアントの登録したジョブに含まれる料金が支払われる．この解探索を効率化するために，クライアントはサーチャーと呼ぶプログラムを実装し，ジョブに含める．サーチャーは遺伝アルゴリズムのような解探索アルゴリズムの実装であり，マイナーにより実行される．サーチャーは内部的にエバリュエータを何度も呼び出す．エバリュエータが呼び出されるごとに，対応するナンスを含むブロック全体のハッシュ値を計算し，それが指定された個数のゼロビットで始まるか調べ，そうである場合は新たなブロックをネットワークにブロードキャストする．サーチャーの実行は新たなブロックがいずれかのノードによってチェーンに追加されるまで継続する．

PoW においては，過去の計算結果を再利用することについて考慮する必要はない．提案手法においては，解候補の評価値の計算量がブロックのハッシュ値の計算量よりも大きいケースがあり，この場合に有効なナンスを異なったノード間で共有することで，一つの有効なナンスに対して複数のハッシュ値を計算する不正が可能になる．これを防止するため，提案手法では評価値とブロックに含まれるナンス以外のデータを関連付ける．エバリュエータがブロックに含まれるナンス以外のデータのハッシュ値を二つ目の入力とし，この値に応じて出力にわずかな誤差を付加する．各クライアントは，この誤差を付加するアルゴリズムを独自に考案して実装する必要がある．評価値に誤差を付加することで，エバリュエータをハッシュ関数の代わりに利用していることになるが，提案手法においては暗号学的ハッシュ関数としての性質はそれほど強く要求されない．エバリュエータが異なった入力に対し同じ値を出力する頻度はある程度低い必要があるが，ある程度の頻度で同じ値が出力されることは許される．

提案手法の最小限の機能を利用するためのブロックチェーンのデータ構造を図 2 に示す．このデータ構造により，単一の固定されたエバリュエータを利用できるが，最適化問題の良い近似解を見つけるための仕組みは持たない．以降，様々な機能を付け加える方法について述べていく．

3.2 ジョブの登録

提案手法においては，下記の性質が成り立つようにする．

- マイナーは良い解を探してそれをクライアントに提供するインセンティブがあること
- クライアントが既に良い解を知っているジョブを登録するインセンティブがないこと
- ジョブの登録・実行のためにコストが必要であること (この性質により価値のないジョブの登録を阻止する)

特に，クライアントが既に良い解を知っているジョブを登録するケースについて対処が必要である．このようなジョブを登録する動機として，以下が考えられる．

(1) 新たなブロックを追加する際に有利になる
(2) 新たなブロックを追加して得られるコインの利益の一部が得られる
(3) 良い解を見つけることで利益が得られる

良い解を見つけることに対して利益があるようにする一方で，不正な行いに対しては利益が得られないようにプロトコルを設計する必要がある．提案手法では，クライアントがジョブの実行に対して料金を支払うようにすることで，不正を防ぐ．上記項目 1 と 2 に関しては，3.1 章で述べたように，良い解を知っていても，新たなブロックを追加する上で有利にならない．項目 3 に関しては，良い解を見つけることに対する対価はクライアントが支払うため，クライアントが利益を得ることはできない．

また，良い解を見つけることに対する対価の授受は，特定のノードを信頼しなくとも自動的に行われるようにしたい．クライアントにより対価の支払いが確実に行われるようにするため，ジョブの受付時に対価がクライアントの口座から差し引かれるようにする．

探索の結果見つかった解がクライアントに安全に通知されるようにしたい．単純にあるノードが見つけた解をネットワークにブロードキャストすると，タイミングによってはこの解を別のノードが盗んでしまうことがあり得る．提案手法では，まず各ノードが見つけた解とノード ID を結合したもののハッシュ値をブロックチェーンに登録し，次のブロック時間に解を登録する．クライアントの払った対価は，最も良い解を見つけたノードに自動的に支払われる．

3.3 複数ジョブの同時実行

ジョブの実行料金を手頃にするため，各ブロック時間に複数のジョブが実行されるようにする．また，各ノードが最も良い解を見つけることで支払われる対価の期待値が，ノードの計算能力に比例するようにしたい．単純にブロック時間をジョブの実行料金に比例するように調整する方法だと，少額の実行料金でジョブを登録した場合にブロック時間が短くなってしまう．ブロック時間が短くなるとブロックチェーンがフォークする可能性が大きくなってしまう問題が生じる．これを避けるため，提案手法ではブロック時間を変化させずに，ブロックの間に複数のミニブロックを設け，各ミニブロックが一つのジョブに対応するようにする．

図 3 に示すように，ミニブロックは前のブロックのハッシュ値，マイナーの ID および対応するナンスからなる．マ

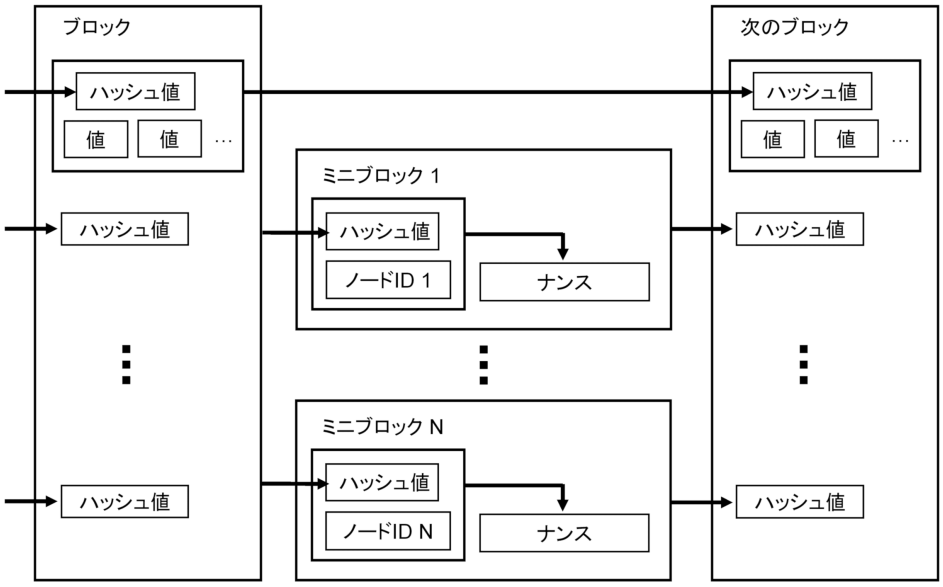

図 3: 提案手法における分散タイムスタンプサーバ

イナーはいずれかのミニブロックのハッシュ値が指定された個数のゼロビットで始まるような有効なナンスを探す．3.1 章で説明したとおり，有効なナンスは，ミニブロックに対応する問題のインスタンスの解候補とその評価値からなる．そのようなナンスが見つかるたびに，マイナーはナンスを含むミニブロックをブロードキャストする．ブロックの間の全てのミニブロックに対応するナンスが見つかると，ミニブロックの前にある最後のブロックがブロックチェーンに加わる．各ミニブロックに対応するナンスを見つけたノード全てに，新たに発行されたコインが支払われる．

メッセージ配送遅延により，ネットワーク中の異なったノードがある時間までに受け取ったメッセージの集合は異なる可能性がある．従って，異なったマイナーは，異なった内容のブロックをネットワークに追加するために計算を行っている可能性がある．これはつまり，これらの二つのマイナーが探しているナンスに対応するミニブロックに含まれる，最後のブロックのハッシュ値が異なる可能性があるということである．ミニブロックに起因するブロックチェーンのフォーク (ミニフォーク) を防止し，より大きな数のマイナーが同じブロックをチェーンに追加するための計算を行うようにするため，提案手法では各マイナーは可能な限り最も長いチェーンに対して新たなブロックを追加するために計算を行うようにする．ここで，あるチェーンが別のチェーンより多くのブロックを含むとき，チェーンの長さがより長いとし，またブロックの数が同じ場合，最後のブロックの後にあるミニブロックの数が多いチェーンの長さがより長いとする．マイナーがミニブロックを受信したときに，そのチェーンがより長い場合は，このチェーンに対して新たなブロックを追加するために計算を行うものとする．マイナーがミニブロックをブロードキャストする際には，最後のブロックとそれ以降のミニブロック全てをブロードキャストするものとする．

3.4 ジョブの実行環境

提案手法を実現するために，エバリュエータとサーチャーを実行するための環境が必要となる．任意のユーザがジョブを登録することができるため，ジョブに含まれるプログラムが適切に実装されてない場合に備える必要がある．まず，プログラムが終了しない場合に備え，定められたステップ数の実行の後にプログラムを強制終了する機能が必要になる．プログラムは決定的に動作する必要があるため，このステップ数のカウントは正確である必要がある．エバリュエータがクラッシュした場合，最も低い評価値が返されたものとして扱う．

上記をふまえて，ジョブの実行環境は下記の条件を全て満たす必要がある．

- 信頼できないコードを安全に実行できること
- エバリュエータを決定的に動作させられること
- アーキテクチャによらず，実行ステップ数を決定的に数える機能を持つこと
- 実行ステップ数を返すこと
- 指定されたステップの実行の後，実行を中断すること

上記の全ての条件を満たす実行環境を，インタプリタとして実装することは容易である．実行環境中で非決定的な動作をする API を提供しないことで，決定的な動作を実現できる．

3.5 提案手法の動作例

以下では，提案手法に基づくブロックチェーンの動作例について，ユーザの視点から説明する．

あるクライアントが，最適化すべき問題のインスタンスを持っていると仮定する．このクライアントは，問題のインスタンスに対するエバリュエータとサーチャを実装し，このインスタンスの解探索に支払う料金を決める．次に，エバリュエータ，サーチャ，料金を組み合わせてジョブを作り，ブロックチェーンに登録する．この料金は自動的に引き落とされる．このジョブのためにマイナーが使用するCPU 資源の期待値は，料金に比例する．マイナーはこの問題の解を探索するために CPU 資源を使う．見つかった解は，いずれブロックチェーンに登録され，クライアントは支払った料金と引き替えに見つかった解を得る．

ここではいつでも十分な数のマイナーがいると仮定する．最も長いチェーンにブロックを追加するために，解探索を行えるジョブは通常複数用意されている．もしジョブを登録するクライアントが居ない場合，通常の PoW にフォールバックし，ブロックのハッシュ値が決められた数のゼロビットで始まるようなナンスを見つけるジョブが自動的に追加される．マイナーはサーチャとエバリュエータをジョブから抽出し，サーチャを実行する．サーチャは，実行中に多数の解候補を生成し，エバリュエータを何度も呼び出すことでこれらの解候補を評価する．マイナーが良い解候補を見つけた場合，その解候補は保存しておく．マイナーが解候補を評価する際，有効なナンスを生成し，このナンスを含むブロック全体のハッシュ値を計算する．もし，このハッシュ値が決められた数のゼロビットで始まる場合，マイナーはこのナンスを含むミニブロックをネットワークにブロードキャストする．新しいミニブロックを追加するのに成功したマイナーは，PoW と同様にコインが授与される．マイナーが，ブロードキャストされた新しいミニブロックを受け取ると，そのマイナーはすぐにそのミニブロックを含む最も長いチェーンに対して新しいミニブロックを追加する処理にとりかかる．これにより，ミニフォークを防ぐことができる．全てのミニブロックが追加されるまでブロックタイムが続く．ブロックタイムの終了後，各マイナーは見つけた中で最も良い解をブロックチェーンに登録する．全てのマイナーの中で最も良い解を見つけたマイナーに対し，ジョブに添付された料金が支払われる．

次に，クライアントとマイナーが共謀することで本来より多くのコインを得ようとするケースについて考える．マイナーが新しいブロックを追加するためには，ブロック全体のハッシュ値が決められた数のゼロビットで始まるようなナンスを探す必要がある．これは，基本的にはランダムな解を選んでハッシュ値を計算することの繰り返しであり，ジョブの問題やそれに対する解とは関係がない．従って，新しいブロックを追加して得られるコインを得るためにクライアントとマイナーが共謀するメリットはない．一方，クライアントは，あるジョブを登録する前に解を計算しておき，その結果を共謀したマイナーに教えるケースが考えられる．しかし，良い解を見つけた報奨金はクライアントから共謀したマイナーに支払われるのであり，これはクライアントからマイナーに送金しているのと同じである．送金は，暗号通貨の基本操作で行うことができる．

4. 終わりに

本稿では，ブロックチェーンの維持のために浪費されている莫大な電力を最適化問題の近似解を探索するために利用できる多数決プロトコルを提案した．このプロトコルを利用し，任意のノード・ユーザが任意の最適化問題の近似解を探索するジョブを登録することができる．Proof-of-work に基づく暗号通貨は広く普及しており，これらの暗号通貨を代替するために提案プロトコルを非常に頑健でセキュアあり，また完全分散で動くよう設計した．提案プロコトルは，外部の組織などに依存しないという点で，既存の Gridcoin における多数決プロトコル等よりも優れている．

参考文献

[1] Anderson, D. P.: BOINC: A System for Public-Resource Computing and Storage, *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, GRID '04, Washington, DC, USA, IEEE Computer Society, pp. 4–10 (2004).

[2] Ball, M., Rosen, A., Sabin, M. and Vasudevan, P. N.: Proofs of Useful Work, *IACR Cryptology ePrint Archive*, Vol. 2017, p. 203 (2017).

[3] Bentov, I., Lee, C., Mizrahi, A. and Rosenfeld, M.: Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake, *ACM SIGMETRICS Performance Evaluation Review*, Vol. 42, No. 3, pp. 34–37 (2014).

[4] Dziembowski, S., Faust, S., Kolmogorov, V. and Pietrzak, K.: Proofs of Space, *Advances in Cryptology – CRYPTO 2015* (Gennaro, R. and Robshaw, M., eds.), Berlin, Heidelberg, Springer Berlin Heidelberg, pp. 585–605 (2015).

[5] Fuchsbauer, G.: Spacemint: A Cryptocurrency Based on Proofs of Space, *ERCIM News*, Vol. 2017 (2015).

[6] Halford, R.: Gridcoin: Crypto-currency using berkeley open infrastructure network computing grid as a proof of work (2014).

[7] King, S.: Primecoin: Cryptocurrency with prime number proof-of-work (2013).

[8] Miller, A., Juels, A., Shi, E., Parno, B. and Katz, J.: Permacoin: Repurposing bitcoin work for data preservation, *Security and Privacy (SP), 2014 IEEE Symposium on*, IEEE, pp. 475–490 (2014).

[9] Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008).

[10] P4Titan: Slimcoin A Peer-to-Peer Crypto-Currency with Proof-of-Burn (2014).

[11] Popper, N.: In Bitcoin's Orbit: Rival Virtual Currencies Vie for Acceptance (2013).

[12] Vries, A.: Bitcoin's Growing Energy Problem (2018).

ⓒ 2019 Information Processing Society of Japan