Vol.2019-CSEC-86 No.44
Vol.2019-SPT-34 No.44
2019/7/24

ブロックチェーンを用いた公正なオンラインゲームの構成手法

佐古　健太郎[1,a)]森　達哉[1,b)]松尾　真一郎[2,c)]

概要：確率的要素によって戦況が決定するゲームにおいて，その公平性が明らかになっているものは多くない．本論文では，ゲームの公平性を担保可能なシステムをブロックチェーンを用いて構成することを目的とする．ゲームとしてトランプを用いるブラックジャックを対象とし，対戦結果によって生じるプレイヤーとディーラーの取引を，スマートコントラクトとしてブロックチェーン上で実現する．さらにゲームの対戦履歴を別のブロックチェーンに記録することで，ゲームの公平性にかかる検証を透明性のある形で実現する．記録された手札の計測値に対して統計的仮説検定を行うことにより，ディーラーによる不正が検証可能であることを示す．

1　はじめに

勝敗に確率の要素が含まれる不完全情報ゲームにおいて，不正行為が不可能であることがそのゲームの核であり保証されている必要がある．不完全情報ゲームとは確率の要素が含まれるトランプやサイコロを用いるものや，麻雀などが挙げられる．このようなゲームは，サイコロの出目や次にどのカードが出るか分からない部分が，これらのゲームの重要な要素になる．確率的要素の存在によりどのプレイヤーにも勝利の可能性が与えられる．その中でどういった戦略を取るかを熟考することが，これらのゲームの楽しむポイントであり魅力となる．しかし，次のカードがどれであるかを知っている場合など，本来はそのゲームの参加者が知ることができない情報を把握している場合，有利にゲームを行うことができる．ユーザ側からの視点でそのゲームが公正であるかどうかは不透明であるものが多いのが現状であり，公正でないゲームも存在する．ゲームの公平性を保つためには，プレイヤーとディーラーの両者が不正行為を起こすことが原理的にできない仕組みを構成することが重要である．

本研究では，確率的要素を含むゲームの例としてトランプカードを用いるブラックジャックゲームを対象とし[1]，ブロックチェーン技術を用いることで不正行為が不可能なシステムを構成する方法を示す．このゲームを選んだ理由の一つは，勝敗結果に基づくチップの取引があるゲームはブロックチェーンの取引と相性が良いことが挙げられる．また，ブロックのデータ構造を取引データに加えて対戦結果のデータを記録することで，不正の実施が困難になる効果が期待される．具体的にはブロックチェーンの改ざん不可能性を利用し，ゲームの結果に人為的な不正行為が入り込んでいないかを統計的仮説検定により検証することが可能である．

ブラックジャックゲームにおいて不正行為が不可能であることを保証する上で，次に配られるカードが何か分からないという点も担保する必要がある．ディーラーはカードの並びを知っている場合，ゲームを進める上で有利になるが，そのような事態を防ぐことを目的とする．そのために，「カッティング」と呼ばれるトランプの山の何枚目から開始するかをプレイヤーが決定するシステムを導入する．これによりトランプの並び方について，シャッフルを行うディーラーだけでなくプレイヤーも関与することが可能となる．また，ディーラーがシャッフルした後にその並びを変更できないようトランプデータと，ある乱数を生成し組み合わせたハッシュ値を対戦する前に公表する．ゲーム終了後にトランプデータと乱数を公開することで，その並び通りにカードが配られたか検証が行うことができる．これらのデータをブロックチェーンに記載することで，検証が容易に可能である．

さらに，勝敗によって発生するゲーム内通貨の移動における不正行為を防ぐためのプロトコルとして，スマートコントラクトを利用する．スマートコントラクトとは，ブロックチェーン上で予め決められた取引の実行を自動的に行うコンピュータプロトコルである．第三者機関を介さず実行できる

[1] 早稲田大学基幹理工学研究科
[2] Computer Science Department, Georgetown University
[a)] ksbowler@nsl.cs.waseda.ac.jp
[b)] mori@nsl.cs.waseda.ac.jp
[c)] Shinichiro.Matsuo@georgetown.edu
[1] ブラックジャックの簡単なルール説明を本論文の付録として掲載している．

トラストレスな部分が最大のメリットである．概念としては暗号通貨が誕生される前から存在しているが，イーサリアムという暗号通貨で初めてスマートコントラクトが実装された．本実験では，プレイヤーがディーラーに勝利した場合にはディーラーからプレイヤーに，負けた場合はプレイヤーからディーラーへゲーム内通貨が送金される取引の契約を結ぶ．これにより，負けた際に故意に支払いを拒否することが両者ともに不可能となる．

ブロックチェーンに記録された対戦履歴データを分析し，ディーラーの手札に不正がないかを検証する．手札がランダムではなく，一定の規則性があるカードを利用するとディーラーにとって有利な偏りをもたらすことが可能である．不正の検証は，ディーラーの手札が 17 以上 21 以下となる Stand 状態，および 22 以上となる Bust 状態の割合を対象とし，割合の実験値と理論値に有意差があるかを，カイ二乗検定を用いて検証する．検証を行った結果，乱数度の高いシャッフルでのカードセットを使用すると，手札がランダムになっている（不正をしていない）帰無仮説が支持された．反対にカードの並びに規則性のあるカードセットを使用すると，手札がランダムになっている（不正をしていない）帰無仮説が棄却された．すなわち，本研究の手法によれば，ディーラーが何らかの規則性を持つカードを使用して不正を行おうとした場合，ブロックチェーン上に記録されたデータからその不正を検出することが可能である．

本論文の構成は以下の通りである．2 章では，本研究の背景技術としてブロックチェーンとスマートコントラクトを簡単に説明する．3 章では，本研究の実験方法を示す．4 章では，3 章で用いた方法での実験結果を示す．5 章は議論として本研究の成約や今後の課題を議論する．6 章では，本研究と関連した研究を紹介し，アプローチの違いや利用した点について述べる．7 章は本研究のまとめである．

2 背景技術

ブロックチェーンは暗号通貨で利用されている分散型台帳である．2008 年にサトシ・ナカモトが発表したビットコインという暗号通貨に関する論文 [1] で提案された．いくつかの取引を一つのブロックにまとめ，そのブロックのデータと過去のブロックのハッシュ値を次のブロックに渡すことで鎖型の台帳となる．そのためブロック内の取引データを改ざんした場合，そのブロックのハッシュ値も変わるためすぐに検知されるようになっている．また，ブロックチェーンは Peer-to-Peer という全てのノードが 1 対 1 で通信する分散型ネットワークで運用しているため，管理者のいない非中央集権的なシステムである．つまり，誰でもブロックを作成することや今までのブロックの中身である取引などを閲覧できるような構造である．

ブロックチェーンにおいてある取引が成立すると，まだブロックに格納されていない取引が集まるトランザクションプールにデータが送られる．ブロックを作成する人は，有効であると思った取引をこのプールから選ぶ．ビットコインに利用されているブロックチェーンの場合，コインにシリアルナンバーが付いている．故に取引データを遡ると，特定のコインがどのようなユーザが持っているかなどの追跡が可能となる．取引の際に使用されたコインが本人のものか検証できることにより，不正取引を選択しないようにすることができる．そして，取引データをいくつか集めたものが一つのブロックに格納される．

ブロックは一つ前のブロックハッシュ値や取引データ，作成日時を示すタイムスタンプや作成するために必要なナンスと呼ばれるビット列が格納されている．ブロック作成にはビットコインの場合 Proof-of-Work (PoW) 方式が採用されている．PoW とは，そのブロックのデータハッシュ値を特定の値 (Target 値) 以下になるようなナンスを見つけるハッシュパズルを解く方式である．このナンスを見つけた者にブロック作成権利が与えられるのである．このパズルは Target 値を変えることで難易度を調整できる．現在の PoW 方式では 10 分でパズルを解けるように設定されている．

ブロック作成者は，既に作られたブロックの中からすべて有効な取引であると思われる最新のブロックのデータハッシュ値を Previous-Hash として作成することで，ブロック同士が繋がっていく．不正な取引データのあるブロックを見つけた場合，そのブロックの直前ブロックと繋げるとそこで分岐 (フォーク) が起こる．フォークが起こると，ブロック作成者は自身が正しいと思うブロックと繋げる．これが続くと分岐したチェーンの長さに差が生じる．短い方は有効であると思われていないとされ，これらの取引データはプールに戻される．そのブロック以降に六個のブロックが繋がっていることで，有効であると判断される．

イーサリアムという暗号通貨は，スマートコントラクトというプロトコルを初めて実装した．スマートコントラクトの概念は，ブロックチェーンが誕生する前よりの 1994 年に Nick Szabo という暗号学者によって最初に提唱された [2]．スマートコントラクトは予め決められた取引の実行を自動的に，検証や交渉をスムーズに行うように意図されている．契約相手からの虚偽情報の申請に対するリスクが小さくなるため相手を信用する必要がなくなる．ブロックチェーン上で行うことにより，契約内容に対する透明性も向上する．他にも，第三者機関を介さず契約内容が実行されるのでコストの低下が見込める．暗号通貨におけるスマートコントラクトは，P2P において改ざん不可能なプログラムのことを指す．ブロックチェーンを利用することにより，改ざん不可能なアプリケーションの作成を可能にしている．

スマートコントラクトは暗号通貨の応用や，それ以外の分野でも活用されることが期待されている．スマートコント

ラクトで取引を行うことで，売り手や代理店などが，購入希望者と分散型ネットワークでつながることによりマッチングが迅速になるといったメリットがある．しかし，売買成立に時間のかかる点や利用料金が高い点など，課題も多い．このような活用事例を参考にし，ゲームの公平性をユーザへ担保することが，本研究での狙いである．

3 ブラックジャックゲームの実装と検証方法

本章では，ブラックジャックゲームの実装について述べる．また，本論文の目的である公平性が保たれているかどうかの検証方法の説明や判断基準の設定を行う．

3.1 ブラックジャックゲームのプログラム

ブラックジャックゲームのルールは付録に記載する．本研究では，Python を使用しプログラムコードを実装した．作成したコードを実行すると，まずプレイヤーに 1,000 コインが配布される．対戦初回時にディーラーがトランプをシャッフルする．このシャッフルの方法に関しては 3.3 節で記述する．シャッフルしたカードのデータと乱数を生成しその乱数を組み合わせたハッシュ値を公開する．そのハッシュ値をプレイヤーは確認し，カッティングを行う．カッティングを行うことで，公開されたハッシュ値に対して合意したと見做すこととする．そしてディーラーとの対戦が開始される．対戦データは新たに生成されるたびに，ブロック作成プログラムへ送信される．このプログラムは 3.2 節において詳しく説明する．また，対戦を行うたびにトランプは使用されていく．最後にディーラーがシャッフルしてからカードを 40 枚以上使用した場合再びシャッフルし，そしてハッシュ値の公開を行いカッティングをプレイヤーが行う．プレイヤーが全てのコインを失う，もしくは対戦が 20 回行われたらこのゲームは終了しこれを 1 プレイとする．ゲーム終了後及びゲーム中にシャッフルする前に，それまでゲームで使用したトランプの並び方と生成した乱数を公開する．

3.2 ブロックを作成するプログラム

前節で述べたように，ブロック作成プログラムはプレイヤーとディーラーの対戦データを受け取る．プレイヤーの勝敗と賭け金を参照し，ゲームのルールに基づいたスマートコントラクトによるプレイヤーとディーラー間でのコインの送金を行う．このプログラムは，一つのトランザクションにプレイヤーとディーラーの手札と二者間のコインの送金データを格納する．ディーラーが，1 プレイで発生したトランザクション全てを一つのブロックに入れたものを格納する．作成されたブロック群はブロックチェーンによりすべて閲覧可能となる．

3.3 シャッフルの方法

乱数度の高いシャッフルを完全シャッフルと名付け Python の random ライブラリの sample 関数を採用して行う．規則性のあるカードのデータセットは一例として，一部 random ライブラリを利用した方法と，そうでないものの二種類のアプローチでカードセットを作成した．52 枚あるトランプのカードを 0 ～ 51 の整数をカード番号として割り当てる．数字を意味するナンバーはカード番号を 13 で割った時の剰余，絵柄を意味するスートはカード番号を 13 で割った時の商と対応する．これらを表 1にまとめる．次節以降の規則性のあるカードセットでは，検証では有意水準を下回らずにディーラーにとって有利になるカードセットの候補を提案する．

表 1 トランプカードとカード番号の対応

13 で割った剰余	ナンバー	13 で割った商	スート
0	A	0	spade
1	2	1	heart
2	3	2	diamond
3	4	3	club
4	5		
5	6		
6	7		
7	8		
8	9		
9	10		
10	J		
11	Q		
12	K		

3.3.1 規則性のあるカードセット　その 1

本節では，一部 random ライブラリを利用したカードセットの作成について説明する．そのアルゴリズムを Algorithm 1に示す．トランプは 4 種類の絵柄と 13 種類の記号によって構成されている．そこで，13 種類の記号を random ライブラリを用いてシャッフルを行い，その並びを 4 種類の絵柄全てで同じものにするカードセットを採用する．Algorithm 1より記号番号 num とスート番号 suit をシャッフルする．suit に 13 を掛け num を足す操作を 52 回繰り返し，これをカードセットとする．13 枚毎にスートの違う同じナンバーが現れるので，この方法でカードセットを作成していることを把握している場合，12 枚使用すると次のカードのナンバーが把握できる．Algorithm 1のように $N, S = 13, 4$ としたカードセットを Part-of-13, 同様に $N, S = 26, 2$ としたものを Part-of-26 と名付ける．

Algorithm 1 Part-of-N のアルゴリズム (N=13 のとき)

$N, S = 13, 4$
$num, suit = [i\ for\ i\ in range(N)], [i\ for\ i\ in\ range(S)]$
$num = random.sample(num, len(num))$
$suit = random.sample(suit, len(suit))$
$card = []$
$for\ i\ in\ suit:$
　　$for\ j\ in\ num:$
　　$card.append(13 * i + j)$
return $card$

3.3.2 規則性のあるカードセット　その２

本節では，random ライブラリを使用しないカードセットの作成について説明する．そのアルゴリズムを Algorithm 2に示す．はじめにカード番号 0 ～ 51 の中からランダムに一つ選ぶ．これがカードセットの先頭となる．次に 52 と互いに素である整数 R を選ぶ．そして次に記述する動作を 51 回繰り返す．カードセットの最後尾のカード番号から R だけ足し，その数字と対応するカードをカードセットの最後尾とする．対応しない場合，52 で割った余りと対応するカードを採用する．整数 R は 52 と互いに素であるため，カードセット内でカードが被るといったことが回避される．この方法でカードセットを作成していることを把握している場合，最初にカードを使用した時点で次のカードのナンバーが把握できる．R は [1,3,5,7,9,11,-1,-3,-5,-7,-9,-11] の 12 種類で検証を行う．最もディーラーの勝率が高かった時の R の値を，次節で説明する Part-of-Arith-L に採用する．このカードセットの作成方法を Arith-R (R が 1 の時は Arith-1) と呼ぶこととする．

Algorithm 2 Arith-R のアルゴリズム (R=1 のとき)

```
R = 1
x = random.randrange(52)
card = []
for i in range(52) :
        card.append((x + R * i) mod 52)
return card
```

3.3.3 規則性のあるカードセット　その３

本節では，完全シャッフルのように見せかけるカードセットの作成について説明する．そのアルゴリズムを Algorithm 3に示す．まず完全シャッフルのカードセットを用意する．次に前節で採用するとした Arith-R を長さ L だけ作成する．長さ L の Arith-R に含まれる要素を完全シャッフルのカードセットから削除する．その後二つに分割し，分割した間に Arith-R を挿入する．このようにすることで長さ $52 - L$ だけ完全シャッフルであり，挿入した Arith-R を使用している時はディーラーの勝率が高くなるということとなる．このカードセットの作成方法を Part-of-Arith-L (L には Arith-R の長さが入る) と呼ぶこととする．

Algorithm 3 Part-of-Arith-L のアルゴリズム (L=10 のとき)

```
L = 10
percard = perfect()
arithcard = arith(L)
for i in arithcard :
        percard.remove(i)
x = random.randrange(52)
s1 = percard[: x]
s2 = percard[x :]
card = s1 + arithcard + s2
return card
```

表 2　Verify-num の理論値 (確率)

カードの枚数	Stand 状態	Bust 状態
2 枚	0.348	—
3 枚	0.271	0.175
4 枚	0.082	0.095
5 枚	0.011	0.017
6 枚以上		0.002

表 3　Verify-up の理論値 (確率)

アップカード	Stand 状態	Bust 状態
A	0.857	0.143
2	0.642	0.358
3	0.619	0.381
4	0.593	0.407
5	0.566	0.434
6	0.578	0.422
7	0.739	0.261
8	0.760	0.240
9	0.765	0.235
K	0.785	0.215

3.4 ブロックチェーンに記載した結果についての検証方法

検証母数について本実験では，最大 20 個の取引を 1 つのブロックに格納し，ブロックは 50 個作成する．この操作は 100 回行う．

3.4.1 ディーラーの手札に関する検証

ブロックチェーンに記載したディーラーの手札の結果が 21 以下である Stand 状態か 22 以上の Bust 状態かどうか，二種類のアプローチで検証を行う．一つはディーラーは手札が何枚目で Stand 状態か Bust 状態になったのかという点 (Verify-num と呼ぶ)．次にプレイヤーは公開されるディーラーの手札の一枚目 (アップカード) を参考にして賭け金を決めるので，アップカードに対して場合分けを行い Stand 状態と Bust 状態のどちらになったのかという点 (Verify-up と呼ぶ) を検証する．これらの検証には，カイ二乗検定を用いる．カイ二乗検定とは，帰無仮説が正しければ検定統計量が漸近的にカイ二乗分布に従うような検定法であり，理論値と実測値の有意差を検定するのに適している．有意水準は 0.05 とする．実測値と理論値の間に有意差はないという帰無仮説を立て検証を行う．理論値は表 2, 3にまとめる．求めた P 値が有意水準より大きい場合，帰無仮説を棄却せずに正しいものとする．そうでない場合，実測値と理論値に有意差があるとして帰無仮説を棄却し，ディーラーが自身が有利になるように不正行為を行ったと見做す．

3.4.2 プレイヤーの勝率に関する検証

本節では，プレイヤーの Bet する額によってプレイヤーの勝率が変化しているか検証について説明する．プレイヤー自身とディーラーの最初のカードを参考にして Bet する Normal-Player と，最初のカードが何であろうと最大額，最小額 Bet する Aggressive-Player，Defensive-Player の三種類のプレイヤーによる対戦を行う．勝率とプレイヤーの収支額

```
{
    txid : 00310842
    time : 2019-06-12 14:16:53.735687
    addr
    {
        src_addr : Pakfja2q45elau78werwalh
        dst_addr : Dj89q33j90g3562gfdqwert24
    }
    price : 400
    hand_data
    {
        player_hand : 37, 7, 15
        dealer_hand : 26, 10
    }
}
```

図 1 トランザクションの一例

表 4 プレイヤーの勝率 (1block あたり)

プレイヤーの種類	収支	勝率
Normal	747.5	0.434
Aggresive	-60.0	0.425
Defensive	-15.3	0.430

を比較する．

4 ブラックジャックゲームの検証結果

4.1 スマートコントラクトを利用したゲーム内でのコインの移動

3.2 節で述べた，ブロック作成プログラムのスマートコントラクトが正しく実行されているか検証する．ブロックに格納されているトランザクションの一例を図 1に示す．ブラックジャックゲームにおいて，ディーラーに勝った場合は賭け金がディーラーからプレイヤーへ，ディーラーに負けた場合はプレイヤーからディーラーに支払うというコントラクトが履行されているのが分かる．

4.2 トランプのシャッフルについて

3.1 節で述べたように，ゲームを開始したときやカードを 40 枚以上使った場合トランプをシャッフルする．その後トランプの並び方のデータと生成した乱数を組み合わせたハッシュ値を公開する．ゲームを開始したとき以外にシャッフルしたとき，またゲーム終了時にトランプの並びと乱数を公開する．先に公開されたハッシュ値とゲーム終了後公開されたトランプと乱数のハッシュ値は一致している場合，ディーラーがシャッフルした後にトランプの並びを変更していないことが分かる．

4.3 プレイヤーの勝率に関する結果

プレイヤーの勝率と 1block あたりの収支額について表 4にまとめる．勝率に関して，Aggressive-Player 及び Defensive-player が Normal-Player と有意差がないかカイ二乗検定を行ったところ有意水準を下回る結果は得られなかった．

表 5 完全シャッフルの結果から求めた P 値

シャッフルの種類	Verify-num		Verify-up	
	P 値	N	P 値	N
完全シャッフル	0.934	0	0.278	0

N: 有意水準を下回った回数 (回/100 回)

表 6 Part-of-N の結果から求めた P 値

シャッフルの種類	Verify-num		Verify-up	
	P 値	N	P 値	N
Part-of-13	0.012	98	0.001	100
Part-of-26	0.733	0	0.057	65

N: 有意水準を下回った回数 (回/100 回)

4.4 ディーラーの手札に関する結果

完全シャッフルを用いた結果

求めた P 値を表 5にまとめる．Verify-num, Verify-up ともに有意水準を下回る結果は得られなかったので，不正なシャッフルではないと判断できる．

Part-of-N を用いた結果

求めた P 値とプレイヤーの収支額及び勝率を表 6, 7にまとめる．Part-of-13 では verify-num, verify-up ともに有意水準を下回る結果となり，Part-of-26 では有意水準を下回る結果は得られない場合もあった．しかしながら，両者ともプレイヤーの収支額が完全シャッフルの時より大きいので，ディーラーにとって効果的なカードセットとはならなかった．

Arith-R を用いた結果

Arith-R ではランダム性がないのでディーラーの手札の結果は有意水準を下回るのは明らかである．プレイヤーの収支額及び勝率を表 8にまとめる．プレイヤーの収支額が最も低い，つまりディーラーの収支額が最も高くなるカードセットは Arith-(-5) であった．

Part-of-Arith-L を用いた結果

前節の結果より，ディーラーの勝率が最も高かった R の値-5 を採用する．求めた P 値とプレイヤーの収支額及び勝率を表 9, 10にまとめる．長さ L が 7 以下の場合は求めた P 値が有意水準を下回る結果は得られなかったが，プレイヤーの収支は収入の方が大きい結果となった．プレイヤーの収支において支出の方が大きくなる時の長さ L は 20 以上の場合であった．このことから，検証結果が不正と判断されずにディーラーが儲けることは，このカードセットの作成方法では実現できなかった．

5 議論

本章では，本研究の制約事項ならびに将来の課題について議論する．

表 7 Part-of-N 利用時の勝率 (1block あたり)

シャッフルの種類	収支	勝敗確率		
		勝利	引分	敗北
Part-of-13	1171.3	0.445	0.072	0.483
Part-of-26	765.2	0.435	0.083	0.482

表 8　Arith-R 利用時の勝率 (1block あたり)

シャッフルの種類	収支	勝敗確率		
		勝利	引分	敗北
Arith-1	-635.9	0.419	0.158	0.423
Arith-3	3061.8	0.457	0.137	0.406
Arith-5	15916.4	0.794	0.103	0.103
Arith-7	-902.2	0.173	0.061	0.766
Arith-9	-623.0	0.224	0.082	0.694
Arith-11	14232.5	0.769	0.115	0.116
Arith-(-1)	15353.4	0.953	0.023	0.024
Arith-(-3)	-191.6	0.356	0.210	0.434
Arith-(-5)	-1000	0.212	0.060	0.728
Arith-(-7)	1279.9	0.503	0.0	0.497
Arith-(-9)	1344.8	0.371	0.212	0.417
Arith-(-11)	122.7	0.651	0.0	0.349

表 9　Part-of-Arith-L の結果から求めた P 値

シャッフルの種類	Verify-num		Verify-up	
	P 値	N	P 値	N
Part-of-Arith-6	0.755	0	0.581	0
Part-of-Arith-7	0.360	0	0.402	0
Part-of-Arith-8	0.058	57	0.141	4
Part-of-Arith-9	0.041	69	0.013	94
Part-of-Arith-10	0.022	100	1.23E-4	100
Part-of-Arith-20	2.56E-09	100	1.84E-11	100

N: 有意水準を下回った回数 (回/100 回)

表 10　Part-of-Arith-L 利用時の勝率 (1block あたり)

シャッフルの種類	収支	勝敗確率		
		勝利	引分	敗北
Part-of-Arith-6	665.4	0.428	0.084	0.488
Part-of-Arith-7	512.0	0.416	0.088	0.496
Part-of-Arith-8	576.1	0.422	0.092	0.486
Part-of-Arith-9	531.8	0.417	0.085	0.498
Part-of-Arith-10	514.9	0.423	0.091	0.486
Part-of-Arith-20	-117.0	0.406	0.092	0.502

5.1　本研究の制約事項

本研究では，ブラックジャックゲームの結果をブロックチェーンに載せ，それをもとにカイ二乗検定を行い不正行為が行われていないか検証した．分散型ネットワークを構築できなかったため，ディーラーのみがブロックを作成できるパーミッションドブロックチェーンを，本実験では採用した．また，このゲームで用いるトランプをシャッフルするのはディーラーであり，Python の random ライブラリの sample 関数を用いている．

5.2　このゲームを非中央集権的にするために

ブロックチェーンはビットコインのネットワークが非中央集権システムで運営するための欠かせない要素である．このゲームを非中央集権システムで運営するにはどのようにすればいいか考える．

管理者のいない非中央集権システムを実現するには，ディーラーを特定の人物に任命するのではなく誰でも行えるようなシステムを導入する．また，カジノで行われているようなディーラーが複数のプレイヤーと対戦するマルチプレイシステムも導入する．これにより，どの参加者もあるディーラーと対戦することが可能となる．それに伴い，エントリーフォームやブロックの構成について見直す必要がある．本実験では，プレイヤーに 1000 コイン配布される．コインが無くなるか 50 回対戦を行うまでゲームが続く．マルチプレイシステムの場合，賭場を開きたい者が主催者となり参加者を受け付ける．その際の終了条件を以下のように 2 通り考える．

- あるプレイヤーの所持コインが無くなるか，決められた値以上になった場合に終了．ディーラーは参加者同士が順番に行うこととする．途中参加や退室は不可．
- 主催者がディーラー，参加者はプレイヤーとなる．途中参加や退室は可能．主催者はいつでも賭場を閉じることができる．

いずれの場合も，賭場が開かれてからあるプレイヤーが終了条件を満たした時までに発生した取引を一つのブロックにまとめる，という本実験と似たブロック作成方法も可能である．それに対し，現在使用されている暗号通貨同様，取引プールを設け取引が生成される度にプールに移動されるシステムも構築できる．

次に，このゲームで用いるブロックチェーンをパーミッションレスにすることで非中央集権的に運用することを考える．パーミッションレスブロックチェーンを利用しているビットコインでは Proof-of-Work (PoW) 方式でブロック作成権利を与えている．このゲームの運用に PoW 方式を採用したとする．そこで考えられる課題は，ブロック取り込み時間である．ブロックを一個作成するのに約 10 分かかり，そのブロックにある取引が有効であるとみなされるには 6 個のブ

ロックが繋がる必要があり，約 1 時間かかる．これにより，ゲームで勝った場合でもそのコインを使えるのが約 1 時間以上先になってしまう．この時間をできるだけ短くすることが，今後の課題である．

5.3 トランプのシャッフルについて

このゲームを公正なものにするために，トランプをシャッフルした後の並び方は「乱数度」の高いものである必要がある．本実験では，Python の random ライブラリの sample 関数を用いてシャッフルし，その後にカッティングを行うことで参加者に公正さを担保している．これはディーラーが自身に有利になるような偏りのあるトランプの並び方でゲームを行ったとしても，トランプの山札の何枚目から使用するのか分からないので，ディーラーが不利になる可能性もあるとしている．言い換えれば，山札のどの位置から開始しても片方にという有利になる並び方は無いとしている．本実験で扱ったカードセット作成方法ではディーラーが有利になるようなものはなかった．しかし，ディーラーが有利になるカードセットの作成方法が存在した場合，この攻撃を検知する方法が必要となる．

このような攻撃の防止策として，ディーラーが行うシャッフルの結果について検定する手法を考える．その検定方法の一例として，文献 [3] 内で紹介されている NIST 乱数検定が挙げられる．NIST 乱数検定とは，ビット列にした際に 0 または 1 の連続した個数の偏りを調べる連の検定，などといった計 15 種類の検定がある米国国立標準技術研究所で開発された乱数検定ツールである．この 15 種類の検定に合格した数が多いほど乱数度が高いとしている．この検定を挙げた理由として，Fan らが発表した Proof-of-Useful-Work 方式が提案されている論文 [4] に活用できると考えたからである．彼らの論文では，従来の PoW 方式のハッシュパズルを解く際の計算力に注目している．ハッシュパズルを解けなかった時の計算力は無駄になってしまう．そこでハッシュパズルではなく，ある分野にとっては有用性のあるパズルを採用し，そのパズルを解いた者にブロックの作成権利を与える方法を提案している．本研究において有用性のあるパズルは，トランプのシャッフルである．本実験では，ディーラーがシャッフルすることになっているが，第三者が行うこととする．そのシャッフル結果が NIST 乱数検定の合格数が基準値を満たした場合，その者にブロック作成権利を与える．Fan らの論文では，有用なパズルはいくつか提案されているが，いずれも解の個数が少ないことや難易度の設定ができないという問題点が指摘されている．しかしながら，トランプの並びは $52! \sim 8.066 \times 10^{67}$ 通りあることや，NIST 乱数検定の合格検定数を調整することでこれらの課題が解決する．検定に合格したカードセットをゲームで用いることで，ディーラーは自身に有利になるようにはできず，より公正なゲームが提供されるであろう．

6 関連研究

本章は，ブロックチェーンやスマートコントラクト，及び対戦型ゲームに関する関連研究を示す．

6.1 ブロックチェーンに関わる研究

前章でも触れたが，Fan らによると，PoW 方式はハッシュパズルが解けなかった際の計算力が無駄であると述べられている [4]．ハッシュパズルの計算結果は，他の分野でも必要とされていない．そこで REM という新しいブロックチェーンのフレームワークを提案した．ブロック作成には PoUW 方式という他の分野に活かされるパズルを用意し，解けなかった場合でもそれらを必要としている企業に提供することで無駄を省くことに寄与されるとしている．機械学習で用いる Support Vector Machine やタンパク質の折り畳み構造に関する問題が，Useful-Work の例として挙げられた．これらの Useful-Work を採用することで，ハッシュパズルを行うことなく無駄を省くことができると結論付けられた．

Patrick らは，ブロックチェーンを用いてインターネット上で行うことができる投票プロトコルを実装した [5]．これらは理事会などの選挙に適しており，Ethereum のスマートコントラクトとして記載されている．このプロトコルは分散型であるため，第三者機関が投票の計算を行うことはない．現在は取引手数料のガスの設定により，最大参加者数は 50 人と推奨している．将来的には，全国規模の選挙が行うことができるように研究が進められている．大きな課題としては，スマートコントラクトに使用している Solidity 言語は符号なしの整数しか扱うことができず，プライバシーを担保するために楕円曲線上でプロトコルを実行することとしている．しかしながら，Solidity 言語は楕円曲線暗号をサポートしておらず，それらのライブラリを利用するとブロックチェーンに投票に関するスマートコントラクトが格納できないという問題が発生した．

6.2 ブラックジャックゲームに関わる研究

ブラックジャックゲームに関して, 本実験では手札の枚数や 1 枚目のカードからディーラーの手札について分析した．Wakin らはマルコフ連鎖を用いたブラックジャックゲームの理論分析を行っている [6]．彼らの手法を適用することにより，より詳細分析が可能となることが期待できる．Jeremy らはオンラインゲームにおける参加者が複数いる場合の公平性や起こりうる問題について研究している [7]．その中で参加者間の地理的距離により起こる伝播時間のラグがゲーム性を悪化させる可能性があるという問題点を指摘している．Griffiths らはオンラインゲームが人気となった社会的要因を研究している [8]．人気であるオンラインゲームのユーザへの

調査が少なかったことから Everquest というオンラインゲームのユーザの情報を調べ，その結果一部ユーザが睡眠などを犠牲にゲームを長時間行っているという点から，ユーザに長時間プレイすることを抑制するべきであると主張した．

7 まとめ

本研究では，ブロックチェーンを用い透明性のある形で公平性を実現するブラックジャックゲームの構成法を示し，その有効性を評価した．中心的なアイディアは，ブラックジャックゲームにおいて不正の元となるトランプのランダム性を誰もが証明できるようにすることである．このために，ディーラーおよびユーザに配布されたカードの履歴を含むゲームの全結果をトランザクションとしてブロックに記録する．このブロックはブロックチェーンの構成であるため改ざんは不可能である．対戦終了後，ブロックに記録されたデータに対して統計的検定を適用することにより，不正な操作の有無を検証することができる．また，ゲーム内で用いるコインの移動については，勝敗の結果に応じて契約を履行するスマートコントラクトを用いることができる．

今後の課題は実世界におけるブロックチェーンサービスとして実現すること，およびそのようなシステムを持続可能なものにするために必要となる課題や要素技術を明らかにすることである．

A ブラックジャックゲーム

ここでは本研究において実装したブラックジャックゲームのルール及びゲームの流れをまとめる．

このゲームではジョーカーを含めないトランプのカード 52 枚を使用する．エース (以下，A と表す) は 1 もしくは 11 として数えることができ，その他の絵札と 10 (絵札はジャック，クイーン，キングの 3 種類あるが総称して K とする) は全て 10 として数える．手札の強さはカードの英数字の合計で決まり，スートは影響しない．ブラックジャックと呼ばれる手札の枚数が 2 枚で 21 となる組み合わせ (A と K) が最も強い．それ以降は手札の枚数に関係なく 21 が強い手札となり，その次は 20 の手札,19 の手札と続く．最も弱い手札は 22 を超えるものであり Bust と呼ばれる．

次にこのゲームの進行手順について説明する．まずはじめにプレイヤーとディーラーに一枚目のカードが配られ，ディーラーの一枚目 (以下，アップカードと呼ぶ) は公開される．アップカードをもとにプレイヤーは賭け金を設定する．この動作が完了後，双方に二枚目のカードが配られるが，ディーラーのカードは公開されない．

そして，プレイヤーは Hit,Stand のどちらかの行動を選択する．Hit は 1 枚カードを引き，Bust 状態でない限り繰り返し選択できる．Stand は手札の決定を行う．

その後ディーラーの手札が決まる．ディーラーは手札の強さが 17 を越えるまで必ず Hit を選択しなければならず，越えた場合は必ず Stand を選択する．両者の手札が確定すると，両者の手札の強さにより勝敗が決まる．プレイヤーが勝利した場合，ディーラーからプレイヤーに賭け金の 2 倍が支払われる．さらに，プレイヤーがブラックジャックだった場合は賭け金の 2.5 倍が支払われる．ディーラーが勝利した場合は，賭け金は全てディーラーに没収される．これが一回の対戦の流れである．

References

[1] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system*. 2009. URL: http://www.bitcoin.org/bitcoin.pdf.

[2] Nick Szabo. *The Idea of Smart Contracts*. 1997. URL: %5Curl%7Bhttps://nakamotoinstitute.org/the-idea-of-smart-contracts/%7D (visited on 06/20/2019).

[3] J. K. M. Sadique Uz Zaman and Ranjan Ghosh. “A Review Study of NIST Statistical Test Suite: Development of an indigenous Computer Package”. In: *CoRR* abs/1208.5740 (2012). arXiv: 1208.5740.

[4] Fan Zhang et al. “REM: Resource-Efficient Mining for Blockchains”. In: *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. 2017, pp. 1427–1444.

[5] Patrick McCorry, Siamak F. Shahandashti, and Feng Hao. “A Smart Contract for Boardroom Voting with Maximum Voter Privacy”. In: *Proceedings of the 21st Financial Cryptography and Data Security*. 2017, pp. 357–375.

[6] Michael B. Wakin and Christopher J. Rozell. *mcbj*. URL: https://scholarship.rice.edu/handle/1911/20431.

[7] Jean-Marc Brun, Farzad Safaei, and Paul Boustead. “Fairness and playability in online multiplayer games”. In: *CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006.* 2 (2006), pp. 1199–1203.

[8] Mark D. Griffiths, Mark N. O. Davies, and Darren Chappell. “Demographic Factors and Playing Variables in Online Computer Gaming”. In: *Cyberpsychology behavior : the impact of the Internet, multimedia and virtual reality on behavior and society* 7 4 (2004), pp. 479–87.