The Game Among Bribers in a Smart Contract System

Lin Chen(⊠), Lei Xu(⊠), Zhimin Gao(⊠), Nolan Shah(⊠), Ton Chanh Le(⊠),
Yang Lu(⊠), and Weidong Shi(⊠)

Department of Computer Science, University of Houston, Houston, TX 77054, USA
chenlin198662@gmail.com, xuleimath@gmail.com, mtion@hotmail.com,
nolanshah212@gmail.com, letonchanh@gmail.com, ylu17@central.uh.edu,
wshi3@uh.edu

Abstract. Blockchain has been used to build various applications, and the introduction of smart contracts further extends its impacts. Most of existing works consider the positive usage of smart contracts but ignore the other side of it: smart contracts can be used in a destructive way, particularly, they can be utilized to carry out bribery. The hardness of tracing a briber in a blockchain system may even motivate bribers. Furthermore, an adversary can utilize bribery smart contracts to influence the execution results of other smart contracts in the same system. To better understand this threat, we propose a formal framework to analyze bribery in the smart contract system using game theory. We give a full characterization on how the bribery budget of a briber may influence the execution of a smart contract if the briber tries to manipulate its execution result by bribing users in the system.

1 Introduction

Various applications are developed on top of blockchain technology [31–33]. However, most of these works assume that the blockchain is a perfect system, e.g., all records stored in the system are correct, and ignore the complexity of the way that the decentralized system achieves consensus. For purely cryptocurrency systems, both static model [24] and game theory model [21,27] have been used to analyze their security features. The introduction of the smart contract makes the situation trickier while extending the applicability of blockchain technology. A smart contract can involve multiple users/participants and have a high value stake. Thus, it has the potential to be more critical than mining in pure cryptocurrency systems (e.g., Bitcoin), in which only a fixed reward is paid to successful miners. The amount of cryptocurrency involved in a contract may be many times and significantly higher than the cost of running the contract itself. Therefore, users involved in a smart contract have the incentive to push through a certain outcome. In particular, they may achieve such a goal through bribery, i.e., offering cryptocurrencies to other users in the system. Interestingly, bribery itself can also be carried out using smart contracts. A recent work discussed this

A. Zohar et al. (Eds.): FC 2018 Workshops, LNCS 10958, pp. 294–307, 2019.
https://doi.org/10.1007/978-3-662-58820-8_20

concept and proposed a straightforward framework to implement bribery on blockchain [20] where the briber offers incentive to the bribee through a smart contract.

Bribery is a serious problem as it may help to compromise the fundamental assumption of smart contract execution model based on consensus or majority accepted outcome. Note that a user is honest in mining does not necessarily means that he/she will remain honest when offered with monetary reward in making decisions. Their honesty is even more questionable when taking into consideration the unlinkability of users' identities to real persons, and the fact that there is no punishment for reporting a wrong execution result in many smart contract systems like Ethereum. Therefore, it is important to investigate the problem whether a briber can succeed in manipulating a smart contract execution result.

It is remarkable that the execution of a smart contract can be cast as an election and we may leverage the research on elections to understand the bribery problem in a smart contract system. Specifically, we can view users in the system as voters, and all the possible outcome of a smart contract as candidates. Each voter (user) will vote for a specific candidate (outcome), and a briber will bribe voters to alter the election result (smart contract execution outcome). We remark that by using an election model we are actually simplifying the consensus protocol implemented in a smart contract system without considering, e.g., the Byzantine behavior of a user who tries to send different messages to different other users. However, note that such kind of behaviors typically influence users who are following the protocol. In this paper, we take a game theoretical point of view by treating all users as rational people who are trying to maximize their own profit, and will therefore stick to the choice which is the best for their own interest regardless of the choices of others. Hence, it is reasonable to adopt an election model.

There exist a series of papers focusing on the bribery problem in an election model, see, .e.g., [1–3,9,10,12,15,18,19,23,26,34,35]. Specifically, researchers have studied extensively the computational complexity of the bribery problem and show that in many settings it is NP-hard for a briber to decide which subset of voters should he/she bribe (see, e.g., [17] for a nice survey). Such hardness results can also be viewed as a way to discourage people from carrying out bribery, if computational complexity is of concern.

Classical hardness results for the election model apply readily to the bribery problem in a blockchain system by viewing a smart contract execution as an election. However, we observe that a briber needs to overcome more difficulties if he/she really wants to carry out bribery in a blockchain system. Indeed, a briber not only needs to handle the computational complexity in determining a suitable subset of voters to be bribed, but he/she may also have to compete with other bribers in the system. Note that in most real-world elections, bribery is carried out in secrecy. A person, once offered a bribe, may either take it and cast his/her vote shortly afterwards, or reject it. The "incorrectness" in the nature of bribery prevents it from becoming a free market where bribers "sell" their bribes to people. However, things change completely in a blockchain system.

As we will provide details in the following section, a briber is able to establish a smart contract with a bribee. The smart contract will be executed by users in the system and a transfer of cryptocurrencies will be carried out once the contract is fulfilled, i.e., once the bribee casts his/her vote accordingly. In this case, a bribee may establish smart contracts with multiple bribers and strategically chooses the best. The unlinkability from a user identity in a blockchain system to a real person behind and the fact that a smart contract may not necessarily be executed immediately allow a user to easily involve in multiple smart contracts. Such a situation poses a severe task to bribers and they end up in competing with each other unavoidably without even knowing their opponents. Under such a competition in a blockchain system, how difficult it is for a specific briber to win? This paper is targeting at such a problem.

Our Contributions. There are two major contributions of this paper. First, we study the bribery problem in a blockchain system from a game theoretical point of view and model it as a smart contract bribery game. This is a first step towards a better understanding of the bribery problem in a blockchain system; and may also be of separate interest to the studies of elections. In this model, every briber is a player and has a bribing budget which can be allocated to voters. Every voter has a bribing price p_j. The voter will only take smart contracts that offer a price no less than p_j. Once he/she is offered multiple smart contracts, he/she will fulfill the one with the highest price (ties are broken arbitrarily). The strategy set of a briber is all possible allocations of the budget to voters.

Second, given a smart contract bribery game, we consider its Nash equilibrium. We are particularly interested in the following problem: if a briber is very lucky, can he/she compromise the smart contract execution by getting the majority of votes through a small amount of budget? The answer is no. We show that, a briber cannot win more than 50% of the votes unless he/she controls more than 20% of the total bribing budgets in *any* Nash equilibrium. That is, even if the briber is lucky enough to end up in a Nash equilibrium that is the best for him/her, he/she still needs to have a significantly large bribery budget, more than 20% of the sum of all the budgets, in order to manipulate the execution result arbitrarily.

Organization of the Paper. The remainder of the paper is organized as follows: In Sect. 2 we give a short review of smart contract and describe the problem we address in this paper. In Sect. 3 we present our main result by studying the Nash equilibria of the smart contract bribery game. In Sect. 4 we give further discussion on our results. Section 5 discusses related work, and we conclude the paper in Sect. 6.

2 Preliminaries and Problem Statement

Smart Contract. We begin by defining smart contracts. The definition provided by Szabo in 1997 is [28]:

Definition 1. *A smart contract is a set of promises, specified in a digital form, including protocols within which the parties perform on these promises.*

A blockchain system, equipped with smart contracts, is a powerful tool that allows users to build various applications on top. In particular, a voting system can be implemented on blockchain. We first briefly describe the election model for a voting system studied in the literature.

Election Model. In an election, given are a set of n candidates $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ and a set of m voters $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$. Each voter V_j has a preference list of candidates, which is essentially a permutation of candidates, denoted as τ_j. The preference of v_j is denoted by $(C_{\tau_j(1)}, C_{\tau_j(2)}, \ldots, C_{\tau_j(m)})$, meaning that v_j prefers candidate $C_{\tau_j(z)}$ to $C_{\tau_j(z+1)}$, where $z = 1, 2, \ldots, m-1$.

An *election rule* is implemented, which takes as input the set of candidates and voters together with their preference lists, and outputs a set of winner(s). There are various election rules studied in the literature. In this paper, we focus on one of the most fundamental rules called *plurality*. In plurality, every voter votes for exactly one candidate which is on top of his/her preference list. The candidate(s) with the highest number of votes then become the winner(s).

The abstract election model is general enough to incorporate a lot of real-world elections as well as other applications that involve voting in their execution. In particular, it is very much relevant to a blockchain system since almost all decisions made in such a system, e.g., block construction and verification [30], are based on the consensus among users. A consensus protocol can be modeled as an election where every user votes for his/her decisions, and eventually one decision is elected by the system.

Bribery in an Election. In recent years, the problem of bribery in an election has received much attention in the literature [1–3, 9, 10, 12, 15, 18, 19, 23, 26, 34, 35]. On a high level, bribery in an election is defined as a way to manipulate the election by giving monetary reward to voters so as to change their preference lists. Researchers have proposed different bribery models. In this paper, we focus on the *constructive bribery model*, that is, the briber tries to make one specific candidate become the winner by bribing a subset of voters. This is particularly the case when bribery happens in a blockchain based system – a briber tries to make the system to reach a specific consensus.

Bribery through Smart Contract. In most real-world elections, briberies are carried out in secrecy. It is, however, interesting that briberies can be carried out "publicly" using smart contracts. Roughly speaking, the briber and the user to be bribed (or bribee) can create a special smart contract that claims a transfer of cryptocurrency upon the condition that the user votes for a specific candidate (decision). Users of the system will execute this smart contract. Once the condition is satisfied, the transfer of the cryptocurrency will be enforced by the system. The anonymous feature of a blockchain system, especially the unlinkability of a user account from the real person behind, allows part of the information of the bribery to be transparent, e.g., the transfer of cryptocurrency from one account to another, while preserves the privacy of the persons involved.

The concept of carrying out bribery through smart contract naturally follows from many real-world contracts that are created to facilitate bribery. However, there is a lack of a systematic study on the creation and execution of such smart contracts for bribery, and its influence on the whole blockchain system. A very recent paper by Kothapalli and Cordi [20] gave the first detailed study on the creation and execution of the smart contracts for bribery and presented pseudo codes. Briefly speaking, the whole bribery procedure, via smart contracts, is divided into three phases: (i) Propose stage. The briber creates a briber contract indicates the incentive that the bribee will receive upon fulfilling the bribe and/or the punishment if the bribee fails to fulfill that. The contract is submitted to the blockchain. (ii) Commit stage. A bribee who decides to participate creates a claim on the blockchain. (iii) Verify stage. After a time period, if the bribe condition is reached, the bribee can get the incentive. Otherwise, the bribee pays the penalty.

Given their research [20], it becomes crucial to understand the impact of such smart contracts for bribery to the whole blockchain system. Although we may leverage the research on bribery in elections, the problem of bribery via smart contracts has its own unique characteristics. Particularly, when there are multiple bribers in the system, the bribee is free to participate in any smart contract for bribery and he/she can thus strategically maximize his/her own profit. In this paper, we try to understand the behavior of bribers and bribees through game theory. Towards this, we first introduce some basic concepts.

Definition 2 ([25]). *A normal form game Γ consists of:*

- A finite set N of players (agents).
- A nonempty set Q_i of strategies available for each player $i \in N$.
- A preference relation $\preceq_i$ on $Q = \times_{j \in N} Q_j$ for each player i.

We restrict our attention to normal form games in this paper. For simplicity, when we say a game, we mean a normal form game. We consider Nash equilibrium in this paper. A Nash equilibrium is a solution concept of a game involving two or more players in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by unilaterally changing his/her own strategy [25].

Taking a game theoretical point of view, we are able to model the bribery problem in a blockchain system with multiple bribers as follows.

Smart Contract Bribery Game. We first describe the basic setting for the smart contract bribery game. Given are a set of n candidates $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$, a set of m voters $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$ and a set of k bribers $\mathcal{B} = \{B_1, B_2, \ldots, B_k\}$. Each briber B_h has a budget b_h for bribing and prefers one specific candidate. Each voter v_j has a preference list τ_j and a bribing price p_j. Each briber can sign a smart contract with a voter, which offers a certain amount of reward in cryptocurrency if the voter changes his/her preference list and votes for the candidate preferred by the briber. A voter v_j can sign a smart contract with every briber and then do the following:

- he/she will discard all smart contracts that offer a price lower than p_j;
- if there are multiple smart contracts offering a price larger than p_j, he/she will pick the one with the highest price and vote for the candidate preferred by this briber;
- ties are broken arbitrarily, i.e., the voter will randomly choose one smart contract if there are several smart contracts offering the same highest price (larger than or equal to p_j).

Note that if all the smart contracts are offering a price lower than p_j, the voter will vote honestly.

Bribers and the candidates need not be the same, however, as each briber prefers a distinct candidate, we assume for simplicity that the briber is *the same as* the candidate he/she prefers, i.e., $\mathcal{B}$ is a subset of the candidates. By re-indexing the candidates, we may assume without loss of generality that $B_h = C_h$ for $1 \leq h \leq k$, i.e., the first k candidates are trying to bribe voters.

Let the bribers be players in the game. The strategy set of a briber is the set of possible smart contracts he/she can make with voters, i.e., every strategy of a briber b_h is an allocation of the budget b_h among all the voters, which can be represented as an m-vector $(b_h^1, b_h^2, \ldots, b_h^m)$ where b_h^j is the price the briber offers to voter V_j and $\sum_j b_h^j \leq b_h$. The goal of each briber, as a player, is to maximize the (expected) number of votes he/she received.

Nash Equilibrium in Smart Contract Bribery Game. A pure Nash equilibrium for the smart contract bribery game, if it exists, is a solution where every briber B_h specifies some strategy $(b_h^1, b_h^2, \ldots, b_h^m)$ such that if B_h changes his/her strategy unilaterally to some $(\bar{b}_h^1, \bar{b}_h^2, \ldots, \bar{b}_h^m)$, the expected number of votes he/she can get will not increase.

3 The Smart Contract Bribery Game

If there is only one briber, then obviously the briber is able to increase the number of his/her votes if his/her bribing budget is at least as large as the cheapest bribing price of some voter who votes for another candidate. When there are multiple bribers, things become much more complicated. Considering an arbitrary briber, say, B_1, can he/she really benefit from bribery in the presence of other bribers? Of course the answer is no if there exists another briber with an infinite or sufficiently larger budget, who is able to bribe every voter with a price larger than b_1 and B_1 will get no votes at all. If, however, B_1 is more powerful, say $b_1 \geq b_i$ for every $2 \leq i \leq k$, is it possible for B_1 to get additional votes? Unfortunately, this may not necessarily be the case and is highly dependent on the strategies of other bribers. In this section, we focus on Nash equilibrium in the smart contract bribery game. We consider the following problem: In a Nash equilibrium, how many votes can B_1 get when competing against bribers who are weaker than him/her? Furthermore, can B_1 get more votes than he/she gets in the absence of bribery in the system?

Theorem 1. *There may exist a pure Nash equilibrium for the smart contract bribery game where the briber B_1 can get at most $\lfloor 1/\epsilon \rfloor$ votes even if $b_1 \geq 1/\epsilon \cdot b_i$ for every $2 \leq i \leq k$, where $\epsilon \in (0, 1)$ is an arbitrary number.*

We remark that a pure Nash equilibrium may not always exist.

Proof. Consider the following smart contract bribery game in which there are $m = k - 1 + \lfloor 1/\epsilon \rfloor$ voters and exactly k candidates (i.e., $\mathcal{C} = \mathcal{B}$). Let $p_j = 1$ for $1 \leq j \leq k-1$, $p_j = 1/\epsilon$ for $k \leq j \leq m$. Let b_1 be an arbitrary integer larger than $1/\epsilon$, and $b_i = \epsilon b_1$ for every $2 \leq i \leq k$.

Consider the following feasible solution: each briber B_i, $2 \leq i \leq k$, bribes V_{i-1} at the price of ϵb_1. The briber B_1 then bribes V_k to V_m, each at the price of ϵb_1.

It is easy to verify that B_1 gets $\lfloor 1/\epsilon \rfloor$ votes. It suffices to argue that the feasible solution above is a Nash equilibrium. First, we claim that every briber B_i, $2 \leq i \leq k$, will not deviate from the current solution. Note that if B_i aims to bribe some other voter instead of V_{i-1}, then he/she needs to pay at least ϵb_1, for otherwise that voter will simply ignore his/her offer. Therefore, B_i has to take away all the money ϵb_1 from V_{i-1} and bribes some V_h for $h \neq i-1$. However, since V_h already receives ϵb_1 amount of money from another briber, thus in expectation B_i only gets 1/2 votes, which is worse than the current solution. Hence, B_i will not unilaterally change his/her strategy. Next, we claim that B_1 will not deviate from the current solution. Note that currently B_1 gets one vote at the cost of ϵb_1. If he/she aims at getting votes from any V_h, $1 \leq h \leq k-1$, he/she has two choices. Either he/she pays the price of ϵb_1 and gets 1/2 votes in expectation, or he/she pays a price strictly larger than ϵb_1 and gets one vote. In both cases, B_1 will lose one vote from the set of voters in $\{V_h : k \leq h \leq m\}$ and get at most one vote from the set of voters $\{V_h : 1 \leq h \leq k-1\}$. □

Note that k is a parameter that can be significantly larger than $1/\epsilon$, Theorem 1 thus implies that a briber may only get a small number of votes even if the bribing budget of any other briber is at most ϵ fraction of his/her budget.

It is worth mentioning that in the proof of Theorem 1 we do not specify which candidate does a voter votes in the absence of bribery. We may assume that without bribery V_h, $1 \leq h \leq k-1$, all vote for B_1, while V_h, $k \leq h \leq m$, all vote for B_2. Therefore, B_1 actually loses an arbitrary amount of votes when bribery happens. More precisely, we have the following corollary.

Corollary 1. *In a smart contract bribery game, a briber may lose an arbitrary number of votes even if he/she is only competing against other bribers whose budget is significantly smaller.*

Theorem 1 implies that the worst Nash equilibrium for a briber can be very bad. However, what if a briber is lucky and ends up in a Nash equilibrium which is the best for him/her? In this case, can the briber win significantly more votes with a very small budget? Unfortunately, even in the best Nash equilibrium, the fraction of the votes a briber can win may not exceed the portion of the bribing budget he/she owns by $O(1)$ times, as is implied by the following theorem.

Theorem 2. *Let $\epsilon < 1/3$ be an arbitrary small constant and suppose $b_i \geq \epsilon b_1$ for $2 \leq i \leq k$. In any Nash equilibrium, B_1 gets at most $1/\epsilon$ votes or a $\frac{4(1+2\epsilon)b_1}{4(1+2\epsilon)b_1+\sum_{i=2}^k b_i}$ fraction of the votes, whichever is larger.*

Proof. Consider an arbitrary Nash equilibrium. If B_1 only gets $1/\epsilon$ votes in expectation, then the theorem is proved. From now on we assume that B_1 receives more than $1/\epsilon$ votes in expectation. In this case, B_1 must have paid less than $b_1\epsilon$ to some voter, say, V_j, who votes for him/her with a positive probability. Since $b_i \geq \epsilon b_1$, the briber B_i must have received a positive number of votes, for otherwise this briber can devote all the budget to V_j and gets one vote, contradicting the fact that the solution is a Nash equilibrium.

Let $\phi_i > 0$ be the expected number of votes received by each briber B_i. We make the following two assumptions.

- Each B_i pays out a total price of exactly b_i to voters;
- If B_i gets 0 vote from a voter in expectation, B_i pays 0 to this voter.

The two assumptions are without loss of generality since each B_i gets a positive number of votes from at least one voter, and we can simply let B_i pays all the remaining money in his/her budget to this voter if he/she does not use up the budget. By doing so, B_i cannot get fewer votes. The fact that the original solution is a Nash equilibrium ensures that B_i will not get more votes. Thus, the modified solution is still a Nash equilibrium.

We define the average cost per vote for B_i as $a_i = b_i/\phi_i$. Let S_j be the set of bribers who offers the same highest price for V_j, then every briber $B_i \in S_j$ gets in expectation $1/|S_j|$ votes from V_j. For simplicity we remove all the voters where $S_j = \emptyset$ from now on. We define $x_{ij} \in \{0,1\}$ as an indicating variable such that $x_{ij} = 1$ if $B_i \in S_j$ and $x_{ij} = 0$ otherwise. Recall that a briber B_i pays b_i^j to V_j, thus we have

$$\sum_{j=1}^{m} x_{ij}/|S_j| = \phi_i, \qquad \forall i \tag{1a}$$

$$\sum_{j=1}^{m} b_i^j x_{ij} = b_i, \qquad \forall i \tag{1b}$$

There are two possibilities with respect to a_1. If $a_1 \geq \epsilon b_1$, then $\phi_1 \leq 1/\epsilon$, which means B_1 gets at most $1/\epsilon$ votes and Theorem 2 is proved. Otherwise $a_1 < \epsilon b_1$ and there are two possibilities.

Case 1. $|\{j : 0 < b_1^j < (1+2\epsilon)a_1\}| \le 1$. Note that a_1 is the average cost. We claim that $\phi_1 < 1/\epsilon$. Otherwise $\sum_{j=1}^m x_{1j} \ge 1/\epsilon$ and it follows that $\sum_{j=1}^m b_i^j x_{1j} \ge (1+2\epsilon)a_1(\sum_{j=1}^m x_{1j} - 1) = (1+2\epsilon)b_1 - (1+2\epsilon)a_1 > b_1$, where the last inequality follows from the fact that $b_1 \ge (1+2\epsilon)(1/\epsilon - 1)a_1 = (1 + 1/\epsilon - 2\epsilon)a_1$, whereas $2\epsilon b_1 > (1+2\epsilon)a_1$. This, however, is a contradiction to Eq (1b). Therefore, B_1 gets in expectation at most $1/\epsilon$ votes and Theorem 2 is proved.

Case 2. $|\{j : 0 < b_1^j < (1+2\epsilon)a_1\}| \ge 2$. In this case, we have the following lemma.

Lemma 1. *If* $|\{j : b_1^j < (1+2\epsilon)a_1\}| \ge 2$, *then for any* $2 \le i \le k$, $a_1 \ge \frac{a_i}{4(1+2\epsilon)}$.

Proof (Proof of Lemma 1). Towards the proof, we need the following claims.

Claim. For every i, there exists some set of voters Γ_i such that $\sum_{j\in\Gamma_i} x_{ij}/|S_j| \le 1$ and $\sum_{j\in\Gamma_i} b_i^j x_{ij} \ge a_i/2$.

To see the claim, we suppose on the contrary that for every set of voters Γ_i satisfying that $\sum_{j\in\Gamma_i} x_{ij}/|S_j| \le 1$, it holds that $\sum_{j\in\Gamma_i} b_i^j x_{ij} < a_i/2$. We list all the variables $x_{i1}, x_{i2}, \ldots, x_{im}$ and divide them into q subsets where the h-th subset consists of $x_{i,\ell_{h-1}}, x_{i,\ell_{h-1}+1}, \ldots, x_{i,\ell_h-1}$ for $1 = \ell_0 < \ell_1 < \ldots < \ell_q = m+1$, such that the followings hold for every h:

$$\frac{x_{i,\ell_{h-1}}}{|S_{\ell_{h-1}}|} + \frac{x_{i,\ell_{h-1}+1}}{|S_{\ell_{h-1}+1}|} + \ldots + \frac{x_{i,\ell_h-1}}{|S_{\ell_h-1}|} \le 1 \tag{2a}$$

$$\frac{x_{i,\ell_{h-1}}}{|S_{\ell_{h-1}}|} + \frac{x_{i,\ell_{h-1}+1}}{|S_{\ell_{h-1}+1}|} + \ldots + \frac{x_{i,\ell_h-1}}{|S_{\ell_h-1}|} + \frac{x_{i,\ell_h}}{|S_{\ell_h}|} > 1 \tag{2b}$$

By Eq (2a) we have

$$\sum_{s=\ell_{h-1}}^{\ell_h-1} b_i^s x_{is} < a_i/2.$$

Taking the summation over $1 \le h \le q$, we have

$$\sum_{s=\ell_{h-1}}^{\ell_h-1} b_i^s x_{is} < a_i q/2.$$

We show in the following that $q \le 2\phi_i$, whereas

$$\sum_{h=1}^{q} \sum_{s=\ell_{h-1}}^{\ell_h-1} b_i^s x_{is} < a_i q/2 \le a_i\phi_i = b_i,$$

contradicting Eq (1b) and the claim is proved. To see $q \le 2\phi_1$, we can view each $x_{ij}/|S_j|$ as an item of size $x_{ij}/|S_j|$. We pack these items into bins of size 1 one by one using the *Next-fit* algorithm in Bin packing [29], i.e., as long as the item fits in the same bin as the previous item, put it there; otherwise, open a new bin and put it in there. It is easy to see that the Next-fit algorithm returns a solution using q bins with the h-th bin containing exactly $x_{i,\ell_{h-1}}/|S_{\ell_{h-1}}|$ to $x_{i,\ell_h-1}/|S_{\ell_h-1}|$. Note that $\phi_i = \sum_{j=1}^m x_{ij}/|S_j|$ is exactly the total size of all items. It is a classical result [29] that the Next-fit algorithm for bin packing returns a solution that uses the number of bins at most twice the total item size (to see this, simply observe that any two consecutive bins have a total size larger than 1), hence $q \le 2\phi_i$.

We are able to prove Lemma 1 now using the above claim. Suppose on the contrary that for some i it holds that $a_1 < a_i/(4+8\epsilon)$. According to the claim, there exists some Γ_i such that $\sum_j \in \Gamma_i x_{ij}/|S_j| \le 1$ and $\sum_{j\in\Gamma_i} b_i^j x_{ij} \ge a_i/2 > 2(1+2\epsilon)a_1$. Hence, the briber B_i pays in total more than $2(1+2\epsilon)a_1$ and only receive in expectation 1 vote. As $|\{j : 0 < b_1^j < (1+2\epsilon)a_1\}| \ge 2$, there exist at least two voters V_{j_1} and V_{j_2} to whom B_1 pays less than $(1+2\epsilon)a_1$. Since B_1 have received a positive number of votes from each of them (otherwise B_1 would have paid 0), V_{j_1} and V_{j_2} receive offers from bribers with a price less than $(1+2\epsilon)a_1$. Hence, if B_i changes his/her solution unilaterally by paying $(1+2\epsilon)a_1$ to V_{j_1} and V_{j_2}, and meanwhile 0 to voters in Γ_i, he/she gets 2 votes instead, contracting the fact that the solution is a Nash equilibrium. Thus, Lemma 1 is true. □

By Lemma 1, we know that in Case 2 every briber B_i gets at least $\frac{b_i}{4(1+2\epsilon)a_1}$ votes. Therefore, B_1 can get at most $\frac{4(1+2\epsilon)b_1}{4(1+2\epsilon)b_1+\sum_{i=2}^k b_i}$ fraction of the total votes. □

Theorem 2 implies that, even if a briber is very lucky and ends up in a Nash equilibrium which is the best for him/her, he/she cannot get more than $\frac{4(1+2\epsilon)b_1}{4(1+2\epsilon)b_1+\sum_{i=2}^k b_i}$ fraction of the total votes if there are significantly many voters (larger than $1/\epsilon$ which is a constant). By taking $\frac{b_1}{\sum_{i=1}^k b_i} = 1/5$, this fractional value becomes $1/2 + O(\epsilon)$, therefore we have the following corollary.

Corollary 2. *Even in a best Nash equilibrium, a briber needs to control more than 20% of the total bribing budgets in order to get more than 50% of the votes.*

4 Further Discussion

We have shown that, although smart contracts can be used to carry out bribery in a blockchain system, it is, however, much more difficult for a briber to do so than in an ordinary real-world election. The major challenge comes from the fact that a voter is free to establish multiple smart contracts with different bribers and can strategically pick the best one.

A natural question is whether a briber can prevent a bribee from establishing smart contracts with other bribers. One potential approach is to introduce a penalty for a bribee if he/she fails to fulfill the smart contract. Indeed, a recent paper by Abhiram and Christopher [20] presents a pseudocode for such kind of smart contracts. It is questionable whether such smart contracts can change our results substantially. Obviously, if the briber can charge an infinite amount of penalty, then surely the bribee has no choice but to follow the smart contract.

However, this is usually unreasonable. A penalty is usually achieved via a deposit from the bribee to the briber, a sufficiently high penalty may exceed the wallet balance of a voter, which means the briber is losing these potential bribees. More critically, the decision whether a smart contract is fulfilled or not is also achieved through consensus. Once the bribee pays a high deposit, even if he/she fulfills the smart contract, the briber may also bribe others to alter the decision and take away the deposit. Hence, even if penalty may be introduced, it should be reasonably low. A low penalty, however, only prevents a voter from making smart contracts with a lot of bribers. It does not prevent a voter from making smart contracts with only a few bribers, which is already enough to yield a non-cooperative game among bribers and our results readily apply.

5 Related Work

In this section, we briefly review related works.

Smart Contract Systems. Ethereum is by far the most popular smart contract system [4] and many works have been done to detect potential vulnerabilities in smart contracts, see, e.g., [22]. Although game theory has been extensively used to analyze mining activities [5,6,13,27], users' behavior in a smart contract system is not well understood.

Bribery in Elections. There are various researches studying the bribery issue in elections. Faliszewski et al. [15] gave the first systematic characterization on the complexity of the bribery problem where the briber can pay a fixed, but voter-dependent, price to arbitrarily manipulate the preference list of a bribed voter. Different bribery models were addressed subsequently in, e.g., [1,7,8,11,14,16,18]. We refer the readers to [17] for a nice survey on this topic and the references therein.

6 Conclusion and Future Work

Bribery is an important issue in real-world elections. Recent studies have shown that smart contracts can be utilized to conduct bribery in a blockchain system; and it is crucial to understand how smart contract based bribery can influence the whole blockchain system. In this paper, we make the first improvement towards this direction. We cast the bribery problem in a blockchain system as an election and leverage the research in voting systems. We observe that, bribery via smart contracts in a blockchain system is likely to end up in a game situation where different bribers compete with each other in bribing users. We model this problem as a smart contract bribery game and study the behavior of bribers under Nash equilibrium. Interestingly, we show that in *any* Nash equilibrium, a briber cannot win the majority of the votes unless he/she controls more than 20% of the total bribing budgets. Therefore, the phenomenon of "anarchy" in game theory actually helps in discouraging people from carrying out bribery in a blockchain system.

There are several interesting open problems along this line of research. In this paper, we assume every voter has the same weight, i.e., each voter can only cast one vote. However, it is common that voters do have weights. It is not clear whether a constant threshold like 20% also exists when voters/users have weights. Another important problem is to study how to protect the blockchain system through other methods, particularly by deploying resources. It is true that the 20% threshold can discourage people from bribing, but it does not fully defend the system from bribery, especially when some briber owns a large amount of cryptocurrencies. There are several works in the research of voting systems which study the problem of protecting an election by awarding honesty or punishing bribery [36]. It is not clear how to implement a similar scheme in a blockchain system.

References

1. Bredereck, R., Chen, J., Faliszewski, P., Nichterlein, A., Niedermeier, R.: Prices matter for the parameterized complexity of shift bribery. Inf. Comput. **251**, 140–164 (2016)
2. Bredereck, R., Faliszewski, P., Niedermeier, R., Talmon, N.: Complexity of shift bribery in committee elections. In: AAAI, pp. 2452–2458 (2016)
3. Bredereck, R., Faliszewski, P., Niedermeier, R., Talmon, N.: Large-scale election campaigns: combinatorial shift bribery. J. Artif. Intell. Res. **55**, 603–652 (2016)
4. Buterin, V.: A next-generation smart contract and decentralized application platform. white paper (2014)
5. Chen, L., Xu, L., Gao, Z., Shah, N., Lu, Y., Shi, W.: Smart contract execution-the (+-)-biased ballot problem. In: LIPIcs-Leibniz International Proceedings in Informatics. vol. 92. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
6. Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., Shi, W.: Decentralized execution of smart contracts: agent model perspective and its implications. In: Brenner, M., et al. (eds.) FC 2017. LNCS, vol. 10323, pp. 468–477. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70278-0_29
7. Chen, L., et al.: Protecting election from bribery: new approach and computational complexity characterization (extended abstract). In: Proceedings of the 2018 International Conference on Autonomous Agents and Multiagent Systems, vol. 1. International Foundation for Autonomous Agents and Multiagent Systems (2018)
8. Dey, P., Misra, N., Narahari, Y.: Frugal bribery in voting. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 2466–2472. AAAI Press (2016)
9. Dorn, B., Krüger, D.: On the hardness of bribery variants in voting with CP-nets. Ann. Math. Artif. Intell. **77**(3–4), 251–279 (2016)
10. Dorn, B., Krüger, D., Scharpfenecker, P.: Often harder than in the constructive case: destructive bribery in CP-nets. In: Markakis, E., Schäfer, G. (eds.) WINE 2015. LNCS, vol. 9470, pp. 314–327. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48995-6_23
11. Elkind, E., Faliszewski, P., Slinko, A.: Swap bribery. In: Mavronicolas, M., Papadopoulou, V.G. (eds.) SAGT 2009. LNCS, vol. 5814, pp. 299–310. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04645-2_27

12. Erdélyi, G., Reger, C., Yang, Y.: The complexity of bribery and control in group identification. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pp. 1142–1150. International Foundation for Autonomous Agents and Multiagent Systems (2017)
13. Eyal, I., Sirer, E.G.: Majority is not enough: bitcoin mining is vulnerable. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 436–454. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_28
14. Faliszewski, P.: Nonuniform bribery. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1569–1572. International Foundation for Autonomous Agents and Multiagent Systems (2008)
15. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: How hard is bribery in elections? J. Artif. Intell. Res. **35**, 485–532 (2009)
16. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: Llull and copeland voting computationally resist bribery and constructive control. J. Artif. Intell. Res. **35**, 275–341 (2009)
17. Faliszewski, P., Rothe, J.: Control and Bribery in Voting. Cambridge University Press, Cambridge (2016)
18. Kaczmarczyk, A., Faliszewski, P.: Algorithms for destructive shift bribery. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pp. 305–313. International Foundation for Autonomous Agents and Multiagent Systems (2016)
19. Knop, D., Kouteckỳ, M., Mnich, M.: Voting and bribing in single-exponential time. In: LIPIcs-Leibniz International Proceedings in Informatics, vol. 66. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)
20. Kothapalli, A., Cordi, C.: A bribery framework using smartcontracts (2017)
21. Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J.S.: Bitcoin mining pools: a cooperative game theoretic analysis. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 919–927. International Foundation for Autonomous Agents and Multiagent Systems (2015)
22. Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 254–269. ACM (2016)
23. Mattei, N., Pini, M.S., Venable, K.B., Rossi, F.: Bribery in voting over combinatorial domains is easy. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 3, pp. 1407–1408. International Foundation for Autonomous Agents and Multiagent Systems (2012)
24. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
25. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
26. Pini, M.S., Rossi, F., Venable, K.B.: Bribery in voting with soft constraints. In: AAAI (2013)
27. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. arXiv preprint arXiv:1507.06183 (2015)
28. Szabo, N.: Formalizing and securing relationships on public networks. First Monday **2**(9) (1997)
29. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-662-04565-7
30. Vukolić, M.: The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: Camenisch, J., Kesdoğan, D. (eds.) iNetSec 2015. LNCS, vol. 9591, pp. 112–125. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39028-4_9

31. Xu, L., Chen, L., Gao, Z., Lu, Y., Shi, W.: CoC: secure supply chain management system based on public ledger. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–6. IEEE (2017)
32. Xu, L., Chen, L., Shah, N., Gao, Z., Lu, Y., Shi, W.: DL-BAC: distributed ledger based access control for web applications. In: Proceedings of the 26th International Conference on World Wide Web Companion, pp. 1445–1450. International World Wide Web Conferences Steering Committee (2017)
33. Xu, L., et al.: Enabling the sharing economy: privacy respecting contract based on public blockchain. In: Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts, pp. 15–21. ACM (2017)
34. Yang, Y., Shrestha, Y.R., Guo, J.: How hard is bribery in party based elections? In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pp. 1725–1726. International Foundation for Autonomous Agents and Multiagent Systems (2015)
35. Yang, Y., Shrestha, Y.R., Guo, J.: How hard is bribery with distance restrictions? In: ECAI, pp. 363–371 (2016)
36. Yin, Y., Vorobeychik, Y., An, B., Hazon, N.: Optimally protecting elections. In: IJCAI, pp. 538–545 (2016)