Protecting Election from Bribery: New Approach and Computational Complexity Characterization

Extended Abstract

Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, Nolan Shah, Yang Lu, Weidong Shi
University of Houston
Houston, TX
chenlin198662@gmail.com,xuleimath@gmail.com, shouhuai.xu@utsa.edu,mtion@msn.com,nshah10@uh.edu,ylu17@central.uh.edu,wshi3@uh.edu

Shouhuai Xu
University of Texas at San Antonio
San Antonio, TX
shouhuai.xu@utsa.edu

ABSTRACT

The *bribery problem* in elections has received a considerable amount of attention. In this paper, we initiate the study of a related, but new problem, the *protection problem*, namely protecting elections from bribery. In this problem, there is a defender who is given a defense budget and can use the budget to award some of the voters such that they cannot be bribed anymore. This naturally leads to the following bi-level decision problem: Is it possible for the defender with a given defense budget to protect an election from being manipulated by the attacker with a given attack budget for bribing voters? We characterize the computational complexity of the protection problem. We show that it is in general significantly harder than the bribery problem. However, the protection problem can be solved, under certain circumstances, in polynomial time.

KEYWORDS

Voting; complexity; bribery

ACM Reference Format:
Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, Nolan Shah, Yang Lu, Weidong Shi and Shouhuai Xu. 2018. Protecting Election from Bribery: New Approach and Computational Complexity Characterization. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

In an election, there are a set of candidates and a set of voters. Each voter has a *preference list* of candidates. Given these preference lists, a winner is determined based on some *voting rule*, examples of which will be elaborated later.

In the context of election, the *bribery problem* has received considerable attention (see, for example, [1–3, 8, 9, 11, 13, 15, 16, 18, 19, 22, 23]). In this problem, there is an attacker (briber) who attempts to manipulate the election by bribing some voters, who will then report preference lists of the attacker's choice (rather than the voters' own preference lists). Each voter has a cost for being bribed, and the attacker has an attack budget for bribing. There are two kinds of attackers: *constructive* vs. *destructive* attacker. A *constructive* attacker that attempts to make a designated candidate win the election, where the designated candidate is chosen by the attacker and would not win the election should there be no bribery attacker. In contrast, a *destructive* attacker that attempts to make a designated candidate lose the election, where the designated candidate is also chosen by the attacker and would win the election should there be no bribery attacker. The research question is: Given an attack budget for bribing, whether or not a (constructive or destructive) attacker can achieve its goal?

In this paper, we study the protection of elections against bribery. More specifically, we initiate the study of the following *protection problem*, which extends the bribery problem as follows. There are also a set of candidates, a set of voters, and an attacker. Each voter also has a *preference list* of candidates. There is also a *voting rule* according to which a winner is determined. Going beyond the bribery problem, the protection problem further considers a defender who is given a defense budget. The defender can use this budget to award a subset of voters such that these awarded voters cannot be bribed by an attacker anymore. This leads to an interesting problem: *Given a defense budget, is it possible to protect an election from an attacker with a given attack budget for bribing?*

Our contributions. We introduce a new defense approach to protecting elections from bribery. Given a defense budget for rewarding and an attack budget for bribery, the protection problem asks whether or not the election can be protected. We investigate the protection problem against the aforementioned two kinds of attackers: *constructive* vs. *destructive* attacker.

We present a characterization on the computational complexity of the protection problem. Our main results are summerized in Table 1). The characterization is primarily with respect to the voting rule of r-approval, which will be elaborated in Section 2. At a high level, our results can be summarized as follows. (i) The *protection problem* is a hard problem and might be much harder than the *bribery problem*, which has been extensively studied in the literature. For example, the protection problem is Σ_2^p-complete in most cases, while the bribery problem is in NP under the same settings. (ii) The protection problem with a destructive attacker is *no* harder than the protection problem with a constructive attacker in any of the settings we considered. In particular, the protection problem with a destructive attacker is Σ_2^p-hard only when the voters are weighted and have arbitrary prices, while the protection problem with a constructive attacker is Σ_2^p-hard even when the voters are unweighted and have the unit price. (iii) Voter weights and prices

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

# of candidates	Model parameters	Destructive security	Constructive security
constant	Weighted, Priced	Σ_2^p-complete ⋄	Σ_2^p-complete ⋄
	Weighted, $p_j = p'_j = 1$	P	coNP-hard
	$w_j = 1$, Priced	NP-complete ⋄	NP-complete ⋄
	$w_j = 1$, Priced, Symmetric	P	P
	$w_j = 1, p_j = p'_j = 1$	P	P
arbitrary	Weighted, Priced	Σ_2^p-complete ⋄	Σ_2^p-complete ⋄
	Weighted, $p_j = p'_j = 1$	NP-complete	Σ_2^p-complete
	$w_j = 1$, Priced	NP-complete	Σ_2^p-complete
	$w_j = 1$, Priced, Symmetric	NP-complete	Σ_2^p-complete
	$w_j = 1, p_j = p'_j = 1$	?	Σ_2^p-complete

Table 1: Summary of results for *single-winner* election under the voting rule of *r-approval*: "Symmetric" means $p_j = p'_j$ for every j, otherwise (without specifying "Symmetric") the values of p_j and p'_j may or may not equal; hardness results that are proved for the case with only two candidates (i.e., $m = 2$) are marked with a "⋄"algorithmic results (marked with a "P") also hold for arbitrary scoring rules.

have completely different effects on the computational complexity of the protection problem. For example, the protection problem with a constructive attacker is coNP-hard in one case, but is in P in another.

Related Work. The most relevant work to ours is [24], which considers the problem of defending elections against an attacker who can delete (groups of) voters. We are not aware of an analysis of a similar defending approach against bribery. Without protection, the bribery problem is much well understood. Faliszewski et al. [13] gave the first systematic characterization on the complexity of the *bribery problem*, followed by a series of further researches [1, 4, 7, 10, 12, 14, 15, 17]. Technically, the protection problem we study is related to the *bi-level optimization* problem, especially the *bi-level knapsack* problem ([5, 6, 20, 21]).

2 PROBLEM DEFINITION

Election model and rules. Consider a set of m candidates $C = \{c_1, c_2, \ldots, c_m\}$ and a set of n voters $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. Every voter v_j has a weight w_j. We focus on r-approval rule, in which every voter votes for r candidates. Let C_i be the set of voters who votes for candidate c_i, then this candidate gets in total $\sum_{j \in C_i} w_j$ votes. The winner of the election is the candidate that receives the highest number of votes. We focus on *single-winner* election, meaning that only one winner is selected.

Attacker and Defender. We consider an attacker who attempts to manipulate the election by bribing some voters. Suppose voter v_j has a *bribing price* p'_j, meaning that v_j, upon receiving a bribery of amount p'_j from the attacker, will change its votes according to the suggestion of the attacker. The attacker has a total budget B. In line with the bribery problem studied in the literature, we consider two kinds of attackers:

- *Constructive attacker*: This attacker attempts to make a designated candidate win the election, meaning that the designated candidate is the only candidate who gets the highest number of votes.
- *Destructive attacker*: This attacker attempts to make a designated candidate lose the election, meaning that there is another candidate that gets a strictly higher number of votes than the designated candidate does.

In the protection problem, voter v_j, upon receiving an award of amount p_j (or *awarding price*) from the defender, will always report its preference list faithfully and cannot be bribed. Note that p_j may have multiple interpretations, such as monetary award, economic incentives or the cost of isolating voters from bribery. We say a voter v_j is *awarded* if v_j receives an award of p_j.

The constructive protection problem (i.e., protecting elections against constructive attackers):
Input: A set C of m candidates. A set $\mathcal{V}$ of n voters, each with a weight w_j, a preference list τ_j, an awarding price of p_j and a bribing price of p'_j. A scoring rule for selecting a single winner. A defender with a defense budget F. An attacker with an attack budget B attempting to make candidate c_i win the election.
Output: Decide whether there exists a $\mathcal{V}_F \subseteq \mathcal{V}$ such that

- $\sum_{j:v_j \in \mathcal{V}_F} p_j \le F$; and
- for *any* subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ with $\sum_{j:v_j \in \mathcal{V}_B} p'_j \le B$, c_i does not get a strictly higher score than any other candidate despite that the attacker bribes the voters belonging to $\mathcal{V}_B$ (i.e., bribing $\mathcal{V}_B$).

The destructive protection problem (i.e., protecting elections against destructive attackers):
Input: A set C of m candidates. A set $\mathcal{V}$ of n voters, each with a weight w_j, a preference list τ_j, an awarding price of p_j and a bribing price of p'_j. A scoring rule for selecting a single winner. Suppose c_m is the winner if every voter is honest. A defender with a defense budget F. An attacker with an attack budget B attempting to make c_m lose the election by making $c \in C \setminus \{c_m\}$ get a strictly higher score than c_m does.
Output: Decide if there exists a $\mathcal{V}_F \subseteq \mathcal{V}$ such that

- $\sum_{j:v_j \in \mathcal{V}_F} p_j \le F$; and
- for *any* subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ such that $\sum_{j:v_j \in \mathcal{V}_B} p'_j \le B$, no candidate $c \in C \setminus \{c_m\}$ can get a strictly higher score than c_m does despite that the attacker bribes $\mathcal{V}_B$.

REFERENCES

[1] Robert Bredereck, Jiehua Chen, Piotr Faliszewski, André Nichterlein, and Rolf Niedermeier. 2016. Prices matter for the parameterized complexity of shift bribery. *Information and Computation* 251 (2016), 140–164.

[2] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. 2016. Complexity of Shift Bribery in Committee Elections.. In *AAAI*. 2452–2458.

[3] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. 2016. Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research* 55 (2016), 603–652.

[4] Robert Bredereck and Nimrod Talmon. 2016. NP-hardness of two edge cover generalizations with applications to control and bribery for approval voting. *Inform. Process. Lett.* 116, 2 (2016), 147–152.

[5] Alberto Caprara, Margarida Carvalho, Andrea Lodi, and Gerhard J Woeginger. 2014. A study on the computational complexity of the bilevel knapsack problem. *SIAM Journal on Optimization* 24, 2 (2014), 823–838.

[6] Lin Chen and Guochuan Zhang. 2013. Approximation algorithms for a bi-level knapsack problem. *Theoretical Computer Science* 497 (2013), 1–12.

[7] Palash Dey, Neeldhara Misra, and Y Narahari. 2016. Frugal bribery in voting. In *AAAI*. AAAI Press, 2466–2472.

[8] Britta Dorn and Dominikus Krüger. 2016. On the hardness of bribery variants in voting with CP-nets. *Annals of Mathematics and Artificial Intelligence* 77, 3-4 (2016), 251–279.

[9] Britta Dorn, Dominikus Krüger, and Patrick Scharpfenecker. 2015. Often harder than in the constructive case: destructive bribery in CP-nets. In *WINE*. Springer, 314–327.

[10] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2009. Swap bribery. In *SAGT*. Springer, 299–310.

[11] Gábor Erdélyi, Christian Reger, and Yongjie Yang. 2017. The Complexity of Bribery and Control in Group Identification. In *AAMAS*. IFAAMAS, 1142–1150.

[12] Piotr Faliszewski. 2008. Nonuniform bribery. In *AAMAS*. IFAAMAS, 1569–1572.

[13] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. 2009. How hard is bribery in elections? *Journal of Artificial Intelligence Research* 35 (2009), 485–532.

[14] Piotr Faliszewski, Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. 2009. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research* 35 (2009), 275–341.

[15] Andrzej Kaczmarczyk and Piotr Faliszewski. 2016. Algorithms for destructive shift bribery. In *AAMAS*. IFAAMAS, 305–313.

[16] Dusan Knop, Martin Koutecký, and Matthias Mnich. 2017. Voting and Bribing in Single-Exponential Time. In *STACS*.

[17] Andrew Lin. 2010. The Complexity of Manipulating k-Approval Elections. *arXiv preprint arXiv:1005.4159* (2010).

[18] Nicholas Mattei, Maria Silvia Pini, K Brent Venable, and Francesca Rossi. 2012. Bribery in voting over combinatorial domains is easy. In *AAMAS*. IFAAMAS, 1407–1408.

[19] Maria Silvia Pini, Francesca Rossi, and Kristen Brent Venable. 2013. Bribery in Voting With Soft Constraints.. In *AAAI*.

[20] Xian Qiu and Walter Kern. 2015. Improved approximation algorithms for a bilevel knapsack problem. *Theoretical computer science* 595 (2015), 120–129.

[21] Zhenbo Wang, Wenxun Xing, and Shu-Cherng Fang. 2010. Two-group knapsack game. *Theoretical Computer Science* 411, 7-9 (2010), 1094–1103.

[22] Yongjie Yang, Yash Raj Shrestha, and Jiong Guo. 2015. How Hard is Bribery in Party Based Elections?. In *AAMAS*. IFAAMAS, 1725–1726.

[23] Yongjie Yang, Yash Raj Shrestha, and Jiong Guo. 2016. How Hard Is Bribery with Distance Restrictions?. In *ECAI*. 363–371.

[24] Yue Yin, Yevgeniy Vorobeychik, Bo An, and Noam Hazon. 2016. Optimally Protecting Elections.. In *IJCAI*. 538–545.