The Complexity of Probabilistic Lobbying ⋆

Gábor Erdélyi[1], Henning Fernau[2], Judy Goldsmith[3],
Nicholas Mattei[3], Daniel Raible[2], and Jörg Rothe[1]

[1] Universität Düsseldorf, Institut für Informatik, 40225 Düsseldorf, Germany
[2] Universität Trier, FB 4—Abteilung Informatik, 54286 Trier, Germany
[3] University of Kentucky, Department of Computer Science, Lexington, KY 40506, USA

Abstract. We propose various models for lobbying in a probabilistic environment, in which an actor (called "The Lobby") seeks to influence the voters' preferences of voting for or against multiple issues when the voters' preferences are represented in terms of probabilities. In particular, we provide two evaluation criteria and three bribery methods to formally describe these models, and we consider the resulting forms of lobbying with and without issue weighting. We provide a formal analysis for these problems of lobbying in a stochastic environment, and determine their classical and parameterized complexity depending on the given bribery/evaluation criteria. Specifically, we show that some of these problems can be solved in polynomial time, some are NP-complete but fixed-parameter tractable, and some are W[2]-complete. Finally, we provide (in)approximability results.

1 Introduction

In the American political system, laws are passed by elected officials who are supposed to represent their constituency. Many factors can affect a representative's vote on a particular issue: a representative's personal beliefs about the issue, campaign contributions, communications from constituents, communications from potential donors, and the representative's own expectations of further contributions and political support.

It is a complicated process to reason about. Earlier work considered the problem of meting out contributions to representatives in order to pass a set of laws or influence a set of votes. However, the earlier computational complexity work on this problem made the assumption that a politician who accepts a contribution will in fact—if the contribution meets a given threshold—vote according to the wishes of the donor.

It is said that "An honest politician is one who stays bought," but that does not take into account the ongoing pressures from personal convictions and opposing lobbyists and donors. We consider the problem of influencing a set of votes under the assumption that we can influence only the *probability* that the politician votes as we desire. The methods for exerting influence on the voters is discussed in the section on bribery criteria while the notion of sufficient influence for a voter is discussed in the section on evaluation criteria.

⋆ Supported in part by DFG grants RO 1202/11-1 and RO 1202/12-1, the European Science Foundation's EUROCORES program LogICCC, the Alexander von Humboldt Foundation's TransCoop program, and NSF grant ITR-0325063.

Lobbying has been studied formally by economists, computer scientists, and special interest groups since at least 1983 [13] and as an extension to formal game theory since 1944 [15]. Each discipline has considered mostly disjoint aspects of the process while seeking to accomplish distinct goals with their respective formal models. Economists have formalized models and studied them as "economic games," as defined by von Neumann and Morgenstern [15]. This analysis is focused on learning how these complex systems work and deducing optimal strategies for winning the competitions [13,1,2]. This work has also focused on how to "rig" a vote and how to optimally dispense the funds among the various individuals [1]. Economists are interested in finding effective and efficient bribery schemes [1] as well as determining strategies for instances of two or more players [1,13,2]. Generally, they reduce the problem of finding an effective lobbying strategy to one of finding a winning strategy for the specific type of game. Economists have also formalized this problem for bribery systems in both the United States [13] and the European Union [6].

In the emerging field of computational social choice, voting and preference aggregation are studied from a computational perspective, with a particular focus on the complexity of winner determination, manipulation, procedural control, and bribery in elections (see, e.g., the survey [9] and the references cited therein), and also with respect to lobbying in the context of direct democracy where voters vote on multiple referenda. In particular, Christian et al. [5] show that "Optimal Lobbying" (OL) is complete for the (parameterized) complexity class $W[2]$. The OL problem is a deterministic and non-weighted version of the problems that we present in this paper. Sandholm noted that the "Optimal Weighted Lobbying" (OWL) problem, which allows different voters to have different prices, can be expressed as and solved via the "binary multi-unit combinatorial reverse auction winner-determination problem" (see [14]).

We extend the models of lobbying, and provide algorithms and analysis for these extended models in terms of classical and parameterized complexity. Our problems are still related to the reverse auction winner-determination problem—in particular, our extensions of the optimal lobbying problem allow the seller to express desire over the objects, thus crucially changing the original problem in both the economic and complexity-theoretic senses. This change is a result of the probabilistic modeling of the seller's reaction to the bribery. We also show novel computational and algorithmic approaches to these new problems. In this way we add breadth and depth to not only the models but also the understanding of lobbying behavior.

2 Models for Probabilistic Lobbying

2.1 Initial Model

We begin with a simplistic version of the PROBABILISTIC LOBBYING PROBLEM (PLP, for short), in which voters start with initial probabilities of voting for an issue and are assigned known costs for increasing their probabilities of voting according to "The Lobby's" agenda by each of a finite set of increments.

The question, for this class of problems, is: Given the above information, along with an agenda and a fixed budget B, can The Lobby target its bribes in order to achieve its agenda? The complexity of the problem seems to hinge on the evaluation criterion for what it means to "win a vote" or "achieve an agenda." We discuss the possible interpretations of evaluation and bribery later in this section. First, however, we will formalize the problem by defining data objects needed to represent the problem instances.

Let $\mathbb{Q}_{[0,1]}^{m\times n}$ denote the set of $m\times n$ matrices over $\mathbb{Q}_{[0,1]}$ (the rational numbers in the interval $[0,1]$). We say $P \in \mathbb{Q}_{[0,1]}^{m\times n}$ is a probability matrix (of size $m\times n$), where each entry $p_{i,j}$ of P gives the probability that voter v_i will vote "yes" for referendum (synonymously, for issue) r_j. The result of a vote can be either a "yes" (represented by 1) or a "no" (represented by 0). Thus, we can represent the result of any vote on all issues as a $0/1$ vector $X = (x_1, x_2, \dots, x_n)$, which is sometimes also denoted as a string in $\{0,1\}^n$.

We now associate with each pair (v_i, r_j) of voter/issue, a discrete price function $c_{i,j}$ for changing v_i's probability of voting "yes" for issue r_j. Intuitively, $c_{i,j}$ gives the cost for The Lobby of raising or lowering (in discrete steps) the ith voter's probability of voting "yes" on the jth issue. A formal description is as follows.

Given the entries $p_{i,j} = a_{i,j}/b_{i,j}$ of a probability matrix $P \in \mathbb{Q}_{[0,1]}^{m\times n}$, choose some $k \in \mathbb{N}$ such that $k+1$ is a common multiple of all $b_{i,j}$, where $1 \le i \le m$ and $1 \le j \le n$, and partition the probability interval $[0,1]$ into $k+1$ steps of size $1/(k+1)$ each. For each $i \in \{1,2,\dots,m\}$ and $j \in \{1,2,\dots,n\}$, $c_{i,j} : \{0, 1/(k+1), 2/(k+1), \dots, k/(k+1), 1\} \to \mathbb{N}$ is the *(discrete) price function for* $p_{i,j}$, i.e., $c_{i,j}(\ell/(k+1))$ is the price for changing the probability of the ith voter voting "yes" on the jth issue from $p_{i,j}$ to $\ell/(k+1)$, where $0 \le \ell \le k+1$. Note that the domain of $c_{i,j}$ consists of $k+2$ elements of $\mathbb{Q}_{[0,1]}$ including 0, $p_{i,j}$, and 1. In particular, we require $c_{i,j}(p_{i,j}) = 0$, i.e., a cost of zero is associated with leaving the initial probability of voter v_i voting on issue r_j unchanged. Note that $k = 0$ means $p_{i,j} \in \{0,1\}$, i.e., in this case each voter either accepts or rejects each issue with certainty and The Lobby can only flip these results.[4] The image of $c_{i,j}$ consists of $k+2$ nonnegative integers including 0, and we require that, for any two elements a, b in the domain of $c_{i,j}$, if $p_{i,j} \le a \le b$ or $p_{i,j} \ge a \ge b$, then $c_{i,j}(a) \le c_{i,j}(b)$. This guarantees monotonicity on the prices.

We represent the list of price functions associated with a probability matrix P as a table C_P whose $m \cdot n$ rows give the price functions $c_{i,j}$ and whose $k+2$ columns give the costs $c_{i,j}(\ell/(k+1))$, where $0 \le \ell \le k+1$. Note that we choose the same k for each $c_{i,j}$, so we have the same number of columns in each row of C_P. The entries of C_P can be thought of as "price tags" that The Lobby must pay in order to change the probabilities of voting.

The Lobby also has an integer-valued budget B and an "agenda," which we will denote as a vector $Z \in \{0,1\}^n$, where n is the number of issues, containing the outcomes The Lobby would like to see on the corresponding issues. For simplicity, we may assume that The Lobby's agenda is all "yes" votes, so the target vector is $Z = 1^n$. This assumption can be made without loss of generality, since if there is a zero in Z at position j, we can flip this zero to one and also change the corresponding probabilities $p_{1,j}, p_{2,j}, \dots, p_{m,j}$ in the jth column of P to $1-p_{1,j}, 1-p_{2,j}, \dots, 1-p_{m,j}$ (see the evaluation criteria in Section 2.3 for how to determine the result of voting on a referendum).

[4] This is the special case of Optimal Lobbying.

Example 1. We create a problem instance with $k = 9$, $m = 2$ (number of voters), and $n = 3$ (number of issues). We will use this as a running example for the rest of this paper. In addition to the above definitions for k, m, and n, we also give the following matrix for P. (Note that this example is normalized for an agenda of $Z = 1^3$, which is why The Lobby has no incentive for lowering the acceptance probabilities, so those costs are omitted below.)

Our example consists of a probability matrix P:

	r_1	r_2	r_3
v_1	0.8	0.3	0.5
v_2	0.4	0.7	0.4

and the corresponding cost matrix C_P:

$c_{i,j}$	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$c_{1,1}$	−−	−−	−−	−−	−−	−−	−−	−−	0	100	140
$c_{1,2}$	−−	−−	−−	0	10	70	100	140	310	520	600
$c_{1,3}$	−−	−−	−−	−−	−−	0	15	25	70	90	150
$c_{2,1}$	−−	−−	−−	−−	0	30	40	70	120	200	270
$c_{2,2}$	−−	−−	−−	−−	−−	−−	−−	0	10	40	90
$c_{2,3}$	−−	−−	−−	−−	0	70	90	100	180	300	450

In Section 2.2, we describe three bribery methods, i.e., three specific ways in which The Lobby can influence the voters. These will be referred to as B_i, $i \in \{1,2,3\}$. In addition to the three bribery methods described in Section 2.2, we also define two ways in which The Lobby can win a set of votes. These evaluation criteria are defined in Section 2.3 and will be referred to as C_j, $j \in \{1,2\}$. They are important because votes counted in different ways can result in different outcomes depending on voting and evaluation systems (cf. Myerson and Weber [11]).

We now introduce the six basic problems that we will study. For $i \in \{1,2,3\}$ and $j \in \{1,2\}$, we define:

Name: B_i-C_j PROBABILISTIC LOBBYING PROBLEM (B_i-C_j-PLP, for short).

Given: A probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ with table C_P of price functions, a target vector $Z \in \{0,1\}^n$, and a budget B.

Question: Is there a way for The Lobby to influence P (using bribery method B_i and evaluation criterion C_j, without exceeding budget B) such that the result of the votes on all issues equals Z?

2.2 Bribery Methods

We begin by first formalizing the bribery methods by which The Lobby can influence votes on issues. We will define three methods for donating this money.

Microbribery (B_1) The first method at the disposal of The Lobby is what we will call *microbribery*. We define microbribery to be the editing of individual elements of the P matrix according to the costs in the C_P matrix. Thus The Lobby picks not only which voter to influence but also which issue to influence for that voter. This bribery method allows the most flexible version of bribery, and models private donations made to candidates in support of specific issues.

Issue Bribery (B_2) The second method at the disposal of The Lobby is *issue bribery*. We can see from the P matrix that each column represents how the voters think about a particular issue. In this method of bribery, The Lobby can pick a column of the matrix and edit it according to some budget. The money will be equally distributed among all the voters and the voter probabilities will move accordingly. So, for d dollars each voter receives a fraction of d/m and their probability of voting "yes" changes accordingly. This can be thought of as special-interest group donations. Special-interest groups such as PETA focus on issues and dispense their funds across an issue rather than by voter. The bribery could be funneled through such groups.

Voter Bribery (B_3) The third and final method at the disposal of The Lobby is *voter bribery*. We can see from the P matrix that each row represents what an individual voter thinks about all the issues on the docket. In this method of bribery, The Lobby picks a voter and then pays to edit the entire row at once with the funds being equally distributed over all the issues. So, for d dollars a fraction of d/n is spent on each issue, which moves accordingly. The cost of moving the voter is generated using the C_P matrix as before. This method of bribery is analogous to "buying" or pushing a single politician or voter. The Lobby seeks to donate so much money to an individual voter that he or she has no choice but to move his or her votes toward The Lobby's agenda.

2.3 Evaluation Criteria

Defining criteria for how an issue is won is the next important step in formalizing our models. Here we define two methods that one could use to evaluate the eventual outcome of a vote. Since we are focusing on problems that are probabilistic in nature, it is important to note that no evaluation criteria will guarantee a win. The criteria below yield different outcomes depending on the model and problem instance.

Strict Majority (C_1) For each issue, a strict majority of the individual voters have probability at least some threshold, t, of voting according to the agenda. In our running example (see Example 1), with $t = 50\%$, the result of the votes would be $X = (0,0,0)$, because none of the issues has a strict majority of voters with above 50% likelihood of voting according to the agenda.

Average Majority (C_2) For each issue, r_j, of a given probability matrix P, we define: $\overline{p_j} = (\Sigma_{i=1}^{m} p_{i,j})/m$. We can now evaluate the vote to say that r_j is accepted if and only if $\overline{p_j} > t$ where t is some threshold. This would, in our running example, with $t = 50\%$, give us a result vector of $X = (1,0,0)$.

2.4 Issue Weighting

Our modification to the model will bring in the concept of issue weighting. It is reasonable to surmise that certain issues will be of more importance to The Lobby than others. For this reason we will allow The Lobby to specify higher weights to the issues that they deem more important. These weights will be defined for each issue.

We will specify these weights as a vector $W \in \mathbb{Z}^n$ with size n equal to the total number of issues in our problem instance. The higher the weight, the more important that particular issue is to The Lobby. Along with the weights for each issue we are also given an objective value $O \in \mathbb{Z}^+$ which is the minimum weight The Lobby wants to see passed. Since this is a partial ordering, it is possible for The Lobby to have an ordering such as: $w_1 = w_2 = \cdots = w_n$. If this is the case, we see that we are left with an instance of B_i-C_j-PLP.

We now introduce the six probabilistic lobbying problems with issue weighting. For $i \in \{1,2,3\}$ and $j \in \{1,2\}$, we define:

Name: B_i-C_j-PROBABILISTIC LOBBYING PROBLEM WITH ISSUE WEIGHTING (B_i-C_j-PLP-WIW, for short).

Given: A probability matrix $P \in \mathbb{Q}^{m\times n}_{[0,1]}$ with table C_P of price functions and a lobby target vector $Z \in \{0,1\}^n$, a lobby weight vector $W \in \mathbb{Z}^n$, an objective value $O \in \mathbb{Z}^+$, and a budget B.

Question: Is there a way for The Lobby to influence P (using bribery method B_i and evaluation criterion C_j, without exceeding budget B) such that the total weight of all issues for which the result coincides with The Lobby's target vector Z is at least O?

3 Complexity-Theoretic Notions

We assume the reader is familiar with standard notions of (classical) complexity theory, such as P, NP, and NP-completeness. Since we analyze the problems stated in Section 2 not only in terms of their classical complexity, but also with regard to their *parameterized* complexity, we provide some basic notions here (see, e.g., Downey and Fellows [7] for more background). As we derive our results in a rather specific fashion, we will employ the "Turing way" as proposed by Cesati [4].

A *parameterized problem* $\mathscr{P}$ is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a fixed alphabet and $\mathbb{N}$ is the set of nonnegative integers. Each instance of the parameterized problem $\mathscr{P}$ is a pair (I,k), where the second component k is called the *parameter*. The language $L(\mathscr{P})$ is the set of all YES instances of $\mathscr{P}$. The parameterized problem $\mathscr{P}$ is *fixed-parameter tractable* if there is an algorithm (realizable by a deterministic Turing machine) that decides whether an input (I,k) is a member of $L(\mathscr{P})$ in time $f(k)|I|^c$, where c is a fixed constant and f is a function whose argument k is independent of the overall input length, $|I|$. The class of all fixed-parameter tractable problems is denoted by FPT.

There is also a theory of parameterized hardness (see, e.g., [7]), most notably the W[t] hierarchy, which complements fixed-parameter tractability: $\mathrm{FPT} = \mathrm{W}[0] \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots$. It is commonly believed that this hierarchy is strict. Only the second level, W[2], will be of interest to us in this paper (see, e.g., [7] for the definition).

Table 1. Complexity results for B_i-C_j-PLP

Bribery Criterion	Evaluation Criterion	
	C_1	C_2
B_1	P	P
B_2	P	P
B_3	W[2]-complete	W[2]-complete

The complexity of a classical problem depends on the chosen parameterization. For problems that involve a budget $B \in \mathbb{N}$ (and hence can be viewed as minimization problems), the most obvious parameterization would be the given budget bound B. In this sense, we state parameterized results in this paper. (For other applications of fixed-parameter tractability and parameterized complexity to problems from computational social choice, see, e.g., [10].)

4 Classical Complexity Results

We now provide a formal complexity analysis of the probabilistic lobbying problems for all combinations of evaluation criteria and bribery methods.

Table 1 summarizes our results for B_i-C_j-PLP, $i \in \{1,2,3\}$ and $j \in \{1,2\}$. Some of these results are known from previous work by Christian et al. [5], as will be mentioned below. In this sense, our results generalize the results of [5] by extending the model to probabilistic settings.

4.1 Microbribery

The following result can be easily seen.

Theorem 1. B_1-C_1-PLP *is in* P.

The complexity of microbribery with evaluation criterion C_2 is somewhat harder to determine. We use the following auxiliary problem. Here, a *schedule S* of q jobs (on a single machine) is a sequence $J_{i(1)},\dots,J_{i(q)}$ such that $J_{i(r)} = J_{i(s)}$ implies $r = s$. The *cost of schedule S* is $c(S) = \sum_{k=1}^{q} c(J_{i(k)})$. S is said to *respect the precedence constraints* of graph G if for every (path)-component $P_i = J_{i,1},\dots,J_{i,p(i)}$ and for each k with $2 \leq k \leq p(i)$, we have: If $J_{i,k}$ occurs in the schedule S then $J_{i,k-1}$ occurs in S before $J_{i,k}$.

Name: PATH SCHEDULE
Given: A set $V = \{J_1,\dots,J_n\}$ of jobs, a directed graph $G = (V,A)$ consisting of pairwise disjoint paths $P_1,\dots,P_z$, two numbers $C, q \in \mathbb{N}$, and a cost function $c : V \to \mathbb{N}$.
Question: Can we find a schedule $J_{i(1)},\dots,J_{i(q)}$ of q jobs of cost at most C respecting the precedence constraints of G?

PATH SCHEDULE is in P by dynamic programming. Then we show how to reduce B_1-C_2-PLP to PATH SCHEDULE, which implies that B_1-C_2-PLP is in P as well.

Table 2. Complexity results for B_i-C_j-PLP-WIW

Bribery Criterion	Evaluation Criterion	
	C_1	C_2
B_1	NP-compl., FPT	NP-compl., FPT
B_2	NP-compl., FPT	NP-compl., FPT
B_3	W[2]-complete	W[2]-complete

Lemma 1. PATH SCHEDULE *is in* P.

Theorem 2. B_1-C_2-PLP *is in* P.

Proof. Let (P, C_P, Z, B) be a given B_1-C_2-PLP instance, where $P \in \mathbb{Q}^{m\times n}_{[0,1]}$, C_P is a table of price functions, $Z \in \{0,1\}^n$ is The Lobby's target vector, and B is its budget. For $j \in \{1,2,\dots,n\}$, let d_j be the minimum cost for The Lobby to bring referendum r_j into line with the jth entry of its target vector Z. If $\sum_{j=1}^{n} d_j \leq B$ then The Lobby can achieve its goal that the votes on all issues equal Z. We now focus on the first task. For every r_j, create an equivalent PATH SCHEDULING instance. First, compute for r_j the minimum number b_j of bribery steps needed to achieve The Lobby's goal on r_j. That is, choose the smallest $b_j \in \mathbb{N}$ such that $\overline{p_j} + {}^{b_j}\!/\!{}_{(k+1)m} > t$. Now, for every voter v_i, derive a path P_i from the price function $c_{i,j}$. Let s, $0 \leq s \leq k+1$, be minimum with the property $c_{i,j}(s) \in \mathbb{N}_{>0}$. Then create a path $P_i = p_s, \dots, p_{k+1}$, where p_h represents the hth entry of $c_{i,j}$ (viewed as a vector). Assign the cost $\hat{c}(p_h) = c_{i,j}(h) - c_{i,j}(h-1)$ to p_h. Observe that $\hat{c}(p_h)$ represents the cost of raising the probability of voting "yes" from ${}^{(h-1)}\!/\!{}_{(k+1)}$ to ${}^{h}\!/\!{}_{(k+1)}$. In order to do so, we must have reached an acceptance probability of ${}^{(h-1)}\!/\!{}_{(k+1)}$ first. Now, let the number of jobs to be scheduled be b_j. Note that one can take b_j bribery steps at the cost of d_j dollars if and only if one can schedule b_j jobs with a cost of d_j. Hence, we can decide whether or not (P, C_P, Z, B) is in B_1-C_2-PLP by using Lemma 1. ❑

4.2 Issue Bribery

A greedy strategy succeeds for proving:

Theorem 3. B_2-C_1-PLP *and* B_2-C_2-PLP *are in* P.

4.3 Probabilistic Lobbying with Issue Weighting

Table 2 summarizes our results for B_i-C_j-PLP-WIW, $i \in \{1,2,3\}$ and $j \in \{1,2\}$. The most interesting observation is that introducing issue weights raises the complexity from P to NP-completeness for all cases of microbribery and issue bribery by using KNAPSACK in the reduction (though it remains the same for voter bribery). Nonetheless, we show later as Theorem 6 that these NP-complete problems are fixed-parameter tractable.

Theorem 4. *For* $i, j \in \{1,2\}$, B_i-C_j-PLP-WIW *is* NP-*complete.*

5 Parameterized Complexity Results

5.1 Voter Bribery

Christian et al. [5] proved that the following problem is W[2]-complete. We state this problem here as is common in parameterized complexity:

Name: OPTIMAL LOBBYING (OL, for short).

Given: An $m \times n$ matrix E and a $0/1$ vector Z of length n. Each row of E represents a voter. Each column represents an issue in the election. The vector Z represents The Lobby's target outcome.

Parameter: A positive integer k (representing the number of voters to be influenced).

Question: Is there a choice of k rows of the matrix (i.e., of k voters) that can be changed such that in each column of the resulting matrix (i.e., for each issue) a majority vote yields the outcome targeted by The Lobby?

Christian et al. [5] proved this problem to be W[2]-complete by a reduction from k-DOMINATING SET to OL (showing the lower bound) and from OL to INDEPENDENT-k-DOMINATING SET (showing the upper bound). To employ the W[2]-hardness result of Christian et al. [5], we show that OL is a special case of B_3-C_1-PLP and thus (parameterized) polynomial-time reduces to B_3-C_1-PLP. The "Turing" approach suggested by Cesati [4] shows membership in W[2]. Analogous arguments apply to B_3-C_2-PLP.

Theorem 5. *For* $j \in \{1,2\}$, B_3-C_j-PLP *(parameterized by the budget) is* W[2]-*complete.*

5.2 Probabilistic Lobbying with Issue Weighting

Recall from Theorem 4 that B_i-C_j-PLP-WIW, where $i, j \in \{1,2\}$, is NP-hard. Theorem 6 says that each of these problems is fixed-parameter tractable when parameterized by the budget, using KNAPSACK again.

Theorem 6. *For* $i, j \in \{1,2\}$, B_i-C_j-PLP-WIW *(parameterized by the budget) is in* FPT.

Voter bribery with issue weighting remains W[2]-complete for both evaluation criteria; the membership proof is somewhat more involved than the one in the unweighted case.

Theorem 7. *For* $j \in \{1,2\}$, B_3-C_j-PLP-WIW *(parameterized by the budget) is* W[2]-*complete.*

6 Approximability

As seen in Tables 1 and 2, many problem variants of probabilistic lobbying are NP-complete. Hence, it is interesting to study them not only from the viewpoint of parameterized complexity, but also from the viewpoint of approximability.

The budget constraint on the bribery problems studied so far gives rise to natural minimization problems: Try to minimize the amount spent on bribing. For clarity, let us denote these minimization problems by prefixing the problem name with MIN, leading to, e.g., MIN-OL.

The already mentioned reduction of Christian et al. [5] (that proved that OL is W[2]-hard) is parameter-preserving (regarding the budget). It further has the property that a possible solution found in the OL instance can be re-interpreted as a solution to the DOMININATING SET instance the reduction started with, and the OL solution and the DOMININATING SET solution are of the same size. This in particular means that inapproximability results for DOMININATING SET transfer to inapproximability results for OL. Similar observations are true for the interrelation of SET COVER and DOMINATING SET, as well as for OL and B_3-C_1-PLP-WIW (or B_3-C_2-PLP-WIW).

The known inapproximability results [3,12] for SET COVER hence give the following result (see also Footnote 4 in [14]).

Theorem 8. *There is a constant $c > 0$ such that* MIN-OL *is not approximable within factor $c \cdot \log(n)$ unless* $\mathrm{NP} \subset \mathrm{DTIME}(n^{\log\log(n)})$*, where n denotes the number of issues.*

Since OL can be viewed as a special case of both B_3-C_i-PLP and B_3-C_i-PLP-WIW for $i \in \{1,2\}$, we have the following corollary.

Corollary 1. *For $i \in \{1,2\}$, there is a constant $c_i > 0$ such that both* MIN-B_3-C_i-PLP *and* MIN-B_3-C_i-PLP-WIW *are not approximable within factor $c_i \cdot \log(n)$ unless* $\mathrm{NP} \subset \mathrm{DTIME}(n^{\log\log(n)})$*, where n denotes the number of issues.*

A *cover number* $c(r_j)$ is associated with each issue r_j, indicating by how many levels voters must raise their acceptance probabilities in order to arrive at average majority for r_j. The cover numbers can be computed beforehand for a given instance. Then, we can also associate cover numbers to sets of issues (by summation), which finally leads to the cover number $N = \sum_{j=1}^{n} c(r_j)$ of the whole instance.

When we interpret an OL instance as a B_3-C_2-PLP instance, the cover number of that resulting instance equals the number of issues, assuming that the votes for all issues need amendment. Thus we have the following corollary:

Corollary 2. *There is a constant $c > 0$ such that* MIN-B_3-C_2-PLP *is not approximable within factor $c \cdot \log(N)$ unless* $\mathrm{NP} \subset \mathrm{DTIME}(N^{\log\log(N)})$*, where N is the cover number of the given instance. A fortiori, the same statement holds for* MIN-B_3-C_2-PLP-WIW.

Let H denote the harmonic sum function, i.e., $H(r) = \sum_{i=1}^{r} 1/i$. It is well known that $H(r) = O(\log(r))$. More precisely, it is known that

$$\lfloor \ln r \rfloor \leq H(r) \leq \lfloor \ln r \rfloor + 1.$$

We show the following theorem by providing and analyzing a greedy approximation algorithm.

Theorem 9. MIN-B_3-C_2-PLP *can be approximated within a factor of* $\ln(N) + 1$, *where* N *is the cover number of the given instance.*

In the strict-majority scenario, cover numbers would have a different meaning—we thus call them *strict cover numbers*: For each referendum, the corresponding strict cover number tells in advance how many voters have to change their opinions (bringing them individually over the given threshold t) to accept this referendum. The strict cover number of a problem instance is the sum of the strict cover numbers of all given issues.

Theorem 10. MIN-B_3-C_1-PLP *can be approximated within a factor of* $\ln(N) + 1$, *where* N *is the strict cover number of the given instance.*

Note that this result is in some sense stronger than Theorem 9 (which refers to the average-majority scenario), since the cover number of an instance could be larger than the strict cover number.

This approximation result is complemented by a corresponding hardness result.

Corollary 3. *There is a constant* $c > 0$ *such that* MIN-B_3-C_1-PLP *is not approximable within factor* $c \cdot \log(N)$ *unless* $\mathrm{NP} \subset \mathrm{DTIME}(N^{\log\log(N)})$, *where* N *is the strict cover number of the given instance. A fortiori, the same statement holds for* MIN-B_3-C_1-PLP-WIW.

Unfortunately, those greedy algorithms do not (immediately) transfer to the case when issue weights are allowed.

7 Conclusions

We have studied six lobbying scenarios in a probabilistic setting, both with and without issue weights. Among the twelve problems studied, we identified those that can be solved in polynomial time, those that are NP-complete yet fixed-parameter tractable, and those that are hard (namely, W[2]-complete) in terms of their parameterized complexity with suitable parameters. It would be interesting to study these problems in different parameterizations. Finally, we investigated the approximability of hard probabilistic lobbying problems (without issue weights) and obtained both approximation and inapproximability results. A number of related results can be found in the full version [8]. An interesting open question is whether one can find logarithmic-factor approximations for voter bribery with issue weights.

References

1. M. Baye, D. Kovenock, and C. de Vries. Rigging the lobbying process: An application of the all-pay auction. *The American Economic Review*, 83(1):289–294, 1993.
2. M. Baye, D. Kovenock, and C. de Vries. The all-pay auction with complete information. *Economic Theory*, 8(2):291–305, 1996.
3. M. Bellare, S. Goldwasser, C. Lund, and A. Russell. Efficient probabilistically checkable proofs and applications to approximations. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 294–304. ACM Press, 1993.

4. M. Cesati. The Turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67:654–685, 2003.
5. R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.
6. C. Crombez. Information, lobbying and the legislative process in the European Union. *European Union Politics*, 3(1):7–32, 2002.
7. R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.
8. G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. Technical Report arXiv:0906.4431 [cs.CC], ACM Computing Research Repository (CoRR), June 2009.
9. P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, chapter 14, pages 375–406. Springer, 2009.
10. C. Lindner and J. Rothe. Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In A. Holder, editor, *Mathematical Programming Glossary*. INFORMS Computing Society, Oct. 2008.
11. R. Myerson and R. Weber. A theory of voting equilibria. *The American Political Science Review*, 87(1):102–114, 1993.
12. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 475–484. ACM Press, 1997.
13. J. Reinganum. A formal theory of lobbying behaviour. *Optimal Control Applications and Methods*, 4(4):71–84, 1983.
14. T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 69–76. ACM Press, July 2002.
15. J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.