arXiv:0906.4431v1 [cs.CC] 24 Jun 2009

The Complexity of Probabilistic Lobbying *

Gábor Erdélyi† and Henning Fernau‡ and Judy Goldsmith§
and
Nicholas Mattei¶ and Daniel Raible‖ and Jörg Rothe**

June 24, 2009

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. Individual entities such as citizens or corporations are not supposed to have undue influence in the wording or passage of a law. However, they are allowed to make contributions to representatives, and it is common to include an indication that the contribution carries an expectation that the representative will vote a certain way on a particular issue.

*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. Work done in part while the first author was visiting Universität Trier, while the third author and the fifth author were visiting Heinrich-Heine-Universität Düsseldorf, and while the sixth author was visiting the University of Rochester.

†URL: ccc.cs.uni-duesseldorf.de/~erdelyi. Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, 40225 Düsseldorf, Germany.

‡URL: www.informatik.uni-trier.de/~fernau. Universität Trier, FB 4—Abteilung Informatik, 54286 Trier, Germany.

§URL: www.cs.uky.edu/~goldsmit. University of Kentucky, Dept. of Computer Science, Lexington, KY 40506, USA.

¶URL: www.cs.uky.edu/~nsmatt2. University of Kentucky, Dept. of Computer Science, Lexington, KY 40506, USA.

‖URL: www.informatik.uni-trier.de/~raible. Universität Trier, FB 4—Abteilung Informatik, 54286 Trier, Germany.

**URL: ccc.cs.uni-duesseldorf.de/~rothe. Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, 40225 Düsseldorf, Germany.

Many factors can affect a representative's vote on a particular issue. There are the representative's personal beliefs about the issue, which presumably were part of the reason that the constituency elected them. There are also the 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.

There are several axes along which we complicate the picture. The first is the notion of sufficiency: What does it mean to say we have donated enough to influence the vote? Does it mean that the probability that a single vote will go our way is greater than some threshold? That the probability that all the votes go our way is greater than that threshold? We discuss these and other criteria in the section on evaluation criteria.

How does one donate money to a campaign? In the United States there are several laws that influence how, when, and how much a particular person or organization can donate to a particular candidate. We examine ways in which money can be channeled into the political process in Section 2.[1]

Lobbying has been studied formally by economists, computer scientists, and special interest groups since at least 1983 [21] and as an extension to formal game theory since 1944 [19]. 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 [19]. This analysis is focused on learning how these complex systems work and deducing optimal strategies for winning the competitions [21,1,2]. 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 [1,21, 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 [21] 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 [14] 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

[1] We stress that when we use the term "bribery" in this paper, it is meant in the sense of lobbying [5], not in the sense Faliszewski et al. [9] define bribery (see also, e.g., [10,13,11]).

Lobbying" (OL) is complete for the (parameterized) complexity class W[2]. The OL problem is a deterministic and nonweighted version of the problems that we present in this paper. Erdélyi et al. [8] extend "Optimal Lobbying" into "Optimal Weighted Lobbying" (OWL) by allowing different voters to have different prices. This in turn can be seen as a special case of a "binary multi-unit combinatorial reverse auction winner-determination problem," see Sandholm et al. [22].

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.[2]

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 $\vec{X} = (x_1, x_2, \ldots, x_n)$, 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

[2] A similar model was first discussed by Reinganum [21] in the continuous case and we translate it here to the discrete case. This will allow us to present algorithms for, and the complexity analysis of, the problem.

interval $[0,1]$ into $k+1$ steps of size $1/(k+1)$ each.[3] For each $i \in \{1,2,\ldots,m\}$ and $j \in \{1,2,\ldots,n\}$, $c_{i,j} : \{0, 1/(k+1), 2/(k+1), \ldots, k/(k+1), 1\} \to \mathbb{N}$ 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 $0 \leq \ell \leq k+1$. 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 $p_{i,j} \leq a \leq b$ or $p_{i,j} \geq a \geq b$, then $c_{i,j}(a) \leq c_{i,j}(b)$. 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 $0 \leq \ell \leq k+1$. 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 $\vec{Z} \in \{0,1\}^n$, 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 $\vec{Z} = 1^n$. This assumption can be made without loss of generality, since if there is a zero in $\vec{Z}$ at position j, we can flip this zero to one and also change the corresponding probabilities $p_{1,j}, p_{2,j}, \ldots, p_{m,j}$ in the jth column of P to $1-p_{1,j}, 1-p_{2,j}, \ldots, 1-p_{m,j}$ (see the evaluation criteria in Section 2.3 for how to determine the result of voting on a referendum).

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 $\vec{Z} = 1^3$, 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

[3]There is some arbitrariness in this choice of k. One might think of more flexible ways of partitioning $[0,1]$. We have chosen this way for the sake of simplifying the representation, but we mention that all that matters is that for each i and j, the discrete price function $c_{i,j}$ is defined on the value $p_{i,j}$, and is set to zero for this value.

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

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 [18]).

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 $\vec{Z} \in \{0,1\}^n$, 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 $\vec{Z}$?

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.

2.2.1 Microbribery (B_1)

The first method at the disposal of The Lobby is what we will call *microbribery*.[5] 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.

[5] Although our notion was inspired by theirs, we stress that it should not be confused with the term "microbribery" used by Faliszewski et al. [10,13,11] in the different context of bribing "irrational" voters in Llull/Copeland elections via flipping single entries in their preference tables.

2.2.2 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.

2.2.3 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.

2.3.1 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 $\vec{X} = (0,0,0)$, because none of the issues has a strict majority of voters with above 50% likelihood of voting according to the agenda.

2.3.2 Average Majority (C_2)

For each issue, r_j, of a given probability matrix P, we define: $\overline{p_j} = \left(\Sigma_{i=1}^{m} p_{i,j}\right)/m$. 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 $\vec{X} = (1,0,0)$.

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 $\vec{W} \in \mathbb{Z}^n$ 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 $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ with table C_P of price functions and a lobby target vector $\vec{Z} \in \{0,1\}^n$, a lobby weight vector $\vec{W} \in \mathbb{Z}^n$, 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 $\vec{Z}$ 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.

The $\mathscr{O}^*(\cdot)$ notation has by now become standard in exact algorithms. It neglects not only constants (as the more familiar $\mathscr{O}(\cdot)$-notation does) but also polynomial factors in the function estimates. Thus, a problem is in FPT if and only if an instance (with parameter k) can be solved in time $\mathscr{O}^*(f(k))$ for some computable function f.

There is also a theory of parameterized hardness, 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. Since only the second level, W[2], will be of interest to us in this paper, we will define only this class below.

A *parameterized reduction* is a function r that, for some polynomial p and some function g, is computable in time $\mathcal{O}(g(k)p(|I|))$ and maps an instance (I,k) of $\mathscr{P}$ onto an instance $r(I,k) = (I',k')$ of $\mathscr{P}'$ such that (I,k) is a YES instance of $\mathscr{P}$ if and only if (I',k') is a YES instance of $\mathscr{P}'$ and $k' \leq g(k)$. We then say that $\mathscr{P}$ *reduces to* $\mathscr{P}'$.

W[2] can be characterized by the following problem on Turing machines:

Name: SHORT NONDETERMINISTIC TURING MACHINE COMPUTATION (SMNTMC, for short).

Given: A multi-tape nondeterministic Turing machine M (with two-way infinite tapes) and an input string x (both M and x are given in some standard encoding).

Parameter: A positive integer k.

Question: Is there an accepting computation of M on input x that reaches a final accepting state in at most k steps?

More specifically, a parameterized problem $\mathscr{P}$ is in W[2] if and only if it can be reduced to SMNTMC via a parameterized reduction, see Cesati [4]. This can be accomplished by giving an appropriate multi-tape nondeterministic Turing machine for solving $\mathscr{P}$. Hardness can be shown by giving a parameterized reduction in the opposite direction, from SMNTMC to $\mathscr{P}$.

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., [17].)

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

Theorem 2 B_1-C_1-PLP *is in* P.

Proof. The aim is to win all referenda. Per referendum r and voter v, we can compute in polynomial time the amount $b(r,v)$ The Lobby has to spend to turn the favor of v in the direction of

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

Table 1: Complexity results for B_i-C_j-PLP

The Lobby (beyond the given threshold t). In particular, set $b(r,v)=0$ if voter v would already vote according to the agenda of The Lobby. For each issue r, sort $\{b(r,v) \mid 1 \leq v \leq m\}$ increasingly, yielding the sequence $b_1(r), \ldots, b_n(r)$. To win referendum r, The Lobby must spend at least $B(r) = \sum_{i \leq (m+1)/2} b_i(r)$ dollars. Hence, all referenda can be won if and only if $\sum_{r=1}^{n} B(r)$ is at most the given bribery budget B. ❑

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 $J_{i(1)}, \ldots, J_{i(q)}$ 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 $P_i = J_{i,1}, \ldots, J_{i,p(i)}$ 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 $V = \{J_1, \ldots, J_n\}$ of jobs, a directed graph $G = (V,A)$ consisting of pairwise disjoint paths $P_1, \ldots, P_z$, two numbers $C, q \in \mathbb{N}$, and a cost function $c : V \rightarrow \mathbb{N}$.

Question: Can we find a schedule $J_{i(1)}, \ldots, J_{i(q)}$ of q jobs of cost at most C respecting the precedence constraints of G?

We first show that PATH SCHEDULE is in P. 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.

Lemma 3 PATH SCHEDULE *is in* P.

Proof. The following dynamic programming approach finally calculates $T[\{P_1, \ldots, P_z\}, q]$, which gives the minimum cost to solve the problem. We build up a table $T[\{P_1, \ldots, P_\ell\}, j]$ storing the minimum cost of scheduling j jobs out of the jobs contained in the paths $P_1, \ldots, P_\ell$. Let $P_i = J_{i,1}, \ldots, J_{i,p(i)}$. Clearly, for $k \leq p(1)$, $T[\{P_1\}, k] = \sum_{s=1}^{k} J_{1,s}$. For $k > p(1)$, set $T[\{P_1\}, k] = \infty$. If $\ell > 1$, $T[\{P_1, \ldots, P_\ell\}, j]$ equals $\min_{0 \leq k \leq \min\{j, p(\ell)\}} T[\{P_1, \ldots, P_{\ell-1}\}, j-k] + \sum_{s=1}^{k} c(J_{\ell,s})$. Consider each possible scheduling of the first k jobs of P_ℓ. For the remaining $j-k$ jobs, look up a solution in the table. Notice that we can re-order each schedule S so that all jobs from one path contiguously appear in S, without violating the precedence constraints by this re-ordering, nor changing the cost of the schedule. Hence, $T[\{P_1, \ldots, P_z\}, q]$ gives the minimum schedule cost. The number of entries in the table is $z \cdot q$, and computing each entry $T[\{P_1, \ldots, P_\ell\}, \cdot]$ is linear in $p(\ell)$ (for each $1 \leq \ell \leq z$), which leads to a run time of the dynamic programming algorithm that is polynomially bounded by the input size. ❑

Theorem 4 B_1-C_2-PLP *is in* P.

Proof. Let $(P, C_P, \vec{Z}, B)$ be a given B_1-C_2-PLP instance, where $P \in \mathbb{Q}_{[0,1]}^{m \times n}$, C_P is a table of price functions, $\vec{Z} \in \{0,1\}^n$ 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 $\vec{Z}$. If $\sum_{j=1}^{n} d_j \leq B$ then The Lobby can achieve its goal that the votes on all issues equal $\vec{Z}$. 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 $(P, C_P, \vec{Z}, B)$ is in B_1-C_2-PLP by using the dynamic program given in the proof of Lemma 3. ❑

Exact Version of Microbribery: The exact variants of probabilistic lobbying via microbribery, denoted by Exact-B_1-C_j-PLP with $j \in \{1,2\}$, ask whether The Lobby can achieve its goal via microbribery (for the given evaluation criterion) by spending exactly B dollars. Thus, in these variants The Lobby is constrained to spending a total amount of exactly B dollars (and no less than that).

Theorem 5 *For* $j \in \{1,2\}$, Exact-B_1-C_j-PLP *is* NP-*complete.*

Proof. We focus on Exact-B_1-C_1-PLP and note that the reduction can be carried over straightforwardly to the case of Exact-B_1-C_2-PLP.

To see membership in NP, observe that an instance I of Exact-B_1-C_j-PLP can be transformed into another instance I' of Exact-B_1-C_j-PLP after guessing which issue and which voter should be bribed by an amount of money specified by the first non-zero entry in the corresponding row of the cost matrix C_P and modifying C_P accordingly. After repeated guesses, we either arrive at an amount of left-over money that cannot be used for any bribery anymore (since it is either too small or possibly no money can be spent at all, since all issues and voters have been "completely bribed"); in this case, the nondeterministic procedure simply stops. Or, all money was (exactly) spent. In that case, it is checked if the evaluation criterion was met. If it is met, the algorithm gives and affirmative answer; otherwise, it rejects. Hence, the nondeterministic procedure will either succeed

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

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

in one branch, yielding an affirmative answer, or there is no solution, and no affirmative answer will be produced.

To show NP-hardness of EXACT-B_1-C_1-PLP, we provide a reduction from SUBSET SUM (see, e.g., Garey and Johnson [15]): Given $a_1, \dots, a_n, S \in \mathbb{N}^+$, does there exist a subset $I \subseteq \{1, \dots, n\}$ such that $\sum_{i \in I} a_i = S$? For such a SUBSET SUM instance, create an EXACT-B_1-C_1-PLP instance with only one referendum r_1 and with voters $v_1, \dots, v_{2n}$. Let $k = 0$. Set $P_{i1} = 1$ and $c_{i1}(1) = 0$ for $1 \leq i \leq n$, and set $P_{i1} = 0$, $c_{i1}(0) = 0$, $c_{i1}(1) = a_{i-n}$ for $n+1 \leq i \leq 2n$. Let $B = S$ and $t = 0.5$. Observe that we have to influence at least one of the voters not in accordance with r_1. Thus we can turn r_1 to The Lobby's favor by spending exactly B dollars on a set of voters $v_{i_1}, \dots, v_{i_\ell}$ ($n+1 \leq i_j \leq 2n$) if and only if there is a subset $I \subseteq \{1, \dots, n\}$, $|I| = \ell$, such that $\sum_{i \in I} a_i = S$. ❑

4.2 Issue Bribery

Theorem 6 B_2-C_1-PLP *and* B_2-C_2-PLP *are in* P.

Proof. We prove that B_2-C_1-PLP is in P; the proof for B_2-C_2-PLP is analogous. Observe that B_i-C_j-PLP (just like the problem OL defined in Section 5) contains a vector that represents the issues that The Lobby would like to see passed. In Theorem 8 below (see Section 5), the constraint from OL, b, is expressed over the number of *voters* that need to be influenced. In B_2-C_1-PLP, however, we are required to influence a certain number of issues. Thus, in order to determine a win we construct a *cost difference* matrix representing how much it would cost The Lobby to win each issue (since all voters receive the same amount of money, this can be determined in time polynomial in the number of voters and issues). Using this cost difference matrix, we greedily select the cheapest issues to influence in order to achieve The Lobby's agenda. ❑

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 (though it remains the same for voter bribery). Nonetheless, we show later as Theorem 10 that these NP-complete problems are fixed-parameter tractable.

Theorem 7 *For* $i, j \in \{1,2\}$, B_i-C_j-PLP-WIW *is* NP-*complete.*

Proof. Membership in NP is easy to observe for each problem B_i-C_j-PLP-WIW, where $i, j \in \{1,2\}$.

To prove that B_1-C_1-PLP-WIW is NP-hard, we give a reduction from KNAPSACK to B_1-C_1-PLP-WIW. In KNAPSACK, we are given a set of objects $U = \{o_1, \ldots, o_n\}$ with weights $w : U \to \mathbb{N}$ and profits $p : U \to \mathbb{N}$, and $W, P \in \mathbb{N}$. The question is whether there is a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} w(o_i) \leq W$ and $\sum_{i \in I} p(o_i) \geq P$. Given a KNAPSACK instance (U, w, p, W, P), create a B_1-C_1-PLP-WIW instance with $k = 0$ and only one voter, v_1, where for each issue, v_1's acceptance probability is either zero or one. For each object $o_j \in U$, create an issue r_j such that the acceptance probability of v_1 is zero. Let the cost of raising this probability on r_j be $c_{1,j} = w(o_j)$ and let the weight of issue r_j be $w_j = p(o_j)$. Let The Lobby's budget be W and its objective value be $O = P$. By construction, there is a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} w(o_i) \leq W$ and $\sum_{i \in I} p(o_i) \geq P$ if and only if there is a subset $I \subseteq \{1, \ldots, n\}$ with $\sum_{i \in I} c_{1,i} \leq W$ and $\sum_{i \in I} w_i \geq O$.

As the reduction introduces only one voter, there is no difference between the bribery methods B_1 and B_2, and no difference either between the evaluation criteria C_1 and C_2. Hence, the above reduction works for all four other problems. ❑

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 $\vec{Z}$ of length n. Each row of E represents a voter. Each column represents an issue in the election. The vector $\vec{Z}$ 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. Analogous arguments apply to B_3-C_2-PLP.

Theorem 8 *For $j \in \{1,2\}$, B_3-C_j-PLP (parameterized by the budget) is W[2]-hard.*

Proof. We first prove that B_3-C_1-PLP is W[2]-hard by providing a parameterized reduction from OL to B_3-C_1-PLP, where the parameter (number of voters to be influenced by The Lobby) is the same in both problems. We are given an instance $(E, \vec{Z}, b)$ of OL, where E is a $m \times n$ $0/1$ matrix, b is the number of votes to be edited, and $\vec{Z}$ is the agenda for The Lobby. We may assume without loss of generality that $\vec{Z} = 1^n$.

We construct an instance of B_3-C_1-PLP consisting of the given matrix $P = E$ (a "degenerated" probability matrix with only the probabilities 0 and 1), a corresponding cost matrix C_P, a target vector $\vec{Z} = 1^n$, and a budget B. C_P has two columns (i.e., we have $k = 0$, since the problem instance is deterministic, see Section 2.1), one column for probability 0 and one for probability 1. All entries of C_P are set to unit cost.

The cost of increasing any value in P is n, since donations are distributed evenly across issues for a given voter. We want to know whether there is a set of bribes of cost at most $b \cdot n = B$ such that The Lobby's agenda passes. This holds if and only if there are b voters that can be bribed so that they vote uniformly according to The Lobby's agenda and that is sufficient to pass all the issues. Thus, the given instance $(E, \vec{Z}, b)$ is in OL if and only if the constructed instance $(P, C_P, \vec{Z}, B)$ is in B_3-C_1-PLP, which shows that OL is a polynomial-time recognizable special case of B_3-C_1-PLP, and thus B_3-C_1-PLP is W[2]-hard.

Note that for the construction above it doesn't matter whether we use the strict-majority criterion (C_1) or the average-majority criterion (C_2). Since the entries of P are 0 or 1, we have $\overline{p_j} > 0.5$ if and only if we have a strict majority of ones in the jth column. Thus, B_3-C_2-PLP is W[2]-hard too. ❑

Theorem 9 *For* $j \in \{1,2\}$, B_3-C_j-PLP *(parameterized by the budget) is* W[2]*-complete.*

Proof. Again, we give only the details for the case of B_3-C_1-PLP; the proof for B_3-C_2-PLP is analogous. W[2]-hardness has been shown in Theorem 8. To show membership in W[2], we reduce B_3-C_1-PLP to SMNTMC, which was defined in Section 3. To this end, it suffices to describe how a nondeterministic multi-tape Turing machine can solve such a lobbying problem.

Consider an instance of B_3-C_1-PLP: a probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ with a table C_P of price functions and a budget B. Again, we may assume that the target vector is $\vec{Z} = 1^n$. Moreover, we assume that the target threshold t is fixed. We can identify t with a certain step level for the price functions.

The reducing machine works as follows. From P, C_P, and t, the machine extracts the information $H_{i,j}(d)$, where $H_{i,j}(d)$ is true if either $p_{i,j} \geq t$ or $c_{i,j}(t) \leq d/n$ (since according to this scenario, the bribery money is distributed among all issues). Note that $H_{i,j}(d)$ captures whether paying d dollars to voter v_i helps to raise the acceptance probability of v_i on referendum r_j above the threshold t. Moreover, for each referendum r_j, we compute the minimum number of voters that need to switch their opinion so that majority is reached for that specific referendum; let $s(j)$ denote this threshold for r_j. Since we assume payments in dollar units, a referendum with $s(j) > B$ yields a NO instance. We can therefore replace any value $s(j) > B$ by the value $B+1$.

The nondeterministic multi-tape Turing machine M we describe next has, in particular, access to $H_{i,j}$ and to $s(j)$. M has $n+1$ working tapes T_j, $0 \leq j \leq n$, all except one of which correspond to issues r_j, $1 \leq j \leq n$. We will use the set of voters, $V = \{v_1, \dots, v_m\}$, as alphabet. The (formal) input tape of M is ignored.

M starts by writing $s(j)$ symbols # onto tape j for each j, $1 \leq j \leq n$. By using parallel writing steps, this needs at most $B+1$ steps, since $s(j) \leq B+1$ as argued above.

Second, for each $i \in \{1, \dots, m\}$, M writes k_i symbols v_i from the alphabet V on the zeroth tape, T_0, such that $\sum_{i=1}^{m} k_i \leq B$. This is the nondeterministic guessing phase where the amount of bribery money spent on each voter is determined. Notice that no more than B voters can be bribed.

In the third phase, for each voter v_i that will be bribed, M counts the corresponding amount k_i of bribery money and determines (by using $H_{i,j}$) if it is enough to change v_i's opinion regarding the jth issue. If so, the head of M on tape j moves one step to the left. Again, all these head moves are performed in parallel. Hence, the string on the zeroth tape is being processed in at most B (parallel) steps.

Finally, it is checked if the left border is reached (again) for all tapes T_j, $j > 0$. This is the case if and only if the guessed bribery was successful. ❑

5.2 Probabilistic Lobbying with Issue Weighting

Recall from Theorem 7 that B_i-C_j-PLP-WIW, where $i, j \in \{1,2\}$, is NP-hard. We now show that each of these problems is fixed-parameter tractable when parameterized by the budget.

Theorem 10 *For $i, j \in \{1,2\}$,* B_i-C_j-PLP-WIW *(parameterized by the budget) is in* FPT.

Proof. Since the four unweighted variants are in P, we can compute the number of dollars to be spent to win referendum r_j in polynomial time in each case. Now re-interpret the given B_i-C_j-PLP-WIW instance as a KNAPSACK instance: Every issue r_j is an object o_j with weight d_j and profit p_j, both set to be the same as weight w_j of issue r_j. Let the KNAPSACK bound be the total number B of dollars allowed to be spent. Now use the pseudo-polynomial algorithm to solve KNAPSACK in time $\mathcal{O}(n2^{|B|})$, where $|B|$ denotes the length of the encoding of B. ❑

Voter bribery with issue weighting remains W[2]-complete for both evaluation criteria.

Theorem 11 *For $j \in \{1,2\}$,* B_3-C_j-PLP-WIW *(parameterized by the budget) is* W[2]*-complete.*

Proof. By Theorem 8, B_3-C_1-PLP is W[2]-hard. Since B_3-C_1-PLP is a special case of B_3-C_1-PLP-WIW, where all the issues have unit weight, B_3-C_1-PLP-WIW is W[2]-hard as well. An analogous argument shows that B_3-C_2-PLP-WIW is W[2]-hard, too.

Membership in W[2] is a bit more tricky than in the unweighted case from Theorem 9. In the following, we indicate only the necessary modifications:

- The reducing machine calculates the difference O' between the target weight and the sum of the weights of the referenda that are already won.
- For each referendum that is not already won, the reducing machine introduces a special letter r_i to be used on the zeroth tape.

- The Turing machine that has been built at the very beginning also guesses at most B referenda that (additionally) should be won. (Note that influencing any issue costs at least one dollar.) Then, the Turing machine will spend $\mathcal{O}(f(B))$ time to calculate if winning those guessed referenda $r_{i1},\dots,r_{ib}$, $b \leq B$, would be sufficient to get beyond the threshold. Only if sufficiency is guaranteed, the Turing machine continues working.

- The Turing machine will then continue to work as described in the proof of Theorem 9.

- At the very end, the Turing machine will verify in at most B steps if all referenda guessed in the very beginning have been won.

Note that it is quite tempting to try to avoid the weight calculations within the Turing machine, letting the reducing machine do this job. However, this seems to necessitate coding the winning situations in the state set of the Turing machine, leading to a possible exponential size of this Turing machine (measured in the overall input size of the voting scenario).

W[2]-completeness of B_3-C_2-PLP-WIW can be proven by an analogous argument. ❑

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.

6.1 Voter Bribery is Hard to Approximate

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 [3,20] for SET COVER hence give the following result (see also Footnote 4 in [22]).

Theorem 12 *There is a constant $c > 0$ such that* MIN-OL *is not approximable within factor $c \cdot \log(n)$ unless* NP $\subset$ 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 13 *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* NP $\subset$ DTIME$(n^{\log\log(n)})$, *where n denotes the number of issues.*

Conversely, a logarithmic-factor approximation can be given for the minimum-budget versions of all our problems, as we will show now. We first discuss the relation to the well-known SET COVER problem, sketching a tempting, yet flawed reduction and pointing out its pitfalls. Avoiding these pitfalls, we then give an approximation algorithm for MIN-B$_3$-C$_2$-PLP. Moreover, we define the notion of cover number, which allows to state inapproximability results for MIN-B$_3$-C$_2$-PLP. Similar results hold for MIN-B$_3$-C$_1$-PLP, the constructions being sketched at the end of the section.

Voter bribery problems are closely related to set cover problems, in particular in the average-majority scenario, so that we should be able to carry over approximability ideas from that area. The intuitive translation of a MIN-B$_3$-C$_2$-PLP instance into a SET COVER instance is as follows: The universe of the derived SET COVER instance should be the set of issues, and the sets (in the SET COVER instance) are formed by considering the sets of issues that could be influenced (by changing a voter's opinion) through bribery of a specific voter. Namely, when we pay voter v a specific amount of money, say d dollars, he or she will invest d/n dollars to each issue and possibly change v's opinion (or at least raise v's acceptance probability to the "next level"). The weights associated to the sets of issues correspond to the bribery costs that are (minimally) incurred to lift the issues in the set to some "next level." There are four differences to classical set covering problems:

1. We cannot neglect the voter who has been bribed, so different voters (with different bribing costs) may be associated with the same set of issues.

2. The sets associated with one voter are not independent. For each voter, the sets of issues that can be influenced by bribing that voter are linearly ordered by set inclusion. Moreover, when bribing a specific voter, we have to first influence the "smaller sets" (which might be expensive) before possibly influencing the "larger ones"; so, weights are attached to set differences, rather than to sets.

3. 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.

4. The money paid "per issue" might not have been sufficient for influencing a certain issue up to a certain level, but it is not "lost"; rather, it would make the next bribery step cheaper, hence (again) changing weights in the set cover interpretation.

To understand these connections better, let us have another look at our running example (under the voter bribery with average-majority evaluation), assuming an all-ones target vector. If we paid 30 dollars to voter v_1, he or she would invest 10 dollars to each issue, which would raise his or her acceptance probability for the second issue from .3 to .4; no other issue level is changed. Hence, this would correspond to a set containing only r_2 with weight 30. Note that by this bribery, the costs for raising the acceptance probability of voter v_1 to the next level would be lowered for the other two issues. For example, spending 15 more dollars on v_1 would raise r_3 from .5 to .6, since all in all 45 dollars have been spent on voter v_1, which means 15 dollars per issue. If the threshold is 60% in that example, then the first issue is already accepted (as desired by The Lobby), but the second issue has gone up from .5 to .6 on average, which means that we have to raise either the acceptance probability of one voter by two levels (for example, by paying 210 dollars to voter v_1), or we have to raise the acceptance probability of each voter by one level (by paying 30 dollars to voter v_1 and another 30 dollars to voter v_2). This can be expressed by saying that the first issue has a cover number of zero, and the second has a cover number of two.

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 14 *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 now show the following theorem.

Theorem 15 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.*

Proof. Consider the following greedy algorithm (given threshold t and assuming (w.l.o.g.) the target vector $\vec{1}$); notice that the cover numbers (per referendum) can be computed from the cost matrix C_P and the threshold t before calling the algorithm the very first time:

Greedy Voter Bribery (GVB):
Input: A probability matrix $P \in \mathbb{Q}_{[0,1]}^{m \times n}$ (implicitly specifying a set V of m voters and a set R of n referenda), a cost matrix C_P, and n cover numbers $c(r_1), \ldots, c(r_n) \in \mathbb{N}$.

1. Delete referenda that are already won (indicated by $c(r_j) = 0$), and modify R and C_P accordingly.

2. If $R = \emptyset$, STOP.

3. For each voter v, compute the cheapest amount of money, d_v, that allows to raise any level in C_P. Let n_v be the number of referenda whose levels are raised when spending d_v dollars on voter v.

4. Bribe voter v such that d_v/n_v is minimum.

5. Modify C_P by subtracting d_v/n from each amount listed for voter v.

6. Modify c by subtracting one from $c(r)$ for those referenda $r \in R$ influenced by this bribery action.

7. Recurse.

Observe that our greedy algorithm influences voters only via raising their acceptance probabilities by only one level, so that the amount d_v possibly spent on voter v in Step 3 of the algorithm actually correponds to a set of referenda; we do not have to consider multiplicities of issues (raised over several levels) here.

Let $S_1, \ldots, S_\ell$ be the sequence of sets of referenda picked by the greedy bribery algorithm, along with the sequence $v_1, \ldots, v_\ell$ of voters and the sequence $d_1, \ldots, d_\ell$ of bribery dollars spent this way. Let $R_1 = R, \ldots, R_\ell, R_{\ell+1} = \emptyset$ be the corresponding sequence of sets of referenda, with the accordingly modified cover numbers c_i. Let $j(r,k)$ denote the index of the set in the sequence influencing referendum r the kth time with $k \le c(r)$, i.e., $r \in S_{j(r,k)}$ and $|\{i < j(r) \mid r \in S_i\}| = k-1$. To cover r the kth time, we have to pay $\chi(r,k) = d_{j(r,k)}/|S_{j(r,k)}|$ dollars. The greedy algorithm will incur a cost of $\chi_{greedy} = \sum_{r \in R} \sum_{k=1}^{c(r)} \chi(r,k)$ in total.

An alternative view on the greedy algorithm is from the perspective of the referenda: By running the algorithm, we implicitly define a sequence $r_1, \ldots, r_N$ of referenda, where $N = c(R) = \sum_{r \in R} c(r)$ is the cover number of the original instance, such that $S_1 = \{r_1, \ldots, r_{|S_1|}\}$, $S_2 = \{r_{|S_1|+1}, \ldots, r_{|S_1|+|S_2|}\}$, etc. Ties (how to list elements within S_i) are broken arbitrarily. This (implicitly) defines two functions $L, R : \{1, \ldots, \ell\} \to \{1, \ldots, N\}$ such that $S_i = \{r_{L(i)} \cdots r_{R(i)}\}$. Slightly abusing notation, we can associate a cost $\chi'(r_i)$ to each element in the sequence (keeping in mind the multiplicities of covering implied by the sequence), so that $\chi_{greedy} = \sum_{i=1}^{N} \chi'(r_i)$. Notice that $d_i = \sum_{L(i) \le r \le R(i)} \chi'(x_r)$.

Consider r_j with $L(i) \le j \le R(i)$. The current referenda set R_i has cover number $N - L(i) + 1$, i.e., of at least $N - j + 1$. Let χ_{opt} be the cost of an optimum bribery strategy $\mathscr{C}^*$ of the original universe. Of course, this also yields a cover of the referenda set R_i with cost at most χ_{opt}. The average cost per element (taking into account multiplicities as given by the cover numbers) is $\chi_{opt}/c(R_i)$. (So, whether or not some new levels are obtained through bribery does not really matter here, as long as the threshold is not exceeded.)

$\mathscr{C}^*$ can be described by a sequence of sets of referenda $C_1, \ldots, C_q$, with corresponding voters $z_1, \ldots, z_q$ and dollars $d_1^*, \ldots, d_q^*$ spent. Hence, $\chi_{opt} = \sum_{\kappa=1}^{q} d_\kappa^*$. To each bribery step we associate the cost factor $d_\kappa^*/|C_\kappa|$, for each issue r contained in C_κ. $\mathscr{C}^*$ could be also viewed as a bribery strategy for R_i. By pigeon hole, there is a referendum r in R_i (to be influenced the kth time) with cost factor at most $d_\kappa^*/|C_\kappa \cap R_i| \le \chi_{opt}/c(R_i)$, where κ is the index such that C_κ contains r for the kth time in $\mathscr{C}^*$ (usually, the cost would be smaller, since part of the bribery has already been paid before). Since (S_i, v_i) was picked to minimize $d_i/|S_i|$, we find $d_i/|S_i| \le d_\kappa^*/|C_\kappa \cap R_i| \le \chi_{opt}/c(R_i)$.

We conclude that

$$\chi'(r_j) \le \chi_{opt}/c(R_i) = \chi_{opt}/N-L(i)+1 \le \chi_{opt}/N-j+1.$$

Hence, $\chi_{greedy} = \sum_{j=1}^{N} \chi'(r_j) \le \sum_{j=1}^{N} \chi_{opt}/N-j+1 = H(N)\chi_{opt} \le (\ln(N)+1)\chi_{opt}$. ❑

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. Again, the strict cover number of a problem instance is the sum of the strict cover numbers of all given referenda.

The corresponding greedy algorithm would therefore choose to influence voter v_i (with d_i dollars) in the ith loop so that v_i changes his or her opinion on some referendum r_j (possibly, there is a whole set ρ_j of referenda influenced this way), so that $d_i/|\rho_j|$ is minimized.

We can now read the approximation estimate proof given for the average-majority scenario nearly literally as before, by re-interpreting the formulation "influencing referendum r" meaning now a complete change of opinion for a certain voter (not just gaining one level somehow). This establishes the following result.

Theorem 16 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 15 (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 17 *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. These weights might also influence the quality of approximation, but a simplistic greedy algorithm might result in covering the "wrong" arguments. Also, the proof of the approximation factor given above will not carry over, since we need as one of the proof's basic ingredients that an optimum solution can be interpreted as a partial one at some point. Those problems tend to have a different flavor.

6.2 Polynomial-Time Approximation Schemes

Those problems for which we obtained FPT results in the case of issue weights actually enjoy a polynomial-time approximation scheme (PTAS). The proof of Theorem 10 can be easily turned into a PTAS using standard techniques, since that result was obtained by transferring pseudo-polynomial time algorithms.

Theorem 18 *For $i, j \in \{1,2\}$,* MIN-B$_i$-C$_j$-PLP-WIW *admits a PTAS.*

The exact version of microbribery also admitted an FPT-result, but this cannot be interpreted as an approximation result, since the entity that should be minimized has to be hit exactly(otherwise, we have polynomial time).

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. 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, in: 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, in: 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, (ACM Press, 1993), pp. 294–304.

[4] M. Cesati, The Turing Way to Parameterized Complexity, in: 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, in: Review of Economic Design **11**(3), 217–224 (2007).

[6] C. Crombez, Information, Lobbying and the Legislative Process in the European Union, in: 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, L. Hemaspaandra, J. Rothe, and H. Spakowski, On Approximating Optimal Weighted Lobbying, and Frequency of Correctness versus Average-Case Polynomial Time, in Proceedings of the 16th International Symposium on Fundamentals of Computation Theory, (Springer-Verlag *Lecture Notes in Computer Science #4639*, August 2007), pp. 300–311.

[9] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra, The complexity of bribery in elections, in: Proceedings of the 21st National Conference on Artificial Intelligence, (AAAI Press, July 2006), pp. 641–646.

[10] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, Llull and Copeland voting broadly resist bribery and control, in: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, (AAAI Press, July 2007), pp. 724–730.

[11] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, Llull and Copeland voting computationally resist bribery and constructive control, Journal of Artificial Intelligence Research. To appear. Full version available as [12].

[12] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, Llull and Copeland Voting Computationally Resist Bribery and Control, Tech. Rep. arXiv:0809.4484v2 [cs.GT], ACM Computing Research Repository (CoRR), September 2008.

[13] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, Copeland voting fully resists constructive control, in: Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management, (Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008), pp. 165–176.

[14] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, A richer understanding of the complexity of election systems, in: Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz, edited by S. Ravi and S. Shukla, (Springer, 2009), chap. 14, pp. 375–406.

[15] M. Garey and D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, (W. H. Freeman and Company, New York, 1979).

[16] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems, (Springer-Verlag, Berlin, Heidelberg, New York, 2004).

[17] C. Lindner and J. Rothe, Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice, in: Mathematical Programming Glossary, edited by A. Holder, (INFORMS Computing Society, October 2008).

[18] R. Myerson and R. Weber, A Theory of Voting Equilibria, The American Political Science Review **87**(1), 102–114 (1993).

[19] J. von Neumann and O. Morgenstern, Theory of Games and Economic Behavior, (Princeton University Press, 1944).

[20] 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, (ACM Press, 1997), pp. 475–484.

[21] J. Reinganum, A Formal Theory of Lobbying Behaviour, in: Optimal Control Applications and Methods **4**(4), 71–84 (1983).

[22] 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, (ACM Press, July 2002), pp. 69–76.