

# A Unifying Framework for Manipulation Problems

## Extended Abstract

Dušan Knop\*

Dept. of Informatics, Univ. of Bergen  
Charles University, Prague  
dusan.knop@uib.no

Martin Koutecký†

Technion - Israel Institute of  
Technology  
koutecky@technion.ac.il

Matthias Mnich‡

Universität Bonn  
Maastricht University  
mmnich@uni-bonn.de

### ABSTRACT

Manipulation models for electoral systems are a core research theme in social choice theory; they include bribery (unweighted, weighted, swap, shift, ...), control (by adding or deleting voters or candidates), lobbying in referenda and others.

We develop a unifying framework for manipulation models with few types of people, one of the most commonly studied scenarios. A critical insight of our framework is to separate the descriptive complexity of the voting rule  $\mathcal{R}$  from the number of types of people. This allows us to finally settle the computational complexity of  $\mathcal{R}$ -SWAP BRIBERY, one of the most fundamental manipulation problems. In particular, we prove that  $\mathcal{R}$ -SWAP BRIBERY is fixed-parameter tractable when  $\mathcal{R}$  is Dodgson’s rule and Young’s rule, when parameterized by the number of candidates. This way, we resolve a long-standing open question from 2007 which was explicitly asked by Faliszewski et al. [JAIR 40, 2011].

Our algorithms reveal that the true hardness of bribery problems often stems from the complexity of the voting rules. On one hand, we give a fixed-parameter algorithm parameterized by number of types of people for complex voting rules. Thus, we reveal that  $\mathcal{R}$ -SWAP BRIBERY with Dodgson’s rule is much harder than with Condorcet’s rule, which can be expressed by a conjunction of linear inequalities, while Dodgson’s rule requires quantifier alternation and a bounded number of disjunctions of linear systems. On the other hand, we give an algorithm for quantifier-free voting rules which is parameterized only by the number of conjunctions of the voting rule and runs in time polynomial in the number of types of people. This way, our framework explains why SHIFT BRIBERY is polynomial-time solvable for the plurality voting rule, making explicit that the rule is simple in that it can be expressed with a single linear inequality, and that the number of voter types is polynomial.

### KEYWORDS

Swap bribery; Dodgson’s rule; Young’s rule; fixed-parameter algorithms

### ACM Reference Format:

Dušan Knop, Martin Koutecký, and Matthias Mnich. 2018. A Unifying Framework for Manipulation Problems. In *Proc. of the 17th International*

\* Author supported by grant projects P202/12/G061 of GA ČR and NFR MULTIVAL.

† Author supported by Technion postdoctoral fellowship.

‡ Author supported by ERC Starting Grant 305465 (BeyondWorstCase).

*Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden. © 2018 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.*

*Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018, IFAAMAS, 9 pages.*

## 1 INTRODUCTION

Problems of manipulation, bribery and control constitute a fundamental part of computational social choice. Many such problems are known to be NP-hard (or worse). However, their input can naturally be partitioned into several parts, like the number of voters, the number of candidates, and others. This motivates the study of such problems by the powerful tools of parameterized complexity.

A by-now classical example in this direction is the  $\mathcal{R}$ -SWAP BRIBERY problem, which takes as input an election consisting of a set  $C$  of candidates and a set  $V$  of voters with their individual preference lists  $\succ_v$  (for  $v \in V$ ), which are total orders over  $C$ . Additionally, for each voter  $v \in V$  and each pair of consecutive candidates  $c \succ_v c'$ , there is some cost  $\sigma^v(c, c') \in \mathbb{Z}$  of swapping the order of  $c$  and  $c'$  in  $\succ_v$ . The objective is to find a minimum-cost set of swaps of consecutive candidates in the preference lists in order to make a designated candidate  $c^* \in C$  the winner of the thus-perturbed election under a fixed voting rule  $\mathcal{R}$ .

This problem was introduced by Elkind et al. [8] and has since been studied for many classical voting rules  $\mathcal{R}$  [6, 11, 18, 23]. In particular, its computational complexity has been thoroughly analyzed with respect to the number of candidates  $|C|$ . The observation that  $|C|$  is often small motivated the search for *fixed-parameter algorithms* for  $\mathcal{R}$ -SWAP BRIBERY parameterized by  $|C|$ , which are algorithms that run in time  $f(|C|) \cdot n^{O(1)}$  for some computable function  $f$ , here  $n$  denotes the size of the input election; if such an algorithm exists, we then say that the problem is *fixed-parameter tractable* with respect to the parameter  $|C|$ .

Despite the problem’s importance, for a long time, only the “uniform cost” case of  $\mathcal{R}$ -SWAP BRIBERY was known to be fixed-parameter tractable for various voting rules, parameterized by the number of candidates; here, *uniform cost* refers to the special case that all voters have the same cost function, that is,  $\sigma^v \equiv \sigma$  for all  $v \in V$ . This is a fundamental result due to Dorn and Schlotter [6], who showed that  $\mathcal{R}$ -SWAP BRIBERY with uniform cost can be solved in time  $2^{2^{|C|^{O(1)}}} \cdot n^{O(1)}$  for all voting rules  $\mathcal{R}$  that are “linearly describable”. Many classical voting rules are indeed linearly describable, like any scoring protocol, Copeland $^\alpha$ , Maximin, or Bucklin.

Recently, Knop et al. [18] gave the first fixed-parameter algorithms for  $\mathcal{R}$ -SWAP BRIBERY for *general cost functions* for most voting rules  $\mathcal{R}$  studied in the literature (scoring protocol, Copeland $^\alpha$ , Maximin, Bucklin etc.), thereby removing the uniform cost assumption. This way, they resolved a long-standing open problem. Moreover, their algorithm runs in time  $2^{|C|^{O(1)}} \cdot n^{O(1)}$  for many rules  $\mathcal{R}$ ,

and thus improves the double-exponential run time by Dorn and Schlotter. Their key idea was to reduce the problem to so-called *n-fold integer programming*, which allowed them to solve the problem efficiently for bounded number of candidates despite their integer program having an unbounded number of variables. Their approach also solved  $\mathcal{R}$ -SWAP BRIBERY for  $\mathcal{R}$  being the Kemeny rule, even for general cost functions, though the Kemeny rule is not known to be linearly describable (cf. [10, p. 338]). However, this does not apply for Dodgson’s and Young’s rules.

## 1.1 The Challenge

Even so, there are some notable voting rules  $\mathcal{R}$  for which the complexity of  $\mathcal{R}$ -SWAP BRIBERY remained open even in the uniform cost case. This includes the Dodgson rule and the Young rule. Those rules are based on the notion of *Condorcet winner*, which is a candidate who beats any other candidate in a head-to-head contest. The Condorcet voting rule is very natural and dates back to the 18<sup>th</sup> century; however, clearly there exist elections without a Condorcet winner. In such a situation one proclaims those candidates as winners who are “closest” to being a Condorcet winner; different notions of closeness then yield different voting rules:

- Closeness measured as the of number of swaps in voter’s preference orders defines the *Dodgson rule*.
- Closeness measured as the number of voter deletions defines the *Young rule*.

Thus, a candidate  $c$  is a Dodgson winner if s/he can be made a Condorcet winner by a minimum number of swaps in the voter’s preference orders over all candidates; analogously for the Young rule and voter deletions.

When considering  $\mathcal{R}$ -SWAP BRIBERY, the Dodgson rule and the Young rule are much more complicated to handle than other rules; the reasons are several. First, for many voting rules  $\mathcal{R}$ , the winner of an election can be found in polynomial time, and solving this winner determination problem is certainly a necessary subtask when solving  $\mathcal{R}$ -SWAP BRIBERY. However, for  $\mathcal{R} \in \{\text{Dodgson, Young}\}$ , already winner determination is NP-hard, and so even *verifying* that  $c^*$  is indeed the winner of the perturbed election is intractable. In fact, winner determination for these voting rules is complete for parallel access to NP [13, 14, 22], denoted  $P_{||}^{\text{NP}}$ -complete<sup>1</sup>. Second, for more than 25 years the winner determination problem for the Dodgson rule and Young rule was only known to be solvable by an ILP-based algorithm [1] with doubly-exponential dependence in  $|C|$ ; a single-exponential algorithm is only known since recently [18]. This provides a sharp contrast to the Kemeny rule, for which simply enumerating all  $|C|!$  possible preference orders suffices to determine the winner. Faliszewski et al. [9] describe these difficulties:

*It is interesting to consider which features of Kemeny elections allow us to employ the above [ILP-based] attack, given that the same approach does not seem to work for either Dodgson or Young elections. One of the reasons is that the universal quantification implicit in Dodgson and Young elections is over an exponentially large search space, but the quantification in Kemeny is, in the case of a fixed candidate set, over a fixed number of options.*

<sup>1</sup>The class  $P_{||}^{\text{NP}}$  contains all problems solvable in polynomial time by a deterministic Turing machine which has access to an NP oracle, but must ask all of its oracle queries at once (i.e., the queries can not depend on each other).

Thus, it is not clear how to solve  $\mathcal{R}$ -SWAP BRIBERY even for uniform cost with *any* fixed-parameter algorithm for  $\mathcal{R}$  being the Dodgson rule or the Young rule. These complications led Faliszewski et al. [10] to explicitly ask for the complexity of  $\mathcal{R}$ -SWAP BRIBERY parameterized by the number of candidates under these rules.

## 1.2 Our Contributions

We start by making a key observation about the majority of fixed-parameter algorithms for  $\mathcal{R}$ -SWAP BRIBERY when  $|C|$  is small. A typical such result is an algorithm for  $\mathcal{R}$ -SWAP BRIBERY for  $\mathcal{R}$  being Condorcet’s voting rule. That algorithm uses two key ingredients:

- (1) There are at most  $|C|!$  preference orders of  $C$ , and hence each voter falls into one of  $|C|!$  types; thus, an input election is expressible as a society  $\mathbf{s} = (s_1, \dots, s_{|C|!})$ , where  $s_i$  is the number of voters of type  $i$ .
- (2) Expressing that a candidate  $c^*$  is a Condorcet winner is possible using a conjunction of  $|C| - 1$  linear inequalities in terms of  $\mathbf{s}$ .

As those key properties hold almost universally for voting rules  $\mathcal{R}$ , one might be tempted to think that if there are many types of voters, the  $\mathcal{R}$ -SWAP BRIBERY problem must be hard, and if there are few types of voters, the problem must be easy. However, two points arise as counter-evidence. First, very recently, Knop et al. [18] showed that even if there are many types of voters who differ by their cost functions, the  $\mathcal{R}$ -SWAP BRIBERY problem remains fixed-parameter tractable for a wide variety of voting rules  $\mathcal{R}$ . Second, as already mentioned, it was open since 2007 whether  $\mathcal{R}$ -SWAP BRIBERY with Dodgson’s and Young’s voting rule are fixed-parameter tractable for few candidates, even for uniform cost functions.

*From voters and candidates to societies.* Here, we take a novel perspective. We observe that the two key ingredients (1) and (2) apply much more widely than for  $\mathcal{R}$ -SWAP BRIBERY; namely, they are also present in many other manipulation, bribery and control problems. We therefore abstract away the specifics of such problems and introduce general notions of “society”, “moves in societies”, and “winning conditions”. Let  $\tau \in \mathbb{N}$  be the number of types of people (e.g., voters in an election or a referendum). A *society*  $\mathbf{s}$  is simply a non-negative  $\tau$ -dimensional integer vector encoding the numbers of people of each type. A *move*  $\mathbf{m}$  is a  $\tau^2$ -dimensional integer vector whose elements sum up to zero; it encodes how many people move from one type to another. A *change*  $\Delta$  is a  $\tau$ -dimensional vector (typically associated with a move) encoding the effect of a move on a society, such that  $\mathbf{s} + \Delta$  is again a society. Finally, a *winning condition*  $\Psi(\mathbf{s})$  is a predicate encoding some desirable property of a society, such as that a preferred candidate has won or that a preferred agenda was selected in a referendum. Specifically, we study winning conditions which are describable by formulas in *Presburger Arithmetic* (PA). PA is a logical language whose atomic formulas are linear inequalities over the integers, which are then joined with logical connectives and quantifiers. Thus, winning conditions describable by PA formulas widely generalize the class of linearly describable voting rules by Dorn and Schlotter [6].

Our main technical contribution informally reads as follows:

**THEOREM 1.1 (INFORMAL).** *Deciding satisfiability of PA formulas with two quantifiers is fixed-parameter tractable with respect to the dimension and length of formula, provided its coefficients and constants are given in unary.*

The importance of Theorem 1.1 arises from its applicability to the following general manipulation problem that we introduce here. This general manipulation problem, which we call **MINIMUM MOVE**, captures that many manipulation problems can be cast as finding a minimum move with respect to some objective function; in particular, it encompasses the well-studied **R-SWAP BRIBERY** problem. We study **MINIMUM MOVE** for linear objective functions and winning conditions  $\Psi$  expressible with PA formulas of the form “ $\exists V$ ”. For all such  $\Psi$ , with the help of Theorem 1.1, we show that **MINIMUM MOVE** is fixed-parameter tractable for combined parameter the descriptive complexity (length) of the winning condition  $\Psi$  and the number  $\tau$  of “types of people”, that is, it is fixed-parameter tractable for parameter length of  $\Psi$  plus  $\tau$ . As an important special case, we obtain the first fixed-parameter algorithm for **R-SWAP BRIBERY** for  $\mathcal{R}$  the Dodgson rule and the Young Rule with uniform costs. To this end, we model the winning condition of the Dodgson rule and Young rules as a PA formula. For intuition, consider the Young rule: a candidate  $c^*$  is a Young winner (with score  $d$ ) if there exists a set  $V^* \subseteq V$  of at most  $d$  voters such that  $c^*$  is a Condorcet winner of the election  $(C, V \setminus V^*)$ , and for all sets  $V' \subseteq V$  of at most  $d - 1$  voters any other candidate  $c \neq c^*$  is not a Condorcet winner of the election  $(C, V \setminus V')$ . This formula has one quantifier alternation, and its length (for a fixed score  $d$ ) is bounded by some function of  $|C|$ ; finally we have to take a disjunction of such formulas over all possible scores  $d$ . For a candidate set  $C$ , the number  $\tau$  of types of people is bounded by  $|C|!$ . Consequently, we finally settle the long-standing open question about the complexity of **R-SWAP BRIBERY** for  $\mathcal{R} \in \{\text{Dodgson, Young}\}$ , that was explicitly raised by Faliszewski [10]:

**THEOREM 1.2.** *R-SWAP BRIBERY with uniform cost is fixed-parameter tractable parameterized by the number of candidates for  $\mathcal{R}$  being the Dodgson rule or the Young rule; it can be solved in time  $f(|C|) \cdot |V|^{O(1)}$  for some computable function  $f$ .*

Beyond this fundamental problem, we show that a host of other well-studied manipulation problems are captured by our fixed-parameter algorithm for **MINIMUM MOVE**:

**COROLLARY 1.3.** *For  $\mathcal{R} \in \{\text{Dodgson, Young}\}$ , the following problems are fixed-parameter tractable for uniform costs when parameterized by the number  $|C|$  of candidates: **R-SBRIBERY**, **R-CCDV/CCAV**, **R-POSSIBLE WINNER**, and **R-EXTENSION BRIBERY**.*

Let us turn our attention to the parameter “number of types of people”  $\tau$ . Our main contribution here is the following:

**THEOREM 1.4 (INFORMAL).** *For any quantifier-free winning condition  $\Psi$ , **MINIMUM MOVE** can be solved in time polynomial in the number of types and exponential only in the number of linear inequalities of  $\Psi$ .*

Note that in many models of bribery and control, the number of *potential* types of people (i.e., types that can occur in any feasible solution) is polynomial in the number of people on input. For example, in **SHIFT BRIBERY**, every voter can be bribed to change their preferences order to one of  $|C| - 1$  orders; thus the number of potential types is  $(|C| - 1)|V|$ . Similarly, in **CCAV / CCDV** (constructive control by adding or deleting voters), every voter has an

active/latent bit; thus the number of potential types is  $2|V|$ . Similar arguments also work for **SUPPORT BRIBERY** where we change voters’ approval counts, and with a more intricate argumentation also for some voting rules and **BRIBERY** and **SBRIBERY**. In this sense, the fact that we need to consider  $|C|!$  potential voter types in **R-SWAP BRIBERY** almost seems like an anomaly, rather than a rule. In summary, the complexity of **MINIMUM MOVE** depends primarily on the descriptive complexity of the winning condition  $\Psi$ , because in many cases the number of types of people is polynomially bounded.

Another consequence of Theorem 1.1 are the first fixed-parameter algorithms for two important manipulation problems beyond **R-SWAP BRIBERY**. The **RESILIENT BUDGET** problem asks, for a given society whether allocating budget  $B$  is sufficient in order to repel any adversary move of cost at most  $B_a$  with a counter-move of cost at most  $B$  (so that the winning condition is still satisfied). Similarly, **ROBUST MOVE** asks for a move of cost at most  $B$  which causes the winning condition to be satisfied even after any adversary move of cost at most  $B_a$ . For formal definitions and results, cf. Sect. 4.2.

### 1.3 Interpretation of Results

Intuitively, the results obtained with Theorem 1.1 can be interpreted as follows. **Dodgson-SWAP BRIBERY** is fixed-parameter tractable parameterized by  $|C|$ ; however, this comes with at least two limitations as compared to prior work for simpler voting rules  $\mathcal{R}$ . First, our methods do not extend beyond the uniform cost scenario, and this remains a major open problem. Second, our result requires the input election to be given in unary, while prior work allows it to be given in binary (this is sometimes called the *succinct* case [9]). This is easily explained by the different descriptive complexities of the respective voting rules: for example, while Condorcet’s voting rule can be formulated as a quantifier-free PA formula, formulating Dodgson’s rule requires a long disjunction of formulas which use two quantifiers and a bounded number of disjunctions.

Theorem 1.4 lets us discuss more specifically the complexity of various voting rules. For example, the Plurality voting rule can be expressed with a single linear inequality encoding that a preferred candidate obtained more points than the remaining candidates altogether. Thus, all problems which can be modeled as **MINIMUM MOVE** are polynomial-time solvable with the Plurality voting rule. This interprets the result of Elkind et al. [8, Theorem 4.1] that **Plurality-SHIFT BRIBERY** is polynomial-time solvable: the number of potential voter types is polynomial, and Plurality has a simple description. Continuing, we may compare  $\mathcal{R}=\text{Borda}$  with  $\mathcal{R}=\text{Copeland}$ . The winning condition for  $\mathcal{R}=\text{Borda}$  can be described with  $|C| - 1$  inequalities, while  $\mathcal{R}=\text{Copeland}$  requires  $O(|C|^2)$  inequalities. Thus **Borda-SWAP BRIBERY** is solvable in time  $|C|^{O(|C|^2)} \log |V|$ , while **Copeland-SWAP BRIBERY** requires time  $|C|^{O(|C|^4)} \log |V|$ . Finally, all descriptions of Kemeny’s voting rule we are aware of require  $|C|!$  inequalities, and thus result in **Kemeny-SWAP BRIBERY** being solvable in time  $|C|!(|C|)^{O(1)} \log |V|$ . We do not claim these complexities to be best possible, but conjecture the existence of lower bounds separating the various voting rules; in particular, we believe that **Kemeny-SWAP BRIBERY** requires double-exponential time.

Finally, our work provides a natural next step in unifying the many different models that have been proposed for voting, bribing

and manipulation problems. In this direction, Faliszewski et al. [10] study what happens when multiple bribery and manipulation actions can occur in an election; e.g., CCAV asks for constructive control by adding voters while CCDV by deleting voters; similarly for CCAC and CCDC for adding/deleting candidates. Faliszewski et al. unify those various (up to that point separately studied) attacks. Similarly, Knop et al. [18] formulate the  $\mathcal{R}$ -MULTI BRIBERY problem, which also incorporates swaps and perturbing approval counts. The problem we put forward in this paper, MINIMUM MOVE, in some sense generalizes and simplifies all those “meta”-problems.

### 1.4 Related Work

We have reviewed most of the relevant computational social choice work already. However, there seems to be some confusion in the literature that deserves clarification. The paper of Faliszewski et al. [9] pioneering the concept of bribery in elections indeed considers the voting rules Kemeny, Dodgson and Young, and provides a fixed-parameter algorithm for Kemeny-BRIBERY. There are three features of their paper that we wish to discuss.

First, turning their attention to Dodgson-BRIBERY, they write:

*Applying the integer programming attack for the case of bribery within Dodgson-like election systems [...] is more complicated. These systems involve a more intricate interaction between bribing the voters and then changing their preferences. For Dodgson elections, after the bribery, we still need to worry about the adjacent switches within voters’s preference lists that make a particular candidate a Condorcet winner. [...] This interaction seems to be too complicated to be captured by an integer linear program, but building on the flavor of the Bartholdi et al. [1] ILP attack we can achieve the following: Instead of making  $p$  a winner, we can attempt to make  $p$  have at most a given Dodgson or Young score.*

They call this problem DodgsonScore-BRIBERY and provide positive results for it. Notice, however, that finding a bribery which makes  $c^*$  have a certain Dodgson score does *not* prevent another candidate to have a lower score and winning the bribed election. Thus, solving DodgsonScore-BRIBERY can be very far from the desired result.

Second, the authors then observe that a brute force approach enumerating all  $|V|^{|C|}$  briberies solves the Dodgson-BRIBERY problem in polynomial time for *constantly* many candidates; however, theirs is *not* a fixed-parameter algorithm for parameter  $|C|$ .

Third, they then introduce another voting system called Dodgson’, which is similar to Dodgson, and provide a fixed-parameter algorithm for winner determination. However, as in the case of Dodgson-BRIBERY, they do not provide a fixed-parameter algorithm for Dodgson’-BRIBERY.

The issue is then that a subsequent paper of Faliszewski et al. [10] claims that the Dodgson rule is “integer-linear-program implementable” and that this implies a certain election control problem generalizing BRIBERY to be fixed-parameter tractable [10, Theorem 6.2]. We believe the authors do not sufficiently differentiate between determining the winner with *one* ILP, as is the case for most simple voting rules, and with *multiple* ILPs, as is the case for Dodgson. Thus, we believe there is no evidence that the Dodgson rule is “integer-linear-program implementable”. Yet, this may be possible and this question still deserves attention. Whatever the reason, we are convinced that their [10, Theorem 6.1] does *not* hold for  $\mathcal{R}$ =Dodgson.

Hence, we believe that ours are the first fixed-parameter algorithms for any BRIBERY-like problem for  $\mathcal{R} \in \{\text{Dodgson, Young}\}$ .

## 2 PRELIMINARIES

Let  $m, n$  be integers. We define  $[m, n] := \{m, m + 1, \dots, n\}$  and  $[n] := [1, n]$ . Throughout, we reserve bold face letters (e.g.  $\mathbf{x}, \mathbf{y}$ ) for vectors. For a vector  $\mathbf{x}$  its  $i$ -th coordinate is  $x_i$ .

Next, we provide notions and notations for  $\mathcal{R}$ -SWAP BRIBERY.

*Elections.* An election  $(C, V)$  consists of a set  $C$  of candidates and a set  $V$  of voters, who indicate their preferences over the candidates in  $C$ , represented via a *preference order*  $>_v$  which is a total order over  $C$ . We often identify a voter  $v$  with their preference order  $>_v$ . Denote by  $\text{rank}(c, v)$  the rank of candidate  $c$  in  $>_v$ ;  $v$ ’s most preferred candidate has rank 1 and their least preferred candidate has rank  $|C|$ . For distinct candidates  $c, c' \in C$ , write  $c >_v c'$  if voter  $v$  prefers  $c$  over  $c'$ .

*Swaps.* Let  $(C, V)$  be an election and let  $>_v \in V$  be a voter. For candidates  $c, c' \in C$ , a *swap*  $s = (c, c')_v$  means to exchange the positions of  $c$  and  $c'$  in  $>_v$ ; denote the perturbed order by  $>_v^s$ . A swap  $(c, c')_v$  is *admissible in*  $>_v$  if  $\text{rank}(c, v) = \text{rank}(c', v) - 1$ . A set  $S$  of swaps is *admissible in*  $>_v$  if they can be applied sequentially in  $>_v$ , one after the other, in some order, such that each one of them is admissible. Note that the perturbed vote, denoted by  $>_v^S$ , is independent from the order in which the swaps of  $S$  are applied. We extend this notation for applying swaps in several votes and denote it  $V^S$ . We specify  $v$ ’s cost of swaps by a function  $\sigma^v : C \times C \rightarrow \mathbb{Z}$ .

*Voting rules.* A voting rule  $\mathcal{R}$  is a function that maps an election  $(C, V)$  to a subset  $W \subseteq C$ , called the *winners*. A candidate  $c \in C$  is a *Condorcet winner* if any other  $c' \in C \setminus \{c\}$  satisfies  $|\{>_v \in V \mid c >_v c'\}| > |\{>_v \in V \mid c' >_v c\}|$ ; then we say that  $c$  *beats*  $c'$  *in a head-to-head contest*. The *Young score* of  $c \in C$  is the size of the smallest subset  $V' \subseteq V$  such that  $c$  is a Condorcet winner in  $(C, V \setminus V')$ . Analogously, the *Dodgson score* of  $c \in C$  is the size of the smallest admissible set of swaps  $S$  such that  $c$  is a Condorcet winner in  $(C, V^S)$ . Then,  $c$  is a *Young (Dodgson) winner* if it has minimum Young (Dodgson) score.

We aim to solve the following problem:

#### $\mathcal{R}$ -SWAP BRIBERY

**Input:** An election  $(C, V)$ , a designated candidate  $c^* \in C$  and swap costs  $\sigma^v : C \times C \rightarrow \mathbb{Z}$  for  $v \in V$ .

**Find:** A set  $S$  of admissible swaps of minimum cost so that  $c^*$  wins the election  $(C, V^S)$  under the rule  $\mathcal{R}$ .

## 3 MOVES IN SOCIETIES AND PRESBURGER ARITHMETIC

Let  $\tau \in \mathbb{N}$  be the number of *types of people*. A *society* is a non-negative  $\tau$ -dimensional integer vector  $\mathbf{s} = (s_1, \dots, s_\tau)$ . In most problems, we are interested in modifying a society by moving people between types. A *move* is a vector  $\mathbf{m} = (m_{1,1}, \dots, m_{\tau,\tau}) \in \mathbb{Z}^{\tau^2}$ . Intuitively,  $m_{i,j}$  is the number of people of type  $i$  turning type  $j$ .

*Definition 3.1.* A *change* is a vector  $\Delta = (\Delta_1, \dots, \Delta_\tau) \in \mathbb{Z}^\tau$  whose elements sum up to 0. We say that  $\Delta$  is the *change associated with a move*  $\mathbf{m}$  if  $\Delta_i = \sum_{j=1}^\tau m_{j,i} - m_{i,j}$ , and we write  $\Delta = \Delta(\mathbf{m})$ .

A change  $\Delta$  is *feasible with respect to society*  $\mathbf{s}$  if  $\mathbf{s} + \Delta \geq \mathbf{0}$ , i.e., if applying the change  $\Delta$  to  $\mathbf{s}$  results in a society.

One more useful notion is that of a “move costs vector”. A *move costs vector* is a vector  $\mathbf{c} = (c_{1,1}, \dots, c_{\tau,\tau})$  in  $(\mathbb{N} \cup \{+\infty\})^{\tau^2}$  which satisfies the triangle inequality, i.e.,  $c_{i,k} \leq c_{i,j} + c_{j,k}$  for all distinct  $i, j, k$ . Let  $k \in \mathbb{N}$  and  $\mathbf{c}$  be a move costs vector. A move  $\mathbf{m}$  is a  $(\mathbf{c}, k)$ -move if  $\mathbf{c}^\top \mathbf{m} \leq k$ .

Finally, we want to check that the society (e.g., resulting from applying some moves) satisfies a certain desired condition. This condition depends on the problem we are modeling: in variants of bribery, it says that a preferred candidate is elected as a winner or to be a part of a committee under a given voting rule; in the context of lobbying, it says that a preferred agenda was selected. To allow large expressibility, we make a very broad definition: A *winning condition of width  $\tau$*  is a predicate  $\Psi(\mathbf{s})$  with  $\tau$  free variables.

### 3.1 Presburger Arithmetic

For two formulas  $\Phi$  and  $\Psi$ , we denote their equivalence by  $\Phi \cong \Psi$ .

*Definition 3.2 (Presburger Arithmetic).* Let

$$\hat{\mathcal{P}}_{0,(n_0),\delta,\gamma,\alpha,\beta} = \{\hat{\Psi}(\mathbf{x}_0)\}$$

be the set of quantifier-free Presburger Arithmetic (PA) formulas with  $n_0$  free variables  $\mathbf{x}_0$  which are a disjunction of at most  $\delta$  conjunctions of linear inequalities  $\mathbf{a}^\top \mathbf{x}_0 \leq b$ , each of length at most  $\gamma$ , where  $\|\mathbf{a}\|_\infty \leq \alpha$  and  $|b| \leq \beta$  for each inequality. Then, let

$$\mathcal{P}_{0,(n_0),\delta,\gamma,\alpha,\beta} = \{\Psi(\mathbf{x}_0) \mid \exists \hat{\Psi}(\mathbf{x}_0) \in \hat{\mathcal{P}}_{0,(n_0),\delta,\gamma,\alpha,\beta} : \hat{\Psi} \cong \Psi\}$$

be the set of PA formulas equivalent to some DNF formula from  $\hat{\mathcal{P}}_{0,(n_0),\delta,\gamma,\alpha,\beta}$ . Finally, let

$$\mathcal{P}_{k,\mathbf{n},\delta,\gamma,\alpha,\beta} = \{\Psi(\mathbf{x}_0) \equiv \exists/\forall \mathbf{x}_1 \exists/\forall \mathbf{x}_2 \dots \exists/\forall \mathbf{x}_k : \Phi(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k)\}$$

be the set of PA formulas with *quantifier depth*  $k$ ,  $n_0$  free variables  $\mathbf{x}_0$ , and *dimension*  $\mathbf{n} = (n_0, n_1, \dots, n_k)$ ; here,  $x_i \in \mathbb{Z}^{n_i}$  for each  $i$  and  $\Phi(\mathbf{x}_0, \dots, \mathbf{x}_k) \in \mathcal{P}_{0,(n),\delta,\gamma,\alpha,\beta}$  with  $n = \sum_{i=1}^k n_i$ . The *length* of  $\Psi(\mathbf{x}_0)$  is the number of symbols it contains, which is polynomially bounded in  $(n, \gamma, \delta)$ . By  $\exists\mathcal{P}$  and  $\forall\mathcal{P}$  we denote the sets of PA formulas whose leading quantifier is  $\exists$  or  $\forall$ , respectively.

**Example.** A simple example of PA is the following formula.

$$\begin{aligned} \Psi(y) \equiv \forall x_1 x_2 \exists z_1 z_2 z_3 : (x_1 + y = z_3 \wedge y \geq 0) \vee \\ (3x_1 + 10y - 3z_1 \leq 13 \wedge 2x_2 + 5y - z_2 \leq 11 \\ \wedge x_1 + 1y - z_3 \geq 9 \wedge z_1 - z_2 + 2z_3 \leq 6) \end{aligned}$$

Here  $k = 2$ ,  $\mathbf{n} = (1, 2, 3)$ ,  $\delta = 2$ ,  $\gamma = 4$ ,  $\alpha = 10$ , and  $\beta = 13$ .

We study winning conditions  $\Psi(\mathbf{s})$  expressible in PA, and state our complexity results with respect to the *descriptive complexity* of  $\Psi$ , which is its number of variables, quantifiers, logical connectives, and unary encoding length of coefficients and constants.

*Vocabulary.* We express relevant definitions by simple PA formulas over integral variables and with integer coefficients and constants:

- $\text{society}(\mathbf{s}) \equiv \mathbf{s} \geq \mathbf{0} \in \mathcal{P}_{0,(\tau),0,\tau,1,0}$ ,
- $\text{move}(\mathbf{m}) \in \mathcal{P}_{0,(\tau^2),0,0,0,0}$ ,
- $\Delta = \Delta(\mathbf{m})$  is a linear map  $\Delta_i = \sum_{j=1}^\tau m_{j,i} - m_{i,j}$ ; thus if we let  $\Psi(\mathbf{m}, \Delta) \equiv \Delta = \Delta(\mathbf{m})$ , then  $\Psi(\mathbf{m}, \Delta) \in \mathcal{P}_{0,(\tau+\tau^2),0,\tau,1,0}$

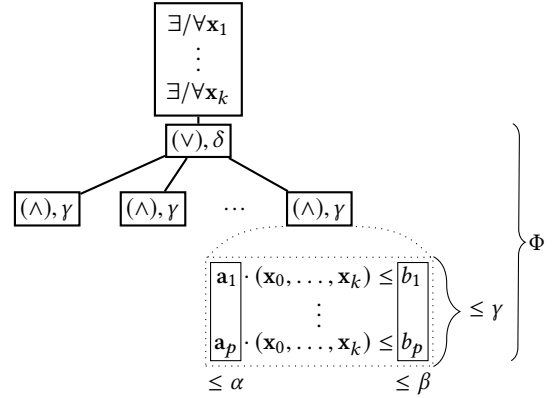


Figure 1: Visualization of PA formula  $\Psi(\mathbf{x}_0)$  in DNF.

- $\text{feasible}(\mathbf{s}, \Delta) \equiv (\mathbf{s} + \Delta(\mathbf{m}) \geq \mathbf{0} \wedge \mathbf{1}^\top \Delta = 0) \in \mathcal{P}_{0,(2\tau),0,\tau,2,0}$ , and,
- $(\mathbf{c}, k)$ -move $(\mathbf{m}) \equiv \mathbf{c}^\top \mathbf{m} \leq k \in \mathcal{P}_{0,(\tau),0,1,\|\mathbf{c}\|_\infty,k}$ .

We note that, for elections, our definition of winning condition generalizes the notion of “linearly-definable voting rules” by Dorn and Schlotter [6]. Precisely, those rules belong to  $\exists\mathcal{P}$  with  $k = 1$ ; we will show that Dodgson and Young are in  $\exists\mathcal{P}$  with  $k = 2$ . Thus, our winning conditions capture an extensive set of voting rules.

### 3.2 Modeling Problems as Minimum Move

We model moves in societies by the following general problem:

MINIMUM MOVE	
<b>Input:</b>	A society $\mathbf{s}$ , an objective function $f : \mathbb{Z}^{\tau^2} \rightarrow \mathbb{Z}$ , a winning condition $\Psi$ .
<b>Find:</b>	A move $\mathbf{m}$ minimizing $f(\mathbf{m})$ s.t. $\Psi(\mathbf{s} + \Delta(\mathbf{m})) \wedge \text{feasible}(\mathbf{s}, \Delta(\mathbf{m}))$ .

It models many well-studied problems:

*Multi bribery.* Knop et al. [18] introduce a generalization of various bribery problems called  $\mathcal{R}$ -MULTI BRIBERY. Informally, we are given an election  $(C, V)$  where each voter further has an approval count, and is either active or latent the status of which can be changed at certain cost; likewise, there are costs for perturbing their preference order or approval count. This problem generalizes BRIBERY, \$BRIBERY, SWAP BRIBERY, SHIFT BRIBERY, SUPPORT BRIBERY, EXTENSION BRIBERY, POSSIBLE WINNER, CONSTRUCTIVE CONTROL BY ADDING/DELETING VOTERS and other problems.

Notice that there are at most  $|C|!$  possible preference orders, at most  $|C|$  possible approval counts, and 2 states “active” or “latent”. Thus, there are at most  $\tau \leq 2|C| \cdot |C|!$  potential types of voters, and we can express the input election as a society  $\mathbf{s}$ . A move costs vector  $\mathbf{c}$  describing the costs of moving a voter from one type to another is obtained by calculating (possibly using a shortest path algorithm) the least costs based on the given cost functions. Let  $\Psi(\mathbf{s})$  be a PA formula which is satisfied if the preferred candidate wins under the voting rule  $\mathcal{R}$  in a society  $\mathbf{s}$ . Then, a bribery of minimum cost in a  $\mathcal{R}$ -MULTI BRIBERY instance can be modeled as solving MINIMUM MOVE with  $f(\mathbf{m}) = \mathbf{c}^\top \mathbf{m}$ . This modeling, combined with Theorem 1.2 and Corollary 4.6, yields Corollary 1.3.

*Multiwinner elections.* Bredebeck et al. [4] study the complexity of SHIFT BRIBERY in committee elections, that is, in elections with multiple winners. The modeling is exactly the same as above, except for the winning condition  $\Psi$  which will be a long disjunction over all committees which include the preferred candidate.

*Lobbying in referenda.* Bredebeck et al. [3] study the complexity of LOBBYING in referenda. There, voters cast ballots with their “yes”/“no” answers to issues. The task is to push an agenda, i.e., a certain outcome. Again, voters fall into groups according to their ballots, the costs of changing their opinions forms a move costs vector, and a winning condition  $\Psi$  expresses that the selected agenda succeeded.

## 4 SENTENCES WITH TWO QUANTIFIERS

We shall now introduce the building blocks of our proof of Theorem 1.1. Woods [24] gives an algorithm that efficiently converts any quantifier-free PA formula  $\Phi$  into an equivalent DNF formula  $\hat{\Phi}$  of bounded length:

LEMMA 4.1 (WOODS [24, PROPOSITION 5.1]). *Let  $\Phi(\mathbf{x})$  be a quantifier-free PA formula with  $\mathbf{x} \in \mathbb{Z}^d$  containing  $N$  inequalities, whose coefficients and right-hand sides are bounded in absolute value by  $\alpha$  and  $\beta$ , respectively. Then  $\Phi(\mathbf{x})$  can be converted into an equivalent DNF formula  $\hat{\Phi}(\mathbf{x})$  with at most  $\delta = N^{O(d)}$  disjunctions, each containing at most  $N$  conjunctions with the same bound on  $\alpha$  and  $\beta$ .*

It is often useful for the quantifiers of a PA formula to range over integer points of polyhedra, e.g.  $\forall \mathbf{x} \in Q$  (we do not write  $Q \cap \mathbb{Z}^n$  for brevity, as we assume everything to be integer); again, our definition is not restrictive by the fact that we can always rearrange:

$$\begin{aligned} \Psi(\mathbf{x}_0) \equiv & \exists \mathbf{x}_1 \in Q_1 \cdots \forall \exists \mathbf{x}_k \in Q_k : \Phi(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k) \equiv \\ & \exists \mathbf{x}_1 \cdots \forall \exists \mathbf{x}_k : \left( \Phi(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k) \right. \\ & \left. \wedge (\mathbf{x}_1 \in Q_1 \wedge \mathbf{x}_3 \in Q_3 \cdots) \wedge \bigvee (\mathbf{x}_2 \notin Q_2 \vee \mathbf{x}_4 \notin Q_4 \cdots) \right) \end{aligned}$$

*Parametric ILP.* A special case of PA are *parametric ILPs*, which can be viewed<sup>2</sup> as deciding the sentence

$$\forall \mathbf{x} \in \mathbb{Z}^p : \mathbf{A}\mathbf{x} \leq \mathbf{b} \exists \mathbf{y} \in \mathbb{Z}^n : B(\mathbf{x}, \mathbf{y}) \leq \mathbf{e},$$

where  $A \in \mathbb{Z}^{\ell \times p}$  and  $B \in \mathbb{Z}^{m \times n}$  are integer matrices. A consequence of an algorithm of Eisenbrand and Shmonin [7] is the following:

COROLLARY 4.2 ([7, THEOREM 4.2], [5, COROLLARY 1]). *Any parametric ILP whose entries of  $A, B, \mathbf{b}$  and  $\mathbf{e}$  are given in unary, is fixed-parameter tractable when parameterized by  $n, m$  and  $p$ .*

*ILP and disjunctions.* We shall use a folklore result about implementing disjunctions in ILP when the domains of variables can be bounded. For that, we need another definition.

*Definition 4.3 (B-bounded, B-small PA formula).* Let  $\Psi(\mathbf{x}_0) \equiv \exists \forall \mathbf{x}_1 \cdots \exists \forall \mathbf{x}_k : \Phi(\mathbf{x}_0, \dots, \mathbf{x}_k)$  be a PA formula, and let  $\Psi_B(\mathbf{x}_0) \equiv \exists \forall \mathbf{x}_1 \in [-B, B]^{n_1} \cdots \exists \forall \mathbf{x}_k \in [-B, B]^{n_k} : \Phi(\mathbf{x}_0, \dots, \mathbf{x}_k)$ . Then we say that  $\Psi(\mathbf{x}_0)$  is *B-bounded* if

$$\{\mathbf{x} \in \mathbb{Z}^{n_0} \mid \Psi_B(\mathbf{x})\} \cap [-B, B]^{n_0} = \{\mathbf{x} \in \mathbb{Z}^{n_0} \mid \Psi(\mathbf{x})\},$$

<sup>2</sup>Parametric ILPs are typically viewed as ILPs with a varying right hand side, that is, deciding the sentence  $\forall \mathbf{b} \exists \mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}$ ; it is known that our formulation is equivalent, as shown by Crampton et al. [5], who call it *ILP Resiliency*.

i.e., the set of feasible solutions does not change by restricting all quantifiers and free variables to the corresponding box of size  $B$ .

Moreover, we say that any  $\Psi \in \mathcal{P}_{k, n, \delta, \gamma, B, B}$  is *B-small* if it is *B-bounded*, that is, its coefficients and constants are bounded by  $B$ .

A special case are ILPs which are *B-small*; they correspond to PA formulas with  $k = 0$  and  $\delta = 0$ ; for such formulas we show:

LEMMA 4.4 (ILP DISJUNCTIONS [FOLKLORE]). *Let  $A_i \mathbf{x} \leq \mathbf{b}_i$  for  $i \in [d]$  be  $B$ -small ILPs with  $A_i \in \mathbb{Z}^{m \times n}$  for each  $i \in [d]$ . Then, a  $(B^2 n)$ -small system  $\mathbf{A}\mathbf{x} \leq \mathbf{b}$  with  $A \in \mathbb{Z}^{(md+d+1) \times (n+d)}$  can be constructed in time  $O(dm + n + \sum_{i=1}^d \langle A_i, \mathbf{b}_i \rangle)$  such that*

$$\exists (\mathbf{x}, \mathbf{y}) \in \mathbb{Z}^{n+d} : \mathbf{A}(\mathbf{x}, \mathbf{y}) \leq \mathbf{b} \iff \exists \mathbf{x} \in \mathbb{Z}^n : \bigvee_{i \in [d]} A_i \mathbf{x} \leq \mathbf{b}_i .$$

PROOF. Let  $M = B^2 n$ , let  $y_i$  for  $i \in [d]$  be binary variables, and consider the following system:

$$\sum_{i=1}^d y_i = 1 \bigwedge y_i \geq 0, A_i \mathbf{x} \leq \mathbf{b}_i + M(1 - y_i) \text{ for all } i \in [d] .$$

Assume it has an integer solution  $\mathbf{y}$ . Then there is an index  $i \in [d]$  such that  $y_i = 1$  and thus  $A_i \mathbf{x} \leq \mathbf{b}_i + \mathbf{0}$  holds; thus, the system  $A_i \mathbf{x} \leq \mathbf{b}_i$  has an integer solution. In the other direction, assume that the system  $A_i \mathbf{x} \leq \mathbf{b}_i$  has a solution  $\mathbf{x}$ ; then let  $y_i = 1$ . We shall prove that  $A_{i'} \mathbf{x} \leq \mathbf{b}_{i'} + M$  holds for all  $i' \neq i$ . Since  $A_{i'} \mathbf{x} \leq \mathbf{b}_{i'}$  is  $B$ -small, each of its row sums has  $n$  terms which are a multiple of two numbers, each bounded by  $B$ , and thus is at most  $B^2 n$ . Moreover, since  $y_i = 1$ , we have  $y_{i'} = 0$  and thus the right hand side is  $\mathbf{b}_{i'} + M$  and every assignment of  $\mathbf{x}$  feasible for  $A_i \mathbf{x} \leq \mathbf{b}_i$  satisfies it. Clearly, the new system has  $n + d$  variables,  $md + d + 1$  inequalities, is  $B$ -bounded and  $\|A\|_\infty = B^2 n$  and thus it is  $(B^2 n)$ -small, and can be constructed in the claimed time.  $\square$

We now prove Theorem 1.1; we restate it in formal terms here:

THEOREM 4.5 (FORMAL VERSION OF THEOREM 1.1). *Let  $\Psi$  be a  $\beta$ -small  $\mathcal{P}_{2, n, \delta, \gamma, \alpha, \beta}$  sentence (i.e., without free variables and thus  $n_0 = 0$ ). Then  $\Psi$  can be decided in time  $g(n, \delta, \gamma) \text{poly}(\alpha, \beta)$  for some computable function  $g$ .*

PROOF. Let  $\Psi \equiv \exists \mathbf{x}_1 \forall \mathbf{x}_2 : \Phi(\mathbf{x}_1, \mathbf{x}_2)$ . Clearly, to decide  $\Psi$  we can instead decide  $\neg \Psi \equiv \forall \mathbf{x}_1 \exists \mathbf{x}_2 : \neg \Phi(\mathbf{x}_1, \mathbf{x}_2)$ . Consider the formula  $\neg \Phi(\mathbf{x}_1, \mathbf{x}_2)$ : by Lemma 4.1, there exists an equivalent  $\beta$ -small DNF formula  $\tilde{\Phi}(\mathbf{x}_1, \mathbf{x}_2)$  such that the number of its disjunctions and conjunctions is a function of just the original  $\delta$  and  $\gamma$ .

Thus, from now on focus on the case  $\Psi \equiv \forall \mathbf{x}_1 \exists \mathbf{x}_2 : \Phi(\mathbf{x}_1, \mathbf{x}_2)$ . Our next task is to construct an instance of PARAMETRIC ILP equivalent to deciding  $\Psi$ . To do that, replace  $\Phi(\mathbf{x}_1, \mathbf{x}_2)$  with  $\Xi(\mathbf{x}_1, \mathbf{x}_2) \equiv \exists \mathbf{x}_3 : A(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \leq \mathbf{b}$ , where  $\mathbf{x}_3$  is of dimension  $\delta$  and the system of linear inequalities  $A(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \leq \mathbf{b}$  has bounded length, coefficients and right sides. Now we use Lemma 4.4. Assume that  $\Phi(\mathbf{x}_1, \mathbf{x}_2)$  is a disjunction of  $\delta$  linear systems, each of at most  $\gamma$  conjunctions, and by the assumptions of the theorem we know that  $\Psi$  is  $\beta$ -small. Plugging into Lemma 4.4, we have  $B = \beta$ ,  $d = \delta$ ,  $m = \gamma$  and  $n = n_1 + n_2$ , and we obtain a formula  $\exists \mathbf{x}_3 : A(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \leq \mathbf{b}$  equivalent to  $\Phi(\mathbf{x}_1, \mathbf{x}_2)$ . Thus, we are left with deciding  $\forall \mathbf{x}_1 \exists (\mathbf{x}_2, \mathbf{x}_3) : A(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \leq \mathbf{b}$  with the following parameters:

- The coefficients and right-hand sides  $\|(A, \mathbf{b})\|_\infty$  are bounded by some computable function  $f(\mathbf{n}, \delta, \gamma) \cdot (\alpha\beta)^2$ .
- The dimensions of  $A$  are bounded by  $f(\mathbf{n}, \delta, \gamma)$ .

Thus, we are in the setting of Corollary 4.2 and we can decide the above sentence by a fixed-parameter algorithm.  $\square$

**COROLLARY 4.6.** *Let  $\Psi(\mathbf{x}_0)$  be a disjunction of  $D$  many  $\beta$ -small  $\exists\mathcal{P}_{2,(\tau, n_1, n_2), \delta, \gamma, \alpha, \beta}$  formulas. Then MINIMUM MOVE with objective  $f(\mathbf{x}_0) = \mathbf{c}^\top \mathbf{x}_0$  and winning condition  $\Psi$  can be solved in time  $g(\tau, n_1, n_2, \delta, \gamma) \text{poly}(D, \alpha, \beta, \mathbf{c})$  for some computable function  $g$ .*

**PROOF.** Let  $\Psi \equiv \Psi_1 \vee \dots \vee \Psi_D$ , and let  $\Xi_i(\mathbf{m}) \equiv \Psi_i(\mathbf{s} + \Delta(\mathbf{m})) \wedge (\mathbf{c}, B)\text{-move}(\mathbf{m}) \wedge \text{feasible}(\Delta(\mathbf{m}), \mathbf{s})$ , for each  $i \in [D]$ . Now we need to perform binary search with parameter  $B$  on the sentence

$$\exists \mathbf{m} \Psi(\mathbf{s} + \Delta(\mathbf{m})) \wedge (\mathbf{c}, B)\text{-move}(\mathbf{m}) \wedge \text{feasible}(\Delta(\mathbf{m}), \mathbf{s}) .$$

This sentence is obviously equivalent to

$$\exists \mathbf{m} \Xi_1(\mathbf{m}) \vee \dots \vee \exists \mathbf{m} \Xi_D(\mathbf{m}),$$

which can be decided by  $D$  application of Theorem 4.5. This is possible in the claimed time, as, for each  $i \in [D]$ ,  $\exists \mathbf{m} \Xi_i(\mathbf{m})$  has quantifier depth 2, is  $\beta'$ -small for  $\beta' = \text{poly}(\beta, \tau + n_1 + n_2, \|\mathbf{c}\|_\infty)$  and has dimension  $\mathbf{n} = (0, \tau + n_1, n_2)$ . A minimum move can then be constructed by coordinate-wise binary search; cf. [19, Thm. 4.1].  $\square$

#### 4.1 Application: Swap Bribery for the Dodgson Rule and Young Rule

We now prove Theorem 1.2 by giving our fixed-parameter algorithm for  $\mathcal{R}$ -SWAP BRIBERY with  $\mathcal{R}$  the Dodgson Rule or the Young Rule.

**PROOF OF THEOREM 1.2.** Fix an instance  $(C, V, c^\star, \sigma)$  of  $\mathcal{R}$ -SWAP BRIBERY. As there are  $|C|!$  possible total orders on  $C$ , each voter has one of these orders. Thus we view the election as a society  $\mathbf{s} = (s_1, \dots, s_{|C|!})$ . Let  $\mathbf{c}_{\text{swap}}$  be a move costs vector defined as  $c_{i,j}$  = “swap distance between types  $i$  and  $j$ ” (this is simply the number of inversions between the permutations  $i$  and  $j$  [8, Proposition 3.2]); observe that a society  $\mathbf{s}$  is in swap distance at most  $d$  from  $\mathbf{s}'$  if  $\mathbf{s}' = \mathbf{s} + \Delta(\mathbf{m})$  and  $\mathbf{m}$  is a feasible  $(\mathbf{c}_{\text{swap}}, d)$ -move for  $\mathbf{s}$ . Moreover, let  $\mathbf{c}_{\text{del}}$  be defined as  $c_{i,0} = 1$  for every type  $i$  and  $c_{i,j} = +\infty$  for every two types  $i, j \neq 0$ , where  $0$  is a type for latent voters; observe that  $\mathbf{s}$  is in voter deletion distance at most  $d$  from  $\mathbf{s}'$  if  $\mathbf{s}' = \mathbf{s} + \Delta(\mathbf{m})$  and  $\mathbf{m}$  is a  $(\mathbf{c}_{\text{del}}, d)$ -move.

Our plan is to express the winning condition for  $\mathcal{R}$  using a PA formula  $\Psi_{\text{Dodgson}}(\mathbf{s})$  which is a disjunction of polynomially many (in  $|V|$  and  $|C|$ ) formulas from  $\exists\mathcal{P}_{2,\mathbf{n},\delta,\gamma,\alpha,\beta}$ , and then solve MINIMUM MOVE with  $f(\mathbf{m}) = \sigma^\top \mathbf{m}$  and  $\Psi = \Psi_{\text{Dodgson}}(\mathbf{s})$  using Corollary 4.6; recall that since the instance is uniform, we have  $\sigma^v \equiv \sigma$  for all  $v \in V$ , and we let  $\sigma$  be the move costs vector obtained from  $\sigma$ . For  $\Psi_{\text{Young}}$ , we proceed analogously. In the end, we will verify that  $\mathbf{n}, \delta, \gamma$  are bounded by a function of  $|C|$ , and that  $\alpha, \beta$  are polynomial in the input size.

*Expressing  $\Psi_{\text{Dodgson}}$  and  $\Psi_{\text{Young}}$ .* The winning condition for candidate  $c^\star$  in both the Dodgson and Young rule can be viewed as  $c^\star$  being closest to being a Condorcet winner with respect to some distance measure. Specifically, for the Dodgson rule, this distance is the number of swaps in the preference orders, and in the Young rule, the distance is the number of voter deletions. For this reason, finding a bribery which makes  $c^\star$  a winner corresponds to finding

a bribery which, for some  $d \in \mathbb{N}$ , makes  $c^\star$  be in distance  $d$  or less from being a Condorcet winner, and, simultaneously, making every other candidate  $c \neq c^\star$  be in distance at least  $d$  from being a Condorcet winner.

Let us fix  $d \in \mathbb{N}$ ; later we will argue that we can go through all  $D$  relevant choices of  $d$ . First, we will express the condition that in a society  $\mathbf{s}$ , candidate  $c$  beats candidate  $c'$  in a head-to-head contest:

$$\text{beats}(c, c', \mathbf{s}) \equiv \sum_{i:c >_i c'} s_i > \sum_{i:c' >_i c} s_i,$$

where  $>_i$  is the preference order shared by all voters of type  $i$ . Then, it is easy to express that  $c^\star$  is a winner in  $\mathbf{s}$  under Condorcet’s rule:

$$\Phi_{\text{Condorcet}}(c, \mathbf{s}) \equiv \bigwedge_{c' \neq c} \text{beats}(c, c', \mathbf{s}) .$$

Finally, we express that  $c^\star$  is a Dodgson-winner with score  $d$  by  $\Psi_{\text{Dodgson},d}$  as follows:

$$\begin{aligned} \Psi_{\text{Dodgson},d}(\mathbf{s}) \equiv & (\exists \mathbf{m} : (\mathbf{c}_{\text{swap}}, d)\text{-move}(\mathbf{m}) \wedge \text{feasible}(\Delta(\mathbf{m}), \mathbf{s}) \wedge \\ & \Phi_{\text{Condorcet}}(c^\star, \mathbf{s} + \Delta(\mathbf{m}))) \wedge \\ & (\forall \mathbf{m} : (\mathbf{c}_{\text{swap}}, d-1)\text{-move}(\mathbf{m}) \wedge \text{feasible}(\Delta(\mathbf{m}), \mathbf{s}) \wedge \\ & \bigwedge_{c \neq c^\star} \neg \Phi_{\text{Condorcet}}(c, \mathbf{s} + \Delta(\mathbf{m}))) \end{aligned}$$

Then,  $\Psi_{\text{Dodgson}} \equiv \Psi_{\text{Dodgson},1} \vee \dots \vee \Psi_{\text{Dodgson},D}$ . Clearly,  $\Psi_{\text{Young},d}$  is obtained simply by replacing  $\mathbf{c}_{\text{swap}}$  with  $\mathbf{c}_{\text{del}}$ .

*Complexity.* The number  $D$  of relevant choices of  $d$  is bounded by  $|C|^2|V|$ : at most  $|C|^2$  swaps suffice for any bribery of a single voter, and there are at most  $|V|$  voters, thus  $D \leq |C|^2|V|$  is a bound on the Dodgson score of a candidate. For the Young score,  $D \leq |V|$ .

Since for each  $d \in [D]$ ,  $\Psi_{\text{Dodgson},d} \equiv \exists \mathbf{m} : \Xi_1(\mathbf{m}) \wedge \forall \mathbf{m}' : \Xi_2(\mathbf{m}')$  can be equivalently rewritten as  $\exists \mathbf{m} \forall \mathbf{m}' : \Xi_1(\mathbf{m}) \wedge \Xi_2(\mathbf{m}')$ , it has quantifier depth 2, and thus belongs to  $\exists\mathcal{P}_{2,\mathbf{n},\delta,\gamma,\alpha,\beta}$ . Let us determine the parameters:

- $\mathbf{n}$  is the vector of dimensions; thus  $n_0 = \tau = |C|!$  are the dimensions of a vector encoding a society, and  $n_1 = \tau^2 = |C|^2$  are the dimensions of a vector encoding a move.
- $\delta$  is the number of disjunctions, and it is polynomial in  $|C|$ ,
- $\gamma$  is the number of conjunctions, and it is polynomial in  $\tau = |C|!$ ,
- $\alpha$  is the largest coefficient, which is  $\|\mathbf{c}_{\text{swap}}\|_\infty \leq |C|^2$ ,
- $\beta$  is the largest right-hand side, which is  $d \leq |C|^2|V|$ .

Thus,  $\mathbf{n}, \delta, \gamma$  are functions of the parameter  $|C|$  and  $\alpha, \beta$  are polynomial in the size of the input election  $(C, V, \{>_v \mid v \in V\})$ , as required by Corollary 4.6. Analogous analysis applies to  $\Psi_{\text{Young},d}$ .  $\square$

Replacing  $\mathbf{c}_{\text{swap}}$  with  $\mathbf{1}^\top$  produces the winning condition for Dodgson’ as introduced by Faliszewski et al. [9]. Furthermore, it is interesting to consider voting rules obtained by replacing Condorcet’s rule in the definition of Dodgson’s and Young’s rule. For example, the Majority rule also might not produce a winner, and most rules (Copeland<sup>a</sup>, Scoring protocol etc.) allow ties. Let the score of a candidate be their distance (swap, deletion, etc.) from being a (unique) winner under rule  $\mathcal{R}$ . We remark that if we replace Condorcet’s rule with  $\mathcal{R}$  with a “simple” PA description, it corresponds to replacing  $\Phi_{\text{Condorcet}}$  in the proof above, thus yielding fixed-parameter tractable algorithms for all such rules as well.

## 4.2 Application: Resilient and Robust Moves

Theorem 1.1 allows us to develop fixed-parameter algorithms for problems related to moves in society. Problem ROBUST MOVE asks for a move that is robust to any adversary move of cost at most  $B_a$ :

### ROBUST MOVE

**Input:** A society  $\mathbf{s}$ , move costs vectors  $\mathbf{c}$  and  $\mathbf{c}_a$ , a winning condition  $\Psi$ , budgets  $B, B_a \in \mathbb{N}$   
**Find:** A  $(\mathbf{c}, B)$ -move  $\mathbf{m}$  such that for every adversary  $(\mathbf{c}_a, B_a)$ -move  $\mathbf{m}_a$ ,  $\Psi(\mathbf{s} + \Delta(\mathbf{m}_a) + \Delta(\mathbf{m}))$  holds.

The second problem, RESILIENT BUDGET, asks if a budget  $B$  suffices to counter any adversary move of cost at most  $B_a$ . Crampton et al. [5] consider a specialization of this problem for  $\mathcal{R}$ -SWAP BRIBERY.

### RESILIENT BUDGET

**Input:** A society  $\mathbf{s}$ , move costs vectors  $\mathbf{c}$  and  $\mathbf{c}_a$ , a winning condition  $\Psi$ , budgets  $B, B_a \in \mathbb{N}$   
**Find:** Does for every adversary  $(\mathbf{c}_a, B_a)$ -move  $\mathbf{m}_a$  exist a  $(\mathbf{c}, B)$ -move  $\mathbf{m}$  such that  $\Psi(\mathbf{s} + \Delta(\mathbf{m}_a) + \Delta(\mathbf{m}))$ ?

**THEOREM 4.7.** *ROBUST MOVE and RESILIENT BUDGET with  $\Psi \in \mathcal{P}_{0, (n_0), \delta, \gamma, \alpha, \beta}$  can be solved in time  $g(\tau, n_0, \delta, \gamma)(\alpha + \beta + B + B_a + \|\mathbf{c}\|_\infty + \|\mathbf{c}_a\|_\infty)^{O(1)}$ , that is, FPT parameterized by  $\tau + n_0 + \delta + \gamma$ .*

**PROOF.** We apply Theorem 1.1 to decide the following formulas, which is clearly equivalent to deciding the problems at hand:

$$\begin{aligned} \Psi(\mathbf{s})_{RB} &\equiv \exists \mathbf{m} : \text{feasible}(\mathbf{s}, \Delta(\mathbf{m})) \wedge (\mathbf{c}, B)\text{-move}(\mathbf{m}) \wedge \\ &\quad \forall \mathbf{m}_a : \text{feasible}(\mathbf{s} + \Delta(\mathbf{m}), \Delta(\mathbf{m}_a)) \wedge (\mathbf{c}_a, B_a)\text{-move}(\mathbf{m}_a) \wedge \\ &\quad \Psi(\mathbf{s} + \Delta(\mathbf{m}) + \Delta(\mathbf{m}_a)) \\ \Psi(\mathbf{s})_{RM} &\equiv \forall \mathbf{m}_a : \text{feasible}(\mathbf{s}, \Delta(\mathbf{m}_a)) \wedge (\mathbf{c}_a, B_a)\text{-move}(\mathbf{m}_a) \wedge \\ &\quad \exists \mathbf{m} : \text{feasible}(\mathbf{s} + \Delta(\mathbf{m}_a), \Delta(\mathbf{m})) \wedge (\mathbf{c}, B)\text{-move}(\mathbf{m}) \wedge \\ &\quad \Psi(\mathbf{s} + \Delta(\mathbf{m}) + \Delta(\mathbf{m}_a)) \end{aligned} \quad \square$$

## 5 POLYNOMIALLY MANY TYPES

We now prove Theorem 1.4, which formally reads:

**THEOREM 5.1.** *Let  $\Psi(\mathbf{s}) \in \mathcal{P}_{0, (\tau, n_1), \delta, \gamma, \alpha, \beta}$  be a winning condition. MINIMUM MOVE can be solved in time  $g(\gamma, \alpha)(\tau + \delta)^{O(1)} \log(\mathbf{s})$  for any linear function  $f(\mathbf{m}) = \mathbf{c}^\top \mathbf{m}$ , where  $g$  is some computable function.*

**PROOF.** Since  $k = 0$ ,  $\Phi(\mathbf{y})$  is a disjunction of  $\delta$  linear systems  $A_i \mathbf{y} \leq \mathbf{b}_i$ , with  $\|A_i\|_\infty \leq \alpha$  and with at most  $\gamma$  rows, for every  $i \in [\delta]$ . Thus we instead solve  $\delta$  instances of MINIMUM MOVE with  $\Phi_i(\mathbf{y}) \equiv A_i \mathbf{y} \leq \mathbf{b}_i$  and pick the best solution among them. So from now on assume that  $\Phi(\mathbf{y}) \equiv A \mathbf{y} \leq \mathbf{b}$  with  $\|A\|_\infty \leq \alpha$  and  $A \in \mathbb{Z}^{\gamma \times \tau}$ .

Observe that  $A$  can have at most  $\alpha^{O(\gamma)}$  different columns. For two types  $i, j \in [\tau]$ , we say they are equivalent and write  $i \sim j$  if the columns  $A_i$  and  $A_j$  are identical. Thus, the  $\tau$  types of people fall into  $C \leq \alpha^{O(\gamma)}$  equivalence classes. For every type  $i \in [\tau]$ , let  $C[i] = \{j \in [\tau] \mid j \sim i\}$  be the equivalence class containing  $i$ , and let  $\bar{A} \in \mathbb{Z}^{\gamma \times C}$  be a matrix with, for every  $i \in [C]$ ,  $\bar{A}_i = A_j$  where  $j \in C[i]$ . Now, for every type  $i \in [\tau]$ , we shall create a *reduced custom move costs vector*  $\mathbf{c}^i \in \mathbb{N}^{C^2}$ . For every  $j \in [C]$ ,  $i \neq j$ , let

$$c_{i,j}^i = \min_{j' \in C[j]} c_{i,j'}$$

be the cost of moving from  $i$  to the cheapest equivalent of  $j$ , and let  $c_{j,k}^i = +\infty$  for any  $i \neq j, k \in [C]$ . Let  $\bar{\mathbf{c}} = (\mathbf{c}^1, \dots, \mathbf{c}^\tau)$

Then, consider the following ILP with variables  $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^\tau) \in \mathbb{Z}^{\tau C}$ ; we obtain the minimum move  $\mathbf{m}$  from its optimal solution by taking  $m_{i,j} = x_{j'}^i$ , where  $j' \sim j$ :

$$\min \bar{\mathbf{c}}^\top \mathbf{x} \quad \text{s.t.} \quad \sum_{i=1, \dots, \tau} \bar{A} \mathbf{x}^i \leq \mathbf{b}, \quad \sum_{j=1, \dots, C} x_j^i = s_i \quad \forall i \in [\tau].$$

This ILP is a *combinatorial pre- $n$ -fold IP*; by Knop et al. [16] (detailed in [17, Corollary 23]), it can be solved in the claimed time.  $\square$

## 6 DISCUSSION

We raise three important questions which naturally arise from this work. First, we ask whether our fixed-parameter algorithm for {Dodgson, Young}-SWAP BRIBERY with uniform cost extends to general cost functions. This parallels [2, Challenge #2]. For much simpler voting rules an analogous result was shown only recently [18]. We believe that if the answer is positive, proving it would require providing new powerful integer programming tools, in particular, some analogue of Theorem 1.1 for  $n$ -fold integer programming, which is the engine behind the recent progress [18].

Second, can the run time of our algorithm be improved? This analogously recalls [2, Challenge #1]. The run time of Corollary 4.6 is double-exponential in the dimension  $O(\tau)$ ; thus Theorem 1.2 shows that Dodgson-SWAP BRIBERY is solvable in triple-exponential time in parameter  $|C|$ . We believe it can be improved, but we are sceptical that it could be made single-exponential, and thus ask if a double-exponential lower bound holds. A related question is to show lower bounds separating the complexity of  $\mathcal{R}$ -SWAP BRIBERY for different voting rules such as Borda, Copeland and Kemeny.

Third, we note that, unlike most previous results, the run time of our algorithm depends polynomially on the number  $|V|$  of voters. Many previous results solve the *succinct* variant of the problem (cf. Falizewski et al. [9]) and depend polynomially only on  $\log |V|$ . Thus we ask whether {Dodgson, Young}-SWAP BRIBERY is fixed-parameter tractable also in the succinct variant.

Finally, we discuss our usage of Presburger arithmetic as a generalization of ILPs. Solving ILPs amounts to deciding  $\exists \mathbf{x} : A \mathbf{x} \leq \mathbf{b}$ ; by Lenstra's algorithm [20] and its improvements by Kannan [15] and Frank and Tardos [12], this task is fixed-parameter tractable parameterized by the dimension of  $\mathbf{x}$  even for unbounded  $\|A, \mathbf{b}\|_\infty$  and if  $A$  has polynomially (in the length of the input) many rows. In 1990, Kannan claimed to show that PARAMETRIC ILP, which amounts to deciding  $\forall \mathbf{b} \in Q \exists \mathbf{x} : A \mathbf{x} \leq \mathbf{b}$  for some polyhedron  $Q$ , is fixed-parameter tractable parameterized by the dimension of  $\mathbf{x}$ ; here,  $\|A\|_\infty$  must be bounded by a polynomial and the number of rows of  $A$  also has to be a parameter. However, Kannan's result relies on Kannan's *Partitioning Theorem* (KPT), which was recently *disproved* by Nguyen and Pak [21]. Nguyen and Pak [21, Theorem 1.9] state that Woods [24] gave a polynomial-time algorithm for deciding  $\forall \mathbf{y} \exists \mathbf{x} : \Phi(\mathbf{x}, \mathbf{y})$  when the dimensions of  $\mathbf{x}$  and  $\mathbf{y}$  are *constant*; however, it is unclear if this is a fixed-parameter algorithm. For this reason, as we have the dimensions of  $\mathbf{x}$  and  $\mathbf{y}$  as (non-constant) *parameter*, we chose to prove Theorem 1.1 as a slightly weaker result (still sufficient for our purposes) but using only elementary techniques.



## REFERENCES

- [1] John J. Bartholdi III, Craig A. Tovey, and Michael A. Trick. 1989. Voting schemes for which it can be difficult to tell who won the election. *Soc. Choice Welfare* 6, 2 (1989), 157–165.
- [2] Robert Bredereck, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, and Gerhard J. Woeginger. 2014. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Sci. Tech.* 19, 4 (2014), 358–373.
- [3] Robert Bredereck, Jiehua Chen, Sepp Hartung, Stefan Kratsch, Rolf Niedermeier, Ondřej Suchý, and Gerhard J. Woeginger. 2014. A multivariate complexity analysis of lobbying in multiple referenda. *J. Artificial Intelligence Res.* 50 (2014), 409–446.
- [4] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. 2016. Complexity of shift bribery in committee elections. In *Proc. AAAI 2016*. 2452–2458.
- [5] Jason Crampton, Gregory Gutin, Martin Koutecký, and Rémi Watrigant. 2017. Parameterized resiliency problems via integer linear programming. In *Proc. CIAC 2017 (Lecture Notes Comput. Sci.)*, Vol. 10236. 164–176.
- [6] Britta Dorn and Ildikó Schlotter. 2012. Multivariate complexity analysis of swap bribery. *Algorithmica* 64, 1 (2012), 126–151.
- [7] Friedrich Eisenbrand and Gennady Shmonin. 2008. Parametric integer programming in fixed dimension. *Math. Oper. Res.* 33, 4 (2008).
- [8] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. 2009. Swap bribery. In *Proc. SAGT 2009 (Lecture Notes Comput. Sci.)*, Vol. 5814. 299–310.
- [9] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. 2009. How hard is bribery in elections? *J. Artificial Intelligence Res.* 40 (2009), 485–532.
- [10] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. 2011. Multimode control attacks on elections. *J. Artificial Intelligence Res.* 40 (2011), 305–351. Issue 1.
- [11] Piotr Faliszewski, Yannick Reisch, Jörg Rothe, and Lena Schend. 2014. Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting. In *Proc. AAMAS 2014*. 1357–1358.
- [12] András Frank and Éva Tardos. 1987. An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica* 7, 1 (1987), 49–65.
- [13] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. 1997. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *J. ACM* 44, 6 (1997), 806–825.
- [14] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. 2005. The complexity of Kemeny elections. *Theoret. Comput. Sci.* 349, 3 (2005), 382–391.
- [15] Ravi Kannan. 1987. Minkowski’s convex body theorem and integer programming. *Math. Oper. Res.* 12, 3 (Aug. 1987), 415–440.
- [16] Dušan Knop, Martin Koutecký, and Matthias Mnich. 2017. Combinatorial  $n$ -fold integer programming and applications. In *Proc. ESA 2017 (Leibniz Int. Proc. Informatics)*, Vol. 87. 54:1–54:14.
- [17] Dušan Knop, Martin Koutecký, and Matthias Mnich. 2017. *Combinatorial  $n$ -fold integer programming and applications*. Technical Report. <https://arxiv.org/abs/1705.08657>.
- [18] Dušan Knop, Martin Koutecký, and Matthias Mnich. 2017. Voting and bribing in single-exponential time. In *Proc. STACS 2017 (Leibniz Int. Proc. Informatics)*, Vol. 66. 46:1–46:14.
- [19] Matthias Köppe, Maurice Queyranne, and Chris T. Ryan. 2010. Parametric integer programming algorithm for bilevel mixed integer programs. *J. Optim. Theory Appl.* 146, 1 (2010), 137–150.
- [20] Hendrik W. Lenstra, Jr. 1983. Integer programming with a fixed number of variables. *Math. Oper. Res.* 8, 4 (1983), 538–548.
- [21] Danny Nguyen and Igor Pak. 2017. Complexity of short Presburger arithmetic. In *Proc. STOC 2017*. 812–820.
- [22] Jörg Rothe, Holger Spakowski, and Jörg Vogel. 2003. Exact complexity of the winner problem for Young elections. *Theory Comput. Syst.* 36, 4 (2003), 375–386.
- [23] Ildikó Schlotter, Piotr Faliszewski, and Edith Elkind. 2017. Campaign management under approval-driven voting rules. *Algorithmica* 77, 1 (2017), 84–115.
- [24] Kevin Woods. 2015. Presburger arithmetic, rational generating functions, and quasi-polynomials. *J. Symbolic Logic* 80, 2 (2015), 433–449.