Ann Math Artif Intell (2013) 68:161–193
DOI 10.1007/s10472-013-9359-5

Challenges to complexity shields that are supposed to protect elections against manipulation and control: a survey

Jörg Rothe · Lena Schend

Published online: 8 June 2013

Abstract In the context of voting, manipulation and control refer to attempts to influence the outcome of elections by either setting some of the votes strategically (i.e., by reporting untruthful preferences) or by altering the structure of elections via adding, deleting, or partitioning either candidates or voters. Since by the celebrated Gibbard–Satterthwaite theorem (and other results expanding its scope) all reasonable voting systems are manipulable in principle and since many voting systems are in principle susceptible to many control types modeling natural control scenarios, much work has been done to use computational complexity as a shield to protect elections against manipulation and control. However, most of this work has merely yielded NP-hardness results, showing that certain voting systems resist certain types of manipulation or control only in the worst case. Various approaches, including studies of the typical case (where votes are given according to some natural distribution), pose serious challenges to such worst-case complexity results and might allow successful manipulation or control attempts, despite the NP-hardness of the corresponding problems. We survey and discuss some recent results on these challenges to complexity results for manipulation and control, including typical-case analyses and experiments, fixed-parameter tractability, domain restrictions (single-peakedness), and approximability.

Keywords Computational social choice · Voting theory · Manipulation · Control

Mathematics Subject Classifications (2010) 68Q17 · 68T99

This work was supported in part by DFG grant RO-1202/15-1, SFF grant "Cooperative Normsetting" of HHU Düsseldorf, and a DAAD grant for a PPP project in the PROCOPE programme.

J. Rothe · L. Schend (✉)
Institut für Informatik, Heinrich-Heine-Univ. Düsseldorf, 40225 Düsseldorf, Germany
e-mail: schend@cs.uni-duesseldorf.de

J. Rothe
e-mail: rothe@cs.uni-duesseldorf.de

1 Introduction

In the emerging area of computational social choice, manipulation and control have been intensely studied from a computational point of view. Manipulation refers to strategic voting—voters who report untruthful preferences so as to either make their favorite candidate win or prevent their most despised candidate's victory. By the celebrated Gibbard–Satterthwaite theorem [64, 102] and other results expanding its scope (see, e.g., the work by Duggan and Schwartz [35]), all reasonable voting systems are manipulable in principle. This fact motivated Bartholdi and Orlin [3] and Bartholdi et al. [4] to study the complexity of manipulation problems in order to protect voting systems against manipulation via such complexity shields. Their work triggered a rich line of research on the computational aspects of manipulation problems, both regarding their classical complexity (see, e.g., the work of Conitzer and Sandholm [26], Conitzer et al. [28], Elkind and Lipmaa [38], Hemaspaandra and Hemaspaandra [66], Faliszewski et al. [54, 55], Betzler et al. [13], and Davies et al. [30]) and regarding other computational approaches (see, e.g., the work of Conitzer and Sandholm [27], Procaccia and Rosenschein [98], Friedgut et al. [58, 59], Dobzinsky and Procaccia [32], Isaksson et al. [74, 75], Mossel and Rácz [91], Xia and Conitzer [113, 114], and Zuckerman et al. [118]), and we will cover some of this work in this paper. Recent surveys by Conitzer [24], Faliszewski et al. [48, 52], Faliszewski and Procaccia [56] and Mossel and Rácz [90] summarize the state of the art in our ongoing "war on manipulation."

Electoral control, on the other hand, refers to attempts of an external actor, commonly called the "chair," to influence the outcome of elections by altering their structure. The common control types, each modeling some natural control scenario, include adding, deleting, and partitioning either candidates or voters. These control types have been introduced and studied by Bartholdi et al. [6] and Hemaspaandra et al. [70]. Although some voting systems are immune to some types of control (i.e., the chair never succeeds in making a favorite candidate win or prevent a most despised candidate's victory), many voting systems have been shown to be susceptible (i.e., not immune) to many control types. Again, this fact motivated Bartholdi et al. [6] to use complexity shields to protect voting systems against electoral control, which triggered many more results on the complexity of control problems (see, e.g., the papers by Hemaspaandra et al. [70, 71], Meir et al. [88], Faliszewski et al. [50, 51], Liu et al. [83], Liu and Zhu [84], Brandt et al. [18], Erdélyi and Fellows [39], Erdélyi et al. [44, 45] and Rothe and Schend [100]), and we will cover some of this work here. Recent surveys by Faliszewski et al. [52] and Baumeister et al. [7] summarize the state of the art in our ongoing "war on control."

The overwhelming number of results on using complexity shields against manipulation or control are NP-hardness results. NP-hardness of some given (decision) problem Y is usually shown by a polynomial-time many-one reduction from a problem X already known to be NP-hard. Such a reduction is implemented by a polynomial-time computable function r transforming any (yes or no) instance x of X into an instance $y = r(x)$ of Y such that $x \in X$ if and only if $y \in Y$.

Undoubtedly, P (*deterministic polynomial time*) and NP (*nondeterministic polynomial time*) are the best known and most central complexity classes; the annoyingly intractable "P = NP?" problem has been the most important open question of theoretical computer science for decades (see also Gasarch's "P vs. NP?" poll [62], whose tenth anniversary will be celebrated by conducting a *new* poll), and the many thousands of important problems that by now are known to be NP-complete (i.e., NP-hard and a member of NP) witness the centrality of the theory of NP-completeness [61]. Thus, the first thing to do when one encounters a seemingly hard problem is to try to prove its NP-hardness. However, NP is defined in the worst-case complexity model. All we can say about the computational hardness of NP-hard problems is that they are hard to solve on *some* instances, and that, unless P = NP, no polynomial-time heuristic can solve the problem correctly on all but a *sparse* set of instances (i.e., on all but a polynomial number of strings at each length), as has been conjectured by Schöning [103] and has been proven by Ogiwara and Watanabe [95] (for further insights and a more detailed discussion regarding this line of research, the reader is referred to the interesting survey by Hemaspaandra and Williams [72]).

This still leaves open the possibility that a polynomial-time heuristic might, for each input size k, correctly solve an NP-hard problem on exponentially (in k) many instances, and even on a vast majority of its instances. Note further that the instances that are really hard to solve might even never occur in practice. That is why manipulability and controllability of voting systems have recently also been investigated, both theoretically and experimentally, with respect to typical-case instances, seeking to circumvent NP-hardness of manipulation and control problems by showing that these problems can be solved by efficient heuristics, or are even exactly polynomial-time solvable for certain typical special cases. Many of these attacks on NP-hardness are very interesting, promising approaches. But are they a real threat? What can be concluded about their success in practice?

We survey and discuss some known results on various challenges to complexity results for manipulation and control, including typical-case analyses (such as some recently published results on analyzing control problems experimentally [100, 101]), fixed-parameter tractability, domain restrictions (single-peakedness), frequency analyses of manipulable instances, probabilistic analyses of degrees of strategyproofness, and approximability.

2 Elections and voting systems

An election is given by a set C of candidates and a list V of voters each having strict preferences over the candidates, which is also referred to as a *preference profile*. A *voting rule* (synonymously, *voting system*) is a function mapping any given preference profile (say, over the candidates in C) to a subset of C, the (possibly empty) set of winners. In social choice theory, such a mapping is termed a *social choice correspondence*. As is most common, preferences are represented by linear orderings, i.e., complete rankings of the candidates. This representation will be used for the following voting systems.

Scoring rules are a very central class of voting systems. For m candidates, a scoring rule is given by a scoring vector of nonnegative integers, $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$, such that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. Every candidate gets α_j points for each vote where c is ranked on the jth position, and all candidates with the most points are winners. Many important voting rules can be described by families of scoring vectors (one for each m). For example, *plurality* has scoring vectors of the form $(1, 0, \dots, 0)$; *veto* (a.k.a. *anti-plurality*) has scoring vectors of the form $(1, \dots, 1, 0)$; *Borda* has scoring vectors of the form $(m-1, m-2, \ldots, 1, 0)$ for m candidates; and k-approval has scoring vectors of the form $(\underbrace{1, \ldots, 1}_{k}, 0, \ldots, 0)$.

Single transferable vote (*STV*) proceeds in rounds, at most as many as there are candidates. Each candidate gets a point for each top position in the votes. In each round, if there is a candidate with a strict majority of points, she wins. Otherwise, a candidate with the smallest number of points is deleted (where ties, when they occur, are broken by a beforehand fixed tie-breaking rule), transferring her points to the candidate placed next, and the procedure is repeated until only one winner remains.[1]

Plurality with runoff proceeds in two rounds. In the first round, each candidate gets a point for each top position in the votes, and the two candidates with the most points (where ties again are broken by some rule if needed) move on to the second round, the "runoff," where their points are compared after deleting all other candidates. The candidate with the most points wins (again breaking ties if needed).

Other voting rules are based on pairwise contests between the candidates. In *Condorcet voting*, the winner is a candidate who is preferred to all other candidates by a strict majority of votes [22]. Note that Condorcet winners don't always exist (due to the *Condorcet paradox*), but when they exist, they are unique. A number of voting systems respect the Condorcet winner but avoid the Condorcet paradox, i.e., Condorcet winners win in these systems and there always is at least one winner. For example, in *maximin* (a.k.a. *Simpson's rule*), let $N(c, d)$ be the number of votes preferring c to d. The *Simpson score* of a candidate $c \in C$ is defined as $\min_{d \neq c} N(c, d)$, and all candidates with maximum Simpson score win. In *Copeland*$^\alpha$ *voting* with α being a rational number in $[0, 1]$, the score of a candidate $c \in C$ is defined as $\|\{d \in C - \{c\} \mid N(c, d) > N(d, c)\}\| + \alpha\|\{d \in C - \{c\} \mid N(d, c) = N(c, d)\}\|$, i.e., c gets one point for each d defeated and α points for each tie. This definition is due to Faliszewski et al. [51]. The original definition of *Copeland voting*, which is obtained by setting $\alpha = 1/2$, is due to Copeland [29]. Nanson's and Baldwin's rules are based on the Borda rule, but unlike this rule, they do respect the Condorcet winner. *Nanson's rule* proceeds in rounds, where in each round all candidates with less than the average Borda score are deleted, and this procedure is repeated until only one winner remains [92]. *Baldwin's rule* successively deletes a candidate with the lowest Borda score in each round until only a single candidate is left, again using a tie-breaking rule if needed [1].

The *ranked pairs* method, introduced by Tideman [107], is also based on pairwise comparisons of the candidates and proceeds in two steps to generate a complete ranking of the candidates: In the first step, all pairs of candidates (a, b) are ordered descendingly with respect to the number of voters preferring a to b. In the second step, beginning with an empty ranking, all pairs in the above ordering are considered one by one to be added to the ranking. If adding the candidates in the order they are given in the considered pair does not create a cycle, the candidates are added to the ranking. If a cycle would occur, the pair is disregarded and the next pair in the ordering is considered. When the ranking is completed, the top choice is the ranked pairs winner. Tideman suggested that whenever ties occur in either step, all candidates that

[1]Note that the complexity of winner determination in STV strongly depends on the tie-handling. As defined here, the procedure has a deterministic polynomial runtime. Without a fixed tie-breaking rule, however, winner determination can become NP-hard, see the work of Conitzer et al. [25]. A similar distinction has to be made for the ranked pairs method, as will be seen later on.

win for some tie-breaking rule are ranked pairs winners—a model Conitzer et al. [25] refer to as *parallel universes tie-breaking*. While this tie-handling ensures neutrality of this voting rule, which was the intention of Tideman's suggestion, it causes winner determination of this voting rule to be NP-hard, as shown by Brill and Fischer [20]. Breaking ties (whenever they occur) by a previously fixed tie-breaking rule does simplify winner determination to be in P, but only at the cost of neutrality.

Bucklin's rule proceeds in rounds (or levels). The *level i score* of a candidate $c \in C$ is defined as the number of votes from V that rank c among their top i positions. The *Bucklin score* of c is the smallest level i such that c's level i score strictly exceeds $\|V\|/2$. All candidates with a smallest Bucklin score, say j, and a largest level j score win.

Brams and Fishburn [16] proposed *approval voting*, a system that unlike the voting systems above doesn't expect linear preference orders as votes, but rather 0-1 vectors of length $\|C\|$: Every voter approves ("1") or disapproves ("0") of each candidate, and all candidates with the most approvals win. Brams and Sanver [17] proposed a hybrid system, called *fallback voting*, that combines Bucklin with approval voting as follows: All voters approve or disapprove of each candidate, and in addition rank their approved candidates (only those contribute to the level i scores). Every Bucklin winner in these partial rankings is also a fallback winner. However, if there exists none (due to disapprovals), then every approval winner is also a fallback winner.

3 Challenges to complexity results for manipulation

Manipulation in the context of voting refers to actions of voters who seek to make their favorite candidate win (in the *constructive* case introduced by Bartholdi et al. [4]) or to prevent their most despised candidate's victory (in the *destructive* case introduced by Conitzer et al. [28]), by reporting insincere preferences. Formally, for any voting system $\mathscr{E}$, *constructive coalitional weighted manipulation* is modeled by the following decision problem:

Constructive Coalitional Weighted Manipulation (CCWM)	
Given:	A set C of candidates, a list V of nonmanipulative voters over C each having a nonnegative integer weight, a list of the weights of the manipulators in S (whose votes over C are still unspecified) with $V \cap S = \emptyset$, and a distinguished candidate $c \in C$.
Question:	Can the preferences of the voters in S be set such that c is an $\mathscr{E}$ winner of $(C, V \cup S)$?

We assume that the strategic voters have complete knowledge of the sincere votes of all nonmanipulators. By asking whether c can be made *an* $\mathscr{E}$ winner of the election $(C, V \cup S)$ with strategic votes, we have defined the above problem in the so-called *co-winner* model (a.k.a. the *nonunique-winner* or simply the *winner* model). In this model, a manipulation is considered to be successful whenever the designated candidate wins, possibly among other winning candidates. By contrast, the *unique-winner* model requires that the designated candidate alone wins the manipulated election for a manipulation to be successful. Alternatively, some papers implicitly use the unique-winner model by requiring that ties among several winners, whenever they occur, are broken adversarially to the designated candidate (see, e.g., Remark 2.2 in the paper by Zuckerman et al. [118]), or they employ the co-winner model by requiring that ties among several winners, whenever they occur, are broken in favor of the designated candidate. As most results we cover in this section on manipulation are in the co-winner model, we use this model by default, and will explicitly state for which results the unique-winner model is used instead.

As special cases of the above problem, Constructive Coalitional Unweighted Manipulation (CCUM) is defined as an analogous problem, except that all weights are set to one, and constructive unweighted manipulation by a single manipulator (CUM) is defined by setting the coalition size $\|S\|$ in CCUM to one. The destructive variants of these problems are obtained by asking whether the preferences of the voters in S can be set such that c is *not* an $\mathscr{E}$ winner of $(C, V \cup S)$ in the co-winner model (respectively, such that c is *not* a unique $\mathscr{E}$ winner of $(C, V \cup S)$ in the unique-winner model).

Bartholdi et al. [4] were the first to analyze the manipulation complexity for voting systems, more precisely, the complexity of constructive unweighted manipulation by a single manipulator. They show that for a large group of voting systems including positional scoring rules, Copeland, and maximin, this problem is efficiently solvable by a simple greedy algorithm. Furthermore, they show NP-hardness of the CUM problem for second-order Copeland, where "second-order" refers to the way ties between several Copeland winners are broken. In a follow-up paper, Bartholdi and Orlin [3] investigate the STV system and show that it is NP-hard to manipulate by a single manipulator. Although the NP-hardness reduction presented in their paper is incorrect, their result holds true, since a small adjustment of the "garbage collector" candidates in some of the votes fixes this problem. The only other natural voting system with a polynomial-time winner determination that is known to have an intractable CUM problem is ranked pairs with a fixed tie-breaking rule, as shown by Xia et al. [116].[2] For the ranked pairs method (as originally defined by Tideman [107]), all variants of manipulation are trivially NP-hard, since winner determination is NP-hard.

As has been stated before, weighted coalitional manipulation has been introduced by Conitzer et al. [28], who show that for many voting rules this problem is NP-hard even for elections with few candidates. In particular, they show NP-hardness of CCWM for Copeland and maximin elections with at least four candidates and for STV elections with at least three candidates. For the latter system, they even show NP-hardness in the destructive case. For positional scoring rules except plurality, they also prove NP-hardness for elections with at least three candidates. Independently, Hemaspaandra and Hemaspaandra [66] established a similar result: CCWM is NP-hard for positional scoring rules defined by a scoring vector $\alpha' = (\alpha_1, \alpha_2 \ldots, \alpha_m)$ that satisfies $\|\{\alpha_i \mid 2 \leq i \leq m\}\| \geq 2$. This property is known as "diversity of dislike," since it means that not all "disliked" candidates (who are ranked behind the top candidate) are assigned the same number of points.

Faliszewski et al. [54, 55] study the unweighted case of coalitional manipulation and show NP-hardness for Copeland$^\alpha$ with $\alpha \in [0, 1] - \{0.5\}$, leaving the complexity of CCUM for Copeland$^{0.5}$ open. Xia et al. [116] show that CCUM is NP-hard for maximin and ranked pairs, and they extend their results to a certain family of positional scoring rules α^* (see their paper [115, p. 8] for the exact definition, which is

[2]Xia et al. actually showed this result for the version of ranked pairs where winner determination is NP-hard, but it also holds when ties are broken by a previously fixed tie-breaking rule, see [20, footnote 10].

somewhat involved). Recently, one of the most glaring open problems in this realm has been solved, independently by Betzler et al. [13] and Davies et al. [30]: Unweighted coalitional manipulation is NP-hard for the Borda rule, even for only two manipulators.

The complexity of manipulation in Bucklin was studied by Xia et al. [116], who provide a polynomial-time algorithm for this problem in the unweighted case, and by Reisch [99], who proves NP-completeness in the weighted case.

Table 1 gives an overview of the above-mentioned complexity results for constructive manipulation problems, and shows one open problem.

A manipulator facing an NP-hard problem, however, does not have to despair! After all, NP-hardness merely shows that this problem is hard to solve in the worst case, and the (often technically sophisticated) reductions used to prove NP-hardness usually provide very particular instances—elections that are unlikely to appear in real-world elections. In this section we survey some of the recent approaches to and advances in tackling NP-hard manipulation problems, including approaches that apply to "typical-case" elections.

3.1 Efficient heuristics for junta distributions

One of the first typical-case challenges to NP-hard manipulation problems is due to Procaccia and Rosenschein [98], who introduced so-called "junta distributions" that focus much weight on hard problem instances and are very light on the remaining ones. Intuitively put, they argue that when a problem is easy to solve relative to a junta distribution, it will be easy to solve relative to every typical distribution. To achieve this goal in a more formal way, they define the notion of "heuristic polynomial time" relative to a junta. We state these notions only informally here. A distribution μ is said to be a *junta* if it satisfies the following properties:

1. **Hardness:** Given an NP-hard problem X, the restriction $X_{|\mu}$ of X to μ (i.e., $X_{|\mu}$ contains the members of X having positive probability weight under μ) is also NP-hard. That is, X's hardness is not hidden by μ putting zero weight on the hard instances.

Table 1 Overview of complexity results for constructive manipulation problems

	CCWM	CCUM	CUM
Bucklin	NP-complete	P	P
Second-order Copeland	NP-complete	NP-complete	NP-complete
Copeland$^{\alpha}$, $\alpha \in [0, 1] - \{0.5\}$	NP-complete	NP-complete	P
Copeland$^{0.5}$	NP-complete	?	P
Maximin	NP-complete	NP-complete	P
STV	NP-complete	NP-complete	NP-complete
Ranked pairs	NP-complete	NP-complete	NP-complete
Plurality with runoff	NP-complete	P	P
Veto	NP-complete	P	P
Borda	NP-complete	NP-complete	P
Family of scoring rules α^*	NP-complete	NP-complete	P
Scoring rules of type α'	NP-complete	NP-complete	P

α^* denotes the family of scoring rules r_{weird} defined by Xia et al. [115, p. 8], α' denotes scoring vectors defined by Hemaspaandra and Hemaspaandra [66, p. 12]

2. **Balance:** Asymptotically, the probability of x being a yes-instance of X is neither too close to zero nor too close to one.
3. **Dichotomy:** There is a polynomial p such that for all n and for all instances x of length n, either $\mu_n(x) \geq 2^{-p(n)}$ or $\mu_n(x) = 0$. That is, the probability weight of each instance that is nonzero exceeds a certain threshold.
4. **Symmetry:** If X is a manipulation problem for a voting system that expects linear preference orderings as votes, then for each nonmanipulator $v \in V$, for any two candidates a and b distinct from the distinguished candidate c, and for each position i in the votes, v ranks a and b at position i with the same probability. In this sense, all candidates distinct from c are treated symmetrically by μ.
5. **Refinement:** If X is a coalitional manipulation problem, then for any length n input string x with $\mu_n(x) > 0$, if all manipulators voted identically, the distinguished candidate c would not be a winner. (Note that Procaccia and Rosenschein [98] restrict themselves to the case of constructive manipulation.)

A *distributional problem* (X, μ) is a (decision or search) problem X on Σ^* paired with a *distribution function* $\mu : \Sigma^* \to [0, 1]$, i.e., μ is a nondecreasing function converging to one: $\mu(0) \geq 0$, $\mu(x) \leq \mu(y)$ for each x and y with x lexicographically preceding y, and $\lim_{x\to\infty} \mu(x) = 1$. Procaccia and Rosenschein [98] define a *heuristic polynomial-time algorithm for* (X, μ) to be a polynomial-time algorithm $\mathscr{A}$ for which there is a polynomial q of degree at least one and a constant $n_0 \in \mathbb{N}$ such that for each $n \geq n_0$,

$$\Pr_{\mu_n}[x \notin X \text{ if and only if } \mathscr{A} \text{ accepts } x] < \frac{1}{q(n)}. \tag{1}$$

Procaccia and Rosenschein [98] show that for each scoring rule with vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ satisfying $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_{m-1} > \alpha_m = 0$, there exists a junta distribution μ^* such that CCWM can be solved in heuristic polynomial time with respect to μ^*. Their heuristic polynomial-time algorithm proceeds greedily. Roughly speaking, it ranks the distinguished candidate on top of the votes of the manipulators, and in each iteration it ranks the remaining candidates by their current scores: a candidate with lowest current score is ranked highest. Their junta distribution $\mu^* = \{\mu_n\}_{n\in\mathbb{N}}$ is defined by the following sampling procedure:

1. For each manipulator $s \in S$, randomly and independently choose the weight of s to be a value in $[0, 1]$ (up to $\mathscr{O}(\log n)$ bits of precision).
2. For each candidate d distinct from the distinguished candidate c, randomly and independently choose the votes of the nonmanipulators such that the initial score of d (before the manipulators cast their votes) is in the range $[(\alpha_1 - \alpha_2)W,\ \alpha_1 W]$ (again, up to $\mathscr{O}(\log n)$ bits of precision), where W is the total weight of the manipulators.
   In addition, c is ranked last by each nonmanipulator.

Adapting this greedy algorithm appropriately, it can also be used to show similar results for other voting systems, such as the maximin and the Copeland system with respect to a junta distribution defined similar to μ^*.

These very interesting results have been discussed by Erdélyi et al. [43], who consider *basic* junta distributions (defined just as junta distributions but disregarding symmetry and refinement, as these two properties are specific to manipulation problems) for general NP-hard problems, i.e., not restricted to manipulation problems. They show that very many NP-hard sets, even problems such as SAT (when suitably encoded) that are widely believed to be "really" hard problems (not only in the worst case), can be solved in heuristic polynomial time with high probability weight of correctness with respect to basic junta distributions (in the sense of (1)). They conclude that "if one were to hope to effectively use on typical NP-complete sets the notion of juntas and of heuristic polynomial time with respect to juntas, one would almost certainly have to go beyond the basic three conditions and add additional conditions" [43, p. 3996]. That is to say that the notion of (basic) junta and the related notion of heuristic polynomial time is not really appropriate to show that NP-hard problems can be easy to solve on typical-case instances: These notions appear to be too broad and unspecific to distinguish "typically hard" from "typically easy" among the NP-hard problems. Nonetheless, Erdélyi et al. [43] stress that the approach of Procaccia and Rosenschein [98], which is restricted to manipulation problems only and isn't meant to speak to general NP-hard problems, is very interesting indeed and should be further pursued and appropriately refined.

Erdélyi et al. [43] also discuss the related but different notion of *average polynomial time*, which—somewhat misleadingly—has been used in the literature for typical-case studies, such as that of Procaccia and Rosenschein [98], that actually concern the frequency of correctness of heuristics with respect to underlying distributions. The theory of average-case complexity was initiated by Levin [81]; see also the surveys by Goldreich [65] and Wang [111, 112]. Crucially, average polynomial time refers to taking an average of running times over the inputs according to some underlying distribution such that this average running time is low: Informally put, AvgP is the class of distributional problems (X, μ) for which there is an algorithm $\mathscr{A}$ solving X such that the running time T of $\mathscr{A}$ is polynomial on the average with respect to distribution μ.

By contrast, heuristic polynomial time with respect to a junta refers to the probability weight according to some underlying distribution for which the heuristic is correct. Note that merely a probability weight of $1 - (1/q(n))$ is required for all except a finite number of length n inputs, for some polynomial q. As the remaining inputs with probability weight $1/q(n)$ have no guarantee as to whether the heuristic solves them correctly, they need to be solved by brute force, which requires exponential time and so destroys any hope of getting a real average polynomial-time algorithm; see the more detailed discussion in [41, Appendix C]. A related discussion can be found in [49, Section 6], see also [42, 73].

A related typical-case approach is due to Homan and Hemaspaandra [73] (see also [86]), who introduce the notion of "*frequently self-knowingly correct algorithm*."[3] They provide such an algorithm that (under certain plausible hypotheses) determines

[3]By the term *self-knowingly correct*, Homan and Hemaspaandra [73] refer to heuristic algorithms that can make errors, but that also "know" with certainty that some of their outputs are correct. When such an algorithm computes a function, it outputs either "definitely" or "maybe" in addition to the value computed, and all computed values with a "definitely" are guaranteed to be correct function values.

the Dodgson winner in a given election with high success probability according to the uniform distribution. This problem (albeit, of course, not a manipulation problem) is known to be NP-hard by a result of Bartholdi et al. [5], and even complete for the complexity class $P_{||}^{NP}$ that captures "parallel access to NP" [68].[4] Erdélyi et al. [42] compare the notion of frequently self-knowingly correct algorithm with average polynomial time, and show that, with respect to the uniform distribution, every (distributional) problem in AvgP has such an algorithm, whereas the converse, provably, does not hold.

3.2 Frequency and probability of manipulable instances

Contemporary to Procaccia and Rosenschein's work [98], Conitzer and Sandholm [27] proposed a different approach to analyzing the hardness of manipulation problems beyond worst-case complexity measures. They show that it is impossible to find a voting rule for which manipulation instances are "*usually*" hard to solve if the voting rule is required to satisfy certain conditions, namely that for a large fraction of manipulable instances it holds that (a) they are weakly monotonic,[5] and (b) the manipulators can make one of exactly two candidates win by strategic voting. While the first property is satisfied by all monotonic— and thus by many commonly used— voting rules, the second property might seem more arguable. However, the authors carefully justify its relevance both theoretically and empirically, and their empirical results show that many voting rules indeed satisfy the second property on instances that are generated according to the distribution model used.

In order to prove an impossibility result, the authors chose to use a stronger definition of the manipulation problem than the version originally suggested by Bartholdi et al. [4]. They formulate a search problem where the manipulators have to find *all* candidates that can be made winners by manipulating the election and the corresponding manipulation action for each candidate has to be computed. An election is said to be manipulable in this context if the manipulators can make *more than one* candidate win. For this definition of manipulation, the authors provide an efficient algorithm[6] that always finds a successful manipulation, provided that the given election satisfies the two properties (a) and (b).

[4] $P_{||}^{NP}$ is defined to be the class of problems solvable by a deterministic polynomial-time Turing machine that can access an NP oracle set in a "parallel" manner (technically speaking, via a so-called polynomial-time truth-table reduction [78]), i.e., it first computes all oracle queries and then asks them all at once in parallel and evaluates the answer vector returned by the oracle—in such a nonadaptive oracle access, new oracle queries do not depend on previously given oracle answers. By definition, $P_{||}^{NP}$ lies inbetween the first and the second level of the polynomial hierarchy. Hemaspaandra et al. [69] survey various results on raising NP-hardness to $P_{||}^{NP}$-hardness lower bounds.

[5] A voting rule is *weakly monotonic* if for every pair of candidates, a and b, either (1) b does not win for any manipulative votes, or (2) a does not win if all manipulators rank b first and a last.

[6] If winner determination for the considered voting rule is in P, then the algorithm runs in deterministic polynomial time as well.

Experimentally, the following monotonic voting rules are investigated: plurality, Borda, veto, Copeland, maximin, and all voting rules respecting the Condorcet winner. For these voting rules, all instances are monotonic, so the analysis is confined to the occurrence of the second property in manipulable instances. Nonmonotonic rules that are investigated and for which both properties have to be analyzed are STV and plurality with runoff.

To conduct the experiments, random instances have to be generated, and the authors chose the following distribution that is due to Condorcet [22]: Assuming a correct ranking of the candidates, the preference for each voter is constructed by deciding for each pair of distinct candidates the order they will have in the preference. With probability p they have the same order as in the correct ranking, and with probability $1 - p$ they are ranked the opposite way. When cycles occur, the vote has to be reconstructed. The distribution of the votes can be varied by the parameter p; for $p = 0.5$, for example, the generated votes are independently and uniformly distributed. For each election size, 1,000 manipulable instances are generated where the number of candidates varies from three to five, the number of nonmanipulative voters is between zero and at most 600, and the number of manipulators is either one or five (these are the election sizes for which results are presented in the paper). For the distribution parameter p, the values 0.5 and 0.6 are chosen. The decision whether a generated instance is manipulable, and thus can be used for further testing, is based on testing sufficient (but not necessary) conditions, and thus the results of the experiments can only give a lower bound.

The results show that in instances with three candidates where one manipulator can be successful, the fraction of instances satisfying both properties goes to one if the distribution parameter is $p = 0.6$, and is still very large when varying it to $p = 0.5$. Increasing the number of manipulators to five shows essentially the same results, but the values for smaller nonmanipulative voter sets are smaller. The same can be observed when the candidate set is enlarged to five candidates, except that the STV rule breaks out by showing very small values whenever property (b) is satisfied.

Since this analysis is based on a stronger definition of manipulation than that due to Bartholdi et al. [4], these results can only give a lower bound for the more common definition of manipulation. As for all experimental studies, further work could be done, for example, by conducting more experiments with larger elections and different vote distributions.

Friedgut et al. [58, 59] introduce a "quantitative version" of the famous Gibbard–Satterthwaite theorem. Their work is motivated by the question about the fraction of profiles needed for a profitable manipulation to actually exist. Just as Conitzer and Sandholm [27], they also alter the definition of a successful manipulation: In this context, a manipulation attempt is *successful* (or *profitable*) if the winner of the manipulated election is more preferred by the manipulators than the winner of the original election. This implies that the winner of the altered election has not to be the top choice of the manipulators and that there is no designated candidate given in the instance that has to be made the winner. In this technically challenging work, the authors show the following very interesting result for neutral voting rules that are "far" from being a dictatorship (see footnote 8): Assuming that the preferences in a given election with three candidates are uniformly distributed, a single manipulator can successfully manipulate it with nonnegligible probability (i.e., with a probability that is at least inverse-polynomially small in the number of voters), by altering her preference to a randomly chosen one.

Dobzinsky and Procaccia [32] follow up this line of research and show that for Pareto-optimal voting rules,[7] the previous result by Friedgut et al. [58, 59] can be generalized to elections that have more than three candidates. The number of voters, however, is limited to two in this result. The authors strongly suspect that it can be generalized to larger voter sets.

Isaksson et al. [74, 75] solve one of the main open problems raised by Friedgut et al. [58], by giving a general quantitative version of the Gibbard–Satterthwaite theorem, which holds for more than three candidates and more than two voters when the voting rule is assumed to be neutral. More recently, Mossel and Rácz [91] were able to further generalize these previous results by proving that the neutrality assumption is in fact not needed: For voting rules on m candidates and n voters that are not neutral and ϵ-far from a dictatorship,[8] a voter profile chosen uniformly at random can be successfully manipulated with a probability of at least $\epsilon^{15}/10^{39}n^{67}m^{166}$.

Certainly, these approaches are still somewhat limited with respect to their implications for real-world elections, since some technical assumptions (e.g., regarding the distribution of preferences, the number of voters or candidates, or the properties of the voting rule used) that are required for the proof to work might not hold. Moreover, the notion of "nonnegligible probability" must be interpreted carefully: Even if, by definition, the probability of a successful random manipulation is large enough that it cannot be disregarded technically, it still might be too small to be practically relevant in real-world elections.

Based on the work of Procaccia and Rosenschein [97], Xia and Conitzer [113] analyze the probability of a successful manipulating coalition depending on the number of voters and manipulators. They generalize previous work (see also the studies of Slinko [104–106]) by showing results for a new class of voting rules, so-called *generalized scoring rules*, which contain many commonly used voting rules such as STV, Copeland, positional scoring rules, and ranked pairs.

Assuming that the preferences of the nonmanipulative voters are independently and identically distributed, they show that the probability that a random election can be manipulated by $\mathcal{O}(n^p)$ manipulators (where n is the total number of nonmanipulative and manipulative voters in the election and $0 \leq p \leq 1/2$) is $\mathcal{O}(n^{p-1/2})$, and thus very small. On the other hand, if the manipulative coalition is of size $o(n)$ and $\Omega(n^p)$ for $1/2 < p < 1$, then the manipulators can make every candidate win with probability $1 - \mathcal{O}(e^{-\Omega(n^{2p-1})})$, which implies that manipulative coalitions of this size are all-powerful. For coalition sizes of order $\sqrt{n}$, no results regarding the probability of manipulable instances are known.

3.3 Approximation algorithms

Another approach to challenge NP-hard manipulation problems in practice is due to Zuckerman et al. [118], who reformulate this as an optimization problem in

[7] A voting rule $\mathcal{E}$ is said to be *Pareto-optimal* if, for all elections, it holds that a candidate a is not an $\mathcal{E}$ winner whenever there exists a candidate b that is ranked before a in every vote.

[8] A voting rule $\mathcal{E}$ is said to be *ϵ-far from a dictatorship* if the fraction of uniformly chosen voter profiles on which $\mathcal{E}$ differs from dictatorial voting rules is at least ϵ.

the constructive, unweighted case for coalitions of manipulators and then seek to approximate a solution to this optimization problem:

Constructive Coalitional Unweighted Optimization (CCUO)	
Input:	A set C of candidates, a list V of nonmanipulative voters over C, and a distinguished candidate $c \in C$.
Output:	The minimal n such that a coalition S of size n of (unweighted) manipulators can make c the unique winner in $(C, V \cup S)$ by casting insincere votes.

We state this problem in the unique-winner model by default, but note that some authors investigate it in the co-winner model. We will always explicitly state when the presented results are in the co-winner model instead of the unique-winner model.

Zuckerman et al. [118] investigate this problem for scoring rules, Borda, and maximin, and they design greedy algorithms that approximate the corresponding problems within a certain factor, i.e., they analyze the algorithms' windows of error. In particular, they show with a series of involved technical proofs that CCUO can be efficiently approximated up to an additive constant of 1 for Borda (i.e., one additional manipulator suffices) by an algorithm called Reverse. We illustrate this algorithm in the following example.

Example 1 We have a Borda election (C, V) with seven candidates, $C = \{a, b, c, d, e, f, g\}$, and five voters with the following preferences. Three voters vote $g > f > e > d > c > a > b$ and two voters have the preference $c > d > e > f > g > a > b$. The Borda scores of the candidates in (C, V) are shown in the first row of Table 2 and we see that candidate g is the unique Borda winner in this election.

Suppose that the manipulators want candidate a to be the unique Borda winner. The algorithm Reverse then constructs the manipulators' votes consecutively as follows. The distinguished candidate a is always positioned in first place, while the remaining candidates are ranked in ascending order with respect to their current Borda scores. So the first manipulator m_1 constructed by the algorithm has the preference $a > b > c > d > e > f > g$. We set $S_1 = \{m_1\}$ and compute the Borda scores in the new election $(C, V \cup S_1)$ (see row 2 of Table 2).

Since the distinguished candidate is not yet a unique Borda winner, a second manipulator m_2 has to be added with a preference constructed as above. This preference might be, e.g., $a > b > c > d > e > f > g$. Note that the candidates $c, d, e, f,$ and g can be ordered arbitrarily behind candidates a and b, since they all have the same

Table 2 Borda scores over different voter sets constructed by the algorithm Reverse (rows 1 to 6) and the scores in an optimally manipulated election (row 7)

	a	b	c	d	e	f	g
(C, V)	5	0	18	19	20	21	22
$(C, V \cup S_1)$	11	5	22	22	22	22	22
$(C, V \cup S_2)$	17	10	26	25	24	23	22
$(C, V \cup S_3)$	23	15	26	26	26	26	26
$(C, V \cup S_4)$	29	20	30	29	28	27	26
$(C, V \cup S_5)$	35	25	30	30	30	30	30
$(C, V \cup S')$	29	20	28	28	28	28	28

Borda score. We add this manipulator m_2 to the set of manipulators, denoted here by $S_2 = S_1 \cup \{m_2\}$, and compute again the Borda scores in the election $(C, V \cup S_2)$ (see row 3 in Table 2).

The algorithm continues constructing manipulators in this way until the distinguished candidate is the unique winner. In this example, five manipulators $m_1, \ldots, m_5$ with the following preferences are constructed:

$$m_1 : a > b > c > d > e > f > g \qquad m_4 : a > b > c > d > e > f > g$$
$$m_2 : a > b > c > d > e > f > g \qquad m_5 : a > b > g > f > e > d > c$$
$$m_3 : a > b > g > f > e > d > c$$

The scores in elections $(C, V \cup S_3)$ to $(C, V \cup S_5)$ are shown in rows 4 to 6 in Table 2. We see that candidate a is the unique Borda winner in election $(C, V \cup S_5)$, but a was not a unique Borda winner in an earlier stage of the construction. Thus the algorithm computes a successful manipulation with five manipulators.

However, for this election the following four manipulating voters $S' = \{m'_1, m'_2, m'_3, m'_4\}$ would actually suffice to exert a successful manipulation:

$$m'_1 : a > b > c > d > e > f > g \qquad m'_3 : a > b > c > d > e > f > g$$
$$m'_2 : a > b > f > g > c > d > e \qquad m'_4 : a > b > e > g > d > f > g$$

As can be seen in row 7 of Table 2, candidate a is the unique Borda winner in the election $(C, V \cup S')$. This example shows that Reverse might construct manipulating coalitions with one more manipulator than optimally needed.

Similarly, manipulation in maximin can be efficiently approximated within a factor of 2 (i.e., doubling the optimal number of manipulators suffices in the worst case).

On the other hand, Zuckerman et al. provide algorithms that solve the unweighted decision problem CCUM efficiently for plurality with runoff and for veto.

Note that these algorithms also apply to the weighted case and make errors only on very few configurations of voters' weights. In particular, the approximation algorithm for Borda improves the error analysis implicit in the above-mentioned more general result that Procaccia and Rosenschein [98] achieve for a certain familiy of scoring rules and CCWM with respect to junta distributions when tailored to Borda only (which, of course, satisfies the required condition $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_{m-1} > \alpha_m = 0$).

By reducing the manipulation problem to a scheduling problem, Xia et al. [115] were able to approximate both the weighted and the unweighted manipulation problem for general positional scoring rules to an additive term of $m - 2$, i.e., the algorithm proposed finds a successful manipulation (if some exists for the given election) with at most $m - 2$ additional manipulators.

Davies et al. [30] propose two other approximation algorithms, Largest Fit and Average Fit, to find optimal manipulations for the Borda rule and compare them both theoretically and experimentally (the latter approach will be treated in more detail later on) with the above-mentioned algorithm Reverse introduced by Zuckerman et al. [118]. Largest Fit and Average Fit use ideas from bin packing and multiprocessor scheduling and solve the manipulation problem in the co-winner model. The theoretical analysis shows that both are incomparable with Reverse in the sense that an infinite family of instances can be found where Reverse performs better than either of them, and in turn there is an infinite familiy of instances where REVERSE does not find the optimal solution but the other algorithms do.

Zuckerman et al. [117] improve the approximation error of 2 for maximin in CCUO to a factor of $5/3$. In addition, they prove that no approximation factor for this problem can be better than $3/2$, unless P = NP. A $5/3$-approximation has great theoretical value, but for practical applications this factor might be unsatisfying. Together with the inapproximability result one can conclude that, from a practical point of view, approximation does not seem to be appropriate to attack maximin's resistance to manipulation. In particular, these results show that maximin in some sense is "safer" against manipulation attempts than Borda.

In general, however, this approach is very promising and should definitely be pursued and further explored, for example, for other voting systems. In particular, additional inapproximability results such as the one mentioned in the previous paragraph would be especially beneficial, as they add to the evidence of protection complexity theory can provide against manipulation.

A more general approach for approximating manipulation problems is presented by Brelsford et al. [19]. They use a different objective function in their problem definition, leading to a different definition of successful manipulation attempts. In this setting, a *legal manipulation* is a manipulation that respects the parameters given in the instance such as, for example, weight constraints for the manipulators in a weighted election. Here, a manipulation is said to be successful if the "performance" of the distinguished candidate is improved by the manipulators' strategic votes, and the goal is to maximize the increase of the distinguished candidate's performance. The *performance* of a candidate is defined to be the difference between her score and the maximal score of any other candidate. Thus, in this setting it is not crucial that the designated candidate wins in the resulting election (considering the constructive case); rather, she has to be "closer to winning" than in the original election. For this more general definition of manipulation, Brelsford et al. [19] provide both approximability and inapproximability results for the family of scoring protocols. For those scoring rules whose scoring vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ satisfies the condition $\alpha_1 > \alpha_2$, the authors give a *fully polynomial-time approximation scheme* (FPTAS) for the weighted coalitional manipulation problem, defined as the above maximization problem. That is, for each rational number ε with $0 < \epsilon < 1$, there exists a $(1-\varepsilon)$-approximation algorithm A_ε for this maximization problem that, given an instance I of the manipulation problem, provides a legal solution (that is, a legal manipulation) and runs in time polynomial both in $|I|$ and in $1/\epsilon$. On the other hand, for the veto rule, k-approval (equivalently, $(m-k)$-veto for m candidates), and generalized versions of k-approval[9] with $k \geq 2$, they show that, unless P = NP, no FPTAS can exist for this variant of weighted coalitional manipulation.

Procaccia [96] suggests a completely different approach to protect elections against manipulation by using approximation methods. The idea is not to analyze the approximability of the manipulation problem for a given voting system but to

[9] Generalized variants of k-approval are defined by scoring vectors $(\underbrace{\alpha_1, \ldots, \alpha_k}_{k}, 0, \ldots, 0)$ where values other than 1 are allowed for $\alpha_1, \alpha_2, \ldots, \alpha_k$ as long as they satisfy $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_k$.

secure the voting system itself against manipulation by approximating it with a strategyproof (i.e., nonmanipulable) randomized voting rule. Procaccia [96] proposes such randomized voting rules providing approximations of score-based voting systems. A strategyproof randomized voting rule f is said to *approximate a given score-based voting rule within an approximation ratio of* γ if the expected score of the winner that f chooses is at least $\gamma \cdot s$, where s is the maximal score. It is shown that for m candidates, positional scoring rules can be approximated by strategyproof randomized voting rules within a factor of $\Omega(1/\sqrt{m})$, and that for plurality voting this is, asymptotically, the best approximation possible. For the Borda rule, on the other hand, it is proven that a $((1/2) + \Omega(1/m))$-approximation can be achieved. Furthermore, Copeland$^\alpha$ and maximin are analyzed. Interestingly, maximin cannot be approximated nontrivially, which means that no strategyproof randomized voting rule exists that provides a better approximation of maximin than the trivial approximation, namely choosing a winner at random. For Copeland$^\alpha$ with $\alpha \in [1/2, 1]$, a lower bound of $1/2 + \Omega(1/m)$ can be shown, whereas for $\alpha \in [0, 1]$ the analysis provides an upper bound of $1/2 + \mathcal{O}(1/m)$. Certainly, in real-life elections, human electorates might have difficulties accepting randomized voting rules, so this approach is clearly limited. But then again, in many other applications of voting systems, for example, multiagent systems, where the computational aspects of manipulation are especially important, since the software agents (the voters) have computational power on their own, acceptance of randomized voting rules is not an issue at all.

3.4 Fixed-parameter tractability

What is "typically" the source of complexity of NP-hard manipulation problems? Does the problem remain hard when there are a fixed number of candidates or voters or when the number of manipulators is bounded by a constant? Parameterized complexity theory—a field pioneered by Downey and Fellows [34], see also [57, 94]—considers questions like that.

While classical complexity theory measures the complexity of problems only in terms of input size, parameterized complexity analyzes a problem's complexity in terms of the input parameters. Typical input parameters for manipulation problems are the number of candidates, the number of voters, and the number of manipulators.[10] Accordingly, such a multivariate analysis seeks to confine to the parameters the seemingly unavoidable combinatorial explosion in NP-hard problems. Technically, a parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$, whose instances are pairs (I, k) such that k is a parameter, is *fixed-parameter tractable* (or in the class FPT) if it can be decided in time $f(k)\|I\|^{\mathcal{O}(1)}$ whether (I, k) is a yes-instance, where f is a computable function of the parameter k, yet independent of $\|I\|$. Intuitively, if the parameter k of an NP-hard but fixed-parameter tractable problem is bounded by a small constant for instances typically occurring in practice, then this problem can be solved efficiently in practice.

[10] Already Bartholdi et al. [5], who proved that the Dodgson winner problem is NP-hard (see [68] for a stronger result), considered these parameters and proved that this problem can be solved in polynomial time when the number of candidates or voters is fixed; see also [11].

Downey and Fellows [34] also introduced concepts that are useful to capture parameterized intractability, including the notions of parameterized reducibility, parameterized hardness, and the W-hierarchy. Most important here are the first two classes of this hierarchy, W[1] and W[2]. We don't give the rather involved definition of these classes here, but refer to the literature [34, 57, 94]. An early treatise of parameterized complexity applied to problems from computational social choice is due to Lindner and Rothe [82]; a more recent one is due to Betzler et al. [10].

Bartholdi et al. [4] showed that when there is only a single manipulator, many unweighted voting systems—including plurality, Borda, Copeland, and maximin—can be manipulated by a greedy algorithm in polynomial time. STV, by contrast, is NP-hard to manipulate by a single strategic voter already in the unweighted case [3]. However, Conitzer et al. [28] proved that this problem, parameterized by the number m of candidates, is fixed-parameter tractable. Their FPT algorithm runs in time $\mathcal{O}(1.62^m \cdot p(n, m))$, where n is the number of votes and p a polynomial. As mentioned earlier, CCUM is NP-complete for Borda, even for only two manipulators [13, 30]. However, when parameterized by the number of candidates, this problem is fixed-parameter tractable. This follows from a more general approach of Betzler et al. [12] who study the parameterized complexity of the Possible Winner problem, which was introduced by Konczak and Lang [76] and generalizes the unweighted coalitional manipulation problem. This problem asks, given an election with only *partial* preference orderings, whether a distinguished candidate can be made win by extending the partial orders to complete orders.[11] Betzler et al. [13] observe another fixed-parameter tractability result for this problem and a different parameter related to "instance tightness."

While these results are very interesting from a theoretical point of view, one should keep in mind, first, that parameterized hardness shares with NP-hardness the feature of being a worst-case measure of complexity and, second, that many of the algorithms showing fixed-parameter tractability are too slow for practical purposes, even for small values of the parameter. Just as in classical algorithmics, an important task is to improve these algorithms.

Many manipulation problems have been classified in terms of classical complexity by now. A detailed multivariate complexity analysis focusing on various specific parameters occurring in practice is still missing for many manipulation scenarios and voting systems. In particular, the number of manipulators seems to be an appealing parameter, especially in light of the approximation algorithms described above that seek to approximate this number. It would also be interesting to obtain parameterized intractability results, such as W[1]- and W[2]-hardness, for certain manipulation problems in suitable parameterizations.

3.5 Single-peaked preferences

What is a "typical" election? Well, it depends. In general, it is a nontrivial problem to say how votes in an election are "typically" distributed. However, there are certain

[11] In addition, Betzler et al. [12] and Betzler [9] obtain FPT results for Possible Winner in Borda, k-approval, and Copeland$^{\alpha}$ voting for different parameters, such as the total number of undetermined candidate pairs. Dorn and Schlotter [33] establish FPT results for an even more general problem, Swap Bribery, which was introduced and studied in terms of classical complexity by Elkind et al. [37].

special cases that may occur in real-world elections and that may outright be easy to solve. For example, suppose society votes on a single issue (such as taxes or health care or war on terror, etc.) that can be nicely embedded in a left-right spectrum. That is, there exists a linear (societal) ordering of candidates on this spectrum (e.g., if taxes are the issue to be voted on, a left-wing candidate would stand for high taxes and a right-wing candidate would stand for low taxes), and relative to this linear ordering, all voters' preference utility curves raise to a single peak (representing this voter's most preferred position on this spectrum) and then fall, or just raise, or just fall. This model of *single-peaked* preferences is a central concept in political science and has been introduced by Black [15]; see also, e.g., [2, 60, 80] for more recent social-choice-theoretic work on single-peakedness.

Formally, an election (C, V) is said to be *single-peaked* if there exists a linear ordering L on C such that for each vote $v_i \in V$ (individually represented by a linear ordering $>_i$ on C) and for each triple of candidates, c, d, and e in C, if $c\,L\,d\,L\,e$ or $e\,L\,d\,L\,c$ then $c >_i d$ implies $d >_i e$ for each i. In other words, for each triple of candidates ordered according to L, it can never happen in an individual vote that the "middle" candidate is ranked last.

Restricting an electorate to only single-peaked preferences may or may not change the computational properties and the complexity of the associated manipulation problems. Such restrictions have only recently be considered (e.g., by Escoffier et al. [47] and Conitzer [23]) from a computational point of view, although the concept in political science is well-established for more than a half century now. In particular, Walsh [108] shows that the weighted manipulation problem for STV for at least three candidates remains NP-complete, even when the given election is restricted to be single-peaked. Here, the underlying linear ordering L of candidates relative to which the votes of the nonmanipulators are single-peaked is part of the input, and the manipulators' votes are supposed to be single-peaked relative to the same ordering L.

Faliszewski et al. [53] prove that, depending on the voting system used, NP-hardness of CCWM can vanish or can remain in place. For example, one of their results says that CCWM for 3-candidate Borda elections is in P when restricted to single-peaked electorates (whereas it is NP-complete in the unrestricted case), while it remains NP-complete for 4-candidate Borda elections, even when restricted to single-peaked electorates (just as in the general case).

Remarkably, for m-candidate 3-veto elections, this manipulation problem is in P whenever $m \leq 4$ or $m \geq 6$, but is NP-complete for $m = 5$ [53]. That is, due to single-peakedness, the complexity of the problem drops down to polynomial time, although the number of candidates is incremented from five to six or more.

Faliszewski et al. [53] also show that (for a certain artificial voting system) restricting the electorate to the single-peaked case may even increase the complexity of manipulation. In addition, they prove a dichotomy result for single-peaked electorates when a scoring rule with vector $\alpha = (\alpha_1, \alpha_2, \alpha_3)$ is used: CCWM is NP-complete whenever $\alpha_1 - \alpha_3 > 2(\alpha_2 - \alpha_3) > 0$; otherwise, it is in P. This dichotomy result has been generalized by Brandt et al. [18] to scoring rules with any (fixed) number of candidates. For further results, see the work of Faliszewski et al. [49].

But, again, one may wonder how "typical" single-peaked elections are in the real world. Just as only "few" (in the sense of subexponentially many) hard instances per length might cause a problem to be hard in the worst-case model (even though it might be easy to solve for all other instances), only a few bad votes might destroy single-peakedness for any societal ordering of candidates (although, for some such ordering, all other votes might be single-peaked). On the other hand, many researchers in political science feel that electorates often are "essentially" single-peaked, even though perhaps not completely. To capture that intuition formally, Faliszewski et al. [49] introduce the notion of *nearly single-peaked* electorates in various settings, making the notion more flexible and more broadly applicable. They prove, for example, that for nearly single-peaked societies the above manipulation problem for 3-candidate Borda, and also for 3-candidate veto, is NP-complete (just as in the general case). On the other hand, they also provide examples where going from single-peaked to nearly single-peaked electorates does not raise the complexity of the manipulative actions.

Mattei [85] empirically investigates huge data sets from real-world elections (drawn from the Netflix Prize data set) with respect to properties such as how likely the Condorcet paradox is to appear and how often single-peaked preference profiles are to occur. In particular, his experiments indicate that single-peaked preferences only very rarely occur in practice.

3.6 Experimental analysis

Another recent line of research studies experimental simulation and evaluates heuristics for solving NP-hard manipulation problems empirically. These investigations were initiated by Walsh [109, 110], who experimentally studied NP-hard manipulation problems for veto and STV, and showed that for many instances the elections generated can be manipulated quickly.

Generating elections for experimental analysis can be done in various ways depending on the electorates one wants to model. The possibility and frequency of successful manipulation can vary greatly for different vote distributions.

For the veto rule, Walsh [109] investigates the problem CCWM restricted to elections with three candidates, which can be directly reduced to 2-Way-Number Partitioning. This reduction allows to make use of known efficient algorithms for the latter problem, such as the CKK algorithm by Korf [77], to solve the manipulation problem.

The votes in the tested elections are generated by randomly choosing one of the three candidates to be vetoed, and the vetoes carry randomly drawn weights as well. The electorate's distribution is varied by the generation of the voters' weights. To generate uniform votes, the weights of the voters are drawn uniformly and independently at random from a given interval. Similarly, normally distributed votes are generated by drawing the voters' weights independently from a normal distribution.

The weight of the manipulating coalition is crucial for the complexity of determining whether a given election is manipulable or not. If the weight is too small, the coalition is hopeless, whereas manipulation is trivially possible if the weight is too big. Instances where the manipulative coalition's weight is between these trivial cases are conjectured to be the hard ones, so this so-called "critical" region is of most interest in the results obtained from the conducted experiments.

Similar results were found for uniformly and for normally distributed votes: Even in the critical region, the decision whether the tested election is manipulable or not could be made with low computational costs. The probability curves for successful manipulation show in both distributions a smooth phase transition in this critical area, similarly to the phase transition observed for polynomial-time solvable decision problems.

Complementary to this setting, electorates with correlated votes have been investigated as well. For the veto system, so-called "hung" elections were generated where the manipulative coalition is twice as heavy as the nonmanipulative voters' total weight, but all nonmanipulators veto the distinguished candidate the coalition wants to make win. This finely balanced situation is exactly what the reduction of Conitzer et al. [28] produces from a given Partition instance in their proof of NP-hardness of the manipulation problem. Not surprisingly, generating such instances at random leads to higher computational costs for deciding them in the critical region. Furthermore, the probability curves resulting from these instances show a typical sharp phase transition, similar to that of other hard decision problems, namely around $(\log k)/m \approx 1$, where m is the number of manipulators and their weights are randomly chosen from $(0, k]$. Interestingly, one randomly vetoing voter in an otherwise perfectly hung election suffices to at least empirically make the problem easier.

Furthermore, the results show that in elections with uniform votes the sizes of the voters' weights do not influence the manipulability of the tested elections, confirming empirically the theoretical conjectures by Procaccia and Rosenschein [97] and Xia and Conitzer [113].

For preference-based voting systems such as STV and Borda, the votes are given by linear orders over all candidates and thus are permutations of the candidates. To generate uniformly distributed votes for these voting systems, each vote is drawn uniformly and independently from an urn containing all possible votes. This is the so-called *Impartial Culture* (*IC*) model in which each vote is equally likely to occur. To model correlated votes, Walsh [110] uses the *Pólya–Eggenberger urn model* (*PE*)—described, e.g., by Berg [8] —in the following sense: The first vote is drawn from an urn containing all possible votes; there are $m!$ different permutations of the m candidates. Before drawing the second vote, $m!$ votes identical to the first one are put back to the urn. Before drawing the third vote, $m!$ votes identical to the second vote are put back to the urn, and so on.[12]

In the literature of social choice theory, the PE model is the standard model to generate electorates with different levels of homogeneity and has been used to investigate the correlation between various interesting properties of voting rules such as similarities regarding the chosen winner or Condorcet efficiency and the homogeneity of the voters' preferences (see, e.g. [63, 87]). An interesting new approach could be to define homogeneity in a more "fragmented" manner meaning that two correlated preferences do not have to be entirely identical but, e.g., have the same candidates on the first, say, ℓ positions but the remaining positions may differ.

For the conducted experiments, an improved version of an algorithm given by Conitzer et al. [28] is used. The experiments show that independent of the

[12] This procedure generates highly correlated votes and models how homogeneity varies in society. By differently choosing the number of votes that are put back in each step, the correlation can be varied. Here we have that the second vote is the same as the first vote with probability $1/2$. This probability can be de- or increased by putting back less or more votes in each step.

underlying model a successful manipulation action for a single manipulator can easily be computed in elections where the number of candidates is not higher than 128. For coalitions of manipulators casting identical votes, the computational cost of deciding whether manipulation is possible depends on the coalition size. Increasing the number of manipulators increases the manipulability of the tested elections.

Complementary to the testing on random elections where no votes exist that are never cast, Walsh [110] also samples real elections: An election to determine a trajectory for NASA's Marine spacecraft and the votes cast in a faculty hiring committee at the University of Irvine (for details see [31, 36]). The sampled elections show similar results as the randomly generated elections.[13] Deciding whether a manipulator can successfully change the outcome of the election or not is easy for up to 128 candidates and voters.

Davies et al. [30] use the experimental approach introduced by Walsh [109, 110] to analyze their approximation algorithms, Largest Fit and Average Fit, for the Borda system and compare them with the algorithm Reverse of Zuckerman et al. [118]. Random elections for their experiments are generated in the IC model and the PE model, and the optimal solution for a given instance is computed with the solver Gecode after having modeled the manipulation problem as a constraint satisfaction problem.[14] The results show that the Largest Fit algorithm finds an optimal solution in roughly 83 % of the elections with uniform votes and in roughly 42 % of the elections generated with the PE model. The Average Fit algorithm, on the other hand, finds an optimal solution in almost all (roughly 99 %) of the tested elections independent of the distribution. Thus, Largest Fit and Average Fit behave better than Reverse for Borda in both distribution models, as Reverse finds an optimal manipulation in roughly 76 % of the tested elections only.

Narodytska et al. [93] study unweighted and weighted manipulation of Nanson's and Baldwin's rules. In the unweighted case, they prove that both rules are NP-hard to manipulate, even for just one manipulator. In the weighted case, they show that coalitional manipulation is NP-hard for Nanson when there are four candidates and is in P for three candidates. Since Coleman and Teague [21] have shown NP-hardness of this problem for Baldwin already for three candidates, Baldwin's rule appears to be computationally more resistant to manipulation than Nanson's rule, as Narodytska et al. [93] point out, at least when restricted to three candidates (and of course assuming that $P \neq NP$).

Narodytska et al. [93] also conduct experiments for these two rules, using the same approximation algorithms as Davies et al. [30], Largest Fit and Average Fit, and

[13]For comparison, the sampled elections should have the same number of candidates and voters as the randomly generated elections. To obtain this, candidate or voter sets containing too many elements are reduced by randomly choosing appropriate subsets. If the list of voters has to be extended, votes are uniformly and independently chosen from the given votes. If the candidate set is too small, the candidates are duplicated and the ranking between the clone and the original candidate is chosen randomly.

[14]The timeout is set to one hour and the variable-ordering heuristic used is the "*domain over weight degree*."

two more, Eliminate and Reverse Eliminate. Their results suggest that, at least for the algorithms studied, Nanson's and Baldwin's rules are harder to manipulate on random elections than Borda's rule.

For Nanson and Baldwin, Reverse works slightly better than Largest Fit and Average Fit, which in turn outperform Eliminate and Reverse Eliminate, especially when the number of candidates is large.

For a deeper experimental analysis following the presented approach, possible future work could be to conduct experiments with larger election sizes, different voter distribution models, or different (and especially larger) real-life elections.

4 Challenges to complexity results for control

Electoral control models structural changes an election's chair (who seeks to influence the outcome of an election—again, we assume that the chair knows all the votes) can make by adding, deleting, or partitioning either candidates or voters. These control actions model real-world election issues such as campaign advertisement, get-out-the-vote-drives, vote suppression, and gerrymandering. Bartholdi et al. [6] introduced seven constructive control types. Hemaspaandra et al. [70] extended this study by introducing destructive control types as well, and also by adding two natural tie-handling rules for the partitioning cases.

Analogously to the formal definition of manipulation, two winner models can be used to define control formally. By default, we will consider control problems in the unique-winner model where a control action is successful only if the designated candidate alone wins the resulting election. The reason for this choice is that the results to be presented in this section typically are in this model, unless stated otherwise. In the other model, the co-winner model, the designated candidate is allowed to be a winner amongst others and the corresponding control action is still considered to be successful.

Control actions can be formalized by decision problems whose instances always contain a distinguished candidate c and an initial election (C, V), and the question always is whether c can be made the (unique) winner by modifying (C, V) according to the control action at hand. We start by defining the constructive control types. For Constructive Control by Adding Voters (CCAV), we in addition are given a list of as yet unregistered votes from which the chair can choose which to add, and a bound on the number of voters that may be added. Similarly, for Constructive Control by Adding Candidates (CCAC), the instances additionally contain a set of spoiler candidates and a bound on the number of candidates that may be added. Note that Bartholdi et al. [6] actually defined this problem without a given bound (and we refer to their control problem as CCAUC, where the U stands for Unlimited). As we will see later on in Table 3, this distinction can be important. For example, in Copeland^0 and Copeland^1 elections, adding this bound to the instance increases the complexity of the problem from P membership to NP-completeness. In the scenarios describing constructive control by deleting either candidates or voters (denoted by CCDC and CCDV), only the limiting bound is given in addition to the initial election and the distinguished candidate.

Table 3 Overview of NP-hardness and tractability results regarding control

	Condorcet		Plurality		Copeland$^\alpha$		Approval		Bucklin		Fallback	
	C	D	C	D	C	D	C	D	C	D	C	D
CAUC	I	V	R	R	V, $\alpha \in \{0, 1\}$ R, $\alpha \notin \{0, 1\}$	V	I	V	R	R	R	R
CAC	I	V	R	R	R	V	I	V	R	R	R	R
CDC	V	I	R	R	R	V	V	I	R	R	R	R
CPC-TE	V	I	R	R	R	V	V	I	R	R	R	R
CRPC-TE	V		R		R		V		R		R	
CPC-TP	V	I	R	R	R	V	I	I	R	R	R	R
CRPC-TP	V		R		R		I		R		R	
CAV	R	V	V	V	R	R	R	V	R	V	R	V
CDV	R	V	V	V	R	R	R	V	R	V	R	V
CPV-TE	R	V	V	V	R	R	R	V	R	R	R	R
CPV-TP	R	V	V	V	R	R	R	V	R	U	R	R

Key: R = **r**esistance, which means NP-hardness; V = **v**ulnerability, which means P membership; I = **i**mmunity, which means that control is impossible; U = **u**nsolved, which means that control is possible but P membership/NP-hardness is not known

Partitioning either candidates or voters changes an election's course by transforming it into a two-stage election with one (or two) pre-round election(s) and one final-stage election. The two tie-handling rules mentioned above are *Ties Eliminate* (*TE*), where only unique pre-round winners move on to the final round, and *Ties Promote* (*TP*), where all pre-round winners participate in the final stage. Which of these two tie-handling rules is more natural depends on the winner model chosen: TE better matches the unique-winner model, whereas TP better fits the co-winner model. As an example, we formally define one of these problems explicitly:

Constructive Control by Partition of Candidates (CCPC)	
Given:	An election (C, V) and a distinguished candidate $c \in C$.
Question:	Is it possible to partition C into C_1 and C_2 such that c is the unique winner (under the election system at hand) of election $(W_1 \cup C_2, V)$, where W_1 is the set of winners of subelection (C_1, V) surviving the tie-handling rule?

In Constructive Control by Runoff Partition of Candidates (CCRPC), there are two pre-rounds, (C_1, V) and (C_2, V), and the final stage is the subelection $(W_1 \cup W_2, V)$, where W_i, $i \in \{1, 2\}$, is the set of winners of subelection (C_i, V) surviving the tie-handling rule. When the list of voters is partitioned (yielding the problem CCPV), the pre-round of the resulting two-stage election consists of two subelections where the voters of each sublist vote over all candidates and those candidates surviving the tie-handling rule run against each other in the final round, considering all votes.

The destructive variants are defined analogously: DCAC, DCAUC, DCAV, DCDC, DCDV, DCPC, DCRPC, and DCPV. Altogether, a total of 22 standard types of control have been defined and studied in the literature (see, e.g., the survey by Baumeister et al. [7] for the formal definitions and the motivation of each single type). However, building on work of Faliszewski et al. [51], Hemaspaandra et al. [67]

have recently observed that these are actually only 20 distinct types of control, as the destructive variants of control by partition and runoff partition of candidates collapse to just one type for both tie-handling rules: DCPC-TE = DCRPC-TE and DCPC-TP = DCRPC-TP, regardless of the voting system.

Bartholdi et al. [6] initiated the study of control complexity by analyzing constructive control for plurality voting and the Condorcet voting system and showed that plurality elections are fully resistant to constructive candidate control, i.e., the control problems are NP-hard, whereas the constructive voter control problems are all in P. In Condorcet elections, however, the voter control problems are NP-hard and the candidate cases yield in some sense mixed results: Control by adding candidates can never be exerted successfully in Condorcet elections—this property is called *immunity*. For the remaining candidate control problems, deterministic polynomial-time algorithms could be found. (Note that immunity results imply P membership as well because the corresponding decision problem can trivially be decided, since the answer is always "no." However, if a voting system is immune to some type of control, it is in fact irrelevant that the control problem can be solved efficiently, as control simply is not possible.)

Hemaspaandra et al. [70] completed the study for plurality and Condorcet and additionally analyzed control complexity in approval elections. They showed that destructive candidate control in plurality elections is NP-hard while the destructive voter control cases can be solved in deterministic polynomial time. For the Condorcet system, on the other hand, voter control becomes tractable when changing the control action from the constructive to the destructive case. Approval voting behaves similarly to Condorcet as it is fully resistant to constructive voter control and the destructive voter control cases are easy to solve. Also the results for the candidate cases are consistent with those for Condorcet voting except that approval voting is also immune to constructive control by partition and runoff partition of candidates in model TP.

The family of Copeland$^\alpha$ systems has been studied in depth by Faliszewski et al. [51], who showed that Copeland$^\alpha$ is fully resistant to voter control for all (rational) values of $\alpha \in [0, 1]$ and is fully resistant to constructive control whenever $\alpha \in (0, 1)$. However, the destructive candidate control cases can be solved in deterministic polynomial time. They also showed that the complexity of CCAUC strongly depends on the value of α: For $\alpha \in \{0, 1\}$, the problem can be solved efficiently, but for all other values of α in [0, 1], the problem is intractable. Another interesting point to note here is that for Copeland$^\alpha$ with $\alpha \in \{0, 1\}$, CCAUC is in P, yet CCAC is NP-complete.

Erdélyi and Rothe [46] and Erdélyi et al. [45] studied the control complexity in Bucklin and fallback elections and showed that both voting systems are fully resistant to both candidate control and to constructive voter control. Only destructive control by deleting and adding voters is known to be solvable efficiently for both voting systems. The remaining voter control problems are each NP-complete in fallback voting, and also in Bucklin voting with the exception of destructive control by partition of voters in model TP which is still an open problem.

Table 3 shows the just presented control complexity results for those voting systems that will be further discussed in the following sections. A number of follow-up papers were concerned with the control complexity of further voting systems and other aspects of control—see, e.g., [44, 50, 71, 88] and the surveys by Faliszewski et al. [48, 52] and Baumeister et al. [7].

Many of the arguments in the previous section on manipulation apply also to control, so an election's chair does not have to despair either when facing an NP-hard control problem. However, some approaches—such as those of Zuckerman et al. [117, 118] that have been proposed for manipulation—may be less suited for control problems.

4.1 Fixed-parameter tractability

Typical parameters for all standard control types are again the number of candidates and the number of voters. For example, Faliszewski et al. [51] establish FPT results for all NP-complete control problems for Copeland$^{\alpha}$ voting when the number of candidates or voters is bounded by a fixed constant, except certain cases of candidate control when the number of voters is bounded. To this end, they employ Lenstra's algorithm for bounded-variable-cardinality integer programming [79]. Betzler and Uhlmann [14] proved NP-completeness for control by adding and deleting candidates when the number of voters is bounded.

Unfortunately, the number of candidates and the number of voters in an election may be large, and in such cases an FPT result is not of much value. Betzler and Uhlmann [14] also consider the parameters "number of candidates added" (in constructive control by adding candidates) and "number of candidates deleted" (in constructive control by deleting candidates). They show that these parameterized problems are equivalent under parameterized reductions to certain digraph problems, parameterized by the minimum indegree or the maximum outdegree, and establish W[2]-completeness results for constructive control by adding and by deleting candidates with respect to the output parameters (e.g., "number of candidates added" for constructive control by adding candidates) in Copeland$^{\alpha}$ voting.

For plurality voting and with respect to the output parameters, W[2]-hardness of constructive control by adding candidates is due to (the proofs of[15]) Bartholdi et al. [6] and Liu et al. [83], and W[2]-hardness of destructive control by adding candidates is due to (the proof of) Hemaspaandra et al. [70]. Betzler and Uhlmann [14] prove W[2]-hardness for constructive and W[1]-hardness for destructive control by deleting candidates. For all these parameterized problems, a proof of an upper bound (membership in W[2] or W[1]) is still missing. Further fixed-parameter tractability and parameterized intractability results concerning control in different voting systems—including approval, maximin, Bucklin, and fallback voting—are due to Liu et al. [83], Liu and Zhu [84] and Erdélyi and Fellows [39] (see also [40]).

4.2 Single-peaked preferences

Regarding the restriction of electorates to single-peaked preferences, Faliszewski et al. [53] showed that the control problem can be solved in polynomial time in all cases in which plurality and approval voting are in general NP-hard to control by adding or deleting either candidates or voters. Brandt et al. [18] achieved similar results for other voting systems as well, in particular those that satisfy the weak

[15] By this we mean that these W[2]-hardness results may not be explicitly stated in these papers but follow immediately from the NP-hardness reductions given there.

Condorcet criterion, and in addition for the case of constructive control by partition of voters.

4.3 Experimental analysis

Among natural voting systems with efficient winner determination, fallback voting is currently known to have the most NP-hardness results (i.e., *resistances*) with respect to the standard control scenarios,[16] and Bucklin voting behaves almost as well [45].

For plurality, Bucklin, and fallback voting, Rothe and Schend [100] have performed an extensive experimental study on the frequency of controllability[17] for randomly generated elections. Inspired by the experimental setup of Walsh [109, 110] described in the section about experimental approaches regarding manipulation, one model used for the vote distribution is the IC model. To simulate correlated votes, however, Rothe and Schend [100] introduce an adaption of the PE model, which they call the *Two Mainstreams* (*TM*) model:

1. Two votes v_1, v_2 are drawn independently from an urn containing all possible, say t, votes.
2. Each of the votes drawn is put back into the urn with t additional identical votes, and the list of votes is then drawn uniformly at random from this urn. Thus, each voter has with probability $1/3$ the same preference as v_1, with probability $1/3$ the same preference as v_2, and again with probability $1/3$ a different preference.

The votes v_1 and v_2 model two mainstreams, such as liberal and conservative, a society may have. Note that, just as the PE model, also this model could be modified so as to allow more deviation by requiring that not exact copies of v_1 and v_2 but only similar preferences (say, with the first ℓ positions being identical but deviating on the remaining ones) are put back into the urn.

Rothe and Schend [100, 101] did not use the variant of the PE model employed by Walsh [109, 110] because of the high probability that all or many preferences in the generated election are identical (see footnote 12). In a manipulation context, this does not cause too much of a problem, since the manipulators can change their votes and can be in some sense independent of the distribution. Control actions, however, do not allow direct modifications of the preferences, and in electorates where, say, all voters have the same preference, for most control types it would then be trivially easy to decide whether or not control is possible.

With these two distributions, random elections are generated, letting the number of candidates and votes, m and n, vary in powers of 2 between 4 and 128. For each data point (i.e., for each pair of m and n), 500 elections are tested.

The algorithms implemented to solve the control problems for the different voting systems apply a heuristic approach: A successful control action for a given election is searched for by basically testing all possible control actions of the given type.

[16] Shortly after the results for Bucklin and fallback voting were published by Erdélyi et al. [40], Menton [89] showed that the voting system *normalized range voting* draws level with fallback voting in terms of the number of resistances to control.

[17] The experiments have been conducted for each of the three voting systems in the NP-hard control cases only, with one exception: DCPV in model TP has nonetheless been studied experimentally for Bucklin voting, as the complexity of this control problem is still unknown.

This testing process is systematized by preordering the candidates or voters such that promising control actions are tested first. To ensure practicability, the algorithm aborts the computation after a fixed timeout, indicating the inconclusive result by its output.

The results of Rothe and Schend [100] allow a fine-grained analysis of those control types the three considered voting systems are resistant to (for a complete collection of results for these experiments, see the corresponding technical report [101]).

So far, the results for both constructive and destructive control by partition of candidates in model TE for Bucklin and fallback voting align with the conclusions arrived at by Rothe and Schend [100] for other control scenarios, which can be summarized as follows.

Comparing the two distribution models used, for all investigated control types and in all voting systems considered, elections generated with the IC model show a higher overall number of yes-instances than elections generated in the TM model. At the same time, the number of timeouts is higher for elections generated with the TM model.

Bucklin and fallback voting show the same tendencies both theoretically (regarding NP-hardness versus membership in P) and experimentally. Note that the complexity status of DCPV in model TP is still open, whereas this problem is known to be NP-complete for fallback voting [45]. Since Bucklin and fallback voting behave empirically very similarly with respect to this control type, we conjecture that this control problem is NP-complete for Bucklin voting as well.

Comparing Figs. 1 and 2 affirms the intuition that destructive control is easier to exert than constructive control: For some election sizes, up to 100 % of the tested elections are controllable by certrain destructive control types, whereas for the constructive cases, especially regarding candidate control, the number of yes-instances is much smaller.

Further comparisons accross the different control types show that in the constructive cases voter control seems to be easier to exert in practice than candidate control. Control by adding candidates and by partition of candidates show particularly few yes-instances for all three voting systems, indicating that these may be the hardest control types investigated. The results also show that more of the tested elections are controllable by deleting voters than by adding voters, and the same can be observed regarding candidate control. But, of course, one should keep in mind that all conclusions drawn from these experiments strongly depend on the specific algorithm

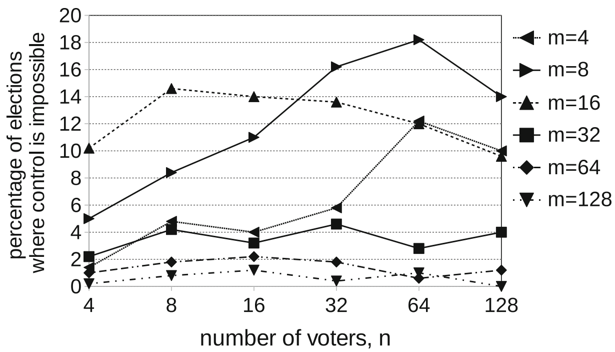
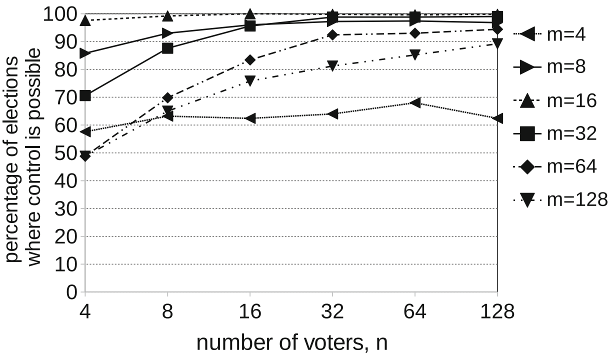


Fig. 1 Experimental results for fallback voting in the TM model for CCPC in model TE, for a fixed number of candidates

Fig. 2 Experimental results for Bucklin voting in the IC model for DCPC in model TE, for a fixed number of candidates



used. Further improved algorithms might provide new insights, especially regarding those cases where only few controllable instances were observed. Furthermore, it is clear that this first approach of an experimental analysis is not exhaustive and should be seen simply as launching this line of research. There certainly are many possibilities for improvement using, for example, more sophisticated algorithms, other voter distribution models, or tests on real data.

Acknowledgements We thank Toby Walsh for interesting discussions on experimental studies of voting systems, and Vince Conitzer for his very helpful comments. Furthermore we thank Volker Aurich for giving us access to his computer lab, Guido Königstein for his help in setting up our experiments, and the anonymous reviewers for their helpful comments and suggestions that very much helped us to improve this paper.

References

1. Baldwin, J.: The technique of the Nanson preferential majority system of election. Proc. R. Soc. Vic. **39**, 42–52 (1926)
2. Ballester, M., Haeringer, G.: A characterization of the single-peaked domain. Soc. Choice Welf. **36**(2), 305–322 (2011)
3. Bartholdi III, J., Orlin, J.: Single transferable vote resists strategic voting. Soc. Choice Welf. **8**(4), 341–354 (1991)
4. Bartholdi III, J., Tovey, C., Trick, M.: The computational difficulty of manipulating an election. Soc. Choice Welf. **6**(3), 227–241 (1989)
5. Bartholdi III, J., Tovey, C., Trick, M.: Voting schemes for which it can be difficult to tell who won the election. Soc. Choice Welf. **6**(2), 157–165 (1989)
6. Bartholdi III, J., Tovey, C., Trick, M.: How hard is it to control an election? Math. Comput. Model. **16**(8–9), 27–40 (1992)
7. Baumeister, D., Erdélyi, G., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Computational aspects of approval voting. In: Laslier, J., Sanver, R. (eds.) Handbook on Approval Voting, chap. 10, pp. 199–251. Springer (2010)
8. Berg, S.: Paradox of voting under an urn model: the effect of homogeneity. Public Choice **47**(2), 377–387 (1985)
9. Betzler, N.: On problem kernels for possible winner determination under the *k*-approval protocol. In: Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science, pp. 114–125. Lecture Notes in Computer Science #6281. Springer-Verlag (2010)
10. Betzler, N., Bredereck, R., Chen, J., Niedermeier, R.: Studies in computational aspects of voting—a parameterized complexity perspective. In: Bodlaender et al. H. (ed.) Fellows Festschrift, pp. 318–363. Lecture Notes in Computer Science #7370. Springer-Verlag (2012)

11. Betzler, N., Guo, J., Niedermeier, R.: Parameterized computational complexity of Dodgson and Young elections. In: Proceedings of the 11th Scandinavian Workshop on Algorithm Theory, pp. 402–413. Lecture Notes in Computer Science #5124. Springer-Verlag (2008)
12. Betzler, N., Hemmann, S., Niedermeier, R.: A multivariate complexity analysis of determining possible winners given incomplete votes. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 53–58. IJCAI (2009)
13. Betzler, N., Niedermeier, R., Woeginger, G.: Unweighted coalitional manipulation under the Borda rule is NP-hard. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 55–60. IJCAI (2011)
14. Betzler, N., Uhlmann, J.: Parameterized complexity of candidate control in elections and related digraph problems. In: Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications, pp. 43–53. Lecture Notes in Computer Science #5165. Springer-Verlag (2008)
15. Black, D.: On the rationale of group decision-making. J. Polit. Econ. **56**(1), 23–34 (1948)
16. Brams, S., Fishburn, P.: Approval voting. Am. Polit. Sci. Rev. **72**(3), 831–847 (1978)
17. Brams, S., Sanver, R.: Voting systems that combine approval and preference. In: Brams, S., Gehrlein, W., Roberts, F. (eds.) The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn, pp. 215–237. Springer (2009)
18. Brandt, F., Brill, M., Hemaspaandra, E., Hemaspaandra, L.: Bypassing combinatorial protections: polynomial-time algorithms for single-peaked electorates. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 715–722. AAAI Press (2010)
19. Brelsford, E., Faliszewski, P., Hemaspaandra, E., Schnoor, H., Schnoor, I.: Approximability of manipulating elections. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, pp. 44–49 (2008)
20. Brill, M., Fischer, F.: The price of neutrality for the ranked pairs method. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, pp. 1299–1305. AAAI Press (2012)
21. Coleman, T., Teague, V.: On the complexity of manipulating elections. In: Proceedings of Computing: the 13th Australasian Theory Symposium, vol. 65, pp. 25–33 (2007)
22. Condorcet, J.: Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix (1785). Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale. English translation appears in I. McLean and A. Urken, Classics of Social Choice, pp. 91–112. University of Michigan Press (1995)
23. Conitzer, V.: Eliciting single-peaked preferences using comparison queries. J. Artif. Intell. Res. **35**, 161–191 (2009)
24. Conitzer, V.: Making decisions based on the preferences of multiple agents. Commun. ACM **53**(3), 84–94 (2010)
25. Conitzer, V., Rognlie, M., Xia, L.: Preference functions that score rankings and maximum likelihood estimation. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 109–115. IJCAI (2009)
26. Conitzer, V., Sandholm, T.: Universal voting protocol tweaks to make manipulation hard. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, pp. 781–788. Morgan Kaufmann (2003)
27. Conitzer, V., Sandholm, T.: Nonexistence of voting rules that are usually hard to manipulate. In: Proceedings of the 21st National Conference on Artificial Intelligence, pp. 627–634. AAAI Press (2006)
28. Conitzer, V., Sandholm, T., Lang, J.: When are elections with few candidates hard to manipulate? J. ACM **54**(3), 14 (2007)
29. Copeland, A.: A "reasonable" social welfare function. Mimeographed Notes from a Seminar on Applications of Mathematics to the Social Sciences. University of Michigan (1951)
30. Davies, J., Katsirelos, G., Narodytska, N., Walsh, T.: Complexity of and algorithms for Borda manipulation. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence, pp. 657–662. AAAI Press (2011)
31. Dobra, J.: An approach to empirical studies of voting paradoxes: an update and extension. Public Choice **41**(2), 241–250 (1983)
32. Dobzinski, S., Procaccia, A.: Frequent manipulability of elections: the case of two voters. In: Proceedings of the 4th International Workshop on Internet and Network Economics, pp. 653–664 (2008)
33. Dorn, B., Schlotter, I.: Multivariate complexity analysis of swap bribery. In: Proceedings of the 5th International Symposium on Parameterized and Exact Computation, pp. 107–122. Lecture Notes in Computer Science #6478. Springer-Verlag (2010)

34. Downey, R., Fellows, M.: Parameterized Complexity. Springer-Verlag (1999)
35. Duggan, J., Schwartz, T.: Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. Soc. Choice Welf. **17**(1), 85–93 (2000)
36. Dyer, J., Miles Jr., R.: An actual application of collective choice theory to the selection of trajectories for the mariner Jupiter/Saturn 1977 project. Oper. Res. **24**(2), 220–244 (1976)
37. Elkind, E., Faliszewski, P., Slinko, A.: Swap bribery. In: Proceedings of the 2nd International Symposium on Algorithmic Game Theory, pp. 299–310. Lecture Notes in Computer Science #5814. Springer-Verlag (2009)
38. Elkind, E., Lipmaa, H.: Hybrid voting protocols and hardness of manipulation. In: Proceedings of the 16th International Symposium on Algorithms and Computation, pp. 206–215. Lecture Notes in Computer Science #3827. Springer-Verlag (2005)
39. Erdélyi, G., Fellows, M.: Parameterized control complexity in Bucklin voting and in fallback voting. In: Conitzer, V., Rothe, J. (eds.) Proceedings of the 3rd International Workshop on Computational Social Choice, pp. 163–174. Universität Düsseldorf (2010)
40. Erdélyi, G., Fellows, M., Rothe, J., Schend, L.: Control complexity in Bucklin and fallback voting. Computing Research Repository (2012). Tech. Rep. arXiv:1103.2230 [cs.CC]. March 11, 2011. Revised August 21, 2012
41. Erdélyi, G., Hemaspaandra, L., Rothe, J., Spakowski, H.: Frequency of correctness versus average-case polynomial time and generalized juntas. Tech. Rep. TR-934, Department of Computer Science, University of Rochester, Rochester, NY (2008)
42. Erdélyi, G., Hemaspaandra, L., Rothe, J., Spakowski, H.: Frequency of correctness versus average polynomial time. Inf. Process. Lett. **109**(16), 946–949 (2009)
43. Erdélyi, G., Hemaspaandra, L., Rothe, J., Spakowski, H.: Generalized juntas and NP-hard sets. Theor. Comput. Sci. **410**(38–40), 3995–4000 (2009)
44. Erdélyi, G., Nowak, M., Rothe, J.: Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Math. Log. Q. **55**(4), 425–443 (2009)
45. Erdélyi, G., Piras, L., Rothe, J.: The complexity of voter partition in Bucklin and fallback voting: solving three open problems. In: Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 837–844. IFAAMAS (2011)
46. Erdélyi, G., Rothe, J.: Control complexity in fallback voting. In: Proceedings of Computing: the 16th Australasian Theory Symposium, vol. 32, no. 8, pp. 39–48. Conferences in Research and Practice in Information Technology Series. Australian Computer Society (2010)
47. Escoffier, B., Lang, J., Öztürk, M.: Single-peaked consistency and its complexity. In: Proceedings of the 18th European Conference on Artificial Intelligence, pp. 366–370. IOS Press (2008)
48. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: Using complexity to protect elections. Commun. ACM **53**(11), 74–82 (2010)
49. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: The complexity of manipulative attacks in nearly single-peaked electorates. In: Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge, pp. 228–237. ACM Press (2011)
50. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.: Multimode control attacks on elections. J. Artif. Intell. Res. **40**, 305–351 (2011)
51. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Llull and Copeland voting computationally resist bribery and constructive control. J. Artif. Intell. Res. **35**, 275–341 (2009)
52. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: A richer understanding of the complexity of election systems. In: Ravi, S., Shukla, S. (eds.) Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz, chap. 14, pp. 375–406. Springer (2009)
53. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. Inf. Comput. **209**(2), 89–107 (2011)
54. Faliszewski, P., Hemaspaandra, E., Schnoor, H.: Copeland voting: ties matter. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 983–990. IFAAMAS (2008)
55. Faliszewski, P., Hemaspaandra, E., Schnoor, H.: Manipulation of Copeland elections. In: Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 367–374. IFAAMAS (2010)
56. Faliszewski, P., Procaccia, A.: AI's war on manipulation: are we winning? AI Mag. **31**(4), 53–64 (2010)

57. Flum, J., Grohe, M.: Parameterized Complexity Theory. EATCS Texts in Theoretical Computer Science. Springer-Verlag (2006)
58. Friedgut, E., Kalai, G., Nisan, N.: Elections can be manipulated often. In: Proceedings of the 49th IEEE Symposium on Foundations of Computer Science, pp. 243–249. IEEE Computer Society (2008)
59. Friedgut, E., Keller, N., Kalai, G., Nisan, N.: A quantitative version of the Gibbard–Satterthwaite theorem for three alternatives. SIAM J. Comput. **40**(3), 934–952 (2011)
60. Gailmard, S., Patty, J., Penn, E.: Arrow's theorem on single-peaked domains. In: Aragonés, E., Beviá, C., Llavador, H., Schofield, N. (eds.) The Political Economy of Democracy, pp. 335–342. Fundación BBVA (2009)
61. Garey, M., Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)
62. Gasarch, W.: The P = ? NP poll. SIGACT News **33**(2), 34–47 (2002)
63. Gehrlein, W., Berg, S.: The effect of social homogeneity on coincidence probabilities for pairwise proportional lottery and simple majority rules. Soc. Choice Welf. **9**(4), 361–372 (1992)
64. Gibbard, A.: Manipulation of voting schemes. Econometrica **41**(4), 587–601 (1973)
65. Goldreich, O.: Notes on Levin's theory of average-case complexity. Tech. Rep. TR97-058. Electronic Colloquium on Computational Complexity (1997)
66. Hemaspaandra, E., Hemaspaandra, L.: Dichotomy for voting systems. J. Comput. Syst. Sci. **73**(1), 73–83 (2007)
67. Hemaspaandra, E., Hemaspaandra, L., Menton, C.: Search versus decision for election manipulation problems. In: Proceedings of the 30th Annual Symposium on Theoretical Aspects of Computer Science, pp. 377–388. Leibniz-Zentrum für Informatik (2013)
68. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. J. ACM **44**(6), 806–825 (1997)
69. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Raising NP lower bounds to parallel NP lower bounds. SIGACT News **28**(2), 2–13 (1997)
70. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Anyone but him: the complexity of precluding an alternative. Artif. Intell. **171**(5–6), 255–285 (2007)
71. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Hybrid elections broaden complexity-theoretic resistance to control. Math. Log. Q. **55**(4), 397–424 (2009)
72. Hemaspaandra, L., Williams, R.: An atypical survey of typical-case heuristic algorithms. SIGACT News **43**(4), 71–89 (2012)
73. Homan, C., Hemaspaandra, L.: Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. J. Heuristics **15**(4), 403–423 (2009)
74. Isaksson, M., Kindler, G., Mossel, E.: The geometry of manipulation: a quantitative proof of the Gibbard–Satterthwaite theorem. In: Proceedings of the 51st IEEE Symposium on Foundations of Computer Science, pp. 319–328. IEEE Computer Society (2010)
75. Isaksson, M., Kindler, G., Mossel, E.: The geometry of manipulation—a quantitative proof of the Gibbard–Satterthwaite theorem. Combinatorica **32**(2), 221–250 (2012)
76. Konczak, K., Lang, J.: Voting procedures with incomplete preferences. In: Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling, pp. 124–129 (2005)
77. Korf, R.: From approximate to optimal solutions: a case study of number partitioning. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 266–272 (1995)
78. Ladner, R., Lynch, N., Selman, A.: A comparison of polynomial time reducibilities. Theor. Comput. Sci. **1**(2), 103–124 (1975)
79. Lenstra Jr., H.: Integer programming with a fixed number of variables. Math. Oper. Res. **8**, 538–548 (1983)
80. Lepelley, D.: Constant scoring rules, Condorcet criteria, and single-peaked preferences. Economic Theory **7**(3), 491–500 (1996)
81. Levin, L.: Average case complete problems. SIAM J. Comput. **15**(1), 285–286 (1986)
82. Lindner, C., Rothe, J.: Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In: Holder, A. (ed.) Mathematical Programming Glossary. INFORMS Computing Society (2008)
83. Liu, H., Feng, H., Zhu, D., Luan, J.: Parameterized computational complexity of control problems in voting systems. Theor. Comput. Sci. **410**(27–29), 2746–2753 (2009)
84. Liu, H., Zhu, D.: Parameterized complexity of control problems in maximin election. Inf. Process. Lett. **110**(10), 383–388 (2010)

85. Mattei, N.: Empirical evaluation of voting rules with strictly ordered preference data. In: Proceedings of the 2nd International Conference on Algorithmic Decision Theory, pp. 165–177. Lecture Notes in Artificial Intelligence #6992. Springer-Verlag (2011)
86. McCabe-Dansted, J., Pritchard, G., Slinko, A.: Approximability of Dodgson's rule. Soc. Choice Welf. **31**(2), 311–330 (2008)
87. McCabe-Dansted, J., Slinko, A.: Exploratory analysis of similarities between common social choice rules. Group Decis. Negot. **15**, 77–107 (2006)
88. Meir, R., Procaccia, A., Rosenschein, J., Zohar, A.: Complexity of strategic behavior in multi-winner elections. J. Artif. Intell. Res. **33**, 149–178 (2008)
89. Menton, C.: Normalized range voting broadly resists control. Computing Research Repository (2012). Tech. Rep. arXiv:1005.5698 [cs.GT]. April 25, 2011. Revised June 26, 2012
90. Mossel, E., Rácz, M.: Election manipulation: the average case. ACM SIGecom Exchanges **11**(2), 22–24 (2012)
91. Mossel, E., Rácz, M.: A quantitative Gibbard–Satterthwaite theorem without neutrality. In: Proceedings of the 44th Symposium on Theory of Computing, pp. 1041–1060. ACM (2012)
92. Nanson, E.: Methods of election. Proc. R. Soc. Vic. **19**, 197–240 (1882)
93. Narodytska, N., Walsh, T., Xia, L.: Manipulation of Nanson's and Baldwin's rules. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence, pp. 713–718. AAAI Press (2011)
94. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)
95. Ogiwara, M., Watanabe, O.: On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. SIAM J. Comput. **20**(3), 471–483 (1991)
96. Procaccia, A.: Can approximation circumvent Gibbard-Satterthwaite? In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 836–841 (2010)
97. Procaccia, A., Rosenschein, J.: Average-case tractability of manipulation in voting via the fraction of manipulators. In: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, vol. 718–720 (2007). Short paper
98. Procaccia, A., Rosenschein, J.: Junta distributions and the average-case complexity of manipulating elections. J. Artif. Intell. Res. **28**, 157–181 (2007)
99. Reisch, Y.: Das Manipulationsproblem für Bucklin–Wahlen. Bachelor thesis, Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, Düsseldorf, Germany (2012)
100. Rothe, J., Schend, L.: Control complexity in Bucklin, fallback, and plurality voting: an experimental approach. In: Proceedings of the 11th International Symposium on Experimental Algorithms, pp. 356–368. Lecture Notes in Computer Science #7276. Springer-Verlag (2012)
101. Rothe, J., Schend, L.: Control complexity in Bucklin, fallback, and plurality voting: an experimental approach. Computing Research Repository (2012). Tech. Rep. arXiv:1203.3967v1 [cs.GT]. March 18, 2012. Revised August 20, 2012
102. Satterthwaite, M.: Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions. J. Econ. Theory **10**(2), 187–217 (1975)
103. Schöning, U.: Complete sets and closeness to complexity classes. Math. Syst. Theory **19**(1), 29–42 (1986)
104. Slinko, A.: On asymptotic strategy-proofness of classical social choice rules. Theory Decis. **52**(4), 389–398 (2002)
105. Slinko, A.: On asymptotic strategy-proofness of the plurality and the run-off rules. Soc. Choice Welf. **19**(2), 313–324 (2002)
106. Slinko, A.: How large should a coalition be to manipulate an election? Math. Soc. Sci. **47**(3), 289–293 (2004)
107. Tideman, N.: Independence of clones as a criterion for voting rules. Soc. Choice Welf. **4**(3), 185–206 (1987)
108. Walsh, T.: Uncertainty in preference elicitation and aggregation. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence, pp. 3–8. AAAI Press (2007)
109. Walsh, T.: Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 324–329. IJCAI (2009)
110. Walsh, T.: An empirical study of the manipulability of single transferable voting. In: Proceedings of the 19th European Conference on Artificial Intelligence, pp. 257–262. IOS Press (2010)
111. Wang, J.: Average-case computational complexity theory. In: Hemaspaandra, L., Selman, A. (eds.) Complexity Theory Retrospective II, pp. 295–328. Springer-Verlag (1997)
112. Wang, J.: Average-case intractable NP problems. In: Du, D., Ko, K. (eds.) Advances in Languages, Algorithms, and Complexity, pp. 313–378. Kluwer Academic Publishers (1997)

113. Xia, L., Conitzer, V.: Generalized scoring rules and the frequency of coalitional manipulability. In: Proceedings of the 9th ACM Conference on Electronic Commerce, pp. 109–118. ACM Press (2008)
114. Xia, L., Conitzer, V.: A sufficient condition for voting rules to be frequently manipulable. In: Proceedings of the 9th ACM Conference on Electronic Commerce, pp. 99–108. ACM Press (2008)
115. Xia, L., Conitzer, V., Procaccia, A.: A scheduling approach to coalitional manipulation. In: Proceedings of the 11th ACM Conference on Electronic Commerce, pp. 275–284. ACM Press (2010)
116. Xia, L., Zuckerman, M., Procaccia, A., Conitzer, V., Rosenschein, J.: Complexity of unweighted coalitional manipulation under some common voting rules. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 348–353. IJCAI (2009)
117. Zuckerman, M., Lev, O., Rosenschein, J.: An algorithm for the coalitional manipulation problem under maximin. In: Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 845–852. IFAAMAS (2011)
118. Zuckerman, M., Procaccia, A., Rosenschein, J.: Algorithms for the coalitional manipulation problem. Artif. Intell. **173**(2), 392–412 (2009)