Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice*

— Mathematical Programming Glossary Supplement —

Claudia Lindner and Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany

October 17, 2008

Abstract

This supplement provides a brief introduction to the field of fixed-parameter tractability and parameterized complexity. Some basic notions are explained and some related results are presented, with a focus on problems arising in the field of computational social choice.

1 Fixed-Parameter Tractability and Parameterized Complexity

The study of fixed-parameter tractability and parameterized complexity has emerged as a new field within computational complexity theory in the late 1980s and the early 1990s. Since the early pioneering work of Downey, Fellows, and other researchers this area has established plenty of results, notions, and methods, and it provides a useful framework for dealing in practice with problems considered intractable by classical complexity theory. This supplement gives only a rough sketch of some basic notions and techniques within parameterized complexity theory; for a deeper and more comprehensive treatise, the textbooks by Downey and Fellows [DF99], Flum and Grohe [FG06], and Niedermeier [Nie06] and the survey by Buss and Islam [BI08] are highly recommendable.

Let P and NP, respectively, denote the classical (worst-case) complexity classes *deterministic polynomial time* and *nondeterministic polynomial time*. Given any two decision problems, A and B, we say *A polynomial-time many-one reduces to B* (denoted $A \leq_{\mathrm{m}}^{\mathrm{p}} B$) if there is a polynomial-time computable function f such that, for each input x, $x \in A$ if and only if $f(x) \in B$. A set B is said to be NP-*hard* if for each NP set A, $A \leq_{\mathrm{m}}^{\mathrm{p}} B$. If $B \in \mathrm{NP}$ is NP-hard then B is said to be NP-*complete*.

*Supported in part by the DFG under grants RO 1202/12-1 (within the European Science Foundation's EUROCORES program LogICCC: "Computational Foundations of Social Choice") and RO 1202/11-1 and by the Alexander von Humboldt Foundation's TransCoop program.

Traditionally, NP-hardness is used to formally capture the notion of intractability of problems in classical complexity theory (see, e.g., [Pap94, Rot05] for an introduction to computational complexity and, more specifically, Garey and Johnson [GJ79] for the theory of NP-completeness).

Let us consider three examples of standard NP-complete problems: INDEPENDENT SET, DOMINATING SET, and SATISFIABILITY. The INDEPENDENT SET problem asks, given an undirected, simple graph $G = (V,E)$ and a positive integer k, does there exist an independent set of size at least k in G, i.e., does there exist a subset $I \subseteq V$ such that $|I| \geq k$ and no two vertices in I are adjacent? The problem DOMINATING SET asks, given an undirected, simple graph $G = (V,E)$ and a positive integer k, does there exist a dominating set of size at most k in G, i.e., does there exist a subset $D \subseteq V$ such that $|D| \leq k$ and every vertex in V either belongs to D or is adjacent to some vertex in D? Independent sets and dominating sets can be illustrated as the result of a database query. Consider a database that lists the top 100 math books and the corresponding authors, and consider a graph G whose vertices are these authors, where any two authors are connected by an edge if and only if they have coauthored some of these books. Every subset of the authors such that no two of them have jointly written a book so far is an independent set of G. On the other hand, every subset D of the authors such that each author not in D has written at least one joint book with one of the authors in D is a dominating set. Graph-theoretic notions such as independent sets and dominating sets arise in many application areas. For example, dominating set is a central notion in computer networks or communication networks (see, e.g., Riege and Rothe [RR06] for more details). Finally, the SATISFIABILITY problem (SAT, for short) from propositional logic asks, given a boolean formula φ, is φ satisfiable, i.e., does there exist an assignment of truth values to φ's variables that makes φ true? As a simple example we consider the formulas $\varphi_1 = (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2})$ and $\varphi_2 = x_1 \wedge \overline{x_1}$. Apparently, φ_1 is satisfiable by setting, say, x_1 to true and x_2 to false, whereas φ_2 is not satisfiable at all, i.e., there is no assignment of truth values that makes φ_2 true.

INDEPENDENT SET, DOMINATING SET, and SAT are well known to be NP-complete [GJ79], and thus they cannot be solved in polynomial time (unless P = NP). Although NP-completeness is often taken as evidence that the problem considered is intractable, in practical applications we are usually interested in solving it for certain small parameters. Parameterized complexity theory measures a problem's complexity based on different parts of the input, whereas traditional complexity theory defines complexity measures only in terms of input size. In this sense, parameterized complexity might be seen as some kind of "higher-dimensional" complexity theory. In particular, fixed-parameter tractability takes advantage of the fact that problem instances constitute more than simply strings of certain sizes, but rather have a structure and contain parameters that represent diverse aspects of the problem at hand. For fixed-parameter tractability to be a reasonable concept, it is crucial that the chosen parameter is expected to be small. In this context, "parameter" stands for an intrinsic value rather than for a value used for getting control over an algorithm's behavior as is common in other disciplines, such as programming. For example, let us look at the complexity of database queries. Evaluating even basic conjunctive queries is NP-complete when measuring the complexity based on the size of the database plus the size of the query. However, since query sizes tend to be significantly smaller than the size of the entire database, the query size is a natural parametrization for the database query evaluation problem. Note that this parameter cannot be controlled explicitly, as it is determined by the query itself.

Formally, a parameterized problem (over some alphabet Σ) is a pair (Π, κ), where $\kappa : \Sigma^* \to \mathbb{N}$

is a polynomial-time computable parametrization and $\Pi \subseteq \Sigma^*$ is a set of strings over Σ. However, if the parameter $k = \kappa(x)$ is clear from context, we will denote the parameterized problem (Π, κ) simply by Π. For example, a natural parametrization for SAT is defined by

$$\kappa(\varphi) = \begin{cases} \text{number of variables of } \varphi & \text{if } \varphi \text{ encodes a boolean formula with at least one variable} \\ 1 & \text{otherwise,} \end{cases}$$

and though the resulting parameterized version of SAT is (SAT, κ), we will simply denote this problem by SAT when the parameter $k = \kappa(\varphi)$ is clear.

In parameterized complexity theory, a problem parameterized by some value k is said to be *fixed-parameter tractable* (or to belong to the class FPT) if for each fixed value of k there is an algorithm that solves this problem for instances of size n in time $f(k) \cdot n^{O(1)}$, where f is some computable function. In other words, the seemingly unavoidable combinatorial explosion in NP-hard problems is then confined to the parameter. For example, the parameterized version of SAT defined above is fixed-parameter tractable via the obvious brute-force algorithm that decides the satisfiability of a given boolean formula with k variables, encoded as a string of size m, in time $O(2^k \cdot m)$. Accordingly, by means of parameterizing the SAT problem, it can be solved efficiently, assuming that the chosen parameter k (the number of variables of the given boolean formula) can be expected to be small.

Let (Π, κ) and (Π', κ') be two given parameterized problems. A *parameterized reduction* from (Π, κ) to (Π', κ') is a mapping that transforms any problem instance (x, k) in time $f(k) \cdot |x|^{O(1)}$ (where f is a computable function) into a problem instance (x', k') such that (a) (x, k) is a yes-instance of (Π, κ) if and only if (x', k') is a yes-instance of (Π', κ'), and (b) $k' = \kappa'(x')$ depends only on $k = \kappa(x)$ but not on $|x|$. For notational convencience, whenever we speak of a reduction between two parameterized problems we mean a parameterized reduction, and whenever we we speak of a reduction between two classical decision problems we mean a $\leq_{\mathrm{m}}^{\mathrm{p}}$-reduction.

The theory of fixed-parameter intractability deals with those problems for which no FPT algorithms have been discovered yet, and thus might be hard to solve even if the chosen parameter is a fixed small constant. The W-hierarchy has been introduced to classify these problems according to their fixed-parameter hardness. There are two main approaches that can be taken to define the W-hierarchy and its classes. The original definition, proposed by Downey and Fellows [DF99], characterizes fixed-parameter intractable problems in terms of weighted satisfiability problems on classes of circuits. More recently, Flum et al. [CF03, CFG03] defined the classes W[t] by using nondeterministic algorithms running on restricted random-access machines. However, both definitions are technically very complex and beyond the scope of this supplement. For a more detailed definition of the entire W-hierarchy and of W[2] in particular, we refer to [DF99, FG06, Nie06] and the survey by Buss and Islam [BI08]. Note that if a problem Π belongs to P then the parameterized problem (Π, κ) is in FPT for every parametrization κ. Note also that

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \mathrm{W}[3] \subseteq \cdots.$$

Natural parametrizations for INDEPENDENT SET and DOMINATING SET are the positive integers k that are given as the second component of the instance in the problem definitions as bounds on the size of, respectively, an independent set and a dominating set. It is widely believed that neither of the corresponding parameterized versions of INDEPENDENT SET and DOMINATING SET is fixed-parameter tractable (see, e.g., [DF99, FG06]). Define the (parameterized) problem INDEPENDENT DOMINATING SET (with parameter k) as follows: Given an undirected, simple graph $G = (V,E)$ and a positive integer k, does there exist an independent dominating set of size k? Downey and Fellows [DF99] proved that both DOMINATING SET and INDEPENDENT DOMINATING SET are complete for the parameterized complexity class W[2],[1] which is widely viewed as strong evidence that these problems are not fixed-parameter tractable. The INDEPENDENT SET problem with the above-mentioned parametrization is in W[1].

Regarding the evaluation of database queries, denoting the input size (i.e., the size of the database) by n and the parameter (i.e., the size of the query) by k, the conjunctive query evaluation problem can be solved in time $O(n^k)$, which is still prohibitive even for small values of k (unless only extremely short queries, such as queries of size two or three, were allowed).[2] The conjunctive query evaluation problem is suspected to be not fixed-parameter tractable, as it is proven to be complete for W[1] (see Flum and Grohe [FG06]). Nonetheless, the fixed-parameter intractability of the query evaluation problem is a strong result, since it shows that the evaluation of database queries is hard even for quite small queries.

2 FPT Results for Kemeny Elections

2.1 Kemeny Elections

Voting systems have been intensely studied within computational social choice, an interesting field currently emerging at the interface of computer science and social choice theory. While voting was previously investigated in such areas as political science, economics, operations research, and social choice theory, a wide range of applications in computer science has been found recently. To give just a few examples, voting can play a central role in planning [ER93], similarity search [FKS03], in the design of recommender systems [GMHS99] or ranking algorithms [DKNS01], and in other fields that require methods for preference aggregation and collective decision-making to be applied in, e.g., multiagent systems within distributed artificial intelligence.

For example, Dwork et al. [DKNS01] designed a ranking algorithm with the purpose of lessening the spam in meta-search web-page rankings. In this scenario, web pages are viewed as candidates (or alternatives) and search engines are viewed as voters. Each voter provides a linear ranking of the candidates (with respect to some search query). Voting systems can then be used to aggregate the voters' individual rankings to obtain a "consensus ranking" of a meta-search engine that is as "close" as possible to the given individual rankings. The voting system used by Dwork et al. [DKNS01] for this purpose is Kemeny's voting system [Kem59, KS60], for this is the unique voting system that is "neutral, consistent, and Condorcet" (see [DKNS01, HSV05]). However, even determining the winners of a given Kemeny election is a difficult computational task (as will be

[1] Completeness and hardness of a problem for a parameterized complexity class such as W[2] is defined with respect to parameterized reductions.

[2] "Prohibitive" here refers to a practical point of view. Of course, technically speaking, $O(n^k)$ is polynomial time whenever k is a fixed constant.

explained in more detail below), which is why Dwork et al. [DKNS01] designed a heuristic called "local Kemenization" on which their ranking algorithm is based.

An *election* $E = (C,V)$ is given by a set C of candidates and a set V of voters. A vote is typically input as a preference list over the candidates in C, i.e., as a linear ordering (or a permutation) of the candidates.[3] Although Kemeny in his original paper [Kem59] also allowed ties to occur in individual preference rankings, there is no consensus in the literature about this question. For clarity and notational convenience, we will use the term "permutation" to indicate that ties are not allowed, and we will use the term "preference ranking" when ties are allowed.

An *election system* (equivalently, a *voting system*) $\mathscr{E}$ is a rule that tells us how to determine the winners of an election. Formally, an election system is a mapping from elections $E = (C,V)$ to subsets of C. In this section, we introduce Kemeny's election system. Later on in Section 3, we are concerned with the election systems by Condorcet, Dodgson, Young, and Copeland as well as the plurality-rule election system.

Kemeny's system, with ties in individual preference rankings explicitly allowed [Kem59, KS60], works as follows. Let $E = (C,V)$ be a given election. To define "closeness" between any two preference rankings r and s on C, the following distance measure is used:

$$dist(r,s) = \sum_{\{c,d\}} D_{r,s}(c,d),$$

where the sum is taken over all unordered pairs $\{c,d\}$ of candidates in C and

$$D_{r,s}(c,d) = \begin{cases} 0 & \text{if } r \text{ and } s \text{ agree on } c \text{ and } d \\ 1 & \text{if one of } r \text{ and } s \text{ prefers } c \text{ to } d \text{ or } d \text{ to } c \text{ and} \\ & \text{the other one is indifferent with respect to } c \text{ and } d \\ 2 & \text{if } r \text{ and } s \text{ strictly disagree on } c \text{ and } d. \end{cases}$$

The *Kemeny score* of a given preference ranking r on C is defined as $KScore_E(r) = \sum_{v \in V} dist(r,v)$. For each candidate $c \in C$, define $KemenyScore_E(c) = \min_r KScore_E(r)$, where the minimum is taken over all preference rankings r on C such that no candidate is ranked higher than c in r. A preference ranking s on C such that $KScore_E(s)$ is minimum is said to be a *Kemeny consensus of* E, and a *Kemeny winner of* E is every candidate that is ranked on top of a Kemeny consensus of E. Let $KemenyScore_E$ denote the Kemeny score of a Kemeny consensus of E. Note that a Kemeny consensus of $E = (C,V)$ is not necessarily an element of V.

The decision problem KEMENY SCORE is defined as follows: Given an election $E = (C,V)$ and a positive integer k, is it true that $KemenyScore_E \leq k$? The decision problem KEMENY WINNER is defined as follows: Given an election $E = (C,V)$ and a distinguished candidate $c \in C$, does there exist a Kemeny consensus of E in which no candidate is ranked higher than c (i.e., is it true that $KemenyScore_E(c) \leq KemenyScore_E(d)$ for each $d \in C$)?

Example 2.1 *Consider the election $E = (C,V)$ with three candidates and four voters that is given by candidate set $C = \{b,c,d\}$ and voter set $V = \{v_1,v_2,v_3,v_4\}$, where the preference rankings of*

[3] By "linear ordering" we here mean a tie-free (i.e., strict) linear ordering.

the voters are

$$\begin{array}{ll} v_1: & b = c > d \\ v_2: & c > b = d \\ v_3: & b > c > d \\ v_4: & b = c > d \end{array}$$

Note that in this case the rankings of v_1 and v_4 are identical, so V can also be seen as a multiset of votes.

To compute the Kemeny consensus of E and to determine E's Kemeny winners, we have to find the minimum $KScore_E(r)$, where the minimum is taken over the 13 possible preference rankings r on C (with ties allowed). Table 1 shows how to obtain the value $KScore_E(r)$ for each such r.

ranking r	$dist(r,v_1)$	+	$dist(r,v_2)$	+	$dist(r,v_3)$	+	$dist(r,v_4)$	=	$KScore_E(r)$
$b = c = d$	2	+	2	+	3	+	2	=	9
$b = c > d$	0	+	2	+	1	+	0	=	3
$b > c = d$	2	+	4	+	1	+	2	=	9
$b > c > d$	1	+	3	+	1	+	1	=	6
$b = d > c$	4	+	4	+	3	+	4	=	15
$b > d > c$	3	+	5	+	2	+	3	=	13
$c > b = d$	2	+	0	+	3	+	2	=	7
$c > b > d$	1	+	1	+	2	+	1	=	5
$c = d > b$	4	+	2	+	5	+	4	=	15
$c > d > b$	3	+	1	+	4	+	3	=	11
$d > b = c$	4	+	4	+	5	+	4	=	17
$d > b > c$	5	+	5	+	4	+	5	=	19
$d > c > b$	4	+	3	+	6	+	4	=	17

Table 1: Determining $KScore_E(r)$ for each preference ranking r on C

For example, the first row of Table 1 is obtained as follows:

$$\begin{aligned} KScore_E(b = c = d) &= \sum_{v_i \in V} dist(b = c = d, v_i) \\ &= \sum_{v_i \in V} (D_{b=c=d,v_i}(b,c) + D_{b=c=d,v_i}(b,d) + D_{b=c=d,v_i}(c,d)) \\ &= (0+1+1)+(1+0+1)+(1+1+1)+(0+1+1) \\ &= 2+2+3+2 \ = \ 9. \end{aligned}$$

Obviously, $KScore_E(b = c > d) = 3$ is the minimum value, so $b = c > d$ is the Kemeny consensus, and since both b and c are the most preferred candidates in this Kemeny consensus, they both are Kemeny winners of E. In general, a Kemeny consensus need not be unique and need not be an element of V (which both happens to be the case in this example).

Bartholdi, Tovey, and Trick [BTT89b] proved that KEMENY SCORE is NP-complete and that KEMENY WINNER is NP-hard. Hemaspaandra, Spakowski, and Vogel [HSV05] precisely pinpointed the complexity of the latter problem by showing it complete for the complexity class $P_{||}^{NP}$, the class of sets solvable via a polynomial-time algorithm with parallel access to an NP oracle. Note that these complexity results hold regardless of whether or not ties are allowed in individual preference rankings (see [HSV05]).

2.2 Fixed-Parameter Tractability for Computing Kemeny Scores

Betzler et al. [BFG$^+$08] studied the problem of computing Kemeny scores from the point of view of parameterized complexity. In contrast with the above-mentioned NP-hardness results, they showed that computing a Kemeny score is fixed-parameter tractable. Again, their results refer to both cases, with and without ties allowed in individual preference rankings.

For simplicity, we focus on the case in which ties are not allowed. So, instead of preference rankings, we now consider permutations of the candidate set C. Then, for any two permutations ρ and σ on C, the distance measure is defined as $dist(\rho,\sigma) = \sum_{\{c,d\}} KT_{\rho,\sigma}(c,d)$, where the sum is again taken over all unordered pairs $\{c,d\}$ of candidates in C and the so-called *Kendall-Tau distance* (a.k.a. the number of inversions between two permutations) is defined as $KT_{\rho,\sigma}(c,d) = 0$ if ρ and σ rank c and d in the same order, and $KT_{\rho,\sigma}(c,d) = 1$ otherwise. All other notions are defined analogously with the notions defined in Section 2.1.

Throughout this paper, for all elections $E = (C,V)$, $m = \|C\|$ denotes the number of candidates and $n = \|V\|$ denotes the number of voters. To highlight the meaning of “parameter” in the context of parameterized complexity as opposed to its traditional meaning in other disciplines such as mathematical programming, we consider two parametrizations below: the number m of candidates and the maximum Kendall-Tau distance δ (to be defined below) between any two votes in the given election. While the number m of candidates in a given election is immediately clear from the problem instance, δ is “intrinsic” to the election. However, recall that it is enough that a function $\kappa : \Sigma^* \to \mathbb{N}$ is polynomial-time computable for it to be a parametrization in our sense. Indeed, note that the Kendall-Tau distance can be computed in time $O(m \log m)$ by a divide-and-conquer algorithm.

2.2.1 First Parametrization: Number of Candidates

Betzler et al. [BFG$^+$08] proved the following result, which says that when the number of candidates is a fixed constant, Kemeny scores can be computed in polynomial time. Note that in most (though not all) scenarios involving elections, assuming that the number of candidates is bounded by a small constant is a reasonable assumption;[4] it thus makes sense to consider the parameterized complexity of computing Kemeny scores with the number of candidates as the parameter.

Theorem 2.2 ([BFG$^+$08]) KEMENY SCORE *can be solved in time* $O(2^m \cdot m^2 \cdot n)$.

Let us sketch the proof of Theorem 2.2. Given an election $E = (C,V)$ and a positive integer k as input, the algorithm works as follows. For each subset $B \subseteq C$, the Kemeny score of the election

[4] An exception is the scenario given in Section 2.1, where web pages are viewed as candidates and search engines are viewed as voters. Clearly, in this scenario it is more reasonable to assume that the number of candidates is huge, whereas the number of voters is small.

$E_B = (B, V_B)$, where V_B denotes the restriction of the voter set V to the candidates in B, is to be computed recursively. The recurrence used for a given subset B is to consider every subset $B - \{c\}$ of B with one candidate deleted. For each $B \subseteq C$ and for each $c \in B$, let $\pi_{B-\{c\}}$ be a Kemeny consensus of election $(B - \{c\}, V_{B-\{c\}})$. Let π_B be the permutation obtained from $\pi_{B-\{c\}}$ by putting c in top position. Compute $K = \min_{B \subseteq C} KScore_{E_B}(\pi_B)$, where the minimum is taken over all subsets B of C. Output "yes" if $K \leq k$, and output "no" otherwise.

For the proof of correctness of this algorithm, we refer to [BFG+08].

2.2.2 Second Parametrization: Maximum Kendall-Tau Distance

Now choose the parameter $\delta = \max_{u,v \in V} dist(u,v)$, the maximum Kendall-Tau distance between any two votes in a given election $E = (C,V)$. Betzler et al. [BFG+08] designed a dynamic programming fixed-parameter algorithm for this parametrization. When δ is a fixed constant, their algorithm runs in polynomial time, in the technical sense of the definition of polynomial time. On the other hand, it should be noted that already for the rather small value of $\delta = 4$, which is far from being unrealistic in many settings, the running time of their algorithm amounts to $49\,816\,166\,400 \cdot a \cdot m \cdot n$, where the constant a is due to the O notation and $49\,816\,166\,400 = (3 \cdot 4 + 1)! \cdot 4 \cdot \log 4$. And for $\delta = 5$, we even have $(3 \cdot 5 + 1)! \cdot 5 \cdot \log 5 > 242\,808\,976\,650\,240$.

Theorem 2.3 ([BFG+08]) KEMENY SCORE *can be solved in time* $O((3\delta + 1)! \cdot \delta \cdot \log \delta \cdot m \cdot n)$, *where* δ *is the maximum Kendall-Tau distance between any two votes in the given election.*

Their algorithm uses the following notation. Let $E = (C,V)$ be a given election, and let π be any given permutation of the candidate set C. For each $c \in C$, the *position of* c *in* π is the number of candidates ranked higher than c in π. For example, the top candidate in π has position 0 and the bottom candidate in π has position $m - 1$. A *block of size* s *with start position* p, denoted by $\text{block}_s(p)$, is the set of candidates whose position in π is between p and $p + s - 1$ for at least one $\pi \in V$. Let $\text{block}(p) = \text{block}_{\delta+1}(p)$.

Let us roughly sketch the proof idea for Theorem 2.3. Given an election $E = (C,V)$ and a positive integer k as input, the fixed-parameter algorithm is based on decomposing blocks of size $\delta + 1$ for computing the Kemeny score of the given election. Consider $\text{block}(p)$. Every candidate $c \notin \text{block}(p)$ has a position that is either less than p for each voter in V or greater than $p + \delta + 1$ for each voter in V. Moreover, there exists a Kemeny consensus of E in which c's position is in a certain range that depends on δ. Thus, the algorithm proceeds by iterating from left to right over the given permutations in V, where it stores the Kemeny scores for all "partial orders" of the candidates in some block of size $\delta + 1$. Candidates that occur left of this block may be ignored.

In some more detail, the algorithm considers in each iteration some block of size $\delta + 1$. Initially, it considers all possible orders of the candidates in $\text{block}(0)$ and it stores in a table the Kemeny scores of all such orders with respect to the subelection $(\text{block}(0), V_{\text{block}(0)})$, where $V_{\text{block}(0)}$ denotes the restriction of the voter set V to the candidates in $\text{block}(0)$. In the ith iteration, $i \geq 1$, the algorithm considers $\text{block}(i)$ and computes the Kemeny scores of all orders of the candidates in $\text{block}(i)$ by looking up the table entries for $\text{block}(i-1)$. The size of the table is $O((3\delta + 1)! \cdot m)$. Finally, the Kemeny score of E is the minimum of the entries for $\text{block}(m - \delta - 1)$.

For the formal description of the algorithm and the proof of correctness, we refer to [BFG+08].

3 A Short Survey of Further FPT and Parameterized Complexity Results in Computational Social Choice

In this section we survey some further FPT and parameterized complexity results for problems arising in computational social choice. As early as in 1989, Bartholdi, Tovey, and Trick [BTT89b] studied NP-hard election problems for a bounded number of candidates or a bounded number of voters, and they obtained efficient fixed-parameter algorithm results for such cases. Within computational social choice, this approach of designing efficient fixed-parameter algorithms for hard problems was pursued in many papers, for example with respect to hard winner problems (see Section 2.2 and [BFG$^+$08, BGN08, RSV03]), with respect to hard problems related to procedural control [FHHR07, FHHR08, BU08], and with respect to hard problems related to bribery [FHH06] and lobbying [CFRS07]. In the remainder of this section, we are concerned with some of these results.

3.1 FPT Results and Parameterized Complexity for Determining Dodgson and Young Winners

As mentioned in Section 2.1, one of the useful properties of Kemeny's election system is that it respects the notion of a Condorcet winner [Con85]. A Condorcet winner is a candidate that defeats every other candidate in head-to-head comparison by a strict majority of votes. It is well known that Condorcet winners do not always exist, but are unique when they exist. An election system $\mathscr{E}$ is said to have the Condorcet property if whenever a Condorcet winner exists, he or she is also a winner of $\mathscr{E}$. Besides the Kemeny system, there are many other election systems that have the Condorcet property (see, e.g., Fishburn [Fis77]), including the systems proposed by Dodgson [Dod76] (who may be better known under his pen name, Lewis Carroll, the author of "Alice in Wonderland" and other beautiful children's books) and Young [You77].

Dodgson's system works as follows. Every candidate c in a given election is assigned a score: the smallest number of sequential swaps of adjacent candidates in the voters' preferences that are needed to make c a Condorcet winner, where swaps are counted separately for each vote and within each vote. Every candidate with a minimum Dodgson score wins. Clearly, a candidate with Dodgson score zero already is the Condorcet winner, as no swap has to be performed. Note that, unlike Condorcet winners, Dodgson winners always exist but are not necessarily unique.

The decision problem DODGSON SCORE is defined as follows: Given an election $E = (C,V)$, a designated candidate $c \in C$, and a positive integer k, is it true that c's Dodgson score is at most k? The decision problem DODGSON WINNER is defined as follows: Given an election $E = (C,V)$ and a distinguished candidate $c \in C$, is c a Dodgson winner of E?

Bartholdi, Tovey, and Trick [BTT89b] proved that DODGSON SCORE is NP-complete and that DODGSON WINNER is NP-hard. Hemaspaandra, Hemaspaandra, and Rothe [HHR97] optimally strengthened the latter result by raising the NP-hardness lower bound of this problem to match its obvious upper bound: DODGSON WINNER is complete for the complexity class $\mathrm{P}_{||}^{\mathrm{NP}}$ (the class capturing "parallel access to NP," as mentioned in Section 2.1).

Betzler, Guo, and Niedermeier [BGN08] investigated the systems by Dodgson and Young from a parameterized complexity perspective. In particular, they studied the complexity of DODGSON

SCORE parameterized by the number k of swaps needed. Their result says that if this number k is a fixed constant then Dodgson scores can be computed in polynomial time. Note that a small value of k for a given candidate means that this candidate is already “close” to being a Condorcet winner.

Note that individual votes in Dodgson elections are usually assumed to be (strictly) linear orders, i.e., they are permutations of the candidates. Hemaspaandra, Hemaspaandra, and Rothe [HHR97] noted that one may allow ties in individual votes and they suggested two natural models of how many swaps are needed in order to alterate the corresponding preference rankings. In the first model, we pay only one swap for changing, say, $a = b > c$ into $c > a = b$. In the second model, the same change costs two swaps. Betzler, Guo, and Niedermeier [BGN08] defined the corresponding problems of computing Dodgson scores with ties allowed in these two models, denoted by DODGSON TIE SCORE 1 and DODGSON TIE SCORE 2. Theorem 3.1 lists their results for Dodgson elections. Interestingly, the two models for how to handle ties yield different parameterized complexities.

Theorem 3.1 ([BGN08]) *The problems mentioned below are parameterized by the number k of swaps needed to make the given candidate a Condorcet winner.*

1. DODGSON SCORE *can be solved in time* $O(2^k \cdot n \cdot k + n \cdot m)$.
2. DODGSON TIE SCORE 2 *can be solved in time* $O(4^k \cdot n \cdot k + n \cdot m)$.
3. DODGSON TIE SCORE 1 *is* W[2]*-complete.*

Young’s election system [You77] works as follows: As in Dodgon’s system the goal is to make a candidate the Condorcet winner via the smallest possible alteration of the given election. However, in Young elections this goal is to be achieved by deleting a minimum number of voters. Formally, for an election $E = (C,V)$, the Young score of a candidate $c \in C$ is defined to be the maximum number of voters remaining in a subset V' of V such that c is the Condorcet winner in (C,V'). Every candidate with a maximum Young score is a Young winner. Clearly, a candidate with Young score $|V|$ already is the Condorcet winner, as no voter has to be deleted.

Define the decision problem YOUNG SCORE as follows: Given an election $E = (C,V)$, a designated candidate $c \in C$, and a positive integer k, is it true that c’s Young score is at least k? The dual probleem, DUAL YOUNG SCORE, focuses on the number of deleted voters: Given an election $E = (C,V)$, a designated candidate $c \in C$, and a positive integer k, can c be made a Condorcet winner by deleting at most k voters from V? Finally, the decision problem YOUNG WINNER is defined as follows: Given an election $E = (C,V)$ and a distinguished candidate $c \in C$, is c a Young winner of E? Rothe, Spakowski, and Vogel [RSV03] proved that, just as KEMENY WINNER and DODGSON WINNER, YOUNG WINNER is $\mathrm{P}_{||}^{\mathrm{NP}}$-complete. They also proved, with a linear program that modifies an integer linear program of Bartholdi et al. [BTT89b], that the winners in Fishburn’s homogeneous[5] variant of Dodgson’s system (see Fishburn [Fis77] for the definition) can be determined in polynomial time.

Betzler, Guo, and Niedermeier [BGN08] proved that both YOUNG SCORE and DUAL YOUNG SCORE are W[2]-complete, where the parameter k is the bound on the solution sizes given in the

[5]An election system $\mathscr{E}$ is said to be *homogeneous* if for each election (C,V) and for all positive integers q, we have $\mathscr{E}(C,V) = \mathscr{E}(C,qV)$, where qV denotes V replicated q times and $\mathscr{E}(C,V)$ denotes the set of winners of election (C,V) according to system $\mathscr{E}$.

instances of these problems. Containment of both problems in W[2] is shown via a reduction to the W[2]-complete problem OPTIMAL LOBBYING (which will be defined in Section 3.3). W[2]-hardness of YOUNG SCORE is shown via a slight modification of a reduction from the W[2]-hard problem SET PACKING presented in [RSV03]. W[2]-hardness of DUAL YOUNG SCORE is shown via a reduction from a W[2]-hard variant of the dominating set problem, which is called $k/2$-RED BLUE DOMINATING SET in [BGN08].

Theorem 3.2 ([BGN08]) *The problems mentioned below are parameterized by the size k of their solution sets.*

1. YOUNG SCORE *is* W[2]*-complete.*
2. DUAL YOUNG SCORE *is* W[2]*-complete.*

3.2 FPT Results for Procedural Control in Copeland Elections

Copeland [Cop51] proposed an election system that is based on pairwise comparisons of candidates (see Merlin and Saari [SM96, MS97] for a comprehensive treatise of Copeland elections). The winner of such a head-to-head contest receives one point, the loser receives no point, and both candidates may receive tie-related points. Since the literature is a bit ambiguous on how ties in such head-to-head contests are to be rewarded, Faliszewski et al. [FHHR07] introduced a parameterized variant of Copeland elections in which ties imply that both candidates receive α points, where the parameter α is a rational number with $0 \leq \alpha \leq 1$. They denote this system by Copeland$^{\alpha}$. Every candidate with the most points from head-to-head contests (including tie-related points) is a Copeland$^{\alpha}$ winner. The most common variant of "Copeland elections" in the literature is Copeland$^{0.5}$. It is interesting to note that the system here denoted by Copeland^{1} was proposed by the philosopher and religious missionary Ramon Llull already in the thirteenth century (see the literature pointers in [FHHR07, FHHR08]).

Since a Condorcet winner defeats every other candidate in head-to-head comparison, Copeland$^{\alpha}$ possesses the Condorcet property for each rational α, $0 \leq \alpha \leq 1$, just like the systems proposed by Kemeny, Dodgson, and Young. However, unlike these three systems, the winner problem and the related scoring problem for Copeland$^{\alpha}$ can be solved in polynomial time, and so is immediately fixed-parameter tractable in every parametrization.

Faliszewski et al. [FHHR07, FHHR08] studied Copeland$^{\alpha}$ with respect to procedural control and bribery. These are settings in which an external agent (called the chair for control and the briber for bribery scenarios) seeks to influence the outcome of a given election either via modifying its structure (namely by such actions as adding/deleting/partitioning voters or candidates), or via bribing certain voters in order to change their votes. We here focus on procedural control of elections. There are 22 different control scenarios that have been studied to date; we will not define them here formally but rather point the reader to the papers by Bartholdi, Tovey, and Trick [BTT89a, BTT92], who introduced the notion of constructive control (where the chair seeks to make a favorite candidate win), and by Hemaspaandra, Hemaspaandra, and Rothe [HHR07], who were the first to study destructive control (where the chair seeks to preclude the victory of a despised candidate), see also [FHHR07, FHHR08].

Among systems with a polynomial-time winner problem, Copeland$^{0.5}$ (and indeed Copeland$^{\alpha}$ for each rational α, $0 < \alpha < 1$) is the first natural election system proven to be fully resistant to constructive control [FHHR07, FHHR08] (see also [ENR08b] for a second such system), where "resistance" means that the corresponding control problem is NP-hard. To list just the numbers (with respect to the 22 different control scenarios considered) of such NP-hardness results for varying α, Faliszewski et al. [FHHR07, FHHR08] establish 14 resistances for Copeland$^{\alpha}$ when $\alpha \in \{0,1\}$, and 15 resistances for Copeland$^{\alpha}$ when $0 < \alpha < 1$.

In addition, Faliszewski et al. [FHHR07, FHHR08] studied the parameterized complexity of these 22 control problems with respect to Copeland$^{\alpha}$ for each rational α, $0 \leq \alpha \leq 1$. They consider two parameterizations, namely bounding the number of candidates and bounding the number of voters. For each of the above-mentioned NP-hardness results, the corresponding control problem is fixed-parameter tractable when the number of candidates is bounded. Also, for each of the above-mentioned NP-hardness results for voter control problems, fixed-parameter tractability holds when the number of voters is bounded. However, when the number of voters is bounded, the corresponding candidate-control cases remain open. The proofs of these FPT results are based on Lenstra's [Len83] algorithm for bounded-variable-cardinality integer programming.

Finally, we mention that Betzler and Uhlmann [BU08] have shown a number of parameterized intractability results (namely W[2]-completeness, W[2]-hardness, and W[1]-hardness) for control by adding candidates and control by deleting candidates within Copeland$^{\alpha}$. In particular, they partially solved the above-mentioned open question raised by Faliszewski et al. [FHHR07, FHHR08]: Constructive and destructive control by adding candidates and by deleting candidates remains NP-hard for Copeland$^{\alpha}$, where $\alpha \in \{0,1\}$, even for a fixed number of voters. In addition, Betzler and Uhlmann [BU08] studied the parameterized complexity of these control problems for Copeland$^{\alpha}$, where $0 \leq \alpha \leq 1$ is a rational, and for plurality voting[6] if the number of added/deleted candidates is viewed as a parameter.

3.3 Parameterized Complexity for Optimal Lobbying

Finally, we consider the problem OPTIMAL LOBBYING, another problem from computational social choice, albeit not related to a particular voting system. Christian et al. [CFRS07] defined this problem as follows: Given a 0-1 matrix (whose entries give the No/Yes votes for multiple referenda in the context of direct democracy), a positive integer *k*, and an external agent's (called the lobby) 0-1 target vector that gives the lobby's desired outcome of the referenda, is it possible for the lobby to reach this target by flipping the votes of at most *k* voters? Here, a column of the matrix (which represents one referendum) gives the outcome Yes if and only if a strict majority of voters cast a Yes vote, i.e., if and only if this column has a strict majority of ones.

The natural parametrization for this problem is the number *k* of voters the lobby needs to influence. Christian et al. [CFRS07] proved that this problem is intractable from the point of view of parameterized complexity: it is W[2]-complete. Theorem 3.3 is proven via a reduction from

[6] In plurality, each voter gives his or her most preferred candidate one point; whoever scores the most points is a plurality winner. Bartholdi, Tovey, and Trick [BTT92] obtained constructive control results for plurality and Hemaspaandra, Hemaspaandra, and Rothe [HHR07] obtained destructive and additional constructive control results for plurality.

the W[2]-complete problem DOMINATING SET to OPTIMAL LOBBYING (which shows OPTIMAL LOBBYING is W[2]-hard) and via a reduction from OPTIMAL LOBBYING to the W[2]-complete problem INDEPENDENT DOMINATING SET (which shows OPTIMAL LOBBYING is in W[2]). Both DOMINATING SET and INDEPENDENT DOMINATING SET were defined in Section 1.

Theorem 3.3 ([CFRS07]) OPTIMAL LOBBYING *is* W[2]-*complete.*

References

[BFG$^+$08] N. Betzler, M. Fellows, J. Guo, R. Niedermeier, and F. Rosamond. Fixed-parameter algorithms for Kemeny scores. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pages 60–71. Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008.

[BGN08] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. In *Proceedings of the 11th Scandinavian Workshop on Algorithm Theory*, pages 402–413. Springer-Verlag *Lecture Notes in Computer Science #5124*, July 2008.

[BI08] J. Buss and T. Islam. The complexity of fixed-parameter problems. *SIGACT News*, 39(1):34–46, March 2008.

[Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[BTT89a] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT89b] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[BTT92] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.

[BU08] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications*. Springer-Verlag *Lecture Notes in Computer Science*, July 2008. To appear.

[CF03] Y. Chen and J. Flum. Machine characterization of the classes of the W-hierarchy. In *Proceedings of the 17th International Workshop on Computer Science Logic*, pages 114–127. Springer-Verlag *Lecture Notes in Computer Science* #2803, August 2003.

[CFG03] Y. Chen, J. Flum, and M. Grohe. Bounded nondeterminism and alternation in parameterized complexity theory. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity*, pages 13–29. IEEE Computer Society Press, July 2003.

[CFRS07] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.

[Con85] J.-A.-N. de Caritat, Marquis de Condorcet. 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*, University of Michigan Press, 1995, pages 91–112.

[Cop51] A. Copeland. A “reasonable” social welfare function. Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan, 1951.

[DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. WWW'01*, pages 613–622. ACM Press, 2001.

[Dod76] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed “not yet published” (see the discussions in [MU95, Bla58], both of which reprint this paper), 1876.

[ENR08a] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proceedings of the 33rd International Symposium on Mathematical Foundations of Computer Science*, pages 311–322. Springer-Verlag *Lecture Notes in Computer Science #5162*, August 2008.

[ENR08b] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Technical Report cs.GT/0806.0535, ACM Computing Research Repository (CoRR), June 2008. A precursor appeared as [ENR08a].

[ER93] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proc. IJCAI'93*, pages 423–429, 1993.

[FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, 2006.

[FHH06] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 641–646. AAAI Press, July 2006.

[FHHR07] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 724–730. AAAI Press, July 2007.

[FHHR08] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pages 165–176. Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008.

[Fis77] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[FKS03] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proc. ACM SIGMOD Intern. Conf. on Management of Data*, pages 301–312. ACM Press, 2003.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proc. 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[HHR97] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, November 1997.

[HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[HSV05] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, December 2005.

[Kem59] J. Kemeny. Mathematics without numbers. *Dædalus*, 88:571–591, 1959.

[KS60] J. Kemeny and L. Snell. *Mathematical Models in the Social Sciences*. Ginn, 1960.

[Len83] H. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8:538–548, 1983.

[MS97] V. Merlin and D. Saari. Copeland method II: Manipulation, monotonicity, and paradoxes. *Journal of Economic Theory*, 72(1):148–172, 1997.

[MU95] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Michigan, 1995.

[Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[Pap94] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[Rot05] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[RR06] T. Riege and J. Rothe. Complexity of the exact domatic number problem and of the exact conveyor flow shop problem. *Theory of Computing Systems*, 39(5):635–668, September 2006.

[RSV03] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, June 2003.

[SM96] D. Saari and V. Merlin. The Copeland method I: Relationships and the dictionary. *Economic Theory*, 8(1):51–76, 1996.

[You77] H. Young. Extending Condorcet's rule. *Journal of Economic Theory*, 16(2):335–353, 1977.