On Determining Dodgson Winners by Frequently Self-Knowingly Correct Algorithms and in Average-Case Polynomial Time[*]

Jörg Rothe[†] and Holger Spakowski

Abstract

In their study of an efficient greedy heuristic for determining Dodgson winners, Homan and Hemaspaandra [HH06] introduced the notion of frequently self-knowingly correct algorithm. We show that this notion is closely related to Impagliazzo's notion of polynomial-time benign algorithm scheme [Imp95], which provides a model of average-case polynomial time. In particular, we show that every distributional problem (with respect to the uniform distribution) that has a polynomial-time benign algorithm scheme also has a frequently self-knowingly correct algorithm. Furthermore, we discuss Homan and Hemaspaandra's greedy heuristic with respect to AvgP, Levin's notion of average polynomial time [Lev86].

Key words: Preference aggregation, election systems, Dodgson elections, frequently self-knowingly correct algorithms, average-case complexity.

1 Introduction

This paper studies a novel type of algorithm, called frequently self-knowingly correct algorithm, and its relation to average-case polynomial time. Frequently self-knowingly correct algorithms were introduced by Homan and Hemaspaandra [HH06], who designed efficient such algorithms for solving the winner problem for Dodgson elections (which is known to be hard in the worst-case complexity model) with a guaranteed high frequency of success.

Before we turn to the main purpose of this paper, let us give a brief overview of recent results on complexity-theoretic issues related to voting in order to motivate this paper's topic in a broader framework. For more background on computational politics and the complexity of electoral problems, we refer to the comprehensive surveys [FHHR06, HH00].

Preference aggregation and election systems have been studied for centuries in social choice theory, political science, and economics, see, e.g., Arrow [Arr63],

[*]Supported in part by the German Science Foundation under grants RO 1202/9-1 and RO 1202/9-3.

[†]Supported in part by the Alexander von Humboldt Foundation in the TransCoop program.

Black [Bla58], Brams and Fishburn [BF83], and McLean and Urken [MU95]. Recently, these topics have become the focus of attention in various areas of computer science as well, such as artificial intelligence (especially with regard to distributed AI in multi-agent settings), computational complexity, and operations research. In the field of computational complexity, much work has been done during the past few years on the following four important classes of problems for various election systems.

Let $\mathcal{E}$ be a given election system. The *winner problem* for $\mathcal{E}$ asks whether a designated candidate has won a given election under $\mathcal{E}$. Bartholdi, Tovey, and Trick [BTT89b] proved that the winner problem for both Dodgson elections [Dod76] and Kemeny elections [Kem59, KS60] is NP-hard. Hemaspaandra, Hemaspaandra, and Rothe [HHR97] optimally improved the former result by proving the Dodgson winner problem complete (under polynomial-time many-one reductions) for $\mathrm{P}_{||}^{\mathrm{NP}}$, the class of problems solvable via parallel access to NP. This class is known to be equal to a number of other classes (including $\mathrm{P}_{\text{truth-table}}^{\mathrm{NP}}$ and $\mathrm{P}^{\mathrm{NP}[\log]}$, see [PZ83, Wag90]) and forms the Θ_2^p level of the polynomial hierarchy.

Rothe, Spakowski, and Vogel [RSV03] proved that the winner problem for Young elections [You77] is also $\mathrm{P}_{||}^{\mathrm{NP}}$-complete, and Hemaspaandra, Spakowski, and Vogel [HSV05] obtained the analogous result for the Kemeny winner problem. Each of these three election systems is based on a certain combinatorial optimization problem, and while each of these systems avoids the Condorcet Paradox, it does respect the Condorcet Principle [Con85, Fis77], i.e., it selects the Condorcet winner whenever one exists.[1] All the above results focus on "winner problems" (which ask if the designated candidate wins regardless of whether there are other winners as well) as opposed to "unique winner problems," and we also adopt the traditional model of winner problem here.[2]

The *control problem* for election system $\mathcal{E}$ asks whether it is feasible for an election chair to alter the outcome of an election by changing its agenda (see, e.g., [BTT92, HHR05]). The *manipulation problem* (a.k.a. the *strategic voting problem*) for $\mathcal{E}$ asks whether it is feasible that the outcome of an election can be altered by having voters strategically change their preferences (see, e.g., [BTT89a, BO91, CS02, CLS03, HH05]). The *bribery problem* for $\mathcal{E}$ (which is somewhat related to manipulation) asks whether it is feasible that the outcome of an election can be altered by an agent who bribes voters to change their votes. Of course, most desirable are voting systems whose winner problem is easy yet which resist electoral control, manipulation, and bribery.

We are concerned with the winner problem only. Since for some election systems with otherwise useful properties the winner problem is hard (in the worst-case complexity model), it is natural to wonder if one at least can find

[1]A Condorcet winner is a candidate who beats all other candidates in pairwise majority-rule contests.

[2]Note, however, that Hemaspaandra, Hemaspaandra, and Rothe [HHR06] have recently shown that also the unique winner problems for Dodgson, Young, and Kemeny elections are $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.

a heuristic algorithm solving the problem for "most of the inputs occurring in practice." Examples of such heuristics are known, e.g., for the Dodgson winner problem [HH06] and for the web page ranking problem in close relation to the Kemeny winner problem [DKNS01]. In particular, we study a heuristic called `GreedyWinner` for the problem of determining the winners of Dodgson elections. This heuristic is due to Homan and Hemaspaandra [HH06], who proved that if the number of voters greatly exceeds the number of candidates (which in many real-world cases is a very plausible assumption), then their heuristic is a "frequently self-knowingly correct algorithm," a notion they introduced in [HH06]. We show that this notion is closely related to average-case complexity.

This paper is organized as follows. In Section 2, we give a brief introduction to social choice theory (and, in particular, to Dodgson elections), we present some foundations of average-case complexity theory, and we define the notion of frequently self-knowingly correct algorithm. Section 3 provides our main result: Every problem in AvgP has a frequently self-knowingly correct algorithm. In Section 4, we discuss the relation of Homan and Hemaspaandra's greedy heuristic for finding Dodgson winners to average-case polynomial time. Finally, in Section 5, we conclude by raising some related open questions.

2 Preliminaries

2.1 Some Background from Social Choice Theory

An election $E = (C, V)$ is given by a set C of candidates and a set V of votes, where each vote is specified by a preference order on all candidates. As is common, we assume that the underlying preference relation is strict (i.e., irreflexive and antisymmetric), transitive, and complete.

In this paper, we focus on Dodgson elections. In 1876, Dodgson proposed an election system [Dod76] that is based on a combinatorial optimization problem: An election is won by those candidates who are "closest" to being a *Condorcet winner*, the unique candidate (if one exists) who defeats each other candidate in pairwise comparison by a strict majority of votes.

More precisely, given a Dodgson election $E = (C, V)$, every candidate c in C is assigned a score, which is denoted by $\text{DodgsonScore}(C, V, c)$ and is defined to be the smallest number of sequential exchanges of adjacent preferences in the voters' preference orders needed to make c a Condorcet winner with respect to the resulting preference orders. Whoever has the lowest Dodgson score wins.

The problem `DodgsonWinner` is defined as follows: Given an election $E = (C, V)$ and a designated candidate c in C, is c a Dodgson winner in E? (The search version of this decision problem can easily be derived.) As mentioned above, Hemaspaandra, Hemaspaandra, and Rothe [HHR97] have shown that this problem is $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.

It certainly is not desirable to have an election system whose winner problem is hard, as only systems that can be evaluated efficiently are actually used in practice. Fortunately, there are a number of positive results on Dodgson elections and related systems as well. In particular, Bartholdi, Tovey, and Trick [BTT89b] proved that for elections with a bounded number of candidates or voters Dodgson winners are easy to determine. Fishburn [Fis77] proposed a "homogeneous" variant of Dodgson elections that Rothe, Spakowski, and Vogel [RSV03] proved to have a polynomial-time winner problem. McCabe-Dansted, Pritchard, and Slinko [MDPS06] proposed a scheme (called Dodgson Quick) that approximates Dodgson's rule with an exponentially fast convergence. Finally, Homan and Hemaspaandra [HH06] proposed a greedy heuristic that finds Dodgson winners with a guaranteed high frequency of success. In particular, they introduced the notion of "frequently self-knowingly correct algorithm," which we define in Section 2.3 below, and they noted:

> "The closest related concepts [...] are probably those involving proofs to be verified, such as NP certificates and the proofs in interactive proof systems."

This statement notwithstanding, we will show that in fact it is the theory of average-case complexity (which Homan and Hemaspaandra also mention in their paper) that is even more closely related to their notion.

2.2 Foundations of Average-Case Complexity Theory

The theory of average-case complexity was initiated by Levin [Lev86]. A problem's average-case complexity can be viewed a more significant measure than its worst-case complexity in many cases, for example in cryptographic applications. We here follow Goldreich's presentation [Gol97]. Another excellent introduction to this theory is due to Wang [Wan97].

Fix the alphabet $\Sigma = \{0, 1\}$, let Σ^* denote the set of strings over Σ, and let Σ^n denote the set of all length n strings in Σ^*. For any $x, y \in \Sigma^*$, $x < y$ means that x precedes y in lexicographic order, and $x - 1$ denotes the lexicographic predecessor of x.

Intuitively, Levin observed that many hard problems—including those that are NP-hard in the traditional worst-case complexity model—may nonetheless be easy to solve "on the average," i.e., for "most" inputs or for "most practically relevant" inputs. He proposed to define the complexity of problems with respect to some suitable distribution on the input strings.

We now define the notion of a distributional problem and the complexity class AvgP. In subsequent sections, we consider two heuristic algorithms: the algorithm `GreedyWinner` intended to solve the decision problem `DodgsonWinner`, and the algorithm `GreedyScore` intended to compute the Dodgson score of some given candidate. Both heuristics work well sufficiently often, provided that the number of voters greatly exceeds the number of candidates.

Here, we define only distributional search problems; the definition of distributional decision problems is analogous.

Definition 1 *1. A* distribution function $\mu : \Sigma^* \to [0,1]$ *is a nondecreasing function from strings to the unit interval that converges to one (i.e., $\mu(0) \geq 0$ and $\mu(x) \leq \mu(y)$ for each $x < y$, and $\lim_{x\to\infty} \mu(x) = 1$). The* density function associated with μ *is defined by $\mu'(0) = \mu(0)$ and $\mu'(x) = \mu(x) - \mu(x-1)$ for each $x > 0$. That is, each string x gets weight $\mu'(x)$ with this distribution.*

2. A distributional (search) problem *is a pair (f, μ), where $f : \Sigma^* \to \Sigma^*$ is a function and $\mu : \Sigma^* \to [0,1]$ is a distribution function.*

3. A function $t : \Sigma^* \to \mathbb{N}$ is polynomial on the average with respect to some distribution μ *if there exists a constant $\epsilon > 0$ such that*

$$\sum_{x\in\Sigma^*} \mu'(x) \cdot \frac{t(x)^\epsilon}{|x|} < \infty.$$

4. The class AvgP *consists of all distributional problems (f, μ) for which there exists an algorithm $\mathcal{A}$ computing f such that the running time of $\mathcal{A}$ is polynomial on the average with respect to the distribution μ.*

In this paper, we focus on the standard uniform distribution μ on Σ^*, which is defined by

$$\mu'(x) = \frac{1}{|x|(|x|+1)2^{|x|}}.$$

That is, we first choose an input size n at random with probability $1/n(n+1)$, and then we choose an input string of that size n uniformly at random. For each $n \in \mathbb{N}$, let μ_n be the restriction of μ to strings of length at most n.

Impagliazzo [Imp95] introduced the notion of polynomial-time benign algorithm scheme to present an alternative view on the definition of average polynomial time.

Definition 2 ([Imp95]) *1. An algorithm computes a function f with* benign faults *if it either ouputs an element of the image of f or "?," and if it outputs anything other than ?, it is correct.*

2. A polynomial-time benign algorithm scheme *for a function f on μ_n is an algorithm $\mathcal{A}(x, \delta)$ such that:*

(a) $\mathcal{A}$ runs in time polynomial in $|x|$ and $1/\delta$.

(b) $\mathcal{A}$ computes f with benign faults.

(c) For each δ, $0 < \delta < 1$, and for each $n \in \mathbb{N}^+$,

$$\mathrm{Prob}_{\mu'_n}[\mathcal{A}(x,\delta) = \ ?] \leq \delta.$$

2.3 A Frequently Self-Knowingly Correct Greedy Algorithm

Homan and Hemaspaandra [HH06] define the following notion.

Definition 3 ([HH06]) *1. Let $f : S \to T$ be a function, where S and T are sets. We say an algorithm $\mathcal{A} : S \to T \times \{$"definitely", "maybe"$\}$ is* self-knowingly correct for f *if, for each $s \in S$ and $t \in T$, whenever $\mathcal{A}$ on input s outputs (t, "definitely") then $f(s) = t$.*

2. An algorithm $\mathcal{A}$ that is self-knowingly correct for $g : \Sigma^* \to T$ is said to be frequently self-knowingly correct for g *if*

$$\lim_{n\to\infty} \frac{||\{x \in \Sigma^n \mid A(x) \in T \times \{\text{"maybe"}\}\}||}{||\Sigma^n||} = 0.$$

In their seminal paper [HH06], Homan and Hemaspaandra present two frequently self-knowingly correct polynomial-time algorithms, which they call `GreedyScore` and `GreedyWinner`. Since `GreedyWinner` can easily be reduced to `GreedyScore`, we focus on `GreedyScore` only and briefly describe the intuition behind this algorithm; for full detail, we refer to [HH06].

If $E = (C, V)$ is an election and c is some designated candidate in C, we call (C, V, c) a *Dodgson triple*. Given a Dodgson triple (C, V, c), `GreedyScore` determines the Dodgson score of c with respect to the given election (C, V). We will see that there are Dodgson triples (C, V, c) for which this problem is particularly easy to solve.

For any $d \in C - \{c\}$, let Deficit[d] be the number of votes c needs to gain in order to have more votes than d in a pairwise contest between c and d.

Definition 4 *Any Dodgson triple (C, V, c) is said to be* nice *if for each candidate $d \in C - \{c\}$, there are at least* Deficit[d] *votes for which candidate c is exactly one position below candidate d.*

Given a Dodgson triple (C, V, c), the algorithm `GreedyScore` works as follows:

1. For each candidate $d \in C - \{c\}$, determine Deficit[d].
2. If (C, V, c) is not nice then output ("anything","maybe"); otherwise, output
$$(\textstyle\sum_{d\in C-\{c\}}\text{Deficit}[d], \text{"definitely"}).$$

Note that, for nice Dodgson triples, we have

$$\text{DodgsonScore}(C, V, c) = \sum_{d\in C-\{c\}} \text{Deficit}[d],$$

It is easy to see that `GreedyScore` is a self-knowingly correct polynomial-time bounded algorithm. To show that it is even *frequently* self-knowingly correct, Homan and Hemaspaandra prove the following key lemma. Their proof uses a variant of Chernoff bounds.

Lemma 5 (see Thm. 4.1(3) in [HH06]) *Let (C, V, c) be a given Dodgson triple with $n = ||V||$ votes and $m = ||C||$ candidates, chosen uniformly at random among all such Dodgson elections. The probability that (C, V, c) is not nice is at most*

$$2(m-1)e^{-\frac{n}{8m^2}}.$$

Homan and Hemaspaandra [HH06] show that the heuristic `GreedyWinner`, which is based on `GreedyScore` and which solves the winner problem for Dodgson elections, also is a frequently self-knowingly correct polynomial-time algorithm. This result is stated formally below.

Theorem 6 ([HH06]) *For all $m, n \in \mathbb{N}^+$, the probability that a Dodgson election E selected uniformly at random from all Dodgson elections having m candidates and n votes (i.e., all $(m!)^n$ Dodgson elections having m candidates and n votes have the same likelihood of being selected) has the property that there exists at least one candidate c such that* `GreedyWinner` *on input (E, c) outputs "maybe" as its second output component is less than $2(m^2 - m)e^{-\frac{n}{8m^2}}$.*

3 On AvgP and Frequently Self-Knowingly Correct Algorithms

Our main result relates polynomial-time benign algorithm schemes to frequently self-knowingly correct algorithms. We show that every distributional problem (with respect to the uniform distribution) that has a polynomial-time benign algorithm scheme must also have a frequently self-knowingly correct algorithm. It follows that all AvgP problems have a frequently self-knowingly correct algorithm.

Theorem 7 *Suppose that $\mathcal{A}(x, \delta)$ is a polynomial-time benign algorithm scheme for a distributional problem f on μ_n. Then there is a frequently self-knowingly correct algorithm $\mathcal{A}'$ for f.*

Proof For each $n \in \mathbb{N}$, let $\delta(n) = 1/n^3$. Define the algorithm $\mathcal{A}'$ as follows:

1. On input $x \in \Sigma^*$, simulate $\mathcal{A}(x, \delta(|x|))$.
2. If $\mathcal{A}(x, \delta(|x|))$ outputs ?, then output (*anything*, *"maybe"*).
3. If $\mathcal{A}(x, \delta(|x|))$ outputs $y \in T$, where $y \neq ?$, then output (y, *"definitely"*).

By definition of "polynomial-time benign algorithm scheme," algorithm $\mathcal{A}'$ is polynomial-time bounded. It remains to show that $\mathcal{A}'$ is frequently self-knowingly correct.

Fix an arbitrary $n \in \mathbb{N}$. Then

$$\begin{aligned}
& \frac{||\{x \in \Sigma^n \mid \mathcal{A}'(x) \in T \times \{\text{“maybe”}\}\}||}{||\Sigma^n||} \\
= \; & \mathrm{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ? \mid |x| = n] \\
= \; & \frac{\mathrm{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ? \text{ and } |x| = n]}{\mathrm{Prob}_{\mu'_n}[|x| = n]} \\
\leq \; & \frac{\mathrm{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?]}{\mathrm{Prob}_{\mu'_n}[|x| = n]}
\end{aligned}$$

$$\begin{aligned}
\leq \; & \frac{\mathrm{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?]}{1/n(n+1)} && (1) \\
= \; & n(n+1) \cdot \mathrm{Prob}_{\mu'_n}[\mathcal{A}(x, \delta(|x|)) = ?] \\
\leq \; & n(n+1) \cdot \delta(|x|) && (2) \\
= \; & \frac{n(n+1)}{n^3} \\
\leq \; & \frac{n+1}{n^2}.
\end{aligned}$$

Here, (1) holds because, by definition of μ,

$$\mathrm{Prob}_{\mu'_n}[|x| = n] \geq 1/n(n+1),$$

and (2) is true by the definition of polynomial-time benign algorithm scheme.

We have shown that

$$\lim_{n \to \infty} \frac{||\{x \in \Sigma^n \mid \mathcal{A}'(x) \in T \times \{\text{“maybe”}\}\}||}{||\Sigma^n||} = 0,$$

which completes the proof. ▮

Corollary 8 *Every distributional problem (under the standard uniform distribution) that is in* AvgP *has a frequently self-knowingly correct algorithm.*

Proof Impagliazzo proved that any distributional problem on input ensemble μ_n is in AvgP if and only if it has a polynomial-time benign algorithm scheme; see Proposition 2 in [Imp95]. The claim now follows from Theorem 7. ▮

4 Dodgson Winners and Average-Case Polynomial Time

Because of the close relationship between the notion of frequently self-knowingly correctness and average-case complexity, one might think that Homan and Hemaspaandra's algorithm could be used to devise an algorithm, call it

DodgsonWinner-AverageGood, witnessing that the problem DodgsonWinner is in AvgP (assuming the uniform probability distribution on all elections with the same number of candidates and voters). A tempting approach towards this goal would be as follows. Given an election, run the GreedyWinner algorithm on it. If that fails (i.e., if GreedyWinner outputs "maybe"), then use the exhaustive algorithm that works in time $O(m^n)$.

We now show that this approach does not work in any obvious way. The reason is that the algorithm is not "frequently enough" self-knowingly correct. That is, DodgsonWinner-AverageGood would have to spend too much time (namely, $O(m^n)$) on a too large fraction of the inputs, namely,

$$2(m^2 - m)e^{-\frac{n}{8m^2}}.$$

Let $\mu'(m, n)$ be the probability that an election chosen randomly under the uniform distribution has m candidates and n voters. We assume that the elections are encoded into strings in a reasonable way. Use, for instance, the encoding scheme given by Homan and Hemaspaandra [HH06].

To prove that DodgsonWinner is in AvgP, we would have to show that there is an $\epsilon > 0$ such that

$$\sum_{x \in \Sigma^*} \mu'(x) \frac{t(x)^\epsilon}{|x|} < \infty,$$

where $t(x)$ is the computation time of the algorithm DodgsonWinner-AverageGood on input x, see Definition 1.

Let $E(m, n)$ be the set of strings in Σ^* that encode elections having m candidates and n voters.

For any $x \in E(m, n)$,

$$\mu'(x) = \frac{1}{||E(m, n)||} \cdot \mu'(m, n).$$

By Theorem 6, for fixed m and n, we obtain

$$\sum_{x \in E(m,n)} \mu'(x) \frac{t(x)^\epsilon}{|x|} \leq \mu'(m, n) \cdot \left(\frac{p(|x|)^\epsilon}{|x|} + \alpha(m, n) \cdot (cm^n)^\epsilon \right),$$

where $\alpha(m, n) = 2(m^2 - m)e^{-\frac{n}{8m^2}}$. Here, p is the polynomial that bounds the computation time of GreedyWinner. We can clearly choose $\epsilon > 0$ small enough such that $p(|x|)^\epsilon/|x|$ contributes only a constant in the above term. However, the crucial fact is that the term

$$\alpha(m, n) \cdot (cm^n)^\epsilon = 2(m^2 - m)e^{-\frac{n}{8m^2}} \cdot (cm^n)^\epsilon$$

grows exponentially, no matter how small we choose ϵ.

Thus, DodgsonWinner-AverageGood does not run in average-case polynomial time with respect to any interesting distribution on the inputs.

5 Conclusions

Homan and Hemaspaandra [HH06] proposed a greedy heuristic for finding Dodgson winners, and they proved that this heuristic is a frequently self-knowingly correct algorithm. We have shown that every distributional problem (with respect to the standard uniform distribution) in AvgP has a frequently self-knowingly correct algorithm. It is easy to see that the converse implication is not true. (For instance, one can define a problem L that consists only of strings in $\{0\}^*$ encoding the halting problem. This problem is clearly not in AvgP, yet it is frequently self-knowingly correct.)

Furthermore, we have argued why it might be hard to show that the problem of finding Dodgson winners—at least via Homan and Hemaspaandra's heuristic—can be solved in polynomial time on the average. We suspect that this problem is hard on the average, but a rigorous proof for this claim has eluded us so far, and we raise this as an open question. In particular, it would be interesting to study the average-case complexity of the Dodgson winner problem with respect to suitable distributions (see, e.g., Procaccia and Rosenschein [PR06]).

Other interesting issues remain open as well. For example, one might study the approximability of the Dodgson winner problem; some first steps in this direction have been taken by McCabe-Dansted, Pritchard, and Slinko [MDPS06]. Also, one could investigate other voting systems with a hard winner problem in the worst-case model, such as the problem of determining Young winners [RSV03] or the problem of determining Kemeny winners [HSV05], and seek to find algorithms that can be shown to be frequently self-knowingly correct or even average-case polynomial-time, or else seek to prove these problems hard on the average.

Acknowledgments: We thank Lane A. Hemaspaandra and Chris Homan for their interest in this work, and for many inspiring discussions on computational issues related to voting. We also thank the COMSOC '06 referees for their helpful comments.

References

[Arr63] K. Arrow. *Social Choice and Individual Values.* John Wiley and Sons, 1951 (revised edition 1963).

[BF83] S. Brams and P. Fishburn. *Approval Voting.* Birkhäuser, Boston, 1983.

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

[BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[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.

[CLS03] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 201–214. ACM Press, 2003.

[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.

[CS02] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, 2002.

[DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, 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.

[FHHR06] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer-Verlag, 2006. To appear.

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

[Gol97] O. Goldreich. Note on Levin's theory of average-case complexity. Technical Report TR97-058, Electronic Colloquium on Computational Complexity, November 1997.

[HH00] E. Hemaspaandra and L. Hemaspaandra. Computational politics: Electoral systems. In *Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science*, pages 64–83. Springer-Verlag *Lecture Notes in Computer Science #1893*, 2000.

[HH05] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. Technical Report TR-861, University of Rochester, Department of Computer Science, Rochester, NY, April 2005.

[HH06] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, pages 528–539. Springer-Verlag *Lecture Notes in Computer Science #4162*, August/September 2006.

[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.

[HHR05] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 95–101. AAAI Press, 2005.

[HHR06] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report TR-900, Department of Computer Science, University of Rochester, Rochester, NY, June 2006. Revised, August 2006.

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

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society Press, 1995.

[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.

[Lev86] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[MDPS06] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson's rule. Technical Report TR-551, Auckland University, Department of Mathematics, Auckland, New Zealand, June 2006.

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

[PR06] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 497–504. ACM Press, May 2006.

[PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.

[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.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[Wan97] J. Wang. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 295–328. Springer-Verlag, 1997.

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

Jörg Rothe
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `rothe@cs.uni-duesseldorf.de`

Holger Spakowski
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
40225 Düsseldorf, Germany
Email: `spakowsk@cs.uni-duesseldorf.de`