On Approximating Optimal Weighted Lobbying, and Frequency of Correctness versus Average-Case Polynomial Time*

Gábor Erdélyi[1], Lane A. Hemaspaandra[2], Jörg Rothe[1], and Holger Spakowski[1]

[1] Inst. für Informatik, Universität Düsseldorf, 40225 Düsseldorf, Germany
[2] Dept. of Computer Science, University of Rochester, Rochester, NY 14627, USA

Abstract. We investigate issues regarding two hard problems related to voting, the optimal weighted lobbying problem and the winner problem for Dodgson elections. Regarding the former, Christian et al. [2] showed that optimal lobbying is intractable in the sense of parameterized complexity. We provide an efficient greedy algorithm that achieves a logarithmic approximation ratio for this problem and even for a more general variant—optimal weighted lobbying. We prove that essentially no better approximation ratio than ours can be proven for this greedy algorithm.

The problem of determining Dodgson winners is known to be complete for parallel access to NP [11]. Homan and Hemaspaandra [10] proposed an efficient greedy heuristic for finding Dodgson winners with a guaranteed frequency of success, and their heuristic is a "frequently self-knowingly correct algorithm." We prove that every distributional problem solvable in polynomial time on the average with respect to the uniform distribution has a frequently self-knowingly correct polynomial-time algorithm. Furthermore, we study some features of probability weight of correctness with respect to Procaccia and Rosenschein's junta distributions [15].

1 Introduction

Preference aggregation and election systems have been studied for centuries in social choice theory, political science, and economics. 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 multiagent settings), systems (e.g., for spam filtering), and computational complexity.

* URLs: `ccc.cs.uni-duesseldorf.de/{~erdelyi/, ~rothe/, ~spakowski/}` (G. Erdélyi, J. Rothe, and H. Spakowski) and `www.cs.rochester.edu/u/www/u/lane/` (L. Hemaspaandra). Supported in part by DFG grants RO 1202/9-1 and RO 1202/9-3, NSF grant CCF-0426761, the Alexander von Humboldt Foundation's TransCoop program, and a Friedrich Wilhelm Bessel Research Award. Work done in part while the second author was visiting Heinrich-Heine-Universität Düsseldorf. Some of the results of Section 3 of this paper were presented at the *First International Workshop on Computational Social Choice*, December 2006.

This paper's topic is motivated by two hard problems that both are related to voting, the optimal weighted lobbying problem and the winner problem for Dodgson elections. Regarding the former problem, Christian et al. [2] defined its unweighted variant as follows: Given a 0-1 matrix that represents the No/Yes votes for multiple referenda in the context of direct democracy, a positive integer k, and a target vector (of the outcome of the referenda) of an external actor ("The Lobby"), is it possible for The Lobby to reach its target by changing the votes of at most k voters? They proved the optimal lobbying problem complete for the complexity class W[2], thus providing strong evidence that it is intractable even for small values of the parameter k. However, The Lobby might still try to find an approximate solution efficiently. We propose an efficient greedy algorithm that establishes the first approximation result for the weighted version of this problem in which each voter has a price for changing his or her 0-1 vector to The Lobby's specification. Our approximation result applies to Christian et al.'s original optimal lobbying problem (in which each voter has unit price), and also provides the first approximation result for that problem. In particular, we achieve logarithmic approximation ratios for both these problems.

The Dodgson winner problem was shown NP-hard by Bartholdi, Tovey, and Trick [1]. Hemaspaandra, Hemaspaandra, and Rothe [11] optimally improved this result by showing that the Dodgson winner problem is complete for $\mathrm{P}_{\|}^{\mathrm{NP}}$, the class of problems solvable via parallel access to NP. Since these hardness results are in the worst-case complexity model, it is natural to wonder if one at least can find a heuristic algorithm solving the problem efficiently for "most of the inputs occurring in practice." Homan and Hemaspaandra [10] proposed a heuristic, called Greedy-Winner, for finding Dodgson winners. They 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 to formally capture a strong notion of the property of "guaranteed success frequency" [10]. We study this notion in relation with average-case complexity. We also investigate Procaccia and Rosenschein's notion of deterministic heuristic polynomial time for their so-called junta distributions, a notion they introduced in their study of the "average-case complexity of manipulating elections" [15]. We show that under the junta definition, when stripped to its basic three properties, every NP-hard set is $\leq_m^p$-reducible to a set in deterministic heuristic polynomial time relative to some junta distribution and we also show a very broad class of sets (including many NP-complete sets) to be in deterministic heuristic polynomial time relative to some junta distribution. We note (see also [17]) that the "average-case complexity" results of [15] are not really average-case complexity results (in the sense of being about some sort of averaging of running times), but rather are frequency of correctness—or, to be more precise, probability weight of correctness—results (as are also the results of Homan and Hemaspaandra).

This paper is organized as follows. In Section 2, we propose and analyze an efficient greedy algorithm for approximating the optimal weighted lobbying problem. In Section 3, we show that every problem solvable in average-case polynomial time with respect to the uniform distribution has a frequently self-knowingly correct polynomial-time algorithm, and we study Procaccia and Rosenschein's junta distributions. The heuristic Greedy-Score on which Greedy-Winner is based [10], some technical definitions from average-case complexity theory [13, 9, 18], and the proofs omitted due to space constraints can be found in the full version of this paper [6].

2 Approximating Optimal Weighted Lobbying

2.1 Optimal Lobbying and its Weighted Version

Christian et al. [2] introduced and studied the following problem. Suppose there are m voters who vote on n referenda, and there is an external actor, which is referred to as "The Lobby" and seeks to influence the outcome of these referenda by making voters change their votes. It is assumed that The Lobby has complete information about the voters' original votes, and that The Lobby's budget allows for influencing the votes of a certain number, say k, of voters. Formally, the Optimal-Lobbying problem is defined as follows: Given an $m \times n$ 0-1 matrix V (whose rows represent the voters, whose columns represent the referenda, and whose 0-1 entries represent No/Yes votes), a positive integer $k \leq m$, and a target vector $x \in \{0,1\}^n$, is there a choice of k rows in V such that by changing the entries of these rows the resulting matrix has the property that, for each j, $1 \leq j \leq n$, the jth column has a strict majority of ones (respectively, zeros) if and only if the jth entry of the target vector x of The Lobby is one (respectively, zero) [2]?

Christian et al. [2] showed that Optimal-Lobbying (with respect to parameter k, the number of voters influenced by The Lobby) is complete for the complexity class W[2]; see, e.g., Downey and Fellows [4] and Flum and Grohe [7] for background on the theory of parameterized complexity and in particular for the definition of W[2].

This result is considered strong evidence that Optimal-Lobbying is intractable, even for small values of the parameter k. However, even though the optimal goal of The Lobby cannot be achieved efficiently, it might be approximable within some factor. That is, given an $m \times n$ 0-1 matrix V and a target vector $x \in \{0,1\}^n$, The Lobby might try to reach its target by changing the votes of as few voters as possible.

We consider the more general problem Optimal-Weighted-Lobbying, where we assume that influencing the 0-1 vector of each voter v_i exacts some price, $\mathit{price}(v_i) \in \mathbb{Q}$, where $\mathbb{Q}$ denotes the set of nonnegative rational numbers. In this scenario, The Lobby seeks to minimize the amount of money spent to reach its goal. The problem Optimal-Lobbying (redefined as an optimization problem rather than a parameterized problem) is the unit-prices special case of Optimal-Weighted-Lobbying, i.e., where $\mathit{price}(v_i) = 1$ for each voter v_i. It follows that Optimal-Weighted-Lobbying (redefined as a parameterized rather than an optimization problem, where the parameter is The Lobby's budget of money to be spent) inherits the W[2]-hardness lower bound from its special case Optimal-Lobbying, and that the logarithmic approximation algorithm we build for Optimal-Weighted-Lobbying will provide the same approximation ratio for Optimal-Lobbying.

In the remainder of this section, we describe and analyze an efficient greedy algorithm for approximating Optimal-Weighted-Lobbying.

2.2 A Greedy Algorithm for Optimal Weighted Lobbying

Let a matrix $V \in \{0,1\}^{m \times n}$ be given, where the columns $r_1, r_2, \dots, r_n$ of V represent the referenda and the rows $v_1, v_2, \dots, v_m$ of V represent the voters. Without loss of generality, we may assume that The Lobby's target vector is of the form $x = 1^n$ (and thus may be dropped from the problem instance), since if there is a zero in x at position j, we can simply flip this zero to one and also flip the corresponding zeros and ones in column r_j.

For each column r_j, define the *deficit* d_j to be the minimum number of zeros that need to be flipped to ones such that there are strictly more ones than zeros in this column. Let $D_0 = \sum_{j=1}^{n} d_j$ be the sum of all initial deficits.

Figure 1 gives the greedy algorithm, which proceeds by iteratively choosing a most "cost-effective" row of V and flipping to ones all those zeros in this row that belong to columns with a positive deficit, until the deficits in all columns have decreased to zero. We assume that ties between rows with equally good cost-effectiveness are broken in any simple way, e.g., in favor of the tied v_i with lowest i.

Let R be the set of columns of V whose deficits have already vanished at the beginning of an iteration, i.e., all columns in R already have a strict majority of ones. Let $v_{i\rceil R^c}$ denote the entries of v_i restricted to those columns not in R, and let $\#_0(v_{i\rceil R^c})$ denote the number of zeros in $v_{i\rceil R^c}$. (For i such that $\#_0(v_{i\rceil R^c}) = 0$, we consider $\mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$ to be $+\infty$.) During an iteration, the *cost per flipped entry in row* v_i (for decreasing the deficits in new columns by flipping v_i's zeros to ones) is $\mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$. We say a voter v_i is *more cost-effective* than a voter v_j if v_i's cost per flipped entry is less than v_j's. When our algorithm chooses to alter a row v_i, we will think of its price being distributed equally among the new columns with decreased deficit, and at that instant will permanently associate with every flipped entry, e_k, in that row its portion of the cost, i.e., $\mathit{cost}(e_k) = \mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$.

Clearly, the greedy algorithm in Figure 1 always stops, and its running time is polynomial, since the while loop requires only linear (in the input size) time and has to be executed at most $D_0 = \sum_{j=1}^{n} d_j \leq n \cdot \lceil (m+1)/2 \rceil$ times (note that at most $\lceil (m+1)/2 \rceil$ flips are needed in each column to achieve victory for The Lobby's position).

Now, enumerate the D_0 entries of V that have been flipped in the order in which they were flipped by the algorithm. Let $e_1, e_2, \dots, e_{D_0}$ be the resulting enumeration. Let OPT be the money that would be spent by The Lobby for an optimal choice of voters such that its target is reached.

```
1. Input: A matrix $V \in \{0,1\}^{m \times n}$.
2. Initialize:
       Compute the deficits $d_j$, $1 \leq j \leq n$.
       $D \leftarrow \sum_{j=1}^{n} d_j$.                    /* Initially, $D = D_0$. */
       $X \leftarrow \emptyset$.
3. While $D \neq 0$ do
       Let $R$ be the set of columns $r_j$ with $d_j = 0$.
       Find a voter whose cost-effectiveness is greatest, say $v_i$.
       Let $\gamma_i = price(v_i)/\#_0(v_{i\rceil R^c})$.
       Choose $v_i$ and flip all zeros in $v_{i\rceil R^c}$ to ones.
       For each flipped entry $e$ in $v_i$, let $cost(e) = \gamma_i$.
                                   /* $cost(e)$ will be used in our analysis. */
       $X \leftarrow X \cup \{i\}$.
       $d_j \leftarrow d_j - 1$, for each column $r_j$ for which a zero was flipped.
       $D \leftarrow \sum_{j=1}^{n} d_j$.
4. Output: $X$.
```

Fig. 1. Greedy algorithm for Optimal-Weighted-Lobbying

Lemma 1. *For each $k \in \{1, 2, \ldots, D_0\}$, we have $cost(e_k) \leq \mathrm{OPT}/(D_0 - k + 1)$.*

The proof of Lemma 1 can be found in the full version of this paper [6].

Theorem 1. *The greedy algorithm presented in Figure 1 approximates the problem Optimal-Weighted-Lobbying with approximation ratio at most*

$$\sum_{i=1}^{D_0} \frac{1}{i} \leq 1 + \ln D_0 \leq 1 + \ln\left(n \left\lceil \frac{m+1}{2} \right\rceil\right).$$

Proof. The total price of the set of voters X picked by the greedy algorithm is the sum of the costs of those entries flipped. That is, $price(X) = \sum_{i \in X} price(v_i) = \sum_{k=1}^{D_0} cost(e_k) \leq \left(1 + \frac{1}{2} + \cdots + \frac{1}{D_0}\right) \cdot \mathrm{OPT}$, where the last inequality follows from Lemma 1. ❑

Since the input size is lower-bounded by $m \cdot n$, Theorem 1 establishes a logarithmic approximation ratio for Optimal-Weighted-Lobbying (and also for Optimal-Lobbying). Note that the proof of Theorem 1 establishes an approximation ratio bound that is (sometimes nonstrictly) stronger than $\sum_{i=1}^{D_0} 1/i$. In particular, if the number of zeros flipped in successive iterations of the algorithm's while loop are $\ell_1, \ell_2, \ldots, \ell_p$, where $\ell_1 + \ell_2 + \cdots + \ell_p = D_0$, then the proof gives a bound on the approximation ratio of

$$\frac{\ell_1}{D_0} + \frac{\ell_2}{D_0 - \ell_1} + \cdots + \frac{\ell_p}{D_0 - (\ell_1 + \cdots + \ell_{p-1})} = \sum_{j=1}^{p} \frac{\ell_j}{D_0 - \sum_{k=1}^{j-1} \ell_k}.$$

	r_1	r_2	r_3	$\cdots$	r_n	$price(v_i)$
v_1	0	1	1	$\cdots$	1	1
v_2	1	0	1	$\cdots$	1	1/2
v_3	1	1	0	$\cdots$	1	1/3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
v_n	1	1	1	$\cdots$	0	$1/n$
v_{n+1}	0	0	0	$\cdots$	0	$1+\epsilon$
v_{n+2}	1	0	0	$\cdots$	0	2
v_{n+3}	0	1	0	$\cdots$	0	2
v_{n+4}	0	0	1	$\cdots$	0	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
v_{2n+1}	0	0	0	$\cdots$	1	2

Table 1. A tight example for the greedy algorithm in Figure 1

This is strictly better than $\sum_{i=1}^{D_0} 1/i$ except in the case that each ℓ_j equals 1. And this explains why, in the example we are about to give that shows that the algorithm can at times yield a result with ratio essentially no better than $\sum_{i=1}^{D_0} 1/i$, each ℓ_j will equal 1.

Now, we show that the $\sum_{i=1}^{D_0} 1/i$ approximation ratio stated in Theorem 1 is essentially the best possible that can be stated for the greedy algorithm of Figure 1. Consider the example given in Table 1. The prices for changing the voters' 0-1 vectors are shown in the right-most column of Table 1: Set $price(v_i) = 1/i$ for each $i \in \{1, 2, \ldots, n\}$, set $price(v_i) = 2$ for each $i \in \{n+2, n+3, \ldots, 2n+1\}$, and set $price(v_{n+1}) = 1 + \epsilon$, where $\epsilon > 0$ is a fixed constant that can be set arbitrarily small. Note that, for each j, $1 \leq j \leq n$, we have $d_j = 1$, and hence $D_0 = n$.

When run on this input, our greedy algorithm sequentially flips, for $i = n, n-1, \ldots, 1$, the single zero-entry of voter v_i to a one. Thus the total money spent is $1 + 1/2 + \cdots + 1/n = 1 + 1/2 + \cdots + 1/D_0$. On the other hand, the optimal choice consists of influencing just voter v_{n+1} by flipping all of v_{n+1}'s entries to ones, which costs only $1 + \epsilon$.

3 Frequency of Correctness versus Average-Case Polynomial Time

3.1 A Motivation: How to Find Dodgson Winners Frequently

An election (C, V) is given by a set C of candidates and a set V of voters, where each vote is specified by a preference order on all candidates and the underlying preference relation is strict (i.e., irreflexive and antisymmetric), transitive, and complete. A Condorcet winner of an election is a candidate i such that for each candidate $j \neq i$, a strict majority of the voters prefer i to j. Not all elections have a Condorcet winner, but when a Condorcet winner exists, he or she is unique. In 1876, Dodgson [5] proposed an election system that is based on a combinatorial optimization problem: An election is won by those candidates who are "closest" to being a Condorcet winner. More precisely, given a Dodgson election (C, V), every candidate c in C is assigned a score, denoted by DodgsonScore(C, V, c), which gives 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 Dodgson-Winner is defined as follows: Given an election (C, V) and a designated candidate c in C, is c a Dodgson winner in (C, V)? (The search version of this decision problem can easily be stated.) As mentioned earlier, Hemaspaandra et al. [11] 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 (see, e.g., [1, 8, 16, 14]). One of these positive results is due to Homan and Hemaspaandra [10] who proposed a greedy heuristic that finds Dodgson winners with a "guaranteed high frequency of success." To capture a strengthened version of this property formally, they introduced the notion of a "frequently self-knowingly correct algorithm."

Definition 1 ([10]). *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 $f(s) = t$. 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*

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

3.2 On AvgP and Frequently Self-Knowingly Correct Algorithms

The theory of average-case complexity was initiated by Levin [13]. A problem's average-case complexity can be viewed as a more significant measure than its worst-case complexity in many cases, for example in cryptographic applications. For an excellent introduction to this theory, we refer to Goldreich [9] and Wang [18]. Formal definitions can be found there and in the full version of this paper [6]. An alternative view of the definition of Levin's class average polynomial time (AvgP) was provided by Impagliazzo [12].

Definition 2 ([12]). *An algorithm computes a function f with* benign faults *if it either outputs an element of the image of f or "?," and if it outputs anything other than "?" it is correct. For any distribution μ on Σ^*, let $\mu_{\leq n}$ denote the restriction of μ to strings of length at most n. A* polynomial-time benign algorithm scheme *for a function f on μ is an algorithm $\mathcal{A}(x, \delta)$ such that:*

1. *$\mathcal{A}$ runs in time polynomial in $|x|$ and $1/\delta$.*
2. *$\mathcal{A}$ computes f with benign faults.*
3. *For each δ, $0 < \delta < 1$, and for each $n \in \mathbb{N}^+$, $\mathrm{Prob}_{\mu_{\leq n}}[\mathcal{A}(x,\delta) = \ ?] \leq \delta$.*

Our main result in this section is that every distributional problem that has a polynomial-time benign algorithm scheme with respect to the uniform distribution must also have a frequently self-knowingly correct polynomial-time algorithm. It follows that all uniformly distributed AvgP problems have a frequently self-knowingly correct polynomial-time algorithm. The proofs of Theorem 2 and Proposition 1 (which says that the converse implication of that of Corollary 1 below is not true) can be found in the full version of this paper [6].

Theorem 2. *Suppose that $\mathcal{A}(x,\delta)$ is a polynomial-time benign algorithm scheme for a distributional problem f on the standard uniform distribution. Then there is a frequently self-knowingly correct polynomial-time algorithm $\mathcal{A}'$ for f.*

Theorem 2 and Proposition 2 in [12] establish the following corollary.

Corollary 1. *Every distributional problem that under the standard uniform distribution is in* AvgP *has a frequently self-knowingly correct polynomial-time algorithm.*

Proposition 1. *There exist (distributional) problems with a frequently self-knowingly correct polynomial-time algorithm that are not in* AvgP *under the standard uniform distribution.*

3.3 A Basic Junta Distribution for SAT

Procaccia and Rosenschein [15] introduced "junta distributions" in their study of NP-hard manipulation problems for elections. The goal of a junta is to be such a hard distribution (that is, to focus so much weight on hard instances) that, loosely put, if a problem is easy relative to a junta then it will be easy relative to any reasonable distribution (such as the uniform distribution). This is a goal, not (currently) a theorem; Procaccia and Rosenschein [15] do not formally establish this, but rather seek to give a junta definition that might satisfy this. Their paper in effect encourages others to weigh in and study the suitability of the notion of a junta and the notion built on top of it, heuristic polynomial time. Furthermore, they repeatedly describe their theory as one of average-case complexity. In the full version of this paper [6] we suggest that it is potentially confusion-inducing to describe their theory as one of average-case complexity. Their theory adds to the study of frequency of correctness the notion of probability weight of correctness. This is a very valuable direction, but we point out (see also [17]) that it is neither explicitly about, nor does it seem to implicitly yield claims about, average-case complexity. Their paper states that work of Conitzer and Sandholm [3] is also about average-case complexity but, similarly, we mention that that work is not about average-case complexity; it is about (and carefully and correctly frames itself as being about) frequency of correctness. We do not mean this as a weakness: We feel that frequency of (or probability weight of) correctness, most especially when as in the work of Homan and Hemaspaandra [10] the algorithm is "self-knowingly" correct a guaranteed large portion of the time, is an interesting and important direction.

Regarding Procaccia and Rosenschein's notion of juntas, they state three "basic" conditions for a junta, and then give two additional ones that are tailored specifically to the needs of NP-hard voting manipulation problems. They state their hope that their scheme will extend more generally, using the three basic conditions and potentially additional conditions, to other mechanism problems. One might naturally wonder whether their junta/heuristic polynomial-time/susceptibility approach applies more generally to studying the probability weight of correctness for NP-hard problems, since their framework in effect (aside from the two "additional" junta conditions just about voting manipulation) is a general one relating problems to probability weight of correctness. We first carefully note that in asking this we are taking their notion beyond the realm for which it was explicitly designed, and so we do not claim to be refuting any claim of their paper. What we will do, however, is show that the three basic conditions for a junta are sufficiently weak that one can construct a junta relative to which the standard NP-complete problem SAT—and a similar attack can be carried out on a wide range of natural NP-complete problems—has a deterministic heuristic polynomial-time algorithm. So if one had faith in the analog of their approach, as applied to SAT, one would have to believe that under essentially every natural distribution SAT is easy (in the sense that there is an algorithm with a high probability weight of correctness under that distribution). Since the latter is not widely believed, we suggest that the right conclusion to draw from the main result of this section is simply that if one were to hope to effectively use on typical NP-complete sets the notion of juntas and of heuristic polynomial time w.r.t. juntas, one would almost certainly have to go beyond the basic three conditions and add additional conditions. Again, we stress that Procaccia and Rosenschein didn't focus on examples this far afield, and even within the world of mechanisms implied that unspecified additional conditions beyond the core three might be needed when studying problems other than voting manipulation problems. This section's contribution is to give a construction indicating that the core three junta conditions, standing on their own, seem too weak.

Since we will use the Procaccia–Rosenschein junta notion in a more general setting than merely manipulation problems, we to avoid any chance of confusion will use the term "basic junta" to denote that we have removed the word "manipulation" and that we are using their three "basic" properties, and not the two additional properties that are specific to voting manipulation. Our definition of "deterministic heuristic polynomial-time algorithm" is identical to theirs, except we have replaced the word "junta" with "basic junta"—and so again we are allowing their notion to be extended beyond just manipulation and mechanism problems.

Definition 3 (see [15]). *Let $\mu = \{\mu_n\}_{n\in\mathbb{N}}$ be a distribution over the possible instances of an* NP*-hard problem L. (In this model, each μ_n sums to 1 over all* *length n instances.) We say μ is a* basic junta distribution *if and only if μ has the following properties:*

1. **Hardness:** *The* restriction of L to μ *is the problem whose possible instances are only $\bigcup_{n\in\mathbb{N}}\{x \mid |x| = n$ and $\mu_n(x) > 0\}$. Deciding this restricted problem is still* NP*-hard.*
2. **Balance:** *There exist constants $c > 1$ and $N \in \mathbb{N}$ such that for all $n \geq N$ and for all instances x, $|x| = n$, we have $1/c \leq \mathrm{Prob}_{\mu_n}[x \in L] \leq 1 - 1/c$.*
3. **Dichotomy:** *There exists some polynomial p such that for all n and for all instances x, $|x| = n$, either $\mu_n(x) \geq 2^{-p(n)}$ or $\mu_n(x) = 0$.*

Let (L, μ) be a distributional decision problem (see, e.g., [6, Definition B.1]). An algorithm $\mathcal{A}$ is said to be a deterministic heuristic polynomial-time algorithm for (L, μ) *if $\mathcal{A}$ is a deterministic polynomial-time algorithm and there exist a polynomial q and $N \in \mathbb{N}$ such that for each $n \geq N$, $\mathrm{Prob}_{\mu_n}[x \notin L \iff \mathcal{A}$ accepts $x] < \frac{1}{q(n)}$.*

We now explore their notion of deterministic heuristic polynomial time and their notion of junta, both however viewed for general NP problems and using the "basic" three conditions. We will note that the notion in such a setting is in some senses not restrictive enough and in other senses is too restrictive. Let us start with the former. We need a definition.

Definition 4. *We will say that a set L is* well-pierced *(respectively,* uniquely well-pierced*) if there exist sets Pos $\in$ P and Neg $\in$ P such that Pos $\subseteq L$, Neg $\subseteq \overline{L}$, and there is some $N \in \mathbb{N}$ such that at each length $n \geq N$, each of Pos and Neg has at least one string at length n (respectively, each of Pos and Neg has exactly one string at length n).*

Each uniquely well-pierced set is well-pierced. Note that, under quite natural encodings, such NP-complete sets as, for example, SAT certainly are well-pierced and uniquely well-pierced. (All this says is that, except for a finite number of exceptional lengths, there is one special string at each length that can easily, uniformly be recognized as in the set and one that can easily, uniformly be recognized as not in the set.) Indeed, under quite natural encodings, undecidable problems such as the halting problem are uniquely well-pierced.

Recall that juntas are defined in relation to an infinite list of distributions, one per length (so $\mu = \{\mu_n\}_{n\in\mathbb{N}}$). The Procaccia and Rosenschein definition of junta does not explicitly put computability or uniformity requirements on such distributions in the definition of junta, but it is useful to be able to make claims about that. So let us say that such a distribution is *uniformly computable in polynomial time* (respectively, is *uniformly computable in exponential time*) if there is a polynomial-time function (respectively, an exponential-time function) f such that for each i and each x, $f(i, x)$ outputs the value of $\mu_i(x)$ (say, as a rational number—if a distribution takes on other values, it simply will not be able to satisfy our notion of good uniform time).

Theorem 3. *Let A be any* NP*-hard set that is well-pierced. Then there exists a basic junta distribution relative to which A has a deterministic heuristic polynomial-time algorithm (indeed, it even has a deterministic heuristic polynomial-time algorithm whose error weight is bounded not merely by $1/poly$ as the definition requires, but is even bounded by $1/2^{n^2-n}$). Moreover, the junta is uniformly computable in exponential time, and if we in addition assume that A is uniquely well-pierced, the junta is uniformly computable in polynomial time.*

The proof of Theorem 3, additional results, and extensive related discussions on the junta approach can be found in the full version of this paper [6].

4 Conclusions

Christian et al. [2] introduced the optimal lobbying problem and showed it complete for W[2], and so generally viewed as intractable in the sense of parameterized complexity. In Section 2, we proposed an efficient greedy algorithm for approximating the optimal solution of this problem, even if generalized by assigning prices to voters. This greedy algorithm achieves a logarithmic approximation ratio and we prove that that is essentially the best approximation ratio that can be proven for this algorithm. We mention as an interesting open issue whether more elaborate algorithms can achieve better approximation ratios.

Section 3 studied relationships between average-case polynomial time, benign algorithm schemes, and frequency (and probability weight) of correctness. We showed that all problems having benign algorithm schemes relative to the uniform distribution (and thus all sets in average-case polynomial time relative to the uniform distribution) have frequently self-knowingly correct algorithms. We also studied, when limited to the "basic" three junta conditions, the notion of junta distributions and of deterministic heuristic polynomial time, and we showed that they admit some extreme behaviors. We argued that deterministic heuristic polynomial time should not be viewed as a model of average-case complexity.

Acknowledgments: We are deeply grateful to Chris Homan for his interest in this work and for many inspiring discussions on computational issues related to voting. We also thank the anonymous FCT 2007 and COMSOC 2006 workshop referees for their helpful comments.

References

1. 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.
2. R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. In U. Endriss and J. Lang, editors, *First International Workshop on Computational Social Choice (COMSOC 2006)*, pages 87–96 (workshop notes). Universiteit van Amsterdam, December 2006.

3. V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, July 2006.
4. R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.
5. C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, 1876.
6. G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. Technical Report TR-914, Department of Computer Science, University of Rochester, Rochester, NY, March 2007.
7. J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, 2006.
8. P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.
9. O. Goldreich. Note on Levin's theory of average-case complexity. Technical Report TR97-058, Electronic Colloquium on Computational Complexity, November 1997.
10. 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.
11. 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.
12. 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.
13. L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.
14. J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson's rule. In U. Endriss and J. Lang, editors, *First International Workshop on Computational Social Choice (COMSOC 2006)*, pages 331–344 (workshop notes). Universiteit van Amsterdam, December 2006.
15. A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.
16. 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.
17. L. Trevisan. Lecture notes on computational complexity. www.cs.berkeley.edu/~luca/notes/complexitynotes02.pdf (Lecture 12), 2002.
18. J. Wang. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 295–328. Springer-Verlag, 1997.