Exact Analysis of Dodgson Elections: Lewis Carroll's 1876 Voting System Is Complete for Parallel Access to NP

EDITH HEMASPAANDRA

Le Moyne College, Syracuse, New York

LANE A. HEMASPAANDRA

University of Rochester, Rochester, New York

AND

JÖRG ROTHE

Friedrich-Schiller-Universität Jena, Jena, Germany

Abstract. In 1876, Lewis Carroll proposed a voting system in which the winner is the candidate who with the fewest changes in voters' preferences becomes a Condorcet winner—a candidate who beats all other candidates in pairwise majority-rule elections. Bartholdi, Tovey, and Trick provided a lower bound—NP-hardness—on the computational complexity of determining the election winner in Carroll's system. We provide a stronger lower bound and an upper bound that matches our lower bound. In particular, determining the winner in Carroll's system is complete for parallel access to NP, that is, it is complete for Θ_2^p, for which it becomes the most natural complete problem known. It

---

This work was done in part while E. Hemaspaandra and L. A. Hemaspaandra were visiting Friedrich-Schiller-Universität Jena and the University of Amsterdam, and while J. Rothe was visiting Le Moyne College and the University of Rochester.

An extended abstract of this paper appeared in *Proceedings of the 24th International Colloquium on Automata*, *Languages*, *and Programming.*

E. Hemaspaandra was supported in part by grant NSF-INT-9513368/DAAD-315-PRO-fo-ab. L. A. Hemaspaandra was supported in part by grants NSF-CCR-9322513 and NSF-INT-9513368/DAAD-315-PRO-fo-ab, and a University of Rochester Bridging Fellowship. J. Rothe was supported in part by grant NSF-INT-9513368/DAAD-315-PRO-fo-ab and a NATO Postdoctoral Science Fellowship from the Deutscher Akademischer Austauschdienst ("Gemeinsames Hochschulsonderprogramm III von Bund und Ländern").

Address: Edith Hemaspaandra, Department of Mathematics, Le Moyne College, Syracuse, NY 13214, USA. Email: edith@bamboo.lemoyne.edu; Lane A. Hemaspaandra, Department of Computer Science, University of Rochester, Rochester, NY 14627, USA. Email: lane@cs.rochester.edu; Jörg Rothe, Institut für Informatik, Friedrich-Schiller-Universität Jena, 07740 Jena, Germany. Email: rothe@informatik.uni-jena.de.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery (ACM), Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
© 1997 ACM 0004-5411/97/1100-0806 $05.00

follows that determining the winner in Carroll's elections is not NP-complete unless the polynomial hierarchy collapses.

Categories and Subject Descriptors: F.1.3 [**Computation by Abstract Devices**]: Complexity Classes; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems; J.4. [**Social and Behavioral Sciences**]

General Terms: Theory

Additional Key Words and Phrases: Completeness, election systems, Lewis Carroll, majority rule

1. *Introduction*

The Condorcet criterion is that an election is won by any candidate who defeats all others in pairwise majority-rule elections [Condorcet 1785], see Black [1958]. The Condorcet Paradox, dating from 1785 [Condorcet 1785], notes that not only is it not always the case that Condorcet winners exist but, far worse, when there are more than two candidates, pairwise majority-rule elections may yield strict cycles in the aggregate preference even if each voter has non-cyclic preferences.[1] This is a widely discussed and troubling feature of majority rule (see, e.g., the discussion in Mueller [1989]).

In 1876, Charles Lutwidge Dodgson—more commonly referred to today by his pen name, Lewis Carroll—proposed an election system that is inspired by the Condorcet criterion,[2] yet that sidesteps the above-mentioned problem [Dodgson 1876]. In particular, a Condorcet winner is a candidate who defeats each other candidate in pairwise majority-rule elections. In Carroll's system, an election is won by the candidate who is "closest" to being a Condorcet winner. In particular, each candidate is given a score that is the smallest number of exchanges of adjacent preferences in the voters' preference orders needed to make the candidate a Condorcet winner with respect to the resulting preference orders. Whatever candidate (or candidates, in the case of a tie) has the lowest score is the winner. This system admits ties but, as each candidate is assigned an integer score, no strict-preference cycles are possible.

Bartholdi et al. [1989] in their paper "Voting Schemes for which It Can Be Difficult to Tell Who Won the Election" raise a difficulty regarding Carroll's election system. Though the notion of winner(s) in Carroll's election system is mathematically well defined, Bartholdi et al. raise the issue of what the *computational complexity* is of determining who is the winner. Though most natural election schemes admit obvious polynomial-time algorithms for determining who won, in sharp contrast Bartholdi et al. prove that Carroll's election scheme has the disturbing property that it is NP-hard to determine whether a given candidate has won a given election (a problem they dub `DodgsonWinner`), and that it is NP-hard even to determine whether a given candidate has tied-or-defeated another given candidate (a problem they dub `DodgsonRanking`).

[1] The standard example is an election over candidates a, b, and c in which 1/3 of the voters have preference $\langle a < b < c \rangle$, 1/3 of the voters have preference $\langle b < c < a \rangle$, and 1/3 of the voters have preference $\langle c < a < b \rangle$. In this case, though each voter individually has well-ordered preferences, the aggregate preference of the electorate is that b trounces a, c trounces b, and a trounces c. In short, individually well-ordered preferences do not necessarily aggregate to a well-ordered societal preference.

[2] Carroll did not use this term. Indeed, Black has shown that Carroll "almost beyond a doubt" was unfamiliar with Condorcet's work [Black 1958, pp. 193–194].

Bartholdi et al.'s NP-hardness results establish lower bounds for the complexity of `DodgsonRanking` and `DodgsonWinner`. *We optimally improve their two complexity lower bounds by proving that both problems are hard for Θ_2^p, the class of problems that can be solved via parallel access to NP*, *and we provide matching upper bounds. Thus*, *we establish that both problems are Θ_2^p-complete*. Bartholdi et al. explicitly leave open the issue of whether `DodgsonRanking` is NP-complete: ". . . Thus `DodgsonRanking` is as hard as an NP-complete problem, but since we do not know whether `DodgsonRanking` is in NP, we can say only that it is NP-hard" [Bartholdi et al. 1989, p. 161]. From our optimal lower bounds, it follows that neither `DodgsonWinner` nor `DodgsonRanking` is NP-complete unless the polynomial hierarchy collapses.

As to our proof method, in order to raise the known lower bound on the complexity of Dodgson elections, we first study the ways in which feasible algorithms can control Dodgson elections. In particular, we prove a series of lemmas showing how polynomial-time algorithms can control oddness and evenness of election scores, "sum" over election scores, and merge elections. These lemmas then lead to our hardness results.

We remark that it is somewhat curious finding "parallel access to NP"-complete (i.e., Θ_2^p-complete) problems that were introduced almost one hundred years before complexity theory itself existed. In addition, `DodgsonWinner`, which we prove complete for this class, is extremely natural when compared with previously known complete problems for this class, essentially all of which have somewhat convoluted forms, for example, asking whether a given list of Boolean formulas has the property that the number of formulas in the list that are satisfiable is itself an odd number. In contrast, the class NP, which is contained in Θ_2^p, has countless natural complete problems. Also, we mention that Papadimitriou [1984] has shown that `UniqueOptimalTravelingSalesperson` is complete for P^{NP}, which contains Θ_2^p.

2. *Preliminaries*

In this section, we introduce some standard concepts and notations from computational complexity theory [Papadimitriou 1994; Bovet and Crescenzi 1993; Garey and Johnson 1979]. NP is the class of languages solvable in nondeterministic polynomial time. The polynomial hierarchy [Meyer and Stockmeyer 1972; Stockmeyer 1977], PH, is defined as $\mathrm{PH} = \mathrm{P} \cup \mathrm{NP} \cup \mathrm{NP}^{\mathrm{NP}} \cup \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}}} \cup \cdots$ where, for any class $\mathscr{C}$, $\mathrm{NP}^{\mathscr{C}} = \bigcup_{C \in \mathscr{C}} \mathrm{NP}^C$, and NP^C is the class of all languages that can be accepted by some NP machine that is given a black box that in unit time answers membership queries to C. The polynomial hierarchy is said to collapse if for some k the kth term in the preceding infinite union equals the entire infinite union. Computer scientists strongly suspect that the polynomial hierarchy does not collapse, though proving (or disproving) this remains a major open research issue.

The polynomial hierarchy has a number of intermediate levels. The Θ_2^p level of the polynomial hierarchy will be of particular interest to us. Θ_2^p, which was first studied by Papadimitriou and Zachos [1983] (see also Wagner [1990]), is the class of all languages that can be solved via $\mathbb{O}(\log n)$ queries to some NP set. Equivalently, and more to the point for the purposes of this paper, Θ_2^p equals the class of problems that can be solved via parallel access to NP [Hemachandra

1989; Köbler et al. 1987], as explained formally later in this section. Θ_2^p falls between the first two levels of the polynomial hierarchy: $\mathrm{NP} \subseteq \Theta_2^p \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \mathrm{NP}^{\mathrm{NP}}$. During the past decade, Θ_2^p has played a quite active role in complexity theory. Kadin [1989] has proven that if NP has a sparse Turing-complete set then the polynomial hierarchy collapses to Θ_2^p, Hemachandra and Wechsung have shown that the question of whether Θ_2^p and *sequential* access to NP yield the same class can be characterized in terms of Kolmogorov complexity [Hemachandra and Wechsung 1991], Wagner [1990] has shown that the definition of Θ_2^p is extremely robust, and Jenner and Torán [1995] have shown that the robustness of the class Θ_2^p seems to fail for its function analogs.

Problems are encoded as languages of strings over some fixed alphabet Σ having at least two letters. Σ^* denotes the set of all strings over Σ. For any string $x \in \Sigma^*$, let $|x|$ denote the length of x. For any set $A \subseteq \Sigma^*$, let $\bar{A}$ denote $\Sigma^* \setminus A$. For any set $A \subseteq \Sigma^*$, let $\|A\|$ denote the cardinality of A. For any multiset A, $\|A\|$ will denote the cardinality of A. For example, if A is the multiset containing one occurrence of the preference order $\langle w < x < y \rangle$ and seventeen occurrences of the preference order $\langle w < y < x \rangle$, then $\|A\| = 18$. As is standard, for each language $A \subseteq \Sigma^*$ we use χ_A to denote the characteristic function of A, that is, $\chi_A(x) = 1$ if $x \in A$ and $\chi_A(x) = 0$ if $x \notin A$. Let $\langle \cdots \rangle$ be any standard, multi-arity, easily computable, easily invertible pairing function. We will also use the notation $\langle \cdots \rangle$ to denote preference orders, for example, $\langle w < x < y \rangle$. Which use is intended will be clear from context. Whenever we speak of a function that takes a *variable* number of arguments, we will assume that the arguments, say $a_1, \ldots, a_z$, are encoded as $a_1\# \cdots \#a_z\#$, where # is a symbol not in the alphabet in which the arguments are encoded. When speaking of a variable-arity function being polynomial-time computable, we mean that the function's running time is polynomial in $|a_1\# \cdots \#a_z\#| = z + |a_1| + \cdots + |a_z|$.

In computational complexity theory, reductions are used to relate the complexity of problems. Very informally, if A reduces to B that means that, given B, one can solve A. For any a and b such that $\leq_a^b$ is a defined reduction type, and any complexity class $\mathscr{C}$, let $\mathrm{R}_a^b(\mathscr{C})$ denote $\{L \mid (\exists C \in \mathscr{C})[L \leq_a^b C]\}$. We refer readers to the standard source, Ladner, Lynch, and Selman [1975], for definitions and discussion of the standard reductions. However, we briefly and informally present to the reader the definitions of the reductions to be used in this paper. $A \leq_m^p B$ ("A polynomial-time many-one reduces to B") if there is a polynomial-time computable function f such that $(\forall x \in \Sigma^*)$ $[x \in A \Leftrightarrow f(x) \in B]$. $A \leq_{tt}^p B$ ("A polynomial-time truth-table reduces to B") if there is a polynomial-time Turing machine that, on input x, computes a query that itself consists of a list of strings and, given that the machine after writing the query is then given as its answer a list telling which of the listed strings are in B, the machine then correctly determines whether x is in A (this is not the original Ladner-Lynch-Selman definition, as we have merged their querying machine and their evaluation machine; however, this formulation is common and equivalent). Since a $\leq_{tt}^p$-reducing machine, on a given input, asks all its questions in a *parallel* (also called *nonadaptive*) manner, the informal statement above that Θ_2^p captures the complexity of "parallel access to NP" can now be expressed formally as the claim $\Theta_2^p = \mathrm{R}_{tt}^p(\mathrm{NP})$, which is known to hold [Hemachandra 1989; Köbler et al. 1987].

As has become the norm, we always use hardness to denote hardness with respect to $\leq_m^p$ reductions. That is, for any class $\mathscr{C}$ and any problem A, we say that

A is $\mathscr{C}$-*hard* if $(\forall C \in \mathscr{C})\ [C \leq_m^p A]$. For any class $\mathscr{C}$ and any problem A, we say that A is $\mathscr{C}$-*complete* if A is $\mathscr{C}$-hard and $A \in \mathscr{C}$. Completeness results are the standard method in computational complexity theory of categorizing the complexity of a problem, as a $\mathscr{C}$-complete problem A is both in $\mathscr{C}$, and is the hardest problem in $\mathscr{C}$ (in the sense that every problem in $\mathscr{C}$ can be easily solved using A).

3. *The Complexity of Dodgson Elections*

Lewis Carroll's voting system [Dodgson 1876] (see also Niemi and Riker [1976] and Bartholdi et al. [1989]) works as follows: Each voter has strict preferences over the candidates. Each candidate is assigned a score, namely, the smallest number of sequential *exchanges of two adjacent candidates in the voters' preference orders* (henceforth called *switches*) needed to make the given candidate a Condorcet winner. We say that a candidate c *ties-or-defeats* a candidate d if the score of d is not less than that of c. (Bartholdi et al. [1989] use the term "defeats" to denote what we, for clarity, denote by ties-or-defeats; though the notations are different, the sets being defined by Bartholdi et al. and in this paper are identical.) A candidate c is said to win the Dodgson-type election if c ties-or-defeats all other candidates. Of course, due to ties it is possible for two candidates to tie-or-defeat each other, and so it is possible for more than one candidate to be a winner of the election.

Recall that all preferences are assumed to be strict. A candidate c is a *Condorcet winner* (with respect to a given collection of voter preferences) if c defeats (i.e., is preferred by strictly more than half of the voters) each other candidate in pairwise majority-rule elections. Of course, Condorcet winners do not necessarily exist for a given set of preferences, but if a Condorcet winner does exist, it is unique.

We now return to Carroll's scoring notion to clarify what is meant by the sequential nature of the switches, and to clarify by example that one switch changes only one voter's preferences. The (*Dodgson*) *score* of any Condorcet winner is 0. If a candidate is not a Condorcet winner, but one switch (recall that a switch is an exchange of two *adjacent* candidates in the preference order of *one* voter) would make the candidate a Condorcet winner, then the candidate has a score of 1. If a candidate does not have a score of 0 or 1, but two switches would make the candidate a Condorcet winner, then the candidate has a score of 2. Note that the two switches could both be in the same voter's preferences, or could be one in one voter's preferences and one in another voter's preferences. Note also that switches are sequential. For example, with two switches, one could change a single voter's preferences from $\langle a < b < c < d\rangle$ to $\langle c < a < b < d\rangle$, where $e < f$ will denote the preference: "f is strictly preferred to e." With two switches, one could also change a single voter's preferences from $\langle a < b < c < d\rangle$ to $\langle b < a < d < c\rangle$. With two switches (not one), one could also change two voters with initial preferences of $\langle a < b < c < d\rangle$ and $\langle a < b < c < d\rangle$ to the new preferences $\langle b < a < c < d\rangle$ and $\langle b < a < c < d\rangle$. As noted earlier in this section, Dodgson scores of 3, 4, etc., are defined analogously, that is, the Dodgson score of a candidate is the smallest number of sequential switches needed to make the given candidate a Condorcet winner. (We note in passing that Dodgson was before his time in more ways than one. His definition is closely related to an important concept that is now known in computer science as "edit-distance"—the minimum number of operations (from some specified set of operations) required to transform one string into another. Though Carroll's single "switch" operation is not the richer set of operations most commonly used today when doing string-to-string editing (see, e.g., Sankoff and Kruskal [1983]), it does form a valid basis operation for transforming between permutations, which after all are what preferences are.)

Bartholdi et al. [1989] define a number of decision problems related to Carroll's system. They prove that given preference lists, and a candidate, and a number k, it is NP-complete to determine whether the candidate's score is at most k in the election specified by the preference lists (they call this problem `DodgsonScore`). They define the problem `DodgsonRanking` to be the problem of determining, given preference lists and the names of two voters, c and d, whether c ties-or-defeats d. They prove that this problem is NP-hard. They also prove that, given a candidate and preference lists, it is NP-hard to determine whether the candidate is a winner of the election.

For the formal definitions of these three decision problems, a preference order is strict (i.e., irreflexive and antisymmetric), transitive, and complete. Since we will freely identify voters with their preference orders, and two different voters can have the same preference order, we define a set of voters as a multiset of preference orders.

We will say that $\langle C, c, V\rangle$ is a *Dodgson triple* if C is a set of candidates, c is a member of C, and V is a multiset of preference orders on C. Throughout this paper, we assume that, as inputs, multisets are coded as lists, that is, if there are m voters in the voter set then $V = \langle P_1, P_2, \ldots, P_m\rangle$, where P_i is the preference order of the ith voter. *Score*$(\langle C, c, V\rangle)$ will denote the Dodgson score of c in the vote specified by C and V. If X is a decision problem, then when we speak of an *instance* of X we mean a string that satisfies the syntactic conditions listed in the "Instance" field of the problem's definition (or implicit in that field in order for the problem to be syntactically well formed—for example, preference lists must be over the right number and right set of candidates). As is standard, since all such syntactic conditions in our decision problems are trivially checkable in deterministic polynomial time, this is equivalent to the language definitions that are also common; in particular, the language corresponding to decision problem X is the set $\{x | x$ is an instance of X, and the "Question" of decision problem X has the answer "yes" for $x\}$. Since reductions map between sets, whenever speaking of or constructing reductions we use this latter formalism.

Decision Problem: `DodgsonScore`

Instance: A Dodgson triple $\langle C, c, V\rangle$; a positive integer k.

Question: Is *Score*$(\langle C, c, V\rangle)$, the Dodgson score of candidate c in the election specified by $\langle C, V\rangle$, less than or equal to k?

Decision Problem: `DodgsonRanking`

Instance: A set of candidates C; two distinguished members of C, c and d; a multiset V of preference orders on C (encoded as a list, as discussed above).

Question: Does c tie-or-defeat d in the election? That is, is *Score*$(\langle C, c, V\rangle) \leq$ *Score*$(\langle C, d, V\rangle)$?

Decision Problem: `DodgsonWinner`

Instance: A Dodgson triple $\langle C, c, V\rangle$.

Question: Is c a winner of the election? That is, does c tie-or-defeat all other candidates in the election?

We now state the complexity of `DodgsonRanking`.

THEOREM 3.1. `DodgsonRanking` *is* Θ_2^p*-complete*.

It follows immediately—since (a) $\Theta_2^p = \mathrm{NP} \Rightarrow \mathrm{PH} = \mathrm{NP}$, and (b) $\mathrm{R}_m^p(\mathrm{NP}) = \mathrm{NP}$—that `DodgsonRanking`, though known to be NP-hard [Bartholdi et al. 1989], cannot be NP-complete unless the polynomial hierarchy collapses quite dramatically.

COROLLARY 3.2. *If* `DodgsonRanking` *is NP-complete, then PH = NP.*

Most of the rest of the paper is devoted to working toward a proof of Theorem 3.1. Wagner has provided a useful tool for proving Θ_2^p-hardness, and we state his result below as Lemma 3.3. However, to be able to exploit this tool we must explore the structure of Dodgson elections. In particular, we have to learn how to control oddness and evenness of election scores, how to add election scores, and how to merge elections. We do so as Lemmas 3.4, 3.5, and 3.7, respectively. On our way toward a proof of Theorem 3.1, using Lemmas 3.3, 3.4, and 3.5 we will first establish Θ_2^p-hardness of a special problem that is closely related to `DodgsonRanking`. This result is stated as Lemma 3.6 below. It is not hard to prove Theorem 3.1 using Lemma 3.6 and Lemma 3.7. Note that Lemma 3.7 gives more than is needed merely to establish Theorem 3.1. In fact, the way this lemma is stated even suffices to provide—jointly with Lemma 3.6—a direct proof of the Θ_2^p-hardness of `DodgsonWinner`.

LEMMA 3.3 [WAGNER 1987]. *Let A be some NP-complete set, and let B be any set. If there exists a polynomial-time computable function g such that, for all $k \geq 1$ and all strings $x_1, \ldots, x_{2k} \in \Sigma^*$ satisfying $\chi_A(x_1) \geq \chi_A(x_2) \geq \cdots \geq \chi_A(x_{2k})$, it holds that*

$$\|\{i | x_i \in A\}\| \text{ is odd} \Leftrightarrow g(x_1, \ldots, x_{2k}) \in B,$$

then B is Θ_2^p-hard.[3]

LEMMA 3.4. *There exists an NP-complete set A and a polynomial-time computable function f that reduces A to* `DodgsonScore` *in such a way that, for every $x \in \Sigma^*$, $f(x) = \langle\langle C, c, V\rangle, k\rangle$ is an instance of* `DodgsonScore` *with an odd number of voters and*

(1) *if $x \in A$, then $Score(\langle C, c, V\rangle) = k$, and*

[3] Recall the comments/conventions of Section 2 regarding the handling of the arguments of variable-arity functions. Wagner did not discuss this issue, but we note that his proof remains valid under the conventions of Section 2. These conventions have been adopted as they shield Wagner's theorem from a pathological type of counterexample (involving large, variable numbers of length zero inputs (ϵ) followed by one other constant-length string) noted by a referee that, without the conventions, could render Wagner's theorem true but never applicable.

Another difference in our statement of the theorem relative to Wagner's is that though we state the theorem for the class Θ_2^p, Wagner used the class "$\mathrm{P}_{\mathrm{bf}}^{\mathrm{NP}}$." However, this is legal as $\mathrm{P}_{\mathrm{bf}}^{\mathrm{NP}}$ is now known to be equal to Θ_2^p (see the discussion in Köbler et al. [1987, Footnote 1]).

(2) *if* $x \notin A$, *then* $Score(\langle C, c, V\rangle) = k + 1$.

LEMMA 3.5. *There exists a polynomial-time computable function DodgsonSum such that, for all k and for all* $\langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle$ *satisfying* $(\forall j)[\|V_j\|$ *is odd*$]$, *it holds that*

$$DodgsonSum(\langle\langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle\rangle)$$

is a Dodgson triple having an odd number of voters and such that

$$\sum_j Score(\langle C_j, c_j, V_j\rangle) =$$

$$Score(DodgsonSum(\langle\langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle\rangle)).$$

Lemmas 3.3, 3.4, and 3.5 together establish the Θ_2^p-hardness of a special problem that is closely related to the problems that we are interested in, `DodgsonRanking` and `DodgsonWinner`. Let us define the decision problem `TwoElectionRanking` (`2ER`).

Decision Problem: `TwoElectionRanking` (`2ER`)
Instance: A pair of Dodgson triples $\langle\langle C, c, V\rangle, \langle D, d, W\rangle\rangle$ both having an odd number of voters and such that $c \neq d$.
Question: Is $Score(\langle C, c, V\rangle) \leq Score(\langle D, d, W\rangle)$?

LEMMA 3.6. `TwoElectionRanking` *is* Θ_2^p*-hard.*

We note in passing that `2ER` is in $R_{tt}^p(\mathrm{NP})$. This fact follows by essentially the same argument that will be used in the proof of Theorem 3.1 to establish that theorem's upper bound. Thus, since $\Theta_2^p = R_{tt}^p(\mathrm{NP})$, we have—in light of Lemma 3.6—that `2ER` is Θ_2^p-complete. We also note in passing that, since one can trivially rename candidates, `2ER` remains Θ_2^p-complete in the variant in which "and such that $c \neq d$" is removed from the problem's definition.

In order to make the results obtained so far applicable to `DodgsonRanking` and `DodgsonWinner`, we need the following lemma that tells us how to merge two elections into a single election in a controlled manner.

LEMMA 3.7. *There exist polynomial-time computable functions Merge and Merge′ such that, for all Dodgson triples* $\langle C, c, V\rangle$ *and* $\langle D, d, W\rangle$ *for which* $c \neq d$ *and both V and W represent odd numbers of voters, there exist* $\hat{C}$ *and* $\hat{V}$ *such that*

(*i*) $Merge(\langle C, c, V\rangle, \langle D, d, W\rangle)$ *is an instance of* `DodgsonRanking` *and* $Merge'(\langle C, c, V\rangle, \langle D, d, W\rangle)$ *is an instance of* `DodgsonWinner`,
(*ii*) $Merge(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle \hat{C}, c, d, \hat{V}\rangle$ *and* $Merge'(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle \hat{C}, c, \hat{V}\rangle$,
(*iii*) $Score(\langle \hat{C}, c, \hat{V}\rangle) = Score(\langle C, c, V\rangle) + 1$,
(*iv*) $Score(\langle \hat{C}, d, \hat{V}\rangle) = Score(\langle D, d, W\rangle) + 1$, *and*
(*v*) *for each* $e \in \hat{C} \setminus \{c, d\}$, $Score(\langle \hat{C}, c, \hat{V}\rangle) < Score(\langle \hat{C}, e, \hat{V}\rangle)$.

We now prove these lemmas.

PROOF OF LEMMA 3.4. Bartholdi et al. [1989] prove the NP-hardness of `DodgsonScore` by reducing `ExactCoverByThreeSets` to it. However, their reduction does not have the additional properties that we need in this lemma.

We will construct a reduction from the NP-complete problem `ThreeDimensionalMatching` (`3DM`) [Garey and Johnson 1979] to `DodgsonScore` that *does* have the additional properties we need. Let us first give the definition of `3DM`:

Decision Problem: `ThreeDimensionalMatching` (`3DM`)

Instance: Sets M, W, X, and Y, where $M \subseteq W \times X \times Y$ and W, X, and Y are disjoint, nonempty sets having the same number of elements.

Question: Does M contain a *matching*, that is, a subset $M' \subseteq M$ such that $\|M'\| = \|W\|$ and no two elements of M' agree in any coordinate?

We now describe a polynomial-time reduction f (from `3DM` to `DodgsonScore`) having the desired properties. Our reduction is defined by $f(x) = f'(f''(x))$, where f' and f'' are as described below. Informally, f'' turns all inputs into a standard format (instances of `3DM` having $\|M\| > 1$), and f' assumes its input has this format and implements the actual reduction.

Let f'' be a polynomial-time function that has the following properties.

(1) If x is not an instance of `3DM` or is an instance of `3DM` having $\|M\| \leq 1$, then $f''(x)$ will output an instance y of `3DM` for which $\|M\| > 1$ and, furthermore, it will hold that $y \in$ `3DM` $\Leftrightarrow x \in$ `3DM`.

(2) If x is an instance of `3DM` having $\|M\| > 1$, then $f''(x) = x$.

It is clear that such functions exist. In particular, for concreteness, let $f''(x)$ be $\langle\{(d, e, p), (d, e, p')\}, \{d, d'\}, \{e, e'\}, \{p, p'\}\rangle$ if x is not an instance of `3DM` or both $x \notin$ `3DM` and x is an instance of `3DM` having $\|M\| \leq 1$; let $f''(x)$ be $\langle\{(d, e, p), (d', e', p')\}, \{d, d'\}, \{e, e'\}, \{p, p'\}\rangle$ if x is an instance of `3DM` having $\|M\| \leq 1$ and such that $x \in$ `3DM`; let $f''(x)$ be x, otherwise.

We now describe f'. Let x be our input. If x is not an instance of `3DM` for which $\|M\| > 1$, then $f'(x) = 0$; this is just for definiteness, as due to f'', the only actions of f' that matter are when the input is an instance of `3DM` for which $\|M\| > 1$. So, suppose $x = \langle M, W, X, Y\rangle$ is an instance of `3DM` for which $\|M\| > 1$. Let $q = \|W\|$. Define $f'(\langle M, W, X, Y\rangle) = \langle\langle C, c, V\rangle, 3q\rangle$ as follows: Let c, s, and t be elements not in $W \cup X \cup Y$. Let $C = W \cup X \cup Y \cup \{c, s, t\}$ and let V consist of the following two subparts:

(1) Voters simulating elements of M. Suppose the elements of M are enumerated as $\{(w_i, x_i, y_i) | 1 \leq i \leq \|M\|\}$. (The w_i are not intended to be an enumeration of W. Rather, they take on values from W as specified by M. In particular, w_j may equal w_k even if $j \neq k$. The analogous comments apply to the x_i and y_i variables.) For every triple (w_i, x_i, y_i) in M, we will create a voter. If i is odd, we create the voter $\langle s < c < w_i < x_i < y_i < t < \cdots\rangle$, where the elements after t are the elements of $C \setminus \{s, c, w_i, x_i, y_i, t\}$ in arbitrary order. If i is even, we do the same, except that we exchange s and t. That is, we create the voter $\langle t < c < w_i < x_i < y_i < s < \cdots\rangle$, where the elements after s are the elements of $C \setminus \{s, c, w_i, x_i, y_i, t\}$ in arbitrary order.

(2) $\|M\| - 1$ voters who prefer c to all other candidates.

We will now show that f has the desired properties. It is immediately clear that f'' and f', and thus f, are polynomial-time computable. It is also clear from our construction that, for each x, $f(x)$ is an instance of `DodgsonScore` having an odd number of voters since, for every instance $\langle M, W, X, Y\rangle$ of `3DM` with $\|M\| > 1$, $f'(\langle M, W, X, Y\rangle)$ is an instance of `DodgsonScore` with $\|M\| + (\|M\| - 1)$ voters, and since f'' always outputs instances of this form. It remains to show that, for every instance $\langle M, W, X, Y\rangle$ of `3DM` with $\|M\| > 1$:

(a) if M contains a matching, then $Score(\langle C, c, V\rangle) = 3q$, and

(b) if M does not contain a matching, then $Score(\langle C, c, V\rangle) = 3q + 1$.

Note that if we prove this, it is clear that f has the properties (1) and (2) of Lemma 3.4, in light of the properties of f''. Note that, recalling that we may now assume that $\|M\| > 1$, by construction c is preferred to s and t by more than half of the voters, and is preferred to all other candidates by $\|M\| - 1$ of the $2\|M\| - 1$ voters.

Now suppose that M contains a matching M'. Then $\|M'\| = q$, and every element in $W \cup X \cup Y$ occurs in M'. $3q$ switches turn c into a Condorcet winner as follows. For every element $(w_i, x_i, y_i) \in M'$, switch c upwards 3 times in the voter corresponding to (w_i, x_i, y_i). For example, if i is odd, this voter changes from $\langle s < c < w_i < x_i < y_i < t < \cdots\rangle$ to $\langle s < w_i < x_i < y_i < c < t < \cdots\rangle$. Let z be an arbitrary element of $W \cup X \cup Y$. Since z occurs in M', c has gained one vote over z. Thus, c is preferred to z by $\|M\|$ of the $2\|M\| - 1$ voters. Since z was arbitrary, c is a Condorcet winner.

On the other hand, c's Dodgson score can never be less than $3q$, because to turn c into a Condorcet winner, c needs to gain one vote over z for every $z \in W \cup X \cup Y$. Since c can gain only *one* vote over *one* candidate for each switch, we need at least $3q$ switches to turn c into a Condorcet winner. This proves condition (a).

To prove condition (b), first note that there is a "trivial" way to turn c into a Condorcet winner with $3q + 1$ switches: Just switch c to the top of the preference order of the first voter. The first voter was of the form $\langle s < c < w_1 < x_1 < y_1 < t < \cdots\rangle$, where the elements after t are exactly all elements in $(W \cup X \cup Y) \setminus \{w_1, x_1, y_1\}$, in arbitrary order. Switching c upwards $3q + 1$ times moves c to the top of the preference order for this voter, and gains one vote for c over all candidates in $W \cup X \cup Y$, which turns c into a Condorcet winner. This shows that $Score(\langle C, c, V\rangle) \leq 3q + 1$, regardless of whether M has a matching or not.

Finally, note that a Dodgson score of $3q$ implies that M has a matching. As before, every switch has to involve c and an element of $W \cup X \cup Y$. (This is because c must gain a vote over $3q$ other candidates—$W \cup X \cup Y$—and so any switch involving s or t would ensure that at most $3q - 1$ switches were available for gaining against the $3q$ members of $W \cup X \cup Y$, thus ensuring failure.) Thus, for every voter, c switches at most three times to become a Condorcet winner. Since c has to gain one vote in particular over each element in Y, and to "reach" an element in Y it must hold that c first switches over the elements of W and X that due to our construction fall between it and the nearest y element (among the $\|M\|$ voters simulating elements of M—it is clear that if any switch involves at least one of the $\|M\| - 1$ dummy voters this could never lead to a Dodgson score of $3q$ for c), it must be the case that c switches upwards exactly three times for exactly q voters corresponding to elements of M. This implies that the q elements of M that correspond to these q voters form a matching, thus proving condition (b). $\square$

PROOF OF LEMMA 3.5. This is trivial if $k = 1$ so we henceforth assume that $k > 1$. We define

$$DodgsonSum(\langle\langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle\rangle) = \langle \hat{C}, c, \hat{V}\rangle,$$

where $\hat{C}$, c, and $\hat{V}$ will be as constructed in this proof.

Let $c = c_1$. Without loss of generality (by renaming if needed), we assume that $c_1 = c_2 = \cdots = c_k$, and that $(\forall i, j)\ [i \neq j \Rightarrow C_i \cap C_j = \{c\}]$.

Also, for each i, enumerate $C_i \backslash \{c\}$ as $\{c_{i,1}, c_{i,2}, \ldots, c_{i,\|C_i\|-1}\}$. To make our preference orders easier to read, whenever in a preference order we write in the text "$\overrightarrow{C_i}$," this should be viewed as being replaced by the text string "$c_{i,1} < c_{i,2} < \cdots < c_{i,\|C_i\|-1}$."

As our candidate set, we will take all the old candidates from the given elections, that is,

$$\{c, c_{1,1}, c_{1,2}, \ldots, c_{1,\|C_1\|-1}, c_{2,1}, c_{2,2}, \ldots, c_{2,\|C_2\|-1}, \cdots, c_{k,1}, c_{k,2}, \ldots, c_{k,\|C_k\|-1}\},$$

plus a set S of new "separator" candidates, whose only purpose is to avoid interference. We will ensure that c is preferred to all elements of S by a majority of the voters.

Formally, let $S = \{s_i | 1 \leq i \leq \Sigma_j \|C_j\| \cdot \|V_j\|\}$, and let $\hat{C} = S \cup \bigcup_j C_j$. As a notational convenience, whenever in a preference order we write in the text "$\overrightarrow{S}$," this should be viewed as being replaced by the string "$s_1 < s_2 < \cdots < s_{\|S\|}$." The voter set $\hat{V}$ consists of two subparts—voters simulating voters from the underlying elections, and voters who are "normalizing" voters. The total number of voters will be $(2\ \Sigma_j \|V_j\|) - 1$, which is odd as required by the statement of the lemma being proven. We now describe the simulating voters (the cases of 1 and k are exactly analogous to the other cases, but are stated separately just for notational reasons):

—There will be voters simulating the voters of V_1. In particular, for each voter $\langle e_1 < e_2 < \cdots < e_{\|C_1\|}\rangle$ in V_1, we create a voter

$$\langle\ \overrightarrow{S}\ < \overrightarrow{C_2} < \cdots < \overrightarrow{C_k} < e_1 < e_2 < \cdots < e_{\|C_1\|}\rangle.$$

Note that c is one of the e_j's.

—For each i, $1 < i < k$, there will be voters simulating the voters of V_i. In particular, for each i, $1 < i < k$, and for each voter $\langle e_1 < e_2 < \cdots < e_{\|C_i\|}\rangle$ in V_i, we create a voter

$$\langle\ \overrightarrow{S}\ < \overrightarrow{C_1} < \cdots < \overrightarrow{C_{i-1}} < \overrightarrow{C_{i+1}} < \cdots < \overrightarrow{C_k} < e_1 < e_2 < \cdots < e_{\|C_i\|}\rangle.$$

Note that c is one of the e_j's.

—There will be voters simulating the voters of V_k. In particular, for each voter $\langle e_1 < e_2 < \cdots < e_{\|C_k\|}\rangle$ in V_k, we create a voter

$$\langle\ \overrightarrow{S}\ < \overrightarrow{C_1} < \cdots < \overrightarrow{C_{k-1}} < e_1 < e_2 < \cdots < e_{\|C_k\|}\rangle.$$

Note that c is one of the e_j's.

For each i, we want c's behavior with respect to candidates in C_i to depend only on voters that simulate V_i. That is, every candidate in $C_i \backslash \{c\}$ should be preferred to c by exactly half of the voters in $\hat{V}$ that do *not* simulate V_i. To accomplish this, we add $(\Sigma_j \|V_j\|) - 1$ normalizing voters.

—There will be $(\Sigma_j \|V_j\|) - 1$ normalizing voters. Each normalizing voter will have preferences of the form

$$\text{“some of the } \overrightarrow{C_j}\text{'s”} < c < \overrightarrow{S} < \text{“the rest of the } \overrightarrow{C_j}\text{'s.”}$$

Within the "some of" and "rest of" blocks, the order of the candidates can be arbitrary. So all that remains to do is to specify, for each particular one of the normalizing voters, how to decide which $\overrightarrow{C_j}$'s go to the left of c (the "some of" block), and which go to the right of $\overrightarrow{S}$ (the "rest of" block). Let us do so. Let the normalizing voters be named $\sigma_1, \ldots, \sigma_{(\Sigma_j \|V_j\|)-1}$. Consider normalizing voter σ_q. Then, for each i, in the preference of σ_q let it be the case that $\overrightarrow{C_i}$ goes to the right of $\overrightarrow{S}$ if

$$q \leq \lfloor \|V_i\|/2 \rfloor + \sum_{j \neq i} \|V_j\|,$$

and otherwise $\overrightarrow{C_i}$ goes to the left of c. Note that, for each i, exactly $\lfloor \|V_i\|/2 \rfloor + \Sigma_{j \neq i} \|V_j\|$ normalizing voters will have $\overrightarrow{C_i}$ to the right of S and exactly $\lfloor \|V_i\|/2 \rfloor$ normalizing voters will have $\overrightarrow{C_i}$ to the left of c.

Recall that $c = c_1 = \cdots = c_k$. We have to prove that $\Sigma_j \, Score(\langle C_j, c, V_j \rangle) = Score(\langle \hat{C}, c, \hat{V} \rangle)$.

First note that c is preferred to each candidate in S by $\Sigma_j \|V_j\|$ of the $(2 \, \Sigma_j \, \|V_j\|) - 1$ voters in $\hat{V}$. Also, for each i, it holds that c is preferred to all candidates in $C_i \backslash \{c\}$ by exactly half of the voters that do *not* simulate V_i. To see this, note that c is preferred to each candidate in $C_i \backslash \{c\}$ by all voters that simulate a V_j with $j \neq i$, and is also preferred by $\lfloor \|V_i\|/2 \rfloor$ of the normalizing voters. Thus, c is preferred to each candidate in $C_i \backslash \{c\}$ by $(\Sigma_{j \neq i} \|V_j\|) + \lfloor \|V_i\|/2 \rfloor$ of the $(\Sigma_{j \neq i} \|V_j\|) + (\Sigma_j \|V_j\|) - 1$ voters not simulating V_i, which indeed is exactly half of the voters not simulating V_i (recall that $\|V_i\|$ is odd).

For each i, let $K_i = Score(\langle C_i, c, V_i \rangle)$. Then after K_i switches in V_i, c is preferred to e by more than $\|V_i\|/2$ voters in V_i, for each $e \in C_i \backslash \{c\}$. This implies that after the analogous K_i switches in $\hat{V}$ (i.e., in the voters in $\hat{V}$ that simulate V_i), c is preferred to e by more than $\|V_i\|/2$ voters in that part of $\hat{V}$ that simulates V_i, for each $e \in C_i \backslash \{c\}$, and thus by more than half of the voters in $\hat{V}$. It follows that $\Sigma_j \, K_j$ switches in voters of $\hat{V}$ turn c into a Condorcet winner. This proves that $Score(\langle \hat{C}, c, \hat{V} \rangle) \leq \Sigma_j \, Score(\langle C_j, c, V_j \rangle)$.

It remains to show that $Score(\langle \hat{C}, c, \hat{V} \rangle) \geq \Sigma_j \, Score(\langle C_j, c, V_j \rangle)$. Let $\hat{K} = Score(\langle \hat{C}, c, \hat{V} \rangle)$. Then $\hat{K}$ switches in $\hat{V}$ turn c into a Condorcet winner. If $\hat{K} \geq \|S\|$, then $\hat{K} > \Sigma_j \, Score(\langle C_j, c, V_j \rangle)$, since $Score(\langle C_j, c, V_j \rangle) \leq \|V_j\| \cdot (\|C_j\| - 1)$ and so $\Sigma_j \, Score(\langle C_j, c, V_j \rangle) \leq \Sigma_j \, \|V_j\| \cdot (\|C_j\| - 1) < \Sigma_j \, \|V_j\| \cdot \|C_j\| = \|S\|$. So $\hat{K} \geq \|S\|$ is impossible, and we thus know that $\hat{K} < \|S\|$. With less than $\|S\|$ switches, c cannot gain extra votes over candidates in $(\bigcup_j C_j) \backslash \{c\}$ in normalizing voters, as can be immediately seen in light of the preferences of the normalizing voters. Also, for each i: Since c is already preferred to all candidates in $C_i \backslash \{c\}$ by all voters that simulate V_j with $j \neq i$, c cannot gain extra votes over candidates in $C_i \backslash \{c\}$ in voters simulating V_j with $j \neq i$. It follows that c can gain extra votes over candidates in $C_i \backslash \{c\}$ only in voters that simulate V_i. After $\hat{K}$ switches, c is still preferred to all candidates in $C_i \backslash \{c\}$ by at most half of the voters that do not simulate V_i, and at the same time, c has become a Condorcet winner. It follows that after these $\hat{K}$ switches, c is preferred to e by more than $\|V_i\|/2$ of the voters that simulate V_i, for each e in $C_i \backslash \{c\}$. Let M_i be the number of switches that take place in the voters of $\hat{V}$ that simulate V_i. Then $M_i \geq Score(\langle C_i, c, V_i \rangle)$.

Since this argument applies for all i, it follows that

$$Score(\langle \hat{C}, c, \hat{V} \rangle) = \hat{K} \geq \sum_j M_j \geq \sum_j Score(\langle C_j, c, V_j \rangle),$$

proving the lemma. □

PROOF OF LEMMA 3.6. Let A and f be the NP-complete set and the reduction from Lemma 3.4, and let *DodgsonSum* be the function from Lemma 3.5. We seek to apply Lemma 3.3, using the A (i.e., 3DM) of Lemma 3.4 as the A of Lemma 3.3, using 2ER as the B of Lemma 3.3, and using a function g that we will define in this proof as the g of Lemma 3.3.

Let $x_1, \ldots, x_{2k} \in \Sigma^*$ be such that $\chi_A(x_1) \geq \cdots \geq \chi_A(x_{2k})$. For $i = 1, \ldots, 2k$, let $f(x_i) = \langle \langle C_i, c_i, V_i \rangle, K_i \rangle$. We will write S_i for the Dodgson triple $\langle C_i, c_i, V_i \rangle$. We will compare the Dodgson score of the sum of the even Dodgson triples with the Dodgson score of the sum of the odd Dodgson triples, that is, we will look at the value of

$$Score(DodgsonSum(\langle S_2, S_4, \ldots, S_{2k} \rangle)) - Score(DodgsonSum(\langle S_1, S_3, \ldots, S_{2k-1} \rangle)).$$

By Lemma 3.5, this is the same as

$$\sum_{1 \leq i \leq k} (Score(S_{2i}) - Score(S_{2i-1})).$$

Recall that $\chi_A(x_1) \geq \cdots \geq \chi_A(x_{2k})$. If $\|\{i | x_i \in A\}\|$ is even then, for all i, $1 \leq i \leq k$, it holds that $x_{2i-1} \in A \Leftrightarrow x_{2i} \in A$. So, by Lemma 3.4, for each i, either $Score(S_{2i-1}) = K_{2i-1}$ and $Score(S_{2i}) = K_{2i}$, or $Score(S_{2i-1}) = K_{2i-1} + 1$ and $Score(S_{2i}) = K_{2i} + 1$. It follows that, for each i, $1 \leq i \leq k$,

$$Score(S_{2i}) - Score(S_{2i-1}) = K_{2i} - K_{2i-1}.$$

On the other hand, if $\|\{i | x_i \in A\}\|$ is odd then, for some j, $1 \leq j \leq k$, $x_{2j-1} \in A$ and $x_{2j} \notin A$ and, for all $i \neq j$, $1 \leq i \leq k$, it holds that $x_{2i-1} \in A \Leftrightarrow x_{2i} \in A$. It follows that $Score(S_{2j}) - Score(S_{2j-1}) = 1 + K_{2j} - K_{2j-1}$ and, for all $i \neq j$, $1 \leq i \leq k$, $Score(S_{2i}) - Score(S_{2i-1}) = K_{2i} - K_{2i-1}$.

To summarize,

$$Score(DodgsonSum(\langle S_2, S_4, \ldots, S_{2k} \rangle)) - Score(DodgsonSum(\langle S_1, S_3, \ldots, S_{2k-1} \rangle))$$

$$= \begin{cases} \sum_{1 \le i \le k} K_{2i} - \sum_{1 \le i \le k} K_{2i-1} & \text{if} \quad \|\{i | x_i \in A\}\| \text{ is even, and} \\ 1 + \sum_{1 \le i \le k} K_{2i} - \sum_{1 \le i \le k} K_{2i-1} & \text{if} \quad \|\{i | x_i \in A\}\| \text{ is odd.} \end{cases}$$

This implies that $\|\{i | x_i \in A\}\|$ is odd if and only if

$$Score(DodgsonSum(\langle S_2, S_4, \ldots, S_{2k}\rangle)) + \sum_{1 \le i \le k} K_{2i-1} \ge$$

$$Score(DodgsonSum(\langle S_1, S_3, \ldots, S_{2k-1}\rangle)) + 1 + \sum_{1 \le i \le k} K_{2i}.$$

For any integer $m \ge 1$, define a Dodgson triple

$$T_m = \langle \{i | 1 \le i \le m + 1\}, 1, \{\langle 1 < 2 < 3 < \cdots < m + 1\rangle\}\rangle.$$

Then, T_m has an odd number of voters (namely one), and $Score(T_m) = m$. Thus, again by Lemma 3.5, $\|\{i | x_i \in A\}\|$ is odd if and only if

$$Score(DodgsonSum(\langle S_2, S_4, \ldots, S_{2k}, T_{\Sigma_{1 \le i \le k} K_{2i-1}}\rangle)) \ge$$

$$Score(DodgsonSum(\langle S_1, S_3, \ldots, S_{2k-1}, T_{1+\Sigma_{1 \le i \le k} K_{2i}}\rangle)).$$

Given $x_1, \ldots, x_{2k}$, define the function $g(x_1, \ldots, x_{2k}) = \langle\langle C, c, V\rangle, \langle D, d, W\rangle\rangle$, where

$$\langle C, c, V\rangle = DodgsonSum(\langle S_1, S_3, \ldots, S_{2k-1}, T_{1+\Sigma_{1 \le i \le k} K_{2i}}\rangle)$$

and

$$\langle D, d, W\rangle = DodgsonSum(\langle S_2, S_4, \ldots, S_{2k}, T_{\Sigma_{1 \le i \le k} K_{2i-1}}\rangle),$$

and (without loss of generality, via trivial renaming if necessary) $c \neq d$.

Note that $g(x_1, \ldots, x_{2k})$ is computable in time polynomial in $|x_1| + |x_2| + \cdots + |x_{2k}| + 2k$ (recall the conventions regarding variable-arity functions discussed in Section 2 and Footnote 3). Since

$$Score(\langle C, c, V\rangle) \le Score(\langle D, d, W\rangle) \Leftrightarrow \|\{i | x_i \in A\}\| \text{ is odd},$$

it follows by Lemma 3.3 that the problem 2ER is Θ_2^p-hard. □

PROOF OF LEMMA 3.7. Without loss of generality, we assume that $\|V\| \ge \|W\|$ and that $C \cap D = \emptyset$. Also, enumerate $C \setminus \{c\}$ as $\{c_1, c_2, \ldots, c_{\|C\|-1}\}$, and $D \setminus \{d\}$ as $\{d_1, d_2, \ldots, d_{\|D\|-1}\}$.

The construction and proof are similar in flavor to the construction and proof of Lemma 3.5. However, in this proof, the number of voters has to be even, as we seek to ensure that c is preferred to d by exactly half of the voters.

We define a set of "separating" candidates: $S = \{s_i | 1 \le i \le 2(\|C\| \cdot \|V\| + \|D\| \cdot \|W\|)\}$. We will also use another set of separating candidates, $T = \{t_i | 1 \le i \le \|S\|\}$, of the same cardinality as S. Let $m = \|S\|/2$. Let $\hat{C} = C \cup D \cup S \cup T$. The set of new voters $\hat{V}$ consists of the following subparts:

(a) Voters simulating V: for each voter $\langle e_1 < e_2 < \cdots < e_{\|C\|}\rangle$ in V, we create a voter

$$\langle d < s_1 < \cdots < s_{\|S\|} < d_1 < \cdots < d_{\|D\|-1} < t_1 < \cdots < t_{\|T\|} < e_1 < e_2 < \cdots < e_{\|C\|}\rangle.$$

(b) Voters simulating W: for each voter $\langle e_1 < e_2 < \cdots < e_{\|D\|}\rangle$ in W, we create a voter

$$\langle t_1 < \cdots < t_{\|T\|} < c < s_1 < \cdots < s_{\|S\|} < c_1 < \cdots < c_{\|C\|-1} < e_1 < e_2 < \cdots < e_{\|D\|}\rangle.$$

In addition, we create $\|V\| + 1$ normalizing voters (recall that $\|V\|$ and $\|W\|$ are both odd), consisting of three subparts:

(c) $\lceil \|V\|/2 \rceil - \lceil \|W\|/2 \rceil$ voters:

$$\langle t_1 < \cdots < t_{\|T\|} < c < s_1 < \cdots < s_{\|S\|} < c_1 < \cdots < c_{\|C\|-1} < d_1 < \cdots < d_{\|D\|-1} < d\rangle.$$

(d) $\lceil \|V\|/2 \rceil$ voters:

$$\langle t_1 < \cdots < t_{\|T\|} < c_1 < \cdots < c_{\|C\|-1} < d_1 < \cdots < d_{\|D\|-1} < s_{\|S\|} < \cdots < s_1 < c < d\rangle.$$

(e) $\lceil \|W\|/2 \rceil$ voters:

$$\langle t_1 < \cdots < t_{\|T\|} < c_1 < \cdots < c_{\|C\|-1} < d_1 < \cdots < d_{\|D\|-1} < s_1 < \cdots < s_{\|S\|} < d < c\rangle.$$

The above construction of $\hat{C}$ and $\hat{V}$ defines our functions $Merge(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle \hat{C}, c, d, \hat{V}\rangle$ and $Merge'(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle \hat{C}, c, \hat{V}\rangle$. These functions clearly satisfy properties (i) and (ii) of Lemma 3.7.

To satisfy properties (iii) and (iv), we have to prove that $Score(\langle \hat{C}, c, \hat{V}\rangle) = Score(\langle C, c, V\rangle) + 1$ and that $Score(\langle \hat{C}, d, \hat{V}\rangle) = Score(\langle D, d, W\rangle) + 1$.

First note that c is preferred to every candidate in $(S \cup D)\backslash\{d\}$ by $\|V\| + \lceil \|V\|/2 \rceil + \lceil \|W\|/2 \rceil$ of the $2\|V\| + \|W\| + 1$ voters in $\hat{V}$. Similarly, d is preferred to every candidate in $(S \cup C)\backslash\{c\}$ by $\|W\| + \|V\| + 1$ of the $2\|V\| + \|W\| + 1$ voters in $\hat{V}$. Similarly, c is preferred to each $t \in T$ by all voters in $\hat{V}$, and d is preferred to each $t \in T$ by $\|V\| + \|W\| + 1$ of the $2\|V\| + \|W\| + 1$ voters in $\hat{V}$.

In addition, c is preferred to all candidates in $C\backslash\{c\}$ by $\lceil \|V\|/2 \rceil + \lceil \|W\|/2 \rceil = (\|V\| + \|W\|)/2 + 1$ of the $\|V\| + \|W\| + 1$ voters that do *not* simulate V. Likewise, d is preferred to all candidates in $D\backslash\{d\}$ by $\|V\| + 1$ of the $2\|V\| + 1$ voters not simulating W. Finally, c is preferred to d by $\|V\| + \lceil \|W\|/2 \rceil = (2\|V\| + \|W\| + 1)/2$ of the $2\|V\| + \|W\| + 1$ voters in $\hat{V}$—exactly half.

Let $K = Score(\langle C, c, V\rangle)$. Then after K switches in $\hat{V}$, c is preferred to e by more than $\|V\|/2$ voters in that part of $\hat{V}$ that simulates V, for every $e \in C\backslash\{c\}$, and thus by more than half of the voters in $\hat{V}$. It follows that after K switches, c is preferred to e by a majority of voters, for all $e \in \hat{C}\backslash\{c, d\}$. If, in addition to these K switches, we switch c and d in a normalizing voter of the form $\langle t_1 < \cdots < t_{\|T\|} < c_1 < \cdots < c_{\|C\|-1} < d_1 < \cdots < d_{\|D\|-1} < s_{\|S\|} < \cdots < s_1 < c < d\rangle$, then c has become a Condorcet winner. Thus, $Score(\langle \hat{C}, c, \hat{V}\rangle) \leq K + 1 = Score(\langle C, c, V\rangle) + 1$. In exactly the same way, we can show that $Score(\langle \hat{C}, d, \hat{V}\rangle) \leq Score(\langle D, d, W\rangle) + 1$.

It remains to show that $Score(\langle \hat{C}, c, \hat{V}\rangle) \geq Score(\langle C, c, V\rangle) + 1$ and that $Score(\langle \hat{C}, d, \hat{V}\rangle) \geq Score(\langle D, d, W\rangle) + 1$. Let $\hat{K} = Score(\langle \hat{C}, c, \hat{V}\rangle)$. Then $\hat{K}$ switches in $\hat{V}$ turn c into a Condorcet winner. Recall that $m = \|S\|/2$. If $\hat{K} \geq m$, then $\hat{K} > Score(\langle C, c, V\rangle) + 1$, since $Score(\langle C, c, V\rangle) < \|C\| \cdot \|V\| < m$ (recall $\|W\|$ is odd and thus nonzero, and without loss of generality we assume $\|D\| > 0$). So $\hat{K} \geq m$ is impossible, which implies that $\hat{K} < m$. In order to become a Condorcet winner, c in particular needs to gain one vote over d. With less than m switches, the *only* way in which c can gain this vote is by switching c and d in a normalizing voter of the form $\langle t_1 < \cdots < t_{\|T\|} < c_1 < \cdots < c_{\|C\|-1} < d_1 < \cdots < d_{\|D\|-1} < s_{\|S\|} < \cdots < s_1 < c < d\rangle$. This uses one of the $\hat{K}$ switches.

With less than m switches, c cannot gain extra votes over candidates in $C \setminus \{c\}$ in normalizing voters, or in voters that simulate W. It follows that c can gain extra votes over candidates in $C \setminus \{c\}$ only in voters that simulate V. After $\hat{K}$ switches, c is still preferred to all candidates in $C \setminus \{c\}$ by at most the smallest possible majority of the (odd) number of voters that do not simulate V, and at the same time, c has become a Condorcet winner. Since $\|V\|$ is odd, it follows that, after these $\hat{K}$ switches, c is preferred to e by more than $\|V\|/2$ voters that simulate V, for every e in $C \setminus \{c\}$. Let $\hat{K}_V$ be the number of switches that take place in the voters of $\hat{V}$ that simulate V. Then $\hat{K}_V \geq Score(\langle C, c, V\rangle)$. Since we had to use one switch to switch c and d in a normalizing voter,

$$Score(\langle \hat{C}, c, \hat{V}\rangle) = \hat{K} \geq \hat{K}_V + 1 \geq Score(\langle C, c, V\rangle) + 1.$$

The same argument can be used to show that

$$Score(\langle \hat{C}, d, \hat{V}\rangle) \geq Score(\langle D, d, W\rangle) + 1,$$

which proves properties (iii) and (iv).

Finally, we prove property (v) of the lemma: For each $e \in \hat{C} \setminus \{c, d\}$, $Score(\langle \hat{C}, c, \hat{V}\rangle) < Score(\langle \hat{C}, e, \hat{V}\rangle)$. First note that we have chosen S sufficiently large to ensure that $Score(\langle \hat{C}, c, \hat{V}\rangle) < m$, since $Score(\langle C, c, V\rangle) < \|C\| \cdot \|V\| < m$ and $Score(\langle \hat{C}, c, \hat{V}\rangle) = Score(\langle C, c, V\rangle) + 1$ by property (iii).

Consider $t_{\|T\|}$. In order to become a Condorcet winner, $t_{\|T\|}$ must in particular outpoll d in pairwise elections. In the specified preferences, $t_{\|T\|}$ is preferred to d by $\|V\|$ of the $2\|V\| + \|W\| + 1$ voters in $\hat{V}$. Thus, more than $\lceil \|W\|/2 \rceil$ of the voters not simulating V must be convinced to prefer $t_{\|T\|}$ to d. However, to gain even one additional vote over d amongst the voter groups (b), (c), (d), and (e), $t_{\|T\|}$ would require more than m switches upwards. Since $Score(\langle \hat{C}, c, \hat{V}\rangle) < m$, the score of c is less than that of $t_{\|T\|}$. The same argument applies to any t_i, $1 \leq i \leq \|T\|$.

Consider $s_{\|S\|}$. In order to become a Condorcet winner, $s_{\|S\|}$ must in particular outpoll c in pairwise elections. Initially, $s_{\|S\|}$ is preferred to c by $\|W\| + \lceil \|V\|/2 \rceil - \lceil \|W\|/2 \rceil = \lceil \|V\|/2 \rceil + \lceil \|W\|/2 \rceil - 1$ voters, namely those belonging to (b) and (c). Thus, more than $\lceil \|V\|/2 \rceil$ of the voters amongst (a), (d), and (e) must be convinced to prefer $s_{\|S\|}$ to c. However, to gain one more vote over c in (a), $s_{\|S\|}$ would need more than m switches upwards. Likewise, for $s_{\|S\|}$ to gain one more vote over c in (d), it would also have to switch more than m times upwards. Finally, to gain one more vote over c in (e), $s_{\|S\|}$ needs only two switches per vote. However, since there are no more than $\lceil \|W\|/2 \rceil \leq \lceil \|V\|/2 \rceil$ voters in (e) and $s_{\|S\|}$ needs to be preferred over c by *more* than $\lceil \|V\|/2 \rceil$ additional voters, $s_{\|S\|}$ cannot become a Condorcet winner by changing only the minds of the voters in (e). It follows that $Score(\langle \hat{C}, c, \hat{V}\rangle) < Score(\langle \hat{C}, s_{\|S\|}, \hat{V}\rangle)$.

Consider s_1. As above, for s_1 to become a Condorcet winner, more than $\lceil \|V\|/2 \rceil$ of the voters amongst (a), (d), and (e) must be convinced to prefer s_1 to

c in particular. Now, to gain one vote in either (a) or (e) requires more than m switches. However, similarly to the previous paragraph, the remaining $\lceil \|V\|/2 \rceil$ voters in (d) alone are too few to make s_1 a Condorcet winner. It follows that $Score(\langle \hat{C}, c, \hat{V} \rangle) < Score(\langle \hat{C}, s_1, \hat{V} \rangle)$. Note that for each $s \in S \setminus \{s_1, s_{\|S\|}\}$, at least one of the two given arguments (the one for $s_{\|S\|}$ and the one for s_1) apply, yielding $Score(\langle \hat{C}, c, \hat{V} \rangle) < Score(\langle \hat{C}, s, \hat{V} \rangle)$, since each such s needs more than m switches (because $m = \|S\|/2$) to gain one vote in either (d) or (e).

Finally, consider $d_{\|D\|-1}$. As was the case for the elements of S, more than $\lceil \|V\|/2 \rceil$ of the voters amongst (a), (d), and (e) must be convinced to prefer $d_{\|D\|-1}$ to c in order for $d_{\|D\|-1}$ to become a Condorcet winner. However, more than m switches would be required to gain even one vote from one of (a), (d), or (e). Thus, $Score(\langle \hat{C}, c, \hat{V} \rangle) < Score(\langle \hat{C}, d_{\|D\|-1}, \hat{V} \rangle)$. The same argument applies to each element in $(C \cup D)\setminus\{c, d\}$. To summarize, we have shown that property (v) holds. □

Having proven these lemmas, we may now turn to the proof of the Theorem 3.1.

PROOF OF THEOREM 3.1. To prove the Θ_2^p-completeness of `DodgsonRanking`, we must according to the definition prove both an upper bound (`DodgsonRanking` $\in \Theta_2^p$) and a lower bound (`DodgsonRanking` is Θ_2^p-hard).

To prove the lower bound, it suffices to provide a $\le_m^p$-reduction, f, from `2ER` to `DodgsonRanking`. f is defined as follows. Let t_o be some fixed string that is not in `DodgsonRanking`. $f(x)$ is defined as being t_o if x is not an instance of `2ER`, and as being $Merge(x_1, x_2)$ otherwise, where $x = \langle x_1, x_2 \rangle$ and $Merge$ is as defined in Lemma 3.7. Note that $Merge$ and thus f are polynomial-time computable. Note also that for any instance $\langle \langle C, c, V \rangle, \langle D, d, W \rangle \rangle$ of `2ER`, it holds that if $\langle \hat{C}, c, d, \hat{V} \rangle = Merge(\langle C, c, V \rangle, \langle D, d, W \rangle)$, then

$$\langle \langle C, c, V \rangle, \langle D, d, W \rangle \rangle \in \texttt{2ER} \Leftrightarrow \langle \hat{C}, c, d, \hat{V} \rangle \in \texttt{DodgsonRanking}$$

by properties (iii) and (iv) of Lemma 3.7. Note also that for any input x that is not an instance of `2ER`, $f(x)$ maps to t_o, a string that is not in `DodgsonRanking`. Thus, f is a $\le_m^p$-reduction from `2ER` to `DodgsonRanking`. From Lemma 3.6, Θ_2^p-hardness of `DodgsonRanking` follows immediately.

Finally, we claim that `DodgsonRanking` is in Θ_2^p. This can be seen as follows: We can in parallel ask all plausible `DodgsonScore` queries for each of the two designated candidates, say c and d, and from this compute the exact score of each of c and d and thus we can tell whether c ties-or-defeats d. Note that there is a polynomial upper bound on the highest possible score (this is what was meant above by "plausible"), and thus this procedure indeed can be implemented via a polynomial-time truth-table reduction to the NP-complete set `DodgsonScore`. However, the class of languages accepted via polynomial-time truth-table reductions to NP sets coincides with Θ_2^p [Hemachandra 1989; Köbler et al. 1987]. This establishes the upper bound, that is, that `DodgsonRanking` $\in \Theta_2^p$. □

`DodgsonWinner` is similarly Θ_2^p-complete.

THEOREM 3.8. *`DodgsonWinner` is Θ_2^p-complete.*

COROLLARY 3.9. *If `DodgsonWinner` is NP-complete, then PH = NP.*

Bartholdi et al. [1989] have stated without proof that `DodgsonRanking` $\leq_m^p$ `DodgsonWinner`. Theorem 3.1 plus this assertion would prove Theorem 3.8. However, as we wish our proof to be complete, we now prove Theorem 3.8. (We note in passing that our paper implicitly provides an indirect proof of their assertion. In particular, given that one has proven Theorem 3.1 and Theorem 3.8, the assertion follows, since it follows from the definition of Θ_2^p-completeness that all Θ_2^p-complete problems are $\leq_m^p$-interreducible.)

PROOF OF THEOREM 3.8. As in the case of `DodgsonRanking`, `DodgsonWinner` $\in \Theta_2^p$ is easily seen to hold, since we can in parallel ask all plausible `DodgsonScore` queries for each of the given candidates (note that the number of candidates and the highest possible score for each candidate are both polynomially bounded in the input length) and thus can compute the exact Dodgson score for each candidate. After having done so, it is easy to decide whether or not the designated candidate c ties-or-defeats all other candidates in the election. This proves the upper bound.

To prove the lower bound, we will provide a polynomial-time many-one reduction from `2ER` to `DodgsonWinner`. By Lemma 3.6, the claim of this theorem then follows. In fact, the following function f provides a polynomial-time many-one reduction from `2ER` to `DodgsonWinner`. Let t_o be some fixed string that is not in `DodgsonWinner`. $f(x)$ is defined as being t_o if x is not an instance of `2ER`, and as being $Merge'(x_1, x_2)$ otherwise, where $x = \langle x_1, x_2 \rangle$ and $Merge'$ is as defined in Lemma 3.7. To see that this is correct, note that f is polynomial-time computable, and that when x is not an instance of `2ER`, then $f(x)$ is not in `DodgsonWinner`.

We now turn to the behavior of $f(x)$ when x is an instance of `2ER`. Given any pair of Dodgson triples, $\langle C, c, V \rangle$ and $\langle D, d, W \rangle$, for which both $\|V\|$ and $\|W\|$ are odd and $c \neq d$, let $\langle \hat{C}, c, \hat{V} \rangle = Merge'(\langle C, c, V \rangle, \langle D, d, W \rangle)$. Assume $Score(\langle C, c, V \rangle) \leq Score(\langle D, d, W \rangle)$. By properties (iii) and (iv) of Lemma 3.7, it follows that $Score(\langle \hat{C}, c, \hat{V} \rangle) \leq Score(\langle \hat{C}, d, \hat{V} \rangle)$ as well. However, since by property (v) of Lemma 3.7 $Score(\langle \hat{C}, c, \hat{V} \rangle) < Score(\langle \hat{C}, e, \hat{V} \rangle)$ for every $e \in \hat{C} \setminus \{c, d\}$, it follows that c is a winner of the election specified by $\hat{C}$ and $\hat{V}$. Conversely, assume $Score(\langle C, c, V \rangle) > Score(\langle D, d, W \rangle)$. Again, properties (iii) and (iv) of Lemma 3.7 imply that $Score(\langle \hat{C}, c, \hat{V} \rangle) > Score(\langle \hat{C}, d, \hat{V} \rangle)$. Thus, c is not a winner of the election specified by $\hat{C}$ and $\hat{V}$. $\square$

Finally, recall that our multisets are specified as a list containing, for each voter, the preference order of that voter. Our main theorem, Theorem 3.8, proves that checking if a candidate is a Dodgson winner is Θ_2^p-complete. Is this complexity coming from the number of candidates, or is the problem already complex with, for example, fixed numbers of candidates? In fact, for each fixed constant k, there clearly is a polynomial-time algorithm to compute all Dodgson scores, and thus all Dodgson winners, in elections having at most k candidates.

PROPOSITION 3.10 [BARTHOLDI ET AL. 1989]. *Let k be any fixed positive integer. There is a polynomial-time algorithm A_k that computes all Dodgson scores (and thus all Dodgson winners) in Dodgson elections having at most k candidates.*

Proposition 3.10 in no way conflicts with Theorem 3.8. In fact, though each A_k is a polynomial-time algorithm, the degree of the polynomial runtimes of the A_k is itself exponential in k. It is also known that, for each fixed constant k, there is a polynomial-time algorithm to compute all Dodgson winners in elections having at most k voters [Bartholdi et al. 1989].

4. *Conclusions*

This paper establishes that testing whether a given candidate wins a Dodgson election is Θ_2^p-complete, thus providing the first truly natural complete problem for the class Θ_2^p.

In this paper, we assumed that no voter views any two candidates as being of equal desirability. However, note that if one allows such ties, our Θ_2^p-hardness result remains valid, as our case is simply a special case of this broader problem. On the other hand, it is not hard to see that the broader problem remains in Θ_2^p (in both of the natural models of switches involving ties, that is, the model in which moving from $\langle a = b < c \rangle$ to $\langle c < a = b \rangle$ requires just one switch, and the model in which this requires two separate switches). Thus, this broader problem is also Θ_2^p-complete.

Since this paper first appeared, some related work has been done that may be of interest to readers of this paper. Hemaspaandra and Wechsung [1997] have shown that the minimization problem for Boolean formulas is Θ_2^p-hard; it remains open whether that problem is Θ_2^p-complete. Hemaspaandra and Rothe [1997] have shown that recognizing the instances on which the greedy algorithm can obtain independent sets that are within a certain fixed factor of optimality is itself a Θ_2^p-complete task. Hemaspaandra et al. [1997] have discussed the relationship between raising a problem's lower bound from NP-hardness to Θ_2^p-hardness and its potential solvability via such modes of computation as randomized and approximate computing.

ACKNOWLEDGMENTS. We are indebted to J. Banks and R. Calvert for recommending Dodgson elections to us as an interesting open topic worthy of study, and for providing us with the literature on this topic. We thank D. Austen-Smith, J. Banks, R. Calvert, A. Rutten, M. Scott, and J. Seiferas for helpful conversations and suggestions. L. Hemaspaandra thanks J. Banks and R. Calvert for arranging, and J. Banks for supervising, his Bridging Fellowship at the University of Rochester's Department of Political Science, during which this project was started. An extended abstract of this paper appeared in the Twenty-Fourth International Colloquium on Automata, Languages, and Programming, and this paper benefited from the comments of its anonymous referees. The authors are also very grateful to two anonymous JACM referees for helpful comments, suggestions, and rewordings. This work was done in part while the first two authors were visiting Friedrich-Schiller-Universität Jena and the University of Amsterdam, and while the third author was visiting Le Moyne College and the University of Rochester, and we thank the host departments for their hospitality.

REFERENCES

BARTHOLDI III, J., TOVEY, C., AND TRICK, M. 1989. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare 6*, 157–165.

BLACK, D. 1958. *Theory of Committees and Elections*. Cambridge University Press, Cambridge, Mass.

BOVET, D., AND CRESCENZI, P. 1993. *Introduction to the Theory of Complexity*. Prentice-Hall, Englewood Cliffs, N.J.

CONDORCET, M. 1785. *Essai sur l'Application de L'Analyse à la Probabilité des Décisions Rendues à la Pluraliste des Voix*. Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale.

DODGSON, C. 1876. 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 McLean and Urken [1995] and Black [1958], both of which reprint this paper).

GAREY, M., AND JOHNSON, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

HEMACHANDRA, L. 1989. The strong exponential hierarchy collapses. *J. Comput. Syst. Sci. 39*, 3, 299–322.

HEMACHANDRA, L., AND WECHSUNG, G. 1991. Kolmogorov characterizations of complexity classes. *Theoret. Comput. Sci. 83*, 313–322.

HEMASPAANDRA, E., HEMASPAANDRA, L., AND ROTHE, J. 1997. Raising NP lower bounds to parallel NP lower bounds. *SIGACT News 28*, 2, 2–13.

HEMASPAANDRA, E., AND ROTHE, J. 1997. Recognizing when greed can approximate maximum independent sets is complete for parallel access to NP. Tech. Rep. Math/Inf/97/14 (May). Institut für Informatik, Friedrich-Schiller-Universität Jena, Jena, Germany.

HEMASPAANDRA, E., AND WECHSUNG, G. 1997. The minimization problem for boolean formulas. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press. New York, pp. 575–584.

JENNER, B., AND TORÁN, J. 1995. Computing functions with parallel queries to NP. *Theoret. Comput. Sci. 141*, 1–2, 175–193. Corrigendum available at http://www.informatik.uni-ulm.de/abt/ti/Personen/jtpapers.html.

KADIN, J. 1989. $P^{NP[\log n]}$ and sparse Turing-complete sets for NP. *J. Comput. Syst. Sci. 39*, 3, 282–298.

KÖBLER, J., SCHÖNING, U., AND WAGNER, K. 1987. The difference and truth-table hierarchies for NP. *RAIRO Theoret. Inf. Appl. 21*, 419–435.

LADNER, R., LYNCH, N., AND SELMAN, A. 1975. A comparison of polynomial time reducibilities. *Theoret. Comput. Sci. 1*, 2, 103–124.

MCLEAN, I., AND URKEN, A. 1995. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Mich.

MEYER, A., AND STOCKMEYER, L. 1972. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*. IEEE, New York, pp. 125–129.

MUELLER, D. 1989. *Public Choice II*. Cambridge University Press, Cambridge, Mass.

NIEMI, R., AND RIKER, W. 1976. The choice of voting systems. *Scient. Am. 234*, 21–27.

PAPADIMITRIOU, C. 1984. On the complexity of unique solutions. *J. ACM 31*, 2, 392–400.

PAPADIMITRIOU, C. 1994. *Computational Complexity*. Addison-Wesley, Cambridge, Mass.

PAPADIMITRIOU, C., AND ZACHOS, S. 1983. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*. Springer-Verlag Lecture Notes in Computer Science, vol. 145. Springer-Verlag, New York, pp. 269–276.

SANKOFF, D., AND KRUSKAL, J., EDS. 1983. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, Pa.

STOCKMEYER, L. 1977. The polynomial-time hierarchy. *Theoret. Comput. Sci. 3*, 1–22.

WAGNER, K. 1987. More complicated questions about maxima and minima, and some closures of NP. *Theoret. Comput. Sci. 51*, 1–2, 53–80.

WAGNER, K. 1990. Bounded query classes. *SIAM J. Comput. 19*, 5, 833–846.

RECEIVED NOVEMBER 1996; REVISED MAY 1997; ACCEPTED SEPTEMBER 1997