The Dodgson Winner Problem

Dhamma Kimpara

August 13, 2020

1 Introduction

Suppose you are a theoretical computer scientist, wandering in the wilds of theory-land and you come upon a complexity class. It has not been seen before. It seems intuitive and natural. So you decide to submit it to the complexity zoo [Aaronson et al., 2005]. But the worry is whether this class will turn out to capture the complexity of important real-world problems. In other words, is anyone or any problems going to come around to your new section of the zoo and join your party? After all, there are 545 classes and counting in the zoo!

This was the case for the class Θ_2^pin the mid-1990s. Θ_2^por $\mathsf{P}_{||}^{\mathsf{NP}}$, the class of languages decidable by a polynomial time Turing machine with parallel access to NP, was introduced in 1983 by [Papadimitriou and Zachos, 1982]. The location of this class in the polynomial hierarchy is:

By the mid 1990s, the theoretical importance of $\Theta_2^p$was recognized in complexity theory. Klaus W. Wagner established half a dozen characterizations of Θ_2^p[Wagner, 1990], several complete problems and a toolkit for establishing Θ_2^p-hardness. Hemachandra [1987] and Köbler, Johannes et al. [1987] showed that $\Theta_2^p$was equivalent to the class of problems that can be solved by $\mathcal{O}(\log(n))$ sequential Turing queries to NP. Furthermore, if NPcontains some Θ_2^p-hard problem, then the polynomial hierarchy collapses to NP. However, these connections and complete problems lived in the pure theory section of the zoo and did not yet have the appeal of problems from "real world" settings.

Along came the Dodgson winner problem, invented in 1876 by Charles Lutwidge Dodgson, better known under the pen name of Lewis Carroll. Hemaspaandra et al. [1999] proved that this problem was Θ_2^p-complete. This was the first "real-world" problem proven complete for the class Θ_2^p.

Dodgson's election system takes the following form. An election is a finite number of voters, who each cast a linear order over a common finite set of candidates. Note that linear orders are "tie-free". The winner is determined by whichever candidate is closest to being a Cordorcet winner, a criteria used by other election systems. A Condorcet winner is a candidate a who for every other candidate b, is preferred to b by strictly more than half of the voters. We naturally want election systems to be Condorcet-consistent, i.e. the system has the property where if a is a Condorcet winner, a is the one and only winner in the election. Dodgson's election system is Condorcet-consistent [Brandt et al., 2016]

The winner(s) in a Dodgson election is defined as the candidate(s) who are the "closest" to being Condorcet winners. The winners are the candidates that have the lowest Dodgson score. The Dodgson score of a candidate a is the smallest number of sequential exchanges of adjacent candidates in preference orders such that after those exchanges a is a Cordorcet winner.

Note that it is remarkable that we find Θ_2^p-complete problems that were defined 100 years before complexity theory itself existed. Dodgson winner is also extremely natural when compared with other complete problems in this class such as determining if the maximum size clique in a graph is of odd size (odd-max-clique). In this project we will use COMP-SAT, which was recently shown to be complete for this class [Lukasiewicz and Malizia, 2017].

The rest of this project presents the theory of and a practical algorithm for the Dodgson winner problem. We present slightly modified proof of completeness of the Dodgson winner problem based on new results for the class Θ_2^p. Then we present, implement, and examine a heuristic algorithm that is self-knowingly correct for most practical instances of the problem.

2 Preliminaries

2.1 Problem Setting

Definition 1. Dodgson Triple
A triple, $\langle C, c, V\rangle$, where C is the set of candidates $1, \ldots, m$, a candidate $c \in C$, and a set of n strict (ie. irreflexive and anti-symmetric) preference orders, one per voter, over all candidates in C.

Definition 2. Preference relations
$<\succ$

Definition 3. Condorcet Winner
In an election, with a set of candidates C and n votes or strict preference orders V, a candidate $a \in C$ is a Condorcet Winner if for every other candidate b, $a > b$ by strictly more than half of the voters.

Definition 4. Dodgson Score
First define a switch as an exchange of two adjacent preferences in the preference order of one voter. Then, the Dodgson Score of a candidate is the smallest number of sequential switches needed to make the candidate a Condorcet winner. The Dodgson Score of any Cordorcet Winner is 0. We denote the Dodgson score of a Dodgson triple as $Score(\langle C, c, V\rangle)$

Decision Problem. DodgsonScore
Instance: $k \in \mathbb{N}$. A Dodgson Election and Candidate $\langle C, c, V\rangle$.
Decide: Is the Dodgson Score of candidate c less than or equal to k?

Decision Problem. DodgsonWinner
Instance: A Dodgson Election and Candidate $\langle C, c, V\rangle$
Decide: Is c a winner of the election? In other words, does c have the minimum Dodgson Score in the election?

Decision Problem. Comp-M
Instance: A pair $\langle A, B\rangle$ of sets of NP-hard decision problems in M.
Decide: Is the number of yes-instances in A greater than or equal to the number of yes-instances in B.

Decision Problem. Comp-SAT [Lukasiewicz and Malizia, 2017]
Instance: A pair $\langle A, B\rangle$ of sets of 3CNF formulas.
Decide: Is the number of satisfiable formulas in A greater than or equal to the number of satisfiable formulas in B.

2.2 Complexity Classes and Definitions

Definition 5. Θ_2^p:
The class of problems solvable with polynomial-time parallel access to an NPoracle. This is equivalent to $\mathcal{O}(\log(n))$ sequential queries to an NPoracle [Hemachandra, 1987, Köbler, Johannes et al., 1987].

Theorem 1. [Lukasiewicz and Malizia, 2017] Comp-SAT is Θ_2^p-complete.

3 The complexity of the Dodgson Winner Problem

Our results largely follow the proof in Hemaspaandra et al. [1999], except instead of using a technical lemma by Wagner [1990], we instead begin the reduction with the recent result that Comp-SAT is Θ_2^p-complete

[Lukasiewicz and Malizia, 2017]. All lemmas and theorems in this section unless cited otherwise are from Hemaspaandra et al. [1999]. We have omitted proofs of technical lemmas and have used our own proof of Lemma 3, which is the main hardness proof.

We first state the main result of this section and a big-picture corollary concerning the polynomial hierarchy.

Theorem 2. DodgsonWinner is Θ_2^p-complete

It follows that though DodgsonWinner is NP-hard [Bartholdi et al., 1989], it cannot be NP-complete unless the polynomial hierarchy collapses.

Corollary 1. If DodgsonWinner is NP-complete then $\mathsf{PH} = \mathsf{NP}$.

3.1 Outline of Proof of Theorem 2

First we briefly outline the reduction from Comp-SAT to DodgsonWinner. Let us start with a Comp-SAT instance $\langle A, B\rangle$, for each 3CNF formula x in A and B, reduce x into the corresponding instance of ThreeDimensionalMatching (3DM), which is possible because 3DM is NP-complete. Now, for each 3DM problem, we reduce it into an instance of DodgsonScore. Using a merger lemma, we will merge these into two elections, one for A and one for B, thus an instance of 2ElectionRanking (2ER). Then 2ER will be reduced to DodgsonWinner. For the formal proof we will use the problems Comp-3DM and Comp-DodgsonScore which are similar to Comp-SAT.

A reference diagram is provided:

$$\text{Comp-SAT} \leq_m^p \text{Comp-3DM} \leq_m^p \text{Comp-DodgsonScore} \leq_m^p \text{2ER} \leq_m^p \text{DodgsonWinner}$$

Now we need to show membership in Θ_2^p.

Theorem 3. DodgsonWinner $\in \Theta_2^p$.

Proof. Ask in parallel all the DodgsonScore queries, one for each candidate in C. Each query is an NP query since DodgsonScore is NP-complete by the second reduction outlined above (lemma 1). We now have the exact Dodgson Score for each candidate. It is easy to decide whether or not the given candidate c ties-or-defeats all other candidates in the election (find the max of the scores and compare the c's score to the max $\sim \mathcal{O}(n)$). □

3.2 Outline of relevant lemmas

Lemma 1. There exists an NP-complete problem A and a polynomial-time computable function f that reduces A to DodgsonScore in such a way that, $\forall x \in \Sigma^*, f(x) = \langle\langle C, c, V\rangle, k\rangle$ 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
2. if $x \notin A$ then $Score(\langle C, c, V\rangle) = k + 1$.

Lemma 2. There exists a polynomial-time computable function *DodgsonSum* such that $\forall k$ 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 $||V_j||$ is odd for all j, 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

$$\sum_j Score(\langle C_j, c_j, V_j\rangle) = Score(DodgsonSum(\langle\langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle\rangle))$$

Theorem 1, Lemma 1 and Lemma 2, establish the Θ_2^p-hardness of a problem related to DodgsonWinner. We now define this decision problem:

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 such that $c \neq d$.

Decide: Is $Score(\langle C, c, V\rangle) \leq Score(\langle D, d, W\rangle)$?

Lemma 3. 2ER is Θ_2^p-hard.

We now need to make the results so far applicable to DodgsonWinner, so we need another merger lemma to merge two elections into a single election.

Lemma 4. There exists a polynomial-time computable function $Merge$ such that, for all Dodgson triples, $\langle C, c, V\rangle$ and $\langle D, d, W\rangle$ for which $c \neq d$ and both having an odd number of voters, there exist $\hat{C}$ and $\hat{V}$ such that

1. $Merge(\langle C, c, V\rangle, \langle D, d, W\rangle)$ is an instance of DogsonWinner,
2. $Merge(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle\hat{C}, c, \hat{V}\rangle$,
3. $Score(\langle\hat{C}, c, \hat{V}\rangle) = Score(\langle C, c, V\rangle) + 1$,
4. $Score(\langle\hat{C}, d, \hat{V}\rangle) = Score(\langle D, d, W\rangle) + 1$ and,
5. for each $e \in \hat{C} \setminus \{c, d\}$, $Score(\langle\hat{C}, c, \hat{V}\rangle) < Score(\langle\hat{C}, e, \hat{V}\rangle)$

3.3 Proof of select lemmas

Proof. Lemma 1 or 3DM $\leq_m^p$ DodgsonScore

[DK: Todo?] Our reduction differs from Bartholdi et al. [1989] in that our reduction has additional properties that are required by the lemma. We will reduce from ThreeDimensionalMatching to DodgsonScore:

Decision Problem. ThreeDimensionalMatching (3DM)

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

Decide: Does M contain a matching, i.e. a subset $M' \subseteq M$ such that $\|M'\| = \|W\|$ and no two elements of M' agree in any coordinate?

□

Proof. Lemma 3

We will reduce from Comp-SAT to 2ER. Let $\langle A, B\rangle$ be a Comp-SAT instance.

For each 3CNF formula $x \in A$ or B, reduce x into the corresponding 3DM instance x' and add x' to A' or B' if $x \in A$ or $x \in B$, respectively. In effect we are reducing Comp-SAT to an instance of Comp-3DM, $\langle A', B'\rangle$. It is easy to see that solving $\langle A', B'\rangle$ solves $\langle A, B\rangle$. One can also see that Comp-3DM is Θ_2^p-complete because it shares the structure where two lists of NP-hard problems are compared.

Now we perform a similar reduction from Comp-3DM to Comp-DodgsonScore. For each $x' \in A'$ or B', use the function in Lemma 1 to reduce x' into the corresponding DodgsonScore instance $\langle C, c, V\rangle$ and add $\langle C, c, V\rangle$ to A'' or B'' if $x' \in A'$ or $x' \in B'$, respectively. It is similarly easy to see that solving $\langle A'', B''\rangle$ solves $\langle A', B'\rangle$. If x' is a yes-instance of 3DM then by Lemma 1,

$$Score(f(x')) = Score(x'') = Score(\langle C, c, V\rangle) = k$$

where f is the function described in Lemma 1. Thus $\langle C, c, V\rangle$ is also a yes-instance of DodgsonScore. If x' is a no-instance of 3DM then $Score(\langle C, c, V\rangle) = k + 1$ and the corresponding triple $\langle C, c, V\rangle$ is a no-instance of DodgsonScore.

Now we reduce from COMP-DODGSONSCORE to 2ER using Lemma 2. First note that the direction of the inequality of the decision problem changes in this reduction by the nature of Lemma 1. Now to begin the reduction, we merge all the Dodgson elections in A'' and B'' into $\langle C, c, V\rangle$ and $\langle D, d, W\rangle$, respectively. This is done using the *DodgsonSum* function in Lemma 2, which we can use because Lemma 1 ensures that conditions of the Lemma are met by each election.

For example if $A'' = \langle C_1, c_1, V_1\rangle, \langle C_2, c_2, V_2\rangle, \ldots, \langle C_k, c_k, V_k\rangle$ then

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

and

$$\sum_j \mathit{Score}(\langle C_j, c_j, V_j\rangle) = \mathit{Score}(\langle C, c, V\rangle).$$

Now $\langle A'', B''\rangle$ is a yes-instance of COMP-DODGSONSCORE if by definition, the number of satisfied DODGSONSCORE instances in A'' being greater than that of B''. This again is equivalent to the sum of the Dodgson scores of A'' being less than that of B'' since we fix k to be the same for each of the reductions using Lemma 1. Let $\|A\|_{yes}$ be the number of yes-instances in a set of decision problems A. Using the reduction of $\langle A'', B''\rangle$ outlined above,

$$\begin{aligned}
&\langle A'', B''\rangle \text{ is a yes-instance of COMP-DODGSONSCORE} \\
&\iff \|A''\|_{yes} \geq \|B''\|_{yes} \\
&\iff \|\{x \in A'' | \mathit{Score}(x) \leq k\}\| \geq \|\{x \in B'' | \mathit{Score}(x) \leq k\}\| \\
&\iff \mathit{Score}(\langle C, c, V\rangle) \leq \mathit{Score}(\langle D, d, W\rangle) \qquad (1) \\
&\iff \langle\langle C, c, V\rangle, \langle D, d, W\rangle\rangle \text{ is a yes-instance of 2ER}
\end{aligned}$$

For line (1), note that by the reduction used from Lemma 1, the elections in $\langle A'', B''\rangle$ can have score of either k or $k + 1$. Hence we have shown a reduction from COMP-DODGSONSCORE to 2ER.

Combining the many-one reductions above, we have shown that:

$$\text{COMP-SAT} \leq_m^p \text{COMP-3DM} \leq_m^p \text{COMP-DODGSONSCORE} \leq_m^p \text{2ER}.$$

So by Theorem 1, 2ER is Θ_2^p-hard and the Lemma is proved. □

3.4 Proof that DODGSONWINNER is Θ_2^p-complete.

Proof. Theorem 2
By Theorem 3, DODGSONWINNER $\in \Theta_2^p$. We now show that 2ER $\leq_m^p$ DODGSONWINNER and so by Lemma 3, the theorem then follows.

We now describe a polynomial time function f for this reduction. Let s_o be some fixed string that is not in DODGSONWINNER. Then

$$f(x) = \begin{cases} \mathit{Merge}(x_1, x_2) & \text{if } x \in \text{2ER} \\ s_o & \text{if } x \notin \text{2ER} \end{cases}$$

Where Merge is the function defined in Lemma 4 and x_1 and x_2 are the two elections in the instance of 2ER. So $f(x)$ is an instance of DODGSONWINNER if and only if x is an instance of 2ER.

Now we show how $\mathit{Merge}(x_1, x_2)$, an instance of DODGSONWINNER, solves the corresponding instance x of 2ER. Let x be a pair of Dodgson triples, $\langle C, c, V\rangle$ and $\langle D, d, w\rangle$, where both have an odd number of voters and $d \neq c$ (so we can apply the Lemma). Then let $\mathit{Merge}(\langle C, c, V\rangle, \langle D, d, W\rangle) = \langle \hat{C}, c, \hat{V}\rangle$ be the corresponding instance of DODGSONWINNER.

First assume that $Score(\langle C, c, V\rangle) \leq Score(\langle D, d, W\rangle)$, or the answer to the 2ER decision problem is yes. Then by Lemma 4, properties 3 and 4, $Score(\langle \hat{C}, c, \hat{V}\rangle) \leq Score(\langle \hat{C}, d, \hat{V}\rangle)$. By property 5, $Score(\langle \hat{C}, c, \hat{V}\rangle) \leq Score(\langle \hat{C}, e, \hat{V}\rangle)$ for all $e \in \hat{C} \setminus \{c, d\}$. Hence c is a Dodgson winner of the election and the answer to the decision problem of this instance of DodgsonWinner is yes.

Finally, assume that $Score(\langle C, c, V\rangle) > Score(\langle D, d, W\rangle)$, or the answer to the 2ER decision problem is no. Then similarly, $Score(\langle \hat{C}, c, \hat{V}\rangle) > Score(\langle \hat{C}, d, \hat{V}\rangle)$ and so c is not a Dodgson winner of the election. Hence the answer to the decision problem of this instance of DodgsonWinner is no.

□

4 Practical Greedy Algorithm

In this section we present the `GreedyWinner` algorithm which runs in polynomial time and is self-knowingly correct with high probability when the number of voters is superquadratic in the number of candidates [Homan and Hemaspaandra, 2005]. This probability is over a uniform random draw of each vote i.e. $m!$ possibilities for each vote.

First we need to define what self-knowingly correct is.

Definition 6. For sets S and T and function $f : S \to T$, an algorithm $\mathcal{A} : S \to T \times \{$"definitely","maybe"$\}$ is self-knowingly correct for all f if, for all $s \in S$ and $t \in T$ whenever $\mathcal{A}$ on input s outputs $(t,$"definitely"$)$ it holds that $f(s) = t$.

The main theorems are as follows:

Theorem 4.
1. `GreedyScore` is self-knowingly correct for *Score*.
2. `GreedyWinner` is self-knowingly correct for DodgsonWinner.
3. `GreedyScore` and `GreedyWinner` both run in polynomial time.

Theorem 5.
1. `GreedyScore` is self-knowingly correct for *Score*.
2. `GreedyWinner` is self-knowingly correct for DodgsonWinner.
3. `GreedyScore` and `GreedyWinner`both run in polynomial time.

Theorem 6. For each $m, n \in \mathbb{N}^+$, the following hold. Let $C = \{1, \ldots, m\}$, $\|V\| = n$

1. For each $c \in C$,

$$\Pr[\texttt{GreedyScore}(\langle C, c, V\rangle) \neq (Score(\langle C, c, V\rangle), \text{“definitely”})] < 2(m-1)e^{\frac{-n}{8m^2}},$$

where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election (i.e. all $(m!)^n$ Dodgson elections having m candidates and n voters are equally likely to be chosen).

2.

$$\Pr[\exists c \in C | \texttt{GreedyWinner}(\langle C, c, V\rangle) \neq (DodgsonWinner(\langle C, c, V\rangle), \text{“definitely”})] < 2(m^2 - m)e^{\frac{-n}{8m^2}},$$

where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election.

Proof. First we need to establish under what conditions `GreedyScore`is self-knowingly correct.

Claim 1. For each $c \in C$, if for all $d \in C \setminus \{c\}$ it holds that

$$\|\{i \in [n] | c <_{v_i} d\}\| \leq \frac{2mn+n}{4m}, \text{ and} \tag{2}$$
$$\|\{i \in [n] | c \prec_{v_i} d\}\| \geq \frac{3n}{4m} \tag{3}$$

then `GreedyScore`$(\langle C, c, V\rangle) = (Score(\langle C, c, V\rangle), \text{“definitely”})$.

Proof: $\frac{2mn+n}{4m} = \frac{n}{2} + \frac{n}{4m}$ so if (2) holds then either c beats d and we are done. Otherwise, d beats c and flipping $\frac{n}{4m}$ votes where $c \prec d$ to $d \prec c$ would ensure that c beats d. If (3) holds then there are strictly more than $\frac{n}{4m}$ flippable votes so `GreedyScore`will be able to make enough swaps to ensure that c beats d.

Now we need to show what is the probability that these conditions are not met among all the possible pairs $c, d \in C, c \neq d$.

Claim 2. For each $c, d \in C$ such that $c \neq d$,

$$\Pr[(\|\{i \in [n] | c <_{v_i} d\}\| > \frac{2mn+n}{4m}) \vee (\{i \in [n] | c \prec_{v_i} d\} < \frac{3n}{4m})] < 2e^{\frac{-n}{8m^2}}.$$

Where the probability is taken over drawing uniformly at random an m-candidate, n-voter Dodgson election.

Proof sketch:

□

5 Experiments

6 Modified Greedy Algorithm

7 Conclusion

Algorithm 1 `GreedyScore`

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

```
for d ∈ C \ {c} do
    Deficit[d] ← 0
    Swaps[d] ← 0
end for
for each vote v ∈ V do
    i ← 1
    while v[i] ≠ c do
        d ← v[i]
        Deficit[d] ← Deficit[d] − 1
        i ← i + 1
    end while
    if i < length(v) then
        d ← v[i + 1]
        Swaps[d] ← Swaps[d] + 1
    end if
    for i ← i + 1 to length(v)  do
        d ← v[i]
        Deficit[d] ← Deficit[d] + 1
    end for
end for
confidence ← "definitely"
score ← 0
for d ∈ C \ {c} do
    if Deficit[d] ≥ 0 then
        score ← score + ⌊Deficit[d]/2⌋ + 1
        if Deficit[d] ≥ 2 · Swaps[d] then
            confidence ← "maybe"
            score ← score + 1
        end if
    end if
end for
```

Output: $(score,\ confidence)$

Algorithm 2 `GreedyWinner`

Input: A Dodgson triple $\langle C, c, V\rangle$ where we want to test whether c is a Dodgson winner in the election.

$(cscore,\ confidence) \leftarrow$ `GreedyScore`$(\langle C, c, V\rangle)$
$winner \leftarrow$ "yes"
for $d \in C \setminus \{c\}$ **do**
  $(dscore,\ confidence) \leftarrow$ `GreedyScore`$(\langle C, d, V\rangle)$
  if $dscore < cscore$ **then**
    $winner \leftarrow$ "no"
    if $dcon =$ "maybe" **then**
      $Confidence \leftarrow$ "maybe"
    end if
  end if
end for
Output: $(winner,\ confidence)$

References

Scott Aaronson, Greg Kuperberg, and Christopher Granade. The complexity zoo, 2005.

J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989. ISSN 01761714, 1432217X. URL `http://www.jstor.org/stable/41105913`.

Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

L. A. Hemachandra. The strong exponential hierarchy collapses. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 110–122, New York, NY, USA, 1987. Association for Computing Machinery. ISBN 0897912217. doi: 10.1145/28395.28408. URL `https://doi.org/10.1145/28395.28408`.

Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Exact analysis of dodgson elections: Lewis carroll's 1876 voting system is complete for parallel access to NP. *CoRR*, cs.CC/9907036, 1999. URL `https://arxiv.org/abs/cs/9907036`.

Christopher M. Homan and Lane A. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding dodgson-election winners. *CoRR*, abs/cs/0509061, 2005. URL `http://arxiv.org/abs/cs/0509061`.

Köbler, Johannes, Schöning, Uwe, and Wagner, Klaus W. The difference and truth-table hierarchies for np. *RAIRO-Theor. Inf. Appl.*, 21(4):419–435, 1987. doi: 10.1051/ita/1987210404191. URL `https://doi.org/10.1051/ita/1987210404191`.

Thomas Lukasiewicz and Enrico Malizia. A novel characterization of the complexity class thetakp based on counting and comparison. *Theoretical Computer Science*, 694: 21 – 33, 2017. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2017.06.023. URL `http://www.sciencedirect.com/science/article/pii/S0304397517305352`.

Christos H. Papadimitriou and Stathis K. Zachos. Two remarks on the power of counting. In Armin B. Cremers and Hans-Peter Kriegel, editors, *Theoretical Computer Science*, pages 269–275, Berlin, Heidelberg, 1982. Springer Berlin Heidelberg. ISBN 978-3-540-39421-1.

Klaus W Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.