Single Transferable Vote: Incomplete Knowledge and Communication Issues

Manel Ayadi, Nahla Ben Amor, Jérôme Lang, Dominik Peters

▶ **To cite this version:**

Manel Ayadi, Nahla Ben Amor, Jérôme Lang, Dominik Peters. Single Transferable Vote: Incomplete Knowledge and Communication Issues. 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 19), May 2019, Montreal QC, Canada. pp.1288-1296. hal-02307486

HAL Id: hal-02307486

https://hal.archives-ouvertes.fr/hal-02307486

Submitted on 15 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Single Transferable Vote: Incomplete Knowledge and Communication Issues

Manel Ayadi
LARODEC, ISG, Université de Tunis,
LAMSADE, Université Paris-Dauphine
manel.ayadi@hotmail.com

Nahla Ben Amor
LARODEC, ISG, Université de Tunis
nahla.benamor@gmx.fr

Jérôme Lang
CNRS, PSL, Université Paris-Dauphine
lang@lamsade.dauphine.fr

Dominik Peters
University of Oxford
dominik.peters@cs.ox.ac.uk

ABSTRACT

Single Transferable Vote (STV) is used in large political elections around the world. It is easy to understand and has desirable normative properties such as clone-proofness. However, voters need to report full rankings, which can make it less practical than plurality voting. We study ways to minimize the amount of communication required to use single-winner STV. In the first part of the paper, voters are assumed to report their top-k alternatives in a single shot. We empirically evaluate the extent to which STV with truncated ballots approximates STV with full information. We also study the computational complexity of the possible winner problem for top-k ballots. For $k = 1$, it can be solved in polynomial time, but is NP-complete when $k \geqslant 2$. In the second part, we consider interactive communication protocols for STV. Building on a protocol proposed by Conitzer and Sandholm (2005), we show how we can reduce the amount of communication required in practice. We then study empirically the average communication complexity of these protocols, based on randomly generated profiles, and on real-world election data. Our conclusion is that STV needs, in practice, much less information than in the worst case.

ACM Reference Format:
Manel Ayadi, Nahla Ben Amor, Jérôme Lang, and Dominik Peters. 2019. Single Transferable Vote: Incomplete Knowledge and Communication Issues. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

The world's largest elections that allow voters to *rank* candidates (and not just select the top choice) are probably the elections for the Australian House of Representatives. Each of 150 districts sends a representative, who is chosen by Single Transferable Vote (STV, also known as instant runoff voting, see Sec. 2 for a definition). Voters are asked to rank-order the candidates in their district. (Partial rankings are not allowed.) While it would be easy to compute the winner from electronic ballots, counting the paper ballots is a major undertaking; in the 2016 elections, it took more than a week. Since this makes for bad TV, the "two-party-preferred" heuristic is used to quickly estimate the winner: officials guess which two candidates are most likely to win, and the majority margin between these is counted and reported. In this paper, we think about better ways to handle STV elections, that can find the election winner quickly enough for TV, and that do not require voters to give full rankings.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

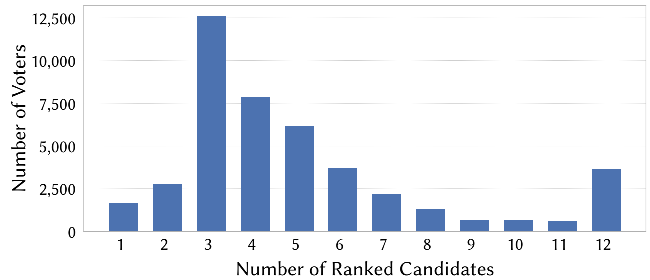

Figure 1: In the Dublin election, most of the 43,942 voters rank only 3 to 5 candidates out of 12.

In many legislatures, such as for the Irish presidential elections, voters are allowed to only submit a *partial* ranking: they rank a subset of the candidates, and leave the rest unranked. If all ranked candidates are eliminated by STV, the vote is then 'exhausted' and ignored during further counting. The freedom to give partial rankings is popular with voters. In the 2002 elections in Dublin, for which full ballot data is available, most voters chose to rank between 3 and 5 of the 12 candidates, with only 8% of voters submitting a full ranking (see Fig. 1). We ask: Are such partial ballots sufficient to correctly identify the STV winner? To do so, we sample random profiles of full rankings and truncate the rankings so that voters only rank their top-k choices. We then compute the probability that the STV winner of the top-k ballots is the same as the STV winner of the full ballots. Even for small k, we find that top-k ballots are enough to identify the correct winner quite frequently, especially for data taken from real elections. We also test empirically the sensitivity of STV to cloning when given truncated ballots.

As specified, STV ignores exhausted votes, which implicitly assumes that voters are indifferent between all unranked candidates. If we were to make fewer assumptions, we would want to compute all *possible winners* of STV, given top-k ballots. (A candidate is a possible winner if it wins for some way of completing the partial ballots into full rankings.) A problem with this proposal is that it is not clear how to efficiently decide which candidates are possible winners. Indeed, we show that the problem is NP-complete, even for top-2 ballots. On the other hand, we show that the problem is fixed-parameter tractable when there are few voters or few candidates. When given top-1 ballots (that is, we only know the plurality scores of the candidates), we give a simple characterization of the set of possible winners, which also yields a polynomial-time algorithm.

Suppose we are not happy to merely approximate the true STV winner. Another way to lower the workload for voters is to allow *interactive* communication protocols, that can ask voters for more information in an adaptive fashion. In 2005, Conitzer and Sandholm [6] studied such protocols for several common voting rules that take rankings as input. They found that, asymptotically and in the worst case, for many popular rules (such as Borda, Copeland, and ranked pairs) we cannot do better than to ask for the entire ranking outright. This can be annoying to voters (as it is to many Australians), and costly. In contrast, Conitzer and Sandholm [6] found a natural communication protocol for STV where the average voter only needs to name a *logarithmic* number of candidates. The Conitzer–Sandholm (CS) protocol begins by asking each voter for their top candidate. Based on this information, the candidate with the lowest plurality score is eliminated. The protocol then asks the supporters of this candidate for their next-most-preferred choice. The key insight is that only a small minority of voters can be supporters of the eliminated candidate (by choice of that candidate); indeed at most n/m supporters are possible, where n and m are the total number of voters and of remaining candidates. Repeating the elimination process, we see that in total we ask at most $\frac{n}{m} + \frac{n}{m-1} + \cdots + \frac{n}{2} \approx n \cdot \log m$ many questions during the elimination phase.[1]

Can this protocol be improved? We do not offer an alternative with a better worst-case guarantee,[2] but we propose a protocol that needs less communication in practice, and that never requires more communication than the CS protocol. Our protocol first asks voters for their initial top choices. Then, we identify a set of candidates that can safely be eliminated simultaneously. This avoids querying the same voter many times in a row, if that voter likes many niche candidates. This basic idea (slightly adapted to handle tie-breaking issues) allows for lower communication cost, on both random profiles and on real election data. On data from Australian elections, we find that many candidates are necessary losers, and can therefore immediately be eliminated, giving a more rigorous alternative to the current two-party-preferred heuristic.

Related work. The chapter by Boutilier and Rosenschein [4] provides background on work on incomplete information and communication in voting. Research on the communication complexity of STV was initiated by Conitzer and Sandholm [6]. Kalech et al. [13] consider a communication protocol where voters are repeatedly queried about their next preferred candidate; they do not study STV. Lu and Boutilier [15, 16] and Dery et al. [8] also study elicitation protocols based on top-k ballots. Adams et al. [21] consider the communication complexity of determining approximate winners for rules based on scores. Using truncated ballots as a way of reducing the amount of information has also been considered in [2, 19, 20, 22]. Filmus and Oren [10] study experimentally, using impartial culture, how often one can find the Borda or Copeland winner when only given top-k ballots.

Freeman *et al.* [11] gave an axiomatic characterization of STV as a social welfare function. They show that STV is the only rule in a family of iterative elimination rules that satisfies clone-proofness [23]. STV is NP-hard to manipulate, even for one voter [1], although this worst-case result does not seem to hold in the average case [14, 24]. Winner determination for the parallel-universe version of STV is NP-hard; however, recent work by Jiang et al. [12] shows that heuristic search algorithms perform very well in practice. Winner determination for the immediate-tie-breaking version of STV is in P; but it is P-complete and thus hard to parallelize efficiently [7].

Outline. Section 2 gives background on voting. Section 3 defines the k-truncated approximation of STV and evaluates it empirically. Section 4 studies the possible winner problem for STV given k-truncated ballots. Section 5 focuses on communication protocols.

2 PRELIMINARIES

An election is a triple $E = (N, A, P)$ where $N = \{1, \dots, n\}$ is a set of *voters*, A is a set of *candidates*, with $|A| = m$; and $P = (\succ_1, \dots, \succ_n)$ is a *(preference) profile*, which specifies a linear order $\succ_i$ over A for each voter $i \in N$. If $a \succ_i b$, then voter i is said to *strictly prefer* candidate a over b. A (resolute) *voting rule* is a function $f : E \mapsto A$ which for each election outputs a single winning candidate.

Given a prespecified linear order $\rhd$ over the candidates, called *tie-breaking priority*, the $STV^{\rhd}$ rule proceeds in (up to $m-1$) rounds. (We will usually write *STV* for short, leaving $\rhd$ implicit.) In each round, the candidate with the smallest number of voters ranking them first is eliminated (using tie-breaking if necessary),[3] and the votes who supported it now support their preferred candidate among those that remain. More formally, given a profile P and a candidate $x \in A$, we write $S(x)$ for the number of voters in P who rank x on top (the *plurality score* of x in P). STV is defined recursively as follows: If $|A| = 1$, then the unique candidate is the winner. Otherwise, select a candidate $x \in A$ with minimum plurality score $S(x)$ (taking x to be least-priority according to $\rhd$ if there are several plurality losers), construct the profile P' by removing x from all votes in P, and then recursively run STV on P'.

3 APPROXIMATING STV WITH TRUNCATED BALLOTS

In this section, we consider *one-shot protocols* where all input information needs to be gathered at the same time. In this model, if we wish to compute the *STV*-winner with *certainty*, we need to ask voters to report their entire preferences, i.e., to report their linear order. Let us relax the goal of certainty and instead aim to compute the *STV*-winner with high-enough probability. In exchange, we ask for less information: Voters report *top-k-ballots* for a fixed $k \geqslant 1$, i.e., they report a ranked list of their k most-preferred candidates.

[1]The existence of this protocol explains why counting STV elections on paper is feasible in the first place. We can view a paper ballot as an agent who is costly to 'communicate' with. First sort the papers into m physcial stacks according to top-ranked candidate. The Conitzer–Sandholm analysis shows that, to find the election winner, we only need to touch each ballot $O(\log m)$ times on average while redistributing.

[2]Conitzer and Sandholm [6] show that their protocol requires $O(n(\log m)^2)$ bits of communication, and they prove a lower bound of $\Omega(n \log m)$ using fooling sets.

[3]This way of breaking ties in STV is called *immediate tie-breaking*. The parallel universe version [5] of STV will not be considered here.

3.1 The rule STV_k

We first define a version of *STV* which takes top-*k* ballots as input. This generalization of *STV* is natural and quite popular. It is used, for example, in local elections in San Francisco (with $k = 3$). The analogous rule that allows submitting partial rankings of any length (not fixed to some k) is also widely used, for example in Ireland.

For each $1 \leqslant k \leqslant m$, we define STV_k as follows: Just like *STV*, in each round, we eliminate a candidate ranked first by the smallest number of voters (breaking ties using $\triangleright$ if necessary). If all the k candidates in some ballot have been eliminated, the vote is ignored in later rounds (and is called *exhausted*). We repeat this process until one candidate remains, who is the winner according to STV_k.

Example 1. *Let $A = \{a, b, c, d, e\}$ and consider the following 21 top-2 ballots: 6 votes $a \succ e$, 5 votes $d \succ e$, 4 votes $c \succ e$ and 6 votes $b \succ c$, with tie-breaking priority $a \triangleright b \triangleright c \triangleright d \triangleright e$. Under STV_2, e is eliminated first, then c. The votes $c \succ e$ are now exhausted. d is eliminated next; a and b are then tied, and a is the winner.*

Although STV_k can be seen as a voting rule on its own, we view it as an approximation of *STV*. In this context, we take STV_k to be a rule which takes profiles of full linear orders as input, truncates the preferences to become top-*k* ballots, and then proceeds as above. (Hence, STV_1 is the plurality rule, and STV_{m-1} and STV_m are just *STV*.) We will ask: How often do STV_k and *STV* declare the same candidate to be the winner?

For this we start by defining voting distributions on which our study will be based. We focus on the *Mallows* ϕ model [17] because of its flexibility and ability to represent a wide class of preferences. We will also discuss experiments using real-world data sets.

When a voting rule is defined via the maximization of a score, it makes sense to measure the quality of its approximations using score ratios. However, STV is not based on score maximization (for a discussion, see [5]). Thus, we measure the quality of approximation by the frequency with which the approximation outputs the true winner. Our main practical objective is to obtain, depending on the context, a value of k small enough to allow for painless communication of preferences, but large enough so that the probability of obtaining the true winner from STV_k is high.

3.2 Evaluation of the Accuracy of STV_k

To experimentally measure the probability that the STV_k-winner coincides with the true *STV*-winner, we repeatedly do the following:

(1) Generate a complete profile P with n voters and m candidates.
(2) For $k = 1$ to $m - 1$: compare $STV_k(P)$ to $STV(P)$.

The details of step (1) depend on whether we are in the random generation setting or the real world data setting. For the former, we sample a profile from a given distribution. For the latter, we randomly generate a profile, by selecting n votes uniformly at random from the collection of votes found in the data set. These two steps are then iterated a sufficient number of times so as to obtain meaningful results. Iterating this process a number of times allows us to evaluate the quality of STV_k for different values of k.

3.2.1 Mallows ϕ. The *Mallows ϕ-model* [17] is a probability distributions over rankings, parameterized by a reference ranking σ and a dispersion parameter $\phi \in [0, 1]$. For any ranking r, the probability of selecting r given σ and ϕ is

$$P(r; \sigma, \phi) = \frac{1}{Z}\phi^{d(r,\sigma)},$$

where d is the Kendall tau distance and

$$Z = \sum_{r'} \phi^{d(r,\sigma)} = 1 \cdot (1 + \phi) \cdot \dots \cdot \left(1 + \phi + \dots + \phi^{m-1}\right)$$

is a normalization constant. For small values of ϕ, the mass is concentrated around σ, while $\phi = 1$ gives the uniform distribution *impartial culture (IC)*, where all profiles are equiprobable.

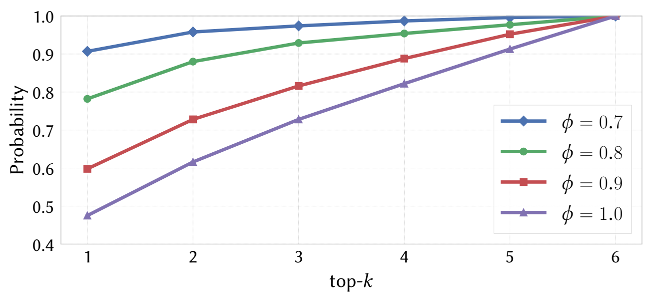

Figure 2: Probability that STV_k coincides with *STV*: Mallows ϕ model with $m = 7$ and $n = 100$.

We present simulation results with $m = 7$ and $n = 100$, and let ϕ vary. For each experiment, we draw 1000 random profiles. We simulate the elicitation of top-*k* ($k \in \{1 \dots 6\}$) preferences for $n = 100$ with $\phi \in \{0.7, 0.8, 0.9, 1\}$. Figure 2 shows the probabilities that STV_k correctly selects the *STV*-winner. For small $\phi = 0.7$, we see that the *STV*-winner is correctly selected with high probability when k is at least 3 or 4. For larger ϕ, the accuracy is much lower. For impartial culture ($\phi = 1$), the success rate is 82% (resp. 91%) when considering top-4 (resp. top-5) truncated ballots. In general, when $\phi < 1$, the accuracy of STV_k improves significantly if the number of voters is increased.

3.2.2 Real Election Data. We use data sets from *Preflib* [18]. Most of them contain some incomplete ballots. Since we need full ballots to run our experiments, we excluded all partial ballots. The data sets used are the following, with the number of voters and of alternatives in parentheses: *Sushi* (5000,10), *Dublin* (3662,12), *ERS* (43,10), *Glasgow City Council* (548,9), and *Debian* (327,7). Except for *Sushi*, all these data sets were obtained from elections where voters knew that STV would be used to count their votes. This makes it more likely that our conclusions will hold for future STV elections.

For each of these data sets, we find that STV_k converges quickly to the correct prediction with 100% accuracy for each fixed value of k, as n increases. Now, we are interested in predicting the result for small and large elections. We consider Dublin data (with $m = 12$ candidates), where we sample n^* voters among the n available votes ($n^* < n$). We start with $n^* = 10$ and increment the number of voters in steps of 10. In each experiment, 1000 random profiles are constructed with n^* voters; then we consider the top-k ballots obtained from these profiles, where $k \in \{1, 2, 3\}$, and we compute the probability of selecting the correct winner (the winner of the complete profile of the n^* sampled votes). Figure 3 shows results for Dublin data with small elections ($n^* \in \{10, \dots, 100\}$) and large elections ($n^* \in \{110, \dots, 1500\}$).

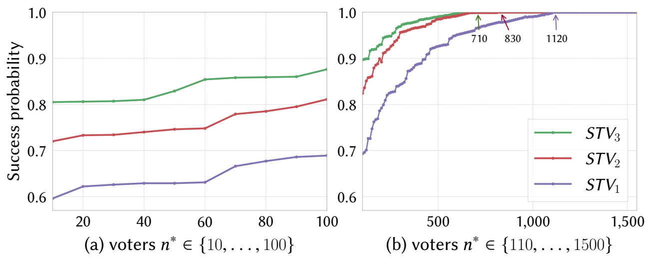

Figure 3: Probability that STV_k coincides with STV on Dublin data: varying $k \in \{1, 2, 3\}$ and n^* ($n^* < n$).

Our results suggest that predicting the correct winner with a small number of voters fails rather often when k is too small ($k \leqslant \frac{1}{4}m$). For instance, when $k = 3$ and $n^* = 100$, the correct winner is predicted only with frequency 87%. We also see that performance increases with the number of voters. Indeed, eliciting only one candidate for each voter ($k = 1$) is sufficient to predict the correct winner when $n^* \geqslant 1120$. Obviously, increasing the value of k leads to a decrease in the number of voters needed for correct winner selection with 100% accuracy: for instance, when $k = \frac{1}{6}m$ (resp. $k = \frac{1}{4}m$) over 12 candidates, $n^* \geqslant 830$ (resp. $n^* \geqslant 710$) are needed to always output the correct result.

3.3 STV_k and Resistance to Cloning

Tideman [23] introduced the notion of *independence of clones*, which requires a voting rule to be robust to the introduction of similar candidates. Notably, while most voting rules fail Tideman's condition,[4] the parallel-universe version of STV satisfies it, and the resolute version essentially satisfies it as well (see below). In this section, we study empirically to what extent STV_k stays clone-proof. Now, for $k = 1$, STV_1 coincides with plurality, which is highly sensitive to cloning, while STV_{m-1} is clone-proof. What happens in between?

Let us make the notion of clone-proofness formal. Given a candidate x, introduce a new set of candidates X' called *clones of* x. Let $A' = (A \setminus \{x\}) \cup X'$. A ranking $\succ'$ over A' is compatible with a ranking $\succ$ over A if all elements of X' are ranked contiguously in $\succ'$. A profile $P' = (\succ'_1, \dots, \succ'_n)$ over A' is compatible with a profile $P = (\succ_1, \dots, \succ_n)$ over A if for every i, $\succ'_i$ is compatible with $\succ_i$. A (possibly irresolute) voting rule is *clone-proof* if, given a profile P and a profile P' compatible with P, x is a winner in P if and only if one of the clones of x is a winner in P, and for any $y \neq x$, y is a winner in P if and only if it is a winner in P'.

The resolute version of STV with immediate tie-breaking is clone-proof, provided that tie-breaking is consistent with cloning: Let P be a profile over A, and P' a profile over $A' = A \setminus \{x\} \cup X'$ compatible with P. Take a tie-breaking relation $\rhd$ over A, and suppose $\rhd'$ over A' is compatible with $\rhd$. In this case, if $STV^{\rhd}(P) = x$ then $STV^{\rhd'}(P') \in X'$, and if $STV^{\rhd}(P) = y \neq x$ then $STV^{\rhd'}(P') = y$.

In order to evaluate the resistance of STV_k to cloning, we propose an empirical approach where we clone the winner and we measure experimentally the probability that cloning significantly changes the outcome. (We clone the winner because cloning a different candidate rarely changes the result.) For doing so we repeatedly generate a complete profile P (with n voters and m candidates), and then for each $k \in \{1, \dots m\}$ (a) we construct a profile P' obtained from P by cloning the winner (note that there are $m + 1$ candidates in P'), and (b) we compare $STV_k(P)$ to $STV_k(P')$. These steps are iterated a sufficient number of times to obtain meaningful results.

Example 2. *Let P contain 4 votes $a \succ c \succ b$, 3 votes $b \succ a \succ c$, and 2 votes $c \succ b \succ a$. The STV_2 ($= STV$) winner is b. Let us clone b to $\{b, b'\}$. A compatible profile P' is $\{4 : a \succ c \succ b' \succ b, 3 : b \succ b' \succ a \succ c, 2 : c \succ b' \succ b \succ a\}$. For $k = 2$, the k-truncated profile is $\{4 : a \succ c, 3 : b \succ b', 2 : c \succ b'\}$ and the STV_2 winner is a.*

3.3.1 Mallows ϕ. For each experiment we draw 1000 profiles. We present simulation results for small and large elections when $m = 5$ as we vary ϕ. We simulate the elicitation of top-k preferences where $k \in \{1, \dots, 5\}$ for $n \in \{30, 500\}$ with $\phi \in \{0.7, 0.8, 0.9, 1\}$. We compute the probability that, after we clone the STV_k winner, one of the clones still wins under STV_k (Figure 4). Unsurprisingly, resistance to cloning increases rapidly with k and decreases with ϕ. Also, it significantly increases with the number of voters.

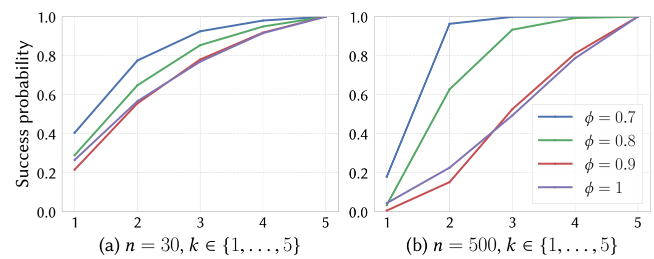

Figure 4: Resistance to cloning: Mallows ϕ model.

3.3.2 Real Data. We test the resistance of STV_k to cloning using Dublin data with samples of n^* voters among n ($n^* < n$) where $n^* = \{100, 1000\}$ then we clone the STV_k winner (Figure 5). In each experiment 1000 random profiles are constructed with n^* voters.

Consistently with the above experiments, STV_k often fails clone-proofness for small k. Again, resistance to cloning increases rapidly with k, even more rapidly than with randomly generated profiles.

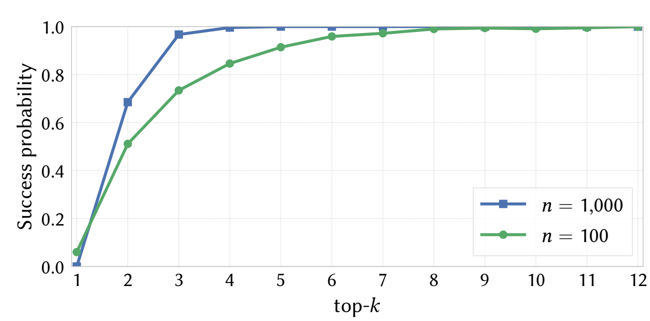

Figure 5: Resistance to cloning: real data.

[4]Notable exceptions are *Ranked Pairs* and *Schulze*. *Approval voting* and *range voting* can also be seen as clone-proof, though they do not use rankings.

4 POSSIBLE WINNERS WITH TOP-k BALLOTS

When votes are only partially known, the correct winning candidate can in general not be determined. We can express our incomplete information using the concept of *possible winners.*

Formally, a *partial profile* is a collection $(R_1, \dots, R_n)$ of partial orders over the candidates A. A *completion* of a partial profile is a profile $(\succ_1, \dots, \succ_n)$ of linear orders such that $\succ_i$ extends R_i for each $i = 1, \dots, n$. Then, given a partial profile, a candidate c is a *possible winner* of *STV* if there exists a completion of the partial profile such that c is the *STV*-winner of that completion.

It is known that it is NP-complete to decide whether a given candidate is a possible winner for *STV* in a given partial profile. This follows immediately from the NP-completeness of constructive manipulation for *STV* [1] which is equivalent to the possible winner problem with a profile where $n-1$ votes are fully specified, but we do not have any information about the last vote. While this case is known to be hard, the complexity of the possible winner problem for *STV* has not been studied for other 'shapes' of partial profiles.

A particularly intuitive way of expressing partial preferences is to use *truncated ballots.* Given $k \leqslant m$, a *top-k ballot* is a linear order of k among the m candidates. A *top-k profile* is a collection of n top-k ballots. In this section, we study the possible winner problem for *STV* with top-k ballots for fixed $k \geqslant 1$. We show that the problem is polynomial-time computable if $k = 1$, but that it is NP-complete for each fixed $k \geqslant 2$. However, the problem is fixed-parameter tractable for profiles that have few voters or few candidates.

4.1 Possible Winners with Top-1 ballots

The next result gives a closed-form characterization of possible winners for STV. This condition (without tie-breaking) was also considered (without proof) in [12] as a reduction technique for computing parallel universe STV cowinners using search algorithms.

Proposition 1. *Let P be a top-1 profile over A. Let $S(x)$ be the plurality score of x in P, that is, the number of votes in P ranking x on top. Relabel candidates so that $S(x_1) \leqslant S(x_2) \leqslant \cdots \leqslant S(x_m)$, and such that if $S(x_i) = S(x_{i+1})$ then $x_{i+1} \rhd x_i$. Then x_i is a possible winner for $STV_{\rhd}$ if there is no $j > i$ such that either $S(x_j) > \sum_{s=1}^{j-1} S(x_s)$, or $S(x_j) = \sum_{s=1}^{j-1} S(x_s)$ and $j \rhd i$.*

Proof. Assume there is a $j > i$ such that $S(x_j) > \sum_{s=1}^{j-1} S(x_s)$, or $S(x_j) = \sum_{s=1}^{j-1} S(x_s)$ and $j \rhd i$. Then, there is no way for x_i to avoid eliminating before x_j, because it can only benefit from the transfers from the votes that initially support x_s for $s < j$.

For the converse, consider any completion Q of P where x_i is ranked second in every vote whose top is not x_i. Assume there is a step where x_i is eliminated, and let j be the smallest index such that $j \neq i$ and such that x_j has not been eliminated yet at that stage. The votes supporting x_j are only those $S(x_j)$ that supported it initially. Since all candidates x_s with $s < j$ and $s \neq i$ have been eliminated, their votes have all been transferred to x_i, and thus x_i has exactly $\sum_{s=1}^{j-1} S(x_s)$ supporting votes. Since x_i is eliminated before x_j, either $\sum_{s=1}^{j-1} S(x_s) < S(x_j)$, or $\sum_{s=1}^{j-1} S(x_s) = S(x_j)$ and $j \rhd i$. □

Example 3. *Let P be such that $S(x_1) = 1$, $S(x_2) = 2$, $S(x_3) = 4$, $S(x_4) = 6$ and $S(x_5) = 12$. The possible winners are x_3, x_4, x_5.*

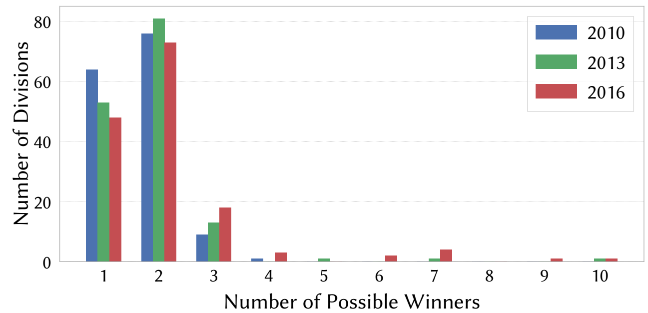

Figure 6: In most divisions in recent elections for the Australian House of Representatives, there were only one or two possible winners, given plurality scores.

If a candidate is not a possible winner, it must be a necessary loser. It turns out that in real STV elections, we can identify many necessary losers, even if we only know the plurality scores. Figure 6 shows the number of possible winners in elections for the Australian House of Representatives, for which plurality scores are publicly available for each of the 150 districts ('divisions') in each election year. We can see that in most divisions, there is only 1 possible winner (because they received a majority of first-place votes), or there are 2 possible winners (usually from the two major parties). It is very uncommon for there to be 3 or more possible winners.

Using Proposition 1, one can find the possible winners for STV given top-1 ballots in polynomial time. The complementary notion of *necessary* winner (i.e., a candidate who wins for all completions of the partial profile), is easily characterized: candidate x is the necessary winner iff it is top-ranked by a majority of voters (assuming n is odd; the characterization for even n depends on tie-breaking).

4.2 Possible Winners with Top-2 ballots

Now we show that the positive result for top-1 ballots does not extend to larger values of k: the possible STV-winner problem for top-k ballots is NP-complete, even for $k = 2$.

Proposition 2. *The possible STV-winner problem is NP-complete given top-2 truncated ballots.*

Proof. We reduce from 3-SAT, restricted to formulas in which each clause has exactly 3 literals, and each literal occurs exactly twice. Let φ be such a 3CNF formula, with clauses $C = \{c_1, \dots, c_m\}$ and variable set $X = \{x_1, \dots, x_n\}$. Let $L = X \cup \overline{X}$ be the set of literals. Let $\varepsilon := 1/1000nm$. We construct a preference profile with alternative set $A = \{c, w\} \cup C \cup L$. In specifying the profile, we use fractional voters to make the argument easier to follow, but we can translate this to integer numbers by multiplying all multiplicities by $1/\varepsilon^2$. Introduce the following voters:

$100 \times$	$c \succ w \succ \cdots$	
$99 \times$	$w \succ c \succ \cdots$	
$99 - \varepsilon \times$	$c_j \succ w \succ \cdots$	for each clause $c_j \in C$
$60 \times$	$\ell \succ \overline{\ell} \succ \cdots$	for each literal $\ell \in L$
$2\varepsilon \times$	$\ell \succ c_j \succ \cdots$	for each occurrence of $\ell \in L$ in $c_j \in C$
$\varepsilon^2 \times$	$\cdot \succ \cdot \succ \cdots$	for each variable $x \in X$

The dots in the last type of voter indicate dummy alternatives that only appear in those positions (and will immediately be eliminated). The value ε is chosen so that the total weight of order-ε voters is less than 1, and the total weight of order-ε^2 voters is less than ε. We wish to know if alternative c is a possible STV winner.

Suppose there is a satisfying assignment to the variables; for each variable $x_i \in X$, let ℓ_i be the literal that is set true. Consider any completion of the above profile such that

$$\begin{array}{rll} 100 \times & c \succ w \succ \cdots & \\ 99 \times & w \succ c \succ \cdots & \\ 99-\varepsilon \times & c_j \succ w \succ c \succ \cdots & \text{for each clause } c_j \in C \\ 60 \times & \ell \succ \overline{\ell} \succ c \succ \cdots & \text{for each literal } \ell \in L \\ 2\varepsilon \times & \ell \succ c_j \succ c \succ \cdots & \text{for each occurrence of } \ell \in L \text{ in } c_j \in C \\ \varepsilon^2 \times & \cdot \succ \cdot \succ \overline{\ell}_i \succ \cdots & \text{for each variable } x \in X \end{array}$$

After eliminating nameless dummy alternatives, true literals ℓ_i have the lowest score $60+4\varepsilon$ (false literals have score $60+4\varepsilon+\varepsilon^2$). So in the next n rounds, the true literals are eliminated. Since the assignment satisfies all clauses, the scores of the remaining alternatives are:

$$c: 100 \quad w: 99 \quad c_j: \geqslant 99+\varepsilon \quad \overline{\ell_i}: >120.$$

Thus, w is eliminated next, and its votes are transferred to c, giving it a score of 199. Then all the c_j are eliminated, again transferring votes to c. Finally, false literals $\overline{\ell_i}$ are eliminated, transferring (all but perhaps ε^2 many) votes to c. Hence, c wins.

For the other direction, suppose the formula is unsatisfiable. We show that c is not an STV winner for any completion of the profile. At first, the unnamed dummy alternatives are eliminated. In each of the next n rounds, the alternative with the lowest score (between 60 and 61) is a literal. So in these rounds, a selection of n literals are eliminated. Note that once we eliminate literal ℓ, the alternative $\overline{\ell}$ has plurality score greater than 120, and so $\overline{\ell}$ will not be eliminated in these n rounds. Thus, the n eliminated literals describe a valid assignment to the variables which sets all eliminated literals to *true*.

Because the formula is unsatisfiable, there is a clause c_j which is not satisfied by this assignment. Thus, c_j did not have any order-ε votes transferred to it, so has score < 99. So in the next round c_j (or another unsatisfied clause) is eliminated, and its votes are transferred to w, so that w now has a score of more than 197. In the next $m-1$ steps, either c is eliminated (and we are done), or all other clause candidates are eliminated, since they all have scores of < 101, while the remaining literals have a score of at least 120. Each time a clause is eliminated, the score of w rises. After all clauses are eliminated, c still has a score between 100 and 101, so c is eliminated next. Since w still remains, c is not an STV winner. □

It is easy to extend the above reduction to top-k ballots for fixed $k \geqslant 3$, by introducing extra dummy candidates in second position, which will immediately be eliminated.

While the possible winner problem is NP-complete, it is fixed-parameter tractable if there are few voters, or few candidates.

Proposition 3. *The possible $\text{STV}_{\triangleright}$-winner problem for top-k ballots is in FPT with respect to the number of voters n, and is in FPT with respect to the number of candidates m.*

Proof. *Number of voters n.* Note that at most n different candidates can appear in top position, and so STV, run on any completion of the profile, will immediately eliminate all candidates that are never in top position. So we may assume that there are at most n candidates overall. This allows us to iterate over all possible completions of the profile (there are at most $(n!)^n$ many) in FPT time, and then run STV on the completion to obtain a possible winner.

Number of candidates m. For this, we formulate the problem as in integer linear program with a number of variables bounded as a function of m. By Lenstra's theorem, it follows that the problem is in FPT. We omit the details due to space constraints. As a sketch, the program contains, for each of the $m!$ possible preference rankings $\succ$, a variable $m_{\succ}$ indicating how many votes are completed to this ranking. Constraints ensure that these variables describe a valid completion of the input profile. Then, we use additional variables and constraints to compute the STV winner of the completion, in a way similar to the STV-ILP presented in [12]. □

5 COMMUNICATION PROTOCOLS FOR STV

We now take a different path: we consider the determination of the STV winner for the complete profile, and consider *interactive* protocols where voters may report their preferences incrementally, when the central authority asks them to do so.

5.1 Conitzer and Sandholm's Protocol

Conitzer and Sandholm [6] studied the communication complexity of several voting rules, by giving specific protocols and by proving lower bounds obtained via fooling sets. For the case of STV, they showed a lower bound on the communication complexity of $\Omega(n \log m)$, which is the cost of communicating every voter's top choice. They were able to match this lower bound up to a log factor using the following protocol for STV, which we call P_1.

Protocol 1: P_1

```
for each voter i ∈ N do
    ask i to send the name of her top candidate
repeat
    d ← candidate ranked first by the fewest voters (breaking ties)
    Remove d from the set of available candidates
    for each voter i ∈ N do
        if the top candidate of i was d then
            ask i to send the name of her next preferred candidate
until there exists a candidate c ranked first by a majority of voters
return c
```

Example 4. *Let $A = \{a, b, c, d\}$ and $P = \{3 : a \succ c \succ d \succ b, 3 : c \succ a \succ b \succ d, 1 : d \succ b \succ a \succ c, 1 : b \succ d \succ c \succ a\}$ with tie-breaking priority $a \triangleright b \triangleright c \triangleright d$. We run P_1. In the first round, d is eliminated. We ask the voter who supports d to name her next preferred candidate. We get $P = \{3 : a, 3 : c, 2 : b\}$. Next, b is eliminated. We ask the two voters supporting b for their next preferred candidate. We get $P = \{4 : a, 4 : c\}$. By tie-breaking, a wins.*

Conitzer and Sandholm [6] show that P_1 requires communication of at most $O(n(\log m)^2)$ bits. To see this, note that at a step where k candidates remain, at most $\frac{n}{k}$ voters rank the eliminated candidate first. Thus, the number of times we need to ask voters for their new favorite is at most $\frac{n}{m} + \frac{n}{m-1} + \ldots + n$, which is bounded by $n \log m$. We can actually say something more precise: the worst case occurs when, in each step, all candidates are tied for elimination; in this case, when k candidates remain, exactly $\frac{m}{k}$ voters will send $\log k$ bits, and this goes on until $k = 1$. This gives an exact worst case cost of $n\left(\log m + \sum_{k=1}^{m-1} \frac{\log k}{k+1}\right)$ bits. In the subsequent discussion, we will not focus on the number of bits transmitted, and instead will count how often voters have to report their favorite remaining candidate. We refer to this as the *number of questions* asked. It is easy to see that, in the worst-case, for fixed n and m, the number of questions asked by protocol P_1 is $P_{Worst} = n \cdot \left(1 + \sum_{k=1}^{m-1} \frac{1}{(k+1)}\right)$.

5.2 An Improved Protocol

At each step in the execution of the protocol, the central authority has partial knowledge of the votes. Therefore, it makes sense to identify those candidates that can still win (the possible winners) and those that cannot (the necessary losers). This is especially useful when interaction with the voters takes time: assume that the vote is about a meeting date; the execution of the protocol can take several days (due to some voters reacting slowly to their emails). If at some point in the execution of the protocol, we know that for sure the meeting will not be on November 22 nor November 24, this is useful information for voters, who can plan something else on these two days, and for the central authority, which does not have to pre-book a room for these days.

We start by noticing that at each step of the protocol P_1, the information known by the protocol is essentially a top-1 profile. Therefore, Proposition 1 is applicable and we can calculate, at each step of the protocol, the remaining possible winners and necessary losers. Knowing the possible winners (and the necessary losers) at each step of the protocol is useful information, but it turns out that eliminating a candidate as soon as it becomes a necessary loser can change the final outcome:

Example 5. *Let us consider* $P = \{5 : c \succ a \succ d \succ e \succ b, 1 : a \succ e \succ b \succ d \succ c, 2 : e \succ b \succ d \succ a \succ c, 2 : b \succ a \succ e \succ c \succ d\}$*, and let the tie-breaking priority be* $a \rhd b \rhd c \rhd d \rhd e$*. The winner for* P *is* c*. The necessary losers given top-1 ballots are* a*,* d *and* e*. If we eliminate those three candidates, the winner is* b*.*

The source of the paradox lies in the tie between the initial score of c and the cumulative score of a, b, d, and e. However, we now identify a subset of necessary losers that can always be eliminated without changing the winner.

Let us rename the candidates so that they are ranked by increasing order of plurality score, then by tie-breaking priority: for each i, either $S(x_i) < S(x_{i+1})$, or $S(x_i) = S(x_{i+1})$ and x_{i+1} has priority over x_i. We say that x_i is *dominant* if either $S(x_i) > \sum_{j=1}^{i-1} S(x_j)$, or $S(x_i) = \sum_{j=1}^{i-1} S(x_j)$ and x_i has priority over x_j, where j is the largest index smaller than i such that x_j is a dominant candidate. Note that to check whether x_i is dominant, we need to iterate over all candidates x_j, $j < i$. We say that x_j is a *strong necessary loser* if there is an index $i > j$ such that x_i is dominant. By Proposition 1, a strong necessary loser is a necessary loser, but the converse is not always true: in Example 5, only d and a are strong necessary losers.

Proposition 4. *The removal of strong necessary losers does not change the winner.*

Proof. Assume x_i is the largest dominant candidate in the sequence ($x_1, \ldots, x_{i-1}$ are thus the strong necessary losers). Then no matter in which order the candidates $x_1, \ldots, x_{i-1}$ are eliminated, they will all be eliminated before all candidates in $\{x_i, \ldots, x_m\}$. Therefore, after $i-1$ elimination steps, the currently eliminated candidates are exactly $x_1, \ldots, x_{i-1}$. Eliminating them directly allows to "jump" over $i-1$ elimination steps, and the rest of the process continues exactly as if this jump had not been performed. □

Protocol P_1 can be improved by checking at each step if there are strong necessary losers, and if so, eliminate them in addition to the current loser, and then send a query to all voters whose current top candidate has just been eliminated. A further improvement is possible by querying one voter at a time: it might be possible to rule out further candidates without knowing every voters' current top choice. To do this we need to generalize the notion of dominant candidate and strong necessary loser to *incomplete plurality profiles*, where the top candidate of some voters may be unknown. An *incomplete plurality profile* is given by the numbers $(s_\emptyset, s_{x_1}, \ldots, s_{x_m})$, where $s_\emptyset$ is the number of voters for whom we do not know what their top candidate is, and s_{x_i} are the number of voters for whom we know that x_i is their top candidate.

Assume again that candidates are renamed so that they are ranked by increasing order of plurality score, then by tie-breaking priority. A candidate x_i is a *safe necessary loser* (SNL) if there is $j > i$ such that $s_{x_j} > s_\emptyset + s_{x_1} + \cdots + s_{x_{j-1}}$. Clearly, if x_i is a safe necessary loser for an incomplete plurality profile then it is a strong necessary loser for every of its completions into a plurality profile, which implies that eliminating an SNL candidate cannot change the winner. This leads us to define the following protocol:

Protocol 2: P_2

```
Query every voters' top candidate
Let (s_∅, s_x1, ..., s_xm) be the resulting plurality profile, with
  s_∅ = 0
repeat
    for each safe necessary loser x_i do
        Remove x_i
        s_∅ ← s_∅ + s_xi
    Select an empty voter and query their new top alternative
    Let x_j be this new top alternative
    s_xj ← s_xj + 1; s_∅ ← s_∅ − 1
until the set of possible winners is a singleton
```

The protocol terminates because at each step, either a candidate is removed, or $s_\emptyset$ is decreased by 1. If $s_\emptyset$ queries are made successively without any candidate being removed, then $s_\emptyset$ reaches 0 and then there is at least one SNL (the current plurality loser with lowest priority). The winner returned at the end of the protocol is the STV winner because removing SNLs does not influence the winner. It can be proven that P_2 is as least as cheap as P_1, in the sense that no voter is queried more often with P_2 than with P_1.

Example 5.1. Let the candidates be a, b, c, d, e. After the first step, the plurality scores are $(s_a : 1, s_b : 1, s_c : 3, s_d : 4, s_e : 8)$. a and b are

SNL and are removed. The resulting incomplete plurality profile is $(s_\emptyset : 2, c : 3, d : 4, e : 8)$. We query a voter who votes for d, giving $(s_\emptyset : 1, s_c : 3, s_d : 5, s_e : 8)$. c is now a SNL and is removed, giving $(s_\emptyset : 4, s_d : 5, s_e : 8)$. We query a voter who votes for e, resulting in $(s_\emptyset : 3, s_d : 5, s_e : 9)$. d is now an SNL, and the winner is e.

5.3 Evaluation of the Protocols

This section evaluates the average communication complexity of P_1 and P_2. We discuss experiments using the Mallows ϕ model and real data. Our objective is to determine the average communication complexity, in terms of the number of questions voters need to answer on average. Note that, for plurality elections, this number is 1, while for (say) Borda elections, this number is m in the worst-case. We find that STV only asks 2 or 3 questions for each voter when $m = 7$, or in the Dublin election data.

5.3.1 Mallows ϕ. For each experiment, we draw 1000 random profiles. In the first set of experiments, we present simulation results with $m = 7$ and $n = 100$, and let ϕ vary. We count the number of questions asked by P_1 and P_2 with $\phi \in \{0.7, 0.8, 0.9, 1\}$. Figure 7 shows the average communication cost of P_1, P_2 and compares it to P_{Worst}, the worst-case cost of P_1.

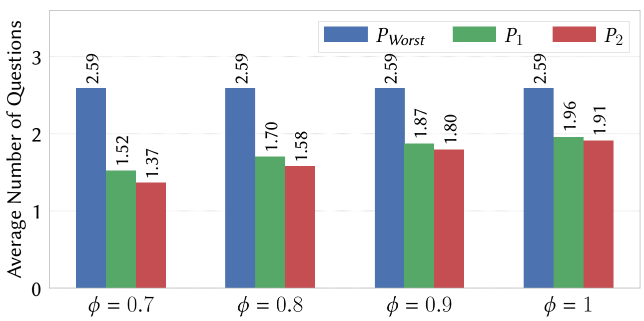

Figure 7: Average communication cost, Mallows.

Results suggest that in practice, that both P_1 and P_2 ask fewer questions than in the worst case (P_{Worst}). P_2 asks fewer questions than P_1 for all parameter values, but the difference is more significant for lower ϕ. When $\phi = 1$, more information is needed from voters under both P_1 and P_2, and the savings of P_2 are smaller.

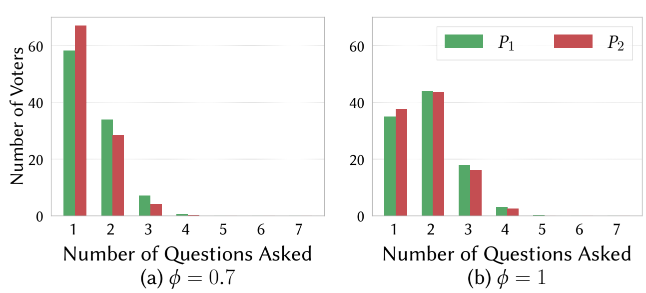

Figure 8: Average number of voters who were asked k questions, Mallows

To better understand the difference in the behavior of P_1 and P_2, we also plot, for each k, how many voters needed to answer k questions, see Fig. 8. The numbers shown are averages over profiles with $n = 100$, $m = 7$ and $\phi \in \{0.7, 1\}$, taken over 1000 samples. The results show that, compared to P_1, P_2 often only needs to ask a voter for their top alternative, and then never asks another question.

5.3.2 Real Data. Figure 9 shows the average number of questions voters are asked by the protocols on real data sets. P_2 performs better than P_1 by asking 5–10% fewer questions. Both protocols are much better than the worst-case analysis suggests, by about 40%.

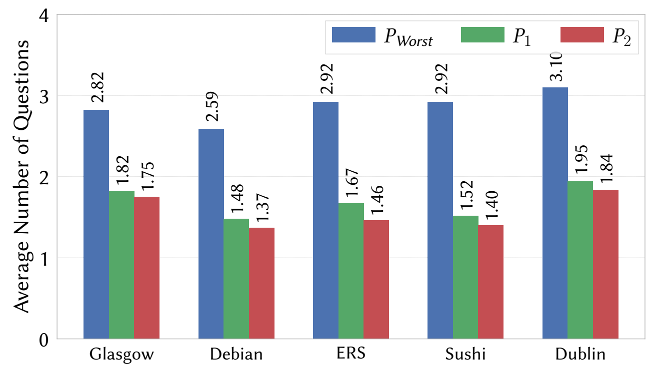

Figure 9: Average communication cost, real data

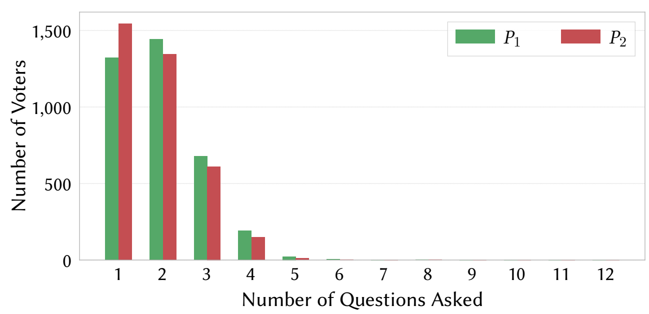

Figure 10: Number of voters who were asked k questions, Dublin data.

Figure 10 shows how many voters in Dublin were asked k questions under P_1 and P_2. Again, P_2 outperforms P_1. Results suggest that almost half of the voters are asked to submit only their top preferred candidate when using P_2. In P_1, it is more common to be asked to submit two preferences.

6 CONCLUSION

Our results show that STV has low communication cost in practice, either through using the STV$_k$-rules of Section 3 or by using the protocols of Section 4. Thus, the extra cost of eliciting full rankings should not be an argument against replacing, say, plurality voting by STV. In fact, the very low communication cost of STV might be a reason to prefer it to other traditional voting rules, such as Borda.

A future direction is to consider STV in another incomplete information context, namely *vote streams*, where voters arrive one at a time [3, 9]. The key practical question is to decide when we have enough information to eliminate one more candidate, so that the next voters will have less information to communicate.

REFERENCES

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

[2] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. 2012. Campaigns for lazy voters: truncated ballots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 577–584.

[3] A. Bhattacharyya and P. Dey. 2015. Fishing out Winners from Vote Streams. *Electronic Colloquium on Computational Complexity (ECCC)* 22 (2015), 135.

[4] C. Boutilier and J. Rosenschein. 2016. Incomplete Information and Communication in Voting. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). Cambridge University Press, Chapter 10.

[5] V. Conitzer, M. Rognlie, and L. Xia. 2009. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 109–115.

[6] V. Conitzer and T. Sandholm. 2005. Communication Complexity of Common Voting Rules. In *Proceedings of the 6th ACM Conference on Electronic Commerce (ACM-EC)*. ACM Press, 78–87.

[7] T. Csar, M. Lackner, R. Pichler, and E. Sallinger. 2017. Winner Determination in Huge Elections with MapReduce. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*. 451–458.

[8] L. N. Dery, M. Kalech, L. Rokach, and B. Shapira. 2014. Reaching a joint decision with minimal elicitation of voter preferences. *Information Sciences* 278, 466–487.

[9] P. Dey, N. Talmon, and O. van Handel. 2017. Proportional representation in vote streams. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 15–23.

[10] Y. Filmus and J. Oren. 2014. Efficient voting via the top-k elicitation scheme: a probabilistic approach. In *Proceedings of the 15th ACM Conference on Economics and Computation (ACM-EC)*. ACM Press, 295–312.

[11] R. Freeman, M. Brill, and V. Conitzer. 2014. On the axiomatic characterization of runoff voting rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 675–681.

[12] C. Jiang, S. Sikdar, J. Wang, L. Xia, and Z. Zhao. 2017. Practical Algorithms for Computing STV and Other Multi-Round Voting Rules. In *EXPLORE-2017: The 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice*.

[13] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman. 2011. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems* 22, 1, 151–182.

[14] J.-F. Laslier. 2016. Heuristic voting under the Alternative Vote: the efficiency of 'sour grapes' behavior. *Homo Oeconomicus* 33, 1 (2016), 57–76.

[15] T. Lu and C. Boutilier. 2011. Robust Approximation and Incremental Elicitation in Voting Protocols. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 287–293.

[16] T. Lu and C. Boutilier. 2011. Vote Elicitation with Probabilistic Preference Models: Empirical Estimation and Cost Tradeoffs. In *Proceedings of the 2nd International Conference on Algorithmic Decision Theory (ADT)*. Springer, 135–149.

[17] C. L. Mallows. 1957. Non-null ranking models. I. *Biometrika* (1957), 114–130.

[18] N. Mattei and T. Walsh. 2013. PrefLib: A Library for Preference Data. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT) (Lecture Notes in Computer Science (LNCS))*, Vol. 8176. Springer-Verlag, 259–270.

[19] L. Naamani-Dery, M. Kalech, L. Rokach, and B. Shapira. 2016. Reducing preference elicitation in group decision making. *Expert Systems with Applications* 61 (2016), 246–261.

[20] J. Oren, Y. Filmus, and C. Boutilier. 2015. Efficient Vote Elicitation under Candidate Uncertainty. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, 309–316.

[21] T. C. Service and J. A. Adams. 2012. Communication complexity of approximating voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 593–602.

[22] P. Skowron, P. Faliszewski, and A. Slinko. 2015. Achieving fully proportional representation: Approximability results. *Artificial Intelligence* 222 (2015), 67–103.

[23] T. N. Tideman. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare* 4, 3 (1987), 185–206.

[24] T. Walsh. 2010. An Empirical Study of the Manipulability of Single Transferable Voting. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI)*. 257–262.