# The Communication Burden of Single Transferable Vote, in Practice

Manel Ayadi[1,2](✉), Nahla Ben Amor[1](✉), and Jérôme Lang[2](✉)

[1] LARODEC, Institut Supérieur de Gestion, Université de Tunis, Tunis, Tunisie
manel.ayadi@hotmail.com, nahla.benamor@gmx.fr
[2] CNRS, LAMSADE, Université Paris-Dauphine, Paris, France
lang@lamsade.dauphine.fr

**Abstract.** We study single-winner STV from the point of view of communication. First, we assume that voters give, in a single shot, their top-$k$ alternatives; we define a version of STV that works for such votes, and we evaluate empirically the extent to which it approximates the standard STV rule. Second, we evaluate empirically the communication cost of the protocol for STV defined by Conitzer and Sandholm (2005) and some of its improvements.

## 1 Introduction

*Single transferable vote* (STV)[1] is an appealing voting rule: it is relatively easy to understand, it is not easy to manipulate, and it enjoys a very important normative property: clone-proofness. It is used in single-winner and multi-winner political elections in several countries. It fails to satisfy a number of other important properties,but in many contexts, being sensitive to cloning may be worse than the failure of these other properties. On the other hand, when compared to other rules that are widely used in practice (such as plurality, $k$-approval for small $k$, approval, or plurality with runoff), STV suffers from a significant drawback: its direct implementation requires an important amount of information to be communicated from the voters, because its input consists of a collection of complete rankings over candidates. Our aim is to get a more accurate idea of the precise amount of information that we need from the voters to compute or to approximate STV. We successively consider two contexts.

First, we assume that voters communicate, in a single shot, their top-$k$ candidates, and we use an approximation of STV which needs only these top-$k$ ballots as input.

Second, we consider interactive communication protocols, to be run between the central authority and the voters until the outcome of the vote is eventually determined. We study empirically the average communication complexity of the protocol defined by Conitzer and Sandholm [2], and of an improved variant of it.

This is a short version of our long paper submitted to SAGT 2018 (this submitted long version is available at https://goo.gl/Knd59d).

[1] For single-winner elections, STV is often called *instant runoff voting*.


X. Deng (Ed.): SAGT 2018, LNCS 11059, pp. 251–255, 2018.
https://doi.org/10.1007/978-3-319-99660-8_23

## 2 Approximating STV with Truncated Ballots

An election is a triple $E = (N, A, P)$ where $N = \{1, ..., n\}$ is the set of *voters*, $A$ is the set of *candidates*, with $|A| = m$; and $P = (\succ_1, ..., \succ_n)$ is the *(preference) profile*, where for each $i$, $\succ_i$. A resolute voting rule maps any election to a single winner.

Given a prespecified linear order $\rhd$ over candidates, called *tie-breaking priority*, the $STV^{\rhd}$ rule proceeds in (up to $m-1$) rounds. (For brevity notation we will simply write $STV$, leaving $\rhd$ implicit.) In each round, the candidate with the smallest number of voters ranking them first is eliminated (using tie-breaking if necessary), and the votes who supported it now support their preferred candidate among those that remain.

Given $k \leq m$, a *top-k ballot* is a linear order of $k$ among the $m$ candidates in $A$. A *top-k profile* is a collection of $n$ top-$k$ ballots. Using truncated ballots as a way of reducing the amount of information in voting has been considered in a few recent works, especially [1,3,6–8].

For each $k \leq m$, $STV_k$ is defined similarly as STV, but with top-$k$ ballots as input. In each round, the candidate ranked first by the smallest number of voters is eliminated (using tie-breaking if needed). When all $k$ candidates in a vote have been eliminated, the vote is ignored in later rounds (such a vote will be said to be *exhausted*). We repeat this process until there exists a candidate ranked first by the majority of non-exhausted truncated votes. $STV_1$ coincides with plurality, and $STV_{m-1}$ (and $STV_m$) with $STV$.

In order to evaluate the quality of $STV_k$, we measure the frequency with which the approximation outputs the true winner using randomy generated date with the Mallows model, and then using real data.

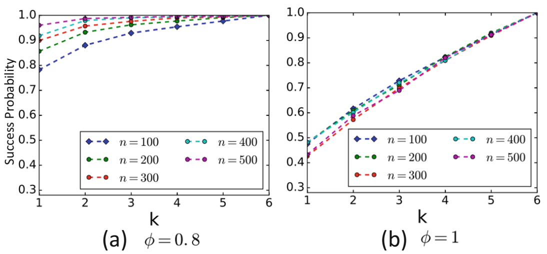


**Fig. 1.** Success probabilities of *top-k* voting for $STV_k$: $m = 7$ varying $n$, $k$ and $\phi$.

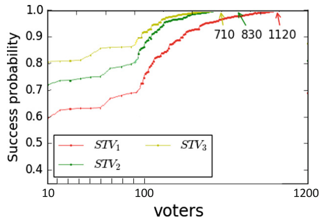


**Fig. 2.** Success probabilities of $STV_k$ with Dublin data: varying $k \in \{1, 2, 3\}$ and $n^*$ $(n^* < n)$.

The Mallows $\phi$ model is described by two parameters: reference ranking $\sigma$ and dispersion parameter $\phi \in [0, 1]$. The probability of a ranking $r$ under this model is: $P(r; \sigma, \phi) = \frac{1}{Z}\phi^{d(r,\sigma)}$ where $d$ is the Kendall tau distance and $Z = \sum_{r'} \phi^{d(r,\sigma)}$ is a normalization constant. We draw 1000 random profiles. then we simulate the elicitation of *top-k* ($k \in \{1 \dots 6\}$) preferences $m = 7$ and let $n$ and $\phi$ vary (Fig. 1).

Our results suggest that the winner is always predicted correctly when $\phi \leq 0.8$, $k = 2$ and with large $n$. When $\phi = 1$, the success rate is 82% with top-4 ballots of 500 voters. In all cases, *top*-2 ballots seem to be always sufficient to predict the correct STV winner with 100% accuracy with small values of $\phi$ and high number of voters.

Next, we use the Dublin data ($n = 3662$, $m = 12$) from the PrefLib library [5], with samples of $n^*$ voters among $n$ ($n^* < n$) where 1000 random profiles are constructed with $n^*$ voters. Then, we consider the top-$k$ ballots obtained from these profiles, where $k \in \{1, 2, 3\}$ over 12 candidates, and we compute the probability of selecting the correct winner (the winner of the complete profile of the $n^*$ sampled votes) (Fig. 2). Our results suggest that predicting the correct winner with a small number of voters fails significantly often when $k$ is too small ($k \leq \frac{1}{4}m$). Also, the performance increases with $n$. Indeed, $k = 1$ is sufficient to predict the correct winner when $n^* \geq 1120$.

Obviously, increasing the value of $k$ leads to a decrease in the number of voters needed for correct winner selection for instance, when $k = \frac{1}{6}m$ (resp. $k = \frac{1}{4}m$) over 12 candidates, $n^* \geq 830$ (resp. $n^* \geq 710$) are needed to always output the correct result.

## 3 Communication Protocols for STV

Now, we allow for more sophisticated, *interactive* protocols where voters may report their preferences incrementally, when the central authority asks them to do so; on the other hand, we are not any longer interested in computing an approximation of STV, but in computing the real STV winner. With the aim of assessing the communication complexity of STV, Conitzer and Sandholm [2] a protocol for STV, which we call $P_1$:

1. each voter submits her most preferred candidate over the set of all available candidates to the central authority ($C$).
2. let $d \in A$ be the candidate ranked first by the fewest voters (using tie-breaking if necessary).
3. $d$ is eliminated; all voters who had $d$ as their current best candidate receive a message from $C$ asking them to send their next preferred candidate among the remaining ones. For each of these voters, their vote is transferred to this next best remaining candidate.
4. this process is repeated until there exists a candidate $x$ ranked first by more than 50% of the votes or only one candidate remains in the set of available candidates.

We say that $x \in A$ is an *immediate loser* if we know that $x$ will be the next candidate eliminated after the currently eliminated one. Formally, let $d$ be the candidate which is about to be eliminated, and $U$ the set of remaining candidates (including $d$); candidate $x$ is an *immediate loser* if for every $y \neq x, d$, either (1)

$S(y, P_U) > S(x, P_U) + S(d, P_U)$, or (2) $S(y, P_U) = S(x, P_U) + S(d, P_U)$ and $y \rhd x$.

Eliminating an immediate necessary loser during the execution of the protocol will never change the final outcome since we know exactly when it will be eliminated, then we can safely remove it.[2] This is the key property used in the next protocol, which we call $P_2$, which is an improvement over $P_1$: in $P_2$, the two first steps are similar as $P_1$. Then, if there is an immediate loser at this point, it is eliminated as well, together with $d$; from the set of available candidates. After $d$ is eliminated, there may be *another* immediate loser; the process is repeated until there is no immediate loser. After removing all immediate losers in $P_U$, we select a voter whose top candidate is $d$ or an immediate loser. We ask this voter to report her next preference among the available ones in $U$. Unlike $P_1$, $P_2$ queries one voter at a time since the new voter's preference may help to detect another immediate losers, thus reduce the set of available candidates. We repeat this process until we obtain a tops-only profile $P$ with candidates among $U$ for each voter. Finally, the process is repeated until there exists a candidate ranked first by more than 50% of the votes or only one candidate remains in $U$.

Now, we evaluate the average communication complexity of $P_1$ and $P_2$ using data generated from the Mallows $\phi$ model. Our objective is to determine the average communication complexity reported from voters in order to return the winner. We refer to $P_{Worst}$ as the theoretical communication complexity.

For each experiment, we draw 1000 random profiles. We simulate the number of bits transferred between the central authority and the voters when with $m = 7$ and let $n$ and $\phi$ vary (see Fig. 3). Results suggest that in practice, we can save a lot in communication costs compared to the theoretical complexity. Even with high $\phi$, using $P_2$, we can save almost 50% of bits communicated. Also, our results suggest that when $\phi \leq 0.8$, $P_2$ is efficient to reduce the communication cost. When $\phi \geq 0.9$, from the results we can detect that $P_1$ and $P_2$ become closer in communication cost.

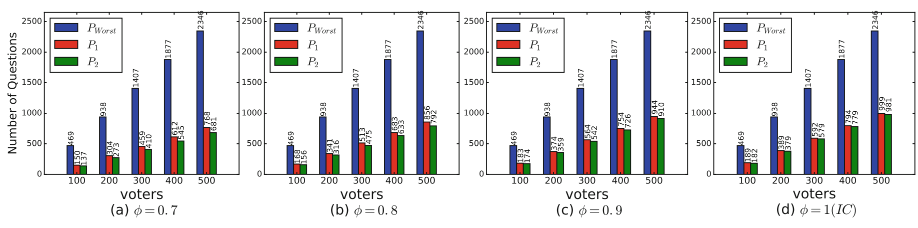


**Fig. 3.** Average communication cost with $P_1$, $P_2$ and $P_{Worst}$

[2] Jiang et al. [4] define a weaker version of necessary losers for STV in the context of a search algorithm for outputting all parallel universe STV winners.

## References


1. Baumeister, D., Faliszewski, P., Lang, J., Rothe, J.: Campaigns for lazy voters: truncated ballots. In: Proceedings AAMAS, pp. 577–584 (2012)
2. Conitzer, V., Sandholm, T.: Communication complexity of common voting rules. In: Proceedings of the 6th ACM Conference on Electronic Commerce, pp. 78–87. ACM (2005)
3. Filmus, Y., Oren, J.: Efficient voting via the top-k elicitation scheme: a probabilistic approach. In: Proceedings of ACM Conference on Economics and Computation, pp. 295–312 (2014)
4. Jiang, C., Sikdar, S., Wang, J., Xia, L., Zhao, Z.: 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 (2017)
5. Mattei, N., Walsh, T.: PrefLib: a library for preferences http://www.preflib.org. In: Perny, P., Pirlot, M., Tsoukiàs, A. (eds.) ADT 2013. LNCS (LNAI), vol. 8176, pp. 259–270. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41575-3_20
6. Naamani-Dery, L., Kalech, M., Rokach, L., Shapira, B.: Reducing preference elicitation in group decision making. Expert Syst. Appl. **61**, 246–261 (2016)
7. Oren, J., Filmus, Y., Boutilier, C.: Efficient vote elicitation under candidate uncertainty. In: Proceedings of IJCAI, pp. 309–316. AAAI Press (2013)
8. Skowron, P., Faliszewski, P., Slinko, A.: Achieving fully proportional representation: approximability results. Artif. Intell. **222**, 67–103 (2015)