Compiling the Votes of a Subelectorate

Yann Chevaleyre Jérôme Lang Nicolas Maudet Guillaume Ravilly-Abadie
LAMSADE
Université Paris-Dauphine
75775 Paris Cedex 16, France
{chevaleyre, lang, maudet, ravilly-abadie}@lamsade.dauphine.fr

Abstract

In many practical contexts where a number of agents have to find a common decision, the votes do not come all together at the same time. In such situations, we may want to preprocess the information given by the subelectorate (consisting of the voters who have expressed their votes) so as to "compile" the known votes for the time when the latecomers have expressed their votes. We study the amount of space necessary for such a compilation, as a function of the voting rule, the number of candidates, and the number of votes already known. We relate our results to existing work, especially on communication complexity.

1 Introduction

Voting is a general method for a group of agents to agree on a common decision. Now, in many practical contexts where a number of agents have to find a common decision via a voting process, the votes do not come all together at the same time. For instance, in some political elections, the votes of the citizens living abroad are known only a few days after the rest of the votes. Or, when voting about a date of a meeting, it often happens that some participants express their preferences later than others. In such situations, we may want to preprocess the information given by the subelectorate (consisting of those voters who have expressed their votes) so as to prepare the ground for the time when the latecomers will have expressed their votes. What does "preparing the ground" exactly mean? We may think of two different criteria:

- *space*: synthesize the information contained in the votes of the subelectorate, using *as little space as possible*, while keeping enough information for computing the outcome once the last votes are known;
- *on-line time*: compile the information, using as much off-line time and space as needed, in such a way that once the last votes are known, the outcome can be computed *as fast as possible*.

These two criteria not only differ, but are, to some extent, opposed, and lead to two very different lines of research. The research area of *knowledge compilation* (*e.g.,* [Cadoli *et al.*, 2002; Darwiche and Marquis, 2002]) lay the focus on on-line time, and typically looks for worst-case exponentially large rewritings of the "fixed part" of the input, enabling on-line time complexity to fall down. While knowledge compilation is definitely relevant to voting (the fixed part being the known votes, and the varying part the votes of the latecomers), and would surely deserve a paper on its own, in this paper, however, we focus on minimizing space.

Why should we care about synthesizing the votes of a subelectorate in as little space as possible in the first place? There are two main reasons for that.

The first reason has to do with the practical acceptance of the voting rule. Suppose the electorate is split into different districts (generally, corresponding to geographical entities, or clusters of agents in a system). Each district can count its ballots separately and communicate the partial outcome to the central authority, which, after gathering the outcomes from all districts, determines the final outcome. *The space needed to synthesize the votes of a district* (with respect to a given voting rule) *is precisely the amount of information that the district has to send to the central authority*. Now, it is important that the voters should be able to check as easily as possible the outcome of the election. In other words, one property that we would like to ensure is the possibility to perform an easy distributed verification of the outcome. Take a simple rule, such as plurality or Borda. Obviously, it is enough (and almost necessary, as we see later) for each district to send only its "local" plurality or Borda scores to the central authority. If the district is small enough, it is not difficult for the voters of this district to check that the local results are sound (for instance, each political party may delegate someone for checking the ballots). Note that this only requires these local results to be made public (which is usually the case in real life: in most countries, they are published in newspapers). Every voter can then check the final outcome from these local outcomes (in the case of plurality or Borda, simply by summing up the local scores). Clearly, if the information about the votes of a district being necessary for computing the final outcome is large (*e.g.*, if one needs to know how many voters have expressed every possible linear order on the candidate set), it will be impractical to publish the results locally, and therefore, difficult to check the final outcome, so that voters may then be reluctant to accept the voting rule.

The second reason is foundational: the size of the minimum information to be extracted from a subset of the votes,

and still guaranteeing that we will have enough information to compute the outcome, is an important property of voting rules that has not been studied yet. Even if this minimum information is, in some cases, easy to determine (such as in the case of plurality), in some other cases it is far from being obvious (see for instance our results about STV).

Although the compilation of the votes of a subelectorate has not been considered before (as far as we know), several related problems have been investigated:

- the *complexity of vote elicitation* [Conitzer and Sandholm, 2002; Walsh, 2008]: given a voting rule r, a set of known votes S, and a set of t new voters, is the outcome of the vote already determined from S?
- the *computation of possible and necessary winners* [Konczak and Lang, 2005; Xia and Conitzer, 2008; Pini *et al.*, 2007]: given a voting rule r, a set of incomplete votes (that is, partial orders on the set of candidates), who are the candidates who can still possibly win the election, and is there a candidate who surely wins it?
- the *communication complexity of voting rules* [Conitzer and Sandholm, 2005]: given a voting rule r and a set of voters, what is the worst-case cost (measured in terms of number of bits transmitted) of the best protocol allowing to compute the outcome of the election?

In the first two cases, the connection is clear. In the most favourable case where the outcome is already determined from S (corresponding to the existence of a necessary winner, or to a positive answer to the vote elicitation problem), the space needed to synthesize the input is just the binary encoding of the winner. The connection to communication complexity will be discussed in Section 2, once communication complexity has been formally introduced. In Section 3, we determine the compilation complexity of some of the most common voting rules. Lastly, in Section 4 we discuss our results and mention further research issues.

2 Compilation complexity as one-round communication complexity

Let X be a finite set of *candidates*, and $p = |X|$. A *vote* is a linear order over X. We sometimes denote votes in the following way: $a \succ b \succ c$ is denoted by abc, etc. For any $m \geq 0$, a *m-profile* is a tuple $P = \langle V_1, \ldots, V_m \rangle$ of votes. Let $\mathcal{P}_X^m$ be the set of all m-profiles over X. We denote by $\mathcal{P}_X^*$ the set of all m-profiles for $m \geq 0$, *i.e.*, $\mathcal{P}_X^* = \cup_{k \geq 0} \mathcal{P}_X^k$. An element of $\mathcal{P}_X^*$ is called a *partial profile*.

Let N a finite set of *voters*, with $n = |N|$. A voting rule is a function r from $\mathcal{P}_X^n$ to X. As the usual definition of most voting rules does not exclude the possibility of ties, we assume these ties are broken by a fixed priority order on candidates. A voting rule is anonymous if it insensitive to the identity of voters.

For $P \in \mathcal{P}_X^m$ and $x \in X$, let $n(P, i, x)$ be the number of votes in P ranking x in position i, and $ntop(P, x) = n(P, 1, x)$ the number of votes in P ranking x first. Let $\vec{s} = \langle s_1, \ldots, s_p \rangle$ be a vector of integers such that $s_1 \geq \ldots \geq s_p$. The scoring rule induced by $\vec{s}$ is defined by: $r_{\vec{s}}(P)$ is the candidate maximizing $\sum_{i=1}^{p} s_i.n(P, i, x)$. Plurality (resp. Borda) is the scoring rule r_P (resp. r_B) corresponding to the vector $\langle 1, 0, \ldots, 0 \rangle$ (resp. $\langle p-1, p-2, \ldots, 0 \rangle$).

We now consider situations where only some of the voters (the "subelectorate") have expressed their votes. Let m be number of voters who have expressed their vote, and $P \in \mathcal{P}_X^m$ the partial profile obtained from these m voters. We say that two partial profiles are r-equivalent if no matter the remaining unknown votes (and no matter how many they are), they will lead to the same outcome[1]:

Definition 1 *Let $P, Q \in \mathcal{P}_X^m$ be two m-voters X-profiles, and r a voting rule. P and Q are* r-equivalent *if for every $k \geq 0$ and for every $R \in \mathcal{P}_X^k$ we have $r(P \cup R) = r(Q \cup R)$.*

Example 1 *Let r_P be plurality and r_B Borda, $X = \{a, b, c\}$, and $m = 4$. Let $P_1 = \langle abc, abc, abc, abc \rangle$, $P_2 = \langle abc, abc, acb, acb \rangle$, and $P_3 = \langle acb, acb, abc, abc \rangle$.*

- *P_2 and P_3 are r_P-equivalent and r_B-equivalent. More generally, they are r-equivalent for every anonymous voting rule r.*
- *P_1 and P_2 are r_P-equivalent. However they are not r_B-equivalent: take $R = \langle bca, bca, bca \rangle$, then we have $r_B(P_1 \cup R) = b$ while $r_B(P_2 \cup R) = a$.*

We denote r-equivalence by $\sim_r$. Obviously, $\sim_r$ is an equivalence relation. Informally, the compilation complexity of r is the minimum space needed to compile a m-voter partial profile P without knowing the remaining profile R. This notion does not take into account the off-line time needed to compute σ, nor the off-line time needed to compute ρ.

Definition 2 *Given a voting rule r, we say that a function σ from $\mathcal{P}_X^m$ to $\{0,1\}^*$ is a* compilation function *for r if there exists a function $\rho : \{0,1\}^* \times \mathcal{P}_X^* \to X$, such that for every $P \in \mathcal{P}_X^m$, every $k \geq 0$, and every $R \in \mathcal{P}_X^k$, $\rho(\sigma(P), R) = r(P \cup R)$. The* compilation complexity *of r is defined by*

$$C(r) = \min\{Size(\sigma) \mid \sigma \text{ is a compilation function for } r\}$$

Up to minor details, compilation complexity coincides with *one-round communication complexity* (even if their intuitive interpretations differ). We start by recalling standard communication complexity. When n agents have to compute a function f, each of them knowing only a part of the input, the deterministic communication complexity (see [Kushilevitz and Nisan, 1997]) of f is the worst-case number of bits that the agents must exchange so as to be able to know the outcome. A *one-round protocol* for two agents A and B is a protocol where A sends only one message to B, and then B sends the output to A (see Section 4.2 of [Kushilevitz and Nisan, 1997]). The *one-round communication complexity* (ORCC) of f is the worst-case number of bits of the best one-round protocol for f. Note that while standard communication complexity does not impose any restriction on the protocol that the agents may use to compute f, ORCC imposes a severe restriction, so that these two notions differ significantly.

The only difference between compilation complexity and ORCC is that compilation complexity does not care about B

[1] We could write a similar definition for a *fixed* number of remaining votes, but we leave it out due to space limitations.

sending back the output to A. Here, A represents the set of voters having already expressed their votes, and B the remaining voters; the space needed to synthesize the votes of A is the amount of information that A must send to B so that B can be able to compute the final outcome.

Since one-round communication complexity is never smaller than standard communication complexity, we could expect the communication complexity of a voting rule is always smaller than its compilation complexity. However, this is not so simple, because in [Conitzer and Sandholm, 2005] (a) there is no partition between two subelectorates: their results mention only the total number of candidates, whereas ours mention the number of candidates who have already expressed their votes, and (b) they count the bits sent by each of the voters, while we do not count the bits sent by our m voters to communicate their votes – we only count the bits needed to communicate the *synthesis* of those votes.

We have the following characterization of $C(r)$. Up to minor details, this is a reformulation (in our own terms) of Exercise 4.18 in [Kushilevitz and Nisan, 1997], and we omit the proof.

Proposition 1 *Let r be a voting rule, m be the number of initial voters, and p the number of candidates. If the number of equivalence classes for $\sim_r$ is $g(m,p)$ then the compilation complexity of $C(r) = \lceil \log g(m,p) \rceil$.*

Proposition 2 *Let r be a voting rule, and r' an anonymous voting rule.*

- $C(r) \leq m \log(p!)$
- $C(r') \leq p! \log\left\{1 + \frac{m}{p!}\right\} + m \log\left\{1 + \frac{p!}{m}\right\}$
  $\leq \min(m \log(p!), p! \log m)$

The proof is easy. For any r, the number of equivalence classes cannot be larger than the number of profiles, and there are $(p!)^m$ possible profiles. For any anonymous voting rule r, the number of equivalence classes corresponds to the number of ways to choose a set of m orders among the set of all possible linear orders on X, which is equal to $M^m_{p!}$. The bound is then the logarithm of the latter number.

Importantly, these bounds are tight: it is possible to design a voting rule (resp. an anonymous voting rule) yielding these bounds. At the other extremity of the spectrum, the compilation complexity of a dictatorship is $\log p$, and that the compilation complexity of r is 0 if and only if r is constant.

3 Some case studies

We now consider a few specific families of voting rules. For each rule r we adopt the following methodology: we first give a characterization of the equivalence classes for $\sim_r$, which we use to count the number of equivalence classes. In simple cases, it will be easy to enumerate exactly these classes, and Proposition 1 will give us the exact compilation complexity of the rule. In more complex cases, we will exhibit a simple upper bound and provide a lower bound of the same order.

3.1 Plurality

Lemma 1 *$P \sim_{r_P} P'$ holds if and only if for every x, $ntop(P,x) = ntop(P',x)$.*

Proof: ($\Leftarrow$) is straightforward. For ($\Rightarrow$), suppose there is an $x \in X$ such that $ntop(P,x) \neq ntop(P',x)$. Without loss of generality, assume $ntop(P,x) > ntop(P',x)$. Now, $\sum_{x \in P} ntop(P,x) = \sum_{x \in P'} ntop(P',x) = m$, therefore there must be an $y \neq x$ such that $ntop(P,y) < ntop(P',y)$. Take a profile Q with $2m - ntop(P,x) - ntop(P,y) + 1$ votes, with $m - ntop(P,x) + 1$ votes having x on top (and whatever below), and $m - ntop(P,y)$ votes having y on top (and whatever below). We have $ntop(P \cup Q, x) = m + 1$, $ntop(P \cup Q, y) = m$, and for every $z \neq x, y$, $ntop(P \cup Q, z) \leq m$. Therefore, $r_P(P \cup Q) = x$. Now, $ntop(P' \cup Q, x) = ntop(P',x) - ntop(P,x) + m + 1 \leq m$, $ntop(P' \cup Q, y) = ntop(P',y) - ntop(P,y) + m \geq m + 1$, and for every $z \neq x, y$, $ntop(P' \cup Q, z) \leq m$. Therefore, $r_P(P \cup Q) = y$. This shows that $P \not\sim_{r_P} P'$. ■

This characterization together with Proposition 1 tells us that the compilation complexity of r_P is exactly $\lceil \log L(m,p) \rceil$, where $L(m,p)$ be the number of vectors of positive integers $\langle \alpha_1, \ldots, \alpha_p \rangle$ such that $\sum_{i=1}^p \alpha_p = m$. The number of such vectors is exactly M^m_p, from which we get:

Corollary 1 $C(r_P) = \Theta\left(p \log(1 + \frac{m}{p}) + m \log(1 + \frac{p}{m})\right)$

Observe that this result yields an *upper bound* in $O(m + p)$, to be compared with the "naive" upper bound that can be derived from the fact that it is sufficient to record the plurality scores of each candidate, which needs $O(p \log m)$ bits.

3.2 Borda

We get the following characterization of $\sim_B$, similar to Lemma 1 for plurality (note that more generally, a similar result holds for any scoring rule). For $P \in \mathcal{P}^m_X$ and $x \in X$, let $score_B(x,P)$ be the Borda score of x obtained from the partial profile P.

Lemma 2 *$P \sim_{r_B} P'$ holds if and only if for every x, $score_B(x,P) = score_B(x,P')$.*

Now, let $B(m,p)$ be the number of vectors of positive integers $\langle \alpha_1, \ldots, \alpha_p \rangle$ corresponding to Borda scores once m votes have been expressed. Note that $\sum_{i=1}^p \alpha_p = \frac{mp(p-1)}{2}$, since each voter distributes $\frac{p(p-1)}{2}$ points among the candidates. However, this alone does not suffice to characterize the set of realizable Borda scores (for instance, if a candidate gets a score of 0, then no other candidate can get less than m).

Proposition 3 $C(r_B) \leq (p-1) \log m(p-1)$

This upper bound is easily obtained by observing that is possible to simply record the scores of $p-1$ candidates, and that this score can be at most $m(p-1)$.[2]

Now we try to exhibit a lower bound approaching this upper bound. The general idea is to focus on a specific subset of vectors of Borda scores. For example, for those vectors where the candidate with the lowest score gets between 0 and m, the second one between m and $2m$, and so on until the penultimate voter, the score of the last candidate can be chosen so as

[2]Note that more generally, for any scoring rule r_S, we have $C(r_S) \leq (p-1) \log m.s_1$.

to make a realizable vector of Borda scores. (Observe that by taking these intervals, the scores of the first $p-1$ candidates can really be chosen independently).

In what follows, we show how to construct profiles that result in such a subset of vectors of Borda scores.

Proposition 4 *Let $\{\delta_1, \ldots, \delta_{p-1}\}$ be a collection of non-negative integers such that $\sum_{i=1}^{p-1} \delta_i \leq \frac{m}{2}$. The vector of Borda scores $\vec{\alpha} = \langle \delta_1, m+\delta_2, 2m+\delta_3, \ldots, m(p-1)+\delta_p \rangle$, where $\delta_p = -\sum_{i=1}^{p-1} \delta_i$, can result from a m-voter profile.*

Proof sketch: We only show that for any $i < p$, the vector of Borda scores $\langle \alpha_1, \alpha_2, \ldots, \alpha_i + 1, \ldots, \alpha_p - 1 \rangle$ of the candidates, where $\forall j \leq p, \alpha_j = 2(j-1)$, can result from a two-voter profile. Denote by $\langle \alpha_1^v, \alpha_2^v, \ldots, \alpha_n^v \rangle$ the vector corresponding to the ballot of voter v. We initially assign to voters 1 and 2 the basic vectors $\langle 0, 1, \ldots, p-1 \rangle$. Now we construct the modified vectors of the two voters as follows: take the scores α_i^1 and α_{i+1}^1 of voter 1 and swap them; then take the scores α_{i+1}^2 and α_{i+2}^2 of voter 2 and swap them; then move back to voter 1 and swap the scores α_{i+2}^1 and α_{i+3}^1, and so on until the last score of voter 1 or voter 2 is reached, in which case no more swap is possible. Now, for any $j \in [i+1, p-1]$, $\alpha_j^1 + \alpha_j^2 = \alpha_j'^1 + \alpha_j'^2$ because the swaps of voters 1 and 2 compensate each other, so the scores of these candidates remain unaffected. On the other hand, the Borda scores of candidates i and p are modified as required (resp. $+1$ and -1). Now, the same principle can be applied to m voters: it is possible to distribute up to $m/2$ points among the first $p-2$ candidates to improve their score obtained from the basic vectors only (with the last candidate compensating by seeing its score decreased by the same amount of points). ∎

Corollary 2 $C(r_B) = \Theta(p \log mp)$

Proof: From Proposition 4, $V_{m/2}^{p-1}$ is the number of profiles with increasing scores. We have $V_{m/2}^{p-1} \geq \frac{1}{2} \times \left(\frac{m}{2}\right)^{p-1}$. The total number of profiles is at least $(p-1)! m^{p-1} 2^{-p}$, and we get the lower bound $(p-1)(\log_2(p-1) + \log_2 m - 2)$. Together with the upper bound, the result holds. ∎

3.3 Rules based on the weighted majority graph

Let $N_P(x,y)$ be the number of voters in the profile P preferring x to y. The *majority graph* M_P is the directed graph whose set of vertices is X and containing an edge from x to y if and only if $N_P(x,y) > N_P(y,x)$. The *weighted majority graph* $\mathcal{M}_P$ is the same as M_P, where each edge from x to y is weighted by $N(x,y)$ (note that there is no edge in $\mathcal{M}_P$ between x and y if and only if $N_P(x,y) = N_P(y,x)$.) A voting rule r is *based on the majority graph* (abridged into "MG-rule") if for any profile P, $r(P)$ can be computed from M_P, and *based on the weighted majority graph* (abridged into "WMG-rule") if for any profile P, $r(P)$ can be computed from $\mathcal{M}_P$. Obviously, a MG-rule is *a fortiori* a WMG-rule. A candidate x is the *Condorcet winner* for a profile P if it dominates every other candidate in M_P. A voting rule r is *Condorcet-consistent* if it elects the Condorcet winner whenever there exists one.

Lemma 3 *Let r be a WMG-rule rule. If $\mathcal{M}_P = \mathcal{M}_{P'}$ then $P \sim_r P'$.*

Proof: for any Q, $\mathcal{M}_{P \cup Q}$ is fully determined from $\mathcal{M}_P$ and $\mathcal{M}_Q$, because $N_{P \cup Q}(x,y) = N_P(x,y) + N_Q(x,y)$. Since r is WMG, $r(P \cup Q)$ is determined from $\mathcal{M}_{P \cup Q}$, therefore from $\mathcal{M}_P$ and $\mathcal{M}_Q$, and *a fortiori* from $\mathcal{M}_P$ and Q. ∎

Note that for rules based on the (non-weighted) majority graph, we still need the *weighted* majority graph of P and P' to coincide – having only the majority graph coinciding is not sufficient for $P \sim_r P'$, since $M_{P \cup Q}$ is generally not fully determined from M_P and M_Q.

Lemma 3 gives an upper bound on the compilation complexity of a WMG-rule. Let $T(m,p)$ be the number of weighted tournaments on X that can be obtained as the weighted majority graph of some m-voter profile.

Proposition 5 *If r is a WMG-rule then $C(r) \leq \log T(m,p)$.*

Getting a lower bound is not possible without a further assumption on r. After all, constant rules are based on the majority graph, yet they have a compilation complexity of 0. We say that a WMG-rule r is *proper* if $P \sim_r P'$ implies $\mathcal{M}_P = \mathcal{M}_{P'}$. It is easy to find a natural sufficient condition for a WMG-rule to be proper:

Lemma 4 *If r is a Condorcet-consistent rule then $P \sim_r P'$ implies $\mathcal{M}_P = \mathcal{M}_{P'}$.*

Proof: assume r is Condorcet-consistent and $\mathcal{M}_P \neq \mathcal{M}_{P'}$, *i.e.*, there exists $(x,y) \in X$ with $N_P(x,y) \neq N_{P'}(x,y)$. Without loss of generality, $N_P(x,y) = N_{P'}(x,y) + k$ (and $N_P(y,x) = N_{P'}(y,x) - k$), with $k > 0$. Let Q be a set of $m+1$ votes, of which $m+1-N_P(x,y)$ votes prefer x to y and y to anyone else and $N_P(x,y)$ votes prefer y to x and x to anyone else. From $N_{P \cup Q}(x,y) = N_P(x,y) + N_Q(x,y) = m+1$, we get for any $z \neq x, y$, $N_{P \cup Q}(x,z) = N_P(x,z) + m + 1 \geq m+1$, and x is Condorcet winner in $P \cup Q$ (which contains $2m+1$ voters) and $r(P \cup Q) = x$. But $N_{P' \cup Q}(y,x) = N_{P'}(y,x) + N_Q(y,x) = N_P(y,x) + k + N_P(x,y) = m + k$, and for any $z \neq x, y$, $N_{P' \cup Q}(y,z) = N_{P'}(y,z) + m + 1 \geq m+1$, so y is Condorcet winner in $P' \cup Q$ and $r(P' \cup Q) = y$. Hence $P \not\sim_r P'$. ∎

This gives us the following lower bound.

Proposition 6 *If r is a Condorcet-consistent rule then $C(r) \geq \log T(m,p)$.*

From Propositions 5 and 6 we finally get that:

Proposition 7 *If r is a Condorcet-consistent WMG-rule, then $C(r) = \log T(m,p)$.*

Corollary 3 *The compilation complexity of the following rules is exactly $\log T(m,p)$:* Copeland, Simpson (maximin), Slater, Banks, uncovered set, Schwartz.

We now have to compute $T(m,p)$. We easily get the following upper bound.

Proposition 8 $\log T(m,p) \leq \frac{p(p-1)}{2}\log(m+2)$.

Proof: From Lemma 3 we know that it is enough to store M_P. Let $>$ be a fixed ordering on the candidates. Storing M_P can be done by storing $N_P(x,y)$ for every pair (x,y) of distinct candidates such that $x > y$. We have $\frac{p(p-1)}{2}$ such pairs, and $0 \leq N_P(x,y) \leq m$, hence the result. ∎

It is easy to observe that this bound is not necessarily reached: for any $x,y,z \in X$ and any profile P we have $N_P(x,z) \geq N_P(x,y) + N_P(y,z) - m$ (*e.g*, if $m = 3$ and $N_P(x,y) = N_P(y,z) = 2$, then $N_P(x,z)$ cannot be 0).

Lemma 5 $T(m,p) \geq \left| V_{\frac{m}{2}}^{\frac{p(p-1)}{2}} \right|$

Proof: Assume m is even. Let $\{c_{i,j} \mid 1 \leq i < j \leq p\}$ be any set non-negative integers such that $\sum_{i<j} c_{i,j} \leq \frac{m}{2}$. We will show how to build a profile such that $N(i,j) = 2c_{i,j}$, where $N(i,j)$ indicates how many voters prefer i to j. Let us divide voters into $\frac{p(p-1)}{2}$ groups $g_{i,j}$ with $1 \leq i < j \leq p$ and a final group g_0, such that each group $g_{i,j}$ contains exactly $2c_{i,j}$ voters and g_0 contains the remaining $m - \sum_{i<j} 2c_{i,j}$ voters. In each group $g_{i,j}$, set the profile of half of the voters to $i \succ j \succ x_1 \succ x_2 \succ \ldots \succ x_{p-2}$, and the other half to $x_{p-2} \succ \ldots \succ x_1 \succ i \succ j$, where $x_1 \ldots x_{p-2}$ refer to the candidates other than i and j in an arbitrary order. In group g_0 set half of the voters to $x_1 \succ \ldots \succ x_p$ and the other half to $x_p \succ \ldots \succ x_1$. Let $N^g(x,y)$ denote the number of voters in group g preferring x to y. Clearly, $N^{g_{ij}}(x,y) = N(x,y)$ if $x = i$ and $y = j$; and 0 otherwise; and $N^{g_0}(x,y) = 0$. Thus, $N(x,y) = \sum N^{g_{i,j}}(x,y) = 2c_{i,j}$. ∎

From Lemma 5, it directly follows that:

Corollary 4 *If r is a Condorcet-consistent WMG-rule then* $C(r) = \Theta(p^2 \log m)$.

3.4 Plurality with runoff

Plurality with runoff (denoted by r_2) consists of two (virtual) rounds: the first round keeps the two candidates with maximum plurality scores (with some tie-breaking mechanism), and the second round is the majority rule.

Proposition 9 *$P \sim_{r_2} Q$ holds if and only if the following two conditions hold:*

1. *for every x, $ntop(P,x) = ntop(Q,x)$;*
2. *for every x and $y \neq x$, $N_P(x,y) = N_Q(x,y)$.*

Due to space limitations, we give only a very quick sketch of the proof. The ($\Leftarrow$) direction (if the two conditions are met then $P \sim_{r_2} Q$) is straightforward. Next, we show that if for some x, $ntop(P,x) \neq ntop(Q,x)$, then $P \not\sim_{r_2} Q$; we do so by taking, without loss of generality, an x such that $ntop(P,x) > ntop(Q,x)$, and then we construct a profile R such that in $P \cup R$, the finalists are x and z, and the winner is x, while in $Q \cup R$, the finalists are y and z (therefore the winner cannot be x). Lastly, we show that if for some $x,y \in X$, $N(P,x,y) \neq N(Q,x,y)$ then $P \not\sim_{r_2} Q$. Assuming without loss of generality that $N(P,x,y) > N(Q,x,y)$, we complete P and Q such that in both $P \cup R$ and $Q \cup R$, the finalists are x and y, with x winning in $P \cup R$ and y in $Q \cup R$.

Observe that the two conditions are not independent, so that the previous proposition only tells us that $\log T(m,p) \leq C(r_2) \leq \log L(m,p) + \log T(m,p)$. It follows that:

Corollary 5 $C(r_2) = \Theta\left(p^2 \log m\right)$.

In words, compiling the votes in the context of plurality with runoff or of a Condorcet-consistent WMG rule requires an asymptotically equivalent number of bits.

3.5 Single Transferable Vote (STV)

We recall that single transferable vote performs in successive rounds: at each round, the candidate with the lowest plurality score gets eliminated, and its votes are transferred to the next preferred candidate in each ballot. Given a collection P of votes, and a subset $Z \subseteq X$ of candidates, we denote by P^{-Z} the collection of votes obtained from P by removing those candidates in Z, leaving the rest unchanged. For instance, $\langle abcde, bcade, beadc\rangle^{-bc} = \langle ade, ade, ead\rangle$.

Proposition 10 *$P \sim_{STV} Q$ holds if and only if for all Z and $x \notin Z$, $ntop(P^{-Z},x) = ntop(Q^{-Z},x)$.*

Proof: The ($\Leftarrow$) part is straightforward. As for the $\Rightarrow$ part, suppose that for the partial profiles P and Q, there exist Z and $x_1 \notin Z$ such that $ntop(P^{-Z},x_1) \neq ntop(Q^{-Z},x_1)$. Without loss of generality, suppose $ntop(P^{-Z},x_1) < ntop(Q^{-Z},x_1)$. Note that if Z had contained all candidates but one, then these two scores would have necessarily been equal. Thus, there exist candidates not in Z and different from x_1. Among these candidates, there exist at least two candidates such that $ntop(P^{-Z},x) > ntop(Q^{-Z},x)$, and we will refer to one of them as x_2. Let $\epsilon = ntop(P^{-Z},x_1) - ntop(P^{-Z},x_2) + 1$. Let $\{e_1,\ldots,e_k\} = Z$ and $\{\overline{e_1},\ldots,\overline{e_l}\}$ be the candidates not in $Z \cup \{x_1,x_2\}$. We now add the profile R to P and to Q:

- $3m$ votes : $x_1 \succ x_2 \succ \overline{e_1} \succ \ldots \succ \overline{e_l} \succ e_1 \succ \ldots \succ e_k$
- $3m+\epsilon$ votes : $x_2 \succ x_1 \succ \overline{e_1} \succ \ldots \succ \overline{e_l} \succ e_1 \succ \ldots \succ e_k$
- $4m+2$ votes: $\overline{e_1} \succ x_1 \succ x_2 \succ \overline{e_2} \succ \ldots \succ \overline{e_l} \succ e_1 \succ \ldots \succ e_k$
- $4m+2$ votes: $\overline{e_2} \succ x_1 \succ x_2 \succ \overline{e_3} \succ \ldots \succ \overline{e_1} \succ e_1 \succ \ldots \succ e_k$
- $4m+2$ votes: $\overline{e_3} \succ x_1 \succ x_2 \succ \overline{e_4} \succ \ldots \succ \overline{e_2} \succ e_1 \succ \ldots \succ e_k$
- etc.

After the first k rounds of STV have been applied to $P \cup R$ and $Q \cup R$, all the candidates in Z are eliminated. Then, the score of $\overline{e_i}$ is between $4m+2$ and $5m+2$. Note that $ntop((P \cup R)^{-Z},x_2) = ntop((P \cup R)^{-Z},x_1) + 1$, and $ntop((Q \cup R)^{-Z},x_2) < ntop((Q \cup R)^{-Z},x_1)$. Thus, the scores of x_1 and x_2 are no larger than $4m+1$ in $P \cup R$ and $Q \cup R$. Moreover, candidate x_1 is removed from $P \cup R$, and x_2 is removed from $Q \cup R$. After this round, the candidate that remains among $\{x_1,x_2\}$ gets at least $6m$ votes. Thus, all candidates $\{\overline{e_1},\ldots,\overline{e_l}\}$ will be removed one after the other, and x_1 will win in $Q \cup R$, and x_2 will win in $P \cup R$. This

proves that $Q \cup R$ and $P \cup R$ are not STV-equivalent. ■

In other words, the previous proposition shows that the number of STV-equivalence classes is exactly the number of ways to define $ntop(P^{-Z}, x)$ for the p candidates.

Lemma 6 *For a given candidate x, the number of score functions $ntop(P^{-Z}, x)$ realizable with a partial profile P of m voters and p candidates is exactly $M^m_{2^{p-1}}$.*

Proof: Observe that it is required and sufficient, for each candidate x and possible subset of candidates Z (not containing x), to record how many voters among m prefers x over any candidate in Z. ■

Proposition 11 *The compilation complexity of STV is:*
- $\Omega\left(2^p \log\left\{1 + \frac{m}{2^p}\right\} + m \log\left\{1 + \frac{2^p}{m}\right\}\right)$
- $O\left(p2^p \log\left\{1 + \frac{m}{2^p}\right\} + mp \log\left\{1 + \frac{2^p}{m}\right\}\right)$

Proof: For a given x, Lemma 6 shows that there are exactly $M^m_{2^{p-1}}$ score functions $ntop(P^{-Z}, x)$. ■

Note that the score functions of the p candidates are not independent (knowing the score of some candidate(s) may constrain the score of other candidates). Therefore, the number of equivalence classes lies somewhere between $M^m_{2^{p-1}}$ and $(M^m_{2^{p-1}})^p$.

4 Conclusion

This paper has introduced a notion that we believe to be of primary importance in many practical situations, especially in large systems where it is unlikely that all the votes will come together at the same time: the compilation of incomplete profiles. In particular, the amount of information that a given polling station (or cluster of agents) needs to transmit to the central authority is a good indicator of the difficulty of the verification process. We have introduced a general technique for determining the compilation complexity of a voting rule, have related it to communication complexity, and have derived results for most of the "common" voting rules.

Our results allow us to clearly rank most of the different voting rules wrt. the quantity of information required to encode the known profiles and still guarantee the correct outcomes when new voters come in. It is noteworthy to observe that the rules are ranked quite differently under the criteria of their communication complexity [Conitzer and Sandholm, 2005]. Consider for instance plurality with runoff: its compilation complexity is higher than Borda, although it exhibits a lower communication complexity. As another example, Borda is less costly to compile than Condorcet-consistent WMG-rules, but its communication complexity is higher.

A question that we plan to consider more carefully concerns the situations where the number k of remaining voters is fixed. In this case, a different approach can be taken: instead of compiling the partial profiles provided by the m voters, it may be more efficient to compile the possible completions of this partial profile together with the associated outcome, or, in other words, to compile the function that takes the remaining k profiles as input (there are $(p!)^k$ such inputs) and returns the outcome. As there are $(p!)^k$ possible profiles, the number of such functions is $p^{(p!)^k}$. Therefore, a general upper bound for $C(r,k)$ is $\leq (p!)^k . \log p$. Hence, overall we have $C(r,k) \leq \min(m . \log(p!), (p!)^k . \log p)$ (the second term becomes interesting only when m is big enough, and p and k small enough).

More generally, the problem of dealing with incomplete profiles in this sense opens a host of related questions, *e.g.* the probability that a voting process could be stopped after m voters, or the probability that the central authority would make a mistake were it forced to commit on a winner in situations when no candidate is guaranteed to prevail yet.

Acknowledgements

We thank the anonymous reviewers for helpful comments. This research is supported by the project ANR-05-BLAN-0384 "Preference Handling and Aggregation in Combinatorial Domains".

References

[Cadoli *et al.*, 2002] M. Cadoli, F. Donini, P. Liberatore, and M. Schaerf. Preprocessing of intractable problems. *Information and Computation*, 176(2):89–120, 2002.

[Conitzer and Sandholm, 2002] V. Conitzer and T. Sandholm. Vote elicitation: complexity and strategy-proofness. In *Proceedings of AAAI-02*, pages 392–397, 2002.

[Conitzer and Sandholm, 2005] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of EC'05*, pages 78–87, 2005.

[Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A knowledge compilation map. *JAIR*, 17:229–264, 2002.

[Konczak and Lang, 2005] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[Kushilevitz and Nisan, 1997] E. Kushilevitz and N. Nisan. *Communication complexity*. Cambridge University Press, 1997.

[Pini *et al.*, 2007] M.S. Pini, F. Rossi, K. Brent Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proceedings of IJCAI'07*, pages 1464–1469, 2007.

[Walsh, 2008] T. Walsh. Complexity of terminating preference elicitation. In *Proceedings of AAMAS-08*, pages 967–974, 2008.

[Xia and Conitzer, 2008] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of AAAI-08*, pages 196–201, 2008.