# Compiling the votes of a subelectorate

Y. Chevaleyre[*], J. Lang[†], N. Maudet[*], G. Ravilly-Abadie[*]

**Abstract**

In many practical contexts where a number agents have so as to find a common decision, the votes do not come all together at the same time (for instance, when voting about a date for a meeting, it often happens that one or two participants express their preferences later than others). In such situations, we might want to preprocess the information given by the subelectorate (consisting of those voters who have expressed their votes) so as to "compile" the known votes for the time when the latecomers will have expressed their votes. We study the amount of space necessary to such a compilation, in function of the voting rule used, the number of candidates, the number of voters who have already expressed their votes and the number of remaining voters. We position our results with respect to existing work, especially on vote elicitation and communication complexity.

**Key words :** Communication Complexity, Computational Social Choice

## 1 Introduction

In many practical contexts where a number agents have so as to find a common decision, the votes do not come all together at the same time. For instance, in some political elections, the votes of the citizens living abroad is known only a few days after the rest of the votes. Or, when voting about a date for a meeting, it often happens that one or two participants express their preferences later than the others. In such situations, we might 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:

[*]LAMSADE, Université Paris-Dauphine, 75775 Paris cedex 16, France. {chevaleyre,maudet,ravilly-abadie}@lamsade.dauphine.fr

[†]IRIT, CNRS 31062 Toulouse, France. {lang}@irit.fr

- *space*: synthesize the information contained in the votes of the subelectorate, using *as less space as possible*, while keeping enough information so as to be able to compute the outcome once the newcomers hace expressed their votes;
- *on-line time*: compile the information, using as much off-line time and space as needed, in such a way that once the newcomers hace expressed their vote, the outcome can be computed *as fast as possible*.

These two criteria not only differ, but are, to some extent, opposed.

The research area of *knowledge compilation* (see for instance [3, 6]) lay the focus on on-line space 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 (and do not care about on-line time).

While should we care about synthesizing the votes of a subelectorate in as less space as possible? After all, one may think, the current cost of storage is so low that one should not care about storing millions of votes. There are two possible objections to this line of argumentation. The first one has to do with the size of the candidate set. In one-seat political elections, the number of candidates is typically no more than a dozen; however, in "profane" votes, such as multiple elections [2], the set of candidates has a combinatorial structure and can be extremely large (possibly much more than a few millions – while it is difficult to imagine an election with more than a few million voters). The second objection has to do with the practical acceptance of the voting rule. Suppose the electorate is split into different districts (generally, corresponding to geographical entities). Each district can count its ballots separately and communicate th! e partial outcome to the central authority (e.g. the Ministry of Innner Affairs), which, after gathering the outcomes from all districts, will determine 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. 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); provided these local results are made public (which is usually the case – in most countries, th! ey are published in newspapers), every voter can check the fin! al 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, and voters may then be reluctant to accept the voting rule. 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* [4]: 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* [7, 11, 9, 10]: 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* [5]: 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 extremely favourable case where the outcome of the vote 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 with communication complexity [8] will be discussed more explicitly in Section 2, after the notion of compilation is introduced formally. Then in Section 3 we determine the compilatioon complexity of some of the most common voting rules.

## 2 Compilation complexity as one-round communication complexity

Let $X$ be a finite set of *candidates* and $N$ a finite set of *voters*. Let $p = |X|$ and $n = |N|$. 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 $m \leq n$, a $(p, m)$*-profile* is a tuple $P = \langle V_1, \ldots, V_m \rangle$ where each $V_i$ is a vote. When $m < n$ (resp. $m = n$), we call such profiles *partial* (resp. *complete*). Let $\mathcal{P}_X^m$ be the set of all $m$-voters profiles over $X$. A voting rule is a function $r$ from $\mathcal{P}_X^n$ to $X$. As the usual definition of most common voting rules does not exclude the possibility of ties, we assume these ties are broken by a fixed priority order on candidates.

We now consider situations where only some of the voters (the “subelectorate”) have expressed their votes. Let $m \leq n$ 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, they will lead to the same outcome. We distinguish between two cases, depending on whether the number of remaining voters is fixed or not.

**Definition 1** *Let $P, Q \in \mathcal{P}_X^m$ be two $m$-voters $X$-profiles and $r$ a voting rule. We say that*

- *given $k \geq 0$, $P$ and $Q$ are $(r,k)$-*equivalent *if for every $R \in \mathcal{P}_X^k$ we have $r(P \cup R) = r(Q \cup R)$.*
- *$P$ and $Q$ are $r$-*equivalent *if they are $(r,k)$-equivalent for every $k \geq 0$.*

**Example 1** *Let $r_P$ be the plurality rule and $r_B$ the Borda rule, $X = \{a,b,c\}$ and $m = 4$. Let $P_1 = \langle abc, abc, abc, abc \rangle$, $P_2 = \langle abc, abc, acb, acb \rangle$, $P_3 = \langle acb, acb, abc, abc \rangle$ and $P_4 = \langle abc, abc, abc, bca \rangle$. Then we have the following:*

- *$P_2$ 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. They are also $(r_B,k)$-equivalent for every $k \leq 2$. However they are not $(r_B,k)$-equivalent for $k \geq 2$. For $k = 3$, this can be seen by considering $R = \langle bca, bca, bca \rangle$. We have $r_B(P_1 \cup R) = b$ but $r_B(P_2 \cup R) = a$; therefore, $P_1$ and $P_2$ are not $r_B$-equivalent.*
- *$P_1$ and $P_4$ are $(r_P,k)$-equivalent for every $k \leq 2$, but not for $k \geq 2$, therefore they are not $r_P$-equivalent (nor $r_B$-equivalent).*

We denote $(r,k)$-equivalence and $r$-equivalence by, respectively, $\sim_{r,k}$ and $\sim_r$. Obviously, $\sim_{r,k}$ and $\sim_r$ are transitive, therefore they are indeed equivalence relations. We now define the *compilation complexity* of a voting rule. We have two notions, depending on whether the number of remaining candidates (i.e. the size of $R$) is fixed or not.

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

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

Informally, the compilation complexity of $(r, k)$ is the minimum space needed to compile the $m$-voter partial profile $P$ without knowing the remaining $k$-voter profile $R$. This notion does not take into account the off-line time needed to compute $\sigma$, nor the off-line time needed to compute $\rho$. The usual knowledge compilation view would focus on minimizing the time needed to compute $\rho$,regardless of the size of $\sigma$ (and the time needed to compute it). The definitions when $k$ is not fixed are similar:

**Definition 3** *Given a voting rule $r$, we say that a function $\sigma$ from $\mathcal{P}_X^m$ to $\{0,1\}^*$ is a* compilation function *for $r$ if there exists a function $\rho : \{0,1\}^* \times \mathcal{P}_X^* \to X$, where $\mathcal{P}_X^* = \cup_{k \geq 0} \mathcal{P}_X^k$, 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*

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

An equivalent way of seeing compilation complexity is related to multiparty communication complexity. When $n$ agents have to compute a function $f$, while each of them only knows a part of the input, the deterministic communication complexity (see [8]) of $f$ is the worst-case number of bits that the agents have to exchange so as to be able to know the outcome. The communication complexity of common voting rules is identified in [5].

While standard communication complexity does not impose any restriction on the protocol that the agents may use to compute $f$, imposing such restrictions leads to variants of communication complexity; especially, 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 [8]). The *one-round communication complexity* of $f$ is the worst-case number of bits of the best one-round protocol for $f$. This is exactly the same as the compilation complexity of $f$, up to a minor difference: we do not care about $B$ 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[2].

[2]Since one-round communication complexity is never smaller than standard communication complexity, we expect the lower communication complexity bounds communication in [5] to be lower bounds of compilation complexity. However, making this more precise is not so simple, because in [5] 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. Let $D(r, n, p)$ the (deterministic) communication complexity of $r$ for $n$ voters and $p$ candidates as in [5]. Let us now introduce this variant of communication complexity: if $m \leq n$, define $D(r, n, m, p)$ as the cost of the optimal protocol for computing $r$, where only the bits sent by the $m$ first voters count for the cost of a protocol (the remaining $n - m$ can communicate for free). Obviously, we have $D(r, n, m, p) \leq D(r, n, p)$. Moreover, if $C(r, m, p)$ is the compilation complexity of $r$ for $m$ voters and $p$ candidates then for every $n \geq m$ we have $C(r, m, p) \geq D(r, n, m, p)$. In order to conclude $C(r, m, p) \geq D(r, m, p)$, we would have to show that for all voting rules considered here, we have $D(r, n, m, p) = D(r, m, p)$ (which we conjecture).

We have this following general characterization of compilation complexity. Up to minor details, this is a reformulation of Exercise 4.18 in [8]. For the sake of the exposition, we reformulate it in our own terms and include its proof.

**Proposition 1** *Let $r$ be a voting rule. Let $m$ be the number of initial voters and $p$ the number of candidates.*

- *given $k \geq 0$, if the number of equivalence classes for $\sim_{r,k}$ is $f(m,p,k)$ then the compilation complexity of $(r,k)$ is exactly $\lceil \log f(m,p,k) \rceil$.*
- *if the number of equivalence classes for the $r$-equivalence relation $\sim_r$ is $g(m,p)$ then the compilation complexity of $r$ is exactly $\lceil \log g(m,p) \rceil$.*

*Proof:* We give the proof only for the case of $(r,k)$; the proof with unbounded $k$ is similar. We first show that $C(r,k) \geq \lceil \log f(m,p,k) \rceil$. Suppose $\sim_{k,r}$ has $f(m,p,k)$ equivalence classes. Assume there is a number $\theta < \lceil \log f(m,p,k) \rceil$, a function $\sigma : \mathcal{P}_X^m \to \{0,1\}^\theta$ and a mapping $\rho : \{0,1\}^* \to X$ such that for every $P \in \mathcal{P}_X^m$ and $R \in \mathcal{P}_X^k$, $\rho(\sigma(P), R) = r(P \cup R)$. We first note that $\theta < \lceil \log f(m,p,k) \rceil$ implies $\theta < \log f(m,p,k)$. Let $\approx_\sigma$ be the equivalence relation on $\mathcal{P}_X^m$ defined by $P \approx_\sigma Q$ if $\sigma(P) = \sigma(Q)$. Because for every $P$, $|\sigma(P)| \leq \theta$, $\approx_\sigma$ has at most $2^\theta$ equivalence classes. Since $2^\theta < f(m,p,k)$, $\approx_\sigma$ has strictly less equivalence classes than $\sim_{k,r}$. Hence there exists a pair $(P,Q)$ such that $\sigma(P) = \sigma(Q)$ but $P \not\sim_{k,r} Q$. $P \not\sim_{k,r} Q$ means that there exists a profile $R \in \mathcal{P}_X^k$ such that $r(P \cup R) \neq r(Q \cup R)$. Now, $r(P \cup R) = \rho(\sigma(P), R) = \rho(\sigma(Q), R) = r(Q \cup R)$, hence a contradiction. We now show that $C(r,k) \leq \lceil \log f(m,p,k) \rceil$. Let us enumerate and number all $f(m,p,k)$ equivalence classes for $\sim_{k,r}$. For every $P$, let $i(P)$ be the index of its equivalence class for $\sim_{k,r}$. Define the translation $\sigma(P) = i(P)$. We note that the size of $\sigma$ is exactly $\lceil \log f(m,p,k) \rceil$.! Now, define $\rho$ by $\rho(j,R) = r(P \cup R)$ for an arbitrary $P$ such that $i(P) = j$. The result follows. ■

Here are now a few simple results about voting rules in general.

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

- *$C(r) \leq m \log(p!)$;*
- *$C(r') \leq \min(m \log(p!), p! \log m)$.*

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 $r$, the bound $p! \log m$ comes from the fact that linear orders on $X$ can be enumerated, together with the number of voters who choose it. $p! \log m$ can be smaller than $m. \log(p!)$ when $m$ becomes large enough and $p$ small enough. At the other extremity of the spectrum, we have:

**Proposition 3**

- *the compilation complexity of a dictatorship is* $\log p$*;*
- *the compilation complexity of* $r$ *is* $0$ *if and only if* $r$ *is constant.*

In these limit cases, whether we know or not the number of remaining voters is irrelevant.

## 3 Some case studies

We now consider a few specific families of voting rules. For each of these we adopt the following methodology: we first seek a characterization of the equivalence classes for the given rule, then we use this characterization 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 and Borda

Let $\vec{s} = \langle s_1, \ldots, s_n \rangle$ be a vector of integers such that $s_1 \geq s_2 \geq \ldots \geq s_n = 0$. The scoring rule induced by $\vec{s}$ is defined by: for every candidate $x$, $score_{\vec{s}}(x, P) = \sum_{i=1}^{n} s_i.n(P, i, x)$, where $n(P, i, x)$ is the number of votes in $P$ that rank $x$ in position $i$; and $r_{\vec{s}}(P)$ is the candidate maximizing $score_{\vec{s}}(x, P)$ (in case of a tie, a priority relation on candidates is applied). The plurality (resp. Borda) rule $r_P$ (resp. $r_B$) is the scoring rule corresponding to the vector $\langle 1, 0, \ldots, 0 \rangle$ (resp. $\langle p-1, p-2, \ldots, 0 \rangle$).

**Plurality.** We begin with the compilation complexity of plurality (antiplurality is similar).

**Lemma 1** *For* $P \in \mathcal{P}_X^m$ *and* $x \in X$*, let* $ntop(P, x)$ *be the number of votes in* $P$ *ranking* $x$ *first.* $P \sim_{r_P} P'$ *holds if and only if for every* $x$*,* $ntop(P, x) = ntop(P', x)$*.*

*Proof:* The ($\Leftarrow$) direction is obvious. For the ($\Rightarrow$) direction, 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, we have $\sum_{x \in P} ntop(P,x) = \sum_{x \in P'} ntop(P',x) = m$, therefore there must be an $y$ such that $ntop(P,y) < ntop(P',y)$. Note that we necessarily have $y \neq x$. Now, let $Q$ be the following profile with $2m - ntop(P,x) - ntop(P,y) + 1$ voters: $m - ntop(P,x) + 1$ voters have $x$ on top (and whatever below), and $m - ntop(P,y)$ voters have $y$ 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, we have $ntop(P' \cup Q, x) = ntop(P',x) - ntop(P,x) + m + 1 \leq m$, $ntop(P' \cup Q, y) = ntop(P',y) - ntop(P,y) + m \geq m + 1$, and for every $z \neq x, y$, $ntop(P' \cup Q, z) \leq m$. Therefore, $r_P(P \cup Q) = y$. This sh! ows 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 known, in fact it is equivalent to the number of ways to choose $m$ elements from a set of size $p$ when repetition is allowed, that is $\binom{p+m-1}{m}$ —see *e.g.* [1]. A more explicit expression can be obtained at the price of a very tight approximation, by using Stirling's formula for factorials. The following result is then obtained after a few algebraic rewritings.

**Corollary 1** *The compilation complexity of* $r_P$ *is* $\Theta\left(p \log(1 + \frac{m}{p}) + m \log(1 + \frac{p}{m})\right)$

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

**Borda.** We get this intuitive characterization of $\sim$ for the Borda rule, in a similar way as Proposition 1 for plurality. More generally, a similar result holds for any scoring rule.

**Lemma 2** *For* $P \in \mathcal{P}_X^m$ *and* $x \in X$, *let* $score_B(x,P)$ *be the Borda score of* $x$ *obtained from the partial profile* $P$. $P \sim_{r_B} P'$ *holds if and only if for every* $x$, $score_B(x,P) = score_B(x,P')$.

Let us denote by $B(m,p)$ the number of vectors of positive integers $\langle \alpha_1, \ldots, \alpha_p \rangle$ corresponding to Borda scores once $m$ votes have been expressed. Observe that we necessarily have 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 $m$). An 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 $m(p-1)$.

**Proposition 4** *The compilation complexity of Borda is at most* $(p-1)\log m(p-1)$.

Now we try to exhibit a lower bound that will approach this upper bound. The general idea is to restrict our attention to a 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 between $m$ and $2m$, and so on until the penultimate voter, the score of the last candidate can be chosen on purpose so as to make a realizable vector of Borda scores. (Observe that by taking these intervals, the scores of the $p-1$ first candidates can really be chosen independently).

In what follows, we show how to construct profiles that result in the desired vectors of Borda scores, albeit for the sake of readability we shall confine ourselves to a slightly more restricted case than the one discussed above. Technically, the bound obtained is slightly less tight, but the proof is easier to follow. Let us call *basic score* the vector of Borda scores obtained when all voters cast their vote similarly $\langle 0, 1, \ldots, p-1\rangle$. The following Lemma shows that two voters can produce a vector where any candidate can obtain one more vote than the basic score, while the last candidate obtains one vote less.

**Lemma 3** *For any $i < p$, the vector of Borda scores $\langle \alpha_1, \alpha_2, \ldots, \alpha_i + 1, \ldots, \alpha_p - 1\rangle$ where $\forall j \leq p, \alpha_j = 2(j-1)$ can result from a two-voter profile.*

*Proof:* We 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 voter 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 $\alpha_i^1$ and $\alpha_{i+1}^1$ of voter 1 and swap them; then take the scores $\alpha_{i+1}^2$ and $\alpha_{i+2}^2$ of voter 2 and swap them; then move back to voter 1 and swap the scores $\alpha_{i+2}^1$ and $\alpha_{i+3}^1$, and so on until the last score of voter 1 or voter 2 is reached, in which case no more swap is possible. Observe now that $\forall j \in [i+1, p-1]$, $\alpha_j^1 + \alpha_j^2 = \alpha_j'^1 + \alpha_j'^2$ because the swaps of voter 1 and 2 compensate each other, so the scores of these candidates remain unaffected. On the other hand, the Borda scores of candidate $i$ and $p$ are modified as required (resp. $+1$ and $-1$). ■

But the same principle can be applied with $m$ voters: in short, it is possible to distribute up to $m/2$ points among the first $p-2$ candidates to improve over their basic score (with the last candidate compensating by seeing its score decreased by the same amount of points):

**Proposition 5** *Let $\{\delta_1,\ldots,\delta_{p-1}\}$ be any set of non-negative integers such that $\sum_{i=1}^{p-1}\delta_i \leq \frac{m}{2}$. The vector of Borda scores $\langle \delta_1, m+\delta_2, 2m+\delta_3,\ldots,2(p-1)+\delta_p\rangle$, where $\delta_p = -\sum_{i=1}^{p-1}\delta_i$, can result from a $m$-voter profile.*

*Proof:* Let $m' = m - \sum_{i=1}^{p-1} 2\delta_i$. In the following, we will consider sums of profiles and multiplications by constants. In particular, $a \times \langle x_1, x_2\ldots\rangle$ will refer to the profile $\langle ax_1, ax_2,\ldots\rangle$. The above profile can be decomposed as follows as a sum of scores

$$\begin{array}{rcl}\vec{\alpha}_1 & = & 2\delta_1 \times \langle 0,1,2\ldots\rangle + \langle \delta_1, 0,0,0,\ldots - \delta_1\rangle \\ \vec{\alpha}_2 & = & 2\delta_2 \times \langle 0,1,2\ldots\rangle + \langle 0,\delta_2,0,0,\ldots - \delta_2\rangle \\ \vec{\alpha}_3 & = & 2\delta_3 \times \langle 0,1,2\ldots\rangle + \langle 0,0,\delta_3,0,0,\ldots - \delta_3\rangle \\ & \ldots & \\ \vec{\alpha}_p & = & m' \times \langle 0,1,2\ldots\rangle\end{array}$$

The last score can be realized by simply summing $m'$ scores $\langle 0,1,2,\ldots\rangle$. As according to Lemma 3 the scores $\alpha_i$ can be obtained by summing $2\delta_i$ scores, the result follows. ■

**Corollary 2** *The compilation complexity of the Borda rule is $\Theta(p\log mp)$.*

*Proof:* Let $\mathbf{1_C}$ be the indicator function valued $1$ if condition $C$ is true and $0$ otherwise. In Proposition 5, we showed that the number of profiles in which candidates $0\ldots p-2$ have increasing scores is at least $\left|\left\{\langle\alpha_0\ldots\alpha_{p-2}\rangle \in \mathbb{N}^{p-1} \mid \sum_{i=0}^{p-2}\alpha_i \leq \frac{m}{2}\right\}\right|$. More generally, the question amounts to enumerating $V_t^s$, the set of vectors of $s$ non-negative integers, whose sum is lower or equal to $t$. This value can be written as $\int_{\alpha_0..\alpha_{s-1}=0}^{\infty}\mathbf{1}_{\sum\lfloor\alpha_i\rfloor\leq t}d\alpha_0\ldots d\alpha_{s-1}$. Clearly, this can be lower bounded by $\int_0^\infty \mathbf{1}_{\sum\alpha_i\leq t}d\alpha_0\ldots d\alpha_{s-1}$. But this is equal to half of the volume of the hypercube of dimension $s$ whose side has length $t$. (For example, with $s=2$, this value becomes half the area of a square $\frac{t^2}{2}$). More generally, we then have $V_t^s \geq \frac{t^s}{2}$. In our case, this gives us $\frac{1}{2}\times\left(\frac{m}{2}\right)^{p-1}$. Note that this lower bounds the number of profiles with increasing scores. Thus, the total number of profiles is at least $(p-1)!m^{p-1}2^{-p}$. Using the fact that $\log n! \geq n\log n$, we get the lower bound $(p-1)(\log_2(p-1)+\log_2 m - 2)$. Together with the upper bound, the result holds. ■

## 3.2 Rules based on the weighted majority graph

We now consider tournament-based rules. Let $P$ be a profile. $N_P(x,y)$ denotes the number voters in $P$ 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 4** *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 $N_{P \cup Q}(x, y) = N_P(x, y) + N_Q(x, y)$. If $r$ is a WMG-rule then $r(P \cup Q)$ is fully 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 4 gives an upper bound on the compilation complexity of a WMG-rule. Let $T(m, p)$ be the set of all weighted tournaments on $X$ that can be obtained as the weighted majority graph of some $m$-voter profile.

**Proposition 6** *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 $\mathcal{M}_P = \mathcal{M}_{P'}$[3]. It is easy to find a natural sufficient condition for a WMG-rule to be proper:

**Lemma 5** *If $r$ is a Condorcet-consistent rule then $P \sim_r P'$ implies $\mathcal{M}_P = \mathcal{M}_{P'}$.*

[3]Examples of WMG-rules that are not proper: constant rules; dictatorial rules; strange rules such as $r(P)$ = first $x_i$ (wrt a fixed ordering $x_1 > ... > x_p$ on candidates) such that for all $x_j \neq x_i$ there is at least one voter who prefers $x_i$ to $x_j$, and $x_p$ if there is no such $x_i$; "restricted" rules such as $r(P)$ being defined as the candidate maximizing the Copeland score among a fixed subset of candidates; etc.

*Proof:* Let $r$ be a Condorcet-consistent rule. Assume $\mathcal{M}_P \neq \mathcal{M}_{P'}$, *i.e.*, there exists $(x, y) \in X$ with $N_P(x, y) \neq N_{P'}(x, y)$. W.l.o.g., $N_P(x, y) = N_{P'}(x, y) + k$ (hence $N_P(y, x) = N_{P'}(y, x) - k$), with $k > 0$. Let $Q$ be a set of $m + 1$ voters where: $m + 1 - N_P(x, y)$ voters prefer $x$ to $y$ and $y$ to anyone else; $N_P(x, y)$ voters prefer $y$ to $x$ and $x$ to anyone else. As we have $N_{P\cup Q}(x, y) = N_P(x, y) + N_Q(x, y) = m + 1$; for any $z \neq x, y$, $N_{P\cup Q}(x, z) = N_P(x, z) + m + 1 \geq m + 1$, $x$ 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 7** *If $r$ is a Condorcet-consistent rule then $C(r) \geq \log T(m, p)$.*

¿From Propositions 6 and 7 we get

**Proposition 8** *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 9** $\log T(m, p) \leq \frac{p(p-1)}{2} \log(m + 2)$.

*Proof:* ¿From Lemma 4 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, for every pair $(x, y)$ of distinct candidates such that $x > y$, (a) a single bit indicating whether $N_P(x, y) > N_P(y, x)$ or $N_P(x, y) \leq N_P(y, x)$ and (b) $\min(N_P(x, y), N_P(y, x))$. Since the latter number can vary between $0$ and $\frac{m}{2}$ if $m$ is even, and between $0$ and $\frac{m-1}{2}$ is $m$ is odd, storing this number requires at most $\log\left(\frac{m}{2} + 1\right)$ bits. This makes a total of $1 + \log\left(\frac{m}{2} + 1\right)$ bits, that is, $\log(m + 2)$ bits. We have $\frac{p(p-1)}{2}$ pairs of distinct candidates, hence the result. ■

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 6** *Consider $V_t^s$ the set of vectors of $s$ non-negative integers whose sum is lower or equal to $t$. Then $T(m,p) \geq \mid V_{\frac{m}{2}}^{\frac{p(p-1)}{2}} \mid$.*

*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 $g_{i,j}$ is assumed to contain exactly $2c_{i,j}$ voters and $g_0$ contains the rest of the voters (i.e. $m - \sum_{i<j} 2c_{i,j}$). 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 $x_{p-2} \succ x_{p-1} \succ \ldots \succ x_1 \succ i \succ j$, 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 $x_1 \succ x_2 \succ \ldots \succ x_p$ 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 the previous Lemma, and using a technique similar to the one used in Corollary 2 to enumerate $V_t^s$, we obtain the compilation complexity of this family of rules:

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

## 3.3 Plurality with runoff

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

**Proposition 10** *Let $r_2$ be the plurality-with-runoff rule. $P \sim_{r_2} Q$ holds if and only if for every $x$, $ntop(P,x) = ntop(Q,x)$ and for every $x, y$, $N_P(x,y) = N_Q(x,y)$.*

**Lemma 7** *If for every $x$, $ntop(P,x) = ntop(Q,x)$ and for every $x, y$, $N_P(x,y) = N_Q(x,y)$, then $P \sim_{r_2} Q$.*

*Proof:* For every $x \in X$, since $ntop(P,x) = ntop(Q,x)$, we also have $ntop(P \cup R, x) = ntop(Q \cup R, x)$: the two plurality winners are the same in $P \cup R$ and $Q \cup R$. Let $x$ and $y$ be these two plurality winners. Since $N_P(x,y) = N_Q(x,y)$, we have $N_{P \cup R}(x,y) = N_{Q \cup R}(x,y)$, therefore, $M_{P \cup R}(x,y)$ if and only if $M_{Q \cup R}(x,y)$ and hence $r_2(P \cup R) = r_2(Q \cup R)$. ■

**Lemma 8** *If for some $x$ $ntop(P,x) \neq ntop(Q,x)$, then $P \not\sim_{r_2} Q$.*

*Proof:* If $p = 2$, this is a corollary of Lemma 1. Assume $p \geq 3$, and w.l.o.g., assume $ntop(P,x) > ntop(Q,x)$. Because $\sum_{c \in X} ntop(P,c) = \sum_{c \in X} ntop(Q,c) (= m)$, there exists an $y \neq x$ such that $ntop(P,y) < ntop(Q,y)$. Almost w.l.o.g., assume $x$ has priority over $y$ for tie-breaking[4]. Let $z \neq x, y$ (which is possible because $p \geq 3$). We now construct an $R$ such that in $P \cup R$, the two finalists are $x$ and $z$, and the winner is $x$, and in $Q \cup R$, the two finalists are $y$ and $z$ (therefore the winner cannot be $x$). Let $R$ be the following partial profile containing $14m + ntop(P,y) - ntop(P,x)$ new votes:

| | |
|---|---|
| $ntop(P,y) - ntop(P,x) + 4m$ votes: | $x \succ \ldots$ |
| $4m$ votes: | $y \succ x \succ z \succ \ldots$ |
| $6m$ votes: | $z \succ \ldots$ |

The plurality scores in $P \cup R$ are:

- $s_{P \cup R}(x) = ntop(P,x) + ntop(P,y) - ntop(P,x) + 4m = ntop(P,y) + 4m$;
- $s_{P \cup R}(y) = ntop(P,y) + 4m$;
- $s_{P \cup R}(z) = ntop(P,z) + 6m$.
- for every other candidate $c$, $s_{P \cup R}(c) = ntop(P,c)$.

Since $ntop(P,c) \leq m$ holds for every $c \neq x, y, z$, we have $s_{P \cup R}(z) > s_{P \cup R}(x) = s_{P \cup R}(y) > s_{P \cup R}(c)$ for every $c \neq x, y, z$. Because $x$ has priority over $y$, the two candidates remaining for the second round are $z$ and $x$. Now, the number of voters in $P \cup R$ preferring $x$ to $z$ is $N(P \cup R, x, z) = N(P, x, z) + ntop(P,y) - ntop(P,x) + 8m \geq 8m$ (because $N(P,x,z) \geq ntop(P,x)$); and $N(P \cup R, z, x) = N(P,z,x) + 6m \leq 7m$. Hence $r_2(P \cup R) = x$. The plurality scores in $Q \cup R$ are:

- $s_{Q \cup R}(x) = ntop(Q,x) + ntop(P,y) - ntop(P,x) + 4m > ntop(P,y) + 4m$;
- $s_{Q \cup R}(y) = ntop(Q,y) + 4m$;
- $s_{Q \cup R}(z) = ntop(Q,z) + 6m$.
- for every other candidate $c$, $s_{Q \cup R}(c) = ntop(Q,c)$.

[4]The proof in the opposite case is very similar and we omit it.

$s_{Q\cup R}(y) - s_{Q\cup R}(x) = ntop(Q, y) - ntop(P, y) + ntop(P, x) - ntop(Q, x)$. Now, by assumption we have $ntop(Q, y) > ntop(P, y)$ and $ntop(P, x) > ntop(Q, x)$, therefore $s_{Q\cup R}(y) > s_{Q\cup R}(x)$.

$s_{Q\cup R}(z) - s_{Q\cup R}(x) = ntop(Q, z) - ntop(Q, x) - ntop(P, y) + ntop(P, x) + 2m$. Now, $ntop(P, x) > ntop(Q, x)$, therefore $s_{Q\cup R}(z) - s_{Q\cup R}(x) > ntop(Q, z) - ntop(P, y) + 2m > 0$, that is, $s_{Q\cup R}(y) > s_{Q\cup R}(x)$.

Because the plurality scores of both $y$ and $z$ in $Q \cup R$ are larger than the plurality score of $x$, $x$ does not pass the first round, therefore $r_2(P \cup R) \neq x$. ■

**Lemma 9** *If for some* $x, y \in X$*,* $N(P, x, y) \neq N(Q, x, y)$ *then* $P \not\sim_{r_2} Q$*.*

*Proof:* Assume w.l.o.g. that $N(P, x, y) > N(Q, x, y)$. We are going to 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 $Q \cup R$. Let $R$ be composed of the following $2N(P, y, x) + 3m + 1$ votes:

| | |
|---|---|
| $2N(P, y, x) + m + 1$ votes: | $x \succ y \succ \dots$ |
| $2m$ votes: | $y \succ x \succ \dots$ |

Obviously, the plurality scores in $P \cup R$ verify $s_{P\cup R}(x) > m$, $s_{P\cup R}(y) > m$, and for any $c \neq x, y$, $s_{P\cup R}(c) \leq m$, therefore, the finalists are $x$ and $y$. Things are the same for $Q \cup R$.

Now, $N(P\cup R, x, y) = N(P, x, y)+2N(P, y, x)+m+1 = m+N(P, y, x)+m+1 = N(P, y, x) + 2m + 1$; and $N(P \cup R, x, y) = 2N(P, y, x) + 4m + 1 - N(P \cup R, x, y) = N(P, y, x) + 2m$. Therefore, $r_2(P \cup R) = x$.

Lastly, $N(Q \cup R, x, y) = N(Q, x, y) + 2N(P, y, x) + m + 1$, and $N(Q \cup R, x, y) = N(Q, y, x) + 2m$. We have now $N(Q \cup R, y, x) - N(Q \cup R, x, y) = N(Q, y, x) + m - N(Q, x, y) - 2N(P, y, x) - m - 1 = N(Q, y, x) + m - N(Q, x, y) - 2N(P, x, y) - 1 = 2(N(Q, y, x) - N(P, y, x)) - 1$. Now, $N(Q, y, x) > N(P, y, x)$, therefore, $N(Q \cup R, y, x) > N(Q \cup R, x, y)$, that is, $r_2(Q \cup R) = y$. ■

Proposition 10 is now a corollary from Lemmas 7, 8 and 9, and it follows that:

**Proposition 11** *The compilation complexity of plurality with runoff is* $\log L(m, p) + \log T(m, p)$*.*

## 4 Conclusion

This paper has introduced a notion that we believe to be of primary importance in many practical situations: the compilation of incomplete profiles. In particular, the amount of information that a given polling station needs to transmit to the central authority is a good indicator of the difficulty of the verification process. We have established a general technique which allows us to derive the compilation complexity of a voting rule, and have related it to other issues in communication complexity. We have derived a number of results for specific classes of voting rules. A question that we have only sketched in this paper and 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 as provided by the $m$ voters, it may be more efficient to compile the possible completion of this partial profile together with the associated outc! ome, or, in other words, to compile the function that takes the remaining $k$ profiles as input (there are $(p!)^k$ such inputs) and return the outcome. As there are $(p!)^k$ possible profiles, the number of such functions is $p^{(p!)^k}$. This tells us that a general upper bound for $C(r, k)$ is $\leq (p!)^k. \log p$. Hence, overall we have $C(r, k) \leq \min(m. \log(p!), (p!)^k. \log p)$ (note that the second term becomes interesting only when $m$ is big enough, and $p$ and $k$ small enough).

The general problem of dealing with incomplete profiles opens a host of related questions, for instance the probability that a voting process could be stopped after only $m$ voters have expressed their opinions, or (a closely related question), the probability that the central authority would make a mistake were it forced to commit on a winner in situations where no candidate is yet guaranteed to prevail. Another interesting issue for further research would consist in designing new ways of computing NP-hard voting rules using an off-line compilation step so that their on-line computation time becomes polynomial in the size of the initial profile. This, of course, implies that the size of $\sigma(P)$ may be much larger (possibly exponentially) than the size of the input, which means that the computation of $\rho$ may be logarithmic in the size of the compilation $\sigma(P)$.

## References

[1] E. Bender and S. Williamson. *The Foundations of Combinatorics with Applications*. Dover, 2006.

[2] S. Brams, D. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

[3] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf. Feasibility and unfeasibility of off-line processing. In *ISTCS*, 1996.

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

[5] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proceedings of EC-05*, 2005.

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

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

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

[9] M.S. Pini, F. Rossi, K. Brent Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *IJCAI*, 2007.

[10] T. Walsh. Complexity issues in preference elicitation and manipulation. In *AAMAS*, 2008.

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