scientific reports

Check for updates

OPEN

Computations of volumes in five candidates elections

Winfried Bruns[1]✉ & Bogdan Ichim[2,3]

We describe several analytical (i.e., precise) results obtained in five candidates social choice elections under the assumption of the Impartial Anonymous Culture. These include the Condorcet and Borda paradoxes, as well as the Condorcet efficiency of plurality, negative plurality and Borda voting, including their runoff versions. The computations are done by Normaliz. It finds precise probabilities as volumes of polytopes in dimension 119, using its recent implementation of the Lawrence algorithm.

In[1], p. 382 Lepelley, Louichi and Smaoui state:

> "Consequently, it is not possible to analyze four candidate elections, where the total number of variables (possible preference rankings) is 24. We hope that further developments of these algorithms will enable the overcoming of this difficulty."

This hope has been fulfilled by previous versions of Normaliz[2]. In connection with the symmetrization suggested by Schürmann[3], it was possible to compute volumes and Ehrhart series for many voting events in four candidates elections; see[4]. As far as Ehrhart series are concerned, we cannot yet offer progress. But the volume computation was already substantially improved by the descent algorithm described in[5]. Examples of Normaliz being used for voting theory computations by independent authors can be found in[6,7] and[8]. The purpose of this paper is to present *precise* probability computations in five candidates elections under the assumption of the Impartial Anonymous Culture (IAC). They are made possible by Normaliz' implementation of the Lawrence algorithm[9].

The connection between rational polytopes and social choice was established independently in[1] and[10]. Solutions for the four candidates quest were proposed for example in[3,4] and[5]. The similar, but much more challenging computational problem of performing precise computations in five candidates elections is wide open. Various authors have used the well known Monte Carlo methods in order to perform computations with five or more candidates, but fundamentally these methods can only deliver *approximative* results, without even clear bounds for errors. We note that methods that were successful in obtaining precise results in the four candidates case are ineffective in the five candidates case due to the huge leap in computational complexity implied by the increase in the dimension of the associated polytopes (from 23 to 119). Therefore a different algorithmic approach is needed in order to obtain the desired precise results.

To the best of our knowledge, we present here the first precise results obtained for computations with five candidates. By precise we mean either absolutely precise rational numbers, or results obtained using the fixed precision mode of Normaliz where the desired precision is set and fully controlled by the user.

The polytopes in five candidates elections have dimension 119, and are defined as subpolytopes of the simplex spanned by the unit vectors of $\mathbb{R}^{120}$. The number of the inequalities cutting out the subpolytope is the critical size parameter, but fortunately we could manage computations with ≤ 8 inequalities (in addition to the 120 sign inequalities) on the hardware at our disposal, although the algorithm allows an arbitrary number of inequalities. This covers the Condorcet paradox[11] (computable on a laptop in a few minutes), the Borda winner and loser paradoxes[12], and the Condorcet efficiency of plurality, negative plurality and Borda voting, including their runoff extensions. We also compute the probabilities of all 12 configurations of the five candidates that are defined by the Condorcet majority relation.

As Table 6 shows, the computations for 5 candidates are very demanding on the hardware in memory and computation time. Therefore we consider it a major value of the new algorithm that it improves the situation in four candidates elections considerably, where it is now possible to allow preference rankings with all types of partial indifference. Moreover one can run series of parameterized computations for four candidates like those that one finds in[13] for three candidates. In order to illustrate this possibility we compute the probability of the Condorcet paradox in the presence of voters with indifference and the Condorcet efficiency of approval voting

[1]Institut für Mathematik, Universität Osnabrück, 49069 Osnabrück, Germany. [2]Faculty of Mathematics and Computer Science, University of Bucharest, Str. Academiei 14, 010014 Bucharest, Romania. [3]Simion Stoilow Institute of Mathematics of the Romanian Academy, Research Unit 2, C.P. 1-764, 010702 Bucharest, Romania. ✉email: wbruns@uni-osnabrueck.de

(see Sect. "Indifference"). Note that potential applications are not only limited to voting theory, as can be seen in[14], Table 3. There the new algorithm is performing better (as the dimension grows) for the first family of examples.

Normaliz computes lattice normalized volume and uses only rational arithmetic without rounding errors or numerical instability. But there is a slight restriction: while it is always theoretically possible to compute the probabilities as absolutely precise rational numbers, the fractions involved can reach sizes which are unmanageable on the available hardware. For these cases Normaliz offers a fixed precision mode whose results are precise up to an error with a controlled bound that can be set by the user.

In contrast to algorithms that are based on explicit or implicit triangulations of the polytope P (or the cone $C(P)$ defined by P) under consideration, the Lawrence algorithm uses a "generic triangulation" of the dual cone $C(P)^*$. We make a brief discussion of the available Lawrence algorithm implementations and their limitations in Section "Implementations of the Lawrence algorithm and their limitations". In order to reach the order of magnitude that is necessary for five candidates elections, one needs a fine tuned implementation. It is outlined in[15]. Moreover, the largest of our computations need a high performance cluster to finish in acceptable time. Section "Computational report" gives an impression on the computation times and memory requirements by listing them for selected examples.

The computations that we report in this note were done by version 3.9.0 of Normaliz. Meanwhile it has been succeeded by version 3.10.1 without changes in the Lawrence algorithm. Both versions are available at https://www.normaliz.uni-osnabrueck.de/.

For details on the implementation and the performance of the previous versions of Normaliz we point the reader to[16–19].

A challenging computational problem arising from social choice

Voting schemes and rational polytopes. The connection between voting schemes and rational polytopes is based on counting integral points in the latter. In this subsection we sketch the connection. As a general reference for discrete convex geometry we recommend[20]. The interested reader may also consult[21] and[22].

The basic assumption in the mathematics of social choice is the existence of *individual preference rankings* $\succ$: every voter ranks the candidates in linear order. Examples for three candidates named by capital letters:

$$A \succ B \succ C, \qquad C \succ A \succ B.$$

For n candidates there exist $N = n!$ preference rankings, usually numbered in lexicographic order. (By an extension it is possible to allow indifferences; for example see[13].)

The *result* or *profile* of the election is the N-tuple

$$(x_1, \dots, x_N), \qquad x_i = \#\{\text{voters of preference ranking } i\}.$$

Thus an election result for three candidates may be written in the following tabular form:

Number of voters	x_1	x_2	x_3	x_4	x_5	x_6
Ranking	A	A	B	B	C	C
	B	C	A	C	A	B
	C	B	C	A	B	A

In the following we want to compute probabilities of certain events related to election schemes. This requires a probability distribution on the set of election results. The *Impartial Anonymous Culture (IAC)* assumes that all election results for a fixed number of voters, in the following denoted by k, have equal probability. In other words, it is the equidistribution on the set of voting profiles for a fixed number of k voters.

The *Marquis de Condorcet* (1743–1794) was a leading intellectual in France before and during the revolution. He already observed that there is no ideal election scheme, a fact now most distinctly manifested by Arrow's impossibility theorem. We say that candidate A *beats* candidate B in *majority*, $A >_M B$, if

$$\#\{\text{voters with } A \succ B\} > \#\{\text{voters with } B \succ A\}.$$

A (necessarily unique) *Condorcet winner* (CW) beats all other candidates in majority. There is general agreement that the CW is the person with the largest common approval. However, Condorcet realized that a CW need not exist: the relation $>_M$ is not transitive: a minimal example is the profile $(1, 0, 0, 1, 1, 0)$. This phenomenon is called the *Condorcet paradox*. From a quantitative viewpoint, the most ambitious goal is to find the exact number of election profiles exhibiting the Condorcet paradox (or the opposite), given the number of voters k. For large k, this number is gigantic. It is much more informative to understand the behavior for $k \to \infty$: what is the probability that an election result exhibits the Condorcet paradox? Since we assume the IAC, this probability is

$$\lim_{k\to\infty} \frac{\#\{\text{election results without CW for } k \text{ voters}\}}{\#\{\text{all election results for } k \text{ voters}\}}.$$

It is a crucial consequence of (IAC) that the event "A is the CW" can be characterized by a system of homogeneous linear inequalities. For three candidates they are

$$\begin{aligned} A >_M B: &\quad x_1 + x_2 + x_5 > x_3 + x_4 + x_6, \\ A >_M C: &\quad x_1 + x_2 + x_3 > x_4 + x_5 + x_6. \end{aligned}$$

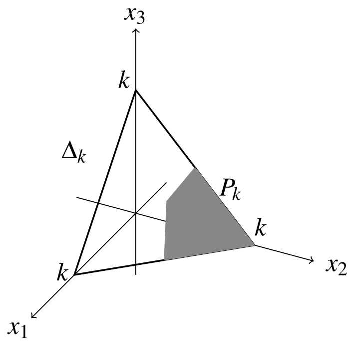

Figure 1. Subpolytope defined by linear inequalities.

If we are only interested in probabilities for $k \to \infty$, standard arguments of measure theory allow ties and replacement of $>$ by $\geq$.

We now consider an event E defined for an n candidates election by a system of homogeneous linear inequalities on the set of election profiles. As above, set $N = n!$. The election profiles $(x_1, \dots, x_N)$ are the lattice points (points with integral coordinates) in the positive orthant $\mathbb{R}^N_+$ satisfying the equation $x_1 + \cdots + x_N = k$. The real points in the positive orthant satisfying this equation form a polytope Δ_k, and the linear inequalities whose validity defines E cut out a subpolytope P_k. We illustrate this assertion by the (necessarily unrealistic) Fig. 1.

For large numbers of voters we want to find the probability $\mathrm{prob}(E)$ of the event E. Under (IAC) it is given by

$$\mathrm{prob}(E) = \lim_{k \to \infty} \frac{\#\{\text{lattice points in } P_k\}}{\#\{\text{lattice points in } \Delta_k\}}.$$

We project Δ_k orthogonally onto Δ_1, and thus P_k onto P_1. The density, roughly speaking, of the projections of the lattice points converges to 1, and therefore

$$\mathrm{prob}(E) = \frac{\mathrm{vol}(P_1)}{\mathrm{vol}(\Delta_1)}.$$

For volume computations in connection with the counting of lattice points one uses the *lattice normalized volume* vol, giving volume 1 to Δ_1. With this choice $\mathrm{prob}(E) = \mathrm{vol}(P_1)$.

It is not difficult, but would take many pages, to write down the linear inequalities for the voting schemes and events discussed in the following. For four candidates the complete systems are contained in[4]. For the inequalities one must often fix the roles that certain candidates play, like the Condorcet winner A above. Then probabilities must be computed carefully, and this may require the inclusion-exclusion principle.

Both from the theoretical as well as from the computational viewpoint it is better to consider the cone C defined by the homogeneous linear inequalities as the prime object, and the polytopes as intersections of C with the hyperplane defined by the equation $x_1 + \cdots + x_N = k$.

It is not difficult to see that a voting event that can be realized by a voting profile has positive probability:

Proposition 1 *Let $\mathscr{E}$ be a subset of all voting profiles defined by strict homogeneous rational inequalities. If $\mathscr{E}$ is nonempty, then it has probability > 0 under (IAC).*

Proof Clearing denominators, one can assume that the coefficients of the inequalities are integers. Let m be the maximum of all their absolute values and $x \in \mathscr{E}$ be a voting profile. Then $x' = (m+1)x \in \mathscr{E}$ as well by homogeneity. It is easily checked that also $x' + e_i \in \mathscr{E}$ where $e_i, i = 1, \dots, N$ is the i-th unit vector. The parallel translation by $-x'$ maps the the polytope P spanned by the $x' + e_i$ bijectively onto Δ_1. Thus P has lattice normalized volume 1, and therefore its orthogonal projection to Δ_1 has positive volume. □

The Condorcet paradox in five candidates elections. The Condorcet paradox, introduced in Section "Voting schemes and rational polytopes", does not occur in the case of two candidates (if draws are excluded). For three candidates the exact probability of an outcome with a Condorcet winner (under IAC) was first computed by Gehrlein and Fishburn[23] while for four candidates it was first determined by Gehrlein in[24].

For five candidates, we have computed in the full precision mode of Normaliz (and the method presented in Section "Implementations of the Lawrence algorithm and their limitations") that

$$p_{\mathrm{CW}} = \frac{a}{b},$$

where

$$a = 76079454795886424149640859153101819802148488422934611165823661592993 5$$

and

$$b = 10108272625512143586304015110040282491020841362579353564838402 64634368.$$

In decimal notation with 100 decimals, we obtain

$$\begin{aligned} p_{\mathrm{CW}} \approx\ & 0.7526454579773643442763921933175624727126581336541 0\\ & 1822868446458340097032754336154259246534570919500 8. \end{aligned}$$

In order to illustrate the fixed precision mode of Normaliz, we compare the above exact result with the result obtained for fixed precision of 100 decimal digits, namely

$$p_{\mathrm{CW}} \approx \frac{a'}{b'},$$

where

$$\begin{aligned} a' =\ & 940806822471705430345490241646953090890822667067627278585558072925121\\ & 29094292019282405816821364997 \end{aligned}$$

and

$$\begin{aligned} b' =\ & 125000000000000000000000000000000000000000000000000000000000000000000\\ & 000000000000000000000000000000. \end{aligned}$$

In decimal notation with 100 decimals, we obtain

$$\begin{aligned} p_{\mathrm{CW}} \approx\ & 0.7526454579773643442763921933175624727126581336541 0\\ & 1822868446458340097032754336154259246534570919976 0. \end{aligned}$$

The reader should observe that in the decimal notation only the last 4 digits are different. The error bound is

$$6572904 \cdot 10^{-100} < 10^{-93},$$

where 6, 572, 904 is the size of the "generic triangulation" (see Section "Implementations of the Lawrence algorithm and their limitations" and Table 5).

This means that using the fixed precision mode of Normaliz is sufficient for many applications, while it saves computation time and is significantly less demanding on the hardware.

For practical reasons, in the following we use shorter decimal representations of the rational numbers. (The full rational representations of these numbers are available on demand from the authors.) A decimal representation is called *rounded to n decimals* when the first $n-1$ printed decimals are exact and only the last decimal may be rounded up.

Rule versus rule runoff, Condorcet efficiencies. The most common voting scheme in elections is the *plurality rule PR*: for each candidate X one counts the voters that have X on first place in their preference ranking, and the winner is the candidate with most first places. However, in many elections one uses a second ballot, called *runoff*, if the winner has not got the votes of more than half of the voters. In the runoff only two candidates are left, namely the two top candidates of the first round. A typical example is the French presidential election.

If the ideal winner of an election is the Condorcet winner CW, then one must ask for the probability that the plurality winner is the CW under the condition that a CW exists. This conditional probability is called the *Condorcet efficiency*, studied intensively by Gehrlein and Lepelley[21] as a quality measure for voting schemes.

Another important question is whether the runoff is a real improvement: (i) what is the probability that the winner of the first ballot also wins the second, and (ii) by how much does the Condorcet efficiency increase by the runoff.

An often discussed variant of plurality is *negative plurality NPR*: the winner is the least disliked candidate X, defined by the least number of voters who have placed X on the last place in their preference ranking. As for plurality one can have a runoff, and again it makes sense to compute the Conndorcet efficiencies and the probability that the first round winner also wins the runoff.

Both plurality and negative plurality are special cases of weighted voting schemes in which the places in the preference ranking have a fixed weight, and every candidate is counted with the sum of the weights in the preference ranking of the voters. In plurality the first place has weight 1 and the other places have weight 0, wheres negative plurality gives weight -1 to the last place. In addition to these two rules we discuss the *Borda rule BR* that for n candidates gives weight $n-p$ to place p.

In the case of four candidates the plurality voting versus plurality runoff problem was first computed by De Loera, Dutra, Köppe, Moreinis, Pinto and Wu in[25] using LattE Integrale[26] for the volume computation.The Condorcet efficiency of plurality voting was first computed by Schürmann in[3], whereas the Condorcet efficiency of the runoff plurality voting was given in[4]. According to[22], it was obtained independently in[4] and[27]. In Section 6[5] we additionally discuss the influence of a third ballot on the Condorcet efficiencies of plurality and negative plurality.

Our results for five candidates are listed in Table 1. The first line contains the probability that the first round winner also wins the runoff. These three computations were done using the full precision mode of Normaliz. The next two lines contain the Condorcet efficiencies, computed the fixed precision mode of Normaliz. For practical reasons we have only included the results rounded to 15 decimals.

In Table 2 we reproduce the results for the Condorcet efficiency of all three rules contained in Table 7.6 of[22], which were obtained using Monte Carlo methods in[28].

	Rule *R*		
	PR	*NPR*	*BR*
`RVsRunoff`	0.673383666340974	0.614598375568014	0.769395916647461
`CondEffR`	0.614270758198443	0.509039971570300	0.854442922091020
`CondEffRRunoff`	0.832220522376460	0.775488383677566	0.991189085613331

Table 1. Probabilities computed by Normaliz.

	Rule *R*		
	PR	*NPR*	*BR*
`CondEffR`	0.6139	0.5090	0.8541

Table 2. Results obtained by Monte Carlo, according to[22] and[28].

The numbers are relatively close, which confirms the correctness of all algorithms involved. However, at least 14 decimals printed in Table 1 are exact, while for the numbers printed in Table 2 we have 2, 4 and 3 exact decimals.

Strong Borda paradoxes.

The Borda paradoxes are named after the Chevalier de Borda who studied them in[12]. The *strict Borda paradox* is the event that for a voting profile plurality and majority rank the candidate in opposite order. A less sharp paradox is the *strong Borda paradox*: the plurality winner is the Condorcet loser, and the *reverse strong Borda paradox* occurs if the Condorcet winner finishes last in plurality. These paradoxes can be discussed for all voting schemes for which every profile defines a linear order of the candidates. There is however no point in computing them for negative plurality. As shown in Section 2.5[4] plurality and negative plurality are dual to each other: the strong Borda paradox and the reverse strong Borda paradox exchange their roles.

For three candidates elections a detailed study of the family of Borda paradoxes[12] is contained in[29], while the case of four candidates is discussed in Section 2.5[4]. According to[22], similar results were obtained independently in[27].

For the time being, the computation of the strict Borda paradox in the case of five candidates seems not to be reachable. The strong paradoxes have been computed in the fixed precision mode of Normaliz. The results are rounded to 15 decimals.

For large numbers of voters the probability of the strong Borda paradox is

$$B_{\mathrm{Sg}} \approx 0.018125801480904$$

and the probability of the reverse strong Borda paradox is

$$B_{\mathrm{SgRev}} \approx 0.019238302806489.$$

Indifference.

We want to point out that the Normaliz implementation of Lawrence's algorithm does not only yield precise results in five candidates elections, but also extends the range of computations for four candidates considerably by allowing preference rankings with partial indifference that increase the dimension of the related polytopes considerably. We demonstrate this by two examples.

In the examples we allow all possible types of indifference except the equal ranking of all candidates: no indifference, equal ranking of two candidates in three possible positions (top, middle, bottom), two groups of two equally ranked candidates, and equal ranking of three candidates (top and bottom). In total one obtains 74 rankings. Compared to the 24 rankings without indifference this is a substantial increase in dimension. We assume that all rankings have the same probability. The authors of[13] allow weights for the types of indifference, for example that the number of voters with a linear order of the candidates is twice the number of voters with indifference. Such weights can easily be realized as a system of homogeneous linear equations in the Normaliz input file.

The first computation is the probability of a Condorcet winner under the Extended Impartial Anonymous Culture (EIAC), as discussed in[13] for 3 candidates (and varying weights for the different types of indifference). This requires only 3 inequalities to fix the Condorcet winner, and the computation is very fast. We obtained the value of

$$0.884041566089553$$

for the probability of the existence of a Condorcet winner under EIAC (rounded to 15 decimals).

The second example is the Condorcet efficiency of approval voting. Under this rule one additionally assumes that every voter casts a vote for each candidate on first place in his or her preference ranking. This requires 6 inequalities, namely 3 to mark the CW and 3 to make the same candidate the winner of the approval voting.

	3 cand	4 cand	5 cand
`Condorcet Par`	0.9375	0.8384	0.7526
`PR vs RunO`	0.8767	0.7545	0.6734
`CondEff PR`	0.8815	0.7426	0.6143
`CondEff PR RunO`	0.9685	0.9117	0.8322
`NPR vs RunO`	0.6389	0.6227	0.6146
`CondEff NPR`	0.6296	0.5516	0.5090
`CondEff NPR RunO`	0.9704	0.8450	0.7755
`BR vs RunO`	0.8750	0.8053	0.7694
`CondEff BR`	0.9111	0.8706	0.8544
`CondEff BR RunO`	1.0000	0.9962	0.9912
`Strong Borda Par`	0.0296	0.0227	0.0181
`Strong RevBorda Par`	0.0315	0.0238	0.0192

Table 3. Probabilities of voting events for 3, 4 and 5 candidates.

Consequently the computation time is going up considerably. See the data for `CondEffAppr 4cand` in Table 6. Normaliz obtains

$$0.695293409282039$$

as the probability that there exists a CW who finishes first in the approval voting. This yields the Condorcet efficiency of

$$0.786494024661739$$

for approval voting (under the assumptions above). The computations were done using the full precision mode of Normaliz.

From three to five candidates. In Table 3 we give an overview of the probabilities of voting events for three, four and five candidates as far as we have computed them for five candidates. We use the shorthands *PR*, *NPR* and *BR* for the plurality rule, negative plurality rule and Borda rule as introduced above. The remaining abbreviations are self explanatory. For better overview we have rounded all probabilities to 4 decimals.

One observes that all probabilities are decreasing from three to five candidates. This reflects the increase in the number of configurations defined by the voting profiles. The Condorcet efficiencies and the probabilities of the Borda paradoxes are conditioned on the probabilities of the existence of a Condorcet winner, which itself is decreasing. But this does not compensate the decrease of the absolute probabilities.

In view of our observations above it is justified to formulate.

Conjecture 2 *All series of probabilities associated to voting events in Table 3 are monotonically decreasing with the number of candidates n.*

Condorcet classes

A voting outcome without ties imposes an asymmetric binary relation on the n candidates that we call a *Condorcet configuration*. A Condorcet configuration is also called a *dominance relation*, according to[30]. Evidently there are $2^{\binom{n}{2}}$ such configurations. The permutation group S_n acts on the set of configurations by permuting the candidates. We call the orbits of this action *Condorcet classes*. For $n = 4$ the classes and their probabilities are discussed in[4].

From the graph theoretical viewpoint the Condorcet configurations are nothing but simple directed complete graphs with n labeled vertices, i.e., graphs with n labeled vertices without loops, in which each two vertices are connected by a single directed edge. These graphs are also know as *tournament graphs*.

In this section we present the precise probabilities of the Condorcet classes under IAC. First we make a presentations of the classes, which is needed in order to understand a reduction critical to be made for the computations to be successful.

For $n = 5$ these Condorcet configurations fall into 12 classes under the action of the group S_5. There are 6 classes that have a Condorcet winner (CW) or a Condorcet loser (CL):

LinOrd	CW4cyc
CW2nd3cyc	3cyc4thCL
CW3cycCL	4cycCL

here "cyc" stands for "cycle". For example, CW2nd3cyc denotes the class that has a Condorcet winner, a candidate in second position majorizing the remaining three, and the latter are ordered in a 3-cycle.

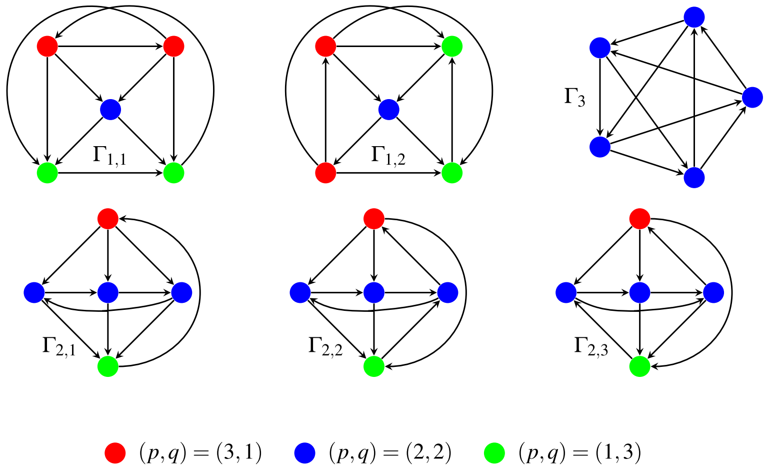

Figure 2. The Condorcet classes without a Condorcet winner or loser.

Class	#config	p(class)
LinOrd	120	0.533665
CW2nd3cyc	40	0.066882
CW3cycCL	40	0.069984
CW4cyc	120	0.082115
3cyc4thCL	40	0.066882
4cycCL	120	0.082115
$\Gamma_{1,1}$	120	0.031467
$\Gamma_{1,2}$	120	0.032172
$\Gamma_{2,1}$	40	0.004509
$\Gamma_{2,2}$	120	0.014644
$\Gamma_{2,3}$	120	0.014203
Γ_3	24	0.001362

Table 4. Condorcet classes, their cardinalities and probabilities.

There are 6 further classes as has been known for a long time. Presumably Davis[31] is the oldest source. (For more sources and cardinalities of the set of classes see[32].) The classes can be structured by the *signatures* (p, q) of a candidate in which p counts the candidates majorized by the chosen candidate and $q = n - 1 - p$ is the number of the candidates majorizing the chosen one. In graph theoretical language, p is the in-degree and q is the out-degree of the chosen node. Without a CW or CL, the signatures (4, 0) and (0, 4) are excluded. The number of signatures (2, 2) must now be odd, and using this observation one easily finds the 6 classes without a CW or CL. They are named in Fig. 2. In the figure candidates of signature (3, 1) are colored red, those of signature (2, 2) are blue, and green indicates the signature (1, 3).

The cardinalities of all classes and their probabilities (rounded to 6 decimals) are listed in Table 4.

We have computed these probabilities not only for aesthetic reasons: that they sum to 1 is an excellent test for the correctness of the algorithm.

For effective computations the following reduction is critical. At first it seems that one must use 10 inequalities representing the relation $>_M$ between the five candidates in addition to the 120 sign inequalities in order to compute the probability of a single class (or configuration). But computations with 130 inequalities are currently not reachable on the hardware at our disposal. Some observations help to reduce the number of inequalities, significantly easing the computational load. For example, LinOrd can be (and is) computed with 128 inequalities if one exploits that it is enough to choose the first two in arbitrary order and the candidate for third place. Once the probability of LinOrd is known, the remaining 5 classes with a CW or CL can be obtained from the Condorcet paradox (124 inequalities), CWand2nd (126), CWandCL (127) and the symmetry between CW and CL (see[4]).

For the other 6 classes it is best to "relax" the direction of some edges and to count which configurations occur if one chooses directions for the relaxed edges. For a proper choice of relaxed edges one gets away with 127 inequalities for $\Gamma_{1,1}$ and only 126 or 125 inequalities for the remaining cases.

It is no surprise that all Condorcet classes have positive probability. In fact, by a theorem of[33] also see Theorem 3.1[30] all Condorcet configurations can be realized by a voting profile. So Proposition 1 implies positive probability.

The problem of finding the minimal number of voters that are necessary to realize a given Condorcet configuration or even a voting event is largely unknown; see[34] for an asymptotic lower bound. Some values for four candidates elections have been computed by Normaliz; see[4], Remark 8.

Implementations of the Lawrence algorithm and their limitations

The Lawrence algorithm is based on the fact that a "signed decomposition" into simplicies of the polytope in the primal space may be obtained from a "generic triangulation" Δ of its dual cone. For each $\delta \in \Delta$ we get a simplex R_δ in the primal space and the volume of the polytope in the primal space is the sum of volumes of simplices R_δ induced by the "generic triangulation" with appropriate signs $e(\delta) = \pm 1$. Thus the following formula can be used for computing the volume of P:

$$\mathrm{vol} P = \sum_{\delta \in \Delta} e(\delta) \mathrm{vol} R_\delta. \tag{1}$$

For mathematical details we refer the reader to Filliman[35]. Details of its implementation in Normaliz are described in[15].

In order to compute a "generic triangulation", Normaliz, following Lawrence's suggestion, finds a "generic element" ω, which in turn induces the "generic triangulation" $\Delta = \Delta_\omega$. Since ω almost inevitably has unpleasantly large coordinates, the induced simplices R_δ have even worse rational vertices, and their volumes usually are rational numbers with very large numerators and denominators. This extreme arithmetical complexity makes computations with full precision sometimes very difficult on the hardware at our disposal. In the fixed precision mode the volumes $\mathrm{vol} R_\delta$ are computed precisely as rational numbers. But the addition of these numbers may result in gigabytes filling fractions. Therefore in order to make computations feasible the precise rational numbers are truncated to a predetermined set of exact decimal digits, which is typically 100 digits. Then the error is bounded above by $T \cdot 10^{-100}$ where T is the size of the "generic triangulation" (i.e. the total number of simplices).

Remark 3 Before Normaliz, the program vinci[36] has provided an implementation of the Lawrence algorithm using floating point arithmetic. As it is noted by the authors in[37], their floating point implementation is numerically unstable. We point out at least one possible reason for this problem, which is indicated by the above discussion.

In any implementation of the Lawrence algorithm the alternating sum 1 must be evaluated. When using floating point arithmetic for subtracting nearby quantities it is possible that the most significant digits are equal and they will cancel each other. This is a severe limitation of the floating point arithmetic that may lead to a phenomenon known as "catastrophic cancelation". It is a fact that, because of the relative error involved, the evaluation of a single subtraction in floating point arithmetic could produce completely meaningless digits.

This problem is visible already when computing voting problems with 4 candidates and only becomes worse for 5 candidates. Consider the problem of comparing 4 voting rules for 4 candidates as it is presented in detail in[5], Sect. 6.1. With its HOT algorithm vinci computes the precise associated Euclidean volume of $1.260510232743 \cdot 10^{-25}$. At the same time, a computation with the Lawrence algorithm as it is implemented in vinci provides the erroneous value of $9.287423132835 \cdot 10^{-8}$ for the same volume. So is clear that the results provided by the vinci implementation of the Lawrence algorithm may lack any kind of precision, therefore it does not make sense to include in this paper a benchmark of the (different) implementation of the Lawrence algorithm in vinci.

Remark 4 The program polymake[38] has also implemented a simplified version of the Lawrence's algorithm. This implementation is restricted to the "smooth" case. Note that smooth implies "simple", which in turn implies that the dual polytope is "simplicial", so its boundary has a trivial triangulation. The polytopes that appear in voting theory are not smooth, in fact they are not even simple. Thus the implementation in polymake of the Lawrence algorithm cannot be compared with the Normaliz implementation for the polytopes presented here.

Computational report

Selected examples.

In order to give the reader an impression of the computational effort, we illustrate it by the data of several selected examples. Except (1) and (2) they are all computations for elections with 5 candidates:

(1) `strictBorda 4cand` is the computation of the probability of the strict Borda paradox for elections with 4 candidates as discussed in[4].
(2) `CondEffAppr 4cand` is the Condorcet efficiency of approval voting for 4 candidates.
(3) `Condorcet` stands for the existence of a Condorcet winner in elections with 5 candidates.
(4) `PlurVsRunoff` computes the probability that the plurality winner also wins the runoff.
(5) `CWand2nd` computes the probability that there exists Condorcet winner and a second candidate dominating the remaining three.
(6) `CondEffPlurRunoff` is used to compute the probability that the Condorcet winner exists and finishes at least second in plurality.
(7) `CondEffPlur` computes the probability that the Condorcet winner exists and wins plurality.

In all cases one has to make choices for the candidates that have certain roles in the computation in order to define the polytope for the computation. Table 5 contains their characteristic combinatorial data.

	dim C	# inequalities	# triangulation	# generic triang
`strictBorda 4cand`	24	33	100,738	324,862
`CondEffAppr 4cand`	74	80	1,620,052	30,564,920
`Condorcet`	120	124	137,105	6,572,904
`PlurVsRunoff`	120	125	4,912,369	93,749,784
`CWand2nd`	120	126	15,529,730	608,572,514
`CondEffPlurRunoff`	120	127	246,310,369	5,456,573,880
`CondEffPlur`	120	128	2,388,564,481	39,390,184,920

Table 5. Combinatorial data.

Parallelized and distributed volume computations. The implementation in Normaliz of the Lawrence algorithm consists of 4 distinct steps that are described in[15]. For effective computations these steps can be separated (and sometimes they must be separated) and run on different machines.

The computation times in Table 6 are "wall clock times" taken on a Dell R640 system with 1 TB of RAM and two Intel™Xeon™Gold 6152 (a total of 44 cores) using 32 parallel threads (of the maximum of 88).

Additional information:

(1) All computations in the table use 64 bit integers for steps (1)–(3). Even step (4) is done with 64 bit integers for `strictBorda 4cand` and `Condorcet`.
(2) The volumes of the first 5 polytopes were computed with full precision, whereas for `CondEffPlur` and `CondEffPlurRunoff` fixed precision was used.
(3) The following rule of thumb can be used to estimate the computation time for a smaller number of threads: if one reduces the number of parallel threads from 32 to 8, then one should expect the computation time to go up by a factor of 3. A further reduction to 1 thread increases it by another factor of 7.
(4) From the selected examples, only `strictBorda` is computable with the algorithms previously implemented in Normaliz. For this example, the data in Table 6 may be compared with the data in[5], Table 2 which was recorded on the same system.
(5) The data in Table 6 shows why computations with more than 128 inequalities are currently not reachable on the hardware at our disposal. Each additional inequality added leads to a significant jump in the required RAM memory and there exists a 1 TB limit on our system.

Stage (4) of the last two polytopes was computed on a high performance cluster (HPC) because the computation time would become extremely long on the R640, despite of the high degree of internal parallelization. The time for `CondEffPlurRunoff` would still be acceptable, but `CondEffPlur` would take several weeks. Instead doing step (4) directly, the result of steps (1)–(3) is written to a series of compressed files on the hard disk. Each of these files contains a certain number of simplices and this number can be chosen by the user, for example 10^6 simplices. For `CondEffPlur` we need 12, 277 s for writing the input files of the distributed computation, and `CondEffPlurRunoff` needs 528 s.

The compressed files are then collected and transferred to the HPC. The Osnabrück HPC has 51 nodes, each equipped with 1 TB of RAM and 2 AMD Epyc 7742 so that 128 threads can be run on each node. In our setup each node ran 16 instances of `chunk` simultaneously and every instance used 8 threads of OpenMP parallelization. Consequently 816 input files could be processed simultaneously. For a `CondEffPlur` input file of 10^6 simplices one needs about 165 MB of RAM and 3 h of computation time. Therefore the volume of `CondEffPlur` could be computed in ≈ 9 h.

Even on a less powerful system it can be advisable to choose this type of approach since one loses only a small amount of data when a system crash should happen and the amount of memory used remains low. Also "small" computations can profit from fixed precision. For example, step (4) of `Condorcet` takes 13.9 s with fixed precision, but 52.5 s with full precision.

		Time		
	RAM in GB	Stages (1)–(3)	Stage (4)	Total
`strictBorda 4cand`	0.35	1.278 s	0.464 s	1.742 s
`CondEffAppr 4cand`	7.4	97.8 s	14:31 m	16:09 m
`Condorcet`	1.67	18.0 s	52.493 s	1:10 m
`PlurVsRunoff`	26.2	12:40 m	1:29:21 h	1:42:01 s
`CWand2nd`	56.4	49:55 m	10:21:36 h	11:11:31 h
`CondEffPlurRunoff`	113	13:30:22 h	HPC	–
`yCondEffPlur`	646	125:27:20 h	HPC	–

Table 6. Memory usage and times for parallelized volume computations.

Data availability

Input and output files for all computations of this paper can be found at https://www.normaliz.uni-osnabrueck.de/documentation/interesting-and-challenging-examples-for-normaliz/.

Received: 15 March 2023; Accepted: 28 July 2023
Published online: 15 August 2023

References

1. Lepelley, D., Louichi, A. & Smaoui, H. On Ehrhart polynomials and probability calculations in voting theory. *Soc. Choice Welf.* **30**, 363–383 (2008).
2. Bruns, W., Ichim, B., Söger, C., & von der Ohe, U. *Normaliz. Algorithms for rational cones and affine monoids*. https://normaliz.uos.de.
3. Schürmann, A. Exploiting polyhedral symmetries in social choice. *Soc. Choice Welf.* **40**, 1097–1110 (2013).
4. Bruns, W., Ichim, B. & Söger, C. Computations of volumes and Ehrhart series in four candidates elections. *Ann. Oper. Res.* **280**, 241–265 (2019).
5. Bruns, W. & Ichim, B. Polytope volume by descent in the face lattice and applications in social choice. *Math. Program. Comput.* **13**, 415–442 (2021).
6. Brandt, F., Geist, C., & Strobel, M. Analyzing the practical relevance of voting paradoxes via Ehrhart theory, computer simulations, and empirical data. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems* 385–393.
7. Brandt, F., Hofbauer, J., & Strobel, M. Exploring the no-show paradox for condorcet extensions using Ehrhart theory and computer simulations. In *Proceedings of the 2019 International Conference on Autonomous Agents & Multiagent Systems* 520–528.
8. Diss, M., Kamwa, E. & Tlidi, A. On some k-scoring rules for committee elections: Agreement and Condorcet Principle. *Revue d'Écon. Polit.* **130**, 699–725 (2020).
9. Lawrence, J. Polytope volume computation. *Math. Comput.* **57**, 259–271 (1991).
10. Wilson, M. C. & Pritchard, G. Probability calculations under the IAC hypothesis. *Math. Soc. Sci.* **54**, 244–256 (2007).
11. de Condorcet, N. Marquis. *Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix* (Imprimerie Royale, Paris, 1785).
12. de Borda, J.-C. Chevalier. Mémoire sur les élections au scrutin. *Histoire de'Académie Royale Des Sci.* **102**, 657–665 (1781).
13. Friese, E., Gehrlein, W. V., Lepelley, D. & Schürmann, A. The impact of dependence among voters' preferences with partial indifference. *Qual. Quant.* **51**, 2793–2812 (2017).
14. Ichim, B., & Moyano-Fernández, J. J. *On the consistency of score sheets of a round-robin football tournament.*https://arxiv.org/abs/2208.12372.
15. Bruns, W. Polytope volume in Normaliz. *São Paulo J. Math. Sci.* **17**, 36–54 (2022).
16. Bruns, W. & Koch, R. Computing the integral closure of an affine semigroup. *Univ. Iagell. Acta Math.* **39**, 59–70 (2001).
17. Bruns, W. & Ichim, B. Normaliz: Algorithms for affine monoids and rational cones. *J. Algebra* **324**, 1098–1113 (2010).
18. Bruns, W., Ichim, B. & Söger, C. The power of pyramid decomposition in Normaliz. *J. Symb. Comput.* **74**, 513–536 (2016).
19. Bruns, W. & Söger, C. Generalized Ehrhart series and Integration in Normaliz. *J. Symb. Comput.* **68**, 75–86 (2015).
20. Bruns, W. & Gubeladze, J. *Polytopes, Rings and K-theory* (Springer, 2009).
21. Gehrlein, W. V. & Lepelley, D. *Voting Paradoxes and Group Coherence* (Springer, 2011).
22. Gehrlein, W. V. & Lepelley, D. *Elections, Voting Rules and Paradoxical Outcomes* (Springer, 2017).
23. Gehrlein, W. V. & Fishburn, P. Condorcet's paradox and anonymous preference profiles. *Public Choice* **26**, 1–18 (1976).
24. Gehrlein, W. V. Condorcet winners on four candidates with anonymous voters. *Econ. Lett.* **71**, 335–340 (2001).
25. De Loera, J. A. *et al.* Software for exact integration of polynomials over polyhedra. *Comput. Geom.* **46**, 232–252 (2013).
26. Baldoni, V., Berline, N., De Loera, J. A., Dutra, B., Köppe, M., Moreinis, S., Pinto, G., Vergne, M., & Wu, J. *A user's guide for LattE integrale v1.7.2, 2013*. Software package LattE is available at https://www.math.ucdavis.edu/~latte/.
27. Lepelley, D., Ouafdi, A. & Smaoui, H. Probabilities of electoral outcomes: From three-candidate to four-candidate elections. *Theory Decis.* **88**, 205–229 (2020).
28. Lepelley, D., Louichi, A. & Valognes, F. Computer simulations of voting systems. *Adv. Complex Syst.* **3**, 181–194 (2000).
29. Gehrlein, W. V. & Lepelley, D. On the probability of observing Borda's paradox. *Soc. Choice Welf.* **35**, 1–23 (2010).
30. Brandt, F., Brill, M. & Harrenstein, P. Tournament solutions. In *Handbook of Computational Social Choice* 56–84 (Cambridge University Press, 2016).
31. Davis, R. L. Structure of dominance relations. *Bull. Math. Biophys.* **16**, 131–140 (1954).
32. The online encyclopedia of integer sequences. http://oeis.org/A000568.
33. McGarvey, D. C. A theorem on the construction of voting paradoxes. *Econometrica* **21**, 608–610 (1953).
34. Erdős, P. & Moser, L. On the representation of directed graphs as unions of orderings. *Publ. Math. Inst. Hung. Acad. Sci.* **9**, 125–132 (1964).
35. Filliman, P. The volume of duals and sections of polytopes. *Mathematika* **39**, 67–80 (1992).
36. Büeler, B., & Enge, A. *Vinci.*https://www.math.u-bordeaux.fr/~aenge/.
37. Büeler, B., Enge, A. & Fukuda, K. Exact volume computation for polytopes: A practical study. In *Polytopes—Combinatorics and Computation (Oberwolfach, 1997)* 131–154, DMV Sem., 29, (Birkhäuser, Basel, 2000).

38. Assarf, B. *et al.* Computing convex hulls and counting integer points with polymake. *Math. Program. Comput.* **9**, 1–38 (2017).

Acknowledgements
The first author was supported by the DFG (German Research Foundation) Grant Br 688/26-1. The second author was partially supported by a Grant of Romanian Ministry of Research, Innovation and Digitization, CNCS/CCCDI - UEFISCDI, project number PN-III-P4-ID-PCE-2020-0878, within PNCDI III. The high performance cluster of the University of Osnabrück that made the computations possible was financed by the DFG Grant 456666331. We cordially thank Lars Knipschild, the administrator of the HPC, for his assistance. Our thanks also go to Ulrich von der Ohe for his careful reading of the manuscript.

Dedication
To the memory of Udo Vetter, our teacher, colleague and friend.

Author contributions
All authors have been equally contributing to the paper.

Funding
Open Access funding enabled and organized by Projekt DEAL.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to W.B.