Computational Social Choice 2026

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

http://www.illc.uva.nl/~ulle/teaching/comsoc/2026/

Plan for Today

Today we are going to discuss a variety of questions that concern the roles of *information* and *communication* in voting:

- The Possible Winner Problem
- Compilation of Intermediate Election Results
- Communication Complexity of Voting Rules

Boutilier and Rosenschein (2016) review several other strands of work that also fit under "information and communication".

At the end we'll also briefly mention some *other research directions*.

C. Boutilier and J.S. Rosenschein. Incomplete Information and Communication in Voting. In F. Brandt et al. (eds.), *Handbook of COMSOC*. CUP, 2016.

Possible Winners

Idea: If we only have partial information about the ballots, we may ask which alternatives are *possible winners* (for a given voting rule F).

Let $\mathcal{P}(A)$ be the class of *partial orders* on the set of alternatives A.

The linear order $(\succ^\ell) \in \mathcal{L}(A)$ *refines* the partial order $(\succ^p) \in \mathcal{P}(A)$ if $(\succ^\ell) \supseteq (\succ^p)$ when seen as a set of pairs of alternatives.

Definition: Given a partial profile $\boldsymbol{R} \in \mathcal{P}(A)^n$, an alternative $x^\star \in A$ is called a *possible winner* under voting rule F if $x^\star \in F(\boldsymbol{R}^\star)$ for *some* profile of linear ballots $\boldsymbol{R}^\star \in \mathcal{L}(A)^n$ that refine the ballots in $\boldsymbol{R}$.

The concept was originally introduced by Konczak and Lang (2005).

K. Konczak and J. Lang. *Voting Procedures with Incomplete Preferences*. 1st Multidisciplinary Workshop on Advances in Preference Handling (MPREF), 2005.

Necessary Winners and Preference Elicitation

Given a profile of partial ballots $\boldsymbol{R} \in \mathcal{P}(A)^n$, an alternative $x^\star \in A$ is called a *necessary winner* under voting rule F if $x^\star \in F(\boldsymbol{R}^\star)$ for *all* profiles of linear ballots $\boldsymbol{R}^\star \in \mathcal{L}(A)^n$ that refine the ballots in $\boldsymbol{R}$.

If we *elicit* preferences, either voter-by-voter ("coarse") or pair-by-pair ("fine"), we can stop once possible and necessary winners coincide.

Special Case: Missing Voters

A first natural special case of having only partial ballot information is when some *voters are missing:*

- Some voters have ballots that are *linear orders* (note: a linear order is a special case of a partial order).
- All other voters have *empty relations* as ballots: $x \not\succ y$ for all $x, y \in A$ (also a special case of a partial order)

Possible scenarios:

- We might be in the midst of a coarse elicitation procedure.
- Postal ballots may only arrive a few days after election day.

Exercise: *How do the possible/necessary winner problems relate to the coalitional manipulation problems we had discussed earlier?*

Special Case: Missing Alternatives

A second natural special case of having only partial ballot information is when some *alternatives are missing:*

- Distinguish "old" alternatives A_1 and "new" alternatives A_2.
- All ballots are complete on A_1: $x \succ y$ or $y \succ x$ for all $x, y \in A_1$.
- All ballots are empty on pairs involving a new alternative: $x \not\succ y$ if $x \in A_2$ or $y \in A_2$.

Possible scenario:

- Some alternatives (e.g., a new plan) become available only after voting has started.

The Possible Winner Problem

Let us now analyse the *computational complexity* of the following variant of the problem (for voting rule F):

$\textsc{PossibleWinner}(F)$
Input: Profile of partial ballots $\boldsymbol{R} \in \mathcal{P}(A)^n$; alternative $x^\star \in A$.
Question: Is $x^\star$ a possible winner under voting rule F?

Note that ballots are *unweighted* and that the number of alternatives is *unbounded* (the crucial parameter for the complexity will be the number of alternatives, not the number of voters).

Possible Winners under Plurality

Even for the very simplest of voting rules, computing possible winners is not trivial. But for plurality it at least is polynomial:

Theorem (Betzler and Dorn, 2010): *Under the plurality rule, the possible winner problem can be decided in polynomial time.*

The original proof of Betzler and Dorn is based on flow networks. Here we instead use a reduction to bipartite matching.

N. Betzler and B. Dorn. Towards a Dichotomy for the Possible Winner Problem in Elections Based on Scoring Rules. *Journal of Computer and System Sciences*, 2010.

Bipartite Matching

A *perfect matching* of a graph $G = (V, E)$ is a set of edges $E' \subseteq E$ such that each vertex in V is adjacent to *exactly* one edge in E'.

A graph $G = (V, E)$ is called *bipartite* if the set of vertices V can be partitioned into sets V_1 and V_2 such that each edge in E is adjacent to both a vertex in V_1 and a vertex in V_2.

BIPARTITE MATCHING
Input: Bipartite graph G.
Question: Does G have a perfect matching?

BIPARTITE MATCHING can be decided in polynomial time, using for instance the Hopcroft-Karp Algorithm (Hopcroft and Karp, 1973).

J.E. Hopcroft and R.M. Karp. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 1973.

Proof

Recall: Given a profile of partial ballots, we want to show that we can decide in polynomial time whether $x^\star$ is a possible plurality winner.

An alternative can get a point from a partial ballot iff it is undominated.

Suppose $x^\star$ is undominated in $K \leqslant n$ ballots. So $x^\star$ can get K points.

Can we choose one undominated alternative for each of the other ballots without one alternative getting more than K points?

If $K \cdot (m-1) < n-K$, then one rival must get more points than $x^\star$. ✓

Otherwise, construct a bipartite graph G:

- Lefthand vertices: one for each of the $n-K$ 'other' ballots (= sets of undominated alternatives) and $K \cdot (m-1) - (n-K)$ 'dummy' vertices
- Righthand vertices: K copies of each alternative other than $x^\star$
- Edges: link each set-vertex to each copy of each of its members; and link each dummy vertex to all vertices on the right

Now we have a one-to-one correspondence between perfect matchings of G and ballot refinements that give each rival of $x^\star$ at most K points. ✓

Possible Winners under Borda

Theorem (Xia and Conitzer, 2008): *Under the Borda rule, deciding whether an alternative is a possible winner is NP-complete.*

Proof: By reduction from EXACT3COVER (details omitted).

The result of Xia and Conitzer covers many positional scoring rules. Betzler and Dorn (2010) showed that POSSIBLEWINNER is NP-complete for almost all PSR's (except plurality and antiplurality).

L. Xia and V. Conitzer. Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders. AAAI-2008.

N. Betzler and B. Dorn. Towards a Dichotomy for the Possible Winner Problem in Elections Based on Scoring Rules. *Journal of Computer and System Sciences*, 2010.

Compilation Complexity

For voting rule F, given a partial profile, how many bits do we need to store (in the worst case) so we can compute the outcome later on, once the profile is complete? This is the *compilation complexity* of F.

Let n be the number of (old) voters and m the number of alternatives.
We need $\lceil \log m \rceil$ bits to represent the name of one alternative.
So the CC of any rule F is at most $n\lceil \log(m!) \rceil$ (just store all ballots!).

Exercise: *Prove the following observations to be correct!*

- for *anonymous* rule it is at most $\min\{n\lceil \log(m!) \rceil, m!\lceil \log(n{+}1) \rceil\}$
- for *dictatorial* rules it is $\lceil \log m \rceil$
- for *constant* rules (always electing the same winner) it is 0

Chevaleyre et al. (2009) establish further such results, e.g., the fact that the CC of *Borda* is $\Theta(m \log(nm))$. Exercise: *Upper bound clear?*

Y. Chevaleyre, J. Lang, N. Maudet, and G. Ravailly-Abadie. Compiling the Votes of a Subelectorate. IJCAI-2009.

Compilation Complexity of Plurality

The compilation complexity of *plurality* can be at most $m\lceil\log(n+1)\rceil$ (store the plurality score $\in \{0,\ldots,n\}$ for each alternative).

But we can do better:

Theorem (Chevaleyre et al., 2009): *The compilation complexity of the plurality rule (for an unknown number of additional voters) is exactly* $\lceil\log\binom{m+n-1}{n}\rceil$, *which is in* $\Theta(m\log(1+\frac{n}{m})+n\log(1+\frac{m}{n}))$.

Recall that $f(n)\in\Theta(g(n))$ means that f is asymptotically bounded both from above and from below by g.

The proof for the exact bound is on the next slide. The rest follows from *Stirling's approximation* [$\log(n!)$ is in $\Theta(n\log n)$].

Y. Chevaleyre, J. Lang, N. Maudet, and G. Ravailly-Abadie. Compiling the Votes of a Subelectorate. IJCAI-2009.

Proof

Profiles $\boldsymbol{R}$ and $\boldsymbol{R}'$ are *plurality-equivalent* iff for each alternative x the number of times x occurs in the the top position in $\boldsymbol{R}$ is equal to the number of times x occurs in the the top position in $\boldsymbol{R}'$.

There are n (= number of voters) top positions available.

How many ways are there to award top positions to m alternatives, so that the sum of numbers of top positions adds up to n?

That is, how many solutions are there for this equation?

$$x_1 + x_2 + \cdots + x_m \quad = \quad n \quad \text{with } x_i \in \{0, 1, 2, \ldots\}$$

This is equal to the number of ways of picking n balls from an urn with m balls *with replacement*, which in turn is equal to picking n balls from an urn with $m + n - 1$ balls *without replacement*: $\binom{m+n-1}{n}$.

The *logarithm* of the above is the compilation complexity: we only need to store the *index of the equivalence class* we are in. ✓

Communication Complexity

Yet another interesting question: How much information do voters need to transmit to allow us to compute the winner of an election?

Let n be the number of voters and m the number of alternatives.
We need $\lceil \log m \rceil$ bits to communicate the identity of an alternative.
Upper bounds are easy to derive:

- Plurality: $O(n \log m)$ — each voter sends one alternative name
- Approval: $O(nm)$ — let each voter communicate a bit-string of length m to encode their approved subset of alternatives
- Borda (any rule with linear orders as ballots): $O(n\, m \log m)$ — each voter sends m alternative names in turn

Conitzer and Sandholm (2005) show that many of these bounds are tight, using tools from *communication complexity*.

V. Conitzer and T. Sandholm. Communication Complexity of Common Voting Rules. EC-2005.

Upper Bound for Instant Runoff

Recall: Under *instant runoff* we keep eliminating plurality losers until one alternative is the majority winner.

In the worst case, the full ranking of some voters may be required to compute the winner. So $O(n\,m\log m)$ is an upper bound. (*Why?*)

Conitzer and Sandholm (2005) show that this is *not* a tight upper bound and give a better one: $O(n\log^2 m)$

- Protocol: in the first round, each voter names their favourite alternative; in subsequent rounds any voter whose favourite alternative has just been eliminated names their new favourite
- Insight: the number of voters who get their favourite alternative eliminated is small, namely $\leqslant \frac{n}{m-i+1}$ in round i
- $\sum_{i=1}^{m-1} \frac{n}{m-i+1} = n \cdot \sum_{i=2}^{m} \frac{1}{i} \leqslant n \ln m$ (harmonic number)
  So: at most $O(n\log m)$ transmissions of size $O(\log m)$ ✓

Voting with Incomplete Preferences

In the classical model of voting all individuals are assumed to have (and, at least eventually, report) *complete preferences*.

But in many scenarios preferences are and remain *incomplete:*

- Bounded rationality: agents cannot reason about all alternatives
- Bounded attention: agents do not care about all alternatives
- Bounded scope: agents are not being asked about all alternatives

Exercise: *Any initial ideas for designing voting rules for this setting?*

Z. Terzopoulou and U. Endriss. Aggregating Incomplete Pairwise Preferences by Weight. IJCAI-2019.

Z. Terzopoulou and U. Endriss. The Borda Class: An Axiomatic Study of the Borda Rule on Top-Truncated Preferences. *J. Math. Econ.*, 2021.

Voting in Combinatorial Domains

Suppose 13 voters are asked to each vote *yes* or *no* on three issues:

- 3 voters each vote for YNN, NYN, NNY.
- 1 voter each votes for YYY, YYN, YNY, NYY.
- *Nobody* votes for NNN.

If we use the *simple majority* rule *issue-by-issue*, then NNN wins.
This is *paradoxical:* the winning combination got not a single vote!

In the field of *voting in combinatorial domains* we try to circumvent such problems by developing voting rules for scenarios of this kind.

S.J. Brams, D.M. Kilgour, and W.S. Zwicker. The Paradox of Multiple Elections. *Social Choice and Welfare*, 1998.

J. Lang and L. Xia. Voting in Combinatorial Domains. In F. Brandt et al. (eds.), *Handbook of COMSOC*. CUP, 2016.

Liquid Democracy and Social Networks

Liquid Democracy is the idea of allowing voters to choose between voting and delegating their votes to others (in a transitive manner).

Used in practice (famous example: *Piratenpartei* in Germany), but development of sound theoretical foundations still lacking.

The entire methodological battery of COMSOC can be brought to bear on the study of such novel models of collective decision making.

Remark: COMSOC + social network theory is a promising combination

M. Brill. Interactive Democracy. AAMAS-2018 Blue Sky Ideas Track.

U. Grandi. Social Choice and Social Networks. In U. Endriss (ed.), *Trends in Computational Social Choice*. AI Access, 2017.

Summary

We have discussed several situations where only *partial information* regarding the ballots is available. Examples:

- Some voters arrive late.
- Some alternatives enter the election late.
- Elicitation proceeds in stages (ballots are partial orders).

This gives rise to a number of questions:

- *Possible winners:* which alternatives have a chance of winning?
- *Necessary winners:* will certain alternatives surely win?
- *Compilation:* how can we efficiently represent the relevant information elicited so far?

What next? Projects. Then a different model of voting.