The Simplest Form of Bribery

- Bribery shares with manipulation the feature that preference lists are being changed, and with electoral control the feature that there is an external actor, here the briber, conducting these changes.
- There are different possibilities as to what kind of changes the briber is allowed to conduct in the various bribery scenarios.
- We start with the constructive case where, in the simplest variant of bribery, the briber's goal is to make a distinguished candidate a winner of a given election by changing at most k votes.

The Simplest Form of Bribery

Definition (Faliszewski, Hemaspaandra, and Hemaspaandra (2009))

Let $\mathcal{E}$ be some voting system.

Name: $\mathcal{E}$-BRIBERY.

Given:
- An election (C, V),
- a distinguished candidate $c \in C$, and
- a nonnegative integer $k \leq \|V\|$.

Question: Is it possible to make c an $\mathcal{E}$ winner of the election that results from changing no more than k votes in V?

How Hard Is Bribery in Elections? P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. *Journal of Artificial Intelligence Research* **35**:485–532, 2009

Llull and Copeland Voting Computationally Resist Bribery and Constructive Control, P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. *Journal of Artificial Intelligence Research* **35**:275–341, 2009

The Simplest Form of Bribery

Remark:

- If k is sufficiently large (e.g., if $k = \|V\|$), the briber will always be successful, at least for each reasonable voting system that satisfies citizens' sovereignty.
- "Immunity" as in control scenarios does not apply here.
- However, computational barriers—such as NP-hardness of the BRIBERY problem for certain voting systems—might still be useful to protect elections against bribery attacks.
- The simplest variant of bribery, as defined above, is closely related to (coalitional, unweighted) manipulation, as in both cases votes are being modified during the attack.

The Simplest Form of Bribery

Remark:

- A difference is that it is not known right from the start which votes to change: The briber has to thoughtfully pick the "right" votes to change so as to achieve his goal of making his favorite candidate a winner.
- In that sense, bribery is also somewhat akin to control where the chair, for example, has to pick the right votes to add or to delete.
- A difference between bribery and control, however, is that the votes can be modified in a bribery, yet not in a control scenario.
- "Bribery" suggests settings where the briber dishonestly tampers with election results, but it can also capture such scenarios as **political campaign management** and **election fraud detection**.

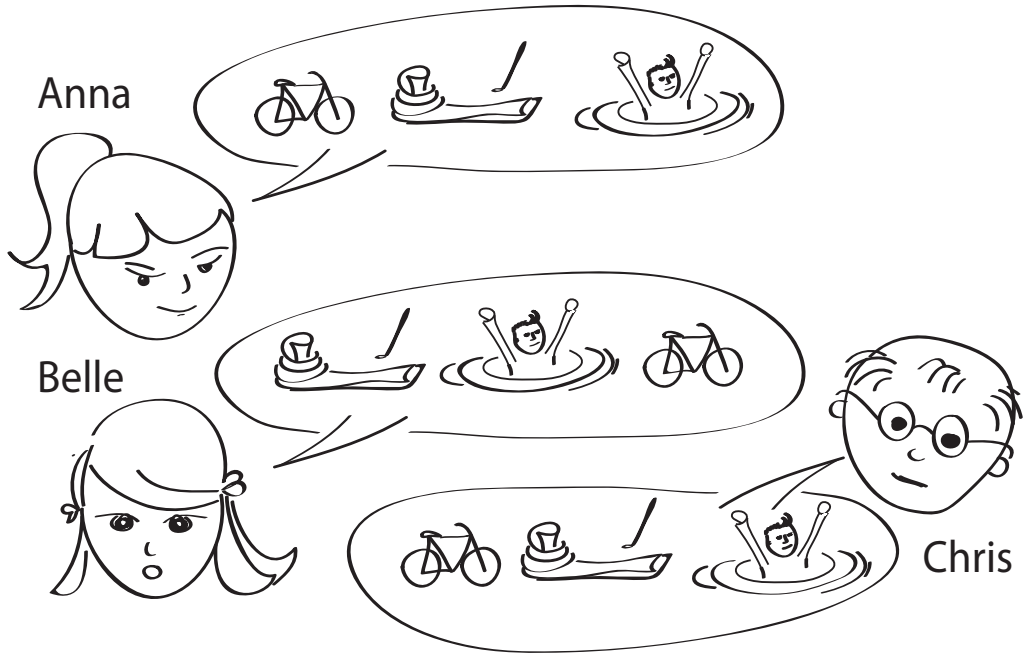
Priced Bribery

In a natural extension of the original bribery problem, we assume that

- a certain budget is available to the briber and
- each voter is willing to change her vote only in exchange for a certain price.

Example

Anna, Belle, and Chris have the following preferences about what to do together:



Priced Bribery

Example (continued)

The owner of the miniature golf facility, Mr. Slugger, happens to come by and asks, "By which rule are you going to vote, kids?"

"Today by a quite simple one," Chris replies, "the plurality rule."

Mr. Slugger takes him aside. "If you vote for miniature golf," he confidentially whispers to him, "I will give you a brand-new golf club! Whatcha think? How cool is that?"

Chris thinks.

Priced Bribery

Example (continued)

"Two!" he eventually demands. "I want to have two clubs. Otherwise, I won't change my vote. And I want to have a golf ball in addition!"

"Are you out of your mind, that's way too expensive!" Mr. Slugger responds, disappointed. "That would totally exceed my budget!"

Chris turns around with a shrug and goes back to the girls to start with them for their bicycle trip, while Mr. Slugger is speculating whether he may have had more luck with making his offer to Anna.

In the *priced bribery problem*, every voter specifies not only a vote but also a price for which he would be willing to change it. *Can the briber make a desired candidate win without exceeding the given budget?*

Priced Bribery

However, it is well possible that a voter

- might be willing to change her vote in one way for a certain price,
- but refuses to change it in another way for the same price.

For example, it might be that in an election with ten candidates, a voter would agree to swap the candidates in the fifth and sixth positions of her vote for 10 bucks, but would not agree to put her most despised candidate on top for the same price.

For such a severe change in her preferences, she perhaps would demand 100 bucks, and even 150 bucks to actually *swap* the candidates in the first and last positions of her vote.

Priced Bribery

Therefore, it would only be reasonable to assume that the price function of a voter depends on the vote the briber has in mind for this voter when bribing her.

However, there are $m!$ possible linear rankings for m candidates.

Thus, if we were to represent price functions by specifying a price for each possible vote, we would get into a hell of a mess algorithmically.

That is why one usually focuses on price functions that can be represented succinctly.

Priced Bribery

Most common are the following families of price functions:

1. *discrete price functions* consist of 0-1-valued functions:
   - Changing a vote costs 1, and
   - leaving it unchanged costs 0.
2. *$discrete price functions* consist of two-valued functions:
   - Changing a vote costs c, where c is a positive integer (and may be a different integer for each voter), and
   - leaving it unchanged costs 0.
3. In *swap-bribery price functions*, for each two adjacent candidates $c, d \in C$, there is a cost that a voter demands for swapping c and d in her vote, and the cost of changing this (original) vote into the vote desired by the briber is the sum of the single costs of swaps needed to transform the original vote into the briber's target vote.

Priced Bribery

More formally, for a preference profile $(\succ_1, \ldots, \succ_n)$, we say that the price functions are

1. *discrete* if for each π_i, $1 \leq i \leq n$, and for each preference order $\succ$, it holds that $\pi_i(\succ) = 0$ if $\succ = \succ_i$, and $\pi_i(\succ) = 1$ otherwise.

2. *$discrete* if for each π_i, $1 \leq i \leq n$, there is an integer c_i such that for each preference order $\succ$, it holds that $\pi_i(\succ) = 0$ if $\succ = \succ_i$, and $\pi_i(\succ) = c_i$ otherwise. (Each voter can have a different value c_i.)

3. *swap-bribery price functions* if for each π_i, $1 \leq i \leq n$, and for each two candidates $x, y \in C$, there is a value $c_i^{\{x,y\}}$ such that for each preference order $\succ$, $\pi_i(\succ)$ is the sum of the values $c_i^{\{x,y\}}$ such that $\succ$ ranks x and y in the opposite order than $\succ_i$ does.

Priced Bribery

Definition (Faliszewski, Hemaspaandra, and Hemaspaandra (2009))

Let $\mathcal{E}$ be some voting system.

Name: $\mathcal{E}$-$BRIBERY.

Given:
- An election (C, V) with $V = (\succ_1, \ldots, \succ_n)$,
- a distinguished candidate $c \in C$, and
- a budget $B \in \mathbb{N}$, and
- a collection of price functions $\Pi = (\pi_1, \ldots, \pi_n)$, where $\pi_i(\succ)$ is the cost of convincing the ith voter to cast vote $\succ$ over C (we require $\pi_i(\succ_i) = 0$), $1 \leq i \leq n$.

Question: Does there exist a preference profile $V' = (\succ'_1, \ldots, \succ'_n)$ such that (i) c is an $\mathcal{E}$ winner of election (C, V'), and (ii) $\sum_{i=1}^{n} \pi_i(\succ'_i) \leq B$?

Priced Bribery

Example (bribery in a Borda election)

points:	5	4	3	2	1	0
voter 1:	a	c	b	f	e	d
voter 2:	b	a	f	c	e	d
voter 3:	c	d	b	a	f	e
voter 4:	e	d	b	f	c	a
voter 5:	e	d	c	b	f	a
Borda winner:	b with score 16					

Suppose that each voter has the same unit price, and that the goal is to ensure the victory of f through bribery.

$score(b) = 16$ and $score(f) = 9$.

Bribe voter 3 to cast vote $f\ d\ a\ c\ e\ b$.

Then b, e, and f have score 13 each, and a, c, and d have score 12 each.

This means that there is a successful bribery with cost one.

Priced Bribery

Example (priced bribery in a Borda election)

points:	5	4	3	2	1	0
voter 1:	*a*	*c*	*b*	*f*	*e*	*d*
voter 2:	*b*	*a*	*f*	*c*	*e*	*d*
voter 3:	*c*	*d*	*b*	*a*	*f*	*e*
voter 4:	*e*	*d*	*b*	*f*	*c*	*a*
voter 5:	*e*	*d*	*c*	*b*	*f*	*a*
Borda winner:	*b* with score 16					

On the other hand,

- if voters 1 and 5 had cost one and
- the remaining voters had cost three each,

then it would be better to bribe voters 1 and 5 to shift *f* to the top positions in their votes.

Bribery, Priced Bribery, and Swap Bribery

- P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra, **How Hard Is Bribery in Elections?** *Journal of Artificial Intelligence Research* **35**:485–532, 2009

  P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, **Llull and Copeland Voting Computationally Resist Bribery and Constructive Control**. *Journal of Artificial Intelligence Research* **35**:275–341, 2009

  focused largely on
  - BRIBERY based on *discrete price functions* and
  - $BRIBERY based on *$discrete price functions*.
- SWAP-BRIBERY based on *swap-bribery price functions* is due to

  P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, **Llull and Copeland Voting Computationally Resist Bribery and Constructive Control**. *Journal of Artificial Intelligence Research* **35**:275–341, 2009

  and was later carefully studied by

  E. Elkind, P. Faliszewski, and A. Slinko, **Swap Bribery**. *Proc. 2nd International Symposium on Algorithmic Game Theory*, pp. 299–310, 2009

Further Variants of Bribery

- In the same way but for weighted elections, we can define
  - WEIGHTED-BRIBERY,
  - WEIGHTED-$BRIBERY, and
  - WEIGHTED-SWAP-BRIBERY.
- SHIFT-BRIBERY is the special form of SWAP-BRIBERY where only the distinguished candidate may be swapped.
- In NEGATIVE-BRIBERY, the briber (wishing to be inconspicuous and unsuspicious) is not allowed to make the distinguished candidate a bribed voter's top choice.
- All these *constructive* variants of bribery also have *destructive* analogues.

Swap Bribery

Example (swap bribery in a Borda election)

points:	5	4	3	2	1	0
voter 1:	*a*	*c*	*b*	*f*	*e*	*d*
voter 2:	*b*	*a*	*f*	*c*	*e*	*d*
voter 3:	*c*	*d*	*b*	*a*	*f*	*e*
voter 4:	*e*	*d*	*b*	*f*	*c*	*a*
voter 5:	*e*	*d*	*c*	*b*	*f*	*a*
Borda winner:	*b* with score 16					

Suppose we want to ensure victory of candidate *d* through swap bribery.

Each swap has unit cost.

	b	*c*	*d*	*e*	*a*	*f*
score	16	15	12	12	11	9

Bribe voter 1 to swap *b* with *f*, then with *e*, finally with *d* to get: *a c f e d b*.

Then *b* loses three points and *d*, *e*, and *f* gain one point each:

	b	*c*	*d*	*e*	*a*	*f*
score	13	15	13	13	11	10

Swap Bribery

Example (swap bribery in a Borda election)

points:	5	4	3	2	1	0
voter 1:	*a*	*c*	*f*	*e*	*d*	*b*
voter 2:	*b*	*a*	*f*	*c*	*e*	*d*
voter 3:	*c*	*d*	*b*	*a*	*f*	*e*
voter 4:	*e*	*d*	*b*	*f*	*c*	*a*
voter 5:	*e*	*d*	*c*	*b*	*f*	*a*
Borda winner:	*b* with score 16					

But *c* still has 15 points:

	b	*c*	*d*	*e*	*a*	*f*
score	13	15	13	13	11	10

So, next bribe voter 3 to swap *c* and *d* to get *d c b a f e* (so *c* and *d* win):

	b	*c*	*d*	*e*	*a*	*f*
score	13	14	14	13	11	10

This is a successful swap bribery of cost four (and, indeed, the cheapest successful swap bribery for *d* here).

Bribery for Plurality

Theorem

For plurality voting it holds that:

1. BRIBERY, WEIGHTED-BRIBERY, *and* $BRIBERY *are each in* P, *but* WEIGHTED-$BRIBERY *is* NP*-complete, and*
2. SWAP-BRIBERY *is in* P.

Proof Sketch. It is easy to see that plurality-BRIBERY can be solved by (repeating in a loop) the following greedy algorithm:

> *If the preferred candidate is not a winner already,*
> *then pick one of the current winners and bribe one of*
> *her voters to vote for the preferred candidate.*

Weighted Bribery for Plurality

Unfortunately, such greedy approaches do not work for plurality-WEIGHTED-BRIBERY. For example, consider an algorithm that works in iterations and in each iteration bribes the heaviest voter among those that vote for one of the current winners.

Counterexample:

Let (C, V) be an election where $C = \{p, a, b, c\}$ and where we have

- 9 weight-1 voters voting for a,
- a single weight-5 voter voting for b, and
- a single weight-5 voter voting for c.

Clearly, it suffices to bribe the two weight-5 voters, but the heuristic would bribe five voters with weight 1 each.

Weighted Bribery for Plurality

On the other hand, bribing the heaviest voter first does not always work either.

Counterexample: $C = \{p, a, b\}$ with

- p receiving no votes at first,
- a receiving three weight-2 votes and one weight-1 vote, and
- b receiving two weight-3 votes.

To make p a winner, it suffices to bribe one weight-2 vote and one weight-3 vote, but the heuristic bribes three votes.

Nonetheless, a combination of these two heuristics does yield a polynomial-time algorithm for plurality-WEIGHTED-BRIBERY.

Weighted Bribery for Plurality

Let us consider some weighted plurality election and let us say that somehow we know that after an optimal bribery, our preferred candidate p has at least T points.

Naturally, all the other candidates have to end up with at most T points (and at least one of them will get exactly T points).

Thus, for each candidate a that has more than T points, we should keep bribing its heaviest voters until its score decreases to at most T.

This corresponds to running the *"bribe the current winner's heaviest voter"* heuristic.

Weighted Bribery for Plurality

If, after bringing each candidate to at most T points, the preferred candidate still does not have T points, we bribe the globally heaviest voters to vote for the preferred candidate.

We do so until the preferred candidate reaches at least T points (this corresponds to running the *"bribe the heaviest voter"* heuristic).

If we chose the value of T correctly, by this point we would have found an optimal bribery strategy.

But how do we choose T?

If the weights were encoded in unary, we could try all possible values, but doing so for binary-encoded weights would give an exponential-time algorithm.

Weighted Bribery for Plurality

Fortunately, we can make the following observation:

- For each candidate a, we bribe a's voters in the order of their nonincreasing weights.
- Thus, after executing the above-described strategy for some optimal value T, a's score is in the set

  $\{a$'s original score,
  a's score without its heaviest voter,
  a's score without its two heaviest voters, $\ldots\}$.

- Hence, it suffices to consider values T of this form only (for each candidate) and to pick one that leads to a cheapest bribery.

Priced Bribery and Swap Bribery for Plurality

Easy exercise: Adapt this algorithm to the case of plurality-$BRIBERY.

On the other hand, solving plurality-SWAP-BRIBERY requires a somewhat different approach.

The reason is that under SWAP-BRIBERY it might not always be optimal to push our preferred candidate to the top of the votes, but sometimes it may be cheaper and more effective to replace some high-scoring candidates with other, low-scoring ones.

To account for such strategies, Elkind et al. (2009) compute, for each vote v, the lowest cost of replacing v's current top-candidate with each other one, and then run a flow-based algorithm (due to Faliszewski, 2008) to find the bribing strategy. We omit the details here.

Weighted and Priced Bribery for Plurality

For plurality-WEIGHTED-$BRIBERY, it is easy to see that the problem is in NP and so we only show NP-hardness.

We give a reduction from the PARTITION problem to plurality-WEIGHTED-$BRIBERY.

Recall that in the PARTITION problem

- the input consists of a sequence of positive integers that sum up to some value S, and
- we ask if it is possible to partition this sequence into two subsequences that both sum up to $S/2$ (naturally, for that S needs to be even).

Weighted and Priced Bribery for Plurality

Let $(s_1, \ldots, s_n)$ be the input sequence and let $S = \sum_{i=1}^{n} s_i$. We form

- an election (C, V), with
  - $C = \{p, d\}$ and
  - with V containing n voters voting for d;
  - for each i, $1 \leq i \leq n$, the ith voter has weight s_i and her price function is *"it costs s_i to change the vote."*
- The budget B is $S/2$.

In effect, any bribery of cost at most B can give p a score of at most $S/2$.

The only such briberies that would ensure that p is among the winners must give p score exactly $S/2$, by solving the original PARTITION instance. ❑

Overview of Complexity of Bribery

Remark: This result is particularly useful because its proof easily adapts to most other typical voting rules, showing that WEIGHTED-$BRIBERY is NP-complete for them as well.

However, in-depth study of f-BRIBERY has shown that the problem is NP-complete for most natural voting rules f.

We survey these results in table on the next slide.

Naturally, the hardness results for BRIBERY immediately transfer to $BRIBERY and WEIGHTED-BRIBERY.

Theorem

For each voting rule $\mathcal{E}$, $\mathcal{E}$-BRIBERY reduces to $\mathcal{E}$-$BRIBERY and to $\mathcal{E}$-WEIGHTED-BRIBERY. **without proof**

Overview of Complexity of Bribery

Table: The complexity of $\mathcal{E}$-BRIBERY for various voting rules.

$\mathcal{E}$	$\mathcal{E}$-BRIBERY	reference
plurality	P	Faliszewski et al. (2009a)
veto	P	Faliszewski et al. (2009a)
2-approval	P	Lin (2012)
k-veto, $k \in \{2, 3\}$	P	Lin (2012)
k-approval, $k \geq 3$	NP-complete	Lin (2012)
k-veto, $k \geq 4$	NP-complete	Lin (2012)
Borda	NP-complete	Brelsford et al. (2008)
STV	NP-complete	Xia (2012)

Overview of Complexity of Bribery

Table: The complexity of $\mathcal{E}$-BRIBERY for various voting rules.

$\mathcal{E}$	$\mathcal{E}$-BRIBERY	reference
Bucklin	NP-complete	Faliszewski et al. (2015)
fallback	NP-complete	Faliszewski et al. (2015)
maximin	NP-complete	Faliszewski et al. (2011)
Copeland	NP-complete	Faliszewski et al. (2009b)
Schulze	NP-complete	Parkes and Xia (2012)
ranked pairs	NP-complete	Xia (2012)
approval	NP-complete	Faliszewski et al. (2009a)
range voting	NP-complete	follows from the approval result

Relations between Bribery and Manipulation

Let $\mathcal{E}$ be a voting rule. Recall that in $\mathcal{E}$-CCM (for "constructive, coalitional manipulation"), we are given (a) an election (C, V), (b) a preferred alternative $p \in C$, and (c) a collection V' of voters with unspecified preference orders.

We ask if it is possible to ensure that p is an $\mathcal{E}$ winner of election $(C, V \cup V')$ by setting the preference orders of the voters in V'.

$\mathcal{E}$-CCWM is defined analogously for weighted elections, with given manipulators' weights.

Theorem

*For each voting rule $\mathcal{E}$, $\mathcal{E}$-*CCM *reduces to $\mathcal{E}$-*$BRIBERY*, and $\mathcal{E}$-*CCWM *reduces to $\mathcal{E}$-*WEIGHTED-$BRIBERY*.* **without proof**

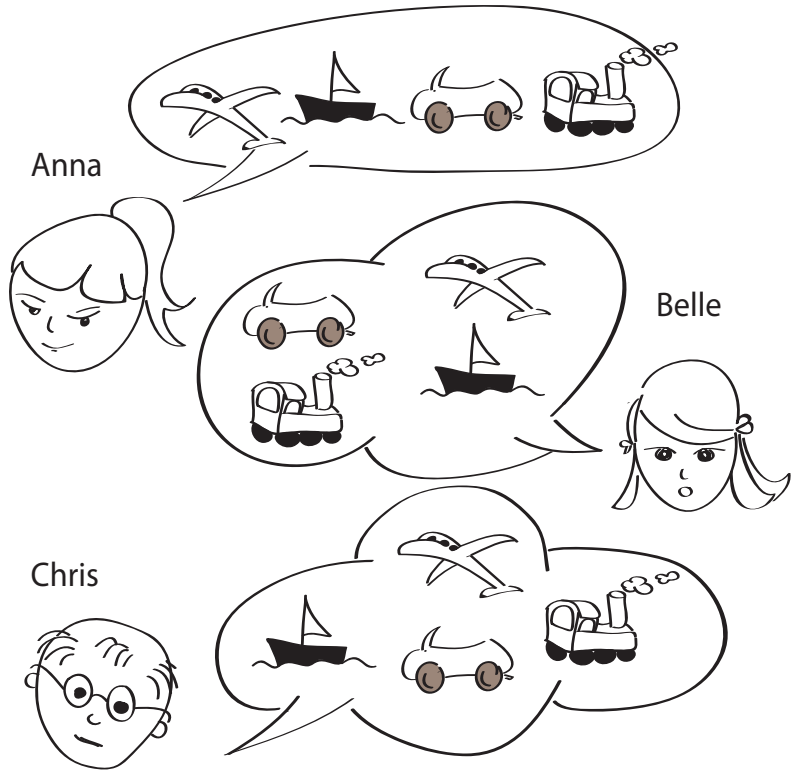
The Possible and Necessary Winner Problem

Konczak and Lang (2005) defined:

- In an election with partial preferences, a *possible winner* is a candidate that has the possibility to win in some total extension of the partial preferences.
- A *necessary winner* is a candidate that wins in *every* total extension of the given partial preferences.

Example:

Anna, Belle, and Chris have the following trip preferences:



The Possible and Necessary Winner Problem

Definition (Konczak and Lang (2005))

Name: $\mathcal{E}$-POSSIBLE-WINNER.

Given:
- An election (C, V), where the votes in V are represented as partial orders over C, and
- a distinguished candidate $c \in C$.

Question: Is c a possible $\mathcal{E}$ winner of (C, V), i.e., is it possible to fully extend every partial vote in V such that c is an $\mathcal{E}$ winner of the resulting election?

$\mathcal{E}$-NECESSARY-WINNER is defined analogously by asking whether c is a necessary $\mathcal{E}$ winner of (C, V).

Possible Winner for Condorcet Voting

Theorem (Konczak and Lang (2005))

POSSIBLE-WINNER *is in* P *for Condorcet voting.*

Proof: For a profile T of linear orders over the set of candidates C and for any two candidates $x, y \in C$, let $D_T(x, y)$ denote

the number of votes in T that prefer x to y minus

the number of votes in T that prefer y to x.

For a profile R of partial orders over C and for any two candidates $x, y \in C$, let $D_R^{\max}(x, y)$ denote the maximal value of $D_T(x, y)$, taken over all total extensions T of the partial votes in R.

Possible Winner for Condorcet Voting

The proof will have two parts:

1. We will first show that for a profile of partial orders $R = (R_1, \ldots, R_n)$ and any two candidates $x, y \in C$, we have

$$D_R^{\max}(x, y) = \|\{i \mid \text{not}(y >_{R_i} x)\}\| - \|\{i \mid y >_{R_i} x\}\|. \quad (1)$$

   This means that $D_R^{\max}(x, y)$ equals
   the number of votes from R not preferring y to x (either because x is preferred to y or because they are incomparable in such a vote)
   minus the number of votes that prefer y to x.

2. In the second part of the proof, we will show that

$$x \in C \text{ is a possible Condorcet winner for } R$$
$$\iff \quad D_R^{\max}(x, y) > 0 \text{ for all } y \neq x.$$

Possible Winner for Condorcet Voting

Together with the first part, this shows that whether a candidate is a possible Condorcet winner can be decided in polynomial time.

1. We start by proving (1). If $y >_{R_i} x$ then it obviously holds that $y >_{T_i} x$ for every total extension T_i of R_i. Hence,

$$\|\{i \mid y >_{T_i} x\}\| \geq \|\{i \mid y >_{R_i} x\}\|. \tag{2}$$

   Furthermore, since T_i is an extension of R_i, $x >_{T_i} y$ implies $\text{not}(y >_{R_i} x)$. Thus

$$\|\{i \mid x >_{T_i} y\}\| \leq \|\{i \mid \text{not}(y >_{R_i} x)\}\|. \tag{3}$$

Possible Winner for Condorcet Voting

Combining (2) and (3) yields

$$\|\{i \mid x >_{T_i} y\}\| - \|\{i \mid y >_{T_i} x\}\| \leq \|\{i \mid \text{not}(y >_{R_i} x)\}\| - \|\{i \mid y >_{R_i} x\}\|. \tag{4}$$

Since this holds for all total extensions T_i of R_i, we get

$$D_R^{\max}(x, y) \leq \|\{i \mid \text{not}(y >_{R_i} x)\}\| - \|\{i \mid y >_{R_i} x\}\|.$$

For the converse inequality, consider a profile $R^x = (R_1^x, \ldots, R_n^x)$ of linear orders that are the best possible total extensions of the profile $R = (R_1, \ldots, R_n)$ with respect to candidate x, i.e., whenever the relation between x and some $z \in C$, $z \neq x$, is undetermined in R it holds that z will be placed behind x in R^x.

Possible Winner for Condorcet Voting

Then

$$\begin{aligned} D_{R^x}(x,y) &= \|\{i \,|\, x >_{R_i^x} y\}\| - \|\{i \,|\, y >_{R_i^x} x\}\| \\ &= \|\{i \,|\, \text{not}(y >_{R_i^x} x)\}\| - \|\{i \,|\, y >_{R_i^x} x\}\| \\ &= \|\{i \,|\, \text{not}(y >_{R_i} x)\}\| - \|\{i \,|\, y >_{R_i} x\}\|. \end{aligned}$$

Hence, we have that

$$D_R^{\max}(x,y) \geq \|\{i \,|\, \text{not}(y >_{R_i} x)\}\| - \|\{i \,|\, y >_{R_i} x\}\|.$$

This completes the proof of (1).

Possible Winner for Condorcet Voting

2. The second part of the proof is to show that a candidate x is a possible Condorcet winner for R if and only if for each $y \neq x$,

$$D_R^{\max}(x, y) > 0.$$

($\Leftarrow$) Assume that $D_R^{\max}(x, y) > 0$ for each $y \neq x$.

For a contradiction, suppose that x is not a possible Condorcet winner.

Then, for all total extensions $T = (T_1, \ldots, T_n)$ of $R = (R_1, \ldots, R_n)$, there is a candidate y such that $\|\{i \mid x >_{T_i} y\}\| \leq \|\{i \mid y >_{T_i} x\}\|$.

Possible Winner for Condorcet Voting

This also holds for the best possible extension R^x, so $D_{R^x}(x, y) \leq 0$, and hence $D_R^{\max}(x, y) \leq 0$.

But this is a contradiction to the assumption that $D_R^{\max}(x, y) > 0$ for each $y \neq x$.

($\Rightarrow$) Assume that x is a possible Condorcet winner for R.

Then there exists a total extension $T = (T_1, \ldots, T_n)$ of R such that for each candidate $y \neq x$, it holds that

$$\|\{i \mid x >_{T_i} y\}\| > \|\{i \mid y >_{T_i} x\}\|.$$

Hence we have $D_T(x, y) > 0$, and thus $D_R^{\max}(x, y) > 0$. ❑

Possible Winner for Pure Scoring Rules

Definition

A scoring protocol is said to be *pure* if for each $m \geq 2$, the scoring vector for m candidates results from the scoring vector for $m-1$ candidates by inserting an additional score value α_i at any position i so that the condition $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$ is still satisfied.

Theorem (Betzler and Dorn (2010); Baumeister and Rothe (2012))

POSSIBLE-WINNER *is in* P *for plurality and veto (and the trivial rule with an all-zero scoring vector), and is* NP*-complete for all other pure scoring rules.* **without proof**

Complexity of Possible and Necessary Winner

	POSSIBLE-WINNER	NECESSARY-WINNER
Condorcet	P	P
Plurality	P	P
Veto	P	P
Borda	NP-complete	P
Simpson	NP-complete	P
Bucklin (simplified)	NP-complete	P
Cup protocol	NP-complete	coNP-complete
Copeland	NP-complete	coNP-complete
STV	NP-complete	coNP-complete
Ranked pairs	NP-complete	coNP-complete

Swap Bribery, Shift Bribery, and Possible Winner

Theorem (Elkind et al. (2009))

For each rule $\mathcal{E}$, $\mathcal{E}$-POSSIBLE-WINNER reduces to $\mathcal{E}$-SWAP-BRIBERY.

Proof Idea:

- Make the already linearly ordered pairs of candidates so costly that swapping them would exceed the briber's budget,
- but a swap between any two candidates that are not yet linearly ordered (in the given partially ordered preference profile) is available for free.

Theorem (Elkind et al. (2009))

There is a polynomial-time 2-approximation algorithm for the cost of a cheapest shift bribery under Borda voting. **without proof**