Bribery

Maria Serna

Fall 2019

1 Bribery

Bribery in elections

- Question: Is it possible, by modifying the preferences of a given number of voters, to make some preferred candidate a winner?

Bribery in elections

- Question: Is it possible, by modifying the preferences of a given number of voters, to make some preferred candidate a winner?
- This is a variant of manipulation.

Bribery in elections

- Question: Is it possible, by modifying the preferences of a given number of voters, to make some preferred candidate a winner?
- This is a variant of manipulation.
- The problem models a type of attack where, the person interested in the success of a particular candidate, picks a group of voters and convinces (or pays) them to vote as he or she says.

Bribery Problem

Bribery Problem

Name: F-BRIBERY
Input: *A preference profile* $\succ$, *a preferred candidate* c *and a nonnegative integer* k.
Question: *Is it possible to make* c *a winner of the* F *election by changing the preference lists of at most* k *voters?*

Bribery Problem

Name: F-BRIBERY
Input: *A preference profile $\succ$, a preferred candidate c and a nonnegative integer k.*
Question: *Is it possible to make c a winner of the F election by changing the preference lists of at most k voters?*

- The problem belongs to NP provided F is computable in polynomial time.

Bribery Problem

- We will study the problem on a variants of plurality.

Bribery Problem

- We will study the problem on a variants of plurality.
- Plurality with weights: Each voter i has weight w_i.

Bribery Problem

- We will study the problem on a variants of plurality.
- Plurality with weights: Each voter i has weight w_i.
- Voter i gives w_i points to its most preferred candidate and 0 to the others.
- The candidate with the higher number of votes wins.

Bribery with money

Bribery with money

Name: PLURALITY-WEIGHTED-$BRIBERY
Input:

Bribery with money

Name: PLURALITY-WEIGHTED-\$BRIBERY
Input: *A set C of m candidates. A collection V of n voters specified via: a preference profile $\succ$, weights $(w_1, \ldots, w_n)$, and their prices $(p_1, \ldots, p_n)$. A distinguished candidate $c \in C$ and a non-negative integer k, the budget.*
Question:

Bribery with money

Name: PLURALITY-WEIGHTED-\$BRIBERY

Input: *A set C of m candidates. A collection V of n voters specified via: a preference profile $\succ$, weights $(w_1, \ldots, w_n)$, and their prices $(p_1, \ldots, p_n)$. A distinguished candidate $c \in C$ and a non-negative integer k, the budget.*

Question: *Is there a set B of voters such that $\sum_{i \in B} p_i \leq k$ and there is a way to bribe the voters from B in such a way that c becomes a winner?*

Plurality

Theorem

Plurality-bribery belongs to P

Plurality

Theorem

Plurality-bribery belongs to P

Proof.

Plurality

Theorem

Plurality-bribery belongs to P

Proof.

Consider the following algorithm.

Plurality

Theorem

Plurality-bribery belongs to P

Proof.

Consider the following algorithm.

- Initially we have bribed zero voters. We check whether c is a winner on $\succ$. If so, we answer yes.

Plurality

Theorem

Plurality-bribery belongs to P

Proof.

Consider the following algorithm.

- Initially we have bribed zero voters. We check whether c is a winner on $\succ$. If so, we answer yes.
- Otherwise, until doing so will exceed the bribe limit, pick any current winner b, bribe one the voters ranking first b to rank first c. If c is now a winner answer yes.

Plurality

Theorem

Plurality-bribery belongs to P

Proof.

Consider the following algorithm.

- Initially we have bribed zero voters. We check whether c is a winner on $\succ$. If so, we answer yes.
- Otherwise, until doing so will exceed the bribe limit, pick any current winner b, bribe one the voters ranking first b to rank first c. If c is now a winner answer yes.
- Answer no.

Plurality

Proof (cont).

Plurality

Proof (cont).

- If the algorithm says yes, obviously bribery is possible.

Plurality

Proof (cont).

- If the algorithm says yes, obviously bribery is possible.
- An easy induction proof shows that, if it is possible to ensure that c is a winner via at most k bribes, our algorithm answer yes

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Proof.

Plurality with weights

Theorem

Plurality-weighted-$bribery is NP-complete, even for two candidates

Proof.

We construct a reduction from PARTITION:
Given integers $x_1, \ldots, x_n$ with $\sum_{i=1}^{n} x_i = 2x$.
Is there a set $S \subseteq \{1, \ldots n\}$ so that $\sum_{i \in S} x_i = \sum_{j \notin S} x_j = x$?

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Proof.

We construct a reduction from PARTITION:
Given integers $x_1, \dots, x_n$ with $\sum_{i=1}^{n} x_i = 2x$.
Is there a set $S \subseteq \{1, \dots n\}$ so that $\sum_{i \in S} x_i = \sum_{j \notin S} x_j = x$?

- The election will have two candidates, a and c, and n voters.

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Proof.

We construct a reduction from PARTITION:
Given integers $x_1, \ldots, x_n$ with $\sum_{i=1}^{n} x_i = 2x$.
Is there a set $S \subseteq \{1, \ldots n\}$ so that $\sum_{i \in S} x_i = \sum_{j \notin S} x_j = x$?

- The election will have two candidates, a and c, and n voters.
- Voter i has weight and prize equal to s_i.

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Proof.

We construct a reduction from PARTITION:
Given integers $x_1, \ldots, x_n$ with $\sum_{i=1}^{n} x_i = 2x$.
Is there a set $S \subseteq \{1, \ldots n\}$ so that $\sum_{i \in S} x_i = \sum_{j \notin S} x_j = x$?

- The election will have two candidates, a and c, and n voters.
- Voter i has weight and prize equal to s_i.
- Every voter prefers a to c.

Plurality with weights

Theorem

Plurality-weighted-\$bribery is NP-complete, even for two candidates

Proof.

We construct a reduction from PARTITION:
Given integers $x_1, \dots, x_n$ with $\sum_{i=1}^{n} x_i = 2x$.
Is there a set $S \subseteq \{1, \dots n\}$ so that $\sum_{i \in S} x_i = \sum_{j \notin S} x_j = x$?

- The election will have two candidates, a and c, and n voters.
- Voter i has weight and prize equal to s_i.
- Every voter prefers a to c.
- $k = x$.

Plurality

Proof (cont).

Plurality

Proof (cont).

- If the partition instance has a solution S, we can bribe the voters in S. We expend al the budget and make c a winner.

Plurality

Proof (cont).

- If the partition instance has a solution S, we can bribe the voters in S. We expend al the budget and make c a winner.
- Otherwise, for any set S with cost $\leq x$, S asigns $\leq x$ points to c, but $V \setminus S$ assigns $> x$ points to a.

Plurality

Proof (cont).

- If the partition instance has a solution S, we can bribe the voters in S. We expend al the budget and make c a winner.
- Otherwise, for any set S with cost $\leq x$, S asigns $\leq x$ points to c, but $V \setminus S$ assigns $> x$ points to a.
  Therefore, the bribery problem as no solution.

Plurality with weights

Theorem

Both Plurality-\$bribery and Plurality-weighted bribery are in P.

Plurality with weights

Theorem

Both Plurality-$bribery and Plurality-weighted bribery are in P.

Proof.

Plurality with weights

Theorem

Both Plurality-\$bribery and Plurality-weighted bribery are in P.

Proof.

- Assume that c will have r votes after the bribery (or in the weighted case, vote weight r), where r is some number to be specified later.

Plurality with weights

Theorem

Both Plurality-$bribery and Plurality-weighted bribery are in P.

Proof.

- Assume that c will have r votes after the bribery (or in the weighted case, vote weight r), where r is some number to be specified later.
- To make c a winner, we need to make sure that everyone else gets at most r votes.

Plurality with weights

Plurality with weights

Proof (cont).

Plurality with weights

Proof (cont).

- We have to choose enough cheapest (heaviest) voters of candidates that defeat c so that after bribing them to vote for c each candidate other than c has at most r votes.

Plurality with weights

Proof (cont).

- We have to choose enough cheapest (heaviest) voters of candidates that defeat c so that after bribing them to vote for c each candidate other than c has at most r votes.
- We have to make sure that c gets at least r votes by bribing the cheapest (the heaviest) of the remaining voters.

Plurality with weights

Proof (cont).

- We have to choose enough cheapest (heaviest) voters of candidates that defeat c so that after bribing them to vote for c each candidate other than c has at most r votes.
- We have to make sure that c gets at least r votes by bribing the cheapest (the heaviest) of the remaining voters.
- If during this process c ever becomes a winner, without exceeding the budget, then we know that bribery is possible.

Plurality with weights

Plurality with weights

Proof (cont).

Plurality with weights

Proof (cont).

- How do we pick the value of r?

Plurality with weights

Proof (cont).

- How do we pick the value of r?
- In the case of plurality-\$bribery, we can simply run the above procedure for all possible values of r, i.e., $0 \leq r \leq n$, and accept exactly if it succeeds for at least one of them.

Plurality with weights

Proof (cont).

- How do we pick the value of r?
- In the case of plurality-\$bribery, we can simply run the above procedure for all possible values of r, i.e., $0 \le r \le n$, and accept exactly if it succeeds for at least one of them.
- For plurality-weighted-bribery it is enough to try all values r that can be obtained as a vote weight of some candidate (other than c) via bribing some number of his or her heaviest voters.

Plurality with weights

Proof (cont).

- How do we pick the value of r?
- In the case of plurality-\$bribery, we can simply run the above procedure for all possible values of r, i.e., $0 \leq r \leq n$, and accept exactly if it succeeds for at least one of them.
- For plurality-weighted-bribery it is enough to try all values r that can be obtained as a vote weight of some candidate (other than c) via bribing some number of his or her heaviest voters.
- There are only polynomially many such values and so the whole algorithm works in polynomial time.

Negative bribery

Negative bribery

- In the previous algorithms voters are bribed to vote for the desired candidate.

Negative bribery

- In the previous algorithms voters are bribed to vote for the desired candidate.
- This might made the bribery easily detectable.

Negative bribery

- In the previous algorithms voters are bribed to vote for the desired candidate.
- This might made the bribery easily detectable.
- To minimize this effect we would lik eto bribe voters to vote for other candidates instead of c.

Negative bribery

- In the previous algorithms voters are bribed to vote for the desired candidate.
- This might made the bribery easily detectable.
- To minimize this effect we would lik eto bribe voters to vote for other candidates instead of c.
- The negative-bribery version of a bribery problem is the same problem with the restriction that it is illegal to bribe people to vote for the designed candidate.

Negative bribery

Theorem

Plurality-negative-$bribery belongs to P.

Negative bribery

Theorem

Plurality-negative-\$bribery belongs to P.

Proof.

- Let (C, V, c, k) be the bribery instance we want to solve.

Negative bribery

Theorem

Plurality-negative-$bribery belongs to P.

Proof.

- Let (C, V, c, k) be the bribery instance we want to solve.
- We need to make c a winner by taking votes away from popular candidates and distributing them among the less popular ones.

Negative bribery

Theorem

Plurality-negative-$bribery belongs to P.

Proof.

- Let (C, V, c, k) be the bribery instance we want to solve.
- We need to make c a winner by taking votes away from popular candidates and distributing them among the less popular ones.
- For a candidate a, define $Sc(a)$ to be the total vote weight of voters who most prefer a.

Negative bribery

Negative bribery

Proof (cont.)

Negative bribery

Proof (cont.)

- We partition the set of all candidates into three sets: candidates that

Negative bribery

Proof (cont.)

- We partition the set of all candidates into three sets: candidates that
  - defeat c, from whom votes need to be taken away
  - are defeated by c, to whom we can give extra votes, and
  - have the same score as p.

Negative bribery

Proof (cont.)

- We partition the set of all candidates into three sets: candidates that
  - defeat c, from whom votes need to be taken away
  - are defeated by c, to whom we can give extra votes, and
  - have the same score as p.

  and define

Negative bribery

Proof (cont.)

- We partition the set of all candidates into three sets: candidates that
  - defeat c, from whom votes need to be taken away
  - are defeated by c, to whom we can give extra votes, and
  - have the same score as p.

  and define

$$C_{above} = \{a|a \in C, Sc(a) > Sc(c)\}.$$
$$C_{below} = \{a|a \in C, Sc(a) < Sc(c)\}.$$
$$C_{equal} = \{a|a \in C, Sc(a) = Sc(c)\}.$$

Negative bribery

Proof (cont.)

Negative bribery

Proof (cont.)

- Since all voters have weight 1, if there is some successful negative bribery then there will be some successful negative bribery that

Negative bribery

Proof (cont.)

- Since all voters have weight 1, if there is some successful negative bribery then there will be some successful negative bribery that
  - will bribe no voters into or out of C_{equal} and
  - won't bribe voters to move within their own "group," e.g., bribing a voter to shift from one C_{below} candidate to another.

Negative bribery

Proof (cont.)

- Since all voters have weight 1, if there is some successful negative bribery then there will be some successful negative bribery that
  - will bribe no voters into or out of C_{equal} and
  - won't bribe voters to move within their own "group," e.g., bribing a voter to shift from one C_{below} candidate to another.
- To make sure that c becomes a winner, for each candidate $a \in C_{above}$, $Sc(a) - Sc(c)$ voters.

Negative bribery

Proof (cont.)

- Since all voters have weight 1, if there is some successful negative bribery then there will be some successful negative bribery that
  - will bribe no voters into or out of C_{equal} and
  - won't bribe voters to move within their own "group," e.g., bribing a voter to shift from one C_{below} candidate to another.
- To make sure that c becomes a winner, for each candidate $a \in C_{above}$, $Sc(a) - Sc(c)$ voters.
- The number of votes that a candidate $a \in C_{below}$ can accept without preventing c from winning is $Sc(c) - Sc(a)$.

Negative bribery

Proof (cont.)

- Since all voters have weight 1, if there is some successful negative bribery then there will be some successful negative bribery that
  - will bribe no voters into or out of C_{equal} and
  - won't bribe voters to move within their own "group," e.g., bribing a voter to shift from one C_{below} candidate to another.
- To make sure that c becomes a winner, for each candidate $a \in C_{above}$, $Sc(a) - Sc(c)$ voters.
- The number of votes that a candidate $a \in C_{below}$ can accept without preventing c from winning is $Sc(c) - Sc(a)$.
- Then a negative bribery is possible if

$$\sum (Sc(a) - Sc(c)) \leq \sum (Sc(c) - Sc(a)).$$

Negative bribery

Theorem

Plurality-weighted-negative-bribery is NP-complete

Negative bribery

Theorem

Plurality-weighted-negative-bribery is NP-complete

Proof.

We construct a reduction from Partition.

Negative bribery

Theorem

Plurality-weighted-negative-bribery is NP-complete

Proof.

We construct a reduction from Partition.

- Let $\{x_1, \dots, x_n\}$ be a sequence of non-negative integers. Let $x_1 + \cdots + x_n = 2X$.
- The election has three candidates $c, a-1, a_2$ and the bribery budget is $k = n + 1$.
- There are $n + 1$ weighted voters:
  - v_0 with weight X, whose preferences are $s > a_1 > a_2$, and
  - $v_1, \dots, v_n$ with weights $s_1, \dots, s_n$, each with preferences $a1 > a2 > c$.

Negative bribery

Proof (cont.)

Negative bribery

Proof (cont.)

- The goal of the briber is to ensure c's victory via bribing up to $n+1$ voters (i.e., all)

Negative bribery

Proof (cont.)

- The goal of the briber is to ensure c's victory via bribing up to $n+1$ voters (i.e., all)
- The only reasonable bribe is to transfer the vote of v_i, $1 \leq i \leq n$, from a_1 to a_2.

Negative bribery

Proof (cont.)

- The goal of the briber is to ensure c's victory via bribing up to $n+1$ voters (i.e., all)
- The only reasonable bribe is to transfer the vote of v_i, $1 \leq i \leq n$, from a_1 to a_2.
- Then, A is a solution to partitiion iff bribing A makes c a winner.

Bribery: Approval voting

Bribery: Approval voting

Theorem

Approval-bribery is NP-complete

Recall that Approval-Manipulation can be solved in polynomial time.

More results

- P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra. How Hard Is Bribery in Elections?. Journal of Artificial Intelligence Research 35 (2009) 485-532
- F. Brandt et al., Eds. Handbook of Computational Choice Theory, Cambridge University Press, 2016.