Lower Bound Issues in Computational Social Choice

Rolf Niedermeier

Fakultät IV,
Institut für Softwaretechnik und Theoretische Informatik,
TU Berlin
www.akt.tu-berlin.de

What is Computational Social Choice?

Computational Social Choice (COMSOC) aims at improving our understanding of

- **social choice mechanisms** and
- **algorithmic decision making**,

including topics such as voting, rank aggregation, fair division, matching (under preferences), and resource allocation.

What is Computational Social Choice?

Computational Social Choice (COMSOC) aims at improving our understanding of

- **social choice mechanisms** and
- **algorithmic decision making**,

including topics such as voting, rank aggregation, fair division, matching (under preferences), and resource allocation.

Involved areas: Artificial Intelligence, Decision Theory, Discrete Mathematics, Logic, Mathematical Economics, Operations Research, Political Sciences, Social Choice, Theoretical Computer Science.

What is Computational Social Choice?

Computational Social Choice (COMSOC) aims at improving our understanding of

- **social choice mechanisms** and
- **algorithmic decision making**,

including topics such as voting, rank aggregation, fair division, matching (under preferences), and resource allocation.

Involved areas: Artificial Intelligence, Decision Theory, Discrete Mathematics, Logic, Mathematical Economics, Operations Research, Political Sciences, Social Choice, Theoretical Computer Science.

Long-term goal: Improve decision support for decision makers.

What is Computational Social Choice?

Computational Social Choice (COMSOC) aims at improving our understanding of

- **social choice mechanisms** and
- **algorithmic decision making**,

including topics such as voting, rank aggregation, fair division, matching (under preferences), and resource allocation.

Involved areas: Artificial Intelligence, Decision Theory, Discrete Mathematics, Logic, Mathematical Economics, Operations Research, Political Sciences, Social Choice, Theoretical Computer Science.

Long-term goal: Improve decision support for decision makers.

"Closest friend": Algorithmic Game Theory.

Motivation

COMSOC problems frequently lead to NP-hardness.

Some typical voting-related examples:

- Gerrymandering.
- Winner determination.

Motivation

COMSOC problems frequently lead to NP-hardness.

Some typical voting-related examples:

- Gerrymandering.
- Winner determination.
- Determination of possible and necessary winners.

Motivation

COMSOC problems frequently lead to NP-hardness.

Some typical voting-related examples:

- Gerrymandering.
- Winner determination.
- Determination of possible and necessary winners.
- Proportional representation (in committees).

Motivation

COMSOC problems frequently lead to NP-hardness.

Some typical voting-related examples:

- Gerrymandering.
- Winner determination.
- Determination of possible and necessary winners.
- Proportional representation (in committees).
- Manipulation.
- Control.
- Bribery.

Motivation

COMSOC problems frequently lead to NP-hardness.

Some typical voting-related examples:

- Gerrymandering.
- Winner determination.
- Determination of possible and necessary winners.
- Proportional representation (in committees).
- Manipulation.
- Control.
- Bribery.

$\rightsquigarrow$ **FPT-related studies highly welcome in the COMSOC world**, also justified by many natural parameterizations (most natural to go multivariate).

http://vignette3.wikia.nocookie.net/villains/images/2/29/Dalek_Parliament.JPG

Motivating Example: Sequential Voting I

1956 Education Act in the USA:

Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Motivating Example: Sequential Voting I

1956 Education Act in the USA:

Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	**b**	$\succ$	**s**	$\succ$	**a**
	100 voters:	**s**	$\succ$	**b**	$\succ$	**a**
	1 voter:	**a**	$\succ$	**b**	$\succ$	**s**

Motivating Example: Sequential Voting I

1956 Education Act in the USA:

Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	**b**	$\succ$	**s**	$\succ$	**a**
	100 voters:	**s**	$\succ$	**b**	$\succ$	**a**
	1 voter:	**a**	$\succ$	**b**	$\succ$	**s**

Agenda $\mathcal{L}$: **b** $\rhd$ **a** $\rhd$ **s**

Motivating Example: Sequential Voting I

1956 Education Act in the USA:

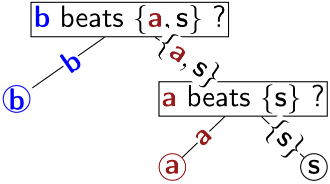
Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	**b**	$\succ$	**s**	$\succ$	**a**
	100 voters:	**s**	$\succ$	**b**	$\succ$	**a**
	1 voter:	**a**	$\succ$	**b**	$\succ$	**s**

Agenda $\mathcal{L}$: **b** $\rhd$ **a** $\rhd$ **s**

Two popular procedures

Euro-Latin procedure:



Motivating Example: Sequential Voting I

1956 Education Act in the USA:

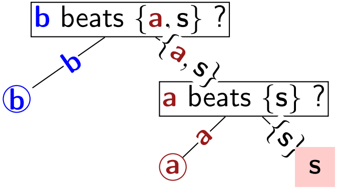
Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	**b**	$\succ$	s	$\succ$	**a**
	100 voters:	s	$\succ$	**b**	$\succ$	**a**
	1 voter:	**a**	$\succ$	**b**	$\succ$	s

Agenda $\mathcal{L}$: **b** $\triangleright$ **a** $\triangleright$ **s**

Two popular procedures

Euro-Latin procedure:



Motivating Example: Sequential Voting I

1956 Education Act in the USA:

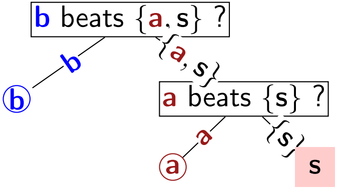
Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	b	$\succ$	s	$\succ$	a
	100 voters:	s	$\succ$	b	$\succ$	a
	1 voter:	a	$\succ$	b	$\succ$	s

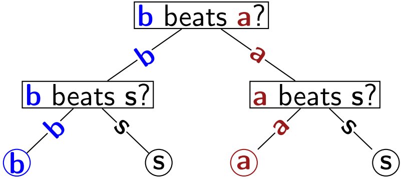
Agenda $\mathcal{L}$: b $\rhd$ a $\rhd$ s

Two popular procedures

Euro-Latin procedure:



Anglo-American procedure:



Motivating Example: Sequential Voting I

1956 Education Act in the USA:

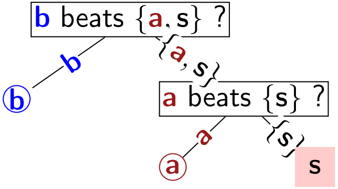
Alternatives	**(b)ill**:	Funding to primary and secondary schools.
	(a)mended bill:	Funding, but not to segregated schools.
	(s)tatus quo:	No bill.

Votes	100 voters:	$b \succ s \succ a$
	100 voters:	$s \succ b \succ a$
	1 voter:	$a \succ b \succ s$

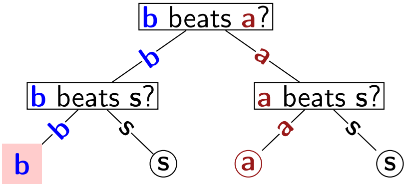
Agenda $\mathcal{L}$: $b \rhd a \rhd s$

Two popular procedures

Euro-Latin procedure:



Anglo-American procedure:



Motivating Example: Sequential Voting II

The agenda (order of alternatives) is essential:

Agenda Control

In: Election $E = (C, V)$ and a specific alternative **a** $\in C$.

Task: Find an agenda for C such that **a** wins.

Motivating Example: Sequential Voting II

The agenda (order of alternatives) is essential:

Agenda Control

In: Election $E = (C, V)$ and a specific alternative **a** $\in C$.
Task: Find an agenda for C such that **a** wins.

Example

1 voter:	**a**	$\succ$	**b**	$\succ$	**c**	$\succ$	**d**
1 voter:	**b**	$\succ$	**c**	$\succ$	**d**	$\succ$	**a**
1 voter:	**c**	$\succ$	**d**	$\succ$	**a**	$\succ$	**b**

Find an agenda such that **a** wins?

- Euro-Latin procedure?
- Anglo-American procedure?

Motivating Example: Sequential Voting II

The agenda (order of alternatives) is essential:

Agenda Control

In: Election $E = (C, V)$ and a specific alternative **a** $\in C$.
Task: Find an agenda for C such that **a** wins.

Example

1 voter:	**a**	$\succ$	**b**	$\succ$	**c**	$\succ$	**d**
1 voter:	**b**	$\succ$	**c**	$\succ$	**d**	$\succ$	**a**
1 voter:	**c**	$\succ$	**d**	$\succ$	**a**	$\succ$	**b**

Find an agenda such that **a** wins?

- Euro-Latin procedure? Yes: **c** $\triangleright$ **d** $\triangleright$ **a** $\triangleright$ **b**.
- Anglo-American procedure?

Motivating Example: Sequential Voting II

The agenda (order of alternatives) is essential:

Agenda Control

In:	Election $E = (C, V)$ and a specific alternative **a** $\in C$.
Task:	Find an agenda for C such that **a** wins.

Example

1 voter:	**a**	$\succ$	**b**	$\succ$	**c**	$\succ$	**d**
1 voter:	**b**	$\succ$	**c**	$\succ$	**d**	$\succ$	**a**
1 voter:	**c**	$\succ$	**d**	$\succ$	**a**	$\succ$	**b**

Find an agenda such that **a** wins?

- Euro-Latin procedure? Yes: **c** $\triangleright$ **d** $\triangleright$ **a** $\triangleright$ **b**.
- Anglo-American procedure? Yes: **d** $\triangleright$ **c** $\triangleright$ **b** $\triangleright$ **a**.

Motivating Example: Sequential Voting III

Two further relevant computational problems for sequential voting:

Manipulation

In: Election $E = (C, V)$, **a** $\in C$, $k \in \mathbb{N}$, and agenda $\mathcal{L}$ for C.

Task: Add k voters such that **a** wins under $\mathcal{L}$.

Motivating Example: Sequential Voting III

Two further relevant computational problems for sequential voting:

Manipulation

In: Election $E = (C, V)$, **a** $\in C$, $k \in \mathbb{N}$, and agenda $\mathcal{L}$ for C.

Task: Add k voters such that **a** wins under $\mathcal{L}$.

Possible (or Necessary) Winner

In: Election $E = (C, V)$ with incomplete preferences, **a** $\in C$, and a partial agenda $\mathcal{B}$ for C.

?: Can **a** win in an (or in every) election completing E under an (or under every) agenda completing $\mathcal{B}$?

Motivating Example: Sequential Voting IV

Summary of computational complexity results ([Bredereck/Chen/N./Walsh, IJCAI 2015]):

- n: number of voters.
- m: number of alternatives.
- k: number of added voters.

Problem	Anglo-American	Euro-Latin
Agenda Control	$O(n \cdot m^2 + m^3)$ ♠	$O(n \cdot m^2)$
Manipulation	$O((k+n) \cdot m^2)$ ♢	$O((k+n) \cdot m)$

Motivating Example: Sequential Voting IV

Summary of computational complexity results ([Bredereck/Chen/N./Walsh, IJCAI 2015]):

- n: number of voters.
- m: number of alternatives.
- k: number of added voters.

Problem	Anglo-American	Euro-Latin
Agenda Control	$O(n \cdot m^2 + m^3)$ ♠	$O(n \cdot m^2)$
Manipulation	$O((k + n) \cdot m^2)$ ♢	$O((k + n) \cdot m)$
Possible Winner	NP-hard	NP-hard
W. Possible Winner	NP-hard when $m = 3$ ♣	NP-hard when $m = 3$

Motivating Example: Sequential Voting IV

Summary of computational complexity results ([Bredereck/Chen/N./Walsh, IJCAI 2015]):

- n: number of voters.
- m: number of alternatives.
- k: number of added voters.

Problem	Anglo-American	Euro-Latin
Agenda Control	$O(n \cdot m^2 + m^3)$ ♠	$O(n \cdot m^2)$
Manipulation	$O((k+n) \cdot m^2)$ ♢	$O((k+n) \cdot m)$
Possible Winner	NP-hard	NP-hard
Necessary Winner	coNP-hard	$O(n \cdot m^3)$
W. Possible Winner	NP-hard when $m = 3$ ♣	NP-hard when $m = 3$
W. Necessary Winner	coNP-hard when $m = 4$ ♣ $O(n)$ when $m \leq 3$	$O(n \cdot m^3)$

Main Venues of COMSOC

Central biannual workshop ("COMSOC Workshop Series"):
Amsterdam (2006), Liverpool (2008), Düsseldorf (2010), Kraków (2012), Pittsburgh (2014), Toulouse (2016).
$\rightsquigarrow$ **Very relaxed, welcoming atmosphere, open to everyone!**

Main Venues of COMSOC

Central biannual workshop ("COMSOC Workshop Series"):
Amsterdam (2006), Liverpool (2008), Düsseldorf (2010), Kraków (2012), Pittsburgh (2014), Toulouse (2016).

$\rightsquigarrow$ **Very relaxed, welcoming atmosphere, open to everyone!**

Dagstuhl seminars: Meanwhile three, last in June 2015.

Main Venues of COMSOC

Central biannual workshop ("COMSOC Workshop Series"):
Amsterdam (2006), Liverpool (2008), Düsseldorf (2010), Kraków (2012), Pittsburgh (2014), Toulouse (2016).
$\rightsquigarrow$ **Very relaxed, welcoming atmosphere, open to everyone!**

Dagstuhl seminars: Meanwhile three, last in June 2015.

Major Conferences with COMSOC sessions:
AAAI, AAMAS (Autonomous Agents & Multiagent Systems), ADT (Algorithmic Decision Theory), EC (Economics and Computation), ECAI, IJCAI, SAGT, WINE, ..., and scattered around in theory conferences.

Main Venues of COMSOC

Central biannual workshop ("COMSOC Workshop Series"):
Amsterdam (2006), Liverpool (2008), Düsseldorf (2010), Kraków (2012), Pittsburgh (2014), Toulouse (2016).
$\rightsquigarrow$ **Very relaxed, welcoming atmosphere, open to everyone!**

Dagstuhl seminars: Meanwhile three, last in June 2015.

Major Conferences with COMSOC sessions:
AAAI, AAMAS (Autonomous Agents & Multiagent Systems), ADT (Algorithmic Decision Theory), EC (Economics and Computation), ECAI, IJCAI, SAGT, WINE, ..., and scattered around in theory conferences.

Journals with a significant fraction of COMSOC stuff:
ACM Transactions on Economics and Computation, Artificial Intelligence, Autonomous Agents and Multi-Agent Systems, Journal of Artificial Intelligence Research, Journal of Mathematical Economics, Mathematical Social Sciences, Social Choice and Welfare, and scattered around in theory journals (e.g., Information and Computation).

COMSOC History

Social choice theory studies mechanisms for collective decision making: voting, preference aggregation, fair division, matching, ...

- Precursors: Condorcet, Borda (18th century) and others.
- Serious scientific discipline since 1950s.

COMSOC History

Social choice theory studies mechanisms for collective decision making: voting, preference aggregation, fair division, matching, ...

- Precursors: Condorcet, Borda (18th century) and others.
- Serious scientific discipline since 1950s.

Computational social choice adds a computational perspective to this, and also explores the use of concepts from social choice in computing (e.g., multi-agent systems).

- Seminal papers from complexity perspective:
  - Determining winners for many voting rules is NP-hard.
    [Bartholdi III/Tovey/Trick, Social Choice and Welfare 1989]
  - STV is NP-hard to manipulate.
    [Bartholdi III/Orlin, Social Choice and Welfare 1991]
  - Many voting rules are NP-hard to control.
    [Bartholdi III/Tovey/Trick, Mathematical and Computer Modelling 1992]

COMSOC History

Social choice theory studies mechanisms for collective decision making: voting, preference aggregation, fair division, matching, ...

- Precursors: Condorcet, Borda (18th century) and others.
- Serious scientific discipline since 1950s.

Computational social choice adds a computational perspective to this, and also explores the use of concepts from social choice in computing (e.g., multi-agent systems).

- Seminal papers from complexity perspective:
  - Determining winners for many voting rules is NP-hard.
    [Bartholdi III/Tovey/Trick, Social Choice and Welfare 1989]
  - STV is NP-hard to manipulate.
    [Bartholdi III/Orlin, Social Choice and Welfare 1991]
  - Many voting rules are NP-hard to control.
    [Bartholdi III/Tovey/Trick, Mathematical and Computer Modelling 1992]
- Active research area with regular contributions since 2002.
- Name "COMSOC" and biannual workshop since 2006.

Outline of the Rest of the Talk

Four COMSOC problems, six issues:

Lobbying: ETH-based lower bounds, ILP-FPT (Lenstra), LOGSNP-completeness.

Outline of the Rest of the Talk

Four COMSOC problems, six issues:

Lobbying: ETH-based lower bounds, ILP-FPT (Lenstra), LOGSNP-completeness.

Shift Bribery: FPT approximation (schemes) and inapproximability.

Outline of the Rest of the Talk

Four COMSOC problems, six issues:

Lobbying: ETH-based lower bounds, ILP-FPT (Lenstra), LOGSNP-completeness.

Shift Bribery: FPT approximation (schemes) and inapproximability.

Network-Based Vertex Dissolution: ParaNP-hardness dichotomies.

Outline of the Rest of the Talk

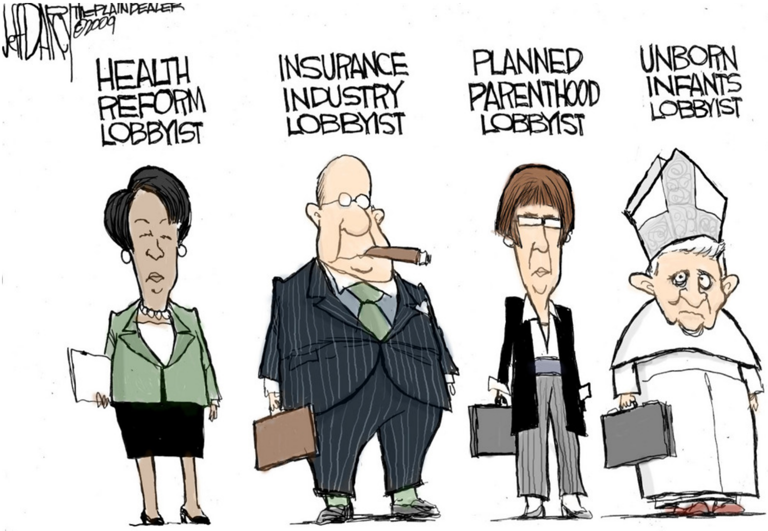
Four COMSOC problems, six issues:

Lobbying: ETH-based lower bounds, ILP-FPT (Lenstra), LOGSNP-completeness.

Shift Bribery: FPT approximation (schemes) and inapproximability.

Network-Based Vertex Dissolution: ParaNP-hardness dichotomies.

Majority-Wise Accepted Ballot: W[2] vs W[2](Maj).



http://media.cleveland.com/darcy/photo/12ggjeffdarcyjpg-21ffa9c8fee1f80c.jpg

Multi-Issue Elections and Lobbying

Multi-issue election

Votes from an electorate who can accept or reject each of several issues.

Election with 3 issues and 5 voters

Issues:	Emissions trading	Nuclear power	Tax raise
Voter 1	×	×	✓
Voter 2	✓	×	✓
Voter 3	×	✓	×
Voter 4	✓	×	×
Voter 5	×	✓	×
Result	×	×	×

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

[Christian/Fellows/Rosamond/Slinko, Review of Economic Design 2007]

Multi-Issue Elections and Lobbying

Multi-issue election
Votes from an electorate who can accept or reject each of several issues.

Election: $A \in \{0,1\}^{n \times m}$
Result: 0^m
Lobbyist's goal: 1^m

Election with 3 issues and 5 voters

Issues:	Emissions trading	Nuclear power	Tax raise
Voter 1	0	0	1
Voter 2	1	0	1
Voter 3	0	1	0
Voter 4	1	0	0
Voter 5	0	1	0
Result	0	0	0
Lobbyist	1	1	1

Lobbying
In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.
?: Can we change at most k rows of A such that each column has more 1s than 0s?

[Christian/Fellows/Rosamond/Slinko, Review of Economic Design 2007]

Multi-Issue Elections and Lobbying

Multi-issue election

Votes from an electorate who can accept or reject each of several issues.

Election: $A \in \{0,1\}^{n \times m}$

Result: 0^m

Lobbyist's goal: 1^m

Election with 3 issues and 5 voters

Issues:	Emissions trading	Nuclear power	Tax raise
Voter 1	0	0	1
Voter 2	1	0	1
Voter 3	0	1	0
Voter 4	1	0	0
Voter 5	0	1	0
Result	0	0	0
Lobbyist	1	1	1

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

[Christian/Fellows/Rosamond/Slinko, Review of Economic Design 2007]

Multi-Issue Elections and Lobbying

Multi-issue election

Votes from an electorate who can accept or reject each of several issues.

Election: $A \in \{0,1\}^{n \times m}$

Result: 0^m

Lobbyist's goal: 1^m

Election with 3 issues and 5 voters

Issues:	Emissions trading	Nuclear power	Tax raise
Voter 1	0	0	1
Voter 2	1	0	1
Voter 3	0	1	0
Voter 4	1	0	0
Voter 5	0	1	0
Result	0	0	0
Lobbyist	1	1	1

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

[Christian/Fellows/Rosamond/Slinko, Review of Economic Design 2007]

Multi-Issue Elections and Lobbying

Multi-issue election

Votes from an electorate who can accept or reject each of several issues.

Election: $A \in \{0,1\}^{n \times m}$
Result: 0^m
Lobbyist's goal: 1^m

Election with 3 issues and 5 voters

Issues:	Emissions trading	Nuclear power	Tax raise
Voter 1	1	1	1
Voter 2	1	0	1
Voter 3	0	1	0
Voter 4	1	0	0
Voter 5	1	1	1
Result	1	1	1

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

[Christian/Fellows/Rosamond/Slinko, Review of Economic Design 2007]

Lobbying: ETH-Based Lower Bounds I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Trivial: Trying all possible subsets of n rows, Lobbying is solvable in $O^*(2^n)$ time.

Lobbying: ETH-Based Lower Bounds I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.
?: Can we change at most k rows of A such that each column has more 1s than 0s?

Trivial: Trying all possible subsets of n rows, Lobbying is solvable in $O^*(2^n)$ time.

Can we do better?

Lobbying: ETH-Based Lower Bounds I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \le n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Trivial: Trying all possible subsets of n rows, Lobbying is solvable in $O^*(2^n)$ time.

Can we do better?

Assuming ETH, Lobbying cannot be solved in $2^{o(n)}(n+m)^{O(1)}$ time.
Proof due to reduction from Vertex Cover (parameterized by the number of vertices).

Lobbying: ETH-Based Lower Bounds II

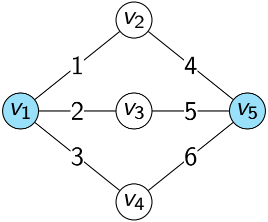
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

Lobbying: ETH-Based Lower Bounds II

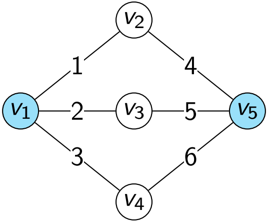
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

	1	2	3	4	5	6
v_1:	1	1	1	0	0	0
v_2:	1	0	0	1	0	0
v_3:	0	1	0	0	1	0
v_4:	0	0	1	0	0	1
v_5:	0	0	0	1	1	1

Lobbying: ETH-Based Lower Bounds II

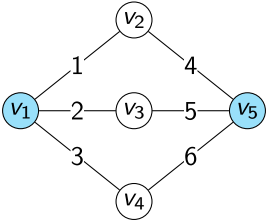
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G=(V,E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G=(V,E)$ of VC and add $|V|-2h+1$ dummy rows and columns.



$h=2$

Vertex Cover

	1	2	3	4	5	6
v_1:	1	1	1	0	0	0
v_2:	1	0	0	1	0	0
v_3:	0	1	0	0	1	0
v_4:	0	0	1	0	0	1
v_5:	0	0	0	1	1	1
d_1:	0	0	0	0	0	0
d_2:	0	0	0	0	0	0

Lobbying: ETH-Based Lower Bounds II

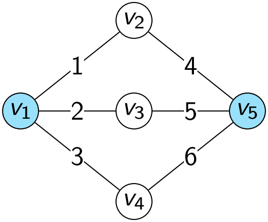
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

	1	2	3	4	5	6	dummies	
v_1:	1	1	1	0	0	0	0	0
v_2:	1	0	0	1	0	0	0	0
v_3:	0	1	0	0	1	0	0	0
v_4:	0	0	1	0	0	1	0	0
v_5:	0	0	0	1	1	1	0	0
d_1:	0	0	0	0	0	0	1	0
d_2:	0	0	0	0	0	0	0	1

Lobbying: ETH-Based Lower Bounds II

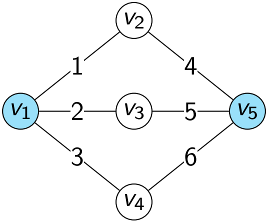
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

	1	2	3	4	5	6	dummies	
v_1:	1	1	1	0	0	0	0	0
v_2:	1	0	0	1	0	0	0	0
v_3:	0	1	0	0	1	0	0	0
v_4:	0	0	1	0	0	1	0	0
v_5:	0	0	0	1	1	1	0	0
d_1:	0	0	0	0	0	0	1	0
d_2:	0	0	0	0	0	0	0	1

$k = |V| - h = 3$

Lobbying

Lobbying: ETH-Based Lower Bounds II

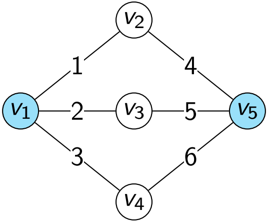
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

	1	2	3	4	5	6	dummies	
v_1:	1	1	1	0	0	0	0	0
v_2:	1	0	0	1	0	0	0	0
v_3:	0	1	0	0	1	0	0	0
v_4:	0	0	1	0	0	1	0	0
v_5:	0	0	0	1	1	1	0	0
d_1:	0	0	0	0	0	0	1	0
d_2:	0	0	0	0	0	0	0	1

$k = |V| - h = 3$

Lobbying

Lobbying: ETH-Based Lower Bounds II

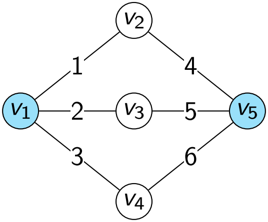
Lobbying not solvable in $2^{o(n)}(n+m)^{O(1)}$ time:

Vertex Cover

In: A graph $G = (V, E)$ and a number h

?: Is there a size-h vertex cover?

Use incidence matrix of input graph $G = (V, E)$ of VC and add $|V| - 2h + 1$ dummy rows and columns.



$h = 2$

Vertex Cover

	1	2	3	4	5	6	dummies	
v_1:	1	1	1	0	0	0	0	0
v_2:	1	0	0	1	0	0	0	0
v_3:	0	1	0	0	1	0	0	0
v_4:	0	0	1	0	0	1	0	0
v_5:	0	0	0	1	1	1	0	0
d_1:	0	0	0	0	0	0	1	0
d_2:	0	0	0	0	0	0	0	1

$k = |V| - h = 3$

Lobbying

$n = 2|V| - 2h + 1 \leq 2|V|$ and Vertex Cover is not solvable in $2^{o(|V|)}$ time.

$\rightsquigarrow$ Lobbying is not solvable in $2^{o(n)} \cdot (n+m)^{O(1)}$ time.

Lobbying: ILP-Based FPT (Lenstra) I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Columns vs. rows: FPT for the $n :=$ #rows was easy to see, what about FPT for $m :=$ #columns?

Lobbying: ILP-Based FPT (Lenstra) I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Columns vs. rows: FPT for the $n := \#$rows was easy to see, what about FPT for $m := \#$columns?

Result

- ILP-FPT for parameter number m of columns (next slide),
- but no poly kernel wrt. (m, number of modified rows k).

[Bredereck/Chen/Hartung/Kratsch/N./Suchý/Woeginger, JAIR 2014]

Lobbying: ILP-Based FPT (Lenstra) I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Columns vs. rows: FPT for the $n := \#$rows was easy to see, what about FPT for $m := \#$columns?

Result

- ILP-FPT for parameter number m of columns (next slide),
- but no poly kernel wrt. (m, number of modified rows k).

[Bredereck/Chen/Hartung/Kratsch/N./Suchý/Woeginger, JAIR 2014]

Challenges:

- Combinatorial algorithm?
- Non-trivial running time lower bounds?

Lobbying: ILP-Based FPT (Lenstra) II

Exact solution via ILP-formulation with 2^m variables

$$\sum_{i=1}^{2^m} x_i \leq k$$

$$\text{constraints: } \forall 1 \leq i \leq 2^m : 0 \leq x_i \leq r_i$$

$$\forall 1 \leq j \leq m : \sum_{i=1}^{2^m} x_i \cdot \bar{A}[i,j] \geq g_j$$

- variable x_i: #rows of type i in the solution
- integer coefficient g_j: #additional ones needed for column j
- integer coefficient r_i: #rows of type i in the matrix
- binary coefficient $\bar{A}[i,j]$: 1 iff row i has a zero in column j

Lobbying: ILP-Based FPT (Lenstra) III

Exact solution via ILP-formulation with 2^m variables

$$\sum_{i=1}^{2^m} x_i \leq k$$

$$\text{constraints: } \forall 1 \leq i \leq 2^m : 0 \leq x_i \leq r_i$$

$$\forall 1 \leq j \leq m : \sum_{i=1}^{2^m} x_i \cdot \bar{A}[i,j] \geq g_j$$

- Linear-time algorithm for $m \leq 4$, factor-log m approximation.
- FPT follows by Lenstra's famous results

  [Lenstra, Mathematics of Operations Research, 1983]
- and implies running time $O^*((2^m)^{2.5 \cdot 2^m + o(2^m)})$

  [Frank and Tardos, Combinatorica, 1987]

  [Kannan, Mathematics of Operations Research, 1987]

Lobbying: ILP-Based FPT (Lenstra) IV

Note: Dozens of other voting problems have ILP-FPT classification wrt. number of alternatives (columns), all lacking lower bounds / combinatorial algorithms.

Lobbying: ILP-Based FPT (Lenstra) IV

Note: Dozens of other voting problems have ILP-FPT classification wrt. number of alternatives (columns), all lacking lower bounds / combinatorial algorithms.

General idea behind these ILP-FPT results:

1. Type of objects in solution can be bounded by parameter.
2. Only the number of objects from each type is important.
3. Plus additional linear constraints.

Lobbying: ILP-Based FPT (Lenstra) IV

Note: Dozens of other voting problems have ILP-FPT classification wrt. number of alternatives (columns), all lacking lower bounds / combinatorial algorithms.

General idea behind these ILP-FPT results:

1. Type of objects in solution can be bounded by parameter.
2. Only the number of objects from each type is important.
3. Plus additional linear constraints.

Remark:

- "Linear costs" induced by x_i objects of type i are easy to model.

Lobbying: ILP-Based FPT (Lenstra) IV

Note: Dozens of other voting problems have ILP-FPT classification wrt. number of alternatives (columns), all lacking lower bounds / combinatorial algorithms.

General idea behind these ILP-FPT results:

1. Type of objects in solution can be bounded by parameter.
2. Only the number of objects from each type is important.
3. Plus additional linear constraints.

Remark:

- "Linear costs" induced by x_i objects of type i are easy to model.
- "Non-linear, but non-negative, convex costs" can be handled with at most $u(x_i)$ additional non-integer variables and constraints per integer variable (with $u(x_i)$ upper-bounding x_i).

Lobbying: ILP-Based FPT (Lenstra) IV

Note: Dozens of other voting problems have ILP-FPT classification wrt. number of alternatives (columns), all lacking lower bounds / combinatorial algorithms.

General idea behind these ILP-FPT results:

1. Type of objects in solution can be bounded by parameter.
2. Only the number of objects from each type is important.
3. Plus additional linear constraints.

Remark:

- "Linear costs" induced by x_i objects of type i are easy to model.
- "Non-linear, but non-negative, convex costs" can be handled with at most $u(x_i)$ additional non-integer variables and constraints per integer variable (with $u(x_i)$ upper-bounding x_i).

⇝ Used to solve Weighted Set Multicover parameterized by "universe size" in FPT time

[Bredereck/Faliszewski/N./Skowron/Talmon, ADT 2015]

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n+m))$ many bits.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:
  - there is always a row containing 0s in at least half of the columns;

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n+m))$ many bits.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:
  - there is always a row containing 0s in at least half of the columns;
  - at most $(\lfloor \log m \rfloor + 1)$ greedy steps decrease g by one;

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n + m))$ many bits.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:
  - there is always a row containing 0s in at least half of the columns;
  - at most $(\lfloor \log m \rfloor + 1)$ greedy steps decrease g by one;
  - altogether $g \cdot (\lfloor \log m \rfloor + 1)$ steps.

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n+m))$ many bits.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.

?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:
  - there is always a row containing 0s in at least half of the columns;
  - at most $(\lfloor \log m \rfloor + 1)$ greedy steps decrease g by one;
  - altogether $g \cdot (\lfloor \log m \rfloor + 1)$ steps.
- Each row can be identified by $O(\log n)$ bits.

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n+m))$ many bits.

Lobbying: LOGSNP-Completeness I

Lobbying

In: $A \in \{0,1\}^{n \times m}$ and $k \leq n$.
?: Can we change at most k rows of A such that each column has more 1s than 0s?

Special situation: Each column needs $\leq g$ many additional ones.

A simple heuristic: Always take a row filled at least half with 0s.

- If $k \geq g \cdot (\lceil \log m \rceil + 1)$ then yes-instance:
  - there is always a row containing 0s in at least half of the columns;
  - at most $(\lfloor \log m \rfloor + 1)$ greedy steps decrease g by one;
  - altogether $g \cdot (\lfloor \log m \rfloor + 1)$ steps.
- Each row can be identified by $O(\log n)$ bits.

Consequently, there is a certificate for yes-instances using only $O(g \cdot \log^2(n+m))$ many bits.

$\rightsquigarrow$ Solvable in $O((n)^{g \log(m)+1} \cdot m)$ time—is this tight?

Lobbying: LOGSNP-Completeness II

[Papadimitriou and Yannakakis, Journal of Computer and System Sciences 1996]

The class LogSNP

- P $\approx$ log non-deterministic bits

Lobbying: LOGSNP-Completeness II

[Papadimitriou and Yannakakis, Journal of Computer and System Sciences 1996]

The class LogSNP

- P $\approx$ log non-deterministic bits
- NP $\approx$ poly non-deterministic bits

Lobbying: LOGSNP-Completeness II

[Papadimitriou and Yannakakis, Journal of Computer and System Sciences 1996]

The class LogSNP

- P $\approx$ log non-deterministic bits
- LogSNP $\approx$ $\log^2$ non-deterministic bits
- NP $\approx$ poly non-deterministic bits

LogSNP-completeness:

Polynomial-time reduction from/to the LogSNP-complete Rich Hypergraph Cover to/from Lobbying with constant g.

Lobbying: LOGSNP-Completeness II

[Papadimitriou and Yannakakis, Journal of Computer and System Sciences 1996]

The class LogSNP

- P $\approx$ log non-deterministic bits
- LogSNP $\approx$ $\log^2$ non-deterministic bits
- NP $\approx$ poly non-deterministic bits

LogSNP-completeness:

Polynomial-time reduction from/to the LogSNP-complete Rich Hypergraph Cover to/from Lobbying with constant g.

Rich Hypergraph Cover

In: A ground set $X = \{x_1, \ldots, x_n\}$ with an even number n of elements, m subsets $S_1, \ldots, S_m$ of X with $|S_i| \geq n/2$ for all i, and an integer k.

?: Does there exist a $Y \subseteq X$ with $Y \cap S_i \neq \emptyset$ for all i and $|Y| \leq k$?

Lobbying: LOGSNP-Completeness II

[Papadimitriou and Yannakakis, Journal of Computer and System Sciences 1996]

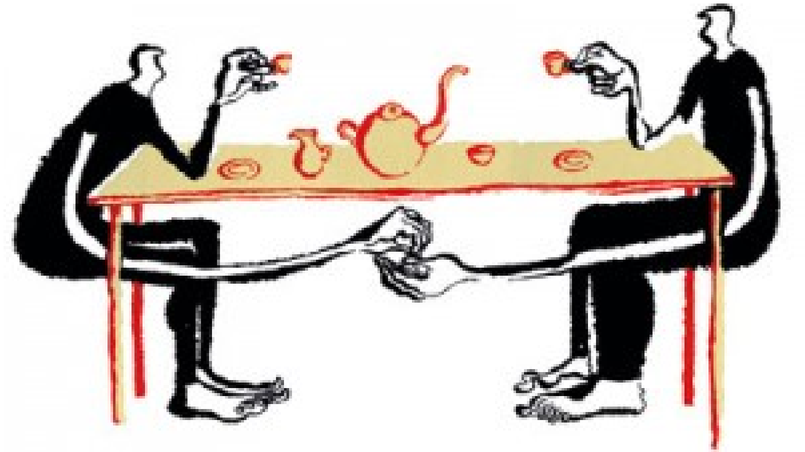
The class LogSNP

- P $\approx$ log non-deterministic bits
- LogSNP $\approx$ $\log^2$ non-deterministic bits
- NP $\approx$ poly non-deterministic bits

LogSNP-completeness:

Polynomial-time reduction from/to the LogSNP-complete Rich Hypergraph Cover to/from Lobbying with constant g.

Challenge: Further examples for LOGSNP-hardness.



/blog/wp-content/uploads/2014/10

Shift Bribery FPT Approximability I

Four symphonies:

a	b	c	d
“Arensky II”	“Brahms I”	“Copland III”	“Dvořák IX”

Three national symphony charts

Austria: $a \succ b \succ c \succ d$

Belgium: $a \succ b \succ c \succ d$

Chile: $b \succ d \succ c \succ a$

Shift Bribery FPT Approximability I

Four symphonies:

a	b	c	d
"Arensky II"	"Brahms I"	"Copland III"	"Dvořák IX"

Three national symphony charts

Austria: $a \succ b \succ c \succ d$

Belgium: $a \succ b \succ c \succ d$

Chile: $b \succ d \succ c \succ a$

Winner: Compute the "Borda top-ranked" symphony.

Shift Bribery FPT Approximability I

Four symphonies:

a	b	c	d
"Arensky II"	"Brahms I"	"Copland III"	"Dvořák IX"

Three national symphony charts

		Borda score
Austria:	$a \succ b \succ c \succ d$	$a: \; 3+3+0=6$
Belgium:	$a \succ b \succ c \succ d$	$b: \; 2+2+3=7$
Chile:	$b \succ d \succ c \succ a$	$c: \; 1+1+1=3$
		$d: \; 0+0+1=2$

Winner: Compute the "Borda top-ranked" symphony.

Shift Bribery FPT Approximability I

Four symphonies:

a	b	c	d
"Arensky II"	"Brahms I"	"Copland III"	"Dvořák IX"

Three national symphony charts

		Borda score
Austria:	$a \succ b \succ c \succ d$	$a: 3 + 3 + 0 = 6$
Belgium:	$a \succ b \succ c \succ d$	$b: 2 + 2 + 3 = 7$
Chile:	$b \succ d \succ c \succ a$	$c: 1 + 1 + 1 = 3$
		$d: 0 + 0 + 1 = 2$

Winner: Compute the "Borda top-ranked" symphony.

Shift Bribery FPT Approximability I

Four symphonies:

a	b	c	d
“Arensky II”	“Brahms I”	“Copland III”	“Dvořák IX”

Three national symphony charts

Austria: $a \succ b \succ c \succ d$

Belgium: $a \succ b \succ c \succ d$

Chile: $b \succ d \succ c \succ a$

Borda score

$a: \; 3+3+0=6$

$b: \; 2+2+3=7$

$c: \; 1+1+1=3$

$d: \; 0+0+1=2$

Winner: Compute the “Borda top-ranked” symphony.

Goal: Make c win by shifting it higher, for as little money as possible.

Shift Bribery: FPT Approximation II

Symphony charts example again

Austria: $a \succ b \succ c \succ d$

Belgium: $a \succ b \succ c \succ d$

Chile: $b \succ d \succ c \succ a$

$\mathcal{R}$ Shift Bribery

In: Election $E = (C, V = (v_1, \ldots, v_n))$,

?:

Shift Bribery: FPT Approximation II

Symphony charts example again

Austria: $a \succ b \succ c \succ d$

Belgium: $a \succ b \succ c \succ d$

Chile: $b \succ d \succ c \succ a$

$\mathcal{R}$ Shift Bribery

In: Election $E = (C, V = (v_1, \ldots, v_n))$, specific candidate $c \in C$,

?:

Shift Bribery: FPT Approximation II

Symphony charts example again

Austria: $a \succ b \succ c \succ d$ (shift prices: 0, 3)

Belgium: $a \succ b \succ c \succ d$ (shift prices: 0, 2)

Chile: $b \succ d \succ c \succ a$ (shift prices: 0, 1)

$\mathcal{R}$ Shift Bribery

In: Election $E = (C, V = (v_1, \ldots, v_n))$, specific candidate $c \in C$, price function list $\Pi = (\pi_1, \ldots, \pi_n)$,

?:

Shift Bribery: FPT Approximation II

Symphony charts example again

0 3

Austria: $a \succ b \succ c \succ d$

0 2

Belgium: $a \succ b \succ c \succ d$

0 1

Chile: $b \succ d \succ c \succ a$

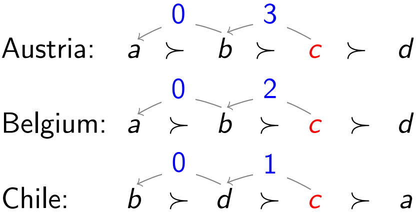
$\mathcal{R}$ Shift Bribery

In: Election $E = (C, V = (v_1, \ldots, v_n))$, specific candidate $c \in C$, price function list $\Pi = (\pi_1, \ldots, \pi_n)$, budget $B \in \mathbb{N}$.

?:

Shift Bribery: FPT Approximation II

Symphony charts example again



$\mathcal{R}$ Shift Bribery

In: Election $E = (C, V = (v_1, \ldots, v_n))$, specific candidate $c \in C$, price function list $\Pi = (\pi_1, \ldots, \pi_n)$, budget $B \in \mathbb{N}$.

?: $\exists$ shift action $\vec{s} = (s_1, \ldots, s_n)$ with $\Pi(\vec{s}) \leq B$ such that p is a winner according to rule $\mathcal{R}$?

Shift Bribery: FPT Approximation III

FPT-Approximation Scheme

A factor-$(1+\varepsilon)$ algorithm solving in FPT time wrt. ε and the parameter.

Theorem

For any voting rule $\mathcal{R}$ where Winner Determination is in FPT wrt. to the number n of voters, $\mathcal{R}$ Shift Bribery admits a factor-$(1+\varepsilon)$ approximation scheme; the running time is $O^*(\lceil n/\varepsilon \rceil^n)$.

Shift Bribery: FPT Approximation III

FPT-Approximation Scheme

A factor-$(1+\varepsilon)$ algorithm solving in FPT time wrt. ε and the parameter.

Theorem

For any voting rule $\mathcal{R}$ where Winner Determination is in FPT wrt. to the number n of voters, $\mathcal{R}$ Shift Bribery admits a factor-$(1+\varepsilon)$ approximation scheme; the running time is $O^*(\lceil n/\varepsilon \rceil^n)$.

Basic idea of the algorithm

- Guess the maximum budget $\pi_{\max}$ to spend on a single voter.
- Rescale and round the price functions to not exceed a given bound (dependent on ε and n).
- Find a cheapest successful shift action $\vec{s}$ for the rescaled instance in $f(\varepsilon, n)$ time.

One can show that this action $\vec{s}$ costs at most $(1+\varepsilon)$OPT.

[Bredereck/Chen/Falizewski/Nichterlein/N., AAAI 2014]

Combinatorial Shift Bribery: Inapproximability

Combinatorial flavor clearly makes the problem more complex...

Can we at least find approximate solutions?

Combinatorial Shift Bribery: Inapproximability

Combinatorial flavor clearly makes the problem more complex...

Can we at least find approximate solutions?

No: we obtain strong inapproximability results like this:

Theorem

Combinatorial Shift Bribery is inapproximable even in FPT-time with respect to the parameter budget B, even if there are only two candidates.

Combinatorial Shift Bribery: Inapproximability

Combinatorial flavor clearly makes the problem more complex...

Can we at least find approximate solutions?

No: we obtain strong inapproximability results like this:

Theorem

Combinatorial Shift Bribery is inapproximable even in FPT-time with respect to the parameter budget B, even if there are only two candidates.

Idea:

- Reduction from the W[2]-complete Set Cover problem.
- No solution for the constructed Combinatorial Shift Bribery instance can cost more than B (negative effects are essential for this).

[Bredereck/Faliszewski/N./Talmon, AAMAS 2015]

http://archive.feedblitz.com/217976/ 4087397

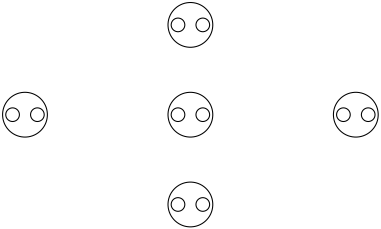
Network-Based Vertex Dissolution: ParaNP-Hardness I

Political (Re-)Districting: Example:
Given: Districts with $s = 2$ voters per district.

Goal: Increase the number of voters per district by $\Delta_s = 3$.
Method: Dissolve some districts and redistribute their voters.

Network-Based Vertex Dissolution: ParaNP-Hardness I

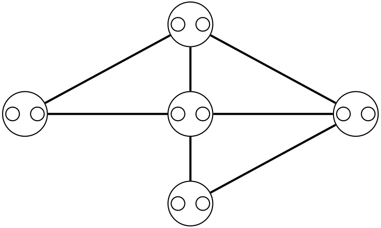
Political (Re-)Districting: Example:
Given: Districts with $s = 2$ voters per district.



Goal: Increase the number of voters per district by $\Delta_s = 3$.
Method: Dissolve some districts and redistribute their voters.

Network-Based Vertex Dissolution: ParaNP-Hardness I

Political (Re-)Districting: Example:
Given: Districts with $s = 2$ voters per district.

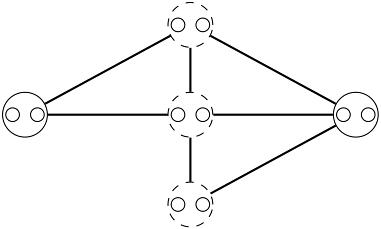


Problem: Cannot move voters arbitrarily!
Move voters of a dissolved district only to an adjacent non-dissolved district.

Network-Based Vertex Dissolution: ParaNP-Hardness I

Political (Re-)Districting: Example:

Given: Districts with $s = 2$ voters per district.

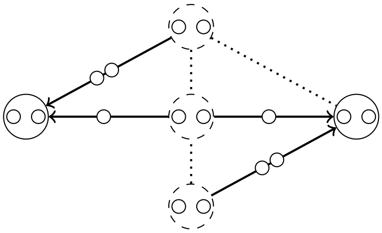


Solution: Dissolve the middle districts.

Network-Based Vertex Dissolution: ParaNP-Hardness I

Political (Re-)Districting: Example:

Given: Districts with $s = 2$ voters per district.



Network-Based Vertex Dissolution: ParaNP-Hardness I

Political (Re-)Districting: Example:

Result: Districts with 5 voters per district.

Network-Based Vertex Dissolution: ParaNP-Hardness II

What do we know from structural results?

$s \backslash \Delta$	1	2	3	4	5	6
1	?	?	?	?	?	?
2	?	?	?	?	?	?
3	?	?	?	?	?	?
4	?	?	?	?	?	?
5	?	?	?	?	?	?
6	?	?	?	?	?	?

Network-Based Vertex Dissolution: ParaNP-Hardness II

$s \backslash \Delta$	1	2	3	4	5	6
1	P	?	?	?	?	?
2	?	P	?	?	?	?
3	?	?	P	?	?	?
4	?	?	?	P	?	?
5	?	?	?	?	P	?
6	?	?	?	?	?	P

Lemma 1: There exists an (s,s)-dissolution for an undirected graph G if and only if G has a perfect matching.

Network-Based Vertex Dissolution: ParaNP-Hardness II

$s \backslash \Delta$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	?	P	?	NP-h	?	NP-h
3	?	?	P	?	?	NP-h
4	?	?	?	P	?	?
5	?	?	?	?	P	?
6	?	?	?	?	?	P

Idea: There exists a $(t \cdot \Delta_s, \Delta_s)$-dissolution for an undirected graph G if and only if G has a t-star partition.
Partitioning a graph into t-stars is NP-hard.

Network-Based Vertex Dissolution III: Para-NP Hardness

Symmetry with respect to s and Δ_s:

Lemma 2: There exists an (s, Δ_s)-dissolution for an undirected graph G if and only if there exists a (Δ_s, s)-dissolution for G.

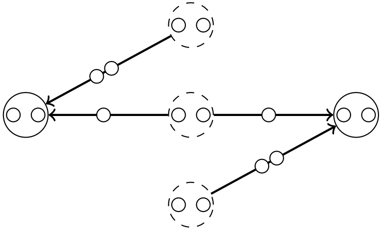
Network-Based Vertex Dissolution III: Para-NP Hardness

Symmetry with respect to s and Δ_s:

Lemma 2: There exists an (s, Δ_s)-dissolution for an undirected graph G if and only if there exists a (Δ_s, s)-dissolution for G.

$s = 2$

$\Delta_s = 3$



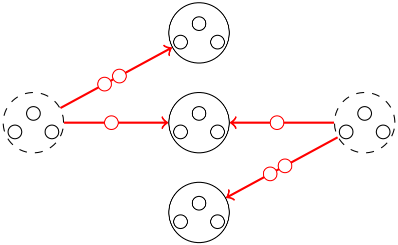
Network-Based Vertex Dissolution III: Para-NP Hardness

Symmetry with respect to s and Δ_s:

Lemma 2: There exists an (s, Δ_s)-dissolution for an undirected graph G if and only if there exists a (Δ_s, s)-dissolution for G.

$s = 3$

$\Delta_s = 2$



Interpret the voter movement backwards.
We put the voters to receive into the districts.
Move to each dissolved districts two voters.

Network-Based Vertex Dissolution II: ParaNP-Hardness III

P vs NP dichotomy:

$s \backslash \Delta_s$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	?	P	?	NP-h	?	NP-h
3	?	?	P	?	?	NP-h
4	?	?	?	P	?	?
5	?	?	?	?	P	?
6	?	?	?	?	?	P

Network-Based Vertex Dissolution II: ParaNP-Hardness III

$s \backslash \Delta$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	NP-h	P	?	NP-h	?	NP-h
3	NP-h	?	P	?	?	NP-h
4	NP-h	NP-h	?	P	?	?
5	NP-h	?	?	?	P	?
6	NP-h	NP-h	NP-h	?	?	P

Idea: There exists an (s, Δ_s)-dissolution for an undirected graph G if and only if there exists a (Δ_s, s)-dissolution for G.

Network-Based Vertex Dissolution II: ParaNP-Hardness III

$s \backslash \Delta_s$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	NP-h	P	NP-h	NP-h	NP-h	NP-h
3	NP-h	NP-h	P	NP-h	NP-h	NP-h
4	NP-h	NP-h	NP-h	P	NP-h	NP-h
5	NP-h	NP-h	NP-h	NP-h	P	NP-h
6	NP-h	NP-h	NP-h	NP-h	NP-h	P

Finally: The rest of the cells are covered by an NP-hardness reduction from **Exact Cover by t-Sets**

Network-Based Vertex Dissolution II: ParaNP-Hardness III

$s \backslash \Delta_s$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	NP-h	P	NP-h	NP-h	NP-h	NP-h
3	NP-h	NP-h	P	NP-h	NP-h	NP-h
4	NP-h	NP-h	NP-h	P	NP-h	NP-h
5	NP-h	NP-h	NP-h	NP-h	P	NP-h
6	NP-h	NP-h	NP-h	NP-h	NP-h	P

Theorem

If $s = \Delta_s$, then DISSOLUTION solvable in $O(n^\omega)$ time (where ω is the matrix multiplication exponent); otherwise NP-complete.

[Bevern/Bredereck/Chen/Froese/N./Woeginger, SIAM J. Discrete Math. 2015]

Network-Based Vertex Dissolution: ParaNP-Hardness IV

P vs. NP dichotomy for planar case:

$s \backslash \Delta_s$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	NP-h	P	?	NP-h	?	NP-h
3	NP-h	?	P	?	?	NP-h
4	NP-h	NP-h	?	P	?	?
5	NP-h	?	?	?	P	?
6	NP-h	NP-h	NP-h	?	?	P

Planar: a lot of open cases!

Network-Based Vertex Dissolution: ParaNP-Hardness IV

P vs. NP dichotomy for planar case:

$s \backslash \Delta_s$	1	2	3	4	5	6
1	P	NP-h	NP-h	NP-h	NP-h	NP-h
2	NP-h	P	?	NP-h	?	NP-h
3	NP-h	?	P	?	?	NP-h
4	NP-h	NP-h	?	P	?	?
5	NP-h	?	?	?	P	?
6	NP-h	NP-h	NP-h	?	?	P

Planar: a lot of open cases!

Interesting subproblem: What is the complexity of **Planar Exact Cover by t-Sets** for $t \geq 3$?

http://www.stephaniemcmillan.org/codegreen/comics/2011-03-21-push-their-agenda.jpg

Majority-wise Accepted Ballot: W[2] vs W[2](Maj) I

Unanimously Accepted Ballot

In: A set $\mathcal{P}$ of m proposals; a set $\mathcal{V}$ of n voters with favorite ballots $B_1, \ldots, B_n \subseteq \mathcal{P}$; an agenda $Q_+ \subseteq \mathcal{P}$.

?: Is there a ballot Q with $Q_+ \subseteq Q \subseteq \mathcal{P}$ which **every** single voter i accepts (that is, $|B_i \cap Q| > |Q|/2$)?

Majority-wise Accepted Ballot: W[2] vs W[2](Maj) I

Unanimously Accepted Ballot

In: A set $\mathcal{P}$ of m proposals; a set $\mathcal{V}$ of n voters with favorite ballots $B_1, \ldots, B_n \subseteq \mathcal{P}$; an agenda $Q_+ \subseteq \mathcal{P}$.

?: Is there a ballot Q with $Q_+ \subseteq Q \subseteq \mathcal{P}$ which **every** single voter i accepts (that is, $|B_i \cap Q| > |Q|/2$)?

Majoritywise Accepted Ballot

In: A set $\mathcal{P}$ of m proposals; a set $\mathcal{V}$ of n voters with favorite ballots $B_1, \ldots, B_n \subseteq \mathcal{P}$; an agenda $Q_+ \subseteq \mathcal{P}$.

?: Is there a ballot Q with $Q_+ \subseteq Q \subseteq \mathcal{P}$ which a **strict majority** of the voters accepts (that is, $|B_i \cap Q| > |Q|/2$)?

Majority-wise Accepted Ballot: Example

Example with five proposals and four voters.

- $\mathcal{P} = \{p_1, p_2, p_3, p_4, p_5\}$
- $B_1 = \{p_2, p_3, p_4\}$, $B_2 = \{p_1, p_3, p_5\}$, $B_3 = \{p_1, p_2, p_4\}$, $B_4 = \{p_1, p_2, p_3\}$

Majority-wise Accepted Ballot: Example

Example with five proposals and four voters.

- $\mathcal{P} = \{p_1, p_2, p_3, p_4, p_5\}$
- $B_1 = \{p_2, p_3, p_4\}$, $B_2 = \{p_1, p_3, p_5\}$, $B_3 = \{p_1, p_2, p_4\}$, $B_4 = \{p_1, p_2, p_3\}$

Ballot matrix:

p_1	p_2	p_3	p_4	p_5
0	1	1	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Note: The ballot $\{p_1, p_2, p_3\}$ would be accepted by all voters. However, although less than half of the voters like p_5, the commission can push through p_5 by proposing $\{p_1, p_2, p_3, p_4, p_5\}$.

Majority-wise Accepted Ballot: Example

Example with five proposals and four voters.

- $\mathcal{P} = \{p_1, p_2, p_3, p_4, p_5\}$
- $B_1 = \{p_2, p_3, p_4\}$, $B_2 = \{p_1, p_3, p_5\}$, $B_3 = \{p_1, p_2, p_4\}$, $B_4 = \{p_1, p_2, p_3\}$

Ballot matrix:

p_1	p_2	p_3	p_4	p_5
0	1	1	1	0
1	0	1	0	1
1	1	0	1	0
1	1	1	0	0

Interesting: No monotonicity w.r.t. solution sizes.

Majority-wise Accepted Ballot: W[2] vs W[2](Maj) II

[Alon, Bredereck, Chen, Kratsch, N., Woeginger, ACM TEAC 2015]

With respect to the parameter size $|Q|$ of the solution, both variants are easily shown to be W[2]-hard (via Hitting Set).

Majority-wise Accepted Ballot: W[2] vs W[2](Maj) II

[Alon, Bredereck, Chen, Kratsch, N., Woeginger, ACM TEAC 2015]

With respect to the parameter size $|Q|$ of the solution, both variants are easily shown to be W[2]-hard (via Hitting Set).

Unanimously Accepted Ballot

W[2]-membership can be show by a quite technical reduction to the W[2]-complete Independent Dominating Set problem.

Majority-wise Accepted Ballot: W[2] vs W[2](Maj) II

[Alon, Bredereck, Chen, Kratsch, N., Woeginger, ACM TEAC 2015]

With respect to the parameter size $|Q|$ of the solution, both variants are easily shown to be W[2]-hard (via Hitting Set).

Unanimously Accepted Ballot

W[2]-membership can be show by a quite technical reduction to the W[2]-complete Independent Dominating Set problem.

Majoritywise Accepted Ballot

Showing W[2]-membership seems quite challenging.

Interestingly: Considering the class W[2](Maj) instead of W[2], that is, allowing only majority gates instead of AND/OR gates in the boolean circuits, one can show membership relatively easy.

Concluding Remarks

COMSOC offers a rich variety of combinatorial problems (touching permutations, partial orders, sets, matrices, graphs, etc.), many of these are NP-hard.

Concluding Remarks

COMSOC offers a rich variety of combinatorial problems (touching permutations, partial orders, sets, matrices, graphs, etc.), many of these are NP-hard.

Lower bounds in COMSOC so far mostly of the form

- W-hardness (W-completeness) or
- ParaNP-hardness or
- no polynomial-size problem kernel unless...

$\rightsquigarrow$ ETH-based lower bounds largely unexplored...

Concluding Remarks

COMSOC offers a rich variety of combinatorial problems (touching permutations, partial orders, sets, matrices, graphs, etc.), many of these are NP-hard.

Lower bounds in COMSOC so far mostly of the form

- W-hardness (W-completeness) or
- ParaNP-hardness or
- no polynomial-size problem kernel unless...

$\rightsquigarrow$ ETH-based lower bounds largely unexplored...

General challenges:

- Lower bounds for "Lenstra-ILP-FPT results"?
- Exploration of LOGSNP-completeness for lower bounds?
- W[2] = W[2](Maj)?

  [Fellows, Flum, Hermelin, Müller, Rosamond, TOCS 2010]
- Running time lower bounds for FPT approximation schemes?

Some General COMSOC Literature

- Yann Chevaleyre, Ulle Endriss, Jérôme Lang, Nicolas Maudet: A Short Introduction to Computational Social Choice. SOFSEM 2007.
- Felix Brandt, Vincent Conitzer, and Ulle Endriss. Computational Social Choice. In Gerhard Weiss, editor, Multiagent Systems, Chapter 6, MIT Press, 2nd edition, 2013.
- Nadja Betzler, Robert Bredereck, Jiehua Chen, Rolf Niedermeier: Studies in Computational Aspects of Voting – A Parameterized Complexity Perspective. The Multivariate Algorithmic Revolution and Beyond, Springer 2012.
- Robert Bredereck, Jiehua Chen, Piotr Faliszewski, Jiong Guo, Rolf Niedermeier, Gerhard J. Woeginger: Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges. Tsinghua Science and Technology 2014
- Jörg Rothe, editor: Economics and Computation, Springer 2015.
- Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, Ariel D. Procaccia, editors: Handbook of Computational Social Choice, Cambridge University Press 2015.