On the Comlpexity of Chamberlin-Courant on Almost Structured Profiles

Chinmay Sonar P R Vaidyanathan Neeldhara Misra

Indian Institute of Technology Gandhinagar

Algorithmic Decision Theory, 2017

Outline

Introduction

Structured Profiles

- In practical settings, elections usually contain some structure

Introduction

Structured Profiles

- In practical settings, elections usually contain some structure
- Examples include Single-peaked(SP) and Single-crossing(SC)

Introduction
Structured Profiles

- In practical settings, elections usually contain some structure
- Examples include Single-peaked(SP) and Single-crossing(SC)
- When a voting profile has such structure, we refer it to as structured profile

Introduction

Structured Profiles

- In practical settings, elections usually contain some structure
- Examples include Single-peaked(SP) and Single-crossing(SC)
- When a voting profile has such structure, we refer it to as structured profile
- Many NP-hard voting rules turn out to be polynomially solvable for structured profiles

Introduction

Nearly Structured profiles

- “Nearly Structured” profiles capture more of real-world scenarios

Introduction

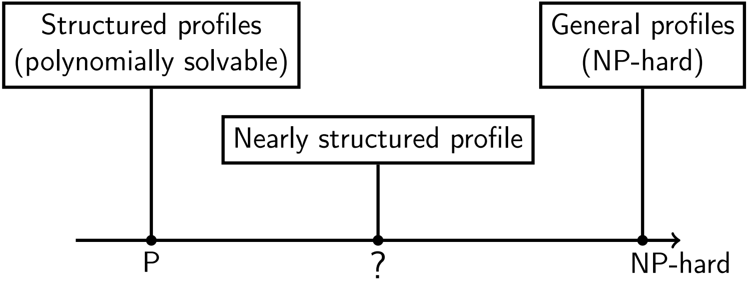
Nearly Structured profiles

- "Nearly Structured" profiles capture more of real-world scenarios
- Our goal was to study the complexity of Chamberlin-Courant in the region between structured profiles and general profiles

Introduction

Nearly Structured profiles

- "Nearly Structured" profiles capture more of real-world scenarios
- Our goal was to study the complexity of Chamberlin-Courant in the region between structured profiles and general profiles



Introduction

What is a Nearly Structured Profile?

- A nearly structured profile is a profile that is “close” to admitting structure

Introduction

What is a Nearly Structured Profile?

- A nearly structured profile is a profile that is “close” to admitting structure
- A popular notion of closeness to structure is by deletion of a small part of the profile.

Introduction

What is a Nearly Structured Profile?

- A nearly structured profile is a profile that is “close” to admitting structure
- A popular notion of closeness to structure is by deletion of a small part of the profile.
- For example, one might say that a profile is k-close to being single-crossing by voter deletion to mean that there exists a subset S of at most k voters such that the election instance projected on $V \setminus S$ is single-crossing.

Related Work

- The problem of finding deletion sets to single-peaked and single-crossing has been studied [1]

[1] Elkind & Lackner: On detecting nearly structured preference profiles

[2] Chamberlin & Courant: Representative deliberations and representative decisions

[3] Procaccia, Rosenschein & Zohar: Multi-Winner Elections: Complexity of Manipulation, Control, and Winner-Determination

[4] Betzler, Slinko & Uhlmann: On the computation of fully proportional representation

[5] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

Related Work

- The problem of finding deletion sets to single-peaked and single-crossing has been studied [1]
- Application of this work to NP-hard voting rules seems promising

[1] Elkind & Lackner: On detecting nearly structured preference profiles

[2] Chamberlin & Courant: Representative deliberations and representative decisions

[3] Procaccia, Rosenschein & Zohar: Multi-Winner Elections: Complexity of Manipulation, Control, and Winner-Determination

[4] Betzler, Slinko & Uhlmann: On the computation of fully proportional representation

[5] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

Related Work

- The problem of finding deletion sets to single-peaked and single-crossing has been studied [1]
- Application of this work to NP-hard voting rules seems promising
- Chamberlin-Courant(CC) is one such rule [2]

[1] Elkind & Lackner: On detecting nearly structured preference profiles

[2] Chamberlin & Courant: Representative deliberations and representative decisions

[3] Procaccia, Rosenschein & Zohar: Multi-Winner Elections: Complexity of Manipulation, Control, and Winner-Determination

[4] Betzler, Slinko & Uhlmann: On the computation of fully proportional representation

[5] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

Related Work

- The problem of finding deletion sets to single-peaked and single-crossing has been studied [1]
- Application of this work to NP-hard voting rules seems promising
- Chamberlin-Courant(CC) is one such rule [2]
- CC is NP-hard for the general setting [3] and polynomially solvable for structured profiles [4] [5]

---

[1] Elkind & Lackner: On detecting nearly structured preference profiles

[2] Chamberlin & Courant: Representative deliberations and representative decisions

[3] Procaccia, Rosenschein & Zohar: Multi-Winner Elections: Complexity of Manipulation, Control, and Winner-Determination

[4] Betzler, Slinko & Uhlmann: On the computation of fully proportional representation

[5] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

Results Summary

- We show tractability results for Chamberlin-Courant on profiles that are k candidates/voters away

[6] MM: Minimax (egalitarian version)

[7] Denotes perfectly structured profiles

Results Summary

- We show tractability results for Chamberlin-Courant on profiles that are k candidates/voters away
- We also show severe intractability results for other natural generalizations of these domains for Chamberlin-Courant

[6] MM: Minimax (egalitarian version)

[7] Denotes perfectly structured profiles

Results Summary

- We show tractability results for Chamberlin-Courant on profiles that are k candidates/voters away
- We also show severe intractability results for other natural generalizations of these domains for Chamberlin-Courant

Table: Parameterized Complexity of considered multiwinner problems

	SP CC	SP MM[6] CC	SC CC	SC MM CC
Struct[7]	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$	$\mathcal{O}(n^2mk)$	$\mathcal{O}(n^2mk)$
VDel	$2^{Rk}\mathcal{O}(nm^2)$	$2^{Rk}\mathcal{O}(nm)$	$2^{Rk}\mathcal{O}(n^2mk)$	$2^{Rk}\mathcal{O}(n^2mk)$
CDel	$2^{k}\mathcal{O}(nm^2)$	$2^{k}\mathcal{O}(nm)$	$2^{k}\mathcal{O}(n^2mk)$	$2^{k}\mathcal{O}(n^2mk)$

[6] MM: Minimax (egalitarian version)

[7] Denotes perfectly structured profiles

Preliminaries

- Misrepresentation function: For an m-candidate election with votes specified as complete order over set of candidates, a *dissatisfaction function* is given by a non-decreasing function $\alpha\colon [m] \to \mathbb{N}$ with $\alpha(1) = 0$.
  Ex. Borda: $\alpha_B^m(i) = \alpha_B(i) = i - 1$

Preliminaries

- Misrepresentation function: For an m-candidate election with votes specified as complete order over set of candidates, a *dissatisfaction function* is given by a non-decreasing function $\alpha\colon [m] \to \mathbb{N}$ with $\alpha(1) = 0$.
  Ex. Borda: $\alpha_B^m(i) = \alpha_B(i) = i - 1$
- Assignment function: k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi\colon V \to C$ such that $|\Phi(V)| \leqslant k$

Preliminaries

- Misrepresentation function: For an m-candidate election with votes specified as complete order over set of candidates, a *dissatisfaction function* is given by a non-decreasing function $\alpha \colon [m] \to \mathbb{N}$ with $\alpha(1) = 0$.
  Ex. Borda: $\alpha_B^m(i) = \alpha_B(i) = i - 1$
- Assignment function: k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi \colon V \to C$ such that $|\Phi(V)| \leqslant k$
- Aggregation Function: Used to measure the quality of Assignment Function. We use following two in our work:

Preliminaries

- Misrepresentation function: For an m-candidate election with votes specified as complete order over set of candidates, a *dissatisfaction function* is given by a non-decreasing function $\alpha\colon [m] \to \mathbb{N}$ with $\alpha(1) = 0$.
  Ex. Borda: $\alpha_B^m(i) = \alpha_B(i) = i - 1$
- Assignment function: k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi\colon V \to C$ such that $|\Phi(V)| \leqslant k$
- Aggregation Function: Used to measure the quality of Assignment Function. We use following two in our work:
  - $\ell_1 = \sum_{i=1,\ldots,n} \alpha(\mathrm{pos}_{v_i}(\Phi(v_i)))$, and

Preliminaries

- Misrepresentation function: For an m-candidate election with votes specified as complete order over set of candidates, a *dissatisfaction function* is given by a non-decreasing function $\alpha\colon [m] \to \mathbb{N}$ with $\alpha(1) = 0$.
  Ex. Borda: $\alpha_B^m(i) = \alpha_B(i) = i - 1$
- Assignment function: k-*CC-assignment function* for an election $E = (C, V)$ is a mapping $\Phi\colon V \to C$ such that $|\Phi(V)| \leqslant k$
- Aggregation Function: Used to measure the quality of Assignment Function. We use following two in our work:
  - $\ell_1 = \sum_{i=1,\dots,n} \alpha(\mathrm{pos}_{v_i}(\Phi(v_i)))$, and
  - $\ell_\infty(\Phi) = \max_{i=1,\dots,n} \alpha(\mathrm{pos}_{v_i}(\Phi(v_i)))$

Chamberlin-Courant Voting Rule

Chamberlin Courant-rule

For every family of dissatisfaction functions $\alpha = (\alpha^m)_{m=1}^{\infty}$, and every $\ell \in \{\ell_1, \ell_\infty\}$, the *$\alpha$-$\ell$-CC voting rule* is a mapping that takes an election $E = (C, V)$ and a positive integer k with $k \leqslant |C|$ as its input, and returns a k-CC-assignment function Φ for E that minimizes $\ell(\Phi)$. [a] [b]

[a] Chamberlin & Courant: Representative deliberations and representative decisions

[b] Betzler, Slinko & Uhlmann: On the Computation of Fully Proportional Representation

Structured Profiles

- Single-crossing profiles

Structured Profiles

- Single-crossing profiles

	3	2	1	0
v_1	a	b	c	d
v_2	b	a	d	c
v_3	c	b	a	d
v_4	c	d	b	a
v_5	d	c	b	a

Structured Profiles

- Single-crossing profiles

	3	2	1	0
v_1	a	b	c	d
v_2	b	a	d	c
v_3	c	b	a	d
v_4	c	d	b	a
v_5	d	c	b	a

Structured Profiles

- Single-crossing profiles

	3	2	1	0
v_1	a	b	c	d
v_2	b	a	d	c
v_3	c	b	a	d
v_4	c	d	b	a
v_5	d	c	b	a

Modulators

k-Modulator

A profile is said to have a candidate/voter modulator of size k, if $\exists$ a subset of size at most k candidates/voters such that the restriction of the profile to all but chosen candidates belongs to domain $\mathcal{D}$.

[8] Bredreck, et al.: Are there any nicely structured preference profiles nearby?

[9] Elkind & Lackner: On detecting nearly structured preference profiles

Modulators

k-Modulator

A profile is said to have a candidate/voter modulator of size k, if $\exists$ a subset of size at most k candidates/voters such that the restriction of the profile to all but chosen candidates belongs to domain $\mathcal{D}$.

- Polynomial time algorithm for finding voter modulator and approximation algorithm for candidate modulator have been shown for Single-crossing profiles [8] [9]

[8] Bredreck, et al.: Are there any nicely structured preference profiles nearby?

[9] Elkind & Lackner: On detecting nearly structured preference profiles

Modulators

k-Modulator

A profile is said to have a candidate/voter modulator of size k, if $\exists$ a subset of size at most k candidates/voters such that the restriction of the profile to all but chosen candidates belongs to domain $\mathcal{D}$.

- Polynomial time algorithm for finding voter modulator and approximation algorithm for candidate modulator have been shown for Single-crossing profiles [8] [9]
- $(\ell, \mathcal{D})$-CC Via χ: denotes aggregation function ℓ over domain D and χ can be candidate or voter modulator

[8] Bredreck, et al.: Are there any nicely structured preference profiles nearby?

[9] Elkind & Lackner: On detecting nearly structured preference profiles

$(\ell, \mathcal{D})$-CC Extension

- We are given an election $E = (C, V)$, along with the partition of the set of candidates $C = D \uplus G \uplus B$, here G, B represents, partially formed committee and candidates which cannot be part of committee respectively. D represents to be decided through the run of algorithm

$(\ell, \mathcal{D})$-CC Extension

- We are given an election $E = (C, V)$, along with the partition of the set of candidates $C = D \uplus G \uplus B$, here G, B represents, partially formed committee and candidates which cannot be part of committee respectively. D represents to be decided through the run of algorithm
- We are given that the election restricted to (D, V) belongs to domain $\mathcal{D}$

$(\ell, \mathcal{D})$-CC Extension

- We are given an election $E = (C, V)$, along with the partition of the set of candidates $C = D \uplus G \uplus B$, here G, B represents, partially formed committee and candidates which cannot be part of committee respectively. D represents to be decided through the run of algorithm
- We are given that the election restricted to (D, V) belongs to domain $\mathcal{D}$
- Objective is to find committee of size b that respects the semantics of (D, G, B) with misrepresentation score at most R

(ℓ, SC)-CC Extension

- Our algorithm builds upon known polynomial time algorithm for Single Crossing profiles [10]

[10] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

(ℓ, SC)-CC Extension

- Our algorithm builds upon known polynomial time algorithm for Single Crossing profiles [10]
- Algorithm is independent of choice of aggregation function

[10] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

(ℓ, SC)-CC Extension

- Our algorithm builds upon known polynomial time algorithm for Single Crossing profiles [10]
- Algorithm is independent of choice of aggregation function
- The "Contiguous block" property continues to hold for modified algorithm which is key for correctness of our algorithm

[10] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

(ℓ, SC)-CC Extension

- Our algorithm builds upon known polynomial time algorithm for Single Crossing profiles [10]
- Algorithm is independent of choice of aggregation function
- The "Contiguous block" property continues to hold for modified algorithm which is key for correctness of our algorithm
- Dynamic programming algorithm admits polynomial running time in terms of number of candidates, voters and committee size

[10] Skowron, Yu, Faliszewski & Elkind: The complexity of fully proportional representation for single-crossing electorates

$(\ell, \mathcal{D})$-CC Extension via Candidates

Construction

- Given a k-sized candidate modulator - X

$(\ell, \mathcal{D})$-CC Extension via Candidates

Construction

- Given a k-sized candidate modulator - X
- Let $Y \subseteq X$ be an arbitrary subset

$(\ell, \mathcal{D})$-CC Extension via Candidates

Construction

- Given a k-sized candidate modulator - X
- Let $Y \subseteq X$ be an arbitrary subset
- Set $D := C \setminus X$, $G := Y$ and $B := X \setminus Y$

$(\ell, \mathcal{D})$-CC Extension via Candidates

Construction

- Given a k-sized candidate modulator - X
- Let $Y \subseteq X$ be an arbitrary subset
- Set $D := C \setminus X$, $G := Y$ and $B := X \setminus Y$
- If resulting $(\ell, \mathcal{D})$-CC Extension instance is a Yes instance for some $Y \subset X$, return Yes

$(\ell, \mathcal{D})$-CC Extension via Candidates

Construction

- Given a k-sized candidate modulator - X
- Let $Y \subseteq X$ be an arbitrary subset
- Set $D := C \setminus X$, $G := Y$ and $B := X \setminus Y$
- If resulting $(\ell, \mathcal{D})$-CC Extension instance is a Yes instance for some $Y \subset X$, return Yes
- If no such Y yields a Yes, return No

$(\ell, \mathcal{D})$-CC Extension via Candidates

Correctness and Runtime

- Let $C^\star$ be a optimal committee, and $Y^\star$ be $C^\star \cap X$

$(\ell, \mathcal{D})$-CC Extension via Candidates

Correctness and Runtime

- Let $C^\star$ be a optimal committee, and $Y^\star$ be $C^\star \cap X$
- $D := C \setminus X$, $G := Y^\star$ and $B := X \setminus Y^\star$

$(\ell, \mathcal{D})$-CC Extension via Candidates

Correctness and Runtime

- Let $C^\star$ be a optimal committee, and $Y^\star$ be $C^\star \cap X$
- $D := C \setminus X$, $G := Y^\star$ and $B := X \setminus Y^\star$
- Since X is a candidate modulator, election induced by (D, V) belongs to domain $\mathcal{D}$

$(\ell, \mathcal{D})$-CC Extension via Candidates

Correctness and Runtime

- Let $C^\star$ be a optimal committee, and $Y^\star$ be $C^\star \cap X$
- $D := C \setminus X$, $G := Y^\star$ and $B := X \setminus Y^\star$
- Since X is a candidate modulator, election induced by (D, V) belongs to domain $\mathcal{D}$
- $E = (C, V); (D, G, B)$ is a valid input to $(\ell, \mathcal{D})$-CC Extension and $C^\star$ is a valid solution

$(\ell, \mathcal{D})$-CC Extension via Candidates

Correctness and Runtime

- Let $C^\star$ be a optimal committee, and $Y^\star$ be $C^\star \cap X$
- $D := C \setminus X$, $G := Y^\star$ and $B := X \setminus Y^\star$
- Since X is a candidate modulator, election induced by (D, V) belongs to domain $\mathcal{D}$
- $E = (C, V); (D, G, B)$ is a valid input to $(\ell, \mathcal{D})$-CC Extension and $C^\star$ is a valid solution
- Runtime(FPT): $2^k q(n, m)$, where $q(n, m)$ is the time required for $(\ell, \mathcal{D})$-CC Extension

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee
- Let $\mu(v)$ denote the candidate representing voter v

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee
- Let $\mu(v)$ denote the candidate representing voter v
- Let $d(v)$ denote the set of candidates ranked higher than $\mu(v)$ by voter v

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee
- Let $\mu(v)$ denote the candidate representing voter v
- Let $d(v)$ denote the set of candidates ranked higher than $\mu(v)$ by voter v
- Setting $G := \bigcup\limits_{v \in X} \mu(v)$, $B := \bigcup\limits_{v \in X} d(v)$ and $D := C \setminus (G \cup B)$ invoke $(\ell, \mathcal{D})$-CC Extension

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee
- Let $\mu(v)$ denote the candidate representing voter v
- Let $d(v)$ denote the set of candidates ranked higher than $\mu(v)$ by voter v
- Setting $G := \bigcup_{v \in X} \mu(v)$, $B := \bigcup_{v \in X} d(v)$ and $D := C \setminus (G \cup B)$ invoke $(\ell, \mathcal{D})$-CC Extension
- To ensure exclusivity of B and G, any guess in which there exists $u, v \in G$, such that $u \in d(v)$ can be rejected

$(\ell, \mathcal{D})$-CC Extension via voters

Construction

- Given k-sized voter modulator - X
- In an arbitrary manner guess the candidates representing each voter in X avoiding conflicts within the sub-committee
- Let $\mu(v)$ denote the candidate representing voter v
- Let $d(v)$ denote the set of candidates ranked higher than $\mu(v)$ by voter v
- Setting $G := \bigcup_{v \in X} \mu(v)$, $B := \bigcup_{v \in X} d(v)$ and $D := C \setminus (G \cup B)$ invoke $(\ell, \mathcal{D})$-CC Extension
- To ensure exclusivity of B and G, any guess in which there exists $u, v \in G$, such that $u \in d(v)$ can be rejected
- If there exists some guess for which $(\ell, \mathcal{D})$-CC Extension yields Yes, return Yes else return No

$(\ell, \mathcal{D})$-CC Extension via Voters

Correctness and Runtime

- Correctness argument similar to (ℓ, SC)-CC Extension via Candidates case

$(\ell, \mathcal{D})$-CC Extension via Voters

Correctness and Runtime

- Correctness argument similar to (ℓ, SC)-CC Extension via Candidates case
- Runtime(XP): $n^k q(n, m)$, where $q(n, m)$ is the time required for $(\ell, \mathcal{D})$-CC Extension

$(\ell, \mathcal{D})$-CC Extension via Voters

Correctness and Runtime

- Correctness argument similar to (ℓ, SC)-CC Extension via Candidates case
- Runtime(XP): $n^k q(n, m)$, where $q(n, m)$ is the time required for $(\ell, \mathcal{D})$-CC Extension
- Open problem: Smarter guessing could yield an FPT runtime, alternatively W-hardness proof could rule out that possibility

Hardness for 3-crossing profiles

Definition of 3-crossing profile

- Another natural way of generalizing the notion of single-crossing profile

Hardness for 3-crossing profiles

Definition of 3-crossing profile

- Another natural way of generalizing the notion of single-crossing profile

r-crossing Profile

There exists an ordering of votes such that the pairwise preference between candidates flips at most r times

Hardness for 3-crossing profiles

Definition of 3-crossing profile

- Another natural way of generalizing the notion of single-crossing profile

r-crossing Profile

There exists an ordering of votes such that the pairwise preference between candidates flips at most r times

- $r = 1$ is the familiar single-crossing setting

Hardness for 3-crossing profiles
Definition of 3-crossing profile

- Another natural way of generalizing the notion of single-crossing profile

r-crossing Profile

There exists an ordering of votes such that the pairwise preference between candidates flips at most r times

- $r = 1$ is the familiar single-crossing setting
- Here, we focus on the $r = 3$ i.e. 3-crossing profiles for CC rule

Hardness for 3-crossing profiles

Definition of LSAT

We show a reduction from LSAT to computing an ℓ_∞ -CC committee on 3-crossing profiles with $R \leqslant 2$

[11] Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard

Hardness for 3-crossing profiles

Definition of LSAT

We show a reduction from LSAT to computing an ℓ_∞ -CC committee on 3-crossing profiles with $R \leqslant 2$

LSAT

- Variant of SAT where each clause has at most three literals

LSAT is known to be NP-hard [11]

[11] Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard

Hardness for 3-crossing profiles
Definition of LSAT

We show a reduction from LSAT to computing an ℓ_∞ -CC committee on 3-crossing profiles with $R \leqslant 2$

LSAT

- Variant of SAT where each clause has at most three literals
- Literals can be sorted such that every clause has consecutive literals

LSAT is known to be NP-hard [11]

[11] Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard

Hardness for 3-crossing profiles

Definition of LSAT

We show a reduction from LSAT to computing an ℓ_∞ -CC committee on 3-crossing profiles with $R \leqslant 2$

LSAT

- Variant of SAT where each clause has at most three literals
- Literals can be sorted such that every clause has consecutive literals
- Each clause can share at most one literal with another clause

LSAT is known to be NP-hard [11]

[11] Arkin, E.M., Banik, A., Carmi, P., Citovsky, G., Katz, M.J., Mitchell, J.S.B., Simakov, M.: Choice is hard

Hardness for 3-crossing profiles

Reduction | Construction

- Let ϕ be the LSAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_n$

Hardness for 3-crossing profiles
Reduction | Construction

- Let ϕ be the LSAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_n$
- Let σ be the LSAT ordering of the literals

Hardness for 3-crossing profiles

Reduction | Construction

- Let ϕ be the LSAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_n$
- Let σ be the LSAT ordering of the literals
- For each variable x_i introduce candidates p_i and q_i corresponding to x_i and $\overline{x_i}$

Hardness for 3-crossing profiles

Reduction | Construction

- Let ϕ be the LSAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_n$
- Let σ be the LSAT ordering of the literals
- For each variable x_i introduce candidates p_i and q_i corresponding to x_i and $\overline{x_i}$
- Also introduce $(n+1)$ dummy candidates for each variable

Hardness for 3-crossing profiles

Reduction | Construction

- Let ϕ be the LSAT instance with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_n$
- Let σ be the LSAT ordering of the literals
- For each variable x_i introduce candidates p_i and q_i corresponding to x_i and $\overline{x_i}$
- Also introduce $(n+1)$ dummy candidates for each variable
- $d[i, j]$ denotes j^{th} dummy candidate for variable x_i

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: x_1 , $\overline{x_2}$, x_3 , $\overline{x_1}$, x_4 , x_2 , $\overline{x_4}$, $\overline{x_3}$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1 \, , \, \overline{x_2} \, , \, x_3 \, , \, \overline{x_1} \, , \, x_4 \, , \, x_2 \, , \, \overline{x_4} \, , \, \overline{x_3}$

$$\underbrace{(\, x_1 \wedge \overline{x_2} \wedge x_3 \,)}_{C_1} \vee \underbrace{(\, x_3 \wedge \overline{x_1} \wedge x_4 \,)}_{C_2} \vee \underbrace{(\, x_2 \wedge \overline{x_4} \wedge \overline{x_3} \,)}_{C_3}$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$$G_1 = p_1, q_2 \qquad G_2 = p_3, q_1, p_4 \qquad G_3 = p_2, q_4, q_3$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$G_1 = p_1, q_2$ $\quad$ $G_2 = p_3, q_1, p_4$ $\quad$ $G_3 = p_2, q_4, q_3$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$G_1 = p_1, q_2$ $\qquad$ $G_2 = p_3, q_1, p_4$ $\qquad$ $G_3 = p_2, q_4, q_3$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$$G_1 = p_1, q_2 \qquad G_2 = p_3, q_1, p_4 \qquad G_3 = p_2, q_4, q_3$$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

$$v_2 : G_2 \succ G_3 \succ G_1 \succ D$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$G_1 = p_1, q_2$ $\quad$ $G_2 = p_3, q_1, p_4$ $\quad$ $G_3 = p_2, q_4, q_3$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

$$v_2 : G_2 \succ G_3 \succ G_1 \succ D$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$G_1 = p_1, q_2$ $\qquad$ $G_2 = p_3, q_1, p_4$ $\qquad$ $G_3 = p_2, q_4, q_3$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

$$v_2 : G_2 \succ G_3 \succ G_1 \succ D$$

$$v_3 : G_3 \succ G_2 \succ G_1 \succ D$$

Hardness for 3-crossing profiles

Reduction | Example

LSAT ordering: $x_1, \overline{x_2}, x_3, \overline{x_1}, x_4, x_2, \overline{x_4}, \overline{x_3}$

Candidate ordering: $p_1, q_2, p_3, q_1, p_4, p_2, q_4, q_3, d[i,j]$

$$(x_1 \wedge \overline{x_2} \wedge x_3) \quad \vee \quad (x_3 \wedge \overline{x_1} \wedge x_4) \quad \vee \quad (x_2 \wedge \overline{x_4} \wedge \overline{x_3})$$

$G_1 = p_1, q_2$ $\qquad G_2 = p_3, q_1, p_4$ $\qquad G_3 = p_2, q_4, q_3$

$$v_1 : G_1 \succ G_2 \succ G_3 \succ D$$

$$v_2 : G_2 \succ G_3 \succ G_1 \succ D$$

$$v_3 : G_3 \succ G_2 \succ G_1 \succ D$$

$$v_i : G_i \; \dots \; G_m \succ G_{i-1} \; \dots \; G_1 \succ D$$

Hardness for 3-crossing profiles

Reduction | Construction

$$v_i : G_i \succ G_{i+1} \succ \cdots \succ G_m \succ G_{i-1} \succ \cdots \succ G_1 \succ D$$

Hardness for 3-crossing profiles

Reduction | Construction

$$v_i : G_i \succ G_{i+1} \succ \cdots \succ G_m \succ G_{i-1} \succ \cdots \succ G_1 \succ D$$

$$v_{i,j} : d[i,j] \succ p_i \succ q_i \succ \cdots \succ D \setminus d[i,j]$$

Hardness for 3-crossing profiles

Reduction | Construction

$$v_i : G_i \succ G_{i+1} \succ \cdots \succ G_m \succ G_{i-1} \succ \cdots \succ G_1 \succ D$$

$$v_{i,j} : d[i,j] \succ p_i \succ q_i \succ \cdots \succ D \setminus d[i,j]$$

A valid committee corresponds to a satisfying assignment when $R \leqslant 2$

Hardness for 3-crossing profiles

Reduction | Construction

$$v_i : G_i \succ G_{i+1} \succ \cdots \succ G_m \succ G_{i-1} \succ \cdots \succ G_1 \succ D$$

$$v_{i,j} : d[i,j] \succ p_i \succ q_i \succ \cdots \succ D \setminus d[i,j]$$

A valid committee corresponds to a satisfying assignment when $R \leqslant 2$

- Using dummy candidates we ensure that exactly one of p_i and q_i is in the committee

Hardness for 3-crossing profiles

Reduction | Construction

$$v_i : G_i \succ G_{i+1} \succ \cdots \succ G_m \succ G_{i-1} \succ \cdots \succ G_1 \succ D$$

$$v_{i,j} : d[i,j] \succ p_i \succ q_i \succ \cdots \succ D \setminus d[i,j]$$

A valid committee corresponds to a satisfying assignment when $R \leqslant 2$

- Using dummy candidates we ensure that exactly one of p_i and q_i is in the committee
- Careful case analysis shows that resultant profile is 3-crossing

Thank You !