Problems in Computational Social Choice on Restricted Domains

Chinmay Sonar

Advisor: Prof. Neeldhara Misra

July 1, 2019

Part I
Introduction

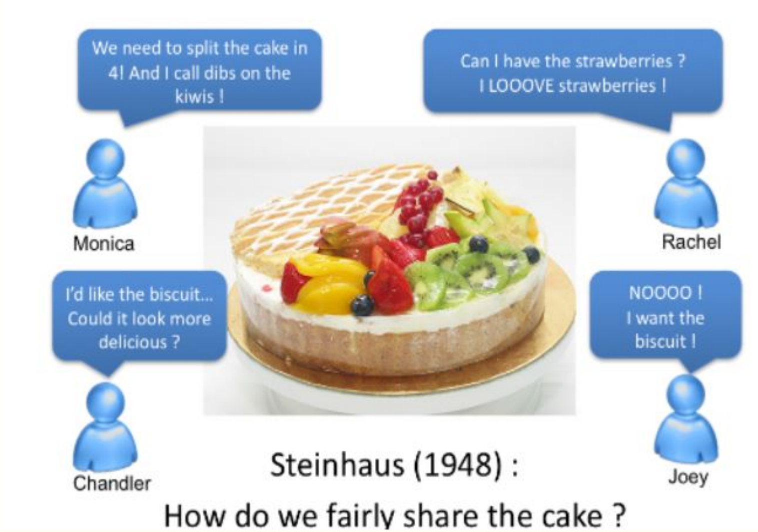
- What is Social Choice?
  $\rightarrow$ Formalized study of aggregating individual preferences towards collective decision making

- What is Social Choice?
  $\rightarrow$ Formalized study of aggregating individual preferences towards collective decision making
- Computational aspects
  $\rightarrow$ Boosted in 1980's. Aim is to find an efficient mechanism or show that it is unlikely to have such mechanism.

- What is Social Choice?
  $\rightarrow$ Formalized study of aggregating individual preferences towards collective decision making
- Computational aspects
  $\rightarrow$ Boosted in 1980's. Aim is to find an efficient mechanism or show that it is unlikely to have such mechanism.
- Three Pillars of COMSOC

Voting Matching Fair Division

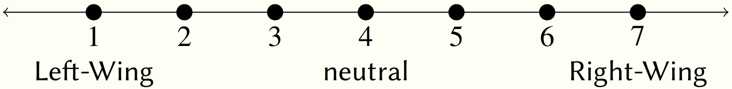
Example
Fair Division



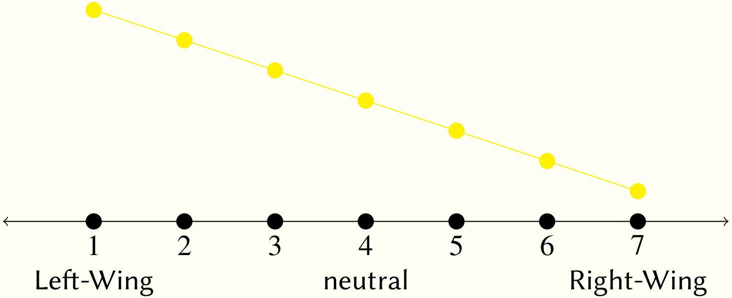
- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.

Restricted Domains
Motivation

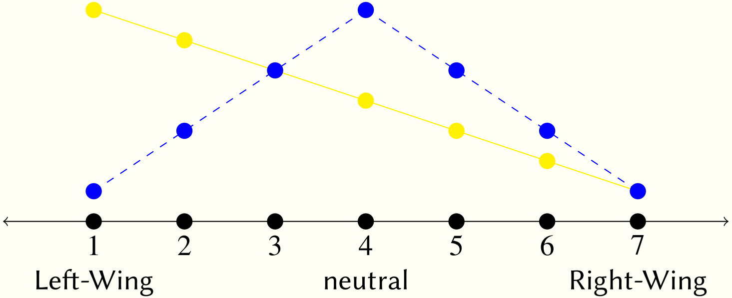
- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.
- Example of Elections in Western countries:



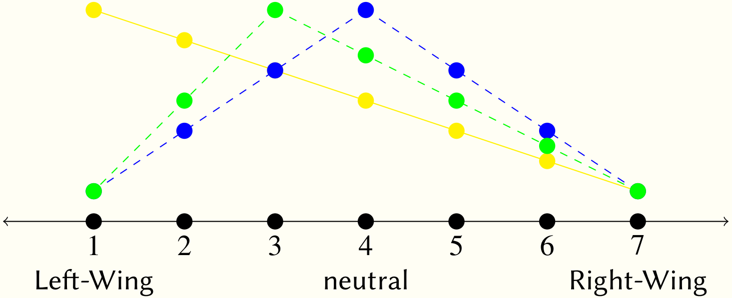
- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.
- Example of Elections in Western countries:



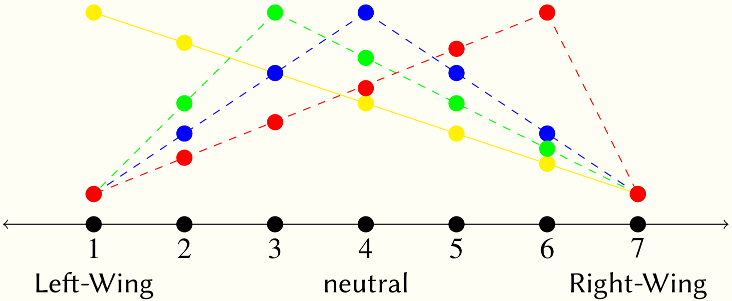
- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.
- Example of Elections in Western countries:



- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.
- Example of Elections in Western countries:



- It is ubiquitous to find some structure in the preferences as they are commonly derived from an underlying philosophy.
- Example of Elections in Western countries:



- Analogous notion of ordering over the voters

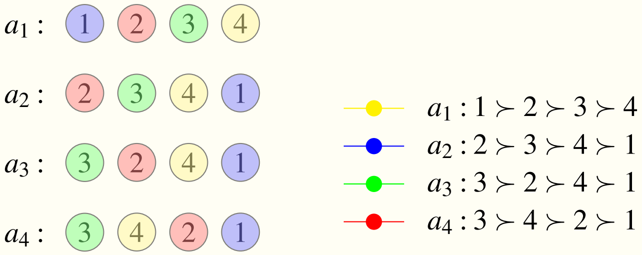


Figure: Single-Crossing Domain

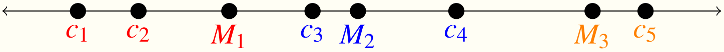


Figure: 1D-Euclidean Domain

1D-Euclidean Preferences

A profile $\mathcal{P}$ is said to be 1D-Euclidean, if there exist a mapping $X : V \rightarrow \mathbb{R}$ which maps every agent on a real line such that for any agent i, $x \succ_i y$ if and only if $|d(i,x)| < |d(i,y)|$.

Restricted Domains
Bounded Range

$$v_i := \quad \cdots \quad \succ c_i \succ \quad \ldots$$
$$v_j := \quad \cdots \quad \succ c_i \succ \quad \ldots$$

$\underbrace{\qquad\qquad\qquad\qquad}_{\text{range of } c_i}$

$$v_i := \quad \cdots \quad \succ c_i \succ \quad \ldots$$
$$v_j := \qquad \cdots \qquad \succ c_i \succ \quad \ldots$$

range of c_i

- **Range** (c_i)**)** is the difference between best and worst position of c_i in the given profile.
- Preference lists of top hundred students in JEE.

Known Results

- Bypass arrow's impossibility theorem by considering mechanisms on restricted domains
- Shield from Manipulation [1]
- Efficient mechanisms for problems hard to solve on general domains [2] [3]

[1] Herv ÌĄe Moulin. On strategy-proofness and single peakedness.Public Choice,pages 437âĂŞ455, 198

[2] Nadja Betzler, Arkadii Slinko, and Johannes Uhlmann. On the Computation of Fully Proportional Representation.J. Artif. Intell. Res., pages 475âĂŞ519,2013.

[3] Piotr Skowron, Lan Yu, Piotr Faliszewski, and Edith Elkind. The complexity of fully proportional representation for single-crossing electorates. Theor.Comput. Sci, pages 43âĂŞ57, 2015

Parameterized Complexity

- Input of a parameterized problem is denoted as
  $(I,k) \in \Sigma^* \times \mathbb{N}$
  where I is input of a classical language and k is the parameter

Parameterized Complexity

- Input of a parameterized problem is denoted as
  $(I, k) \in \Sigma^* \times \mathbb{N}$
  where I is input of a classical language and k is the parameter
- FPT $\rightarrow \exists$ algorithm that decides in time $\mathcal{O}(f(k)n^{\mathcal{O}(1)})$

Parameterized Complexity

- Input of a parameterized problem is denoted as
  $(I, k) \in \Sigma^* \times \mathbb{N}$
  where I is input of a classical language and k is the parameter
- FPT $\rightarrow \exists$ algorithm that decides in time $\mathcal{O}(f(k)n^{\mathcal{O}(1)})$
- XP $\rightarrow \exists$ algorithm that decides in time $\mathcal{O}(n^{f(k)})$

Multiwinner Elections

- Multiwinner elections are ubiquitous. Eg. Shortlisting candidates for a fixed number of fellowships, Movie selection on the airplane
- Fully proportional representation
- Popular Committee selection Rules – Chamberlin-Courant [4] and Monroe [5].

[4] John R Chamberlin and Paul N Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule.American Political Science Review, pages 718âĂŞ733, 1983

[5] Burt L Monroe. Fully proportional representation.American Political ScienceReview, pages 925âĂŞ940, 1995

Example:

Finding a collection of movies to include in Airplane

$n = 4, m = 5, k = 2$

Figure: Ordinal Preferences

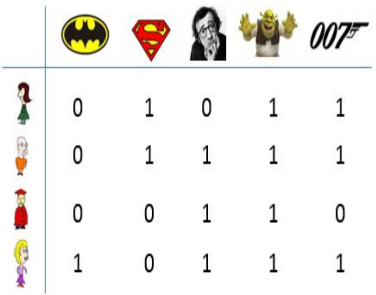


Figure: Dichotomous Preferences

Note: Figures are taken from [1][6]

[6] Piotr Faliszewski: rapc-session-3-committees.pptx

Idea of Chamberlin-Courant voting

- Every voter should find a representative within the determined committee.
- This would make every voter happy.

Known results

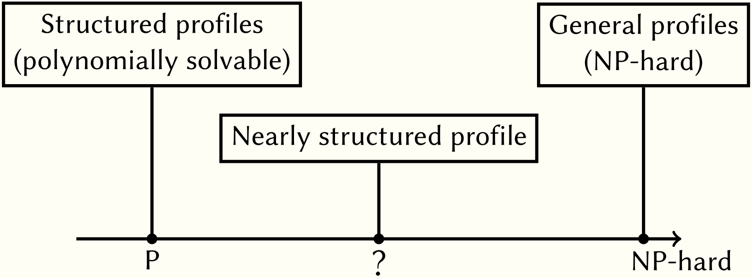


Figure: Chamberlin Courant Winner Determination

Notion of Almost Structured Domains

- There are many ways to generalize the restricted domains [7] [8]. We consider following two ways.

1. k-maverick domains [9]:
   - *k-candidate deletion*
   - *k-voter deletion*

---

[7] Salvador Barber'a and Bernardo Moreno. Top monotonicity: A common root for single peakedness, single crossing and the median voter result.Games and Economic Behavior, pages 345âĂŞ359, 2011

[8] Denis Cornaz, Lucie Galand, and Olivier Spanjaard. Bounded Single-Peaked Width and Proportional Representation. In Proceedings of the 20th Euro-pean Conference on Artificial Intelligence (ECAI)., Frontiers in Artificial Intelligence and Applications, pages 270âĂŞ275, 2012.

[9] Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger. Are there any nicely structured preference profiles nearby? Mathematical Social Sciences,pages 61âĂŞ73, 2016

k-candidate deletion
Example

a_1 : 1 2 3 4 5

a_2 : 2 3 1 4 5

a_3 : 3 2 4 1 5

a_4 : 5 1 3 4 2

a_5 : 5 1 2 3 4

Figure: Original Profile

k-candidate deletion
Example

a_1 : 1 2 3 4 5

a_2 : 2 3 1 4 5

a_3 : 3 2 4 1 5

a_4 : 5 1 3 4 2

a_5 : 5 1 2 3 4

Figure: Original Profile

C-Del Set:= 1 2

k-candidate deletion

Example

C-Del Set:= 1 2

a_1 : 1 2 3 4 5

a_2 : 2 3 1 4 5

a_3 : 3 2 4 1 5

a_4 : 5 1 3 4 2

a_5 : 5 1 2 3 4

Figure: Original Profile

a_1 : 3 4 5

a_2 : 3 4 5

a_3 : 3 4 5

a_4 : 5 3 4

a_5 : 5 3 4

Figure: Single-Crossing Profile

k-voter deletion

Example

a_1 : 1 2 3 4 5

a_2 : 3 2 1 4 5

a_3 : 3 2 4 5 1

a_4 : 5 4 3 2 1

a_5 : 5 1 2 3 4

Figure: Original Profile

k-voter deletion

Example

a_1 : 1 2 3 4 5

a_2 : 3 2 1 4 5

a_3 : 3 2 4 5 1

a_4 : 5 4 3 2 1

a_5 : 5 1 2 3 4

Figure: Original Profile

V-Del Set:= a_5

k-voter deletion
Example

V-Del Set:= a_5

a_1 : 1 2 3 4 5

a_2 : 3 2 1 4 5

a_3 : 3 2 4 5 1

a_4 : 5 4 3 2 1

a_5 : 5 1 2 3 4

Figure: Original Profile

a_1 : 1 2 3 4 5

a_2 : 3 2 1 4 5

a_3 : 3 2 4 5 1

a_4 : 5 4 3 2 1

Figure: Single-Crossing Profile

Notion of Almost Structured Domains

- There are many ways to generalize the restricted domains [10] [11]. We consider following two ways.

1. k-maverick domains [12]:
   - *k-candidate deletion*
   - *k-voter deletion*
2. Generalizations of Single-Peaked and Single-Crossing:
   - *k-composite single-peaked domains*
   - *k-crossing domains*

---

[10] Salvador Barber'a and Bernardo Moreno. Top monotonicity: A common root for single peakedness, single crossing and the median voter result.Games and Economic Behavior, pages 345âĂŞ359, 2011

[11] Denis Cornaz, Lucie Galand, and Olivier Spanjaard. Bounded Single-Peaked Width and Proportional Representation. In Proceedings of the 20th Euro-pean Conference on Artificial Intelligence (ECAI)., Frontiers in Artificial Intelligence and Applications, pages 270âĂŞ275, 2012.

[12] Robert Bredereck, Jiehua Chen, and Gerhard J. Woeginger. Are there any nicely structured preference profiles nearby? Mathematical Social Sciences,pages 61âĂŞ73, 2016

Summary of Results

	SP CC	SP MM[13] CC	SC CC	SC MM CC
Struct[14]	$\mathcal{O}(nm^2)$	$\mathcal{O}(nm)$	$\mathcal{O}(n^2mk)$	$\mathcal{O}(n^2mk)$
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	$2^{k}\mathcal{O}(nm^2)$	$2^{k}\mathcal{O}(nm)$	$2^{k}\mathcal{O}(n^2mk)$	$2^{k}\mathcal{O}(n^2mk)$

Table: Summary of algorithms for CC-Winner Determination

- NP-hardness results:
  1. 3-composite single-peaked domains
  2. 3-crossing domains

[13] MM: Minimax (egalitarian version)

[14] Denotes perfectly structured profiles

Hardness for 3-crossing Domains

Theorem

Computing an ℓ_∞ -CC committee on 3-crossing profiles with $R \leqslant 2$ is NP-complete.

Proof

Robustness Radius for Chamberlin-Courant Rule

Robustness Radius

- **Definition:** For multiwinner voting rule $\mathcal{R}$, and input $E = (C,V)$, a committee k and an integer r, we ask if it is possible to obtain an election E' by
  - making *at most r swaps of adjacent candidates* within rankings of E, or by
  - changing at most r bits for the case of approval ballots,

  s.t. $\mathcal{R}(E,k) \neq \mathcal{R}(E',k)$.

Robustness Radius

- **Definition:** For multiwinner voting rule $\mathcal{R}$, and input $E = (C,V)$, a committee k and an integer r, we ask if it is possible to obtain an election E' by
  - making *at most r swaps of adjacent candidates* within rankings of E, or by
  - changing at most r bits for the case of approval ballots,

  s.t. $\mathcal{R}(E,k) \neq \mathcal{R}(E',k)$.
  - YES $\rightarrow RR \leqslant r$
  - NO $\rightarrow RR > r$

Related Work

- 'Robustness Among Multiwinner Voting Rules' [SAGT'18] [15] – First defined the concept of **RR**

[15] Bredereck, R., Faliszewski, P., Kaczmarczyk, A., Niedermeier, R., Skowron, P., & Talmon, N. (2017, September). Robustness among multiwinner voting rules. In International Symposium on Algorithmic Game Theory (pp. 80-92). Springer, Cham.

Related Work

- 'Robustness Among Multiwinner Voting Rules' [SAGT'18] [15] – First defined the concept of **RR**
- The paper considers the problem of **RR** for many voting rules (SNTV, *k*-Bloc, Copeland, NED, STV and CC) and shows many polynomial time results, but shows the hardness for CC.

[15] Bredereck, R., Faliszewski, P., Kaczmarczyk, A., Niedermeier, R., Skowron, P., & Talmon, N. (2017, September). Robustness among multiwinner voting rules. In International Symposium on Algorithmic Game Theory (pp. 80-92). Springer, Cham.

Related Work

- 'Robustness Among Multiwinner Voting Rules' [SAGT'18] [15] – First defined the concept of **RR**
- The paper considers the problem of **RR** for many voting rules (SNTV, *k*-Bloc, Copeland, NED, STV and CC) and shows many polynomial time results, but shows the hardness for CC.
- We consider the exact algorithms for hard instances and ask the question on restricted domain for **CC** rule

[15] Bredereck, R., Faliszewski, P., Kaczmarczyk, A., Niedermeier, R., Skowron, P., & Talmon, N. (2017, September). Robustness among multiwinner voting rules. In International Symposium on Algorithmic Game Theory (pp. 80-92). Springer, Cham.

Similar Notions

- **RR** captures the impact of small changes in the input preferences on the set of the winning committees for given voting rule.

[16] Elkind, E., Faliszewski, P., & Slinko, A. (2009, October). Swap bribery. In International Symposium on Algorithmic Game Theory (pp. 299-310). Springer, Berlin, Heidelberg

Similar Notions

- **RR** captures the impact of small changes in the input preferences on the set of the winning committees for given voting rule.
- **MoV** (Margin of Victory) – It measures the number of voters to be changed rather than the number of swaps. Hence *MoV* is more powerful model than *RR*.

[16] Elkind, E., Faliszewski, P., & Slinko, A. (2009, October). Swap bribery. In International Symposium on Algorithmic Game Theory (pp. 299-310). Springer, Berlin, Heidelberg

Similar Notions

- **RR** captures the impact of small changes in the input preferences on the set of the winning committees for given voting rule.
- **MoV** (Margin of Victory) – It measures the number of voters to be changed rather than the number of swaps. Hence *MoV* is more powerful model than *RR*.
- **Swap Bribery** – Also cares about the outcome after the change in profile [16].

[16] Elkind, E., Faliszewski, P., & Slinko, A. (2009, October). Swap bribery. In International Symposium on Algorithmic Game Theory (pp. 299-310). Springer, Berlin, Heidelberg

Results Summary

Results Summary

Note: All results are parameterized by parameter k, the size of the committee.

- For general profiles:
  1. **RR** is **W[2]**-hard for *Approval Chamberlin-Courant.*
  2. **XP** algorithm for determining **RR** for *complete rankings/ approval ballots.*

Results Summary

Note: All results are parameterized by parameter k, the size of the committee.

- For general profiles:
  1. **RR** is **W[2]**-hard for *Approval Chamberlin-Courant.*
  2. **XP** algorithm for determining **RR** for *complete rankings/ approval ballots.*
- On nearly restricted domains:
  1. **RR** is **NP**-hard for $\ell_1 - CC$ even for *6-crossing* domains.
  2. **RR** is **NP**-hard for $\ell_\infty - CC$ even for *4-crossing* domains.
  3. **RR** is **NP**-hard for $\ell_\infty - CC$ even for *4-composite SP* domains.

W[2]-Hardness for Approval CC

Theorem

Computing Robustness Radius for of Chamberlin-Courant is $\mathrm{W}[2]-hard$ even for Dichotomous Preferences.

Proof

Part III
Matching

1 Stable Matching Problems on Restricted Domains

Stable Matching Problems for Restricted domains

- Aim was to study classical problems in stable matching on restricted domains.

Stable Matching Problems for Restricted domains

- Aim was to study classical problems in stable matching on restricted domains.
- Classical bipartite marriage setting vs matching on general graphs (Stable Roommates)
- Stable Matching with ties and incomplete preferences (SMTI)

Stable Matching Problems for Restricted domains

- Aim was to study classical problems in stable matching on restricted domains.
- Classical bipartite marriage setting vs matching on general graphs (Stable Roommates)
- Stable Matching with ties and incomplete preferences (SMTI)
- *Hard* variants of the matching problem:
  1. Maximum sized SMTI
  2. Egalitarian SMTI
  3. Minimum-regret SMTI
  4. Sex-equal Stable Matching (SESM)

Stable Matching Problems for Restricted domains

- Aim was to study classical problems in stable matching on restricted domains.
- Classical bipartite marriage setting vs matching on general graphs (Stable Roommates)
- Stable Matching with ties and incomplete preferences (SMTI)
- *Hard* variants of the matching problem:
  1. Maximum sized SMTI
  2. Egalitarian SMTI
  3. Minimum-regret SMTI
  4. Sex-equal Stable Matching (SESM)
- We study these problems for structured preference domains with "degree of incompleteness" as a parameter

Degree of Incompleteness

Example

$$a_1 := a_2 \sim a_3 \sim a_4 \succ a_5$$

$$a_2 := a_3 \sim a_4 \succ a_1 \succ a_5$$

$$a_3 := a_2 \sim a_4 \succ a_1 \sim a_5$$

$$a_4 := a_1 \succ a_2 \succ a_3 \succ a_5$$

$$a_5 := a_4 \succ a_1 \succ a_2 \sim a_3$$

Degree of Incompleteness
Example

$$a_1 := a_2 \sim a_3 \sim a_4 \succ a_5$$
$$a_2 := a_3 \sim a_4 \succ a_1 \succ a_5$$
$$a_3 := a_2 \sim a_4 \succ a_1 \sim a_5$$
$$a_4 := a_1 \succ a_2 \succ a_3 \succ a_5$$
$$a_5 := a_4 \succ a_1 \succ a_2 \sim a_3$$

- $\kappa_1 = 5$ (number of ties)
- $\kappa_2 = 3$ (maximum length of tie)
- $\kappa_3 = 7$ (total length of ties)

Summary of results
For restricted domains

- Parameterized hardness with parameter κ_1 (number of ties in the instance) for SP-SC domains:
  1. SESM (para-NP hard)
  2. Max-sized SMTI (W[1]-hard)
  3. Egalitarian SMTI (W[1]-hard)

Summary of results
For restricted domains

- Parameterized hardness with parameter κ_1 (number of ties in the instance) for SP-SC domains:
  1. SESM (para-NP hard)
  2. Max-sized SMTI (W[1]-hard)
  3. Egalitarian SMTI (W[1]-hard)
- Existence of a stable matching in SRTI is NP-complete even when κ_2 is constant

Manipulation in Stable Marriages

- **Manipulation:**
  As opposed to general profiles [17], it is not always possible to obtain an optimal manipulation inconspicuously or otherwise when we restrict the preferences to structured domains

[17] Vaish, R., Garg, D. (2017, August). Manipulating gale-shapley algorithm: preserving stability and remaining inconspicuous. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17 (pp. 437-443)

Manipulation on SP-SC Domains
Example

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4 : 1 \succ 2 \succ 3 \succ 4 \succ 5$	$w_4 : 5 \succ 4 \succ \underline{3} \succ 2 \succ 1$
$m_1 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_5 : 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$m_5 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_2 : 4 \succ 3 \succ 2 \succ \underline{5} \succ 1$
$m_2 : 3 \succ 2 \succ 4 \succ 1 \succ 5$	$w_1 : 3 \succ \underline{4} \succ 2 \succ 5 \succ 1$
$m_3 : 3 \succ 4 \succ 2 \succ 1 \succ 5$	$w_3 : \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

Manipulation on SP-SC Domains
Example

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4: 1 \succ 2 \succ 3 \succ 4 \succ 5$	$w_4: 5 \succ 4 \succ \underline{3} \succ 2 \succ 1$
$m_1: 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_5: 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$m_5: 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_2: 4 \succ 3 \succ 2 \succ \underline{5} \succ 1$
$m_2: 3 \succ 2 \succ 4 \succ 1 \succ 5$	$w_1: 3 \succ \underline{4} \succ 2 \succ 5 \succ 1$
$m_3: 3 \succ 4 \succ 2 \succ 1 \succ 5$	$w_3: \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

Matching in original profile:
$(w_1, m_4), (w_2, m_5), (w_3, m_2), (w_4, m_3), (w_5, m_1)$

Manipulation on SP-SC Domains
Example

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4 : 1 \succ 2 \succ 3 \succ 4 \succ 5$
$m_1 : 2 \succ 1 \succ 3 \succ 4 \succ 5$
$m_5 : 2 \succ 1 \succ 3 \succ 4 \succ 5$
$m_2 : 3 \succ 2 \succ 4 \succ 1 \succ 5$
$m_3 : 3 \succ 4 \succ 2 \succ 1 \succ 5$

$w_4 : \underline{5} \succ 4 \succ 3 \succ 2 \succ 1$
$w_5 : 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$w_2 : \underline{4} \succ 3 \succ 2 \succ 5 \succ 1$
$w_1 : \underline{3} \succ 2 \succ 1 \succ 4 \succ 5$
$w_3 : \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

$w_1(\mathit{True})$: $w_1 : 3 \succ 4 \succ 2 \succ 1 \succ 5$
Matching with non-inconspicuous manipulation:
$(w_1, m_3), (w_2, m_4), (w_3, m_2), (w_4, m_5), (w_5, m_1)$

Inconspicuous is not enough

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4 : 1 \succ 2 \succ 3 \succ 4 \succ 5$	$w_4 : 5 \succ 4 \succ \underline{3} \succ 2 \succ 1$
$m_1 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_5 : 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$m_5 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_2 : 4 \succ 3 \succ 2 \succ \underline{5} \succ 1$
$m_2 : 3 \succ 2 \succ 4 \succ 1 \succ 5$	$w_1 : 3 \succ \underline{4} \succ 2 \succ 5 \succ 1$
$m_3 : 3 \succ 4 \succ 2 \succ 1 \succ 5$	$w_3 : \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

- There does not exist any lucrative inconspicuous manipulation in this case

Inconspicuous is not enough

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4 : 1 \succ 2 \succ 3 \succ 4 \succ 5$	$w_4 : 5 \succ 4 \succ \underline{3} \succ 2 \succ 1$
$m_1 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_5 : 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$m_5 : 2 \succ 1 \succ 3 \succ 4 \succ 5$	$w_2 : 4 \succ 3 \succ 2 \succ \underline{5} \succ 1$
$m_2 : 3 \succ 2 \succ 4 \succ 1 \succ 5$	$w_1 : 3 \succ \underline{4} \succ 2 \succ 5 \succ 1$
$m_3 : 3 \succ 4 \succ 2 \succ 1 \succ 5$	$w_3 : \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

- There does not exist any lucrative inconspicuous manipulation in this case
- Matching: $(w_1, m_4), (w_2, m_5), (w_3, m_2), (w_4, m_3), (w_5, m_1)$

Inconspicuous is not enough

SP axis: $m_1, m_2, m_3, m_4, m_5, w_1, w_2, w_3, w_4, w_5$

$m_4 : 1 \succ 2 \succ 3 \succ 4 \succ 5$
$m_1 : 2 \succ 1 \succ 3 \succ 4 \succ 5$
$m_5 : 2 \succ 1 \succ 3 \succ 4 \succ 5$
$m_2 : 3 \succ 2 \succ 4 \succ 1 \succ 5$
$m_3 : 3 \succ 4 \succ 2 \succ 1 \succ 5$

$w_4 : 5 \succ 4 \succ \underline{3} \succ 2 \succ 1$
$w_5 : 5 \succ 4 \succ 3 \succ 2 \succ \underline{1}$
$w_2 : 4 \succ 3 \succ 2 \succ \underline{5} \succ 1$
$w_1 : 3 \succ \underline{4} \succ 2 \succ 5 \succ 1$
$w_3 : \underline{2} \succ 3 \succ 1 \succ 4 \succ 5$

- There does not exist any lucrative inconspicuous manipulation in this case
- Matching: $(w_1, m_4), (w_2, m_5), (w_3, m_2), (w_4, m_3), (w_5, m_1)$
- We strengthen our observation by showing similar example even when we relax inconspicuousness.

Manipulation in Stable Marriages

- **Manipulation:**
  As opposed to general profiles [18], it is not always possible to obtain an optimal manipulation inconspicuously or otherwise when we restrict the preferences to structured domains
- **Unique SM:** 1D-Euclidean preferences admit a unique stable matching

[18] Vaish, R., Garg, D. (2017, August). Manipulating gale-shapley algorithm: preserving stability and remaining inconspicuous. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17 (pp. 437-443)

Problem Setup

- We are given a set of $[n] = \{1, 2, \ldots, n\}$ agents, and an undirected graph $G := (V, E)$ such that each $v \in V$ corresponds to a good. Let $|V| = m, \mathcal{C}(V) \subseteq 2^V$.

Problem Setup

- We are given a set of $[n] = \{1, 2, \ldots, n\}$ agents, and an undirected graph $G := (V, E)$ such that each $v \in V$ corresponds to a good. Let $|V| = m, \mathcal{C}(V) \subseteq 2^V$.
- Allocation $\mathcal{A} : [n] \rightarrow \mathcal{C}(V)$ is connected if $\mathcal{A}(i) \in \mathcal{C}(V)$ forms a *connected subgraph* of G.

Problem Setup

- We are given a set of $[n] = \{1, 2, \ldots, n\}$ agents, and an undirected graph $G := (V, E)$ such that each $v \in V$ corresponds to a good. Let $|V| = m, \mathcal{C}(V) \subseteq 2^V$.
- Allocation $\mathcal{A} : [n] \to \mathcal{C}(V)$ is connected if $\mathcal{A}(i) \in \mathcal{C}(V)$ forms a *connected subgraph* of G.
- Preferences of agent $i \in [n]$ are specified with valuation function $u_i : \mathcal{C}(V) \to \mathbb{N} \cup \{0\}$.

Problem Setup

- We are given a set of $[n] = \{1, 2, \ldots, n\}$ agents, and an undirected graph $G := (V, E)$ such that each $v \in V$ corresponds to a good. Let $|V| = m, \mathcal{C}(V) \subseteq 2^V$.
- Allocation $\mathcal{A} : [n] \rightarrow \mathcal{C}(V)$ is connected if $\mathcal{A}(i) \in \mathcal{C}(V)$ forms a *connected subgraph* of G.
- Preferences of agent $i \in [n]$ are specified with valuation function $u_i : \mathcal{C}(V) \rightarrow \mathbb{N} \cup \{0\}$.
- We consider binary valuations – For every agent $i \in [n]$ and $v \in V$, $u_i(v) \in \{0, 1\}$.

- Sharing an access to high performance computer among researchers

- Sharing an access to high performance computer among researchers
- Dividing a coast between countries

- Sharing an access to high performance computer among researchers
- Dividing a coast between countries
- Cake Division

Fainess Notions

- **Envyfreeness**: An allocation $\mathcal{A} := \{A_1, A_2, \ldots, A_n\}$ is *envyfree* if for all pair of agents $i, j \in [n]$, $u_i(A_i) \geqslant u_i(A_j)$. For indivisible goods we relax this to EF1 which demands $u_i(A_i) \geqslant u_j(A_j \setminus u)$ where u is a good from A_j.

Fainess Notions

- **Envyfreeness**: An allocation $\mathcal{A} := \{A_1, A_2, \ldots, A_n\}$ is *envyfree* if for all pair of agents $i, j \in [n]$, $u_i(A_i) \geqslant u_i(A_j)$. For indivisible goods we relax this to EF1 which demands $u_i(A_i) \geqslant u_j(A_j \setminus u)$ where u is a good from A_j.
- **Equitability**: An allocation $\mathcal{A} := \{A_1, A_2, \ldots, A_n\}$ is *equitable* if for all pair of agents $i, j \in [n]$, $u_i(A_i) \geqslant u_j(A_j)$. The notion of EQ1 is defined analogously.

Efficiency Notions

- **Completeness**: All goods should be allocated

Efficiency Notions

- **Completeness**: All goods should be allocated
- **Pareto Optimality (PO)**: An allocation $\mathcal{A}$ is PO if there does not exist $\mathcal{A}'$ which gives every agent utility at least as much as $\mathcal{A}$ and improves utility for at least one agent.

Efficiency Notions

- **Completeness**: All goods should be allocated
- **Pareto Optimality (PO)**: An allocation $\mathcal{A}$ is PO if there does not exist $\mathcal{A}'$ which gives every agent utility at least as much as $\mathcal{A}$ and improves utility for at least one agent.
- **Non-Wastefulness**: Agents are allocated *only* the goods which they approve

Connected Envyfree Allocations

Summary of results

Restrictions ↓	Existence			Computation		
	EF1 + Comp	EF1 + PO	EF1 + NW	EF1 + Comp	EF1 + PO	EF1 + NW
Binary	Yes	No	No	?	NP-hard	NP-hard
Binary interval	Yes	No	No	?	?	?
Binary k-interval	Yes	Yes	Yes	P	P	P
Binary extremal	Yes	Yes	Yes	P	P	P
Binary left-extremal	Yes	Yes	Yes	P	P	P

Table: Summary of results for Connected Fair Division EF1

Connected Equitable Allocations
Summary of results

Restrictions ↓	Existence			Computation		
	EQ1 + Comp	EQ1 + PO	EQ1 + NW	EQ1 + Comp	EQ1 + PO	EF1 + NW
Binary	Yes	No	No	P	NP-hard	NP-hard
Binary interval	Yes	No	No	P	?	P
Binary k-interval	Yes	No	No	P	P	P
Binary extremal	Yes	No	No	P	P	P
Binary left-extremal	Yes	No	No	P	P	P

Table: Summary of results for CONNECTED FAIR DIVISION EQ1

.

- Results table for *EQ1*

Conclusion

- Restricted domains capture many real-life scenarios, hence, it is useful to study them.

Conclusion

- Restricted domains capture many real-life scenarios, hence, it is useful to study them.
- Exploiting the structure of the domain is necessary to obtain efficient mechanisms.

Conclusion

- Restricted domains capture many real-life scenarios, hence, it is useful to study them.
- Exploiting the structure of the domain is necessary to obtain efficient mechanisms.
- Restricted domains also provide a workaround from axiomatic results in social choice.

Conclusion

- Restricted domains capture many real-life scenarios, hence, it is useful to study them.
- Exploiting the structure of the domain is necessary to obtain efficient mechanisms.
- Restricted domains also provide a workaround from axiomatic results in social choice.
- In stable marriages, manipulation seems harder on restricted domains as opposed to the general domains.

Thank You !

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 $R \leqslant 2$

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

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 [19]

[19] 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 [19]

[19] 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 [19]

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

- 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

- 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 $\overline{x_i}$

- 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

- 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Lemma

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

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

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

Lemma

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

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

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

Lemma

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

Theorem

Checking if **RR=1** for Approval Chamberlin Courant is **W[2]** hard parameterized by size of committee (k).

W[2]-hardness for RR for Approval CC
Construction

Theorem

Checking if **RR=1** for Approval Chamberlin Courant is **W[2]** hard parameterized by size of committee (k).

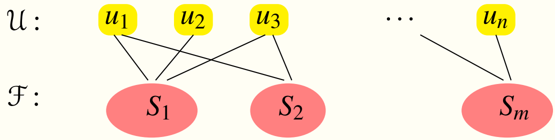
- Reduction Hitting Set instance.

Hitting Set Instance:

$\mathcal{U}$: u_1 u_2 u_3 $\cdots$ u_n

Hitting Set
Definition

Hitting Set Instance:



$\forall S_i \in \mathcal{F}; S_i \subseteq \mathcal{U}$

Given k, does there exist $S \subseteq U$ s.t. $|S \cap S_i| \neq \phi$ & $|S| \leqslant k$

Hitting Set
Example

Hitting Set Instance:

$\mathcal{U} = \{1, 2, 3, 4\}$

$\mathcal{F} = \{\{1, 2, 3\}, \{1, 2\}, \{3, 4\}, \{2, 4\}, \{1, 4\}\}$

$k = 1$

NO instance

Hitting Set Instance:

$\mathcal{U} = \{1, 2, 3, 4\}$

$\mathcal{F} = \{\{1, 2, 3\}, \{1, 2\}, \{3, 4\}, \{2, 4\}, \{1, 4\}\}$

$k = 2$

$S = \{1, 2\}$

Hitting Set Instance:

$\mathcal{U} = \{1,2,3,4\}$

$\mathcal{F} = \{\{1,2,3\},\{1,2\},\{3,4\},\{2,4\},\{1,4\}\}$

$|\mathcal{U}| = n$

$k = 2$ (size of the hitting set)

Construction
General Case

Hitting Set Instance:

$\mathcal{U} = \{1,2,3,4\}$

$\mathcal{F} = \{\{1,2,3\},\{1,2\},\{3,4\},\{2,4\},\{1,4\}\}$

$|\mathcal{U}| = n$

$k = 2$ (size of the hitting set)

RR Instance for Approval Voting Rule:

$k' = k$ (Committee size)

$$\mathcal{A} := \underbrace{\{c_1, c_2, \ldots, c_n\}}_{\mathcal{U}} \cup \underbrace{\{d_1, d_2, \ldots, d_k\}}_{\mathcal{D}}$$

Construction

Hitting Set Instance:
$\mathcal{U} = \{1,2,3,4\}$
$\mathcal{F} = \{\{1,2,3\},\{1,2\},\{3,4\},\{2,4\},\{1,4\}\}$
$|\mathcal{U}| = n = 4$
$k = 2$ (size of the hitting set)

RR Instance for Approval Voting Rule:
$k' = k = 2$ (Committee size)

$$\mathcal{A} := \underbrace{\{c_1, c_2, c_3, c_4\}}_{\mathcal{U}} \cup \underbrace{\{d_1, d_2\}}_{\mathcal{D}}$$

Construction

Hitting Set Instance:

$\mathcal{U} = \{1, 2, 3, 4\}$

$\mathcal{F} = \{\{1,2,3\}, \{1,2\}, \{3,4\}, \{2,4\}, \{1,4\}\}$

$k = 2$ (size of the hitting set)

RR Instance for Approval Voting Rule:

$k' = k = 2$ (Committee size)

$\mathcal{A} := \{c_1, c_2, c_3, c_4\} \cup \{d_1, d_2\}$

Voting profile:

		c_1	c_2	c_3	c_4	d_1	d_2
v_1	:	1	1	1	0	1	0
v_2	:	1	1	0	0	1	0
v_3	:	0	0	1	1	1	0
v_4	:	0	1	0	1	1	0
v_5	:	1	0	0	1	1	0

Construction

Hitting Set Instance:
$\mathcal{U} = \{1,2,3,4\}$
$\mathcal{F} = \{\{1,2,3\},\{1,2\},\{3,4\},\{2,4\},\{1,4\}\}$
$k = 2$ (size of the hitting set)

RR Instance for Approval Voting Rule:
$k' = k = 2$ (Committee size)
$\mathcal{A} := \{c_1, c_2, c_3, c_4\} \cup \{d_1, d_2\}$
Voting profile:

		c_1	c_2	c_3	c_4	d_1	d_2
v_1	:	1	1	1	0	1	0
v_2	:	1	1	0	0	1	0
v_3	:	0	0	1	1	1	0
v_4	:	0	1	0	1	1	0
v_5	:	1	0	0	1	1	0

		c_1	c_2	c_3	c_4	d_1	d_2
v_1	:	1	1	1	0	0	1
v_2	:	1	1	0	0	0	1
v_3	:	0	0	1	1	0	1
v_4	:	0	1	0	1	0	1
v_5	:	1	0	0	1	0	1

Equivalence of two instances:

- **Forward Direction:**
  - 'YES' instance of **HS**
    $\Longrightarrow$ at least two 2-sized winning committees
    (one by candidates corresponding to Hitting Set of size $\leqslant k$;
    the other one is a trivial committee by dummy candidates)

Equivalence of two instances:

- **Forward Direction:**
  - 'YES' instance of **HS**
    $\implies$ at least two 2-sized winning committees
    (one by candidates corresponding to Hitting Set of size $\leqslant k$;
    the other one is a trivial committee by dummy candidates)
  - Both the committees have dissatisfaction score of 0,
    hence they are optimal.

Equivalence of two instances:

- **Forward Direction:**
  - 'YES' instance of **HS**
    $\implies$ at least two 2-sized winning committees
    (one by candidates corresponding to Hitting Set of size $\leqslant k$; the other one is a trivial committee by dummy candidates)
  - Both the committees have dissatisfaction score of 0, hence they are optimal.
  - We make entry $(v_1, d_1) = 0$ to knock off committee $\{d_1, d_2\}$ from the winning set since the misrepresentation score for this committee is now 1.

Equivalence of two instances:

- **Forward Direction:**
  - 'YES' instance of **HS**
    $\implies$ at least two 2-sized winning committees
    (one by candidates corresponding to Hitting Set of size $\leqslant k$;
    the other one is a trivial committee by dummy candidates)
  - Both the committees have dissatisfaction score of 0,
    hence they are optimal.
  - We make entry $(v_1, d_1) = 0$ to knock off committee $\{d_1, d_2\}$ from the winning set since the misrepresentation score for this committee is now 1.
  - Hence RR=1.

Equivalence of two instances: Reverse Direction

If there is more than one winning committee in the constructed election, then there exists a hitting set of size at most k.

If RR = 1 for the constructed election, then there are at least two winning committees of size k.

$$\Downarrow$$

If RR = 1 for the constructed election, the instance of HS on which the election is based is a YES-instance.

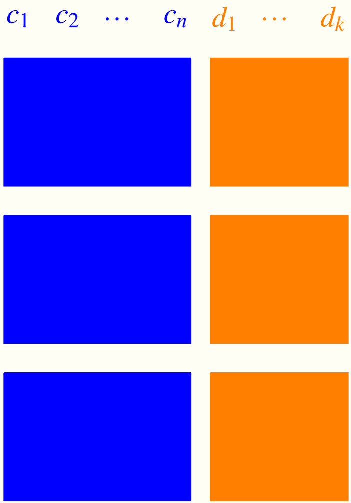
Equivalence of two instances: Reverse Direction

If there is exactly one winning committee in the constructed election, then any other committee has a dissatisfaction score of at least two.

$$\Downarrow$$

If RR = 1 for the constructed election, the instance of HS on which the election is based is a YES-instance.

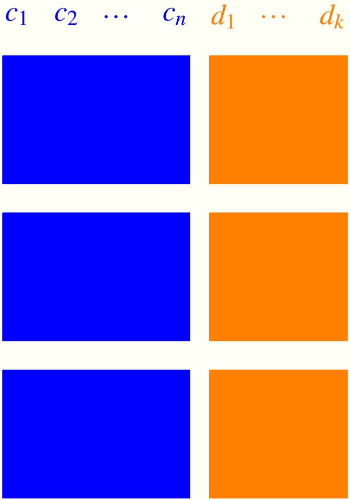
Case I
More than one winning committee



Let W be another winning committee different from D.

Case I

More than one winning committee



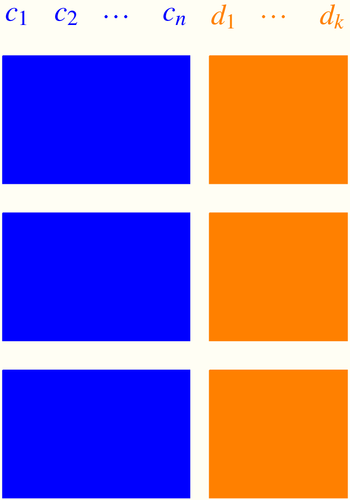
Let W be another winning committee different from D.

$Score(W) = Score(D)$

i.e. every voter has a representative in W

Case I

More than one winning committee

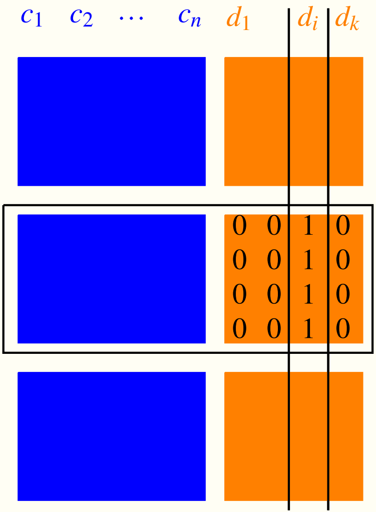


Let W be another winning committee different from D.

$Score(W) = Score(D)$
i.e. every voter has a representative in W

W omits some candidate from D, say d_i.

Case I
More than one winning committee



Let W be another winning committee different from D.

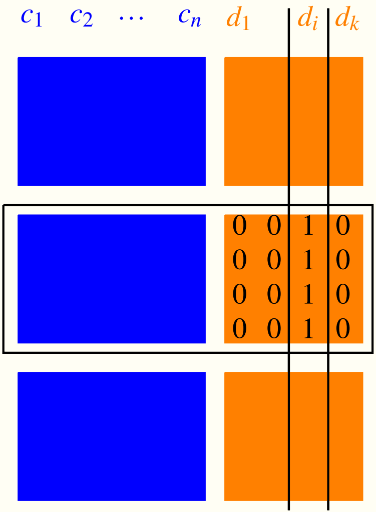
$Score(W) = Score(D)$
i.e. every voter has a representative in W

W omits some candidate from D, say d_i.

Consider block corresponding to d_i.

Case I

More than one winning committee



Let W be another winning committee different from D.

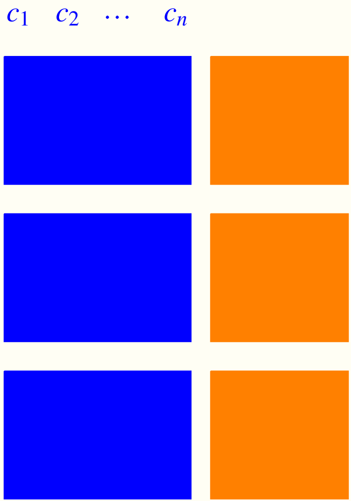
$Score(W) = Score(D)$
i.e. every voter has a representative in W

W omits some candidate from D, say d_i.

Consider block corresponding to d_i.

Here, every voter (set) is represented by a non-dummy candidate.
Hence, $W \setminus D$ corresponds to a hitting set.

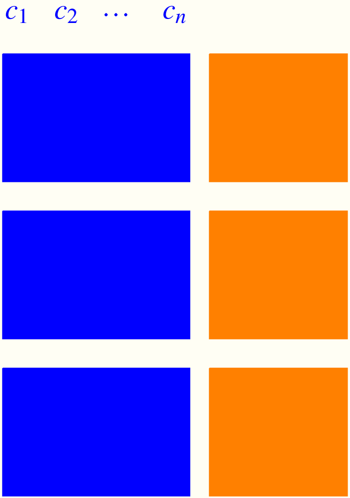
Case II
Unique winning committee



Suppose D is the only committee that represents every voter.

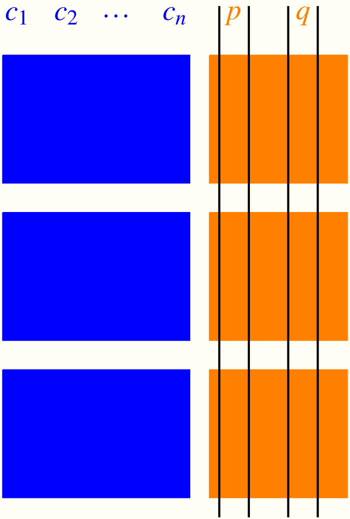
Case II

Unique winning committee



Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

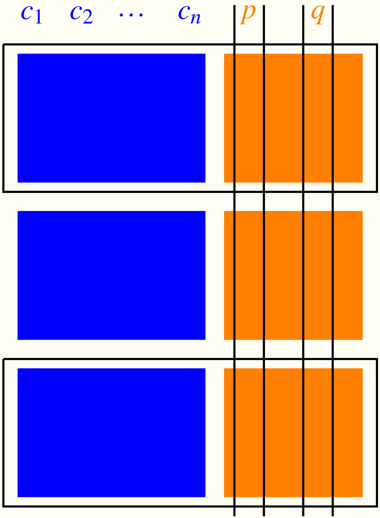
Case II
Unique winning committee



Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits two candidates from D, say p and q.

Case II
Unique winning committee

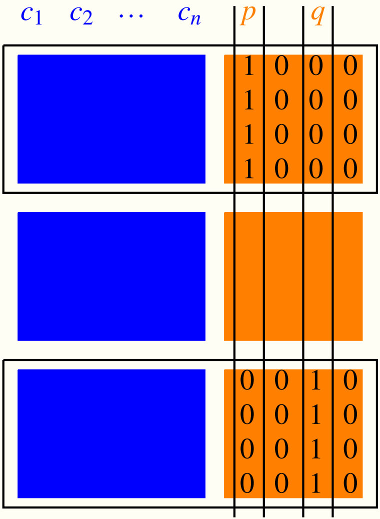


Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits two candidates from D, say p and q.
Consider the voter blocks corresponding to p and q.

Case II

Unique winning committee



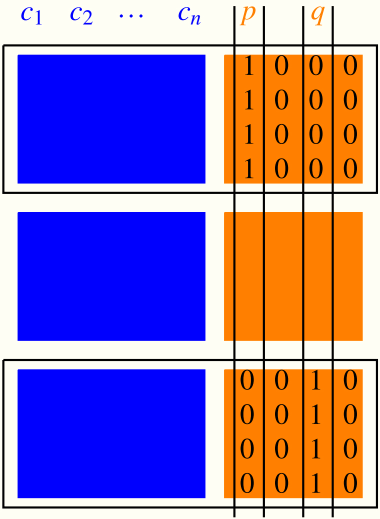
Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits two candidates from D, say p and q.
Consider the voter blocks corresponding to p and q.

Since $W \setminus D$ is not a hitting set*, there is at least one voter in each block that is not represented by W in each block.

Case II

Unique winning committee



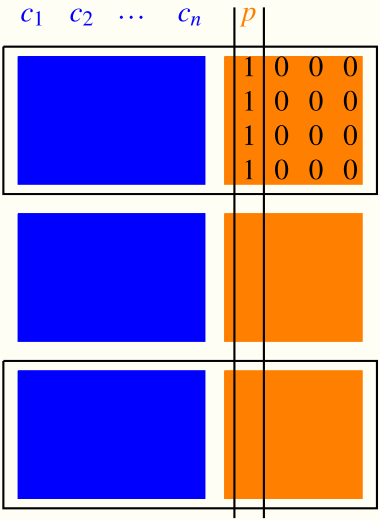
Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits two candidates from D, say p and q.
Consider the voter blocks corresponding to p and q.

Since $W \setminus D$ is not a hitting set*, there is at least one voter in each block that is not represented by W in each block.
Hence the dissatisfaction of W is at least 2. (done)

Case II

Unique winning committee

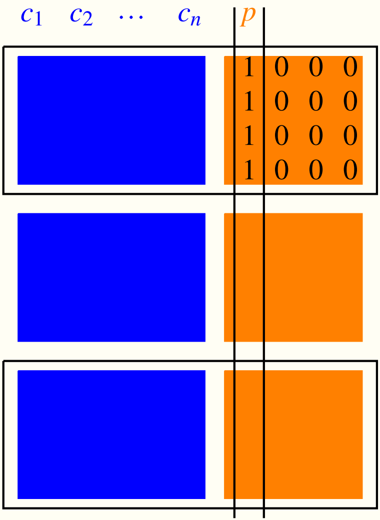


Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits one candidate from D, say p.

Case II

Unique winning committee

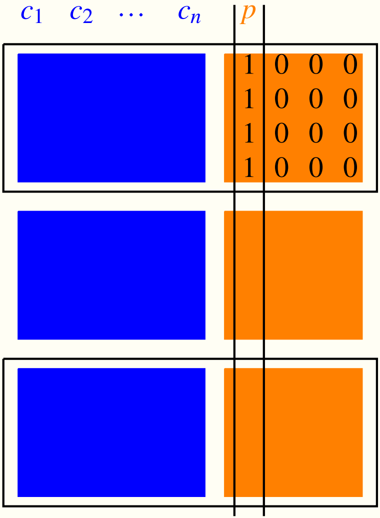


Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits one candidate from D, say p.

Consider the block corresponding to p

Part-III



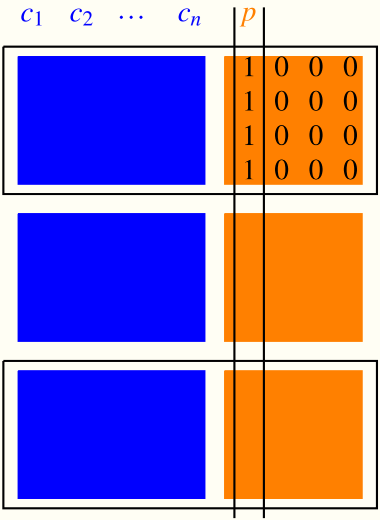
Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits one candidate from D, say p.

Since $W \setminus D$ is not a hitting set*, there is at least one voter that is not represented by W in $p's$ block.

Case II

Unique winning committee



Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

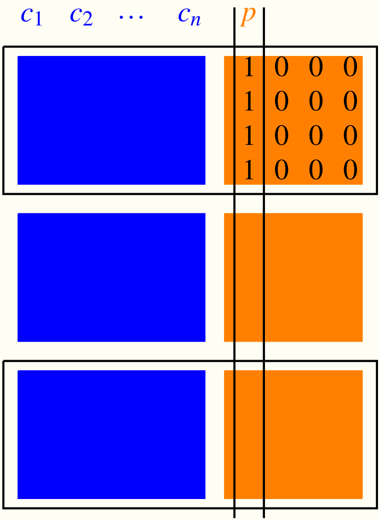
Suppose W omits one candidate from D, say p.

Since $W \setminus D$ is not a hitting set*, there is at least one voter that is not represented by W in p's block.

Part-III

Case II

Unique winning committee



Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits one candidate from D, say p.

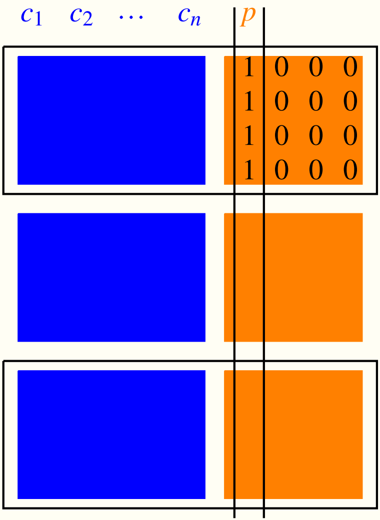
Since $W \setminus D$ is not a hitting set*, there are at least two voters that is not represented by W in $p's$ block. (Sub-case (a))

Hence, $dissatisfaction(W) > 1$.

Part-III

Case II

Unique winning committee



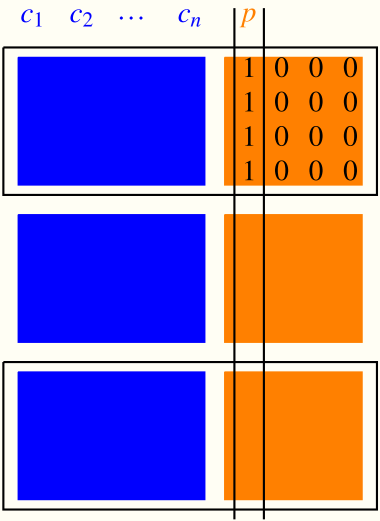
Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose W omits one candidate from D, say p.

Since $W \setminus D$ is not a hitting set*, there is exactly one voter v that is not represented by W in p's block. (Sub-case (b))

Part-III

Case II
Unique winning committee



Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

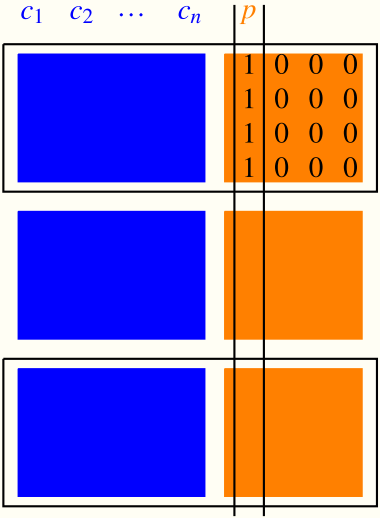
Suppose W omits one candidate from D, say p.

Since $W \setminus D$ is not a hitting set*, there is exactly one voter v that is not represented by W in $p's$ block. (Sub-case (b))

$W \setminus D$ combined with any element from the set corresponding to v, gives a hitting set of size at most k.

Part-III

Case II
Unique winning committee



(Contradicts this assumption) Suppose D is the only committee that represents every voter.
Let W be any other committee s.t. $|D| = |W|$.

Suppose D is the only committee that represents every voter.

Since $W \setminus D$ is not a hitting set*, there is exactly one voter v that is not represented by W in $p's$ block.

$W \setminus D$ combined with any element from the set corresponding to v, gives a hitting set of size $\leqslant k$.