Parameterized Complexity and its Use in Computational Social Choice

Jörg Rothe

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

EXPLORE 2014

Paris, May 6, 2014

Surveys on Parameterized Complexity in COMSOC

- C. Lindner and J. Rothe: **Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice**, A. Holder (ed.), *Mathematical Programming Glossary*, INFORMS Computing Society, 2008.

Surveys on Parameterized Complexity in COMSOC

- C. Lindner and J. Rothe: **Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice**, A. Holder (ed.), *Mathematical Programming Glossary*, INFORMS Computing Society, 2008.

- N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier: **Studies in Computational Aspects of Voting — a Parameterized Complexity Perspective**, In Bodlaender et al. (eds.): *Fellows Festschrift*, Springer-Verlag, LNCS 7370, 2012.

Surveys on Parameterized Complexity in COMSOC

- C. Lindner and J. Rothe: **Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice**, A. Holder (ed.), *Mathematical Programming Glossary*, INFORMS Computing Society, 2008.

- N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier: **Studies in Computational Aspects of Voting — a Parameterized Complexity Perspective**, In Bodlaender et al. (eds.): *Fellows Festschrift*, Springer-Verlag, LNCS 7370, 2012.

Surveys on Parameterized Complexity in COMSOC

- C. Lindner and J. Rothe: **Fixed-Parameter Tractability and Parameterized Complexity, Applied to Problems From Computational Social Choice**, A. Holder (ed.), *Mathematical Programming Glossary*, INFORMS Computing Society, 2008.
- N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier: **Studies in Computational Aspects of Voting — a Parameterized Complexity Perspective**, In Bodlaender et al. (eds.): *Fellows Festschrift*, Springer-Verlag, LNCS 7370, 2012.

- R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. Woeginger: **Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges**, *Tsinghua Science and Technology* 19(4), 2014.

Overview

- Definitions
  - Parameterized Complexity
  - Elections and Voting Systems
- Voting Problems
  - Winner and Score Problems
  - Possible Winner
  - Bribery
  - Control
  - Single-Peaked Elections
- Other COMSOC Problems:
  - Optimal Lobbying
  - Judgment Aggregation
  - Cake Cutting

Overview

- Definitions
  - Parameterized Complexity
  - Elections and Voting Systems
- Voting Problems
  - Winner and Score Problems
  - Possible Winner
  - Bribery
  - Control
  - Single-Peaked Elections
- Other COMSOC Problems: BUT THERE WILL BE NO TIME!
  - Optimal Lobbying
  - Judgment Aggregation
  - Cake Cutting

Fixed-Parameter Tractability

Definition

- A *parameterized decision problem* is a language $\mathscr{L} \subseteq \Sigma^* \times \mathbb{N}$. $\mathscr{L}$ is *fixed-parameter tractable* if there exists some computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each input (x, k), it can be determined in time $f(k) \cdot |x|^{\mathscr{O}(1)}$ whether or not (x, k) is in $\mathscr{L}$.

Fixed-Parameter Tractability

Definition

- A *parameterized decision problem* is a language $\mathscr{L} \subseteq \Sigma^* \times \mathbb{N}$.
  $\mathscr{L}$ is *fixed-parameter tractable* if there exists some computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each input (x, k), it can be determined in time $f(k) \cdot |x|^{\mathscr{O}(1)}$ whether or not (x, k) is in $\mathscr{L}$.
- FPT is the class of fixed-parameter tractable problems.

Fixed-Parameter Tractability

Definition

- A *parameterized decision problem* is a language $\mathscr{L} \subseteq \Sigma^* \times \mathbb{N}$. $\mathscr{L}$ is *fixed-parameter tractable* if there exists some computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each input (x, k), it can be determined in time $f(k) \cdot |x|^{\mathscr{O}(1)}$ whether or not (x, k) is in $\mathscr{L}$.
- FPT is the class of fixed-parameter tractable problems.
  Note: $\mathrm{FPT} \neq$ polynomial-time solvability for constant k.

Fixed-Parameter Tractability

Definition

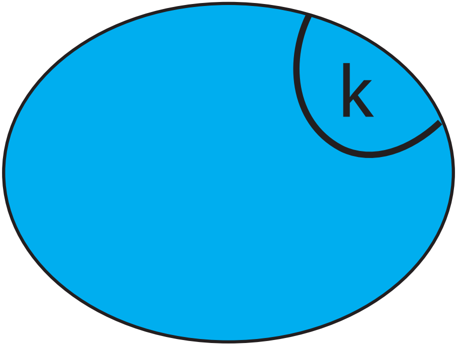
- A *parameterized decision problem* is a language $\mathscr{L} \subseteq \Sigma^* \times \mathbb{N}$. $\mathscr{L}$ is *fixed-parameter tractable* if there exists some computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each input (x,k), it can be determined in time $f(k) \cdot |x|^{\mathscr{O}(1)}$ whether or not (x,k) is in $\mathscr{L}$.
- FPT is the class of fixed-parameter tractable problems.
  Note: $\mathrm{FPT} \neq$ polynomial-time solvability for constant k.
- XP is the class of problems solvable in time $\mathscr{O}(|x|^{f(k)})$, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function depending only on k.

Fixed-Parameter Tractability

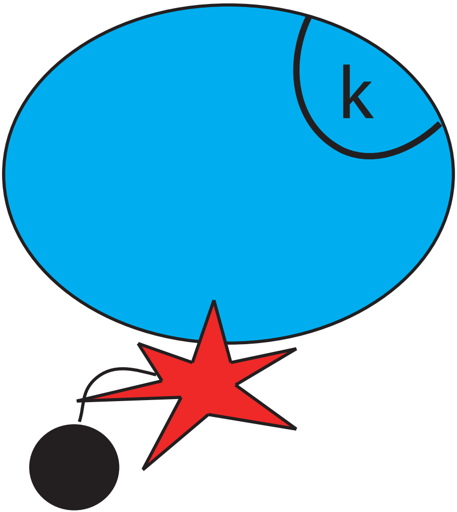
Definition

- A *parameterized decision problem* is a language $\mathscr{L} \subseteq \Sigma^* \times \mathbb{N}$. $\mathscr{L}$ is *fixed-parameter tractable* if there exists some computable function $f : \mathbb{N} \to \mathbb{N}$ such that for each input (x, k), it can be determined in time $f(k) \cdot |x|^{\mathscr{O}(1)}$ whether or not (x, k) is in $\mathscr{L}$.
- FPT is the class of fixed-parameter tractable problems.
  Note: $\mathrm{FPT} \neq$ polynomial-time solvability for constant k.
- XP is the class of problems solvable in time $\mathscr{O}(|x|^{f(k)})$, where $f : \mathbb{N} \to \mathbb{N}$ is a computable function depending only on k.
  Note: $\mathrm{FPT} \subseteq \mathrm{XP}$.

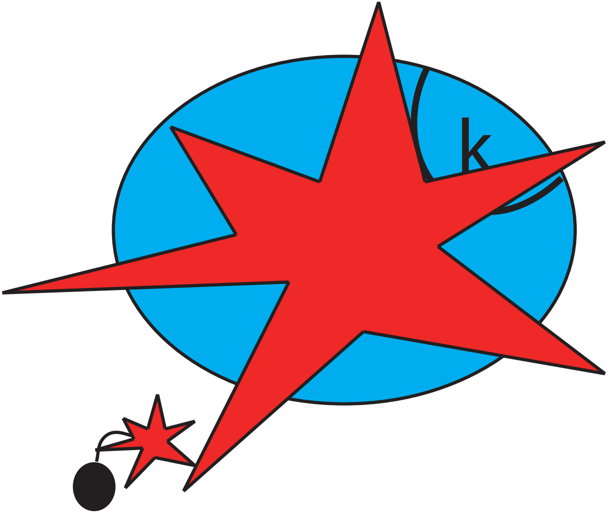
Fixed-Parameter Tractability



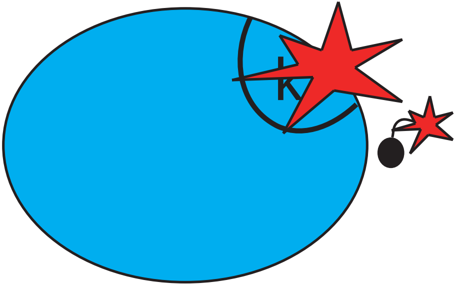
Fixed-Parameter Tractability



Fixed-Parameter Tractability



Fixed-Parameter Tractability



Kernelization

Definition

A parameterized problem $\mathscr{L}$ *has a (polynomial-size) problem kernel* if there is a polynomial-time algorithm (called *kernelization*) that on input (x, k) computes (x', k') such that

1. $(x, k) \in \mathscr{L}$ if and only if $(x', k') \in \mathscr{L}$, and
2. $|(x', k')| \leq f(k)$ for some (polynomial) function $f : \mathbb{N} \to \mathbb{N}$ depending only on k.

Kernelization

Definition

A parameterized problem $\mathscr{L}$ *has a (polynomial-size) problem kernel* if there is a polynomial-time algorithm (called *kernelization*) that on input (x,k) computes (x',k') such that

1. $(x,k) \in \mathscr{L}$ if and only if $(x',k') \in \mathscr{L}$, and
2. $|(x',k')| \leq f(k)$ for some (polynomial) function $f : \mathbb{N} \to \mathbb{N}$ depending only on k.

Note:

- $\mathscr{L} \in \mathrm{FPT}$ if and only if $\mathscr{L}$ has a problem kernel.

Kernelization

Definition

A parameterized problem $\mathscr{L}$ *has a (polynomial-size) problem kernel* if there is a polynomial-time algorithm (called *kernelization*) that on input (x,k) computes (x',k') such that

1. $(x,k) \in \mathscr{L}$ if and only if $(x',k') \in \mathscr{L}$, and
2. $|(x',k')| \leq f(k)$ for some (polynomial) function $f : \mathbb{N} \to \mathbb{N}$ depending only on k.

Note:

- $\mathscr{L} \in \mathrm{FPT}$ if and only if $\mathscr{L}$ has a problem kernel.
- Kernelization typically employs polynomial-time executable *data reduction rules* to shrink the input size.

Parameterized Reducibility

Definition

- Given two parameterized problems $\mathscr{L}$ and $\mathscr{L}'$ (both encoded over $\Sigma^* \times \mathbb{N}$), we say *$\mathscr{L}$ parameterizedly reduces to $\mathscr{L}'$* if there are two functions, $f : \Sigma^* \to \Sigma^*$ and $g : \mathbb{N} \to \mathbb{N}$, such that for each instance (x, k) of $\mathscr{L}$,
  1. $(x, k) \in \mathscr{L}$ if and only if $(f(x), g(k)) \in \mathscr{L}'$, and
  2. f can be computed in FPT time.

Parameterized Reducibility

Definition

- Given two parameterized problems $\mathscr{L}$ and $\mathscr{L}'$ (both encoded over $\Sigma^* \times \mathbb{N}$), we say *$\mathscr{L}$ parameterizedly reduces to $\mathscr{L}'$* if there are two functions, $f : \Sigma^* \to \Sigma^*$ and $g : \mathbb{N} \to \mathbb{N}$, such that for each instance (x, k) of $\mathscr{L}$,
  1. $(x, k) \in \mathscr{L}$ if and only if $(f(x), g(k)) \in \mathscr{L}'$, and
  2. f can be computed in FPT time.
- A parameterized problem $\mathscr{L}$ is *hard for a parameterized complexity class $\mathscr{C}$* if every problem in $\mathscr{C}$ parameterizedly reduces to $\mathscr{L}$.

Parameterized Reducibility

Definition

- Given two parameterized problems $\mathscr{L}$ and $\mathscr{L}'$ (both encoded over $\Sigma^* \times \mathbb{N}$), we say *$\mathscr{L}$ parameterizedly reduces to $\mathscr{L}'$* if there are two functions, $f : \Sigma^* \to \Sigma^*$ and $g : \mathbb{N} \to \mathbb{N}$, such that for each instance (x, k) of $\mathscr{L}$,
  1. $(x, k) \in \mathscr{L}$ if and only if $(f(x), g(k)) \in \mathscr{L}'$, and
  2. f can be computed in FPT time.
- A parameterized problem $\mathscr{L}$ is *hard for a parameterized complexity class $\mathscr{C}$* if every problem in $\mathscr{C}$ parameterizedly reduces to $\mathscr{L}$.
- $\mathscr{L}$ is *complete for $\mathscr{C}$* if it both belongs to $\mathscr{C}$ and is hard for $\mathscr{C}$.

Parameterized Complexity

WEIGHTED WEFT-t DEPTH-d CIRCUIT SATISFIABILITY (WCS(t,d))	
Given:	A boolean circuit of weft t and depth d, and an integer bound k.
Question:	Is there a satisfying assignment of weight k (i.e., setting k variables to true)?

Parameterized Complexity

WEIGHTED WEFT-t DEPTH-d CIRCUIT SATISFIABILITY (WCS(t, d))

Given:	A boolean circuit of weft t and depth d, and an integer bound k.
Question:	Is there a satisfying assignment of weight k (i.e., setting k variables to true)?

- Here, a boolean circuit may contain
  - NOT, AND, and OR gates of fan-in at most 2,
  - large AND and OR gates of unbounded fan-in.

Parameterized Complexity

WEIGHTED WEFT-t DEPTH-d CIRCUIT SATISFIABILITY (WCS(t, d))

Given:	A boolean circuit of weft t and depth d, and an integer bound k.
Question:	Is there a satisfying assignment of weight k (i.e., setting k variables to true)?

- Here, a boolean circuit may contain
  - NOT, AND, and OR gates of fan-in at most 2,
  - large AND and OR gates of unbounded fan-in.
- The *weft* of a circuit is the maximum number of large gates on any path from input to output gates.

Parameterized Complexity

WEIGHTED WEFT-t DEPTH-d CIRCUIT SATISFIABILITY (WCS(t,d))

Given:	A boolean circuit of weft t and depth d, and an integer bound k.
Question:	Is there a satisfying assignment of weight k (i.e., setting k variables to true)?

- Here, a boolean circuit may contain
  - NOT, AND, and OR gates of fan-in at most 2,
  - large AND and OR gates of unbounded fan-in.
- The *weft* of a circuit is the maximum number of large gates on any path from input to output gates.
- The *depth* of a circuit is the maximum number of gates on any path from input to output gates.

Parameterized Complexity: The W-Hierarchy

Definition

The *W-hierarchy* consists of the classes $\mathrm{W}[t]$, $t \geq 1$, where $\mathrm{W}[t]$ is the class of parameterized problems parameterizedly reducible (w.r.t. the given parameter) to $\mathrm{WCS}(t,d)$ for some constant $d \geq 1$.

Parameterized Complexity: The W-Hierarchy

Definition

The *W-hierarchy* consists of the classes $\mathrm{W}[t]$, $t \geq 1$, where $\mathrm{W}[t]$ is the class of parameterized problems parameterizedly reducible (w.r.t. the given parameter) to $\mathrm{WCS}(t, d)$ for some constant $d \geq 1$.

Note:

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\mathrm{t}] \subseteq \cdots \subseteq \mathrm{XP}.$$

Parameterized Complexity: The W-Hierarchy

Definition

The *W-hierarchy* consists of the classes $\mathrm{W}[t]$, $t \geq 1$, where $\mathrm{W}[t]$ is the class of parameterized problems parameterizedly reducible (w.r.t. the given parameter) to $\mathrm{WCS}(t, d)$ for some constant $d \geq 1$.

Note:

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[t] \subseteq \cdots \subseteq \mathrm{XP}.$$

- To classify some problem $\mathscr{L}$, parameterizedly reduce
  - $\mathscr{L}$ to some known problem in $\mathrm{W}[t]$ (membership in $\mathrm{W}[t]$) and
  - some known $\mathrm{W}[t]$-hard problem to $\mathscr{L}$ ($\mathrm{W}[t]$-hardness).

Elections and Voting Systems

Definition (Election)

An *election* is a pair (C, V) with

- a finite set C of candidates (or alternatives) and
- a finite list V of votes expressing the voters' preferences over the candidates in C.

Elections and Voting Systems

Definition (Election)

An *election* is a pair (C, V) with

- a finite set C of candidates (or alternatives) and
- a finite list V of votes expressing the voters' preferences over the candidates in C.

Definition (Voting System)

A *voting system* is a set of rules that

- define the form of the voters' ballots (representation of the voters' preferences) in V and
- determine the winner(s) in C according to the ballots in V.

Scoring Rules: Plurality, k-Approval, and Borda

- Scoring vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$

Scoring Rules: Plurality, k-Approval, and Borda

- Scoring vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$
- Plurality: $\alpha = (1, 0, \ldots, 0)$
- k-Approval: $\alpha = (\underbrace{1, \ldots, 1}_{k}, 0, \ldots, 0)$
- Borda: $\alpha = (m-1, m-2, \ldots, 0)$

Scoring Rules: Plurality, k-Approval, and Borda

- Scoring vector $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$
- Plurality: $\alpha = (1, 0, \dots, 0)$
- k-Approval: $\alpha = (\underbrace{1, \dots, 1}_{k}, 0, \dots, 0)$
- Borda: $\alpha = (m-1, m-2, \dots, 0)$

	Plurality			2-Approval			Borda		
Preference profile	A	B	C	A	B	C	A	B	C
$A > B > C$	1	0	0	1	1	0	2	1	0
$B > C > A$	0	1	0	0	1	1	0	2	1
$A > B > C$	1	0	0	1	1	0	2	1	0
Scores:	**2**	1	0	2	**3**	1	**4**	**4**	1

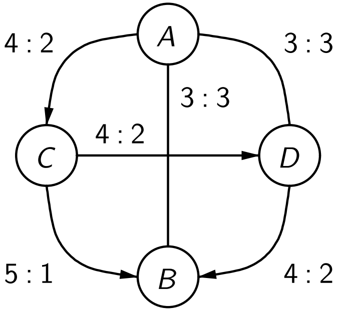
Voting Systems Based on Pairwise Comparison: Example

	Pairwise comparison					
Preference profile	$A?B$	$A?C$	$A?D$	$B?C$	$B?D$	$C?D$
$A > D > C > B$	A	A	A	C	D	D
$C > D > B > A$	B	C	D	C	D	C
$C > D > B > A$	B	C	D	C	D	C
$B > D > A > C$	B	A	D	B	B	D
$A > C > D > B$	A	A	A	C	D	C
$A > C > B > D$	A	A	A	C	B	C
Winner of the comparison:	?	A	?	C	D	C

Table: Example of an election without a Condorcet winner

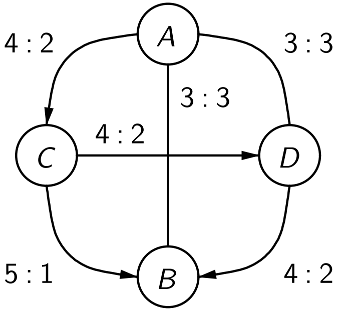
Voting Systems Based on Pairwise Comparison: Copeland

Majority graph



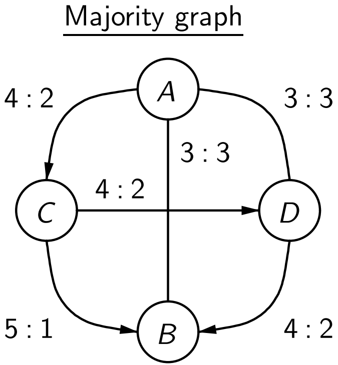
Voting Systems Based on Pairwise Comparison: Copeland

Majority graph



Copeland$^\alpha$ score of a candidate:
1 point for each pairwise win
plus α points for each tie, where
$\alpha \in [0,1]$ is a rational number.

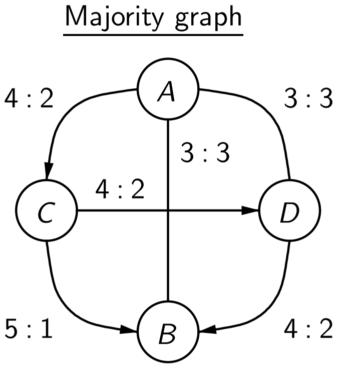
Voting Systems Based on Pairwise Comparison: Copeland



Copeland$^\alpha$ score of a candidate:
1 point for each pairwise win
plus α points for each tie, where
$\alpha \in [0,1]$ is a rational number.

$$\begin{aligned} Copeland^{\alpha}Score(A) &= 1+2\alpha, \\ Copeland^{\alpha}Score(B) &= \alpha, \\ Copeland^{\alpha}Score(C) &= 2, \\ Copeland^{\alpha}Score(D) &= 1+\alpha. \end{aligned}$$

Voting Systems Based on Pairwise Comparison: Copeland

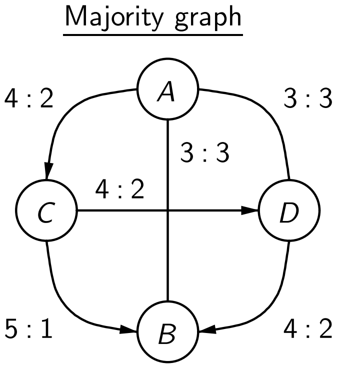


Copeland$^\alpha$ score of a candidate:
1 point for each pairwise win
plus α points for each tie, where
$\alpha \in [0,1]$ is a rational number.

$$
\begin{aligned}
\mathit{Copeland}^\alpha\mathit{Score}(A) &= 1+2\alpha,\\
\mathit{Copeland}^\alpha\mathit{Score}(B) &= \alpha,\\
\mathit{Copeland}^\alpha\mathit{Score}(C) &= 2,\\
\mathit{Copeland}^\alpha\mathit{Score}(D) &= 1+\alpha.
\end{aligned}
$$

C wins if $\alpha = 0$;

Voting Systems Based on Pairwise Comparison: Copeland



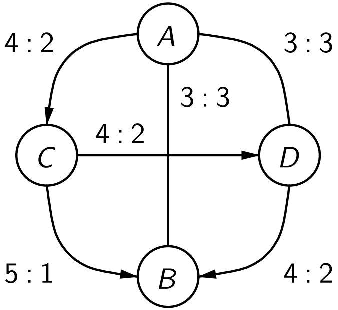
Copeland$^\alpha$ score of a candidate: 1 point for each pairwise win plus α points for each tie, where $\alpha \in [0,1]$ is a rational number.

$$
\begin{aligned}
\mathit{Copeland}^{\alpha}\mathit{Score}(A) &= 1+2\alpha,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(B) &= \alpha,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(C) &= 2,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(D) &= 1+\alpha.
\end{aligned}
$$

C wins if $\alpha = 0$; A and C win if $\alpha = 1/2$;

Voting Systems Based on Pairwise Comparison: Copeland

Majority graph



Copeland$^\alpha$ score of a candidate:
1 point for each pairwise win
plus α points for each tie, where
$\alpha \in [0,1]$ is a rational number.

$$
\begin{aligned}
\mathit{Copeland}^{\alpha}\mathit{Score}(A) &= 1+2\alpha,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(B) &= \alpha,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(C) &= 2,\\
\mathit{Copeland}^{\alpha}\mathit{Score}(D) &= 1+\alpha.
\end{aligned}
$$

C wins if $\alpha = 0$; A and C win if $\alpha = 1/2$; A wins if $\alpha = 1$.

Winner and Score Problems: Definition

$\mathscr{E}$-Winner	
Given:	An election (C, V) and a distinguished candidate $c \in C$.
Question:	Is c a winner of (C, V) according to voting system $\mathscr{E}$?

Winner and Score Problems: Definition

$\mathscr{E}$-WINNER	
Given:	An election (C, V) and a distinguished candidate $c \in C$.
Question:	Is c a winner of (C, V) according to voting system $\mathscr{E}$?

- For most voting systems (scoring rules, Condorcet, approval, Copeland, ...), the winners can be determined in polynomial time.

Winner and Score Problems: Definition

$\mathscr{E}$-WINNER	
Given:	An election (C, V) and a distinguished candidate $c \in C$.
Question:	Is c a winner of (C, V) according to voting system $\mathscr{E}$?

- For most voting systems (scoring rules, Condorcet, approval, Copeland, ...), the winners can be determined in polynomial time.
- A few exceptions: Winner determination is complete for $\mathrm{P}_{||}^{\mathrm{NP}}$ in
  - **Dodgson** (Hemaspaandra, Hemaspaandra, Rothe, *JACM* 44(6), 1997),
  - **Young** (Rothe, Spakowski, Vogel, *TOCS* 36(4), 2003), and
  - **Kemeny** voting (Hemaspaandra, Spakowski, Vogel, *TCS* 349(3), 2005).

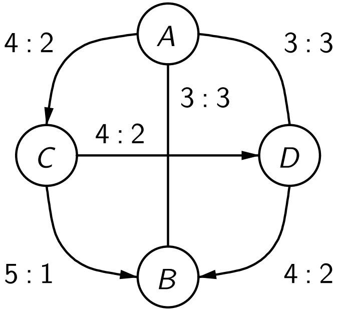
Winner and Score Problems: Example of Dodgson Election

	Pairwise comparison					
Preference profile	$A?B$	$A?C$	$A?D$	$B?C$	$B?D$	$C?D$
$A > D > C > B$	A	A	A	C	D	D
$C > D > B > A$	B	C	D	C	D	C
$C > D > B > A$	B	C	D	C	D	C
$B > D > A > C$	B	A	D	B	B	D
$A > C > D > B$	A	A	A	C	D	C
$A > C > B > D$	A	A	A	C	B	C
Winner of the comparison:	?	A	?	C	D	C

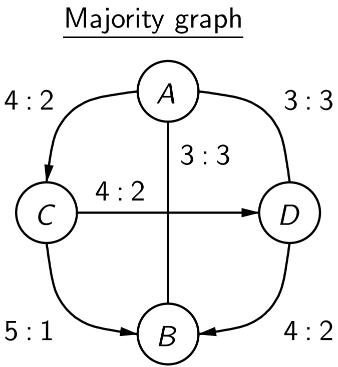
Table: Example of an election without a Condorcet winner

Winner and Score Problems: Example of Dodgson Election

Majority graph

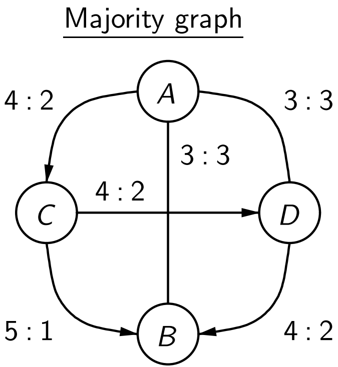


Winner and Score Problems: Example of Dodgson Election



Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

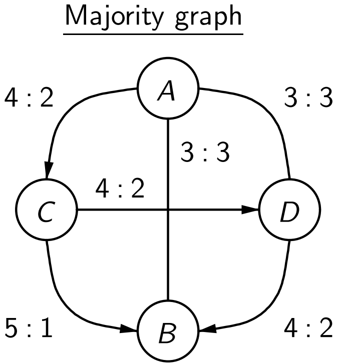
Winner and Score Problems: Example of Dodgson Election



Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

A has a *Dodgson score* of 2.

Winner and Score Problems: Example of Dodgson Election



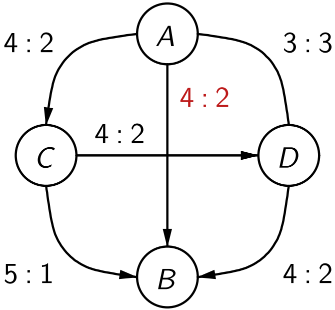
Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

A has a *Dodgson score* of 2.

2^{nd} vote: $C > D > B > A$

Winner and Score Problems: Example of Dodgson Election

Majority graph

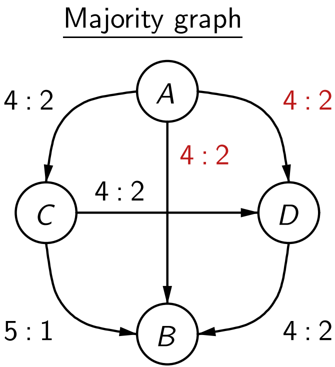


Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

A has a *Dodgson score* of 2.

2nd vote: $C > D > B > A \quad \rightsquigarrow \quad C > D > A > B$

Winner and Score Problems: Example of Dodgson Election



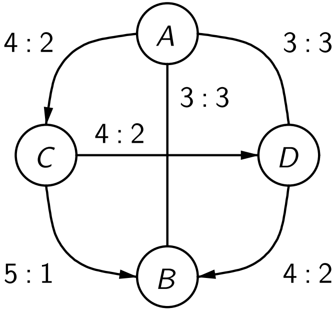
Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

A has a *Dodgson score* of 2.

2nd vote: $C > D > B > A \rightsquigarrow C > D > A > B \rightsquigarrow C > A > D > B$

Winner and Score Problems: Example of Dodgson Election

Majority graph



5^{th} vote: $A > C > D > B$

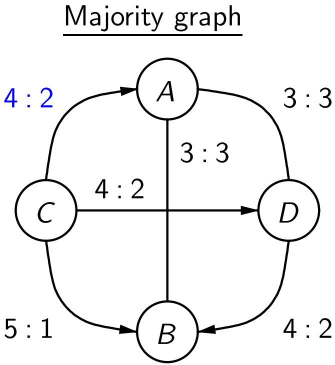
6^{th} vote: $A > C > B > D$

Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

A has a *Dodgson score* of 2.

C has a *Dodgson score* of 2.

Winner and Score Problems: Example of Dodgson Election



Dodgson score of a candidate: smallest number of swaps needed to make her a Condorcet winner.

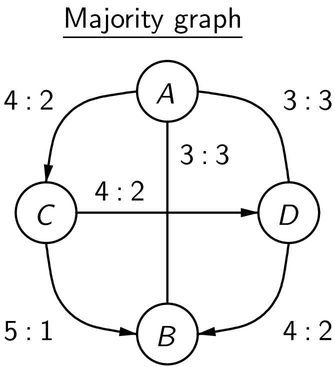
A has a *Dodgson score* of 2.

C has a *Dodgson score* of 2.

5th vote: $A > C > D > B \quad \rightsquigarrow \quad C > A > D > B$

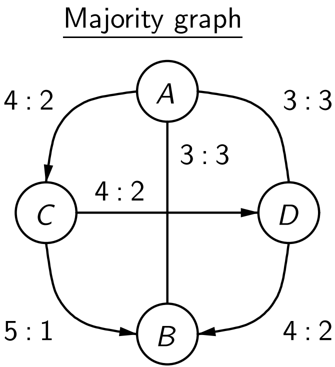
6th vote: $A > C > B > D \quad \rightsquigarrow \quad C > A > B > D$

Winner and Score Problems: Example of Young Election



Young score of a candidate: largest number of votes for which she is a weak Condorcet winner.

Winner and Score Problems: Example of Young Election

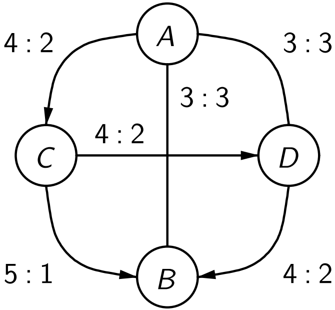


Young score of a candidate: largest number of votes for which she is a weak Condorcet winner.

Dual Young score of a candidate: smallest number of votes that need to be deleted to make her a weak Condorcet winner.

Winner and Score Problems: Example of Young Election

Majority graph

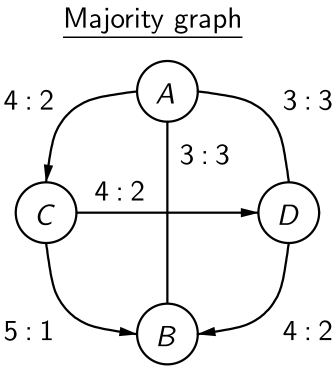


Young score of a candidate: largest number of votes for which she is a weak Condorcet winner.

Dual Young score of a candidate: smallest number of votes that need to be deleted to make her a weak Condorcet winner.

A has a *dual Young score* of 0.

Winner and Score Problems: Example of Young Election



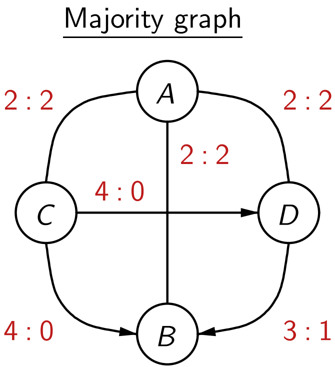
Young score of a candidate: largest number of votes for which she is a weak Condorcet winner.

Dual Young score of a candidate: smallest number of votes that need to be deleted to make her a weak Condorcet winner.

A has a *dual Young score* of 0.

C has a *dual Young score* of 2.

Winner and Score Problems: Example of Young Election



~~$A > D > C > B$~~
$C > D > B > A$
$C > D > B > A$
~~$B > D > A > C$~~
$A > C > D > B$
$A > C > B > D$

C has a *dual Young score* of 2.

Dodgson and Young Score: Definition and Overview

Dodgson Score	
Given:	An election (C, V), a distinguished candidate $c \in C$, and an integer $k > 0$.
Question:	Is the Dodgson score of c in (C, V) at most k?

Dodgson and Young Score: Definition and Overview

Dodgson Score	
Given:	An election (C, V), a distinguished candidate $c \in C$, and an integer $k > 0$.
Question:	Is the Dodgson score of c in (C, V) at most k?

- Young Score and Dual Young Score are defined analogously.

Dodgson and Young Score: Definition and Overview

Dodgson Score	
Given:	An election (C, V), a distinguished candidate $c \in C$, and an integer $k > 0$.
Question:	Is the Dodgson score of c in (C, V) at most k?

- Young Score and Dual Young Score are defined analogously.
- These problems are NP-complete (Bartholdi, Tovey, Trick, *SCW* 6(2), 1989).

Dodgson and Young Score: Overview

Parameter	Dodgson	Dual Young	Young
$m = \#$ candidates	FPT	FPT	FPT
$n = \#$ votes	W[1]-hard	FPT ($\mathcal{O}^*(2^n)$)	FPT ($\mathcal{O}^*(2^n)$)
$k = \#$ swaps	FPT ($\mathcal{O}^*(2^k)$)	—	—
$k = \#$ deleted votes	—	W[2]-complete	—
$k = \#$ remaining votes	—	—	W[2]-complete

Table: Overview of parameterized complexity for Score problems

Bartholdi, Tovey, Trick: **Voting Schemes for Which it Can Be Difficult to Tell Who Won the Election**, *SCW* 6(2), 1989
Young: **Extending Condorcet's Rule**, *JET* 16(2), 1977
Fellows, Jansen, Lokshtanov, Rosamond, Saurabh: **Determining the Winner of a Dodgson Election is Hard**, *FSTTCS*, 2010
Betzler, Guo, Niedermeier: **Parameterized Computational Complexity of Dodgson and Young Elections**, *I&C* 208(2), 2010
Rothe, Spakowski, Vogel: **Exact Complexity of the Winner Problem for Young Elections**, *TOCS* 36(4), 2003

Dodgson Score is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \qquad \text{subject to}$$

$$\forall i \in \tilde{V} \quad : \quad \sum_{j} x_{i,j} = N_i$$

$$\forall y \in C \quad : \quad \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y$$

$$x_{i,j} \geq 0$$

Dodgson Score is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \qquad \text{subject to}$$

$$\forall i \in \tilde{V} \;:\; \sum_{j} x_{i,j} = N_i$$

$$\forall y \in C \;:\; \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y$$

$$x_{i,j} \geq 0$$

where

- $\tilde{V}$ lists the *different* preference types

Dodgson Score is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \qquad \text{subject to}$$

$$\forall i \in \tilde{V} \quad : \quad \sum_{j} x_{i,j} = N_i$$

$$\forall y \in C \quad : \quad \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y$$

$$x_{i,j} \geq 0$$

where

- $\tilde{V}$ lists the *different* preference types
- N_i is the number of votes of type i

Dodgson Score is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \qquad \text{subject to}$$

$$\forall i \in \tilde{V} \quad : \quad \sum_{j} x_{i,j} = N_i$$

$$\forall y \in C \quad : \quad \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y$$

$$x_{i,j} \geq 0$$

where

- $\tilde{V}$ lists the *different* preference types
- N_i is the number of votes of type i
- $x_{i,j}$ is the number of type-i votes for which the designated candidate c will be moved upward by j positions

DODGSON SCORE is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \qquad \text{subject to}$$

$$\begin{aligned} \forall i \in \tilde{V} &: \sum_{j} x_{i,j} = N_i \\ \forall y \in C &: \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y \\ & \quad x_{i,j} \geq 0 \end{aligned}$$

where

- $\tilde{V}$ lists the *different* preference types
- N_i is the number of votes of type i
- $x_{i,j}$ is the number of type-i votes for which the designated candidate c will be moved upward by j positions
- d_y is c's deficit with respect to y

Dodgson Score is FPT by Integer Linear Program

$$\min \sum_{i,j} j \cdot x_{i,j} \quad \text{subject to}$$

$$\forall i \in \tilde{V} \;:\; \sum_{j} x_{i,j} = N_i$$

$$\forall y \in C \;:\; \sum_{i,j} e_{i,j,y} \cdot x_{i,j} \geq d_y$$

$$x_{i,j} \geq 0$$

where

- $\tilde{V}$ lists the *different* preference types
- N_i is the number of votes of type i
- $x_{i,j}$ is the number of type-i votes for which the designated candidate c will be moved upward by j positions
- d_y is c's deficit with respect to y

- $e_{i,j,y} = \begin{cases} 1 & \text{if } c \text{ gains an additional voter support against } y \text{ when } c \text{ is moved upward by } j \text{ positions in a type-}i\text{ vote} \\ 0 & \text{otherwise} \end{cases}$

FPT by ILPs

- Many further FPT results are based on ILPs:
  - Betzler, Hemmann, Niedermeier (IJCAI-2009)
  - Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR* 35, 2009)
  - Betzler, Niedermeier, Woeginger (IJCAI-2011)
  - Dorn, Schlotter (*Algorithmica* 64(1), 2012)
  - Bredereck, Chen, Hartung, Kratsch, Niedermeier, Suchý (AAAI-2012)
  - Alon, Bredereck, Chen, Kratsch, Niedermeier, Woeginger (ADT-2013)

FPT by ILPs

- Many further FPT results are based on ILPs:
  - Betzler, Hemmann, Niedermeier (IJCAI-2009)
  - Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR* 35, 2009)
  - Betzler, Niedermeier, Woeginger (IJCAI-2011)
  - Dorn, Schlotter (*Algorithmica* 64(1), 2012)
  - Bredereck, Chen, Hartung, Kratsch, Niedermeier, Suchý (AAAI-2012)
  - Alon, Bredereck, Chen, Kratsch, Niedermeier, Woeginger (ADT-2013)
- For a bounded number of variables, such ILPs can be solved in polynomial time by the famous algorithm due to H. Lenstra Jr.: **Integer Programming with a Fixed Number of Variables**, *MOR* 8, 1983.

FPT by ILPs

- Many further FPT results are based on ILPs:
  - Betzler, Hemmann, Niedermeier (IJCAI-2009)
  - Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR* 35, 2009)
  - Betzler, Niedermeier, Woeginger (IJCAI-2011)
  - Dorn, Schlotter (*Algorithmica* 64(1), 2012)
  - Bredereck, Chen, Hartung, Kratsch, Niedermeier, Suchý (AAAI-2012)
  - Alon, Bredereck, Chen, Kratsch, Niedermeier, Woeginger (ADT-2013)
- For a bounded number of variables, such ILPs can be solved in polynomial time by the famous algorithm due to H. Lenstra Jr.: **Integer Programming with a Fixed Number of Variables**, *MOR* 8, 1983.
- Advantage: Great classification tool, mainly of theoretical interest.

FPT by ILPs

- Many further FPT results are based on ILPs:
  - Betzler, Hemmann, Niedermeier (IJCAI-2009)
  - Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR* 35, 2009)
  - Betzler, Niedermeier, Woeginger (IJCAI-2011)
  - Dorn, Schlotter (*Algorithmica* 64(1), 2012)
  - Bredereck, Chen, Hartung, Kratsch, Niedermeier, Suchý (AAAI-2012)
  - Alon, Bredereck, Chen, Kratsch, Niedermeier, Woeginger (ADT-2013)
- For a bounded number of variables, such ILPs can be solved in polynomial time by the famous algorithm due to H. Lenstra Jr.: **Integer Programming with a Fixed Number of Variables**, *MOR* 8, 1983.
- Advantage: Great classification tool, mainly of theoretical interest.
- Disadvantage: HUGE exponential function in number of variables $\Rightarrow$ not *practically* feasible; e.g., **above ILP has $\mathbf{m \cdot m!}$ variables $\mathbf{x_{i,j}}$.**

Research Challenge 1: ILP $\Rightarrow$ direct FPT Algorithms

Research Challenge 1

Can one replace the ILPs in these known ILP-based FPT results by direct *combinatorial* fixed-parameter algorithms?

Possible Winner Problem: Example

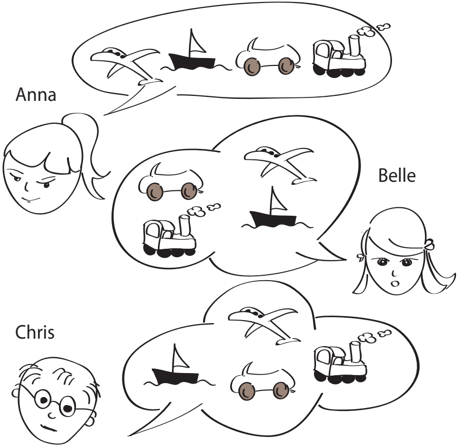


Figure: Trip preferences of Anna, Belle, and Chris

Possible Winner Problem: Definition

$\mathscr{E}$-Possible-Winner	
Given:	An election (C, V), where the votes are represented as partial orders over C, and a distinguished candidate c.
Question:	Is c a possible $\mathscr{E}$ winner of (C, V), i.e., is it possible to fully extend each vote in V such that c wins the election?

- Introduced by Konczak and Lang: **Voting Procedures with Incomplete Preferences**, *IJCAI Workshop on Advances in Preference Handling, 2005*

Possible Winner Problem: Definition

$\mathscr{E}$-Possible-Winner

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

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

- Introduced by Konczak and Lang: **Voting Procedures with Incomplete Preferences**, *IJCAI Workshop on Advances in Preference Handling, 2005*
- Classical complexity has been studied by many authors, e.g., by:
  - Walsh, *AAAI*, 2007
  - Betzler and Dorn, *JCSS* 76(8), 2010
  - Xia and Conitzer, *JAIR* 41, 2011
  - Baumeister and Rothe, *IPL* 112(5), 2012

Possible Winner Problem: Overview

Parameter	Borda	k-Approval	Copeland$^{\alpha}$
$m = \#$ candidates	FPT	FPT	FPT
$n = \#$ votes	para-NP-comp	para-NP-comp	?
$s = \#$ undetermined candidate pairs	$\mathcal{O}^*(1.82^s)$	$\mathcal{O}^*(2^s)$	$\mathcal{O}^*(2^s)$
$u = \max\#$ undetermined candidate pairs	para-NP-comp	para-NP-comp	para-NP-comp

Table: Overview of classical and parameterized complexity of POSSIBLE WINNER

ILP based on Lenstra: **Integer Programming with a Fixed Number of Variables**, *MOR* 8, 1983

Betzler, Hemmann, Niedermeier: **A Multivariate Complexity Analysis of Determining Possible Winners Given Incomplete Votes**, *IJCAI*, 2009

Xia and Conitzer: **Determining Possible and Necessary Winners Given Partial Orders**, *JAIR* 41, 2011

Possible Winner Problem: Overview for k-Approval

Parameter	Result	Remark
$k = \#$ of ones in vector	NP-comp	for each fixed $k \geq 2$
(t, k), $t = \#$ incomplete votes	FPT	super-exponential kernel
(t, k'), $k' = \#$ of zeros in vector	FPT	$\mathcal{O}\left(\min\left\{2^{t^2 k'}, 2^{tk'} \cdot (tk')^{k'}\right\}\right)$

Table: Overview

Xia and Conitzer: **Determining Possible and Necessary Winners Given Partial Orders**, *JAIR* 41, 2011

Betzler: **On Problem Kernels for Possible Winner Determination Under the k-Approval Protocol**, *MFCS*, 2010

Research Challenge 2: Possible Winner

Research Challenge 2

- Previous classical results on Possible Winner consider only voting systems with efficient winner determination.

Research Challenge 2: Possible Winner

Research Challenge 2

- Previous classical results on POSSIBLE WINNER consider only voting systems with efficient winner determination.
  Do the FPT results for DODGSON SCORE, YOUNG SCORE, DUAL YOUNG SCORE, and KEMENY SCORE transfer to POSSIBLE WINNER?

Research Challenge 2: Possible Winner

Research Challenge 2

- Previous classical results on Possible Winner consider only voting systems with efficient winner determination.
  Do the FPT results for Dodgson Score, Young Score, Dual Young Score, and Kemeny Score transfer to Possible Winner?
- What about the parameters
  - average number of candidate pairs
  - maximum number of candidate pairs

  in which a candidate is involved?

Bribery: Definition

$\mathscr{E}$-BRIBERY	
Given:	An election (C, V), a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V\|$.
Question:	Is it possible to make c an $\mathscr{E}$ winner of the election that results from changing no more than k votes in V?

Bribery: Definition

$\mathscr{E}$-Bribery	
Given:	An election (C, V), a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V\|$.
Question:	Is it possible to make c an $\mathscr{E}$ winner of the election that results from changing no more than k votes in V?

- $\mathscr{E}$-\$Bribery: Each voter has an individual price and the briber a budget.

Faliszewski, Hemaspaandra, Hemaspaandra: **How Hard Is Bribery in Elections?**, *JAIR* 35, 2009

Bribery: Definition

$\mathscr{E}$-BRIBERY	
Given:	An election (C, V), a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V\|$.
Question:	Is it possible to make c an $\mathscr{E}$ winner of the election that results from changing no more than k votes in V?

- $\mathscr{E}$-$BRIBERY: Each voter has an individual price and the briber a budget.

Faliszewski, Hemaspaandra, Hemaspaandra: **How Hard Is Bribery in Elections?**, *JAIR* 35, 2009

- $\mathscr{E}$-SWAP BRIBERY: Each voter has a swap-bribery price function that gives the cost of swapping any two adjacent candidates.

Bribery: Definition

$\mathscr{E}$-Bribery	
Given:	An election (C, V), a distinguished candidate $c \in C$, and a nonnegative integer $k \leq \|V\|$.
Question:	Is it possible to make c an $\mathscr{E}$ winner of the election that results from changing no more than k votes in V?

- $\mathscr{E}$-$Bribery: Each voter has an individual price and the briber a budget.

Faliszewski, Hemaspaandra, Hemaspaandra: **How Hard Is Bribery in Elections?**, *JAIR* 35, 2009

- $\mathscr{E}$-Swap Bribery: Each voter has a swap-bribery price function that gives the cost of swapping any two adjacent candidates.
- $\mathscr{E}$-Shift Bribery: Like above, except that each swap must involve c.

Elkind, Faliszewski, Slinko: **Swap Bribery**, *SAGT*, 2009

Swap Bribery: Overview for k-Approval

Parameter	Result	Remark
β = budget	W[1]-hard	for $n = 1$; reduction from MULTI-COLORED CLIQUE
k = # of ones	W[1]-hard	reduction from CLIQUE
m = # candidates	FPT	for constant k; ILP
n = # votes	FPT	for constant k; color-coding
(β, n)	FPT	kernel with $n^2\beta^2$ cand's, $n^2\beta$ votes
(β, n, k)	FPT	kernel with $(n+k)\beta$ cand's, $n^2\beta$ votes

Table: Overview

Dorn and Schlotter: **Multivariate Complexity Analysis of Swap Bribery**, *Algorithmica* 64(1), 2012

Research Challenge 3: Bribery

- For most natural voting systems $\mathscr{E}$, when parameterized by the number of candidates,
  - $\mathscr{E}$-BRIBERY tends to be FPT, whereas
  - the other bribery variants are only known to be in XP.

Research Challenge 3: Bribery

- For most natural voting systems $\mathscr{E}$, when parameterized by the number of candidates,
  - $\mathscr{E}$-Bribery tends to be FPT, whereas
  - the other bribery variants are only known to be in XP.

Research Challenge 3

For natural voting systems $\mathscr{E}$, what is the exact parameterized complexity of the problems

- $\mathscr{E}$-\$Bribery,
- $\mathscr{E}$-Swap Bribery, and
- $\mathscr{E}$-Shift Bribery

when parameterized by the number of candidates?

Research Challenge 4: FPT Approximation Schemes

- MAX VERTEX COVER is known to be W[1]-complete w.r.t. the parameter k of vertices to pick.

Research Challenge 4: FPT Approximation Schemes

- Max Vertex Cover is known to be W[1]-complete w.r.t. the parameter k of vertices to pick.
- Best known approximation algorithm (due to Ageev and Sviridenko, IPCO-1999) achieves a ratio of $\frac{3}{4}$ (i.e., $\frac{3}{4}\mathrm{OPT}$ edges are guaranteed to be covered).

Research Challenge 4: FPT Approximation Schemes

- MAX VERTEX COVER is known to be W[1]-complete w.r.t. the parameter k of vertices to pick.
- Best known approximation algorithm (due to Ageev and Sviridenko, IPCO-1999) achieves a ratio of $\frac{3}{4}$ (i.e., $\frac{3}{4}$OPT edges are guaranteed to be covered).
- Marx (2008) provided an FPT approximation scheme that, for each positive ε,
  - covers at least $(1-\varepsilon)$OPT edges and
  - runs in FPT time w.r.t. k and ε.

Research Challenge 4: FPT Approximation Schemes

- MAX VERTEX COVER is known to be W[1]-complete w.r.t. the parameter k of vertices to pick.
- Best known approximation algorithm (due to Ageev and Sviridenko, IPCO-1999) achieves a ratio of $\frac{3}{4}$
(i.e., $\frac{3}{4}$OPT edges are guaranteed to be covered).
- Marx (2008) provided an FPT approximation scheme that, for each positive ε,
    - covers at least $(1-\varepsilon)$OPT edges and
    - runs in FPT time w.r.t. k and ε.

Research Challenge 4

For which computationally hard voting problems (in particular those related to bribery) are there FPT approximation schemes?

Control: Definition

Electoral Control

Structural change exerted by an external actor, the "chair," intending

- **constructive**: to make a distinguished candidate win
- **destructive**: to prevent a distinguished candidate from winning

Bartholdi, Tovey, Trick: **How hard is it to control an election?**, *Mathematical Comput. Modelling*, 16(8/9), 1992

Hemaspaandra, Hemaspaandra, Rothe: **Anyone but him: The complexity of precluding an alternative**, *Artificial Intelligence*, 171(5-6), 2007.

Types of Control

Candidate Control:

Voter Control:

Types of Control

Candidate Control:

- Adding Candidates (limited and unlimited number)

Voter Control:

- Adding Voters

Types of Control

Candidate Control:

- Adding Candidates (limited and unlimited number)
- Deleting Candidates

Voter Control:

- Adding Voters
- Deleting Voters

Types of Control

Candidate Control:

- Adding Candidates (limited and unlimited number)
- Deleting Candidates
- Partition of Candidates (with or without run-off)

Voter Control:

- Adding Voters
- Deleting Voters
- Partition of Voters

Types of Control

Candidate Control:

- Adding Candidates (limited and unlimited number)
- Deleting Candidates
- Partition of Candidates (with or without run-off)

Voter Control:

- Adding Voters
- Deleting Voters
- Partition of Voters

$\mathscr{E}$-Constructive-Control-by-Deleting-Voters ($\mathscr{E}$-CCDV)

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

Question: Does there exist a sublist V' of V with $\|V \setminus V'\| \leq k$ such that c is an $\mathscr{E}$ winner of (C, V')?

Bucklin Voting (BV)

Example (Bucklin Voting)

$C = \{a, b, c, d\}$ and $V = (v_1, v_2, v_3, v_4, v_5)$, so $maj(V) = 3$

$v_1 :\quad b\ c\ a\ d$

$v_2 :\quad c\ d\ a\ b$

$v_3 :\quad a\ d\ c\ b$

$v_4 :\quad c\ a\ d\ b$

$v_5 :\quad b\ d\ c\ a$

Bucklin Voting (BV)

Example (Bucklin Voting)

$C = \{a, b, c, d\}$ and $V = (v_1, v_2, v_3, v_4, v_5)$, so $maj(V) = 3$

v_1 : b c a d

v_2 : c d a b

v_3 : a d c b

v_4 : c a d b

v_5 : b d c a

	a	b	c	d
score^1	1	2	2	0

Bucklin Voting (BV)

Example (Bucklin Voting)

$C = \{a, b, c, d\}$ and $V = (v_1, v_2, v_3, v_4, v_5)$, so $maj(V) = 3$

v_1 : $b\ c\ a\ d$

v_2 : $c\ d\ a\ b$

v_3 : $a\ d\ c\ b$

v_4 : $c\ a\ d\ b$

v_5 : $b\ d\ c\ a$

	a	b	c	d
score^1	1	2	2	0
score^2	2	2	3	3

Bucklin Voting (BV)

Example (Bucklin Voting)

$C = \{a, b, c, d\}$ and $V = (v_1, v_2, v_3, v_4, v_5)$, so $maj(V) = 3$

v_1 : $b\ c\ a\ d$

v_2 : $c\ d\ a\ b$

v_3 : $a\ d\ c\ b$

v_4 : $c\ a\ d\ b$

v_5 : $b\ d\ c\ a$

	a	b	c	d
score^1	1	2	2	0
score^2	2	2	3	3

$\Rightarrow$ c and d are level 2 Bucklin winners in (C, V)

Control by Partition of Voters

Example (BV)

$C = \{a, b, c, d, e\}$, $V = (v_1, \ldots, v_5)$, $V_1 = (v_1, v_2)$, $V_2 = (v_3, v_4, v_5)$

$$
\begin{array}{ll}
 & (C, V) \\
v_1 & b\,a\,c\,d\,e \\
v_2 & b\,d\,c\,a\,e \\
v_3 & c\,a\,d\,b\,e \\
v_4 & a\,d\,c\,b\,e \\
v_5 & c\,e\,b\,a\,d \\
 & \to a
\end{array}
$$

Control by Partition of Voters

1^{st} stage: (C, V_1) (C, V_2)

Example (BV)

$C = \{a, b, c, d, e\},\ V = (v_1, \ldots, v_5),\ V_1 = (v_1, v_2),\ V_2 = (v_3, v_4, v_5)$

	(C, V)	(C, V_1)	(C, V_2)
v_1	$b\,a\,c\,d\,e$	$b\,a\,c\,d\,e$	
v_2	$b\,d\,c\,a\,e$	$b\,d\,c\,a\,e$	
v_3	$c\,a\,d\,b\,e$		$c\,a\,d\,b\,e$
v_4	$a\,d\,c\,b\,e$		$a\,d\,c\,b\,e$
v_5	$c\,e\,b\,a\,d$		$c\,e\,b\,a\,d$
	$\rightarrow a$		

Control by Partition of Voters

1^{st} stage: (C, V_1) (C, V_2)

W_1 W_2

Example (BV)

$C = \{a, b, c, d, e\},\ V = (v_1, \ldots, v_5),\ V_1 = (v_1, v_2),\ V_2 = (v_3, v_4, v_5)$

	(C, V)	(C, V_1)	(C, V_2)
v_1	$b\,a\,c\,d\,e$	$b\,a\,c\,d\,e$	
v_2	$b\,d\,c\,a\,e$	$b\,d\,c\,a\,e$	
v_3	$c\,a\,d\,b\,e$		$c\,a\,d\,b\,e$
v_4	$a\,d\,c\,b\,e$		$a\,d\,c\,b\,e$
v_5	$c\,e\,b\,a\,d$		$c\,e\,b\,a\,d$
	$\rightarrow a$	$W_1 = \{b\}$	$W_2 = \{c\}$

Control by Partition of Voters

1^{st} stage: (C, V_1) $\quad$ (C, V_2)

W_1 $\quad$ W_2

2^{nd} stage: $(W_1 \cup W_2, V)$

Example (BV)

$C = \{a, b, c, d, e\}$, $V = (v_1, \ldots, v_5)$, $V_1 = (v_1, v_2)$, $V_2 = (v_3, v_4, v_5)$

	(C, V)	(C, V_1)	(C, V_2)	$(W_1 \cup W_2, V)$
v_1	$b\,a\,c\,d\,e$	$b\,a\,c\,d\,e$		$b\,c$
v_2	$b\,d\,c\,a\,e$	$b\,d\,c\,a\,e$		$b\,c$
v_3	$c\,a\,d\,b\,e$		$c\,a\,d\,b\,e$	$c\,b$
v_4	$a\,d\,c\,b\,e$		$a\,d\,c\,b\,e$	$c\,b$
v_5	$c\,e\,b\,a\,d$		$c\,e\,b\,a\,d$	$c\,b$
	$\rightarrow a$	$W_1 = \{b\}$	$W_2 = \{c\}$	

Control by Partition of Voters

1^{st} stage: (C, V_1) (C, V_2)

2^{nd} stage: $(W_1 \cup W_2, V)$

$$\begin{tikzcd} (C, V_1) \arrow[dr, dash, "W_1"'] & & (C, V_2) \arrow[dl, dash, "W_2"] \\ & (W_1 \cup W_2, V) & \end{tikzcd}$$

Example (BV)

$C = \{a, b, c, d, e\}$, $V = (v_1, \ldots, v_5)$, $V_1 = (v_1, v_2)$, $V_2 = (v_3, v_4, v_5)$

	(C, V)	(C, V_1)	(C, V_2)	$(W_1 \cup W_2, V)$
v_1	$b\,a\,c\,d\,e$	$b\,a\,c\,d\,e$		$b\,c$
v_2	$b\,d\,c\,a\,e$	$b\,d\,c\,a\,e$		$b\,c$
v_3	$c\,a\,d\,b\,e$		$c\,a\,d\,b\,e$	$c\,b$
v_4	$a\,d\,c\,b\,e$		$a\,d\,c\,b\,e$	$c\,b$
v_5	$c\,e\,b\,a\,d$		$c\,e\,b\,a\,d$	$c\,b$
	$\rightarrow a$	$W_1 = \{b\}$	$W_2 = \{c\}$	$\rightarrow c$

Classical Control Complexity: Overview

voting rule	CAUC		CAC		CDC		CPC-TE		CPC-TP		CRPC-TE		CRPC-TP		CAV		CDV		CPV-TE		CPV-TP	
	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D	C	D
plurality	R	R	R	R	R	R	R	R	R	R	R	R	R	R	V	V	V	V	V	V	R	R
Condorcet	I	V	I	V	V	I	V	I	V	I	V	I	V	I	R	V	R	V	R	V	R	V
approval	I	V	I	V	V	I	V	I	I	I	V	I	I	I	R	V	R	V	R	V	R	V
Copeland$^{\alpha}$																						
for $\alpha = 0$	V	V	R	V	R	V	R	V	R	V	R	V	R	V	R	R	R	R	R	R	R	R
$0 < \alpha < 1$	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	R	R	R	R	R	R	R
$\alpha = 1$	V	V	R	V	R	V	R	V	R	V	R	V	R	V	R	R	R	R	R	R	R	R
SP-AV	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	V	R	V	R	V	R	R
fallback	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	V	R	V	R	R	R	R
Bucklin	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	V	R	V	R	R	R	S
RV	I	V	I	V	V	I	V	I	I	I	V	I	I	I	R	V	R	V	R	V	R	V
NRV	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	V	R	V	R	R	R	R
Schulze	R	S	R	S	R	S	R	V	R	V	R	V	R	V	R	V	R	V	R	R	R	R

Table: The complexity of control problems for various voting rules. Key: "I" means immunity, "S" susceptibility, "V" vulnerability, and "R" resistance.

Parameterized Control Complexity: Overview

	Plurality	Condorcet	Maximin	Copeland$^{\alpha}$	BV/FV
CCAC	W[2]-hard	P	W[2]-hard	W[2]-comp	W[2]-hard
CCDC	W[2]-hard	P	?	W[2]-comp	W[2]-hard
CCAV	P	W[1]-hard	W[1]-hard	?	W[2]-hard
CCDV	P	W[2]-comp	W[1]-hard	?	W[2]-hard
DCAC	W[2]-hard	P	?	P	W[2]-hard
DCDC	W[1]-hard	P	?	P	W[2]-hard
DCAV	P	P	W[1]-hard	?	P
DCDV	P	P	W[1]-hard	?	P

Table: Overview of classical and parameterized complexity of control problems. All W-hardness results are w.r.t. the output parameter.

Parameterized Control Complexity: Any FPT Results?

Parameterized Control Complexity: Any FPT Results?

- Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR*, 2009) give some FPT results for control by adding/deleting candidates/voters in Copeland$^\alpha$ obtained via ILPs w.r.t. the parameters:
  - $m = \#$ of candidates
  - $n = \#$ of votes

Parameterized Control Complexity: Any FPT Results?

- Faliszewski, Hemaspaandra, Hemaspaandra, Rothe (*JAIR*, 2009) give some FPT results for control by adding/deleting candidates/voters in Copeland$^{\alpha}$ obtained via ILPs w.r.t. the parameters:
  - $m = \#$ of candidates
  - $n = \#$ of votes
- Wang, Yang, Guo, Feng, Chen (*COCOA-2013*) show that, w.r.t. the parameter $d = \#$ of deleted votes, k-Approval-CCDV is
  - $\mathrm{W}[2]$-hard for unbounded k, yet
  - FPT with a polynomial problem kernel for constant k.

Research Challenge 5: Kernelization Complexity

Research Challenge 5

- What is the kernelization complexity of FPT voting problems w.r.t.
  - the number m of candidates,
  - the number n of votes, or
  - some parameter less than m or n?

Research Challenge 5: Kernelization Complexity

Research Challenge 5

- What is the kernelization complexity of FPT voting problems w.r.t.
  - the number m of candidates,
  - the number n of votes, or
  - some parameter less than m or n?
- Can one find polynomial (or even linear) problem kernels for these parameters?

Single-Peaked Elections: Example

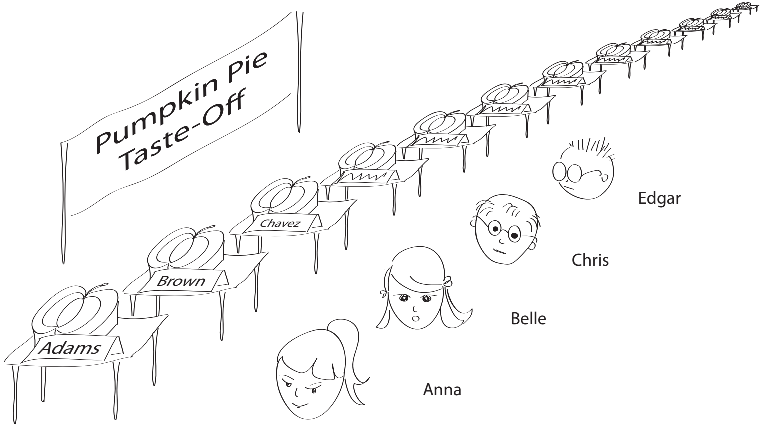


Figure: The annual charity Pumpkin Pie Taste-Off

Single-Peaked Elections: Example

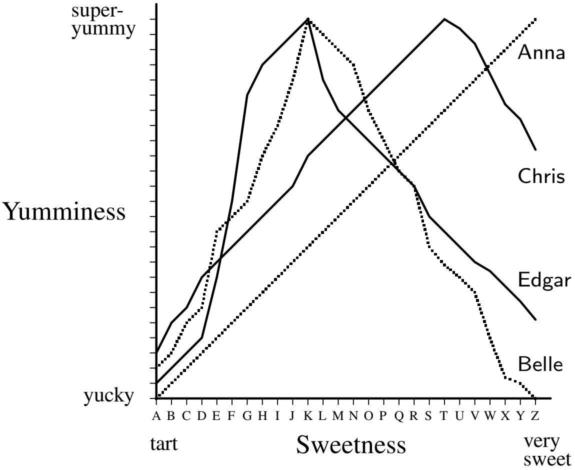


Figure: Preferences regarding sweetness of pumpkin pie

Single-Peaked Elections Generalized

- Yang and Guo (*arXiv*, 2013) consider k-peaked elections, where each voter's preference can have up to k peaks.

Single-Peaked Elections Generalized

- Yang and Guo (*arXiv*, 2013) consider k-peaked elections, where each voter's preference can have up to k peaks.
- One can similarly generalize
  - single-crossing elections:
    The voters can be linearly ordered such that along this order, for each pair of candidates,
    - either all voters agree on the ordering of these two candidates
    - or there is a single crossing point where the voters switch from preferring one candidate to the other.

Single-Peaked Elections Generalized

- Yang and Guo (*arXiv*, 2013) consider k-peaked elections, where each voter's preference can have up to k peaks.
- One can similarly generalize
  - single-crossing elections:
    The voters can be linearly ordered such that along this order, for each pair of candidates,
    - either all voters agree on the ordering of these two candidates
    - or there is a single crossing point where the voters switch from preferring one candidate to the other.
  - one-dimensional Euclidean elections:
    Candidates and voters can be embedded into $\mathbb{R}$ such that each voter prefers the closer one among any pair of candidates.

Research Challenge 6: Single-peaked $\Rightarrow$ k-peaked

Research Challenge 6

How does the complexity of standard voting problems depend on the parameter k in

- k-peaked
- k-crossing
- k-dimensional Euclidean

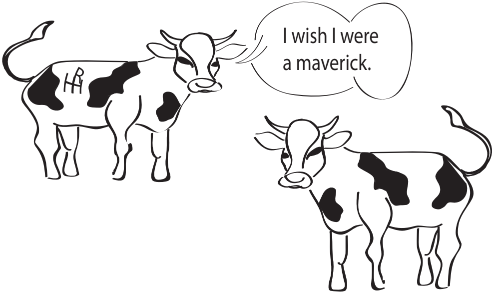
elections?

Nearly Single-Peaked Elections

- There are various measures for "nearness to single-peakedness". Elections can be made single-peaked by
  - deleting k voters

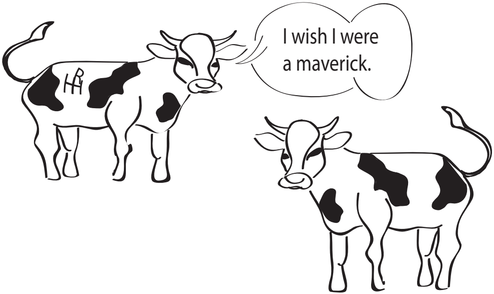
Nearly Single-Peaked Elections

- There are various measures for "nearness to single-peakedness". Elections can be made single-peaked by
  - deleting k voters (a.k.a. **maverick voters**)



Nearly Single-Peaked Elections

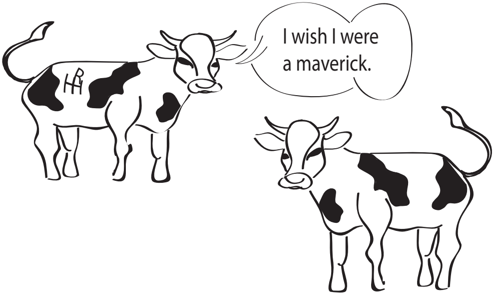
- There are various measures for "nearness to single-peakedness".
  Elections can be made single-peaked by
  - deleting k voters (a.k.a. **maverick voters**)



  - deleting k candidates

Nearly Single-Peaked Elections

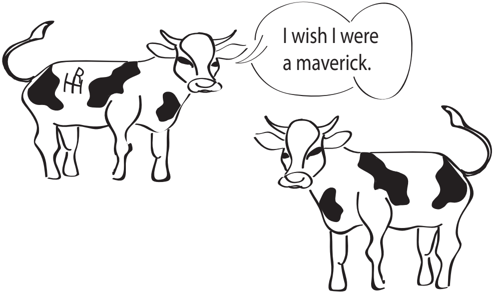
- There are various measures for "nearness to single-peakedness". Elections can be made single-peaked by
  - deleting k voters (a.k.a. **maverick voters**)



  - deleting k candidates
  - k swaps in the preferences of each voter

Nearly Single-Peaked Elections

- There are various measures for "nearness to single-peakedness". Elections can be made single-peaked by
  - deleting k voters (a.k.a. **maverick voters**)



  - deleting k candidates
  - k swaps in the preferences of each voter
  - contracting groups of up to k candidates (showing up as a block in each vote) into a single candidate

Research Challenge 7: Nearness to Single-peakedness

Research Challenge 7

How can one use such "nearness to single-peakedness" parameters to obtain FPT results for NP-hard voting problems?

No Time for Other COMSOC Problems

No Time for Other COMSOC Problems

Questions?