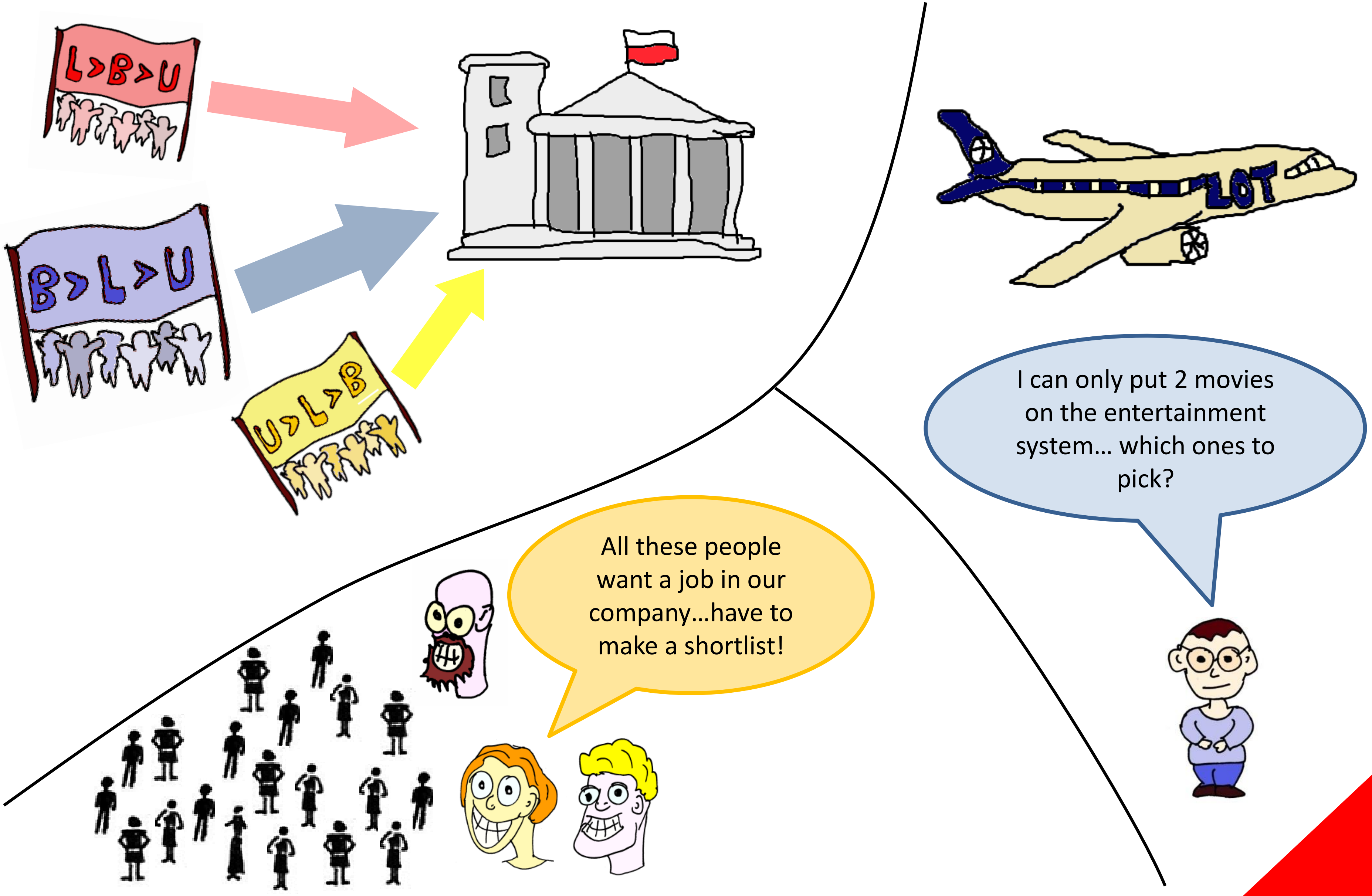
Multiwinner Elections:
Theory and Experiments

Piotr Faliszewski
AGH University
Kraków, Poland

Based on joint work with Edith Elkind (University of Oxford), Jerome Lang (Universite Paris Dauphine), Piotr Skowron (Uniwersytet Warszawski & Google Polska), Arkadii Slinko (University of Auckland), Lan Yu (Google Inc.), Robert Schaefer (AGH), and Nimrod Talmon (TU Berlin, Niemcy)

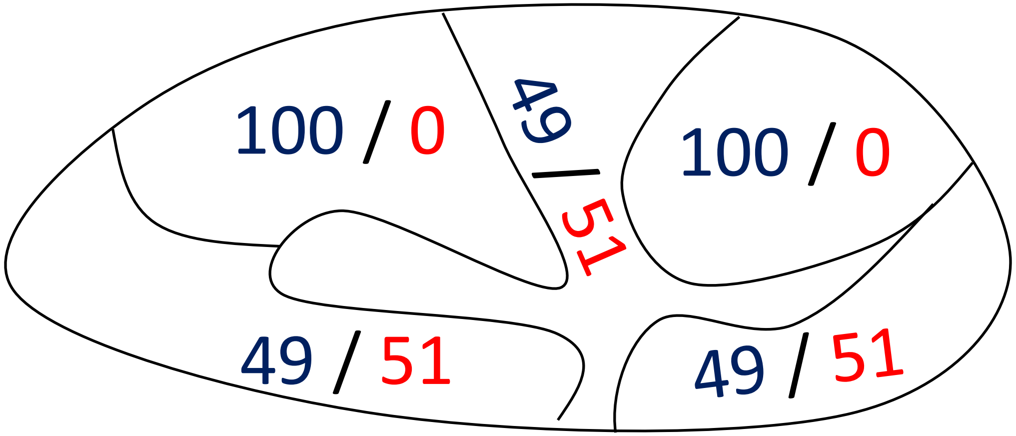
Supported by NCN grant 2012/06/M/ST1/00358

Multiwinner Elections?



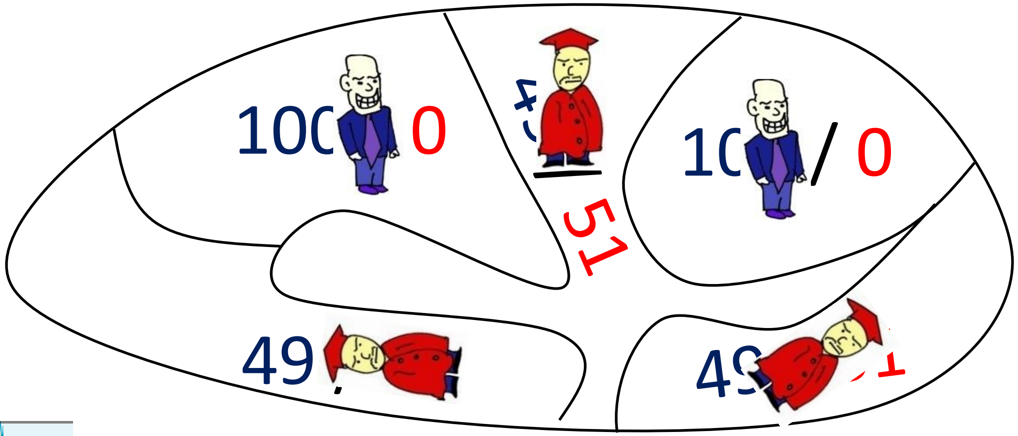
How to choose a parliament?

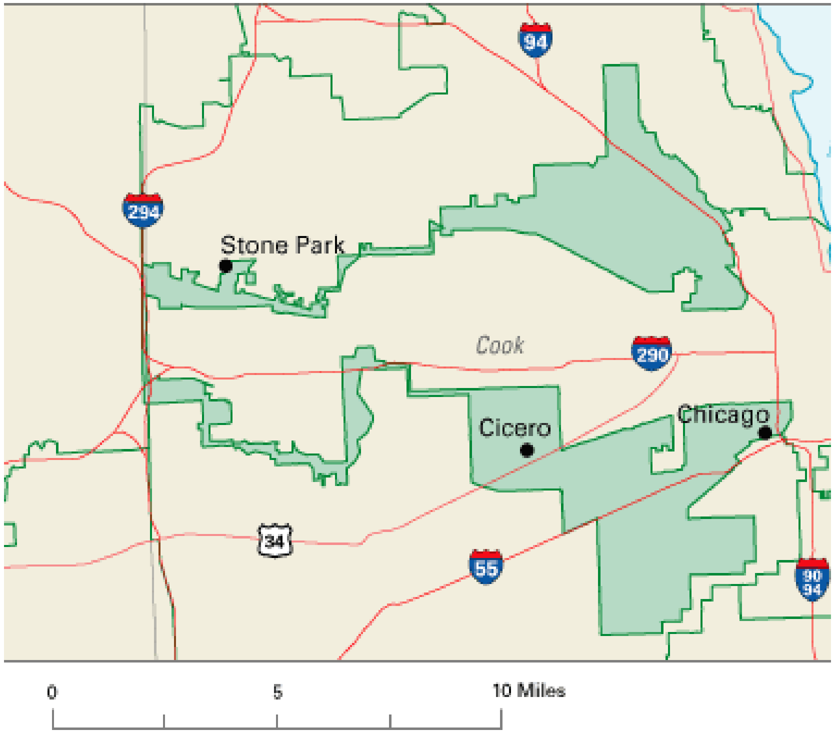
Single-winner districts



How to choose a parliament?

Single-winner districts

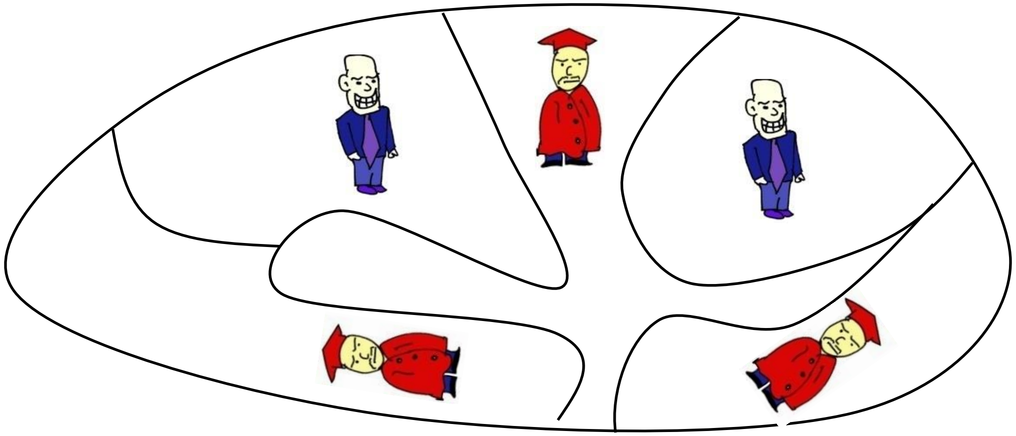




25% support sufficient to form a majority government

How to choose a parliament?

Single-winner districts



Party lists

25% support sufficient to form a majority government

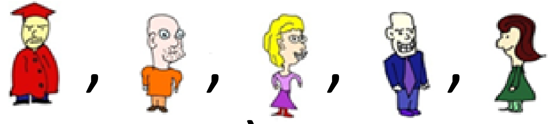
Our focus: A single multiwinner distrct

Agenda

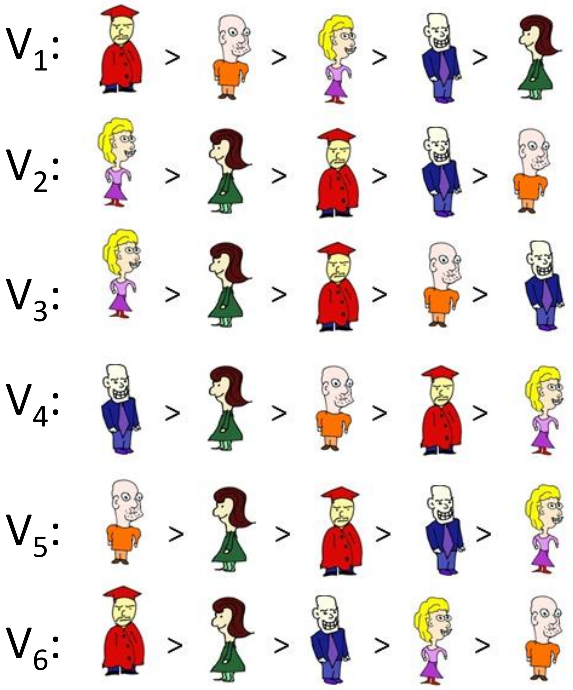
1. Introduction
2. Multiwinner elections
   - Election model
   - Basic rules and how they work
3. Committee scoring rules
   - Analogues of single-winner scoring rules
   - Important subclasses of CSRs
   - Complexity results
   - Example of an axiomatic approach
4. Conclusions

Election Model

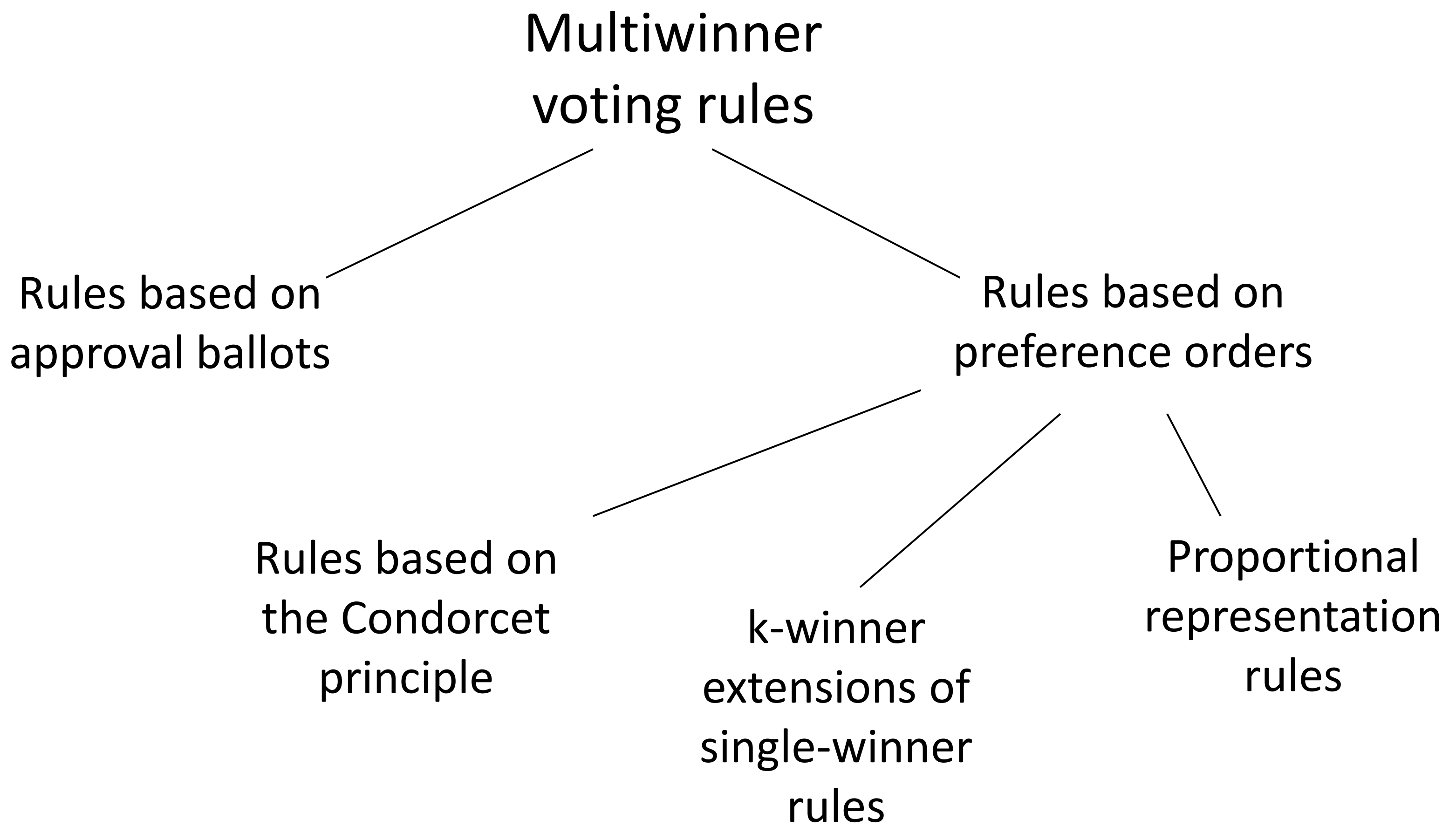
- Election E = (C, V)
  - C – set of candidates
  - V – set of voters
- Parameter k
  - k – the committee size
- … and a voting rule…

C = {  }

$V = (v_1, \dots , v_6)$



Main Families of Multiwinner Rules



Multiwinner voting rules

Rules based on approval ballots

Rules based on preference orders

Rules based on the Condorcet principle

k-winner extensions of single-winner rules

Proportional representation rules

Election Model

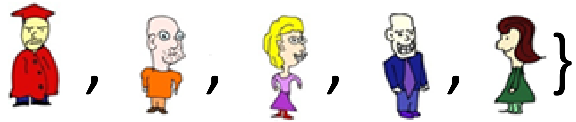
- Election E = (C, V)
  - C – set of candidates
  - V – set of voters

- Parameter k
  - k – the committee size

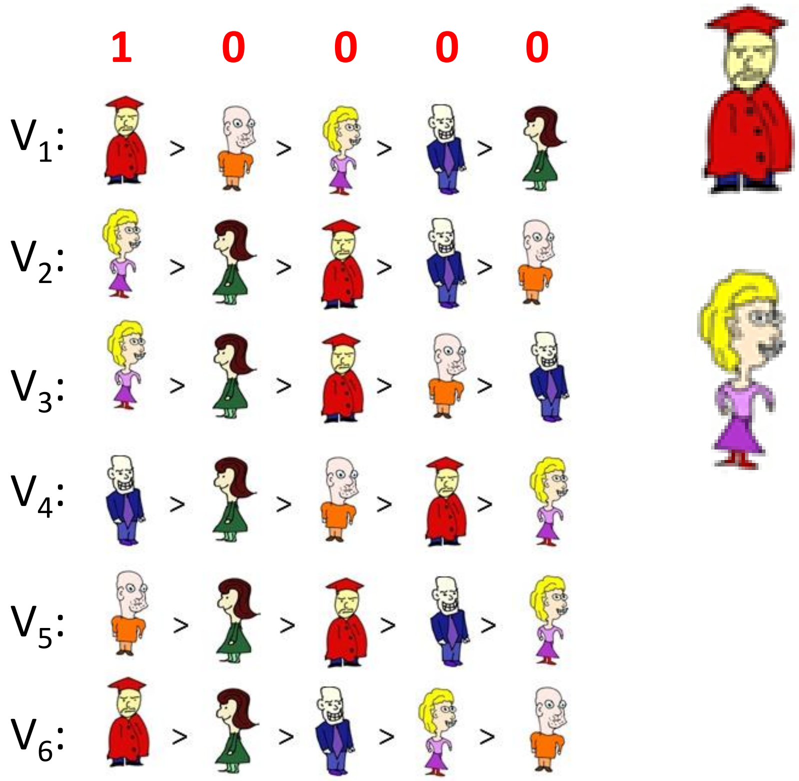
- … and a voting rule…

SNTV

$C = \{ \; , \; , \; , \; , \; \}$



$V = (v_1, \dots , v_6)$



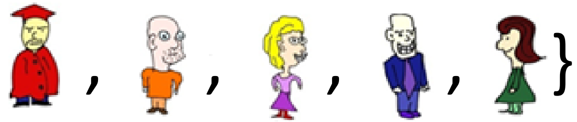
Election Model

- Election E = (C, V)
  - C – set of candidates
  - V – set of voters

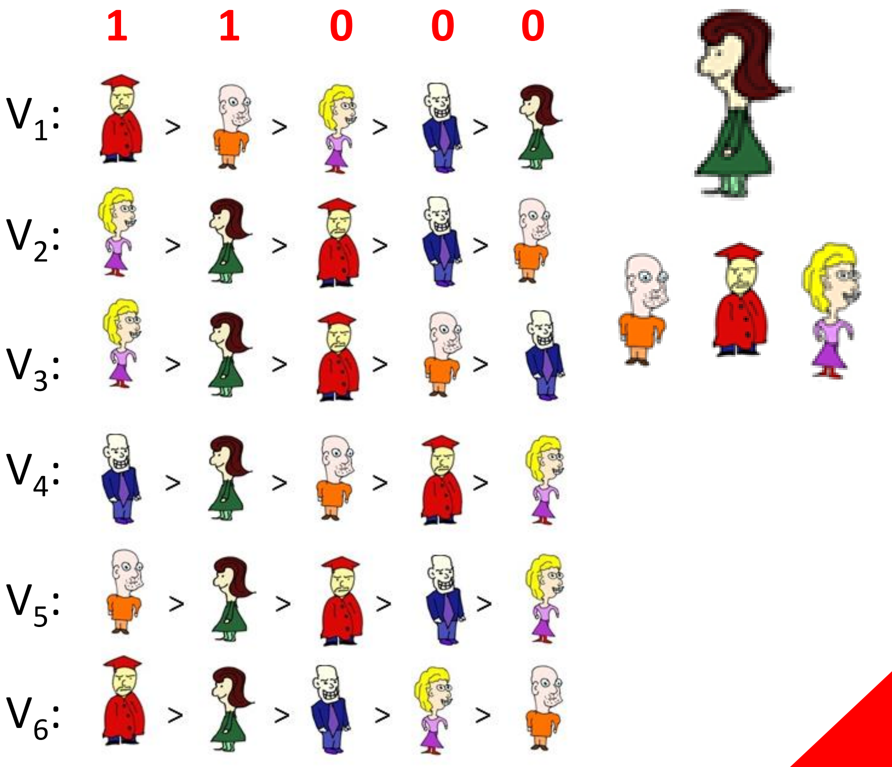
- Parameter k
  - k – the committee size

- … and a voting rule…

Bloc

C = {  }

V = (v_1, … , v_6)



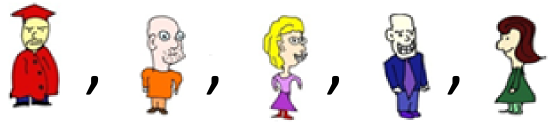
Election Model

- Election $E = (C, V)$
  - C – set of candidates
  - V – set of voters

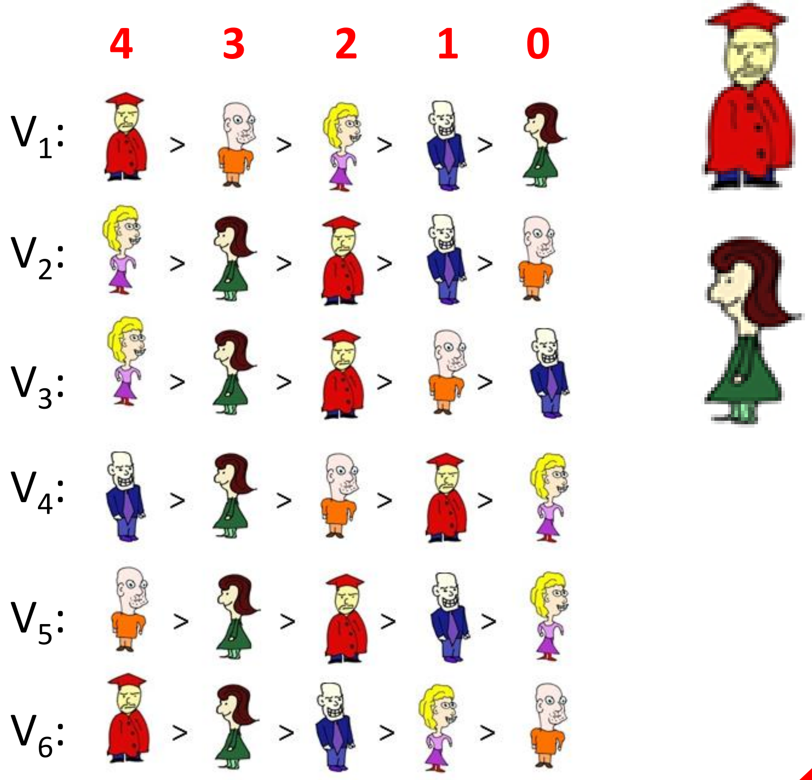
- Parameter k
  - k – the committee size

- … and a voting rule…

k-Borda

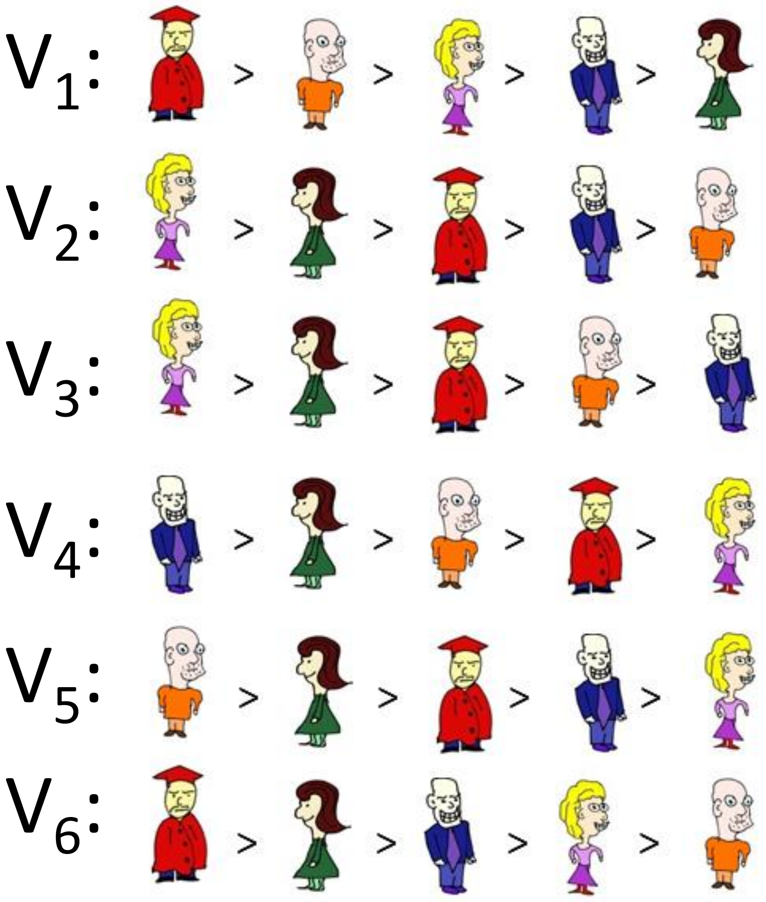
$C = \{$  $\}$

$V = (v_1, \dots, v_6)$



Proportional Representation Rule of Chamberlin—Courant

Choosing a parliament is a resource allocation problems

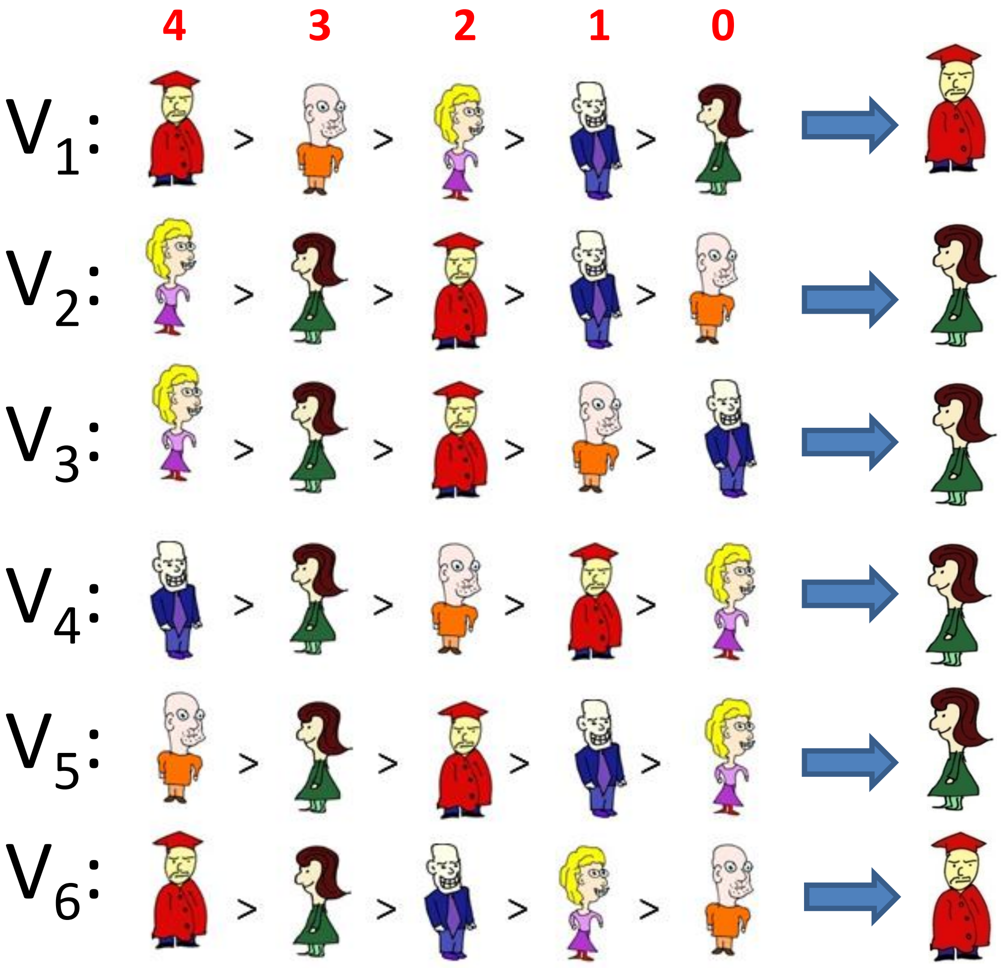


Candidates = Resources

Voting rule assigns candidates to the voters

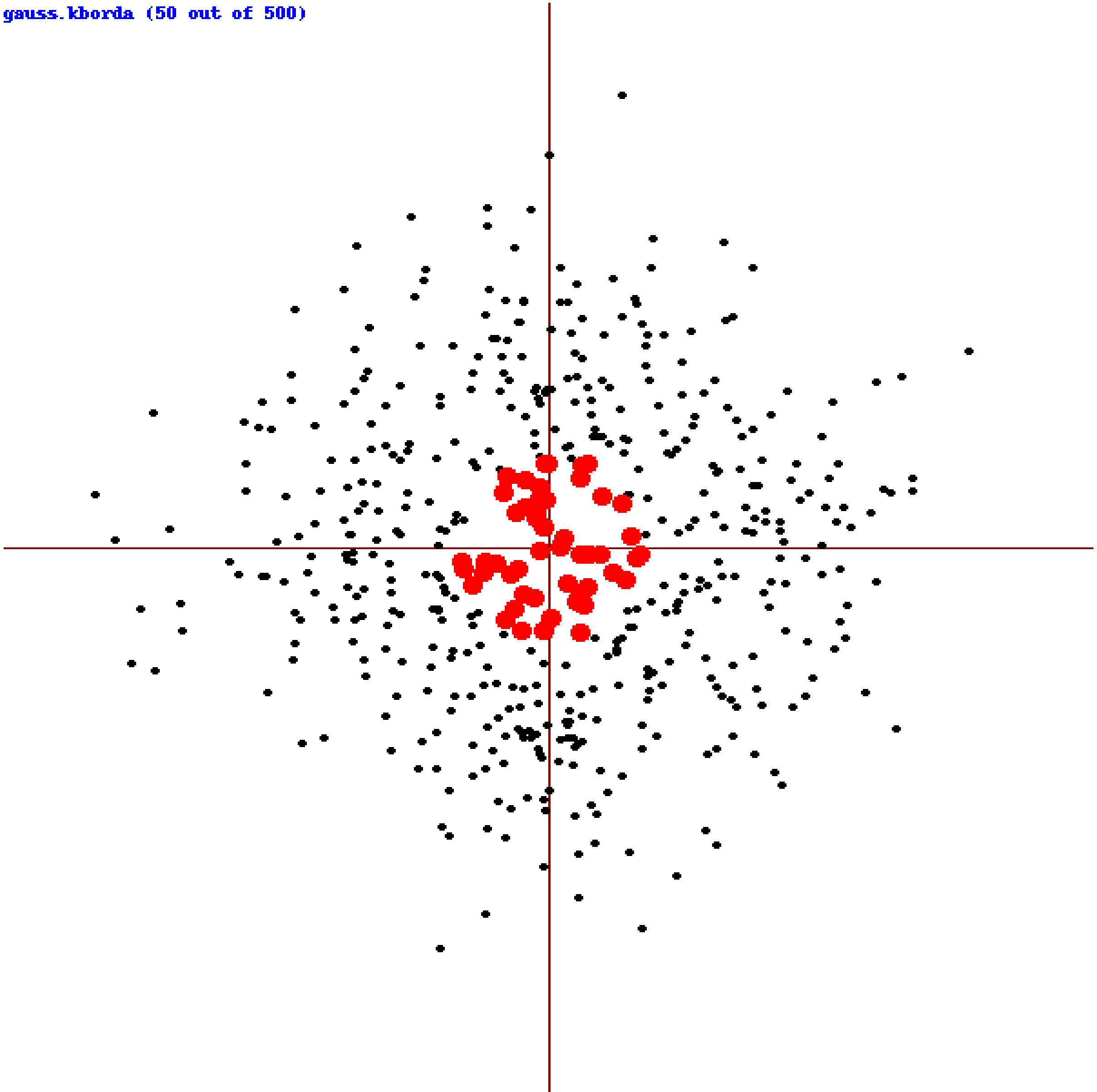
Proportional Representation Rule of Chamberlin—Courant

Choosing a parliament is a resource allocation problems

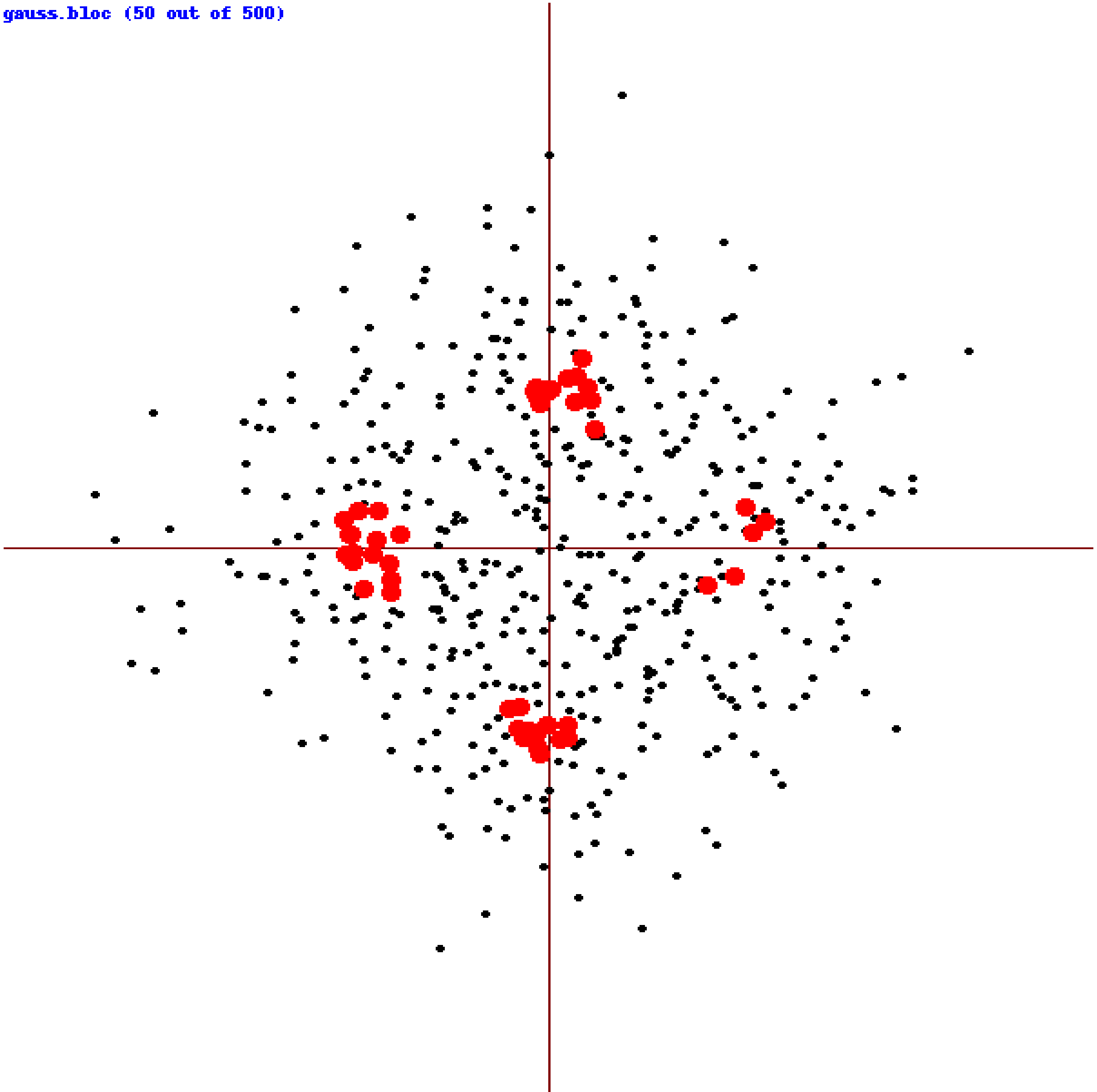


Chamberlin-Courant
Pick k candidates and assign them to the voters to maximze the score that the voters give to their representatives

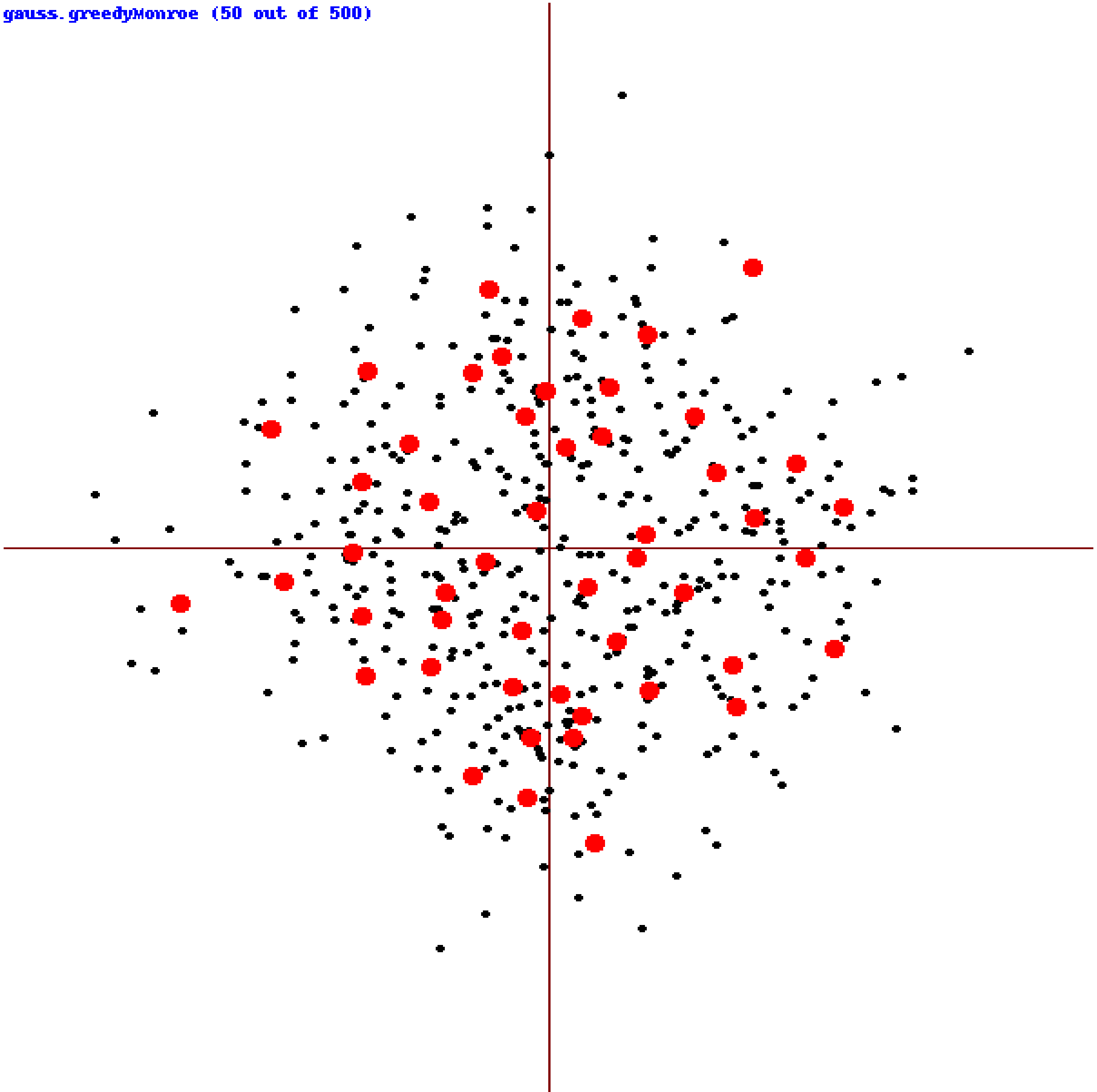
How Do These Rules Work: **k-Borda**



How Do These Rules Work: **Bloc**



How Do These Rules Work: **Chamberlin-Courant**



Single-Winner Scoring Rules

A single-winner scoring function:

f(i) = score for position i

The candidate with the highest sum of scores is the winner

Examples:

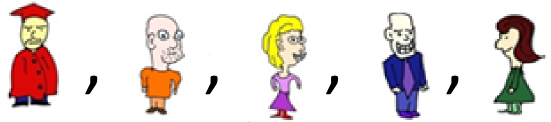
Borda score

$B(i) = m-i$

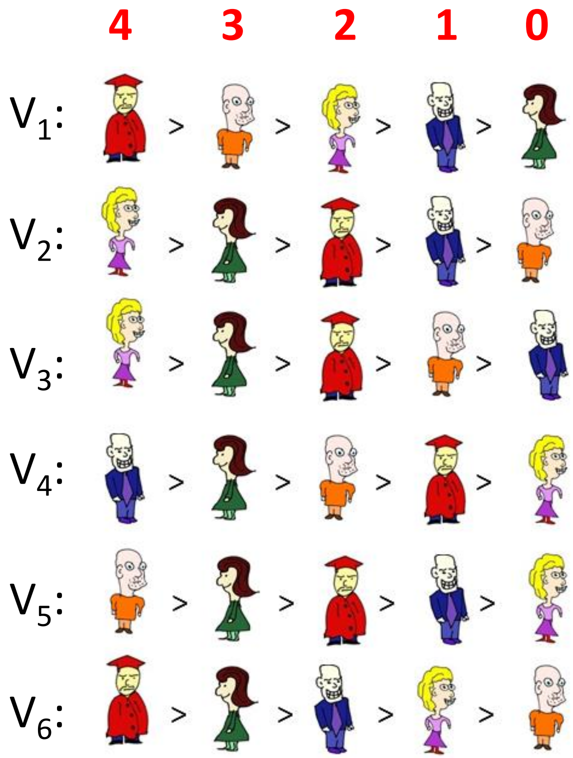
t-Approval score

$A_t(i) = 1$ if $i \leq t$ and 0 otherwise

C = { , , , , }

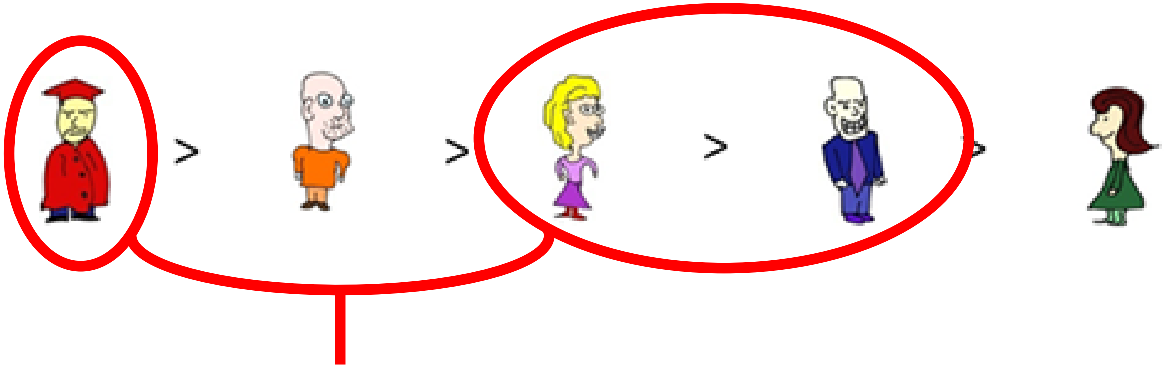


$V = (v_1, \dots, v_6)$



Committee Scoring Rules

Consider a preference order:



winning committee

Position of the winning committee = (1, 3, 4)

$f(i_1, i_2, \ldots, i_k)$ = the score of the committee

Assuming $i_1 < i_2 < \ldots < i_k$

Committee Scoring Rules

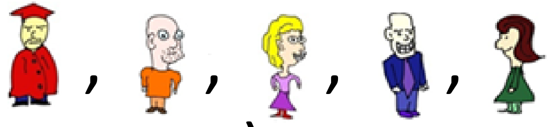
Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

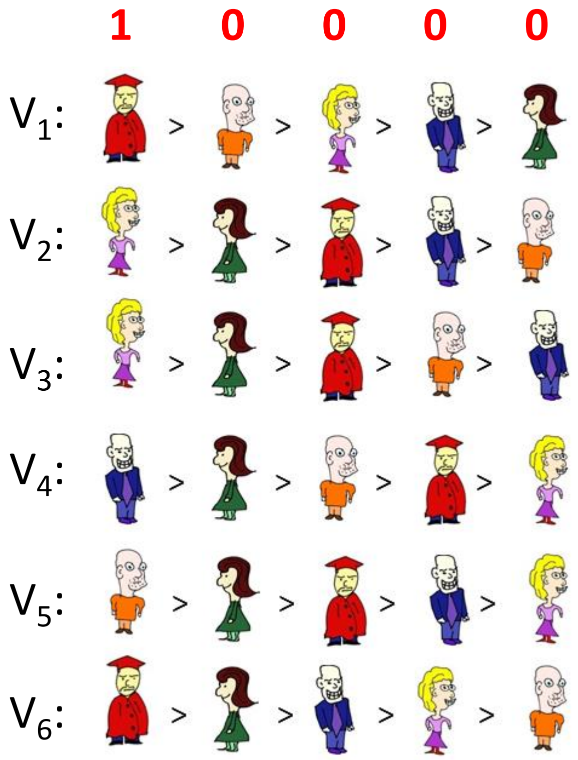
The committee with the highest sum of scores is the winner

Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$C = \{$  $\}$

$V = (v_1, \ldots, v_6)$



Committee Scoring Rules

Committee scoring function:

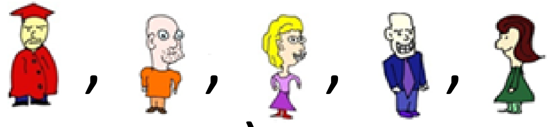
$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

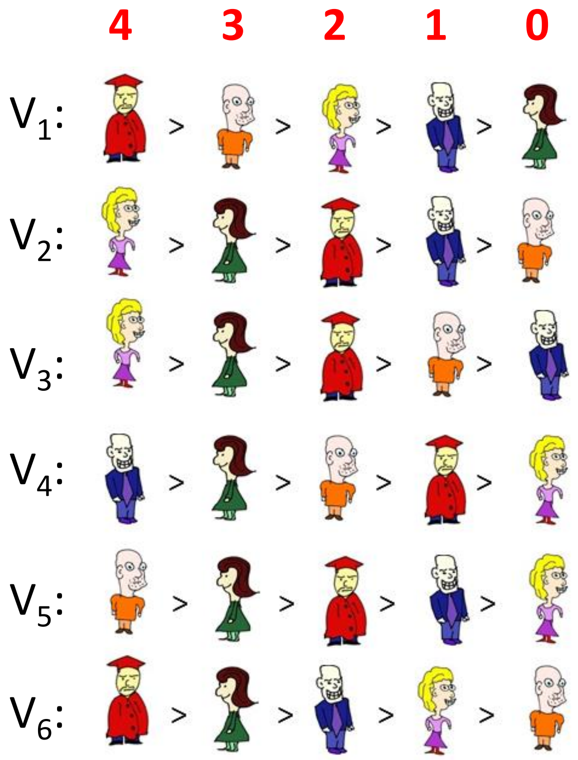
Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-}Borda}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$C = \{$  $\}$

$V = (v_1, \ldots, v_6)$



Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

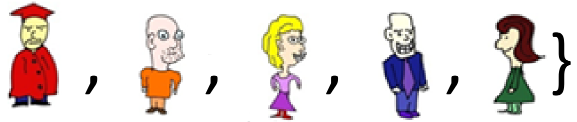
Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-}Borda}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

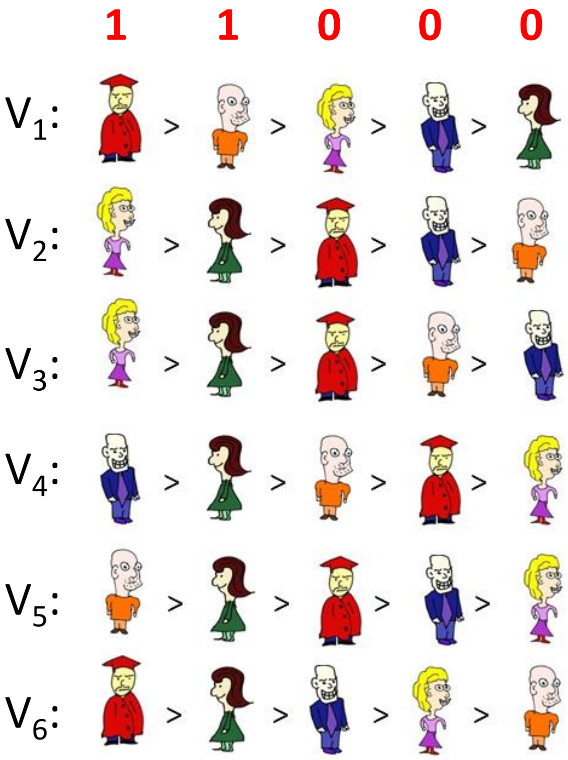
$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

C = {



}

$V = (v_1, \ldots, v_6)$



Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \dots, i_k)$ = score for pos. $(i_1, i_2, \dots, i_k)$

The committee with the highest sum of scores is the winner

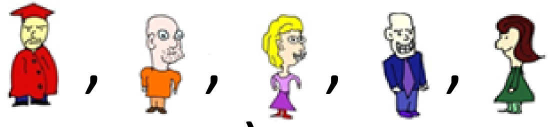
Examples:

$f_{SNTV}(i_1, i_2, \dots, i_k) = A_1(i_1) + \dots + A_1(i_k)$

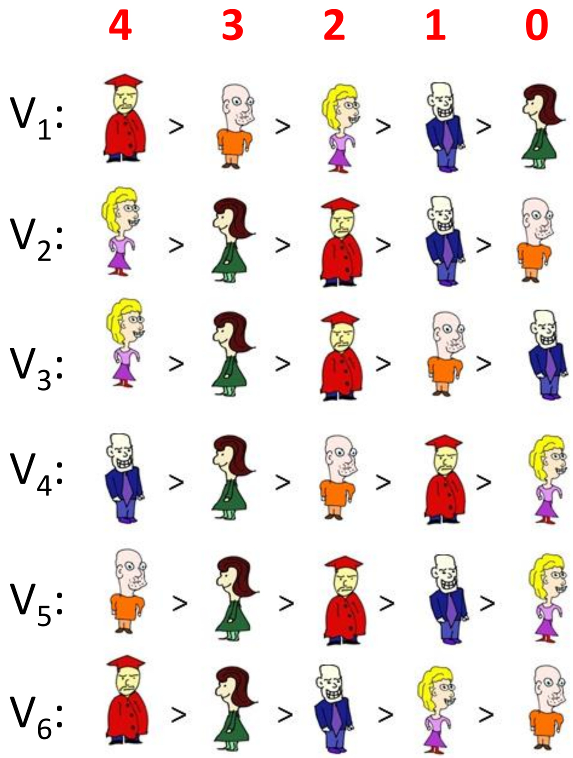
$f_{k\text{-Borda}}(i_1, i_2, \dots, i_k) = B(i_1) + \dots + B(i_k)$

$f_{Bloc}(i_1, i_2, \dots, i_k) = A_k(i_1) + \dots + A_k(i_k)$

$f_{CC}(i_1, i_2, \dots, i_k) = B(i_1)$

C = {  }

$V = (v_1, \dots, v_6)$



Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

Examples:

$f_{\mathrm{SNTV}}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{\mathrm{k\text{-}Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{\mathrm{Bloc}}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{\mathrm{CC}}(i_1, i_2, \ldots, i_k) = B(i_1)$

Basic classes of CSRs

Separable rules:

$f(i_1, i_2, \ldots, i_k) = g(i_1) + \ldots + g(i_k)$

Weakly separable rules:

$f(i_1, i_2, \ldots, i_k) = h_k(i_1) + \ldots + h_k(i_k)$

Representation focused rules:

$f(i_1, i_2, \ldots, i_k) = q(i_1)$

Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-}Borda}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

Basic classes of CSRs

Separable rules:

$f(i_1, i_2, \ldots, i_k) = g(i_1) + \ldots + g(i_k)$

Weakly separable rules:

$f(i_1, i_2, \ldots, i_k) = h_k(i_1) + \ldots + h_k(i_k)$

Representation focused rules:

$f(i_1, i_2, \ldots, i_k) = q(i_1)$

Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

Basic classes of CSRs

Separable rules:

$f(i_1, i_2, \ldots, i_k) = g(i_1) + \ldots + g(i_k)$

Weakly separable rules:

$f(i_1, i_2, \ldots, i_k) = h_k(i_1) + \ldots + h_k(i_k)$

Representation focused rules:

$f(i_1, i_2, \ldots, i_k) = q(i_1)$

Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

Basic classes of CSRs

Separable rules:

$f(i_1, i_2, \ldots, i_k) = g(i_1) + \ldots + g(i_k)$

Weakly separable rules:

$f(i_1, i_2, \ldots, i_k) = h_k(i_1) + \ldots + h_k(i_k)$

Representation focused rules:

$f(i_1, i_2, \ldots, i_k) = q(i_1)$

Committee Scoring Rules

Committee scoring function:

$f(i_1, i_2, \ldots, i_k)$ = score for pos. $(i_1, i_2, \ldots, i_k)$

The committee with the highest sum of scores is the winner

Examples:

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

$f_{k\text{-Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

Basic classes of CSRs

Separable rules:

$f(i_1, i_2, \ldots, i_k) = g(i_1) + \ldots + g(i_k)$

Weakly separable rules:

$f(i_1, i_2, \ldots, i_k) = h_k(i_1) + \ldots + h_k(i_k)$

Representation focused rules:

$f(i_1, i_2, \ldots, i_k) = q(i_1)$

OWA-Based Committee Scoring Rules

An OWA operator is a sequence of k numbers

$$W = (w_1, \dots , w)$$

Given a single-winner scoring rule *g* and OWA opertor *W*, we define CSR:

$$f(i_1, i_2, \dots, i_k) = w_1\, g(i_1) + w_2\, g(i_2) + \dots + w_k\, g(i_k)$$

Examples of OWA-Based Rules

Approval scores

				
	0	0	1	1
	1	1	0	0
	1	0	1	0
	0	1	0	1

Essentially the hardest case.

Most problems are NP-hard and hard to approximate (in an appropriate sense)

Borda scores

				
	0	1	2	3
	2	3	0	1
	2	1	3	0
	1	0	2	3

The „easiest” case

Still NP-hard, but very good approximations possilbe (PTASes in many cases)

└─**e.g., for OWAs with a fixed number of nonzero entries**

Examples of OWA-Based Rules

Special OWA families

Approval scores

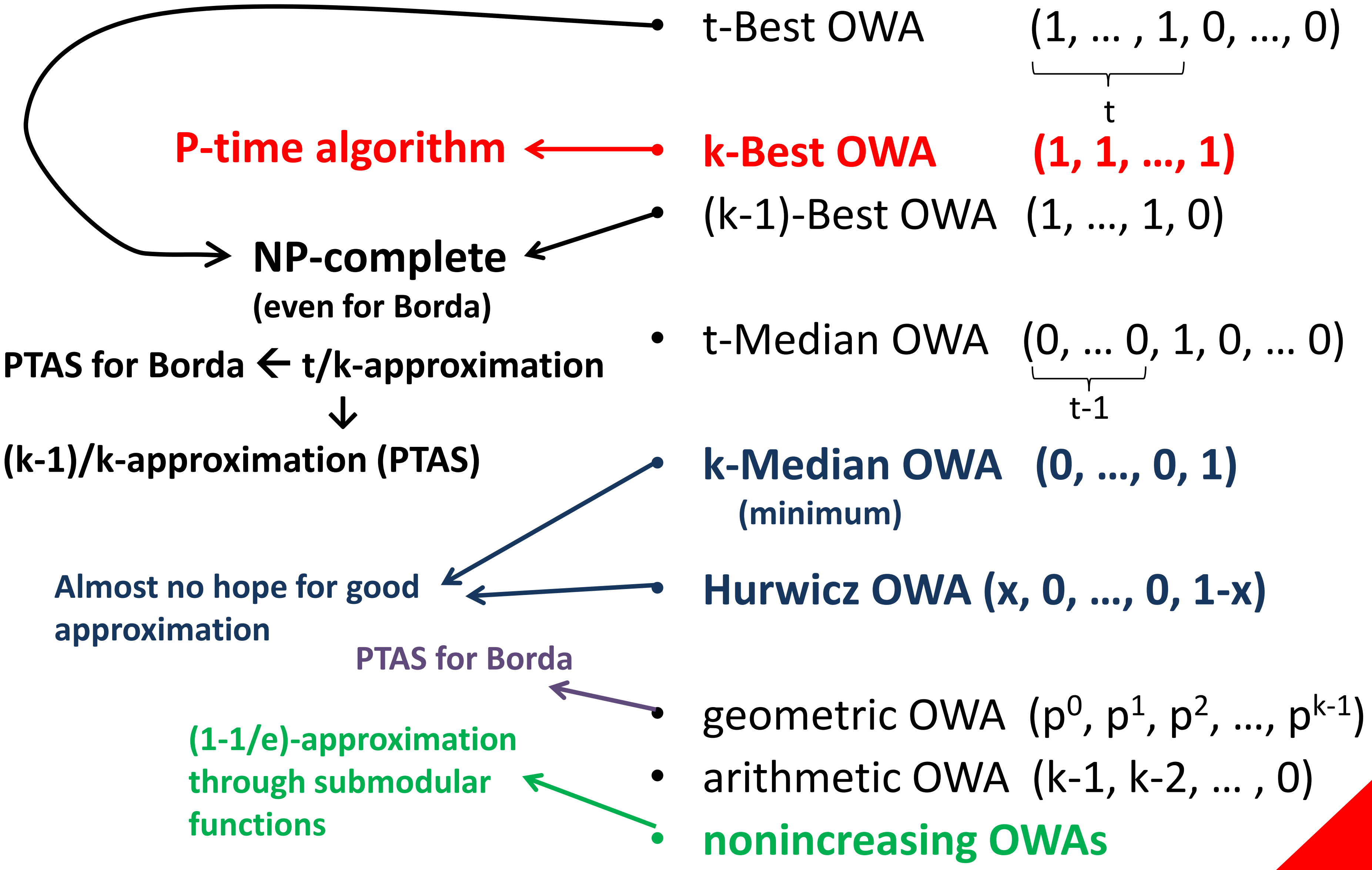
				
	0	0	1	1
	1	1	0	1
	1	0	1	0
	0	0	0	1

Borda scores

				
	0	1	2	3
	2	3	0	1
	2	1	3	0
	1	0	2	3

- t-Best OWA $(\underbrace{1, \dots, 1}_{t}, 0, \dots, 0)$
- **k-Best OWA (1, 1, …, 1)**
- (k-1)-Best OWA (1, …, 1, 0)
- t-Median OWA $(\underbrace{0, \dots 0}_{t-1}, 1, 0, \dots 0)$
- **k-Median OWA (0, …, 0, 1)**
  (minimum)
- **Hurwicz OWA (x, 0, …, 0, 1-x)**
- geometric OWA $(p^0, p^1, p^2, \dots, p^{k-1})$
- arithmetic OWA (k-1, k-2, … , 0)
- **nonincreasing OWAs**

Special OWA families



- t-Best OWA $(\underbrace{1, \dots , 1}_{t}, 0, \dots, 0)$
- **k-Best OWA $(1, 1, \dots, 1)$**
- (k-1)-Best OWA $(1, \dots, 1, 0)$

- t-Median OWA $(\underbrace{0, \dots 0,}_{t-1} 1, 0, \dots 0)$
- **k-Median OWA $(0, \dots, 0, 1)$**
  (minimum)
- **Hurwicz OWA $(x, 0, \dots, 0, 1-x)$**

- geometric OWA $(p^0, p^1, p^2, \dots, p^{k-1})$
- arithmetic OWA $(k-1, k-2, \dots , 0)$
- **nonincreasing OWAs**

P-time algorithm

NP-complete
(even for Borda)

PTAS for Borda ← t/k-approximation
↓
(k-1)/k-approximation (PTAS)

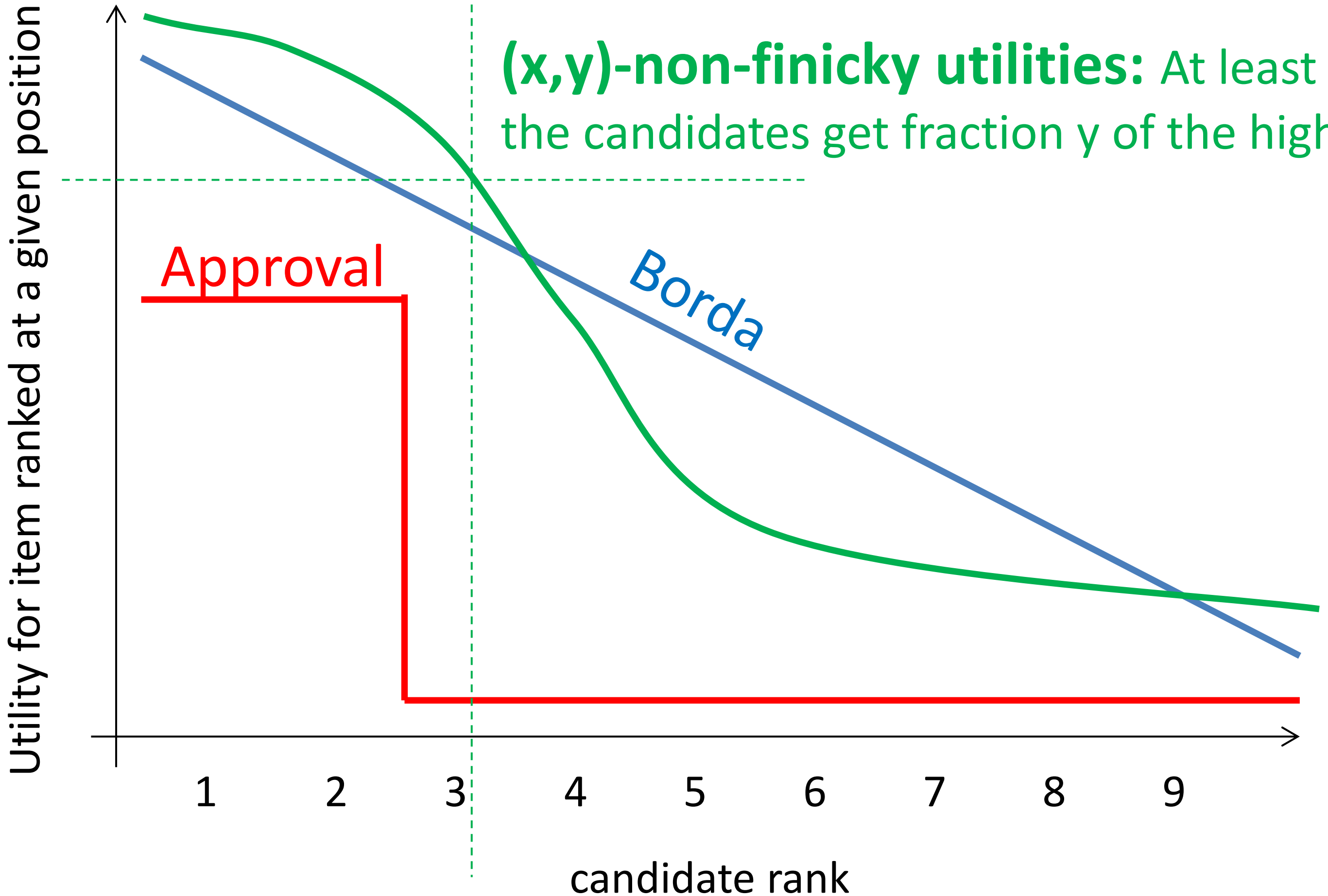
Almost no hope for good approximation

PTAS for Borda

(1-1/e)-approximation through submodular functions

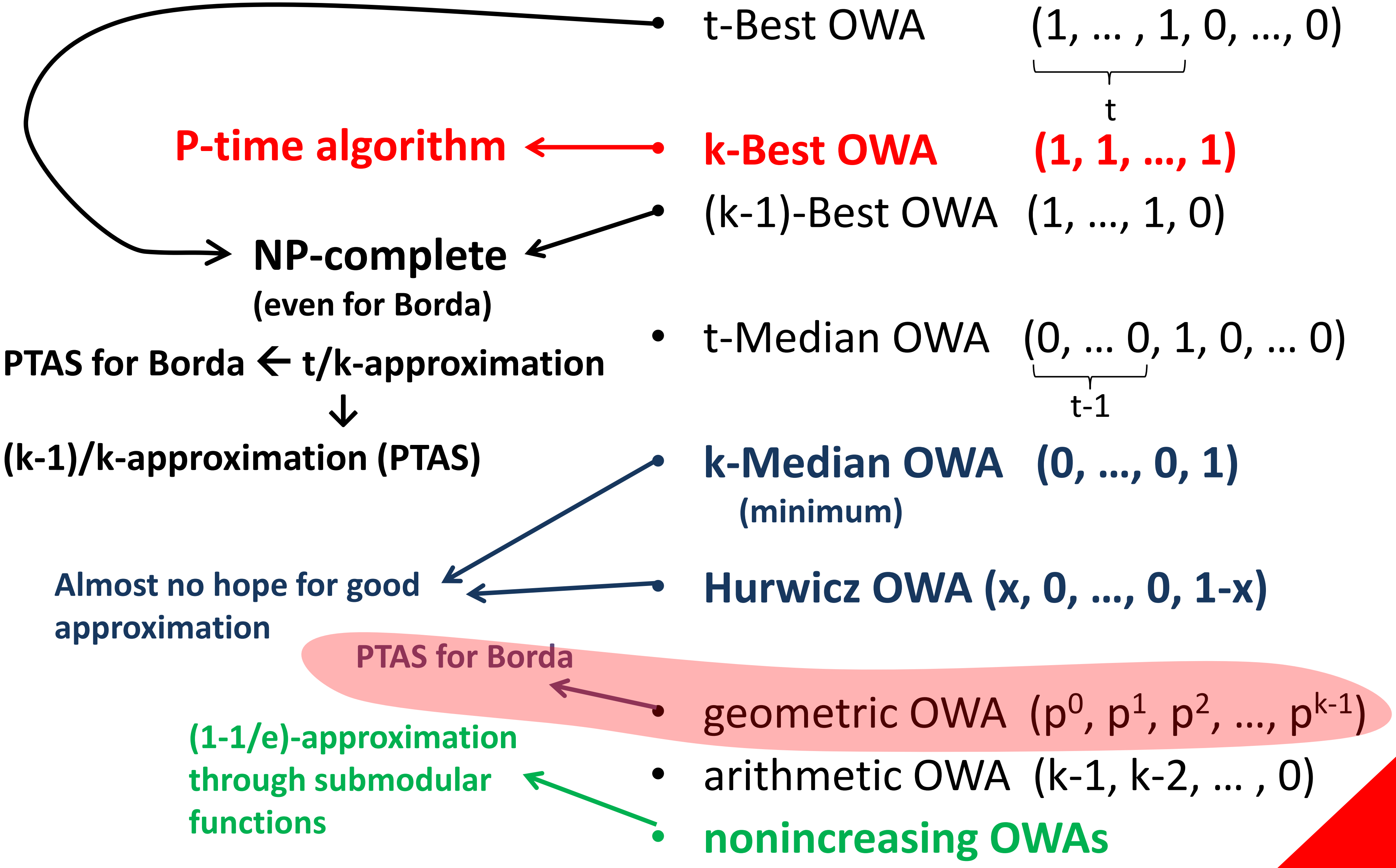
Borda versus Approval

Why is Borda easier to deal with than approval?



(x,y)-non-finicky utilities: At least fraction x of the candidates get fraction y of the highest score

Special OWA families



- t-Best OWA $(\underbrace{1, \dots, 1}_{t}, 0, \dots, 0)$
- **k-Best OWA $(1, 1, \dots, 1)$**
- (k-1)-Best OWA $(1, \dots, 1, 0)$
- t-Median OWA $(\underbrace{0, \dots 0}_{t-1}, 1, 0, \dots 0)$
- **k-Median OWA $(0, \dots, 0, 1)$**
  (minimum)
- **Hurwicz OWA $(x, 0, \dots, 0, 1-x)$**
- geometric OWA $(p^0, p^1, p^2, \dots, p^{k-1})$
- arithmetic OWA $(k-1, k-2, \dots, 0)$
- **nonincreasing OWAs**

P-time algorithm

NP-complete
(even for Borda)

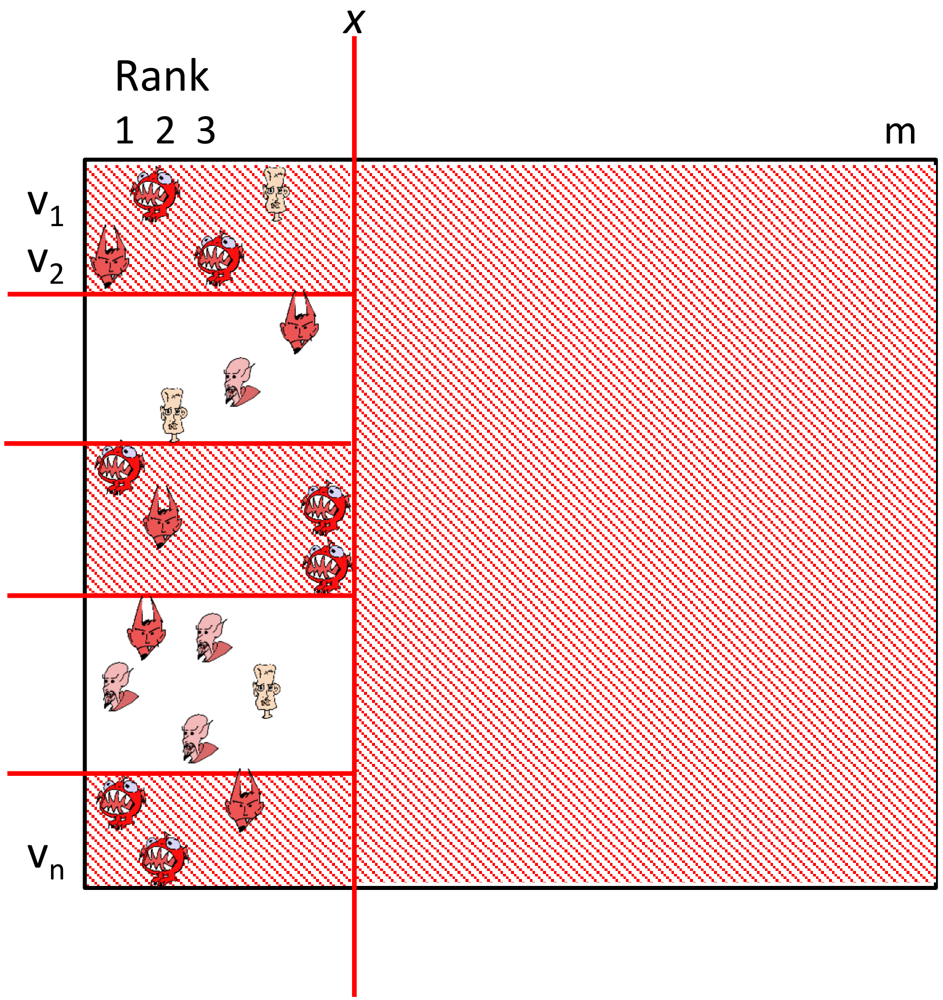
PTAS for Borda ← t/k-approximation
↓
(k-1)/k-approximation (PTAS)

Almost no hope for good approximation

PTAS for Borda

(1-1/e)-approximation through submodular functions

PTAS for Borda Utilities, 1-Best OWA = (1,0, … , 0)*



Goal: pick *K* winners among *m* candidates, to get the highest utility

Initialize: Forget about the whole profile beyond rank *x*:

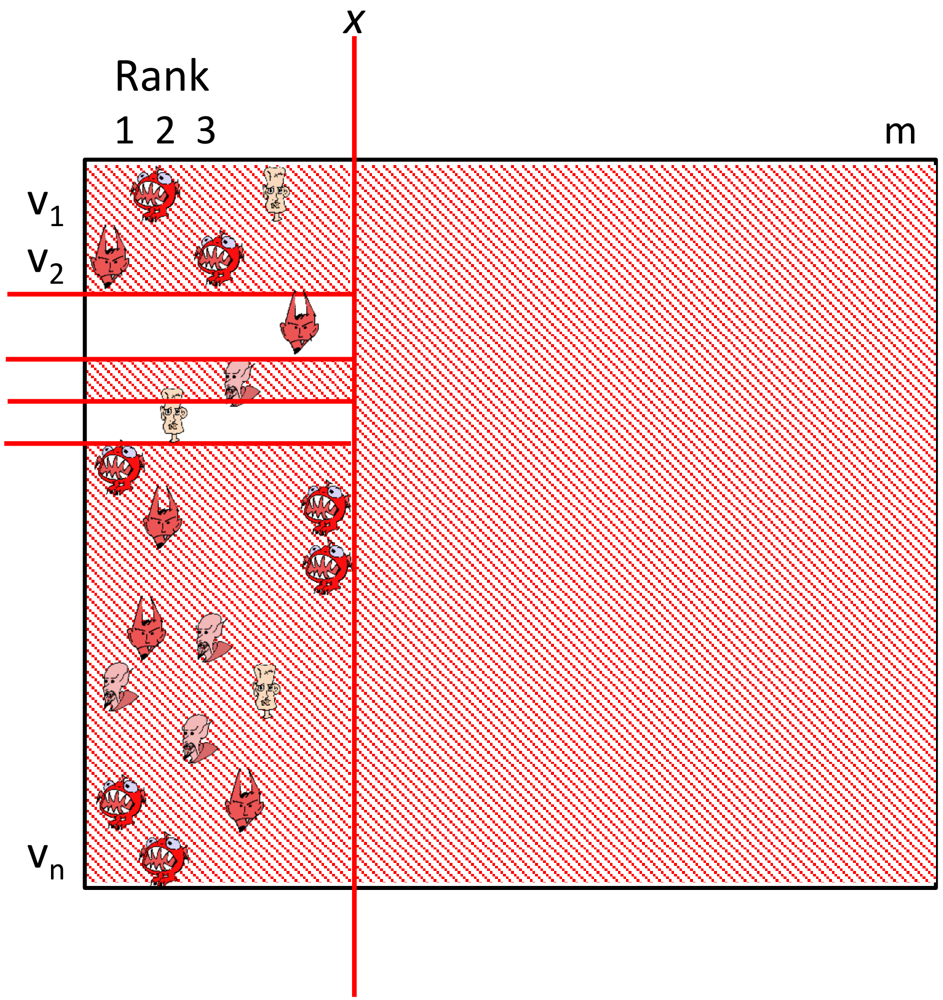
$x = mw(K) / K$ (w(K) is Lambert's W function, O(log K))

Loop: Keep picking the candidate that appears in the „available" part of the profile most frequently.

---

[*] Achieving Fully Proportional Representation: Approximability Results, P. Skowron, P. Faliszewski, A. Slinko, *Artificial Intelligence*, Vol. 222, pp. 67--103, 2015.

PTAS for Borda Utilities, 1-Best OWA = (1,0, … , 0)*

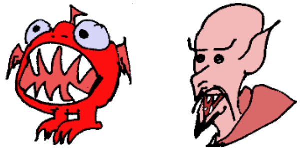


Goal: pick *K* winners among *m* candidates, to get the highest utility

Initialize: Forget about the whole profile beyond rank *x*:

$x = mw(K) / K$ (w(K) is Lambert's W function, O(log K))

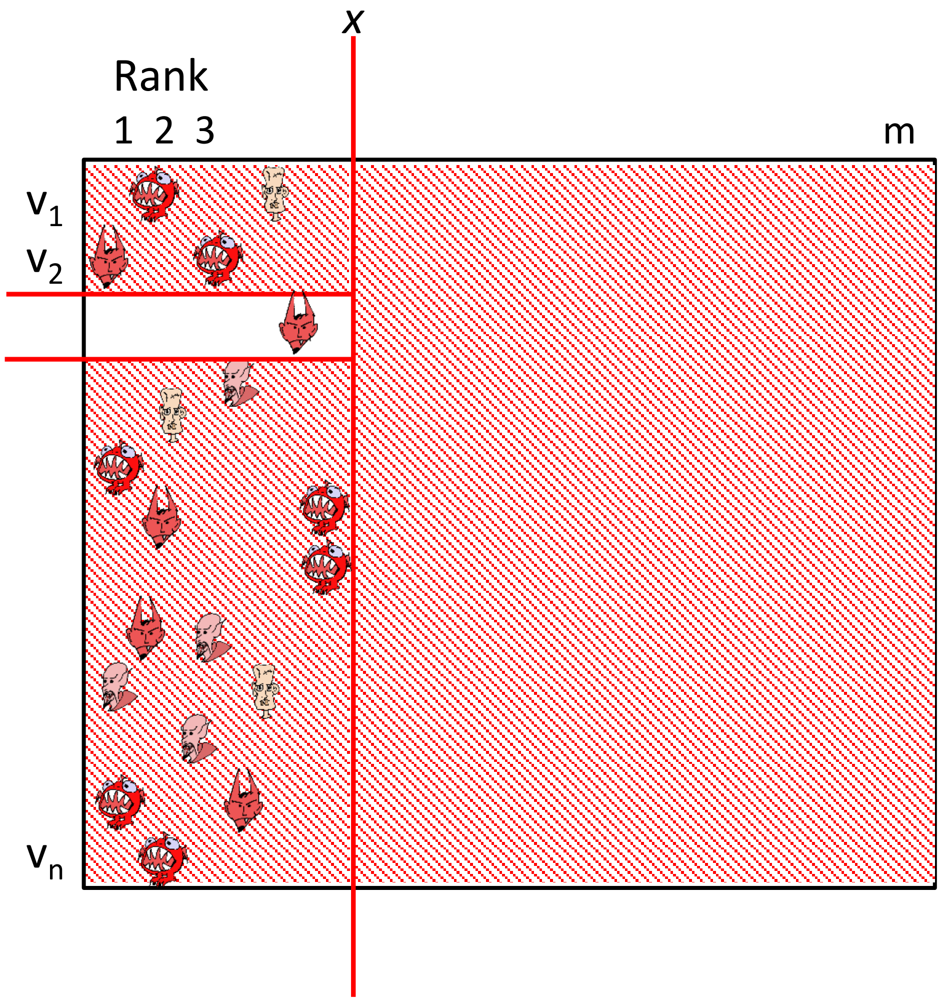
Loop: Keep picking the candidate that appears in the „available" part of the profile most frequently.



---

*) Achieving Fully Proportional Representation: Approximability Results, P. Skowron, P. Faliszewski, A. Slinko, *Artificial Intelligence*, Vol. 222, pp. 67--103, 2015.

PTAS for Borda Utilities, 1-Best OWA = (1,0, … , 0)[*]



Goal: pick *K* winners among *m* candidates, to get the highest utility

Initialize: Forget about the whole profile beyond rank *x*:

$x = mw(K) / K$ (w(K) is Lambert's W function, O(log K))

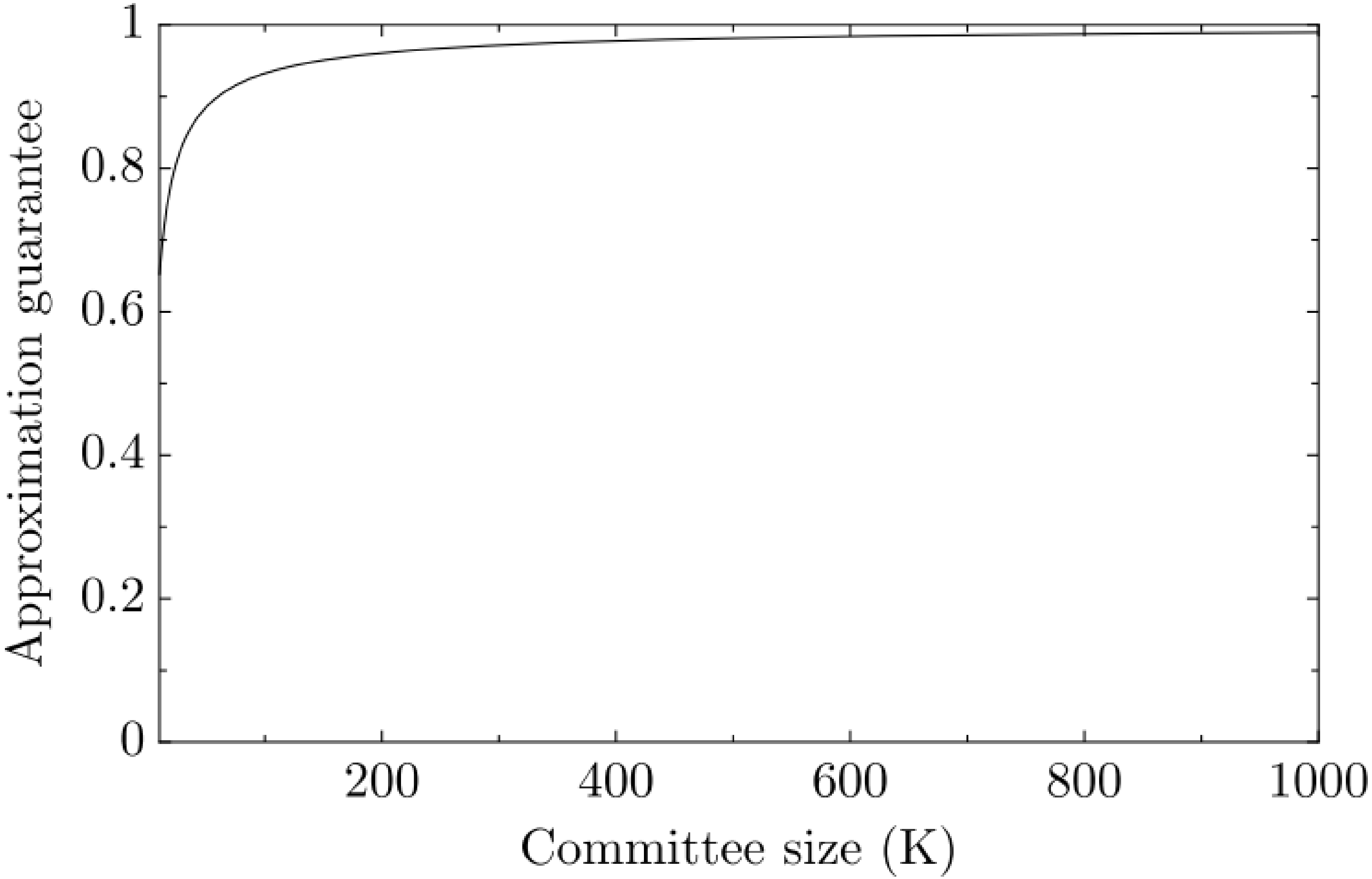
Loop: Keep picking the candidate that appears in the „available" part of the profile most frequently.

Guarantee: n(m-1)(1 – 2w(K)/K) utility

---

[*]) Achieving Fully Proportional Representation: Approximability Results, P. Skowron, P. Faliszewski, A. Slinko, *Artificial Intelligence*, Vol. 222, pp. 67--103,

PTAS for Borda Utilities, 1-Best OWA = (1,0, … , 0)



PTAS of OWA-Winner: Borda Utilities, Fixed Number of Top Nonzero Positions in the OWA Vector

OWA α with t top positions that are nonzero

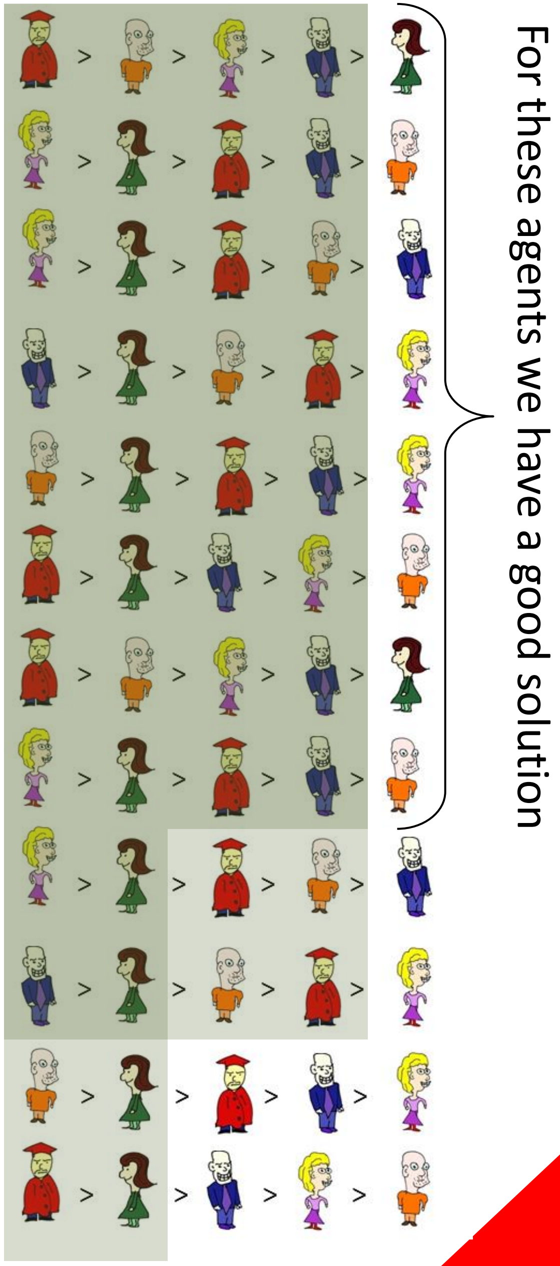
Select: k winners

Main idea:

- Select t groups of k/t winners

Technical idea:

- Use the PTAS for 1-Best-OWA-Winner (from literature)
  - Look at some top guys of all agents
  - Pick k/t winners to „cover" as many of agents
  - Most of the voters can be covered
  - Repeat for a following small group of „second to top preferences"
  - …



PTAS of OWA-Winner: Borda Utilities, Fixed Number of Top Nonzero Positions in the OWA Vector

OWA α with t top positions that are nonzero

Select: k winners

Geometric OWAs

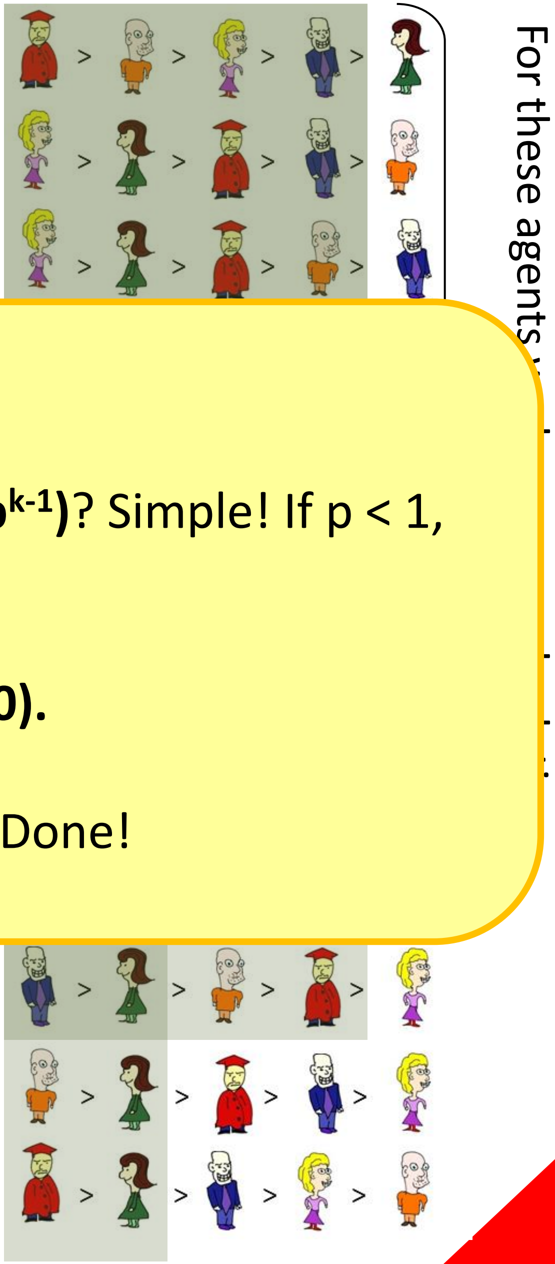
But… what with the geometric **OWA $(p^0, p^1, p^2, \ldots, p^{k-1})$**? Simple! If p < 1, then already for very small t, pt is negligible. Use:

$$\mathbf{OWA = (p^0, p^1, p^2, \ldots, p^{t-1}, 0, \ldots, 0).}$$

This OWA satisfies the assumption of out theorem. Done!

of agents

- Most of the voters can be covered
- Repeat for a following small group of „second to top preferences”
- …



Change of Focus: Axiomatic Approach

Single-winner **plurality rule**: Pick whoever is ranked first most often

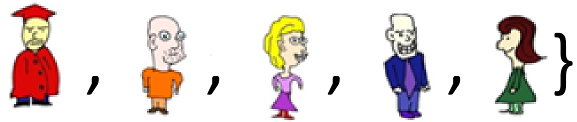
Is there a multiwinner plurality rule?

$f_{SNTV}(i_1, i_2, \dots, i_k) = A_1(i_1) + \dots + A_1(i_k)$

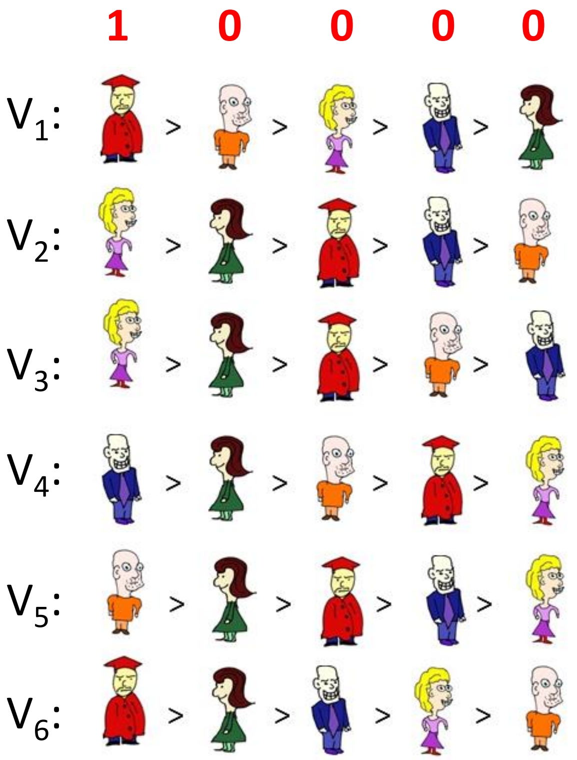
$f_{k\text{-Borda}}(i_1, i_2, \dots, i_k) = B(i_1) + \dots + B(i_k)$

$f_{Bloc}(i_1, i_2, \dots, i_k) = A_k(i_1) + \dots + A_k(i_k)$

$f_{CC}(i_1, i_2, \dots, i_k) = B(i_1)$

$C = \{$  $\}$

$V = (v_1, \dots, v_6)$



Change of Focus: Axiomatic Approach

Single-winner **plurality rule**: Pick whoever is ranked first most often

Is there a multiw

$f_{SNTV}(i_1, i_2, \dots, i_k)$

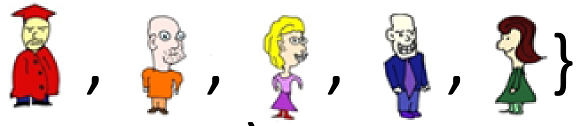
$f_{k\text{-Borda}}(i_1, i_2, \dots, i_k)$

$f_{Bloc}(i_1, i_2, \dots, i_k)$

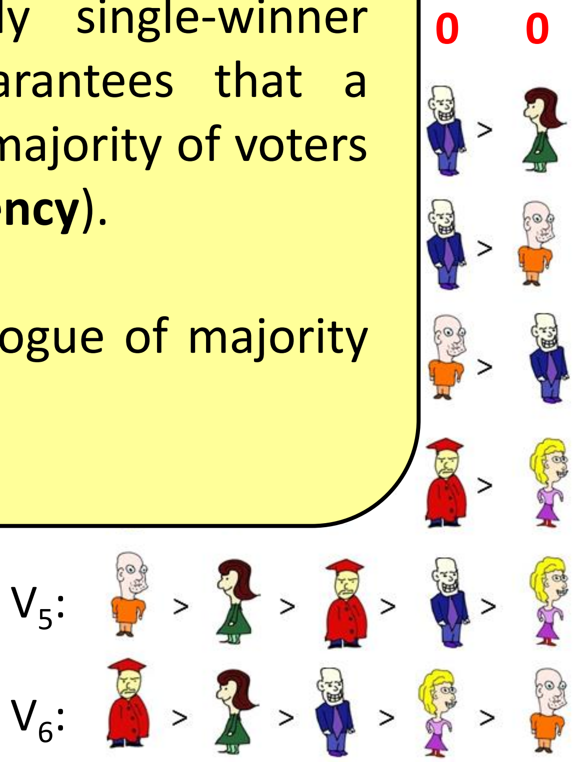
$f_{CC}(i_1, i_2, \dots, i_k)$

Plurality rule is the only single-winner scoring protocol that guarantees that a candidate ranked first by a majority of voters is elected (**majority consistency**).

Is there a multiwinner analogue of majority consistency?

C = {  }

$V = (v \quad v)$



Fixed-Majority Consistency

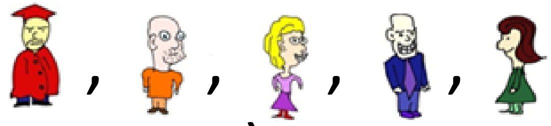
A multiwinner rule is fixed-majority consistent if it always elects a committee that a majority of voters ranks among top k positions.

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

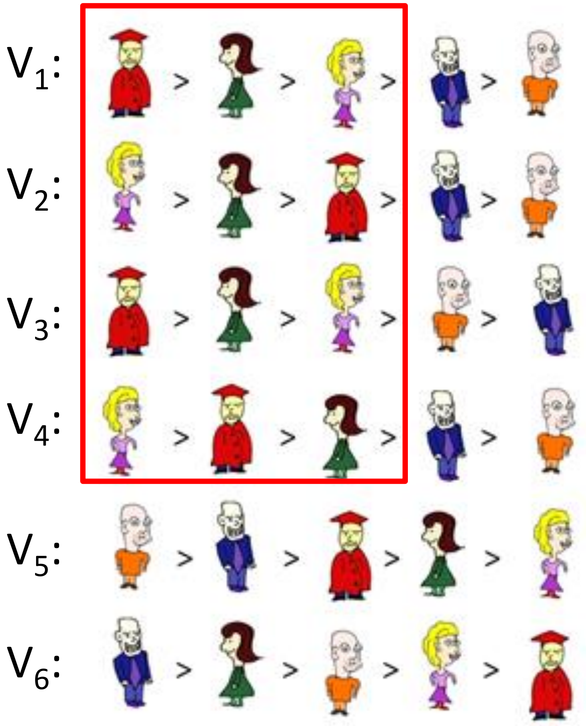
$f_{k\text{-Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

C = {  }

$V = (v_1, \ldots, v_6)$



Fixed-Majority Consistency

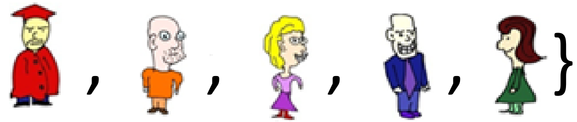
A multiwinner rule is fixed-majority consistent if it always elects a committee that a majority of voters ranks among top k positions.

$f_{SNTV}(i_1, i_2, \ldots, i_k) = A_1(i_1) + \ldots + A_1(i_k)$

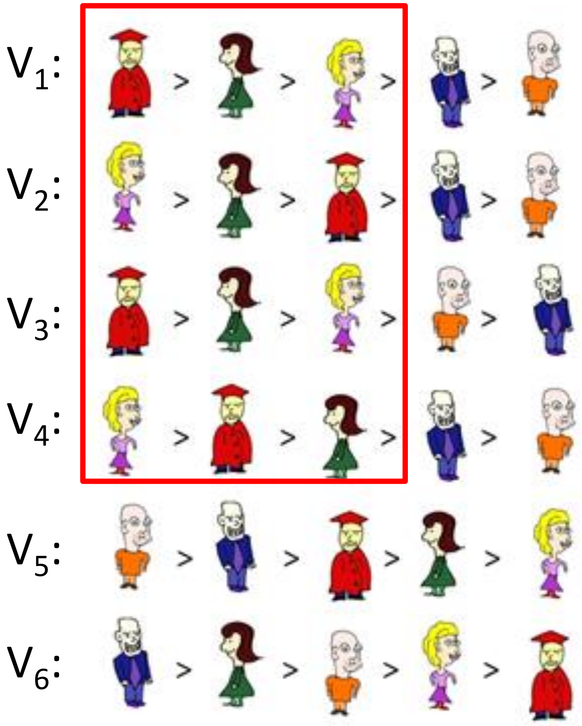
$f_{k\text{-Borda}}(i_1, i_2, \ldots, i_k) = B(i_1) + \ldots + B(i_k)$

$f_{Bloc}(i_1, i_2, \ldots, i_k) = A_k(i_1) + \ldots + A_k(i_k)$

$f_{CC}(i_1, i_2, \ldots, i_k) = B(i_1)$

$C = \{$  $\}$

$V = (v_1, \ldots , v_6)$



Fixed-Majority Consistency

A multiwinner rule is fixed-majority consistent if it always elects a committee that a majority of voters ranks among top k positions.

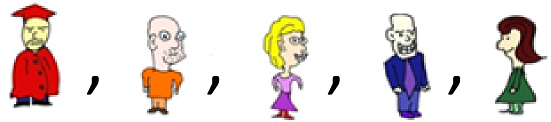
$f_{SNTV}(i_1, i_2, \dots, i_k) = A_1(i_1) + \dots + A_1(i_k)$

$f_{k\text{-}Borda}(i_1, i_2, \dots, i_k) = B(i_1) + \dots + B(i_k)$

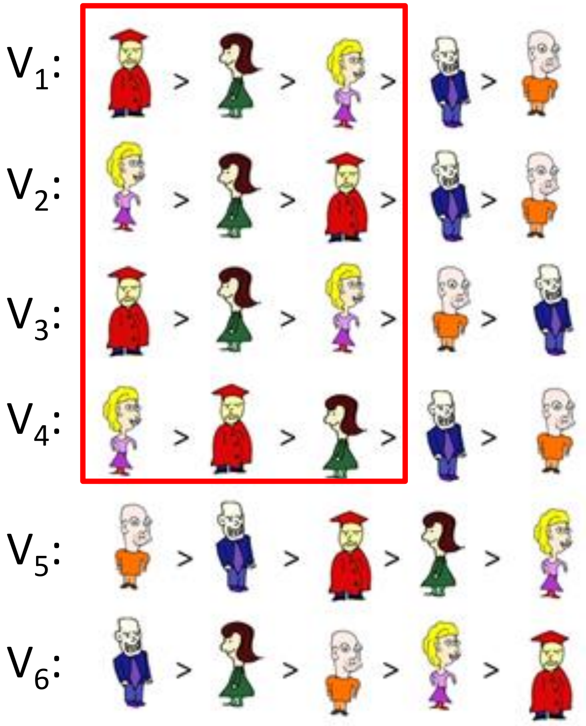
$f_{Bloc}(i_1, i_2, \dots, i_k) = A_k(i_1) + \dots + A_k(i_k)$

$f_{CC}(i_1, i_2, \dots, i_k) = B(i_1)$

$f_{Perf}(i_1, i_2, \dots, i_k) = A_k(i_k)$

C = {  }

$V = (v_1, \dots, v_6)$



Fixed-Majority Consistent CSRs

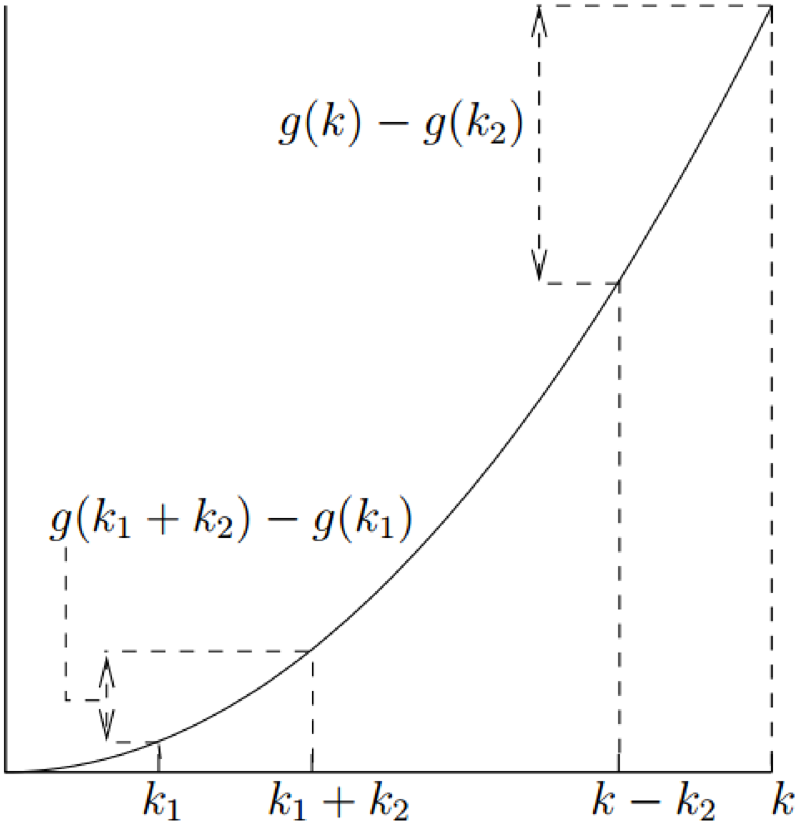
Theorem: Every fixed-majority consistent CSR is an OWA-based, with k-Approval rule:

$$f(i_1, i_2, \ldots, i_k) = w_1 A_k(i_1) + w_2 A_k(i_2) + \ldots + w_k A_k(i_k)$$

where function values w_1, w_1+w_2, $w_1+w_2+w_3$, … satisfy a convexity-like property.

Interpretation: Such rules are top-k-counting rules. There is a function g such that:

$$f(i_1, i_2, \ldots, i_k) = g(\#\{ t : i_t \leq k \})$$



Fixed-Majority Consistent CSRs

Theorem: Every fixed-majority
consistent
with k-A

f(i$_1$*, i*$_2$

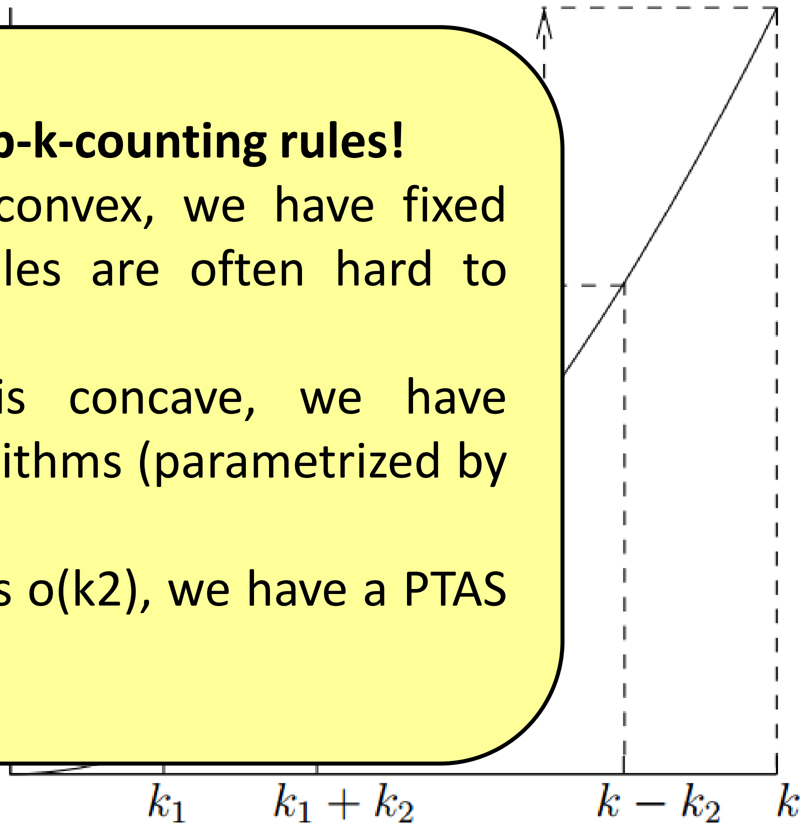
where f
w$_1$+w$_2$+
propert

Interpr
counting
such that:

$$f(i_1, i_2, \ldots, i_k) = g(\#\{\, t : i_t \le k \,\})$$

Lot's of interesting results for top-k-counting rules!

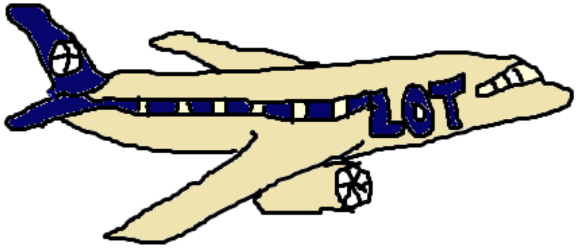
- If the counting function is convex, we have fixed majority consistency, but rules are often hard to approximate
- If the counting function is concave, we have approximability and FPT algorithms (parametrized by the number of voters)
- If the number of candidates is o(k2), we have a PTAS (non-finicky utilities)



Summary

- A new, very general, family of multiwinner election rules:
  - Borda, Chamberlin—Courant, …
  - Turns out to model rules we did not think of!
    - Proportional Approval Voting (PAV): OWA (1, 1/2, 1/3, 1/4, …)
    - NP-hardness even if each aggent „approves" at most **two candidates**, and if each candidate is „approved" by at most **three voters**
    - **Discussed in more detail by:** H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, T. Walsh: Computational Aspects of Multi-Winner Approval Voting.

- Broad NP-hardness results
- Examples of approximation results
- Developing axiomatizations



Thank you!