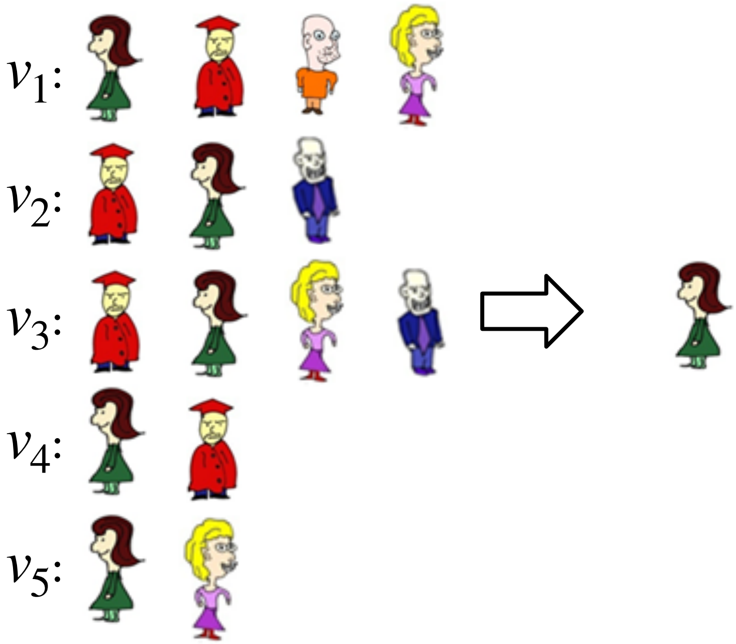
Computational Social Choice

Introduction to Voting



Piotr Skowron
University of Warsaw

The model

1. A set of n voters $V = \{v_1, v_2, \dots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \dots, c_m\}$.

The model

1. A set of n voters $V = \{v_1, v_2, \ldots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \ldots, c_m\}$.

Each voter v_i has a preference relation $\succ_i$ over the set of candidates.

The model

1. A set of n voters $V = \{v_1, v_2, \ldots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \ldots, c_m\}$.

Each voter v_i has a preference relation $\succ_i$ over the set of candidates.

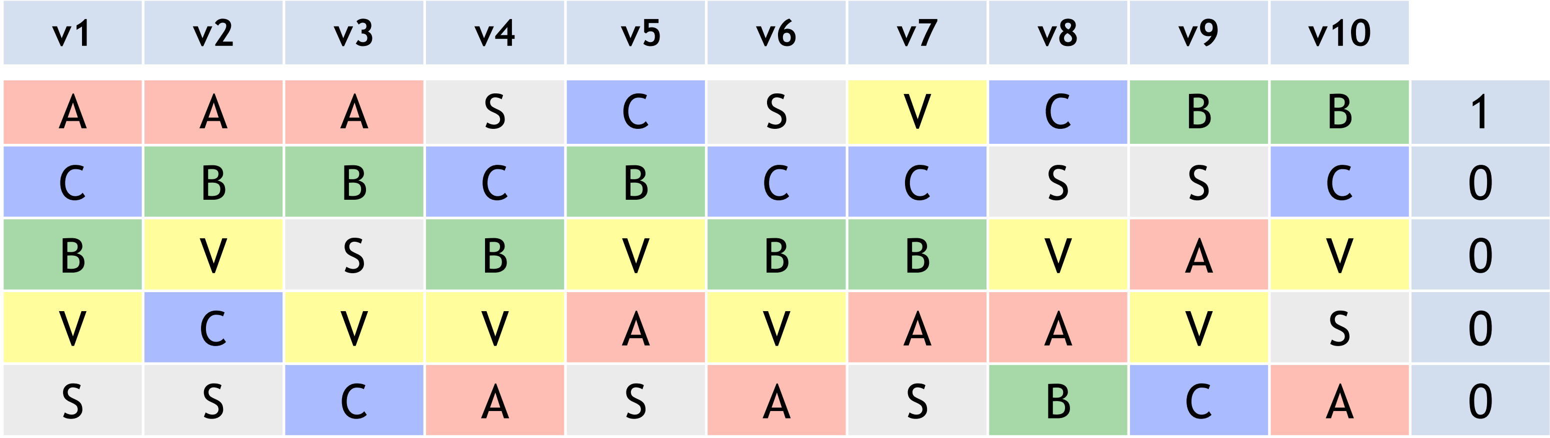
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

The model

1. A set of n voters $V = \{v_1, v_2, \dots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \dots, c_m\}$.

Each voter v_i has a preference relation $\succ_i$ over the set of candidates.

PLURALITY



v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	
A	A	A	S	C	S	V	C	B	B	1
C	B	B	C	B	C	C	S	S	C	0
B	V	S	B	V	B	B	V	A	V	0
V	C	V	V	A	V	A	A	V	S	0
S	S	C	A	S	A	S	B	C	A	0

A : 3

B : 2

C : 2

V : 1

S : 2

The model

1. A set of n voters $V = \{v_1, v_2, \ldots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \ldots, c_m\}$.

Each voter v_i has a preference relation $\succ_i$ over the set of candidates.

BORDA

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	
A	A	A	S	C	S	V	C	B	B	4
C	B	B	C	B	C	C	S	S	C	3
B	V	S	B	V	B	B	V	A	V	2
V	C	V	V	A	V	A	A	V	S	1
S	S	C	A	S	A	S	B	C	A	0

A : 17

B : 25

C : 24

V : 17

S : 17

The model

1. A set of n voters $V = \{v_1, v_2, \ldots, v_n\}$.
2. A set of m candidates $C = \{c_1, c_2, \ldots, c_m\}$.

Each voter v_i has a preference relation $\succ_i$ over the set of candidates.

VETO

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	
A	A	A	S	C	S	V	C	B	B	1
C	B	B	C	B	C	C	S	S	C	1
B	V	S	B	V	B	B	V	A	V	1
V	C	V	V	A	V	A	A	V	S	1
S	S	C	A	S	A	S	B	C	A	0

A : 7

B : 9

C : 8

V : 10

S : 6

The class of positional scoring rules

Each rule in this class is defined by a scoring vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$.

- A candidate ranked at position j by a voter v gets α_j points from v.
- The candidate with the highest number of points wins the election.

The class of positional scoring rules

Each rule in this class is defined by a scoring vector $\alpha = \left(\alpha_1, \alpha_2, \ldots, \alpha_m\right)$.

- A candidate ranked at position j by a voter v gets α_j points from v.
- The candidate with the highest number of points wins the election.

Examples of positional scoring rules:

Plurality: $\alpha = (1, 0, 0, \ \ldots, 0)$

Borda: $\alpha = (m - 1, m - 2, m - 3, \ \ldots, 0)$

The class of positional scoring rules

Each rule in this class is defined by a scoring vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$.

- A candidate ranked at position j by a voter v gets α_j points from v.
- The candidate with the highest number of points wins the election.

Examples of positional scoring rules:

Plurality: $\alpha = (1, 0, 0, \ \ldots, 0)$

Borda: $\alpha = (m-1, m-2, m-3, \ \ldots, 0)$

Veto: $\alpha = (1, 1, 1, \ \ldots, 1, \ 0)$

The class of positional scoring rules

Each rule in this class is defined by a scoring vector $\alpha = \left(\alpha_1, \alpha_2, \ldots, \alpha_m\right)$.

- A candidate ranked at position j by a voter v gets α_j points from v.
- The candidate with the highest number of points wins the election.

Examples of positional scoring rules:

Plurality: $\alpha = (1, 0, 0, \ \ldots, 0)$

Borda: $\alpha = (m-1, m-2, m-3, \ \ldots, 0)$

Veto: $\alpha = (1, 1, 1, \ \ldots, 1, \ 0)$

k-approval: $\alpha = (\underbrace{1, 1, \ \ldots, 1,}_{k \text{ ones}} \ 0, 0, \ \ldots, 0)$

The class of positional scoring rules

Each rule in this class is defined by a scoring vector $\alpha = \left(\alpha_1, \alpha_2, \ldots, \alpha_m\right)$.

- A candidate ranked at position j by a voter v gets α_j points from v.
- The candidate with the highest number of points wins the election.

Examples of positional scoring rules:

Plurality: $\alpha = (1, 0, 0, \ldots, 0)$

Borda: $\alpha = (m-1, m-2, m-3, \ldots, 0)$

Veto: $\alpha = (1, 1, 1, \ldots, 1, 0)$

k-approval: $\alpha = (\underbrace{1, 1, \ldots, 1}_{k \text{ ones}}, 0, 0, \ldots, 0)$

Formula One: $\alpha = \left(25, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0, \ldots, 0\right)$

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

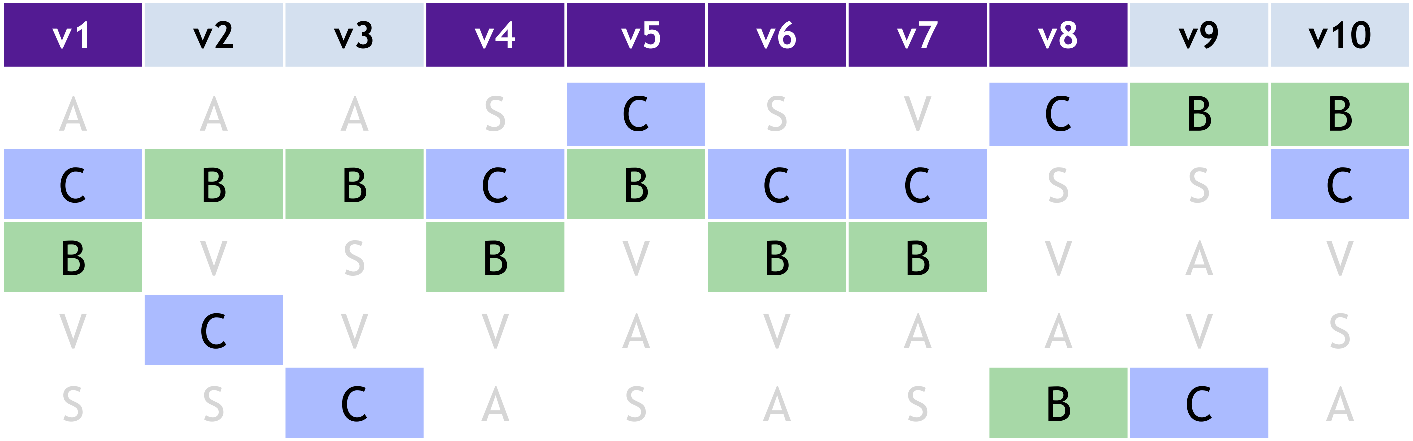
for each c' we have $\left|\{v_i \in V \colon c \succ_i c'\}\right| > \left|\{v_i \in V \colon c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$



v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

Condorcet cycle: a majority prefers A to B, and B to C, but also a majority prefers C to B.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
A	A	A	B	B	B	B	C	C	C	C	C
B	B	B	C	C	C	C	A	A	A	A	A
C	C	C	A	A	A	A	B	B	B	B	B

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

Condorcet cycle: a majority prefers A to B, and B to C, but also a majority prefers C to B.

Condorcet consistency: a rule selects the Condorcet winner, whenever such exists.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
A	A	A	B	B	B	B	C	C	C	C	C
B	B	B	C	C	C	C	A	A	A	A	A
C	C	C	A	A	A	A	B	B	B	B	B

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V: c \succ_i c'\}\right| > \left|\{v_i \in V: c' \succ_i c\}\right|$

Copeland rule: each candidate c gets one point for each candidate she defeats in a head-to-head comparison.

$\text{score}(c) = \left|\{c': c \succ_{\text{maj}} c'\}\right|$, where

$c \succ_{\text{maj}} c'$ iff $\left|\{v_i \in V: c \succ_i c'\}\right| > \left|\{v_i \in V: c' \succ_i c\}\right|$

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
A	A	A	B	B	B	B	C	C	C	C	C
B	B	B	C	C	C	C	A	A	A	A	A
C	C	C	A	A	A	A	B	B	B	B	B

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V \colon c \succ_i c'\}\right| > \left|\{v_i \in V \colon c' \succ_i c\}\right|$

Copeland rule: each candidate c gets one point for each candidate she defeats in a head-to-head comparison.

$\text{score}(c) = \left|\{c' \colon c \succ_{\text{maj}} c'\}\right|$, where

$c \succ_{\text{maj}} c'$ iff $\left|\{v_i \in V \colon c \succ_i c'\}\right| > \left|\{v_i \in V \colon c' \succ_i c\}\right|$

Who will win in the election below?

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
A	A	A	B	B	B	B	C	C	C	C	C
B	B	B	C	C	C	C	A	A	A	A	A
C	C	C	A	A	A	A	B	B	B	B	B

Condorcet rules

Condorcet winner: the candidate c that beats each other candidate in the head-to-head comparison.

for each c' we have $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

Copeland rule: each candidate c gets one point for each candidate she defeats in a head-to-head comparison.

$\text{score}(c) = \left|\{c'\colon c \succ_{\text{maj}} c'\}\right|$, where

$c \succ_{\text{maj}} c'$ iff $\left|\{v_i \in V\colon c \succ_i c'\}\right| > \left|\{v_i \in V\colon c' \succ_i c\}\right|$

Who will win in the election below? (Very irresolute rule!)

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
A	A	A	B	B	B	B	C	C	C	C	C
B	B	B	C	C	C	C	A	A	A	A	A
C	C	C	A	A	A	A	B	B	B	B	B

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V \colon c \succ_i c'\} \right|$$

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V \colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

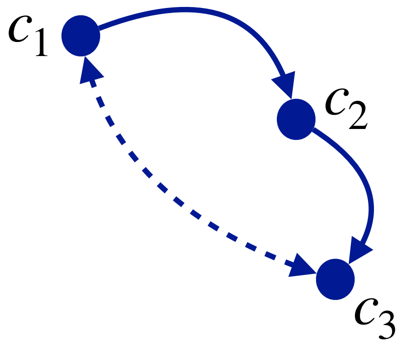
$$d(c, v') = \left| \{v_i \in V\colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1,\ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left|\{v_i \in V \colon c \succ_i c'\}\right|$$

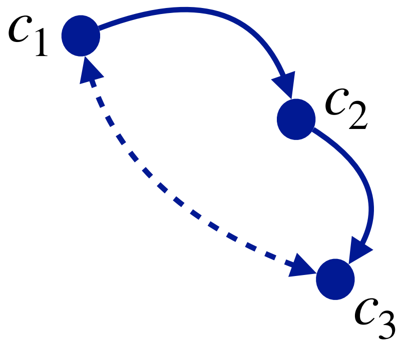
A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \min(\mathrm{str}(c_1 \to c_2), \mathrm{str}(c_2 \to c_3))$$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V: c \succ_i c'\} \right|$$

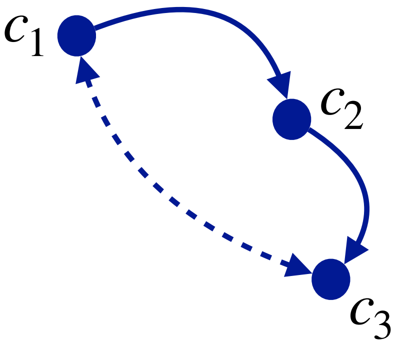
A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \min(\mathrm{str}(c_1 \to c_2), \mathrm{str}(c_2 \to c_3))$$
$$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V\colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \dots, c_{i_r}$ such that:

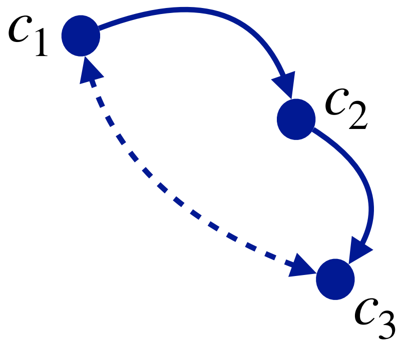
1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1,\dots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \min(\mathrm{str}(c_1 \to c_2), \mathrm{str}(c_2 \to c_3))$$
$$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$

Case 1: $\mathrm{str}(c_1 \to c_2) \leq \mathrm{str}(c_2 \to c_3)$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V : c \succ_i c'\} \right|$$

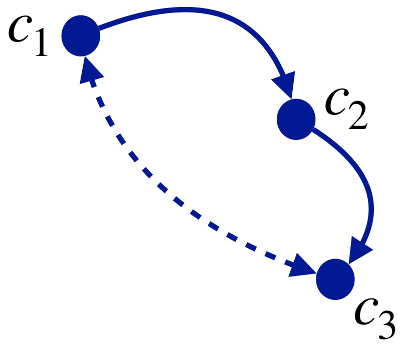
A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

Case 1: $\mathrm{str}(c_1 \rightarrow c_2) \leq \mathrm{str}(c_2 \rightarrow c_3)$



$$\mathrm{str}(c_1 \rightarrow c_3) \geq \mathrm{str}(c_1 \rightarrow c_2)$$
$$\mathrm{str}(c_1 \rightarrow c_2) > \min(\mathrm{str}(c_2 \rightarrow c_3), \mathrm{str}(c_3 \rightarrow c_1))$$
$$\mathrm{str}(c_2 \rightarrow c_3) > \min(\mathrm{str}(c_3 \rightarrow c_1), \mathrm{str}(c_1 \rightarrow c_2))$$

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V \colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

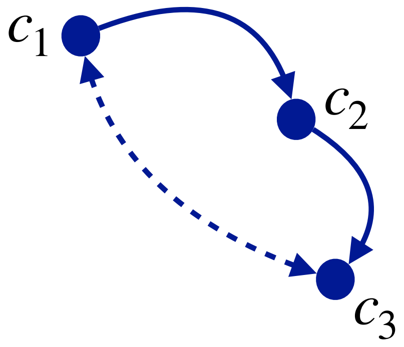
1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \mathrm{str}(c_1 \to c_2)$$
$$\mathrm{str}(c_1 \to c_2) > \mathrm{str}(c_3 \to c_1)$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$

Case 1: $\mathrm{str}(c_1 \to c_2) \leq \mathrm{str}(c_2 \to c_3)$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

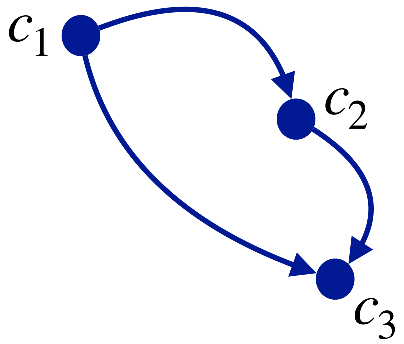
$$d(c, v') = \left| \{v_i \in V : c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

Case 1: $\mathrm{str}(c_1 \to c_2) \leq \mathrm{str}(c_2 \to c_3)$



This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \mathrm{str}(c_1 \to c_2)$$
$$\mathrm{str}(c_1 \to c_2) > \mathrm{str}(c_3 \to c_1)$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left|\{v_i \in V \colon c \succ_i c'\}\right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

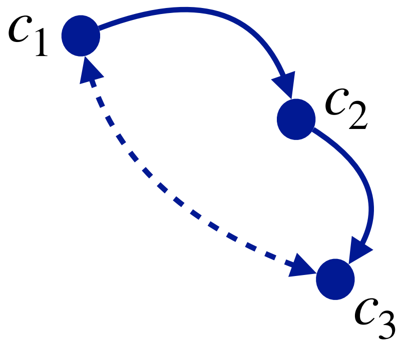
1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \min(\mathrm{str}(c_1 \to c_2), \mathrm{str}(c_2 \to c_3))$$
$$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$

Case 2: $\mathrm{str}(c_1 \to c_2) \geq \mathrm{str}(c_2 \to c_3)$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left|\{v_i \in V \colon c \succ_i c'\}\right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \dots, c_{i_r}$ such that:

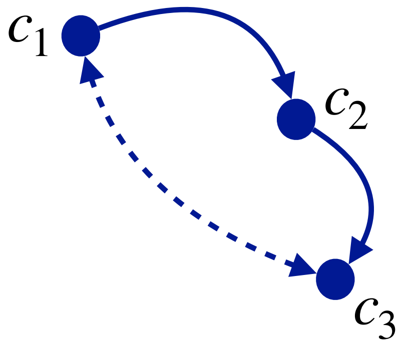
1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \dots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

$$\mathrm{str}(c_1 \to c_3) \geq \mathrm{str}(c_2 \to c_3)$$
$$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$$
$$\mathrm{str}(c_2 \to c_3) > \min(\mathrm{str}(c_3 \to c_1), \mathrm{str}(c_1 \to c_2))$$

Case 2: $\mathrm{str}(c_1 \to c_2) \geq \mathrm{str}(c_2 \to c_3)$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V\colon c \succ_i c'\} \right|$$

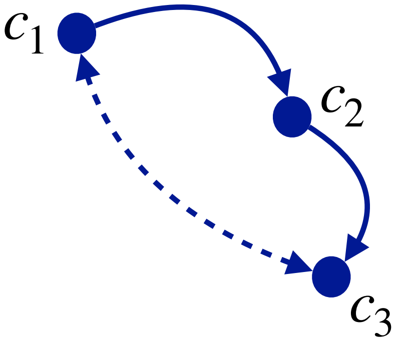
A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1,\ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

Case 2: $\mathrm{str}(c_1 \to c_2) \geq \mathrm{str}(c_2 \to c_3)$



$$\mathrm{str}(c_1 \to c_3) \geq \mathrm{str}(c_2 \to c_3)$$
$$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$$
$$\mathrm{str}(c_2 \to c_3) > \mathrm{str}(c_3 \to c_1)$$

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V \colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

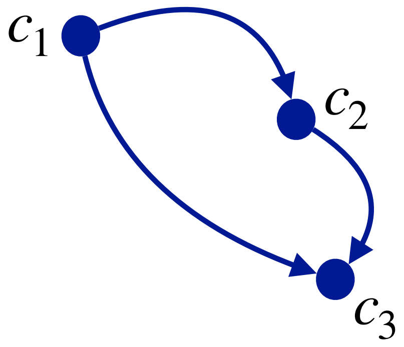
This relation is transitive!

$\mathrm{str}(c_1 \to c_3) \geq \mathrm{str}(c_2 \to c_3)$

$\mathrm{str}(c_1 \to c_2) > \min(\mathrm{str}(c_2 \to c_3), \mathrm{str}(c_3 \to c_1))$

$\mathrm{str}(c_2 \to c_3) > \mathrm{str}(c_3 \to c_1)$

Case 2: $\mathrm{str}(c_1 \to c_2) \geq \mathrm{str}(c_2 \to c_3)$



Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left| \{v_i \in V \colon c \succ_i c'\} \right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1,\ldots,r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

Condorcet rules

Schulze method

$d(c, c')$: the number of voters who prefer c over c', i.e.,

$$d(c, v') = \left|\{v_i \in V \colon c \succ_i c'\}\right|$$

A **path of strength p from c to c'** is a sequence $c_{i_1}, c_{i_2}, \ldots, c_{i_r}$ such that:

1. $c = c_{i_1}$ and $c' = c_{i_r}$, and,
2. for each $k \in \{1, \ldots, r-1\}$ we have $d(c_k, c_{k+1}) > p$.

A candidate **c is better than c'** if the best path from c to c' is stronger than the best path from c' to c.

This relation is transitive!

A candidate that is weakly better than every other candidate is a winner.

Condorcet rules

Schulze method: Example

45 voters

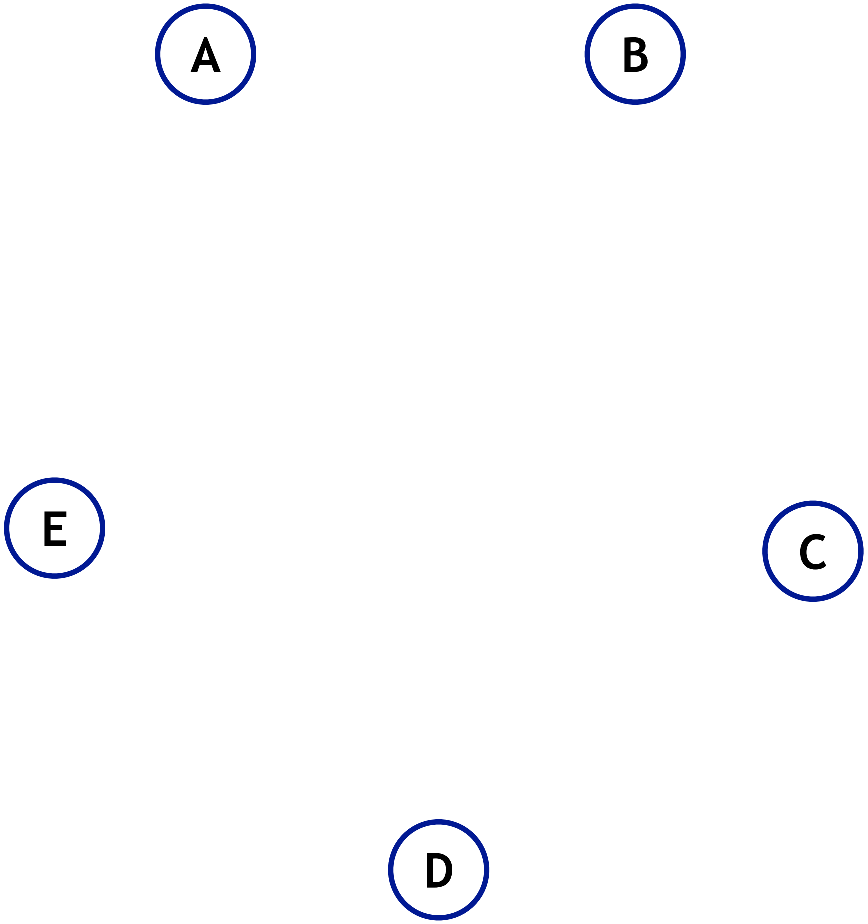
5	5	8	3	7	2	7	8
A	A	B	C	C	C	D	E
C	D	E	A	A	B	C	B
B	E	D	B	E	A	E	A
E	C	A	E	B	D	B	D
D	B	C	D	D	E	A	C

Condorcet rules

Schulze method: Example

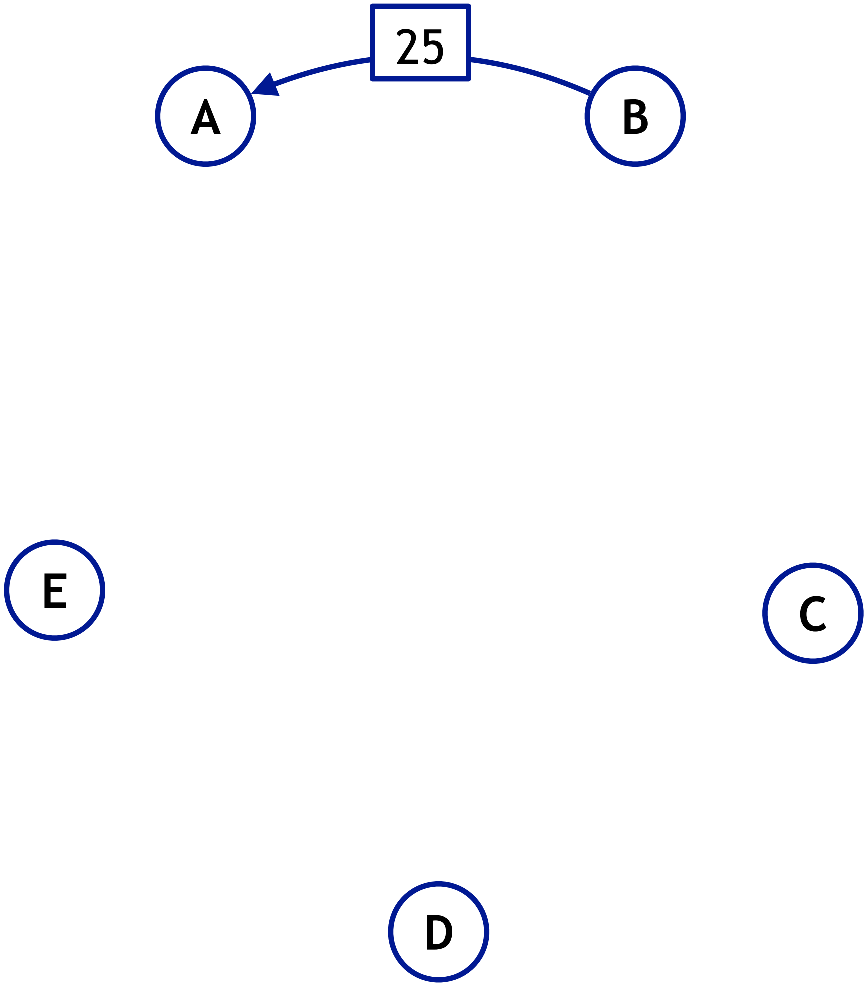
45 voters

5	5	8	3	7	2	7	8
A	A	B	C	C	C	D	E
C	D	E	A	A	B	C	B
B	E	D	B	E	A	E	A
E	C	A	E	B	D	B	D
D	B	C	D	D	E	A	C

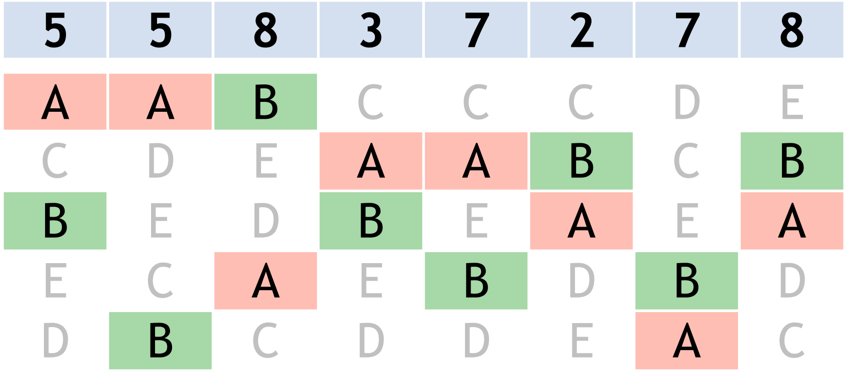


Condorcet rules

Schulze method: Example



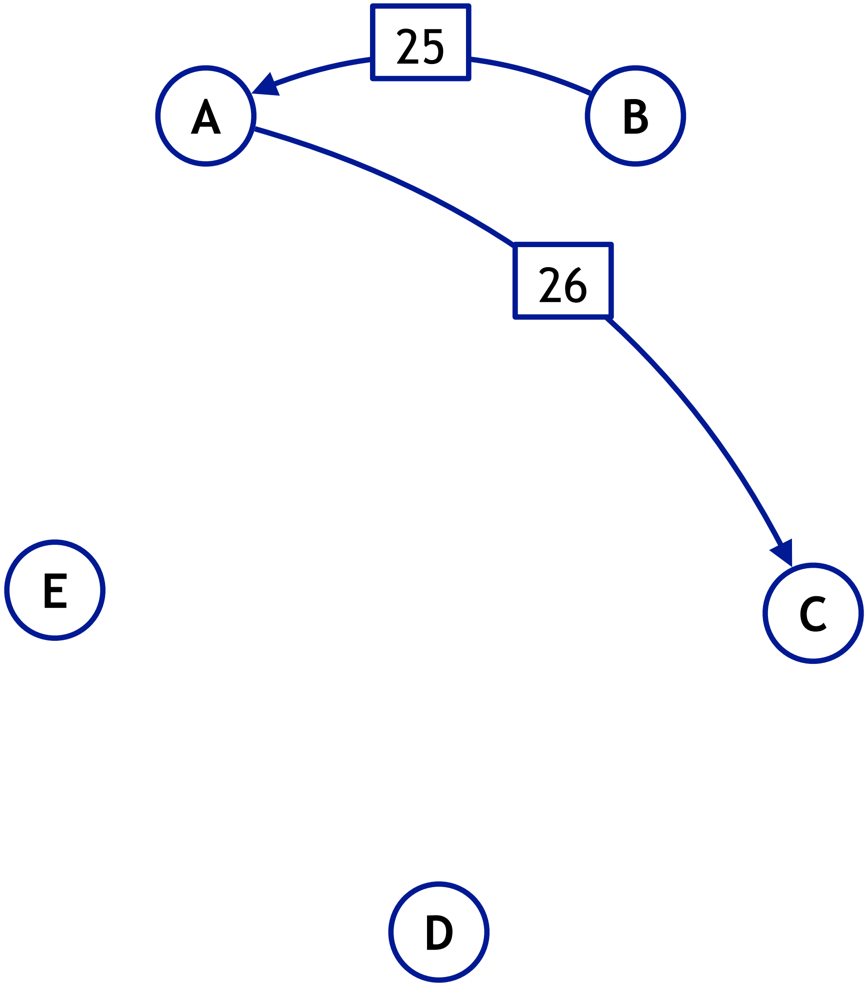
45 voters



5	5	8	3	7	2	7	8
A	A	B	C	C	C	D	E
C	D	E	A	A	B	C	B
B	E	D	B	E	A	E	A
E	C	A	E	B	D	B	D
D	B	C	D	D	E	A	C

Condorcet rules

Schulze method: Example

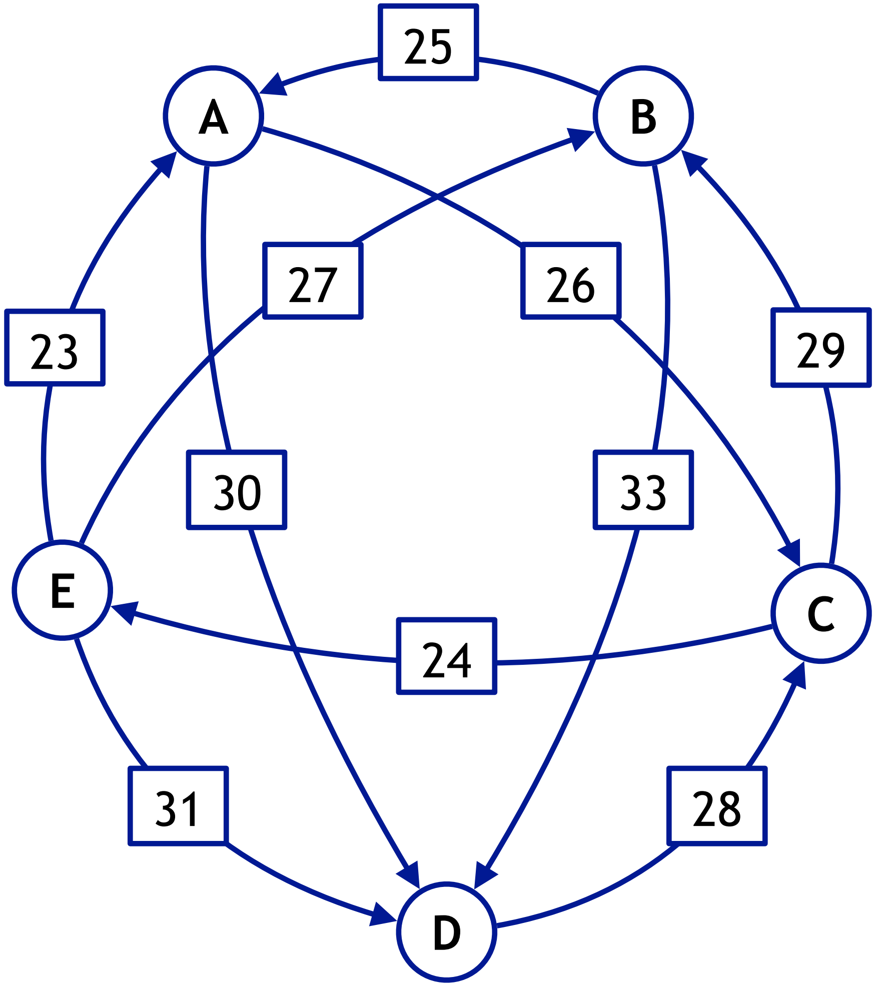


45 voters

5	5	8	3	7	2	7	8
A	A	B	C	C	C	D	E
C	D	E	A	A	B	C	B
B	E	D	B	E	A	E	A
E	C	A	E	B	D	B	D
D	B	C	D	D	E	A	C

Condorcet rules

Schulze method: Example

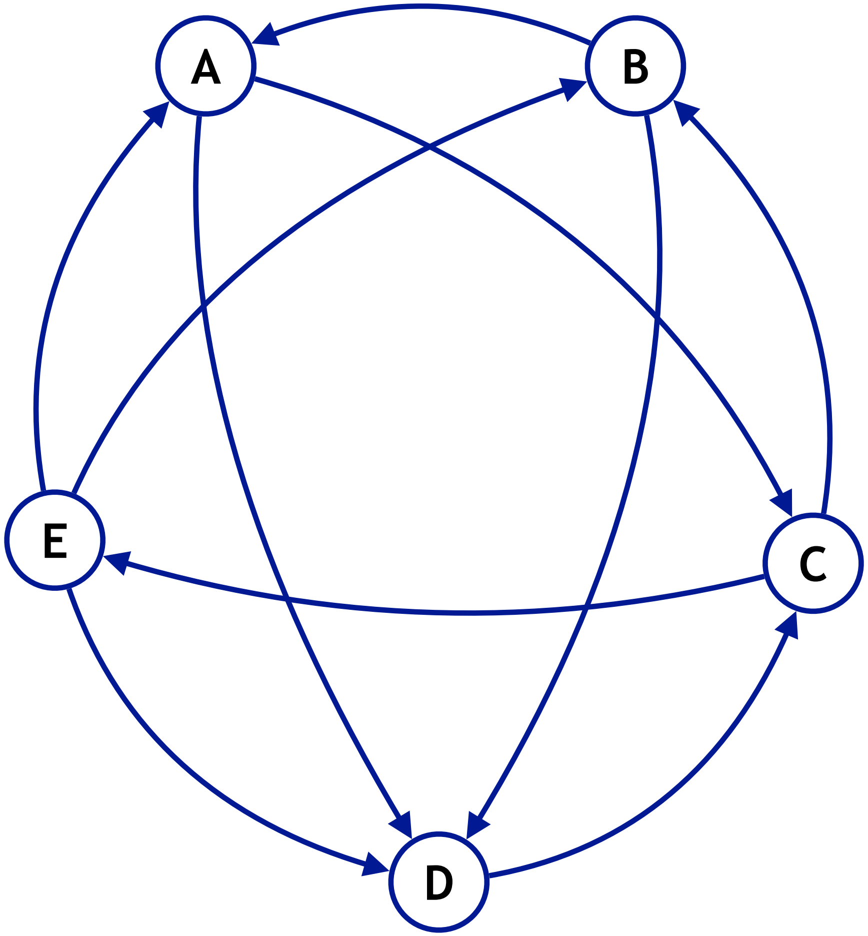


45 voters

5	5	8	3	7	2	7	8
A	A	B	C	C	C	D	E
C	D	E	A	A	B	C	B
B	E	D	B	E	A	E	A
E	C	A	E	B	D	B	D
D	B	C	D	D	E	A	C

Condorcet rules

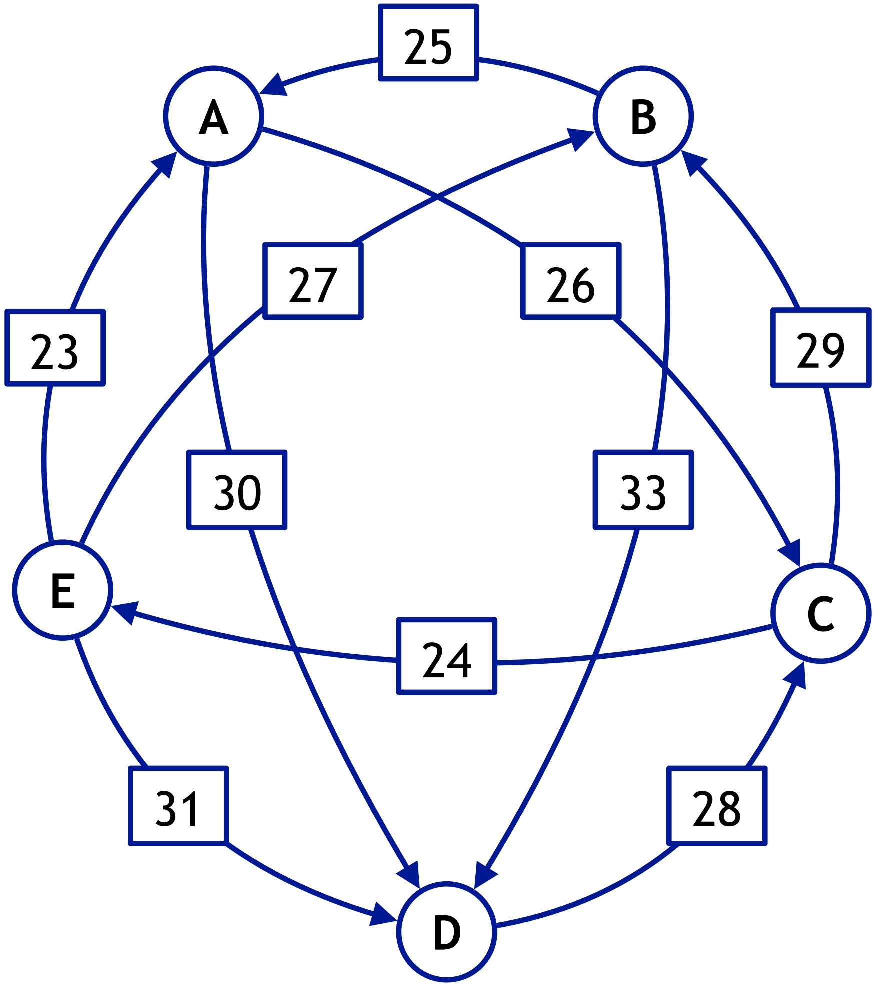
Schulze method: Example



No Condorcet winner!

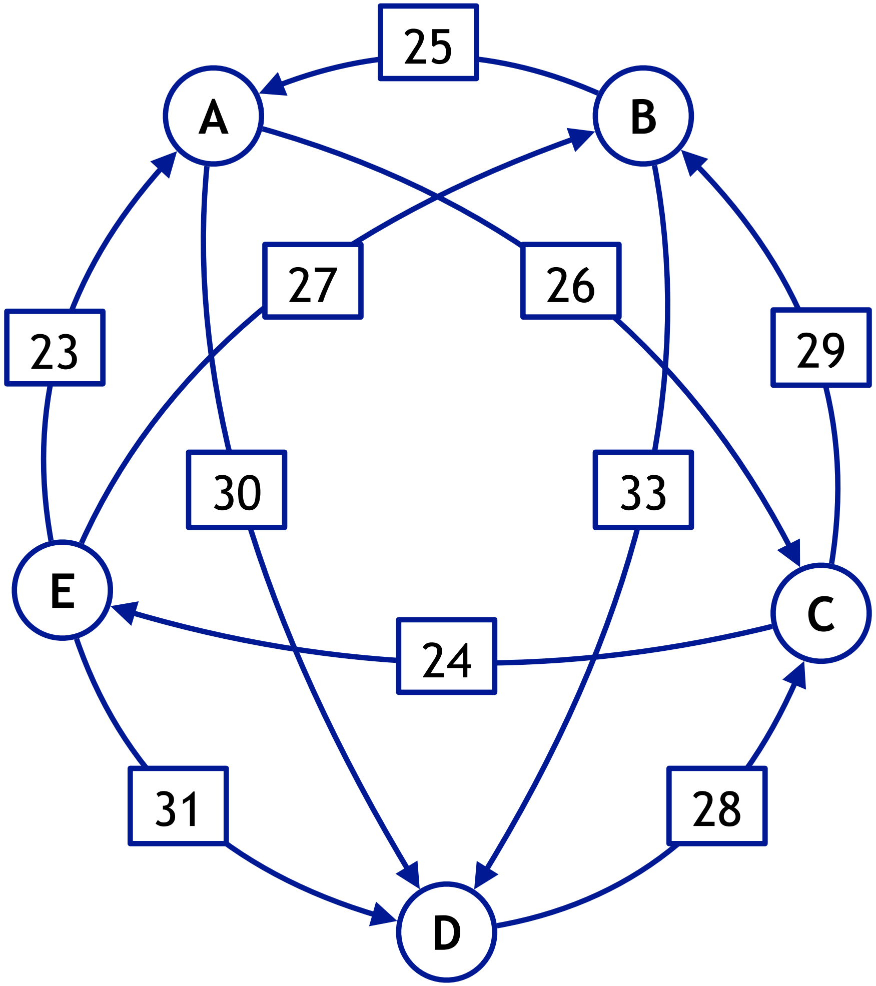
Condorcet rules

Schulze method: Example



Condorcet rules

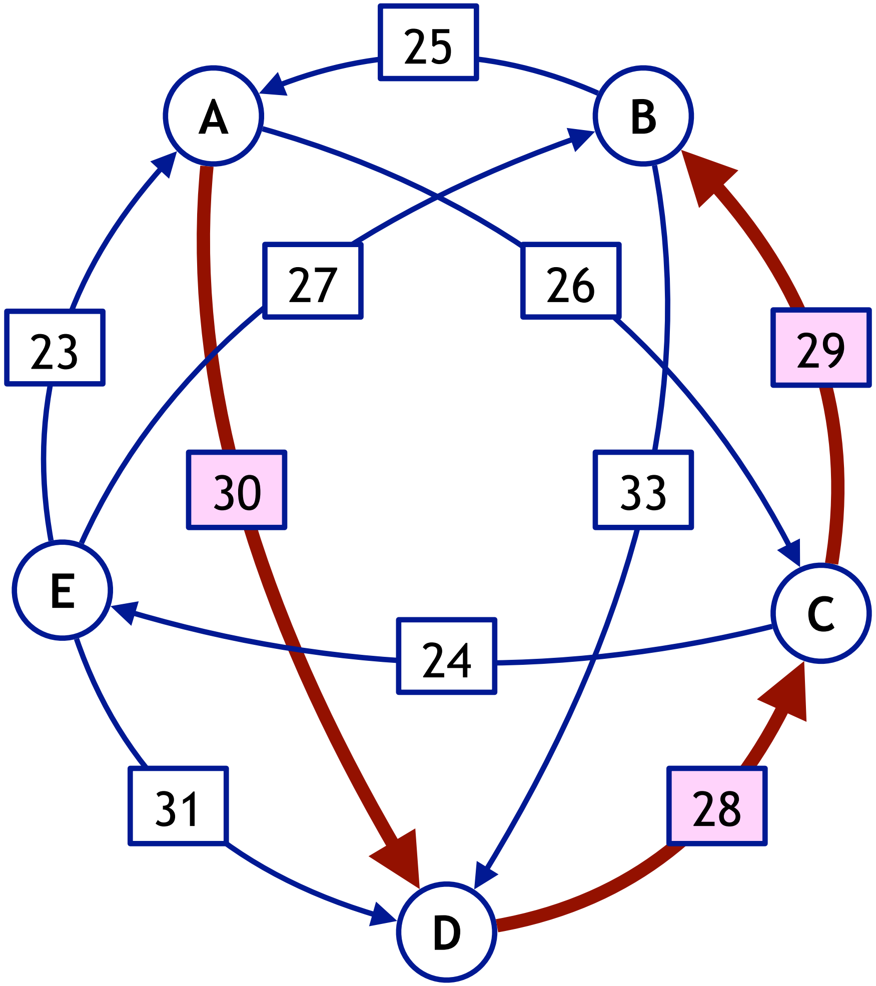
Schulze method: Example



	A	B	C	D	E
A	-				
B		-			
C			-		
D				-	
E					-

Condorcet rules

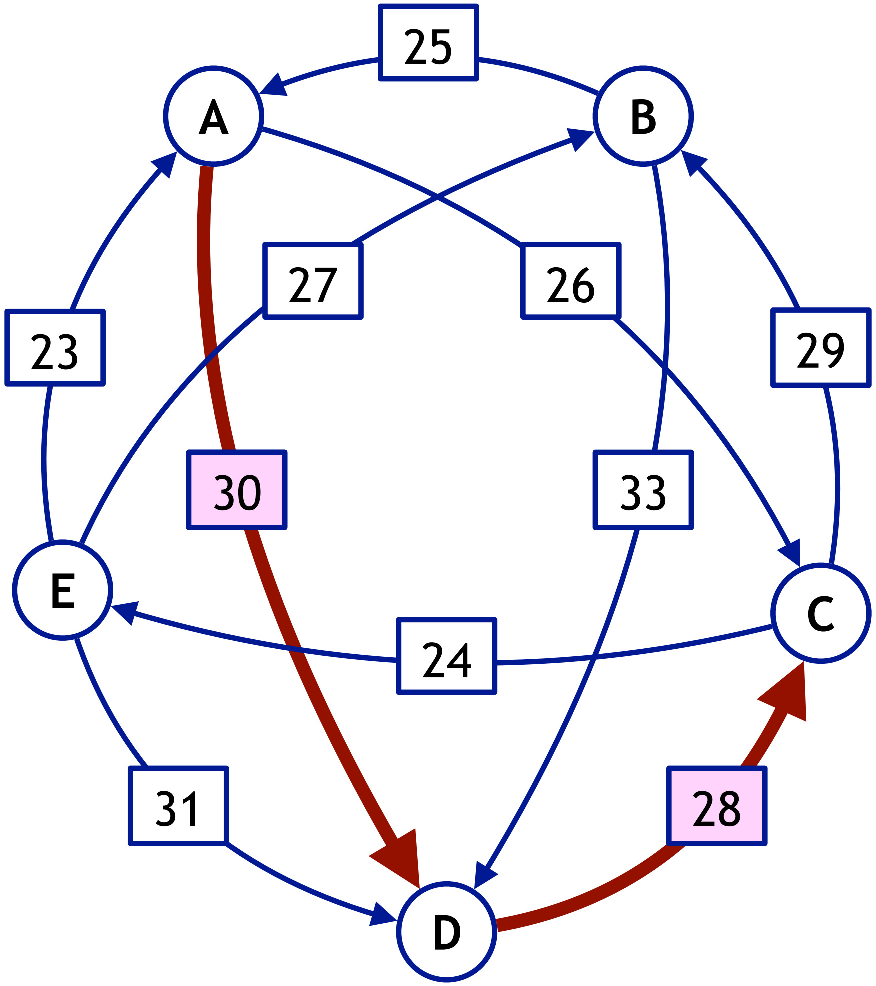
Schulze method: Example



	A	B	C	D	E
A	-	28			
B		-			
C			-		
D				-	
E					-

Condorcet rules

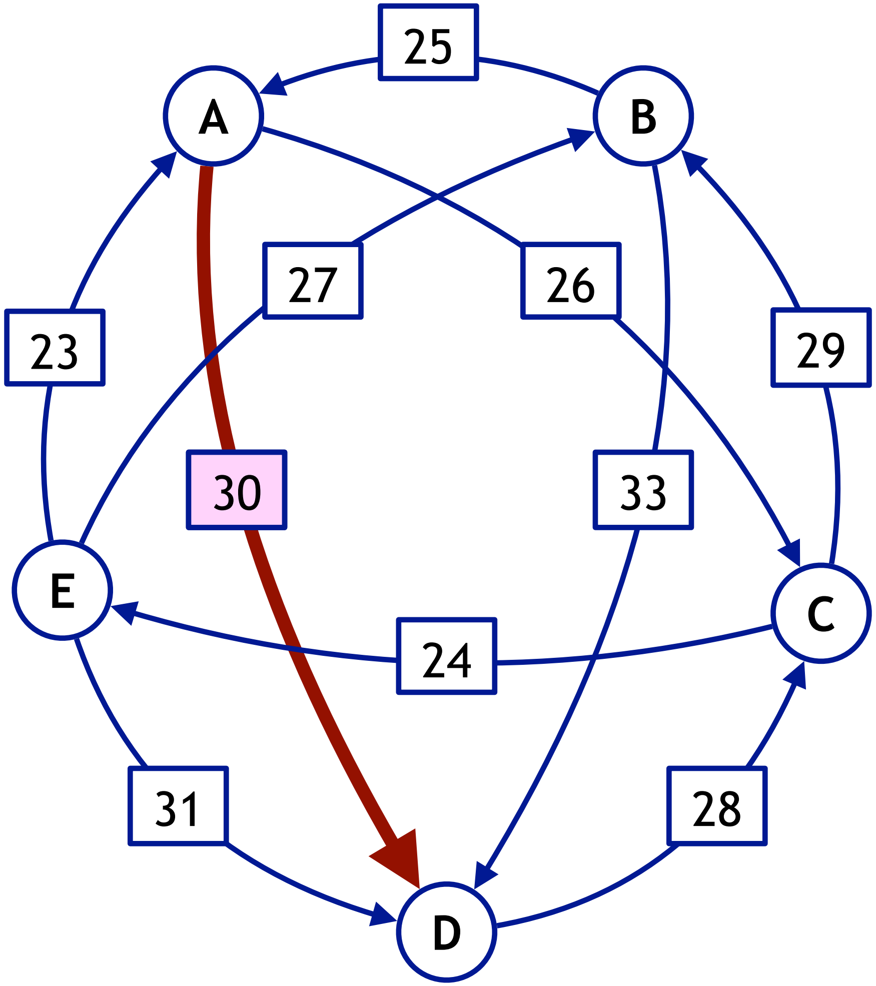
Schulze method: Example



	A	B	C	D	E
A	-	28	28		
B		-			
C			-		
D				-	
E					-

Condorcet rules

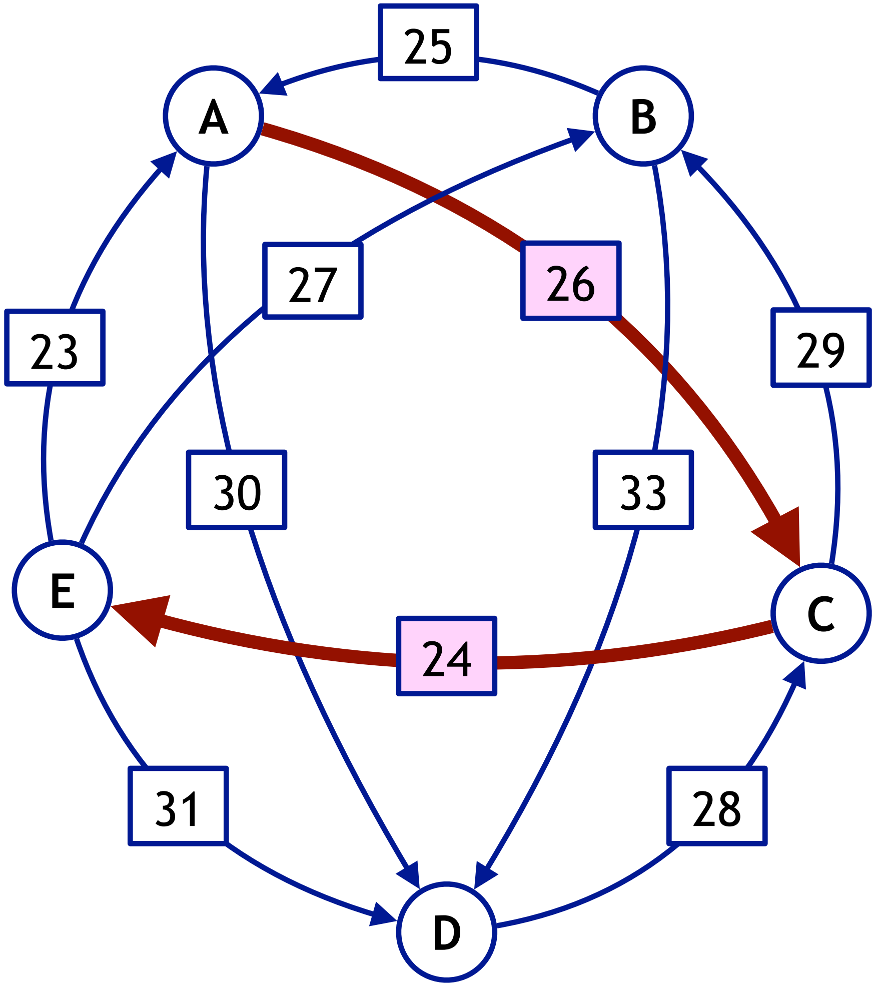
Schulze method: Example



	A	B	C	D	E
A	-	28	28	30	
B		-			
C			-		
D				-	
E					-

Condorcet rules

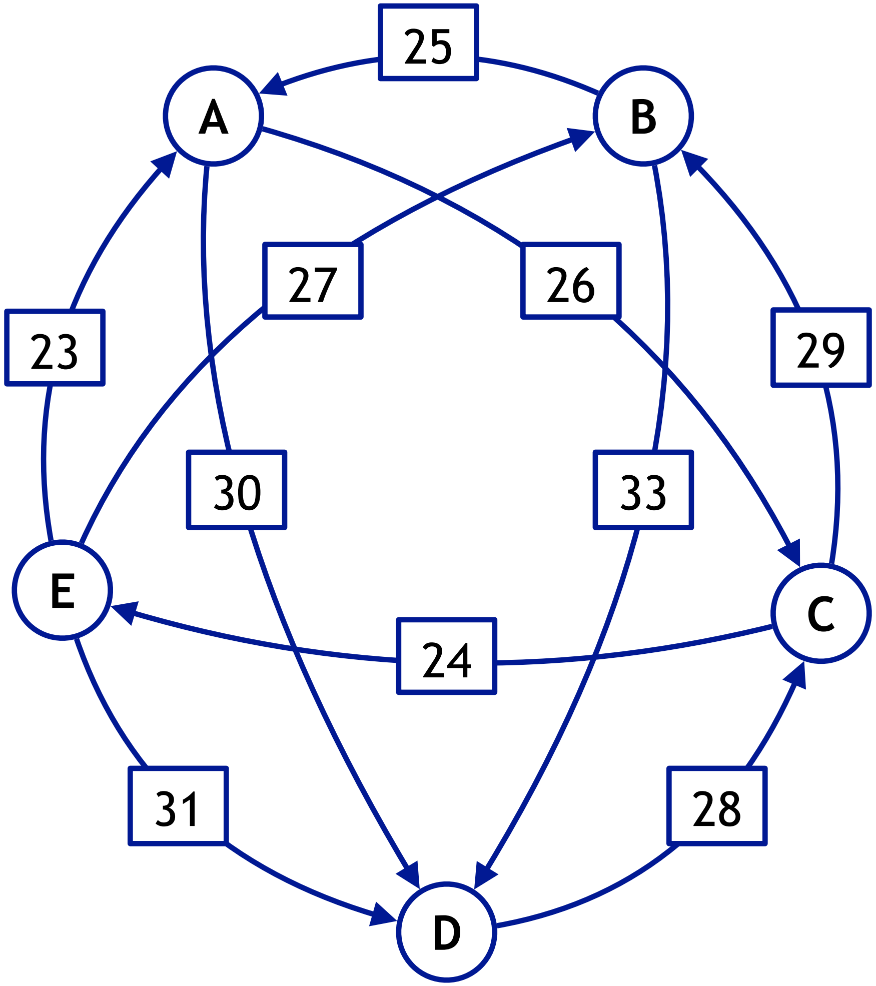
Schulze method: Example



	A	B	C	D	E
A	-	28	28	30	24
B		-			
C			-		
D				-	
E					-

Condorcet rules

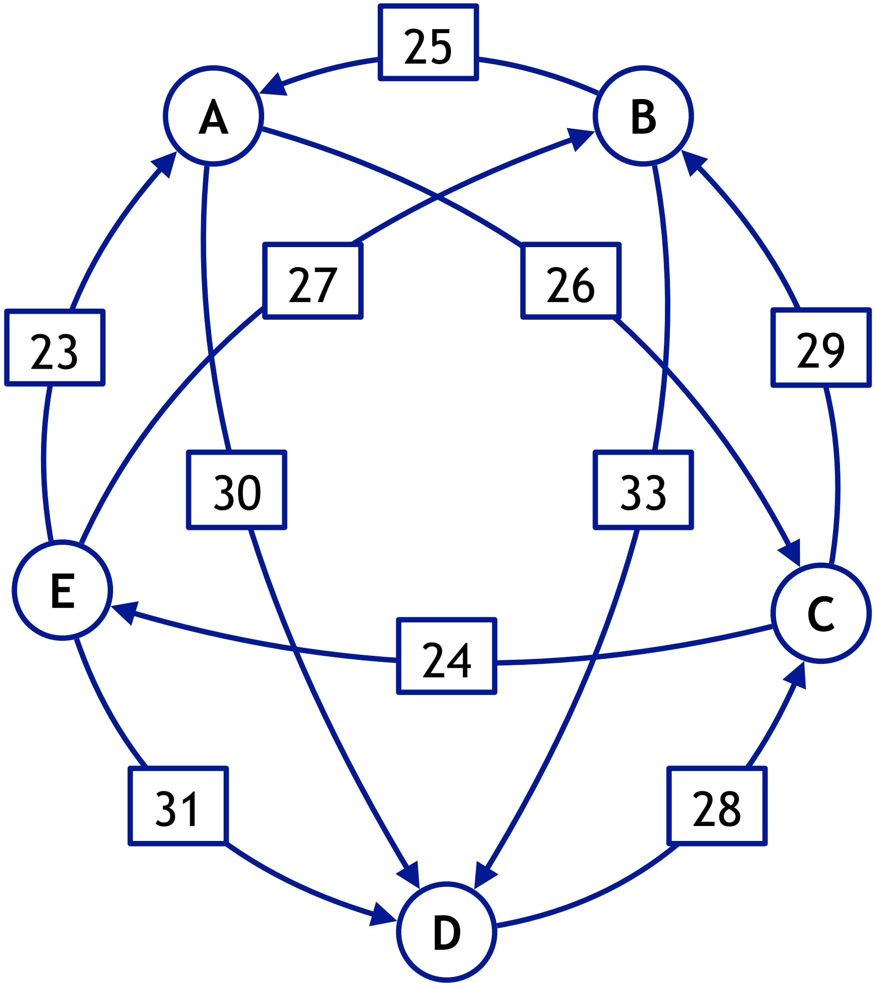
Schulze method: Example



	A	B	C	D	E
A	-	28	28	30	24
B	25	-	28	33	24
C	25	29	-	29	24
D	25	28	28	-	24
E	25	28	28	31	-

Condorcet rules

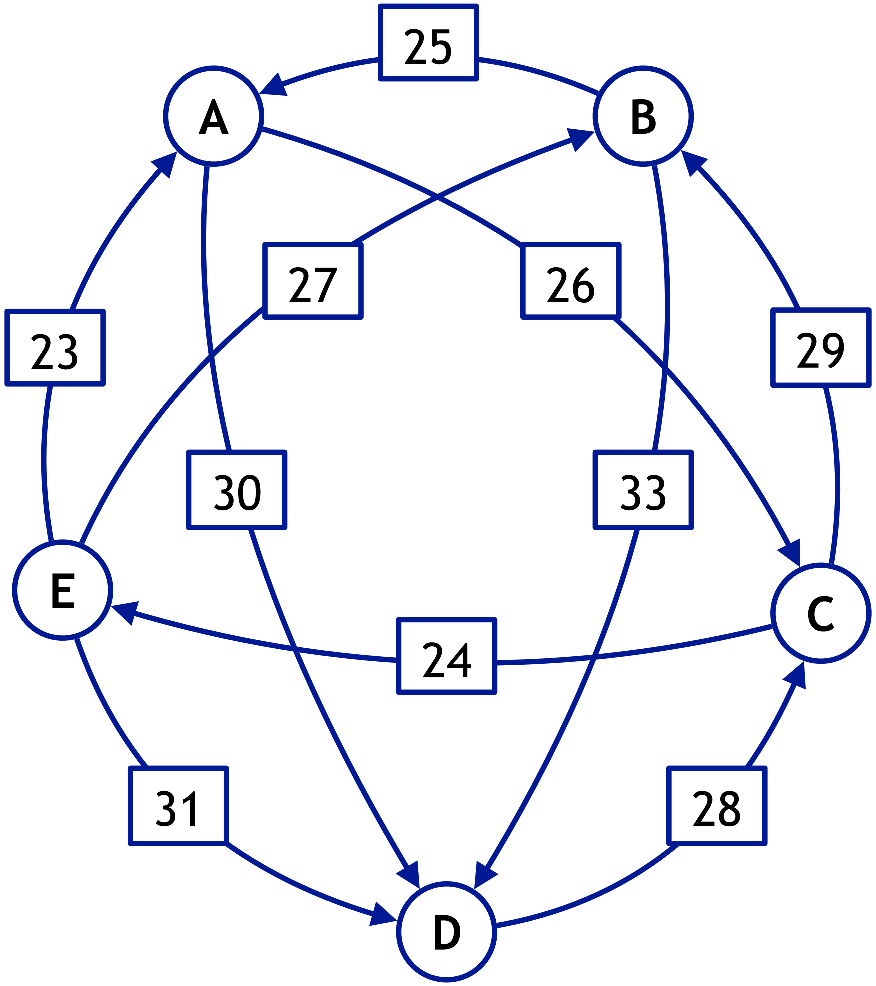
Schulze method: Example



	A	B	C	D	E
A	-	28	28	30	24
B	25	-	28	33	24
C	25	29	-	29	24
D	25	28	28	-	24
E	25	28	28	31	-

Condorcet rules

Schulze method: Example



	A	B	C	D	E
A	-	28	28	30	24
B	25	-	28	33	24
C	25	29	-	29	24
D	25	28	28	-	24
E	25	28	28	31	-

Schulze ranking:

$E \succ A \succ C \succ B \succ D$

Condorcet rules

Schulze method: Computation

Condorcet rules

Schulze method: Computation

$A[i,j,k]$: the best path from i to j using only the vertices from $\{1,2,\ldots k\}$

Condorcet rules

Schulze method: Computation

$A[i, j, k]$: the best path from i to j using only the vertices from $\{1,2,\ldots k\}$

$$A[i, j, k+1] = \max\Big(\min\big(A[i, k+1, k], A[k+1, i, k]\big),\ \ A[i, j, k]\Big)$$

Condorcet rules

Schulze method: Computation

$A[i,j,k]$: the best path from i to j using only the vertices from $\{1,2,\ldots k\}$

$$A[i,j,k+1] = \max\left(\min\left(A[i,k+1,k], A[k+1,i,k]\right),\ A[i,j,k]\right)$$

An dynamic-programming algorithm running in $O(nm+m^3)$.

Condorcet rules

Schulze method: Computation

$A[i, j, k]$: the best path from i to j using only the vertices from $\{1, 2, \ldots k\}$

$$A[i, j, k+1] = \max \left(\min \left(A[i, k+1, k], A[k+1, i, k]\right), \; A[i, j, k] \right)$$

An dynamic-programming algorithm running in $O(nm + m^3)$.

(in fact can be even improved to $O\left(nm + m^{2.69}\right)$

Krzysztof Sornat, Virginia Vassilevska Williams, Yinzhan Xu: Fine-Grained Complexity and Algorithms for the Schulze Voting Method. EC 2021: 841-859

Condorcet rules: other approaches

Top cycle: a generalisation of the Condorcet winner

Condorcet rules: other approaches

Top cycle: a generalisation of the Condorcet winner

A dominant set is a nonempty subset $A \subseteq C$ such that every candidate from A wins a head-to-head comparison with every candidate from $C \backslash A$.

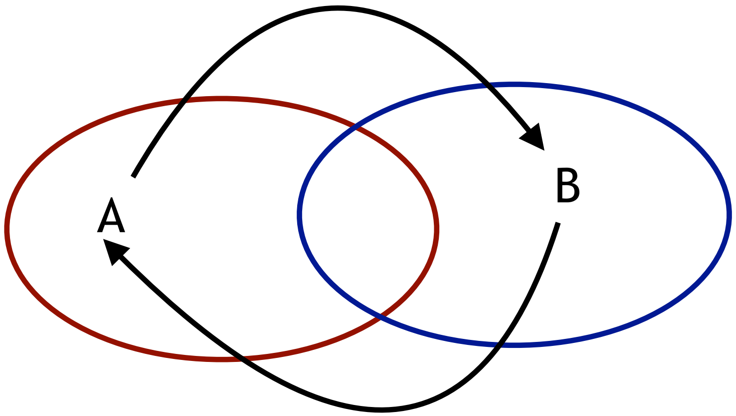
Condorcet rules: other approaches

Top cycle: a generalisation of the Condorcet winner

A dominant set is a nonempty subset $A \subseteq C$ such that every candidate from A wins a head-to-head comparison with every candidate from $C \backslash A$.

Dominant sets can be ordered by inclusion.

Proof:



Top cycle: the smallest dominating set.

Condorcet rules: other approaches

Theorem: Schulze method always selects a member of the top cycle.

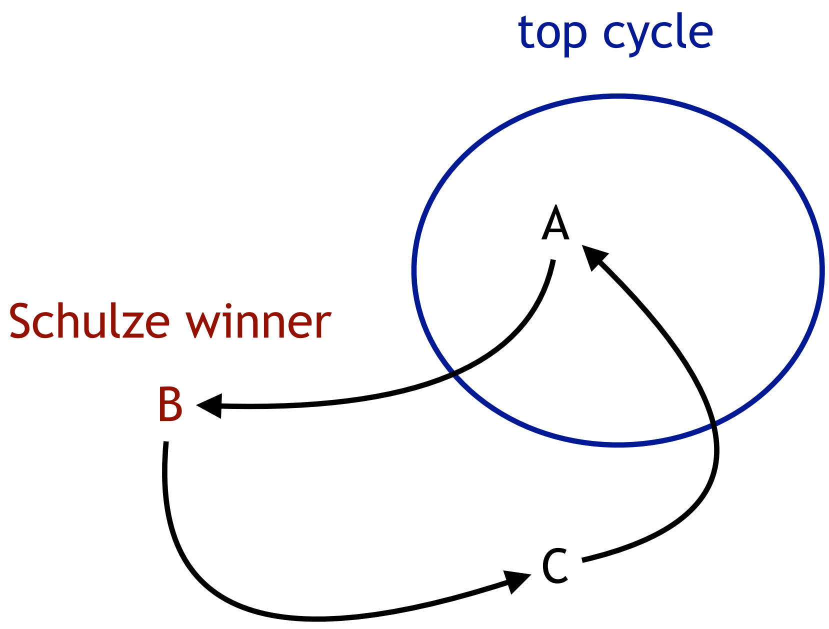
Condorcet rules: other approaches

Theorem: Schulze method always selects a member of the top cycle.

$\mathrm{str}(A \to B) > n/2.$

The strongest path from B to A must cross the boarder of the top cycle.

Thus, $\mathrm{str}(B \to A) < n/2.$



Condorcet rules

Schulze method: list of users from Wikipedia in 2024 (only selected)

Cities: Silla, Turin, San Donà di Piave, London Borough of Southwark

Political parties: Five Star Movement, Pirate Party (in many countries), Volt Party

Organisations: Annodex Association, Berufsverband der Kinder- und Jugendärzte (BVKJ), BoardGameGeek, Cloud Foundry Foundation, County Highpointers, Dapr, Debian, EuroBillTracker, European Democratic Education Community (EUDEC), FFmpeg, Free Geek, Free Hardware Foundation of Italy, Gentoo Foundation, GNU Privacy Guard (GnuPG), Haskell, Homebrew, Internet Corporation for Assigned Names and Numbers (ICANN) (until 2023), Kanawha Valley Scrabble Club, KDE e.V., Knight Foundation, Kubernetes, Kumoricon, League of Professional System Administrators (LOPSA), LiquidFeedback, Madisonium, Metalab, MTV, Neo, Noisebridge, OpenEmbedded, Open Neural Network Exchange, OpenStack, OpenSwitch, RLLMUK, Squeak, Students for Free Culture, Sugar Labs, Sverok, TopCoder, Ubuntu, Vidya Gaem Awards, Wikimedia (2008), Wikipedia in French, Hebrew, Hungarian, Russian, and Persian.

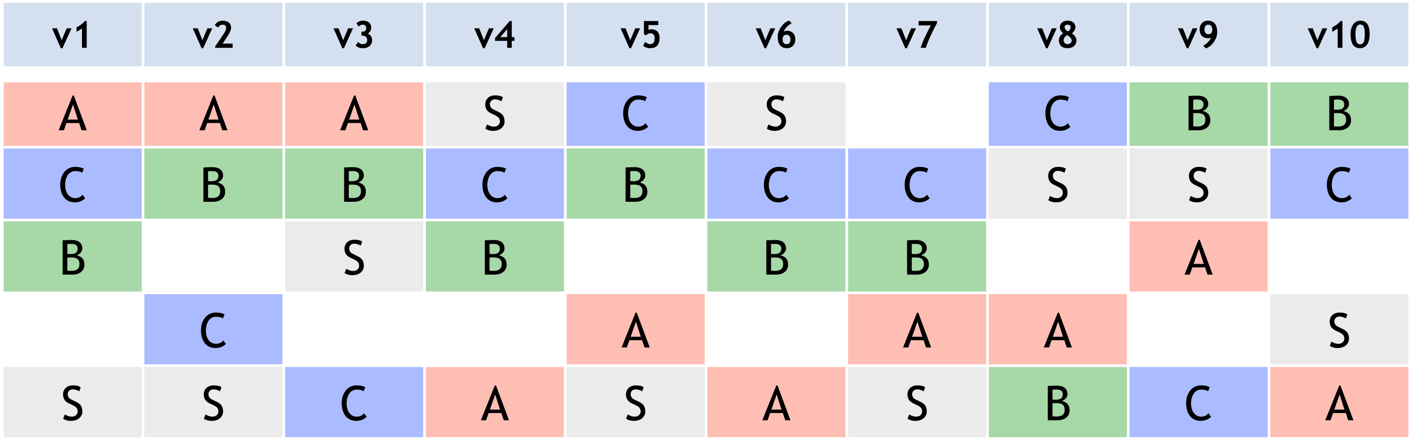
Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	V	C	B	B
C	B	B	C	B	C	C	S	S	C
B	V	S	B	V	B	B	V	A	V
V	C	V	V	A	V	A	A	V	S
S	S	C	A	S	A	S	B	C	A

Single Transferrable Vote

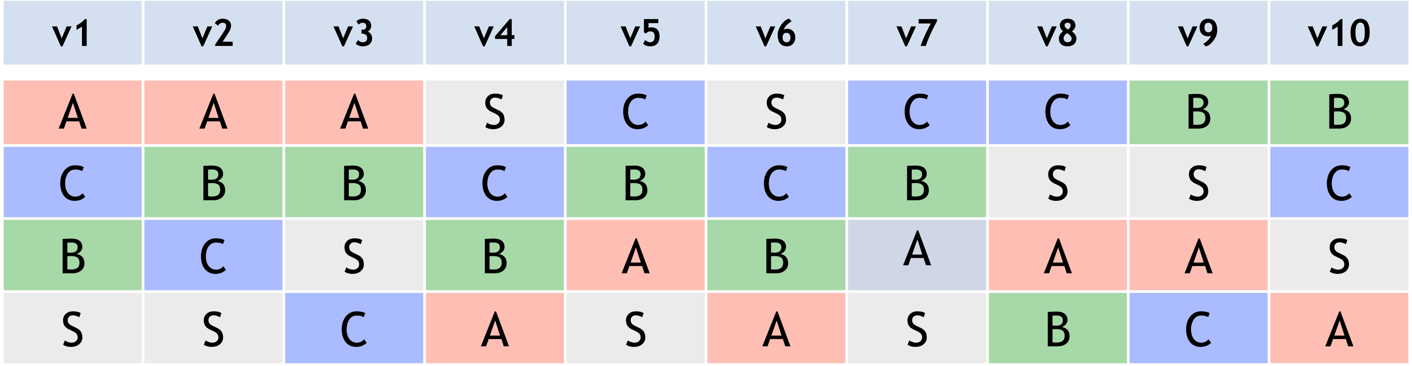
In each round we eliminate the candidate with the lowest number of top-votes.



v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S		C	B	B
C	B	B	C	B	C	C	S	S	C
B		S	B		B	B		A	
	C			A		A	A		S
S	S	C	A	S	A	S	B	C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are “transferred” accordingly.



v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	S	C	S	C	C	B	B
C	B	B	C	B	C	B	S	S	C
B	C	S	B	A	B	A	A	A	S
S	S	C	A	S	A	S	B	C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are "transferred" accordingly.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A		C		C	C	B	B
C	B	B	C	B	C	B			C
B	C		B	A	B	A	A	A	
		C	A		A		B	C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are “transferred” accordingly.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	C	C	C	C	C	B	B
C	B	B	B	B	B	B	A	A	C
B	C	C	A	A	A	A	B	C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are “transferred” accordingly.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	C	C	C	C	C		
C							A	A	C
	C	C	A	A	A	A		C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are “transferred” accordingly.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	C	C	C	C	C	A	C
C	C	C	A	A	A	A	A	C	A

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are “transferred” accordingly.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
			C	C	C	C	C		C
C	C	C						C	

Single Transferrable Vote

In each round we eliminate the candidate with the lowest number of top-votes. The votes are "transferred" accordingly.

C is a winner.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
C	C	C	C	C	C	C	C	C	C

Rules, summary

Positional scoring rules:

- Plurality, $\alpha = (1, 0, 0, \ \ldots, 0)$,
- Borda, $\alpha = (m - 1, m - 2, m - 3, \ \ldots, 0)$,
- Veto, $\alpha = (1, 1, 1, \ \ldots, 1, \ 0)$.

Condorcet rules:

- Copeland (perhaps simplest)
- Schulze rule (many good properties)

Rules based on iterative elimination:

- Single Transferable Vote (STV)
- Baldwin (a candidate with the lowest Borda score is eliminated)

So which rule is the best?

Axiomatic approach: formulate desired properties and check if they are satisfied.

So which rule is the best?

Axiomatic approach: formulate desired properties and check if they are satisfied.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	A	B	B	B	B	B	B
B	B	B	B	A	A	A	A	A	A

Consider the Plurality rule. Which candidate would won?

So which rule is the best?

Axiomatic approach: formulate desired properties and check if they are satisfied.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	A	B	B	B	B	B	B
B	B	B	B	A	A	A	A	A	A

Consider the Plurality rule. Which candidate would won? What happens
If a similar candidate to B runs in an election?

So which rule is the best?

Axiomatic approach: formulate desired properties and check if they are satisfied.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	A	C	C	C	B	B	B
C	C	B	B	B	B	B	C	C	C
B	B	C	C	A	A	A	A	A	A

Consider the Plurality rule. Which candidate would won? What happens If a similar candidate to B runs in an election?

So which rule is the best?

Axiomatic approach: formulate desired properties and check if they are satisfied.

Clone: a set of candidates that each voter ranks consecutively (e.g., B and C).

A rule is **cloneproof** if removing any copy in a clone does not increase nor decrease the chances of the clone in the election.

v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	A	A	A	C	C	C	B	B	B
C	C	B	B	B	B	B	C	C	C
B	B	C	C	A	A	A	A	A	A

Consider the Plurality rule. Which candidate would won? What happens
If a similar candidate to B runs in an election?

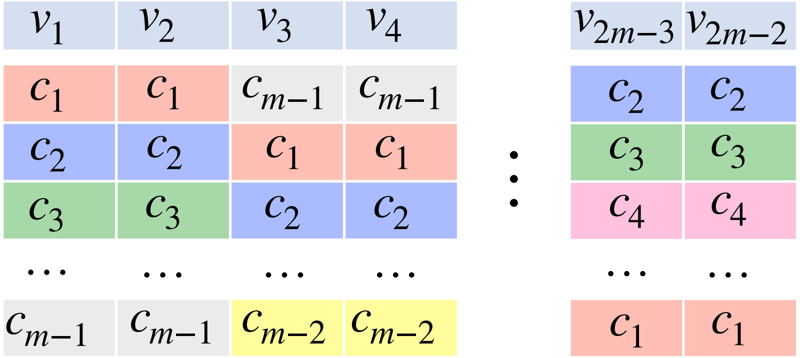
So which rule is the best?

Positional scoring rules are in general not cloneproof.

So which rule is the best?

Positional scoring rules are in general not cloneproof.

Consider a cyclic preference profile with $m-1$ candidates.



v_1	v_2	v_3	v_4		v_{2m-3}	v_{2m-2}
c_1	c_1	c_{m-1}	c_{m-1}		c_2	c_2
c_2	c_2	c_1	c_1	$\vdots$	c_3	c_3
c_3	c_3	c_2	c_2		c_4	c_4
$\dots$	$\dots$	$\dots$	$\dots$		$\dots$	$\dots$
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}		c_1	c_1

So which rule is the best?

Positional scoring rules are in general not cloneproof.

Consider a cyclic preference profile with $m-1$ candidates.

Each candidate is ranked once on each position, so all candidates are winning.

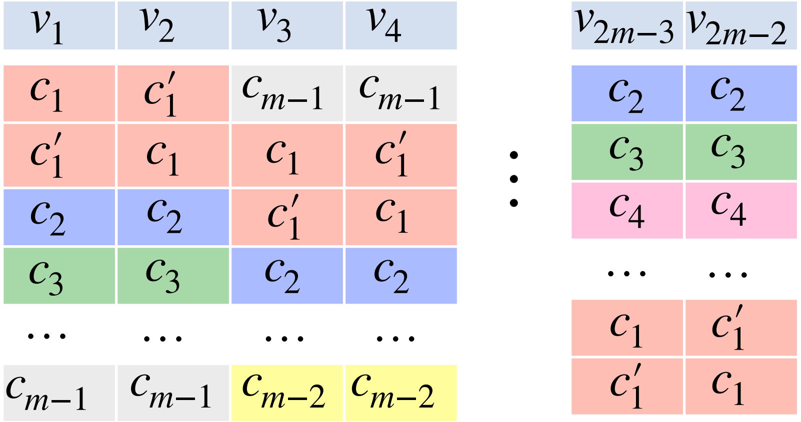
v_1	v_2	v_3	v_4		v_{2m-3}	v_{2m-2}
c_1	c_1	c_{m-1}	c_{m-1}		c_2	c_2
c_2	c_2	c_1	c_1	$\vdots$	c_3	c_3
c_3	c_3	c_2	c_2		c_4	c_4
$\dots$	$\dots$	$\dots$	$\dots$		$\dots$	$\dots$
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}		c_1	c_1

So which rule is the best?

Positional scoring rules are in general not cloneproof.

Consider a cyclic preference profile with $m-1$ candidates.

Each candidate is ranked once on each position, so all candidates are winning.



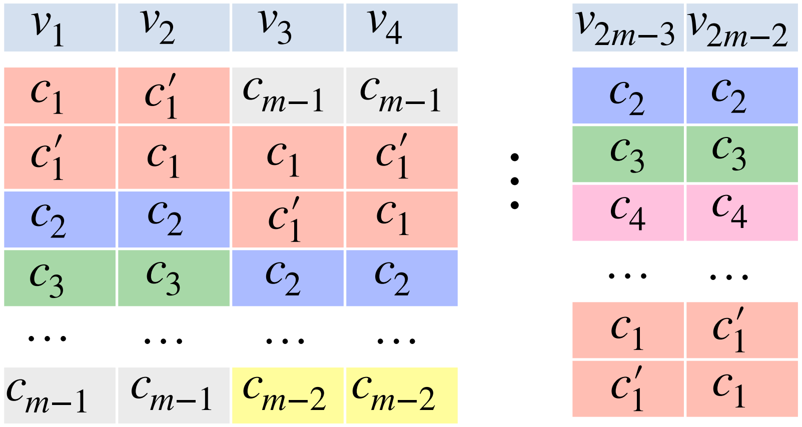
v_1	v_2	v_3	v_4		v_{2m-3}	v_{2m-2}
c_1	c_1'	c_{m-1}	c_{m-1}		c_2	c_2
c_1'	c_1	c_1	c_1'	$\vdots$	c_3	c_3
c_2	c_2	c_1'	c_1		c_4	c_4
c_3	c_3	c_2	c_2		$\dots$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$		c_1	c_1'
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}		c_1'	c_1

What if we clone c_1?

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$



v_1	v_2	v_3	v_4
c_1	c_1'	c_{m-1}	c_{m-1}
c_1'	c_1	c_1	c_1'
c_2	c_2	c_1'	c_1
c_3	c_3	c_2	c_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}

$\vdots$

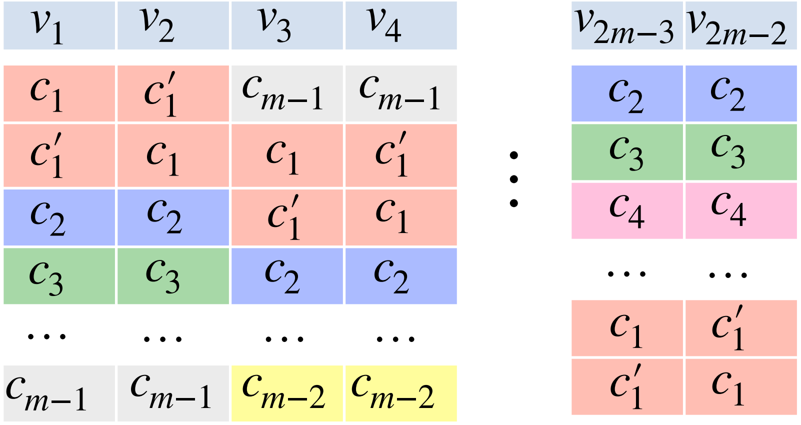
v_{2m-3}	v_{2m-2}
c_2	c_2
c_3	c_3
c_4	c_4
$\ldots$	$\ldots$
c_1	c_1'
c_1'	c_1

What if we clone c_1?

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$



v_1	v_2	v_3	v_4
c_1	c_1'	c_{m-1}	c_{m-1}
c_1'	c_1	c_1	c_1'
c_2	c_2	c_1'	c_1
c_3	c_3	c_2	c_2
…	…	…	…
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}

⋮

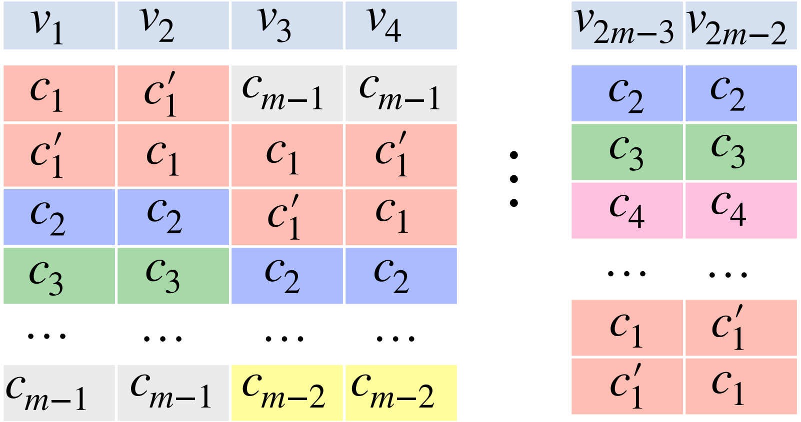
v_{2m-3}	v_{2m-2}
c_2	c_2
c_3	c_3
c_4	c_4
…	…
c_1	c_1'
c_1'	c_1

What if we clone c_1?

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$
- candidate c_{m-2}: $(2, 2, 2, \ldots, 2, 0, 2, 2)$
- etc.



v_1	v_2	v_3	v_4
c_1	c_1'	c_{m-1}	c_{m-1}
c_1'	c_1	c_1	c_1'
c_2	c_2	c_1'	c_1
c_3	c_3	c_2	c_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$
c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}

$\vdots$

v_{2m-3}	v_{2m-2}
c_2	c_2
c_3	c_3
c_4	c_4
$\ldots$	$\ldots$
c_1	c_1'
c_1'	c_1

What if we clone c_1?

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$
- candidate c_{m-2}: $(2, 2, 2, \ldots, 2, 0, 2, 2)$
- etc.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$
- candidate c_{m-2}: $(2, 2, 2, \ldots, 2, 0, 2, 2)$
- etc.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

Candidate c_{m-1} will loose if $\alpha_1 + \alpha_m < 2\alpha_{m-1}$. **(cloning will help)**

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$
- candidate c_{m-2}: $(2, 2, 2, \ldots, 2, 0, 2, 2)$
- etc.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

Candidate c_{m-1} will loose if $\alpha_1 + \alpha_m < 2\alpha_{m-1}$. **(cloning will help)**

For cloneproofness we need $\alpha_1 + \alpha_m = 2\alpha_{m-1}$.

So which rule is the best?

Vector of occurrences on positions for:

- clones: $(1, 2, 2, \ldots, 2, 1)$
- candidate c_{m-1}: $(2, 2, 2, \ldots, 2, 2, 0, 2)$
- candidate c_{m-2}: $(2, 2, 2, \ldots, 2, 0, 2, 2)$
- etc.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

Candidate c_{m-1} will loose if $\alpha_1 + \alpha_m < 2\alpha_{m-1}$. **(cloning will help)**

For cloneproofness we need $\alpha_1 + \alpha_m = 2\alpha_{m-1}$.

Analogously: we need $\alpha_1 + \alpha_m = 2\alpha_{m-2}$, etc.

So which rule is the best?

Intuition: If a scoring vector is convex (like Plurality), then introducing a similar clone typically hurts a candidate. If the vector is very concave (veto), then the candidate could benefit from introducing a clone.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

Candidate c_{m-1} will loose if $\alpha_1 + \alpha_m < 2\alpha_{m-1}$. **(cloning will help)**

For cloneproofness we need $\alpha_1 + \alpha_m = 2\alpha_{m-1}$.

Analogously: we need $\alpha_1 + \alpha_m = 2\alpha_{m-2}$, etc.

So which rule is the best?

Intuition: If a scoring vector is convex (like Plurality), then introducing a similar clone typically hurts a candidate. If the vector is very concave (veto), then the candidate could benefit from introducing a clone.

Candidate c_{m-1} will win if $\alpha_1 + \alpha_m > 2\alpha_{m-1}$. **(cloning will hurt)**

Candidate c_{m-1} will loose if $\alpha_1 + \alpha_m < 2\alpha_{m-1}$. **(cloning will help)**

For cloneproofness we need $\alpha_1 + \alpha_m = 2\alpha_{m-1}$.

Analogously: we need $\alpha_1 + \alpha_m = 2\alpha_{m-2}$, etc.

Thus, $\alpha_2 = \alpha_3 = \ldots = a_{m-1}$.

So which rule is the best?

Thus, $\alpha_2 = \alpha_3 = \ldots = a_{m-1}$ and $\alpha_1 + \alpha_m = \alpha_2$.

The scoring rules that satisfy this are: $(\alpha + c, \alpha, \alpha, \ldots, \alpha, \alpha - c)$.

So which rule is the best?

Thus, $\alpha_2 = \alpha_3 = \ldots = a_{m-1}$ and $\alpha_1 + \alpha_m = \alpha_2$.

The scoring rules that satisfy this are: $(\alpha + c, \alpha, \alpha, \ldots, \alpha, \alpha - c)$.

But for this rule, the candidate can benefit by introducing a “slightly worse” clone.

v_1	v_2	v_3	v_4	v_5	v_6
c_1	c_1	c_1'	c_{m-1}	c_{m-1}	c_{m-1}
c_1'	c_1'	c_1	c_1	c_1	c_1'
c_2	c_2	c_2	c_1'	c_1'	c_1
c_3	c_3	c_3	c_2	c_2	c_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
c_{m-1}	c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}	c_{m-2}

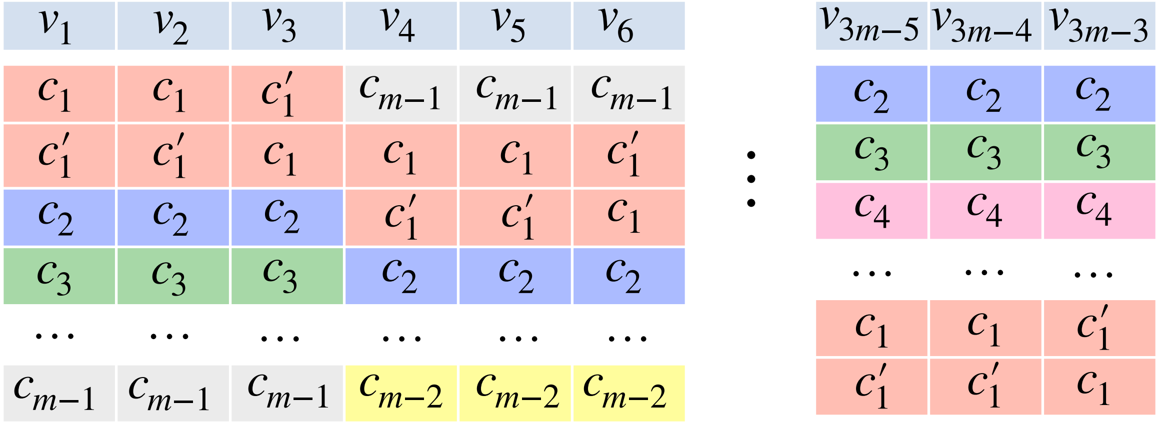
$\vdots$

v_{3m-5}	v_{3m-4}	v_{3m-3}
c_2	c_2	c_2
c_3	c_3	c_3
c_4	c_4	c_4
$\cdots$	$\cdots$	$\cdots$
c_1	c_1	c_1'
c_1'	c_1'	c_1

So which rule is the best?

Vector of occurrences on positions for:

- better clone: $(2, 3, 3, \ldots, 3, 1)$



v_1	v_2	v_3	v_4	v_5	v_6
c_1	c_1	c_1'	c_{m-1}	c_{m-1}	c_{m-1}
c_1'	c_1'	c_1	c_1	c_1	c_1'
c_2	c_2	c_2	c_1'	c_1'	c_1
c_3	c_3	c_3	c_2	c_2	c_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
c_{m-1}	c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}	c_{m-2}

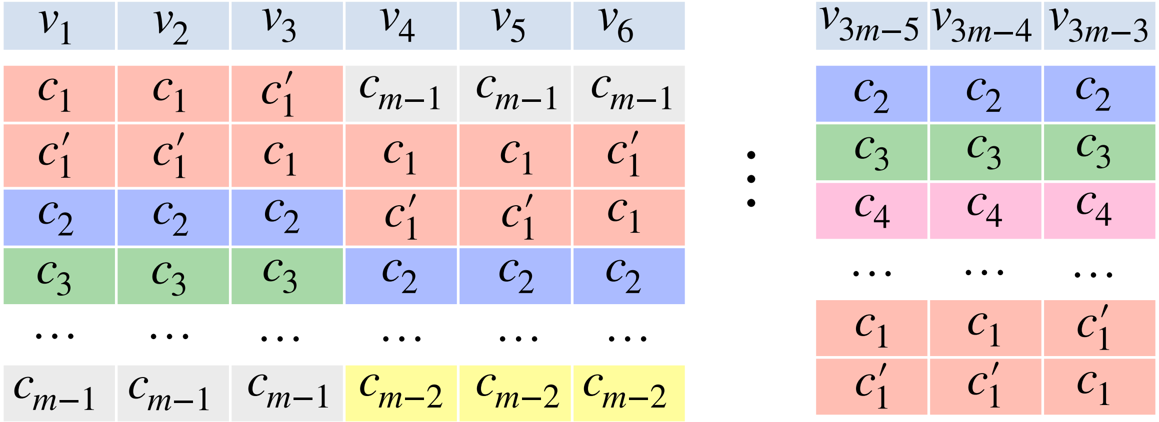
$\vdots$

v_{3m-5}	v_{3m-4}	v_{3m-3}
c_2	c_2	c_2
c_3	c_3	c_3
c_4	c_4	c_4
$\ldots$	$\ldots$	$\ldots$
c_1	c_1	c_1'
c_1'	c_1'	c_1

So which rule is the best?

Vector of occurrences on positions for:

- better clone: $(2, 3, 3, \ldots, 3, 1)$
- candidate c_{m-1}: $(3, 3, 3, \ldots, 3, 3, 0, 3)$



v_1	v_2	v_3	v_4	v_5	v_6
c_1	c_1	c_1'	c_{m-1}	c_{m-1}	c_{m-1}
c_1'	c_1'	c_1	c_1	c_1	c_1'
c_2	c_2	c_2	c_1'	c_1'	c_1
c_3	c_3	c_3	c_2	c_2	c_2
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
c_{m-1}	c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}	c_{m-2}

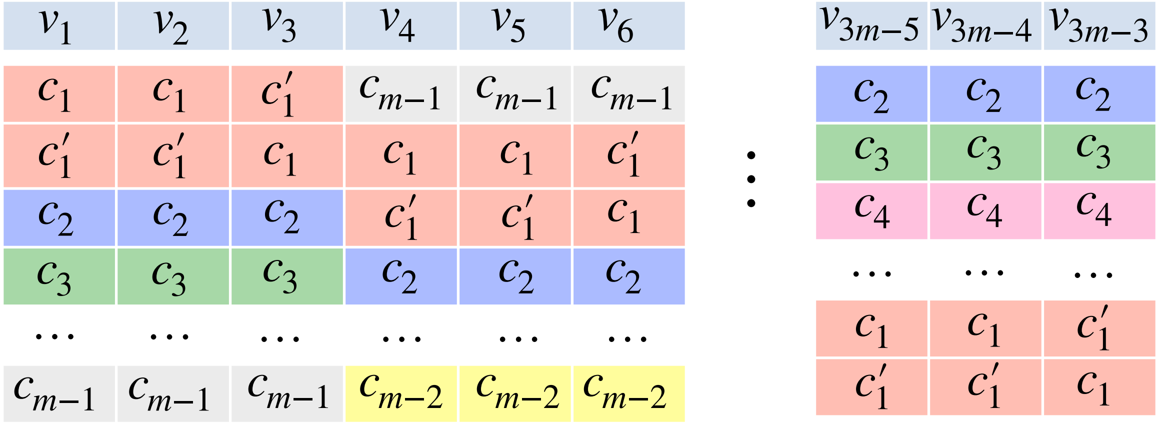
$\vdots$

v_{3m-5}	v_{3m-4}	v_{3m-3}
c_2	c_2	c_2
c_3	c_3	c_3
c_4	c_4	c_4
$\cdots$	$\cdots$	$\cdots$
c_1	c_1	c_1'
c_1'	c_1'	c_1

So which rule is the best?

Vector of occurrences on positions for:

- better clone: $(2, 3, 3, \ldots, 3, 1)$
- candidate c_{m-1}: $(3, 3, 3, \ldots, 3, 3, 0, 3)$
- candidate c_{m-2}: $(3, 3, 3, \ldots, 3, 0, 3, 3)$
- etc.



v_1	v_2	v_3	v_4	v_5	v_6
c_1	c_1	c_1'	c_{m-1}	c_{m-1}	c_{m-1}
c_1'	c_1'	c_1	c_1	c_1	c_1'
c_2	c_2	c_2	c_1'	c_1'	c_1
c_3	c_3	c_3	c_2	c_2	c_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
c_{m-1}	c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}	c_{m-2}

$\vdots$

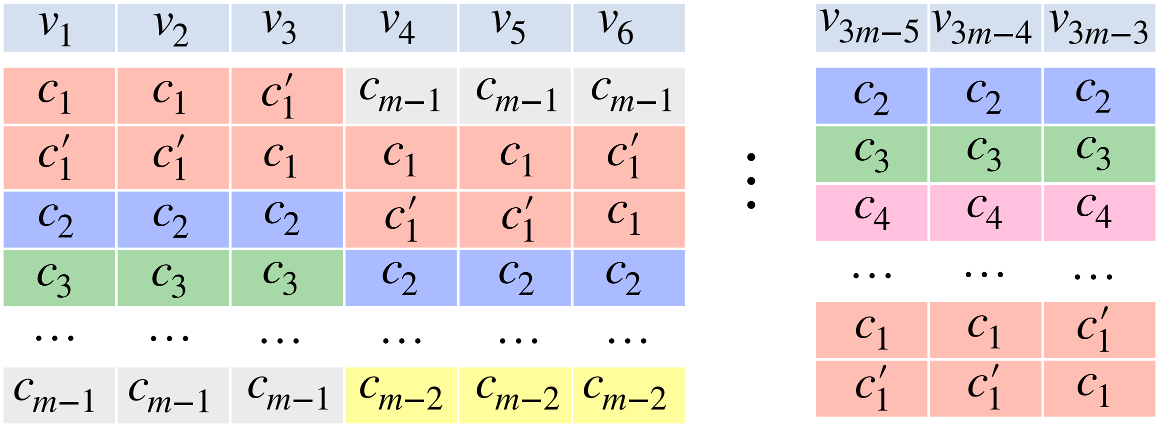
v_{3m-5}	v_{3m-4}	v_{3m-3}
c_2	c_2	c_2
c_3	c_3	c_3
c_4	c_4	c_4
$\ldots$	$\ldots$	$\ldots$
c_1	c_1	c_1'
c_1'	c_1'	c_1

So which rule is the best?

Vector of occurrences on positions for:

- better clone: $(2, 3, 3, \ldots, 3, 1)$
- candidate c_{m-1}: $(3, 3, 3, \ldots, 3, 3, 0, 3)$
- candidate c_{m-2}: $(3, 3, 3, \ldots, 3, 0, 3, 3)$
- etc.

Clone will win with c_{m-1} because $\alpha_1 + 2\alpha_m = 3\alpha - c < 3\alpha = 3\alpha_{m-1}$.



v_1	v_2	v_3	v_4	v_5	v_6
c_1	c_1	c_1'	c_{m-1}	c_{m-1}	c_{m-1}
c_1'	c_1'	c_1	c_1	c_1	c_1'
c_2	c_2	c_2	c_1'	c_1'	c_1
c_3	c_3	c_3	c_2	c_2	c_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
c_{m-1}	c_{m-1}	c_{m-1}	c_{m-2}	c_{m-2}	c_{m-2}

$\vdots$

v_{3m-5}	v_{3m-4}	v_{3m-3}
c_2	c_2	c_2
c_3	c_3	c_3
c_4	c_4	c_4
$\ldots$	$\ldots$	$\ldots$
c_1	c_1	c_1'
c_1'	c_1'	c_1

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

Assume in round r in one instance we eliminated the last clone. Then in the other instance we will eliminate all other clones.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

Assume in round r in one instance we eliminated the last clone. Then in the other instance we will eliminate all other clones. Thus, either all instances have clones or none.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

Assume in round r in one instance we eliminated the last clone. Then in the other instance we will eliminate all other clones. Thus, either all instances have clones or none.

Assume in round r in one instance we eliminated a no-clone candidate c.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to
each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

Assume in round r in one instance we eliminated the last clone. Then in the other instance we will eliminate all other clones. Thus, either all instances have clones or none.

Assume in round r in one instance we eliminated a no-clone candidate c. Then the next no-clone eliminated in the other instance will be c.

So which rule is the best?

Single Transferable Rule (STV) is cloneproof.

The sequence of eliminating the no-clone candidates is the same before and after removing the clone.

By induction (assume lexicographic tie-breaking, where clones are next to each other in the tie-breaking order):

I_o : original instance, I_r : an instance after removing one of the clones.

Assume in round r in one instance we eliminated the last clone. Then in the other instance we will eliminate all other clones. Thus, either all instances have clones or none.

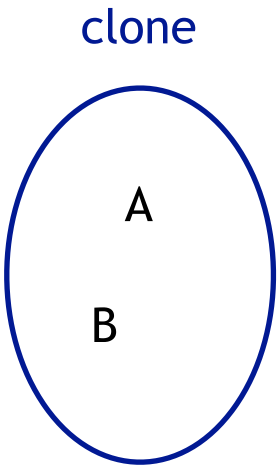
Assume in round r in one instance we eliminated a no-clone candidate c.

Then the next no-clone eliminated in the other instance will be c.

(They have the same number of top-votes.)

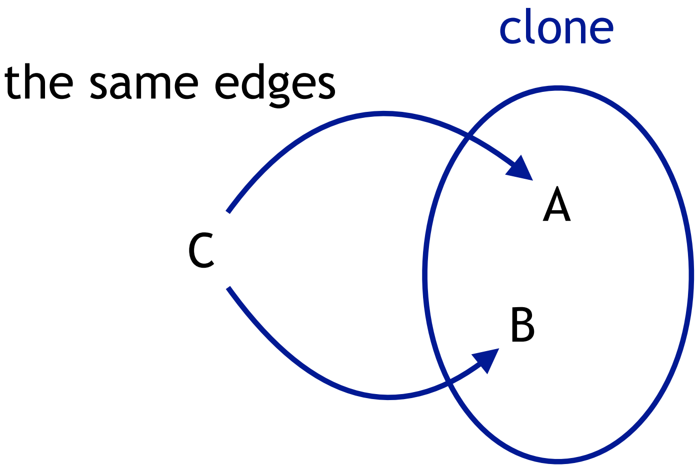
So which rule is the best?

Schulze method is cloneproof.



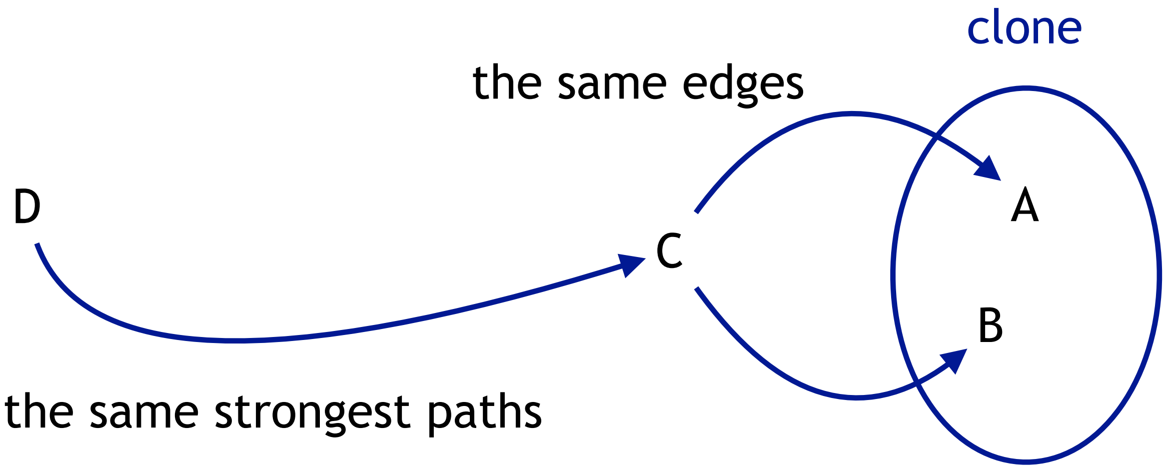
So which rule is the best?

Schulze method is cloneproof.



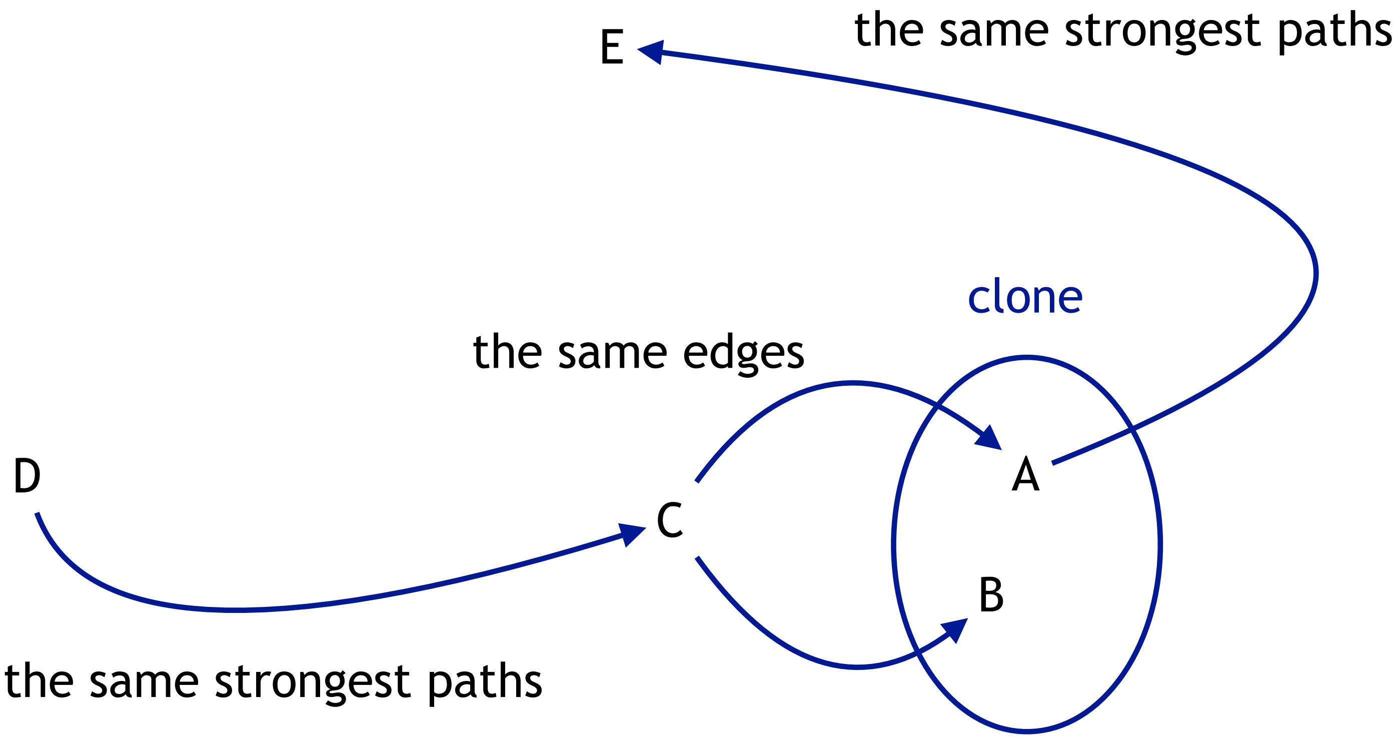
So which rule is the best?

Schulze method is cloneproof.



So which rule is the best?

Schulze method is cloneproof.



So which rule is the best?

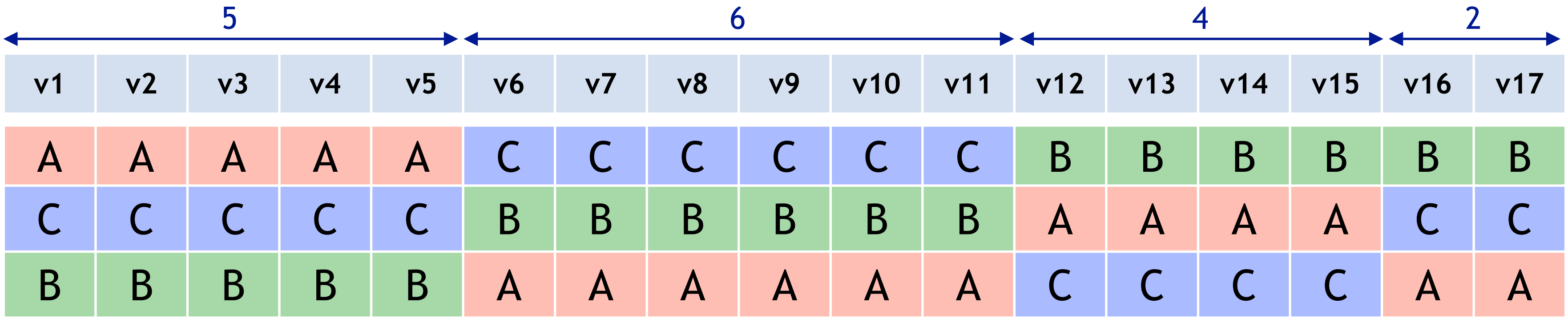
STV and Schulze method are cloneproof. How about other axioms?

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate A will be eliminated first.



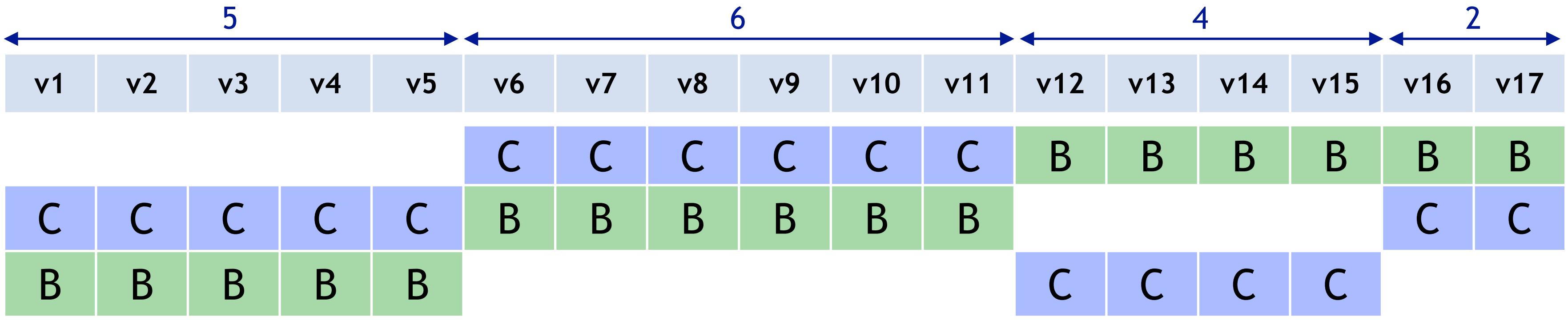
5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	B	B
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	C	C
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate A will be eliminated first.



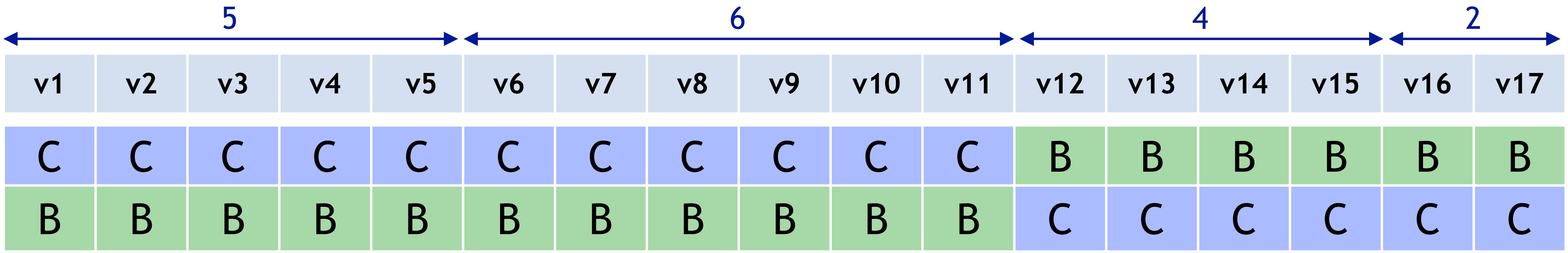
5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
					C	C	C	C	C	C	B	B	B	B	B	B
C	C	C	C	C	B	B	B	B	B	B					C	C
B	B	B	B	B							C	C	C	C		

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate A will be eliminated first.



5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
C	C	C	C	C	C	C	C	C	C	C	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	C	C	C	C	C	C

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate A will be eliminated first.
- Candidate B will be eliminated next, and so **candidate C wins the election!**

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
C	C	C	C	C	C	C	C	C	C	C	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	C	C	C	C	C	C

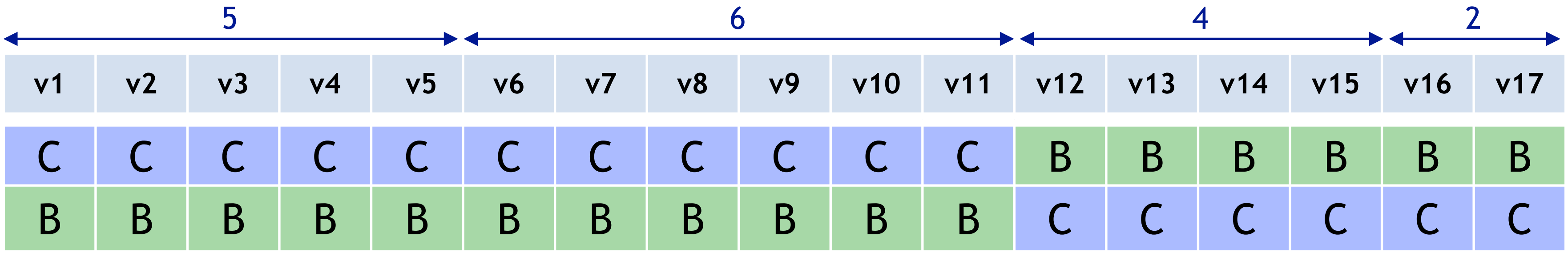
So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate A will be eliminated first.
- Candidate B will be eliminated next, and so **candidate C wins the election!**

What happens if the last two voters increase their support for C?



v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
C	C	C	C	C	C	C	C	C	C	C	B	B	B	B	B	B
B	B	B	B	B	B	B	B	B	B	B	C	C	C	C	C	C

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

What happens if the last two voters increase their support for C?

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	B	B
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	C	C
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

What happens if the last two voters increase their support for C?

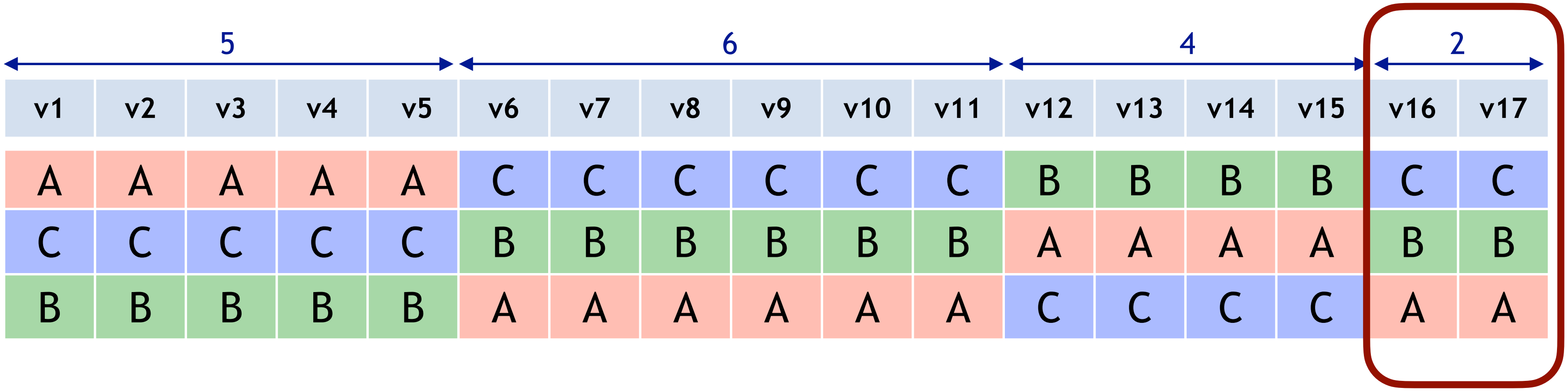
5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	B	B
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	C	C
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

What happens if the last two voters increase their support for C?



5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	C	C
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	B	B
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate B will be eliminated first.

What happens if the last two voters increase their support for C?

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	C	C
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	B	B
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

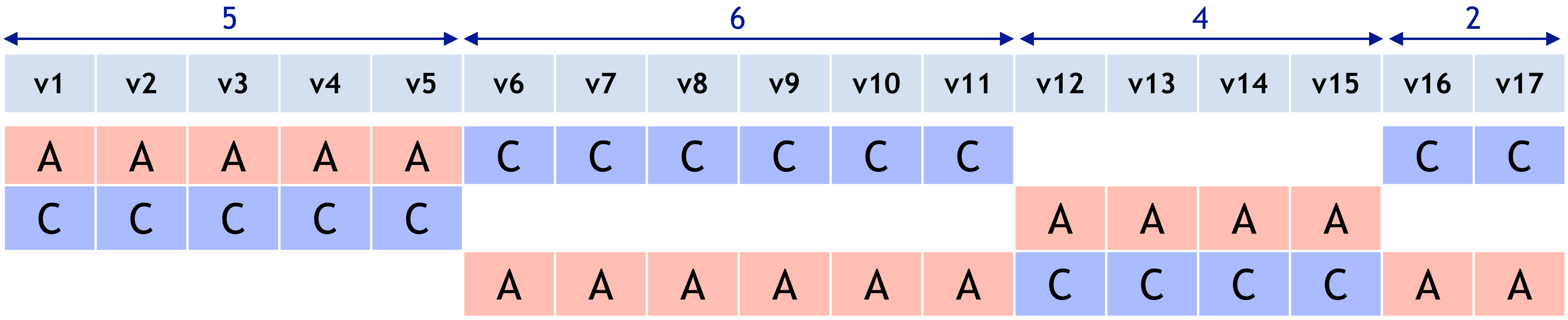
So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate B will be eliminated first.

What happens if the last two voters increase their support for C?



5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C					C	C
C	C	C	C	C							A	A	A	A		
					A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate B will be eliminated first.

What happens if the last two voters increase their support for C?

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	A	A	A	A	C	C
C	C	C	C	C	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Consider STV in this example:

- Candidate B will be eliminated first.
- Candidate C will be eliminated next, and so **candidate A wins the election!**

What happens if the last two voters increase their support for C?

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	A	A	A	A	C	C
C	C	C	C	C	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Monotonicity: if a voter pushes a winning candidate up in her ranking, then this candidate should still be winning.

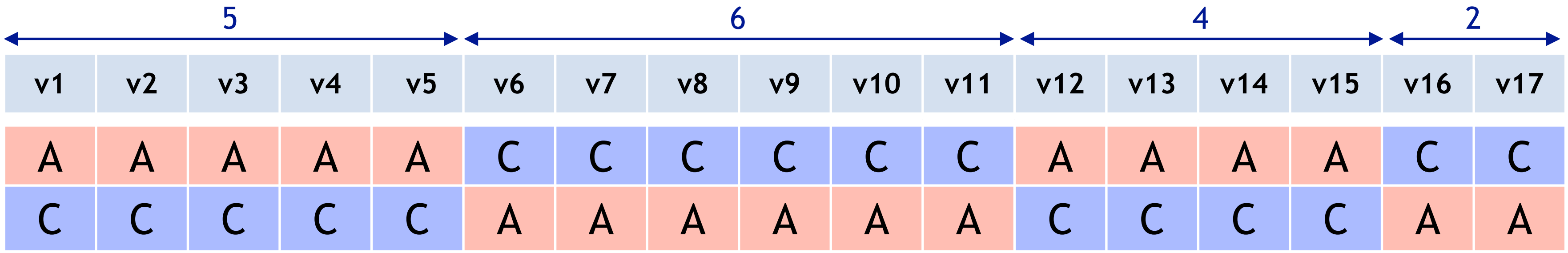
5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	A	A	A	A	C	C
C	C	C	C	C	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Monotonicity: if a voter pushes a winning candidate up in her ranking, then this candidate should still be winning.

STV is non-monotonic!



5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	A	A	A	A	C	C
C	C	C	C	C	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Theorem: Schulze method is monotonic.

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Theorem: Schulze method is monotonic.

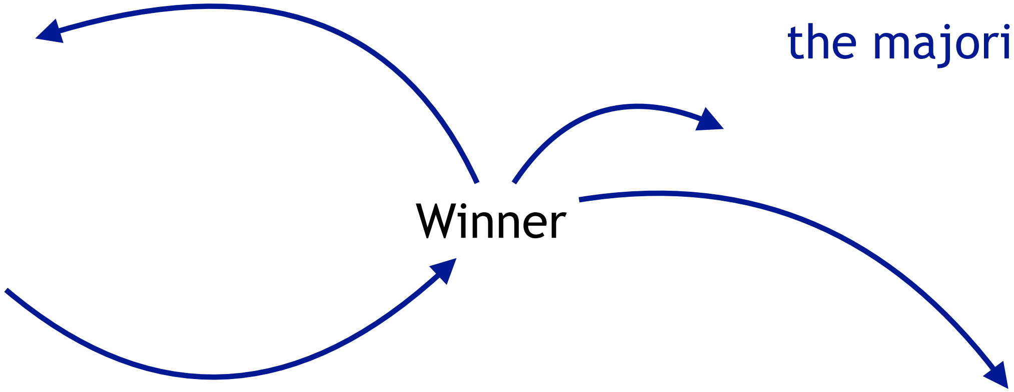
In our example it selects the ranking C ≻ B ≻ A.

5					6						4				2	
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17
A	A	A	A	A	C	C	C	C	C	C	B	B	B	B	B	B
C	C	C	C	C	B	B	B	B	B	B	A	A	A	A	C	C
B	B	B	B	B	A	A	A	A	A	A	C	C	C	C	A	A

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

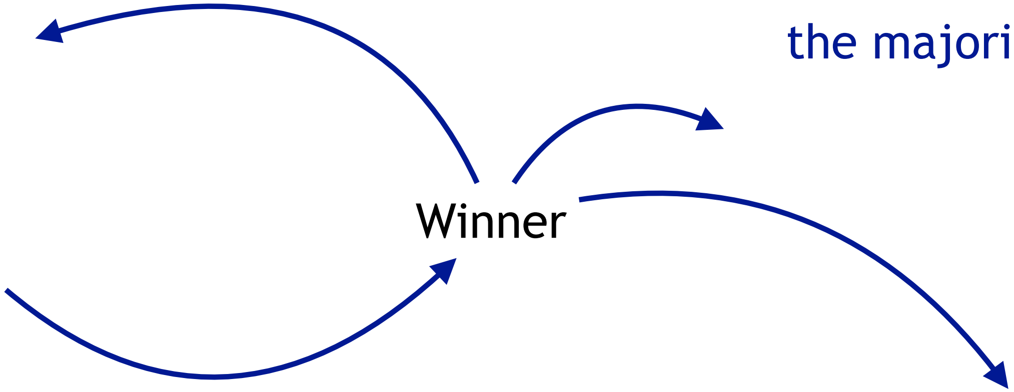
Theorem: Schulze method is monotonic.



So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Theorem: Schulze method is monotonic.



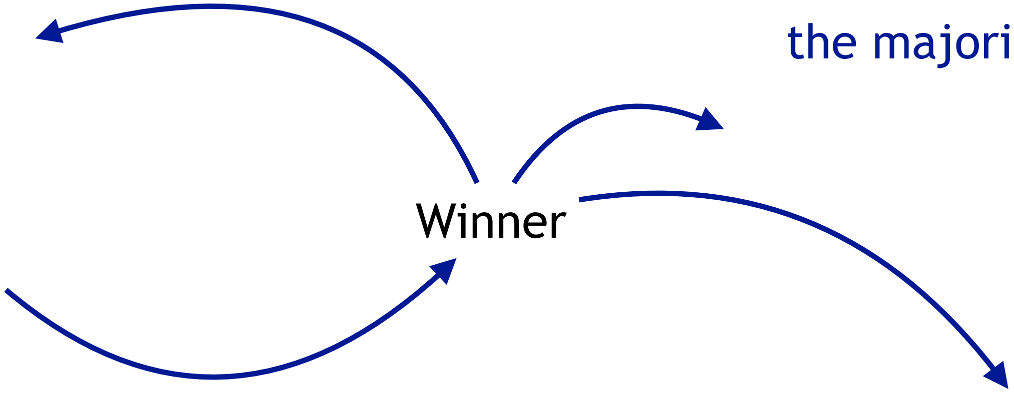
the majoritarian graph

The swap only changes the edges going in and out of the winner.

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Theorem: Schulze method is monotonic.



the majoritarian graph

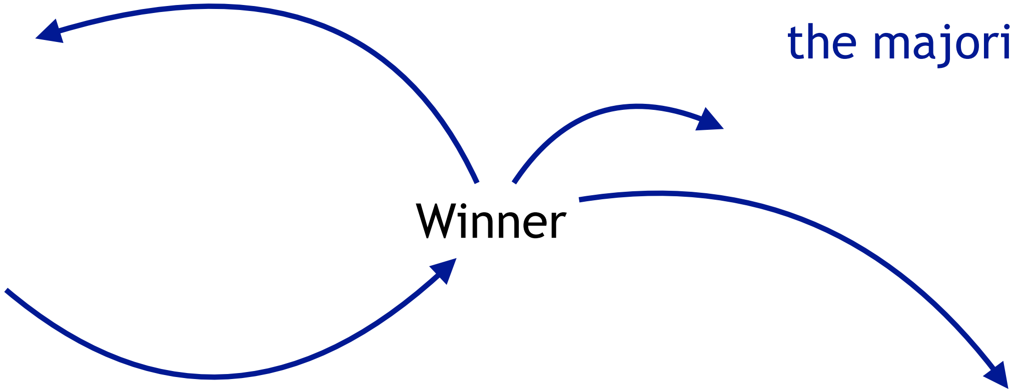
The swap only changes the edges going in and out of the winner.

Each path from the winner that does not repeat vertices can only improve.

So which rule is the best?

STV and Schulze method are cloneproof. How about other axioms?

Theorem: Schulze method is monotonic.



the majoritarian graph

The swap only changes the edges going in and out of the winner.

Each path from the winner that does not repeat vertices can only improve.

A path to the winner can only loose.

So which rule is the best?

No-show paradox

A voting rule $\mathscr{R}$ satisfies **participation** if each voter always weakly prefers voting to not voting, i.e., for each preference profile P it holds that $\mathscr{R}(P) \succ_i \mathscr{R}(P_{-i})$, where P_{-i} is the profile P with vote v_i removed.

So which rule is the best?

No-show paradox

A voting rule $\mathscr{R}$ satisfies **participation** if each voter always weakly prefers voting to not voting, i.e., for each preference profile P it holds that $\mathscr{R}(P) \succ_i \mathscr{R}(P_{-i})$, where P_{-i} is the profile P with vote v_i removed.

v1	v2	v3	v4	v5
A	B	C	B	B
B	C	B	A	C
C	A	A	C	A

Assume that A is a winner

So which rule is the best?

No-show paradox

A voting rule $\mathscr{R}$ satisfies **participation** if each voter always weakly prefers voting to not voting, i.e., for each preference profile P it holds that $\mathscr{R}(P) \succ_i \mathscr{R}(P_{-i})$, where P_{-i} is the profile P with vote v_i removed.

v1	v2	v3	v4	v5
A	B	C	B	B
B	C	B	A	C
C	A	A	C	A

Assume that A is a winner

v1	v2	v3	v4	v5	V6
A	B	C	B	B	B
B	C	B	A	C	A
C	A	A	C	A	C

So which rule is the best?

No-show paradox

A voting rule $\mathscr{R}$ satisfies **participation** if each voter always weakly prefers voting to not voting, i.e., for each preference profile P it holds that $\mathscr{R}(P) \succ_i \mathscr{R}(P_{-i})$, where P_{-i} is the profile P with vote v_i removed.

v1	v2	v3	v4	v5
A	B	C	B	B
B	C	B	A	C
C	A	A	C	A

Assume that A is a winner

v1	v2	v3	v4	v5	V6
A	B	C	B	B	B
B	C	B	A	C	A
C	A	A	C	A	C

Then here A or B must win.

So which rule is the best?

No-show paradox

A voting rule $\mathscr{R}$ satisfies **participation** if each voter always weakly prefers voting to not voting, i.e., for each preference profile P it holds that $\mathscr{R}(P) \succ_i \mathscr{R}(P_{-i})$, where P_{-i} is the profile P with vote v_i removed.

Theorem: no Condorcet rule satisfies participation.

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.

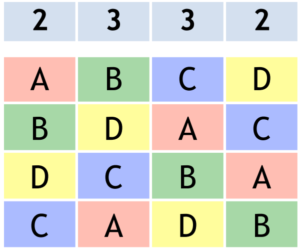
2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.

case 1: A or B wins

case 2: C or D wins



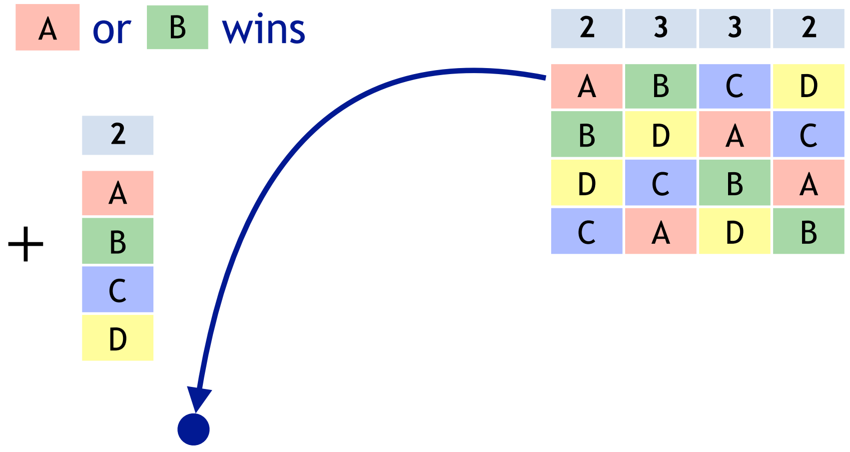
2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.

case 1: A or B wins

case 2: C or D wins



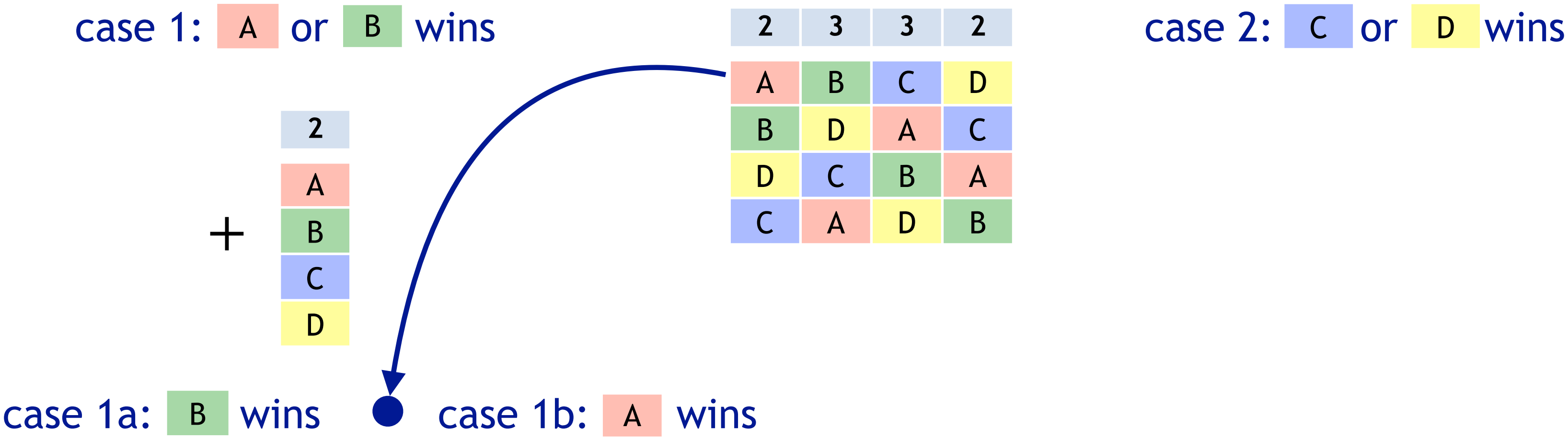
2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

+

2
A
B
C
D

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

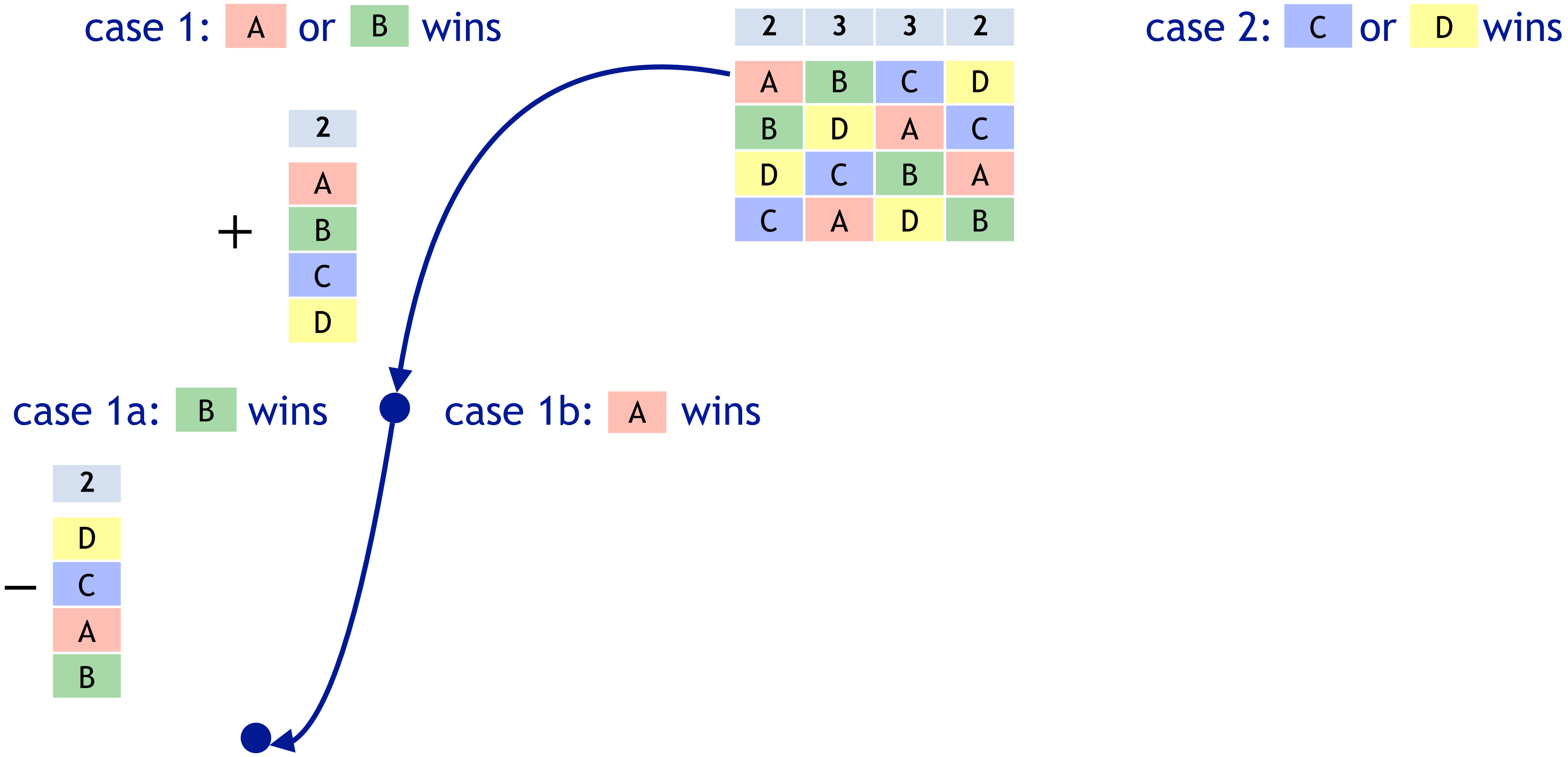
\+

2
A
B
C
D

case 1a: B wins ● case 1b: A wins

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

+

2
A
B
C
D

case 1a: B wins

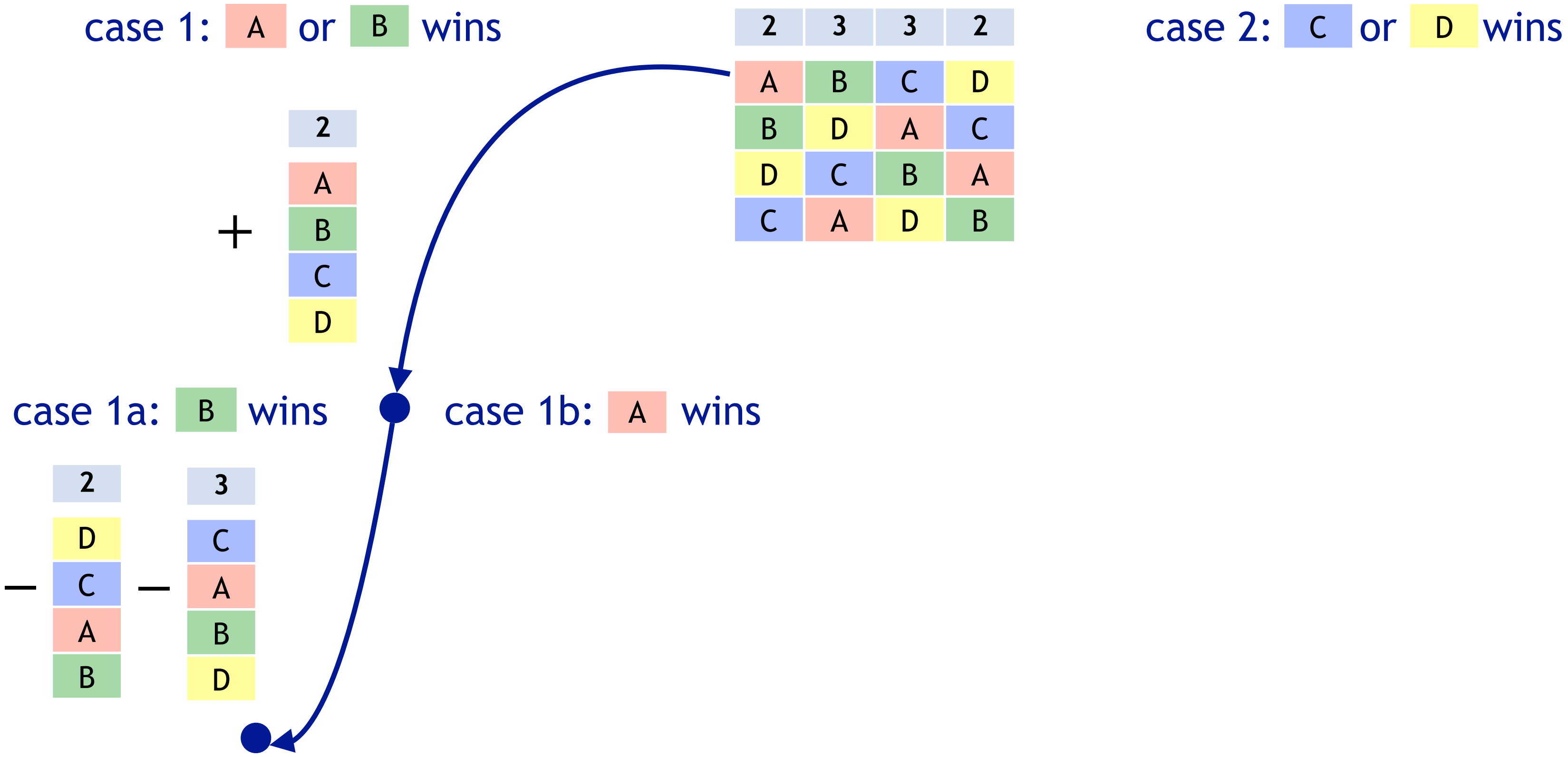
case 1b: A wins

–

2
D
C
A
B

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

+

2
A
B
C
D

case 1a: B wins

case 1b: A wins

−

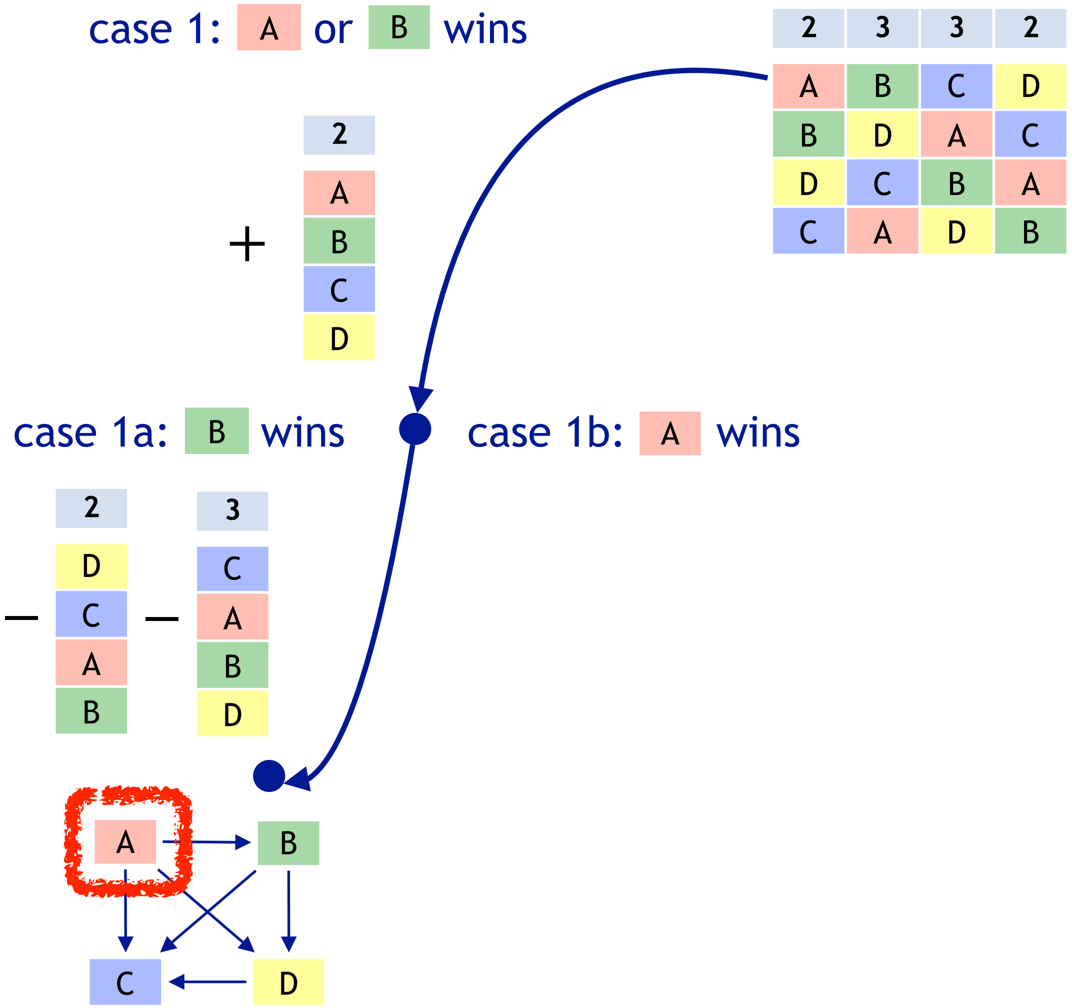
2
D
C
A
B

−

3
C
A
B
D

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

case 2: C or D wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

\+

2
A
B
C
D

case 1a: B wins

case 1b: A wins

−

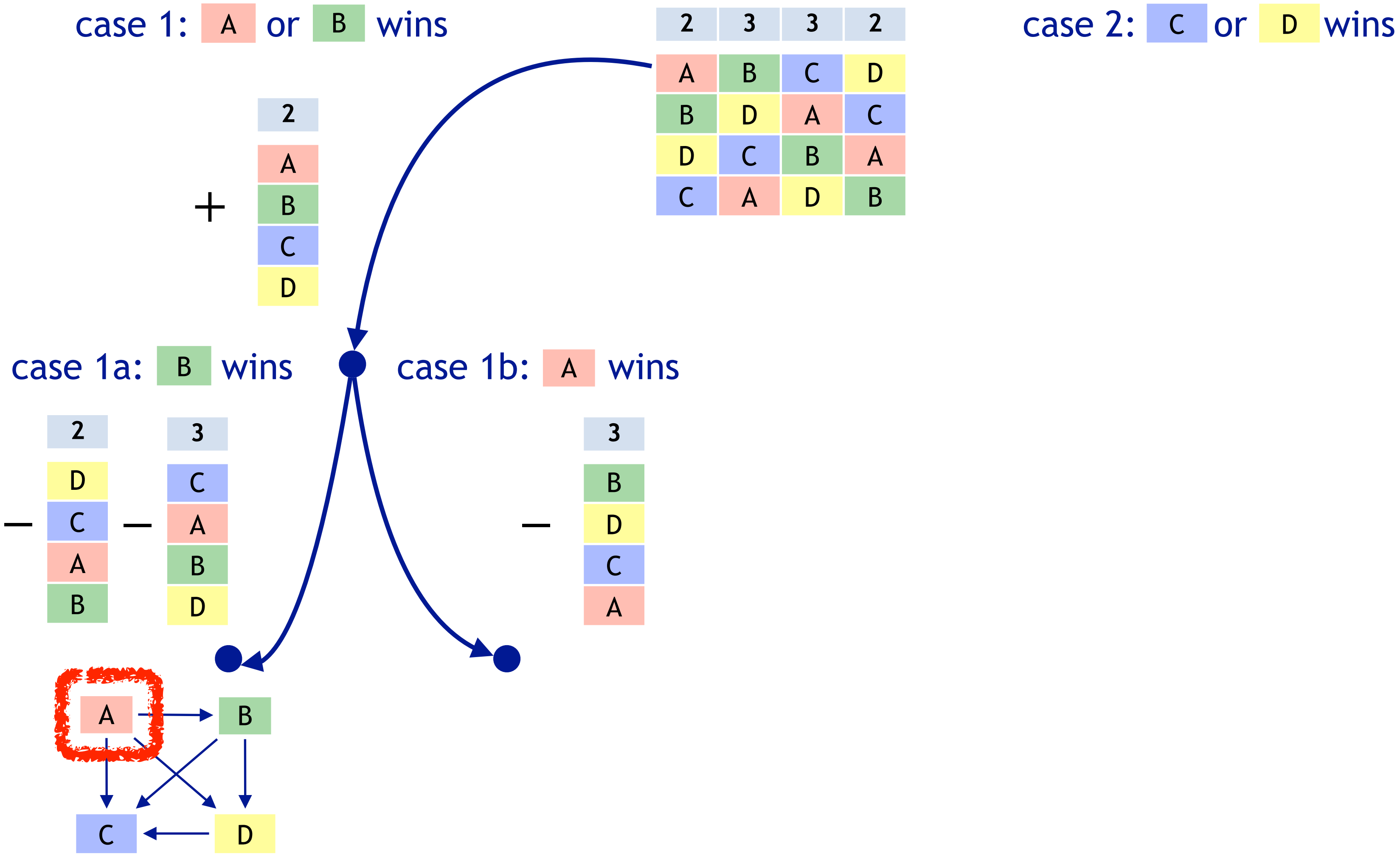
2
D
C
A
B

−

3
C
A
B
D

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

+

2
A
B
C
D

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

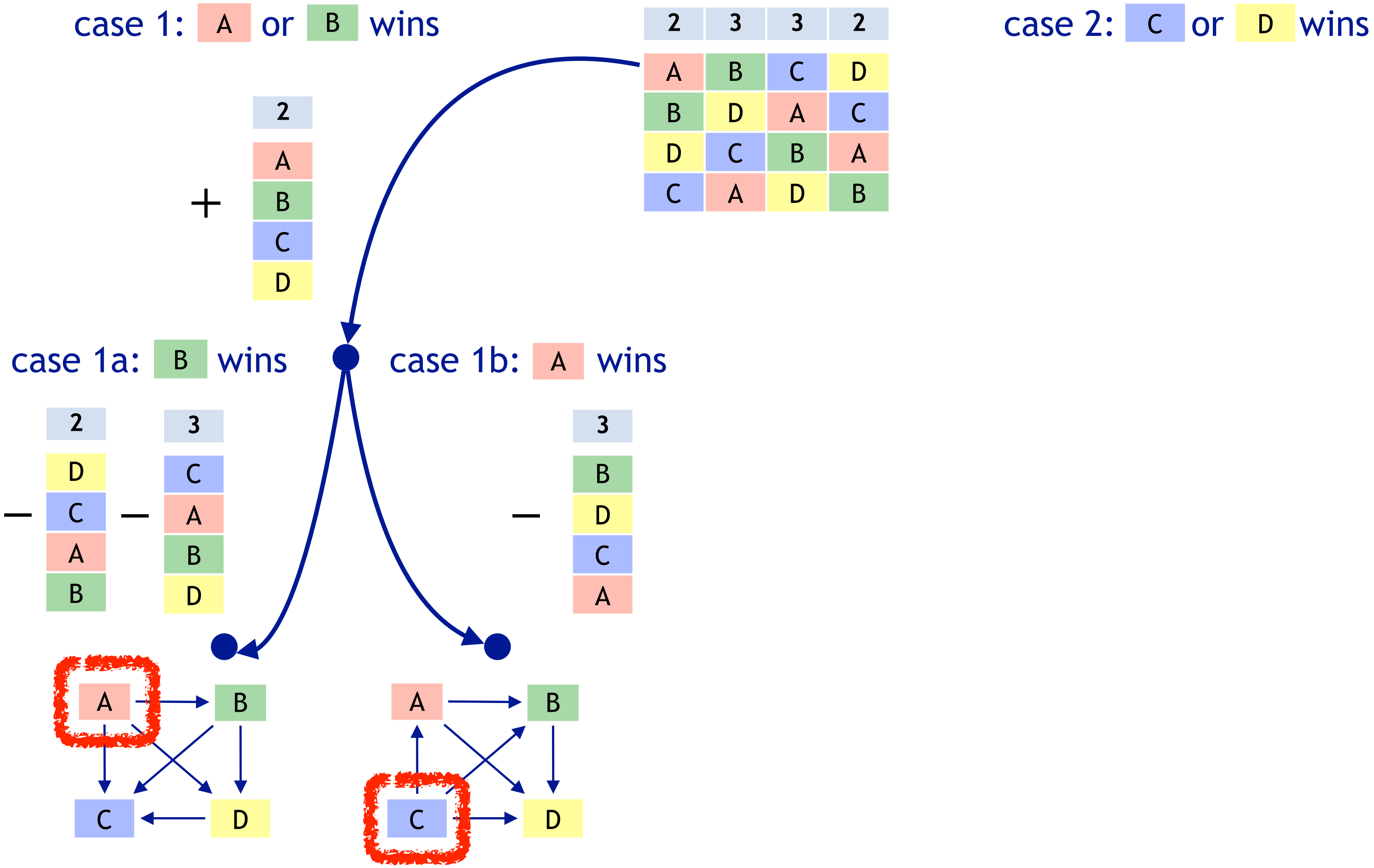
case 1b: A wins

−

3
B
D
C
A

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

case 2: C or D wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

\+

2
A
B
C
D

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

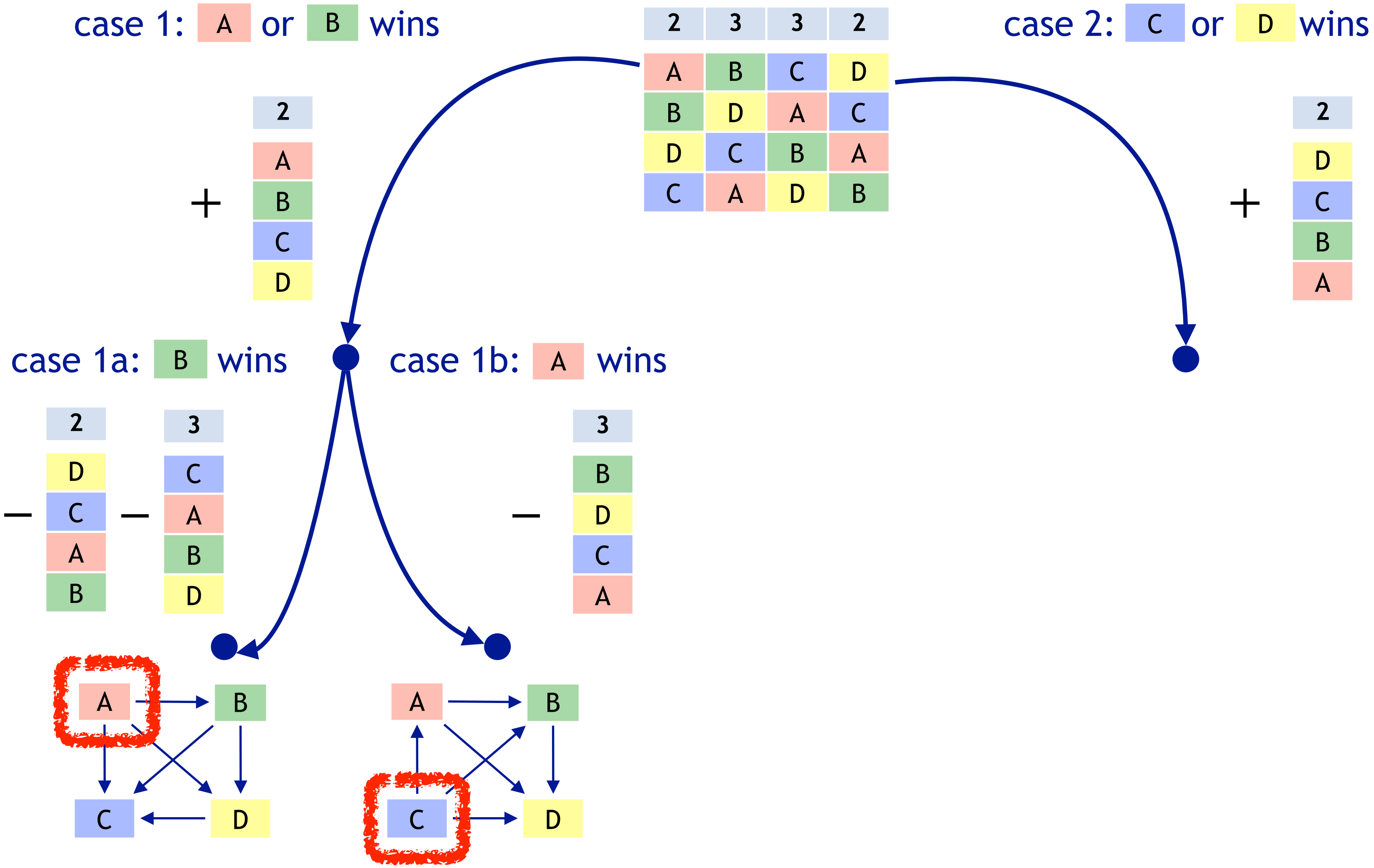
case 1b: A wins

−

3
B
D
C
A

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

\+

2
A
B
C
D

\+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

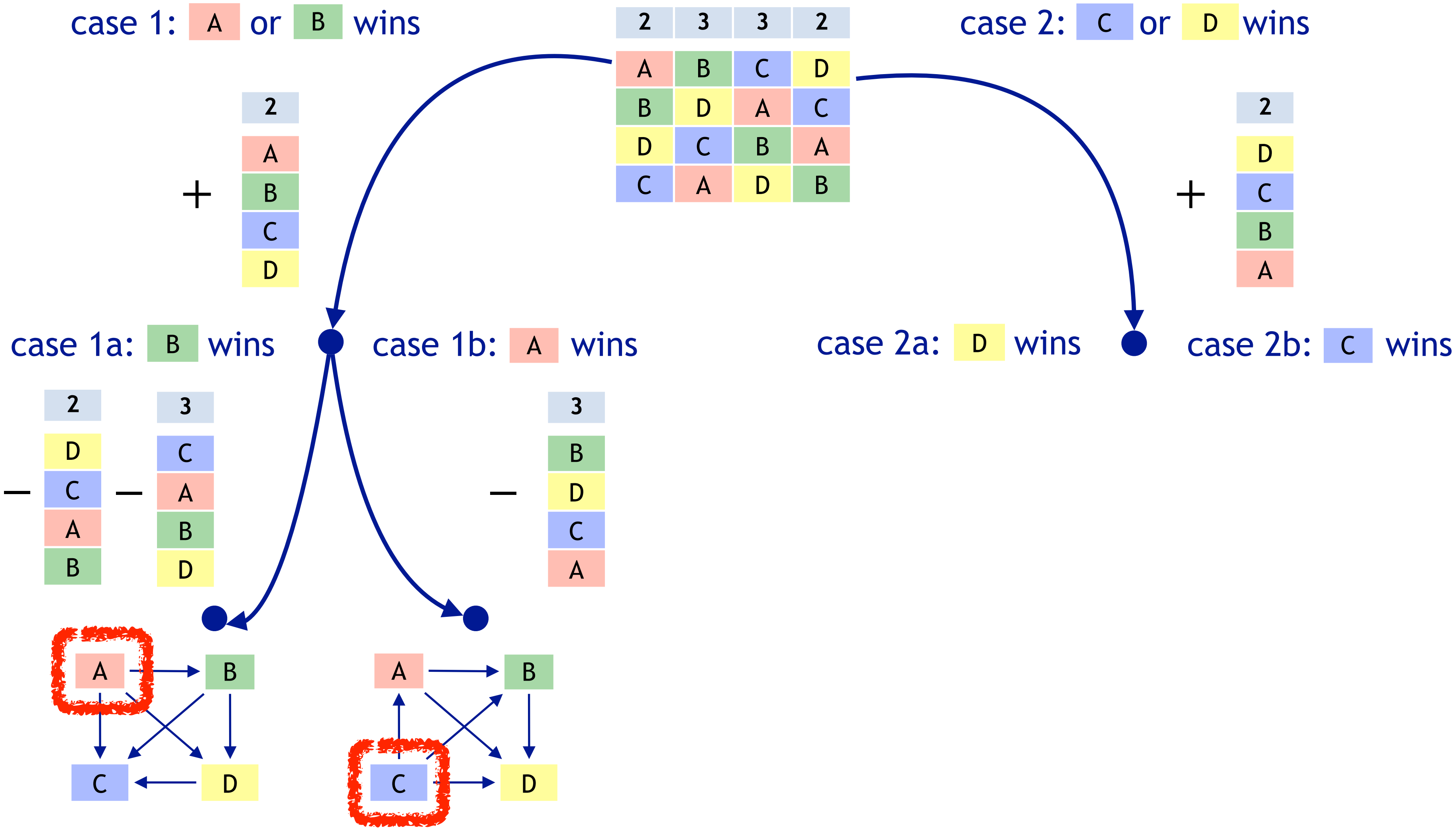
case 1b: A wins

−

3
B
D
C
A

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

+

2
A
B
C
D

+

2
D
C
B
A

case 1a: B wins

case 1b: A wins

case 2a: D wins

case 2b: C wins

−

2
D
C
A
B

−

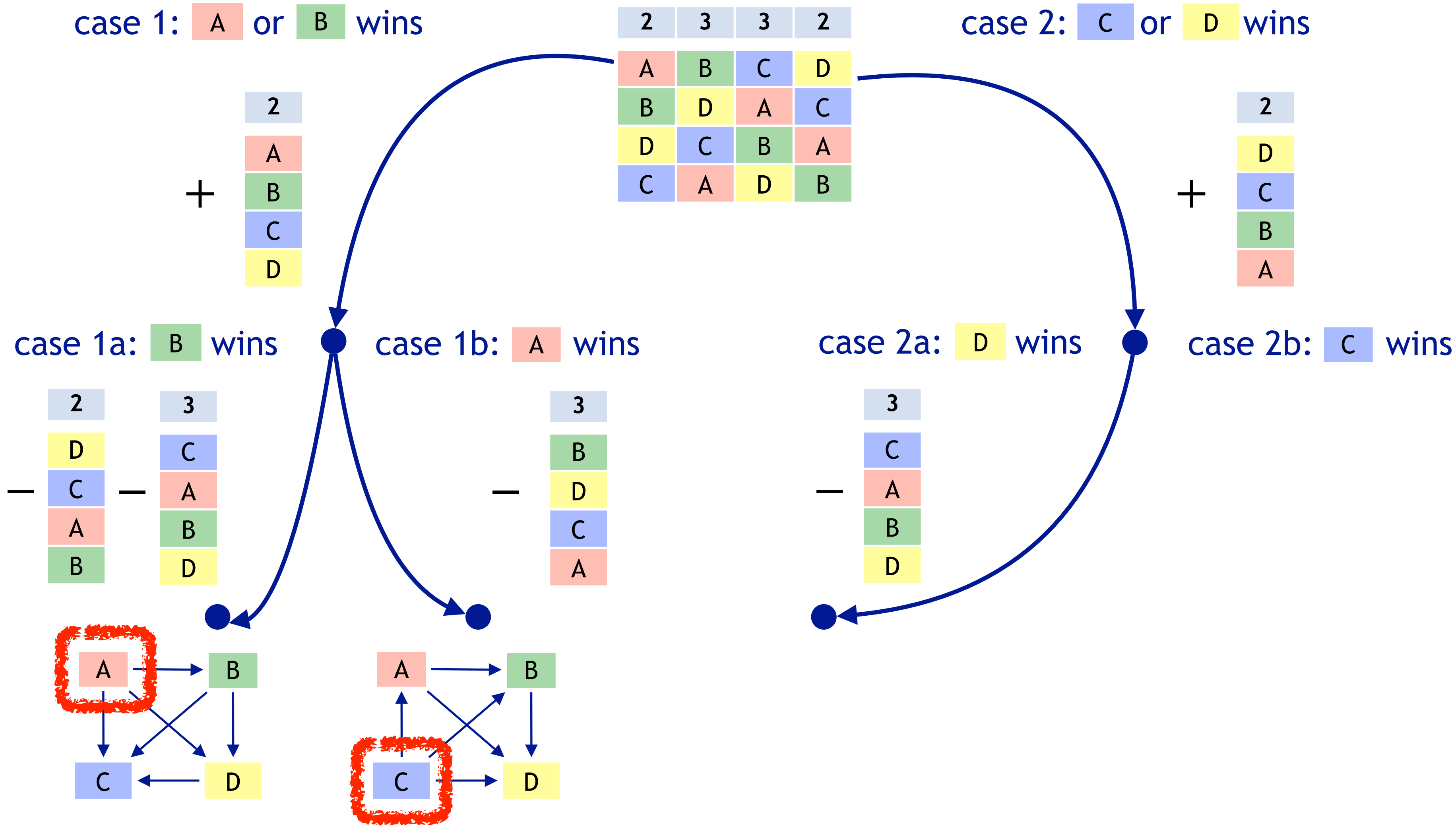
3
C
A
B
D

−

3
B
D
C
A

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

case 2: C or D wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

\+

2
A
B
C
D

\+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

case 1b: A wins

−

3
B
D
C
A

case 2a: D wins

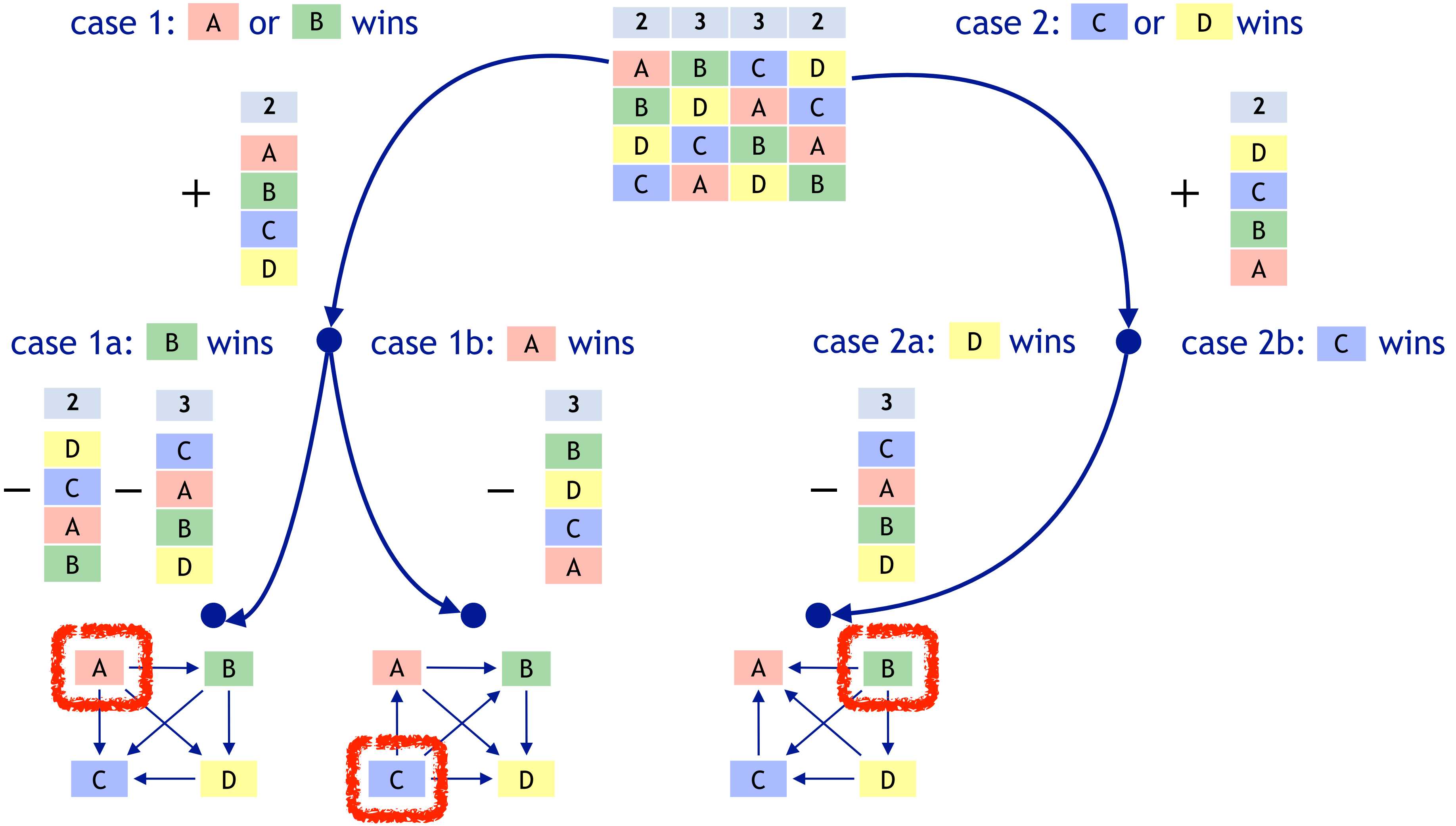
−

3
C
A
B
D

case 2b: C wins

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

\+

2
A
B
C
D

\+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

case 1b: A wins

−

3
B
D
C
A

case 2a: D wins

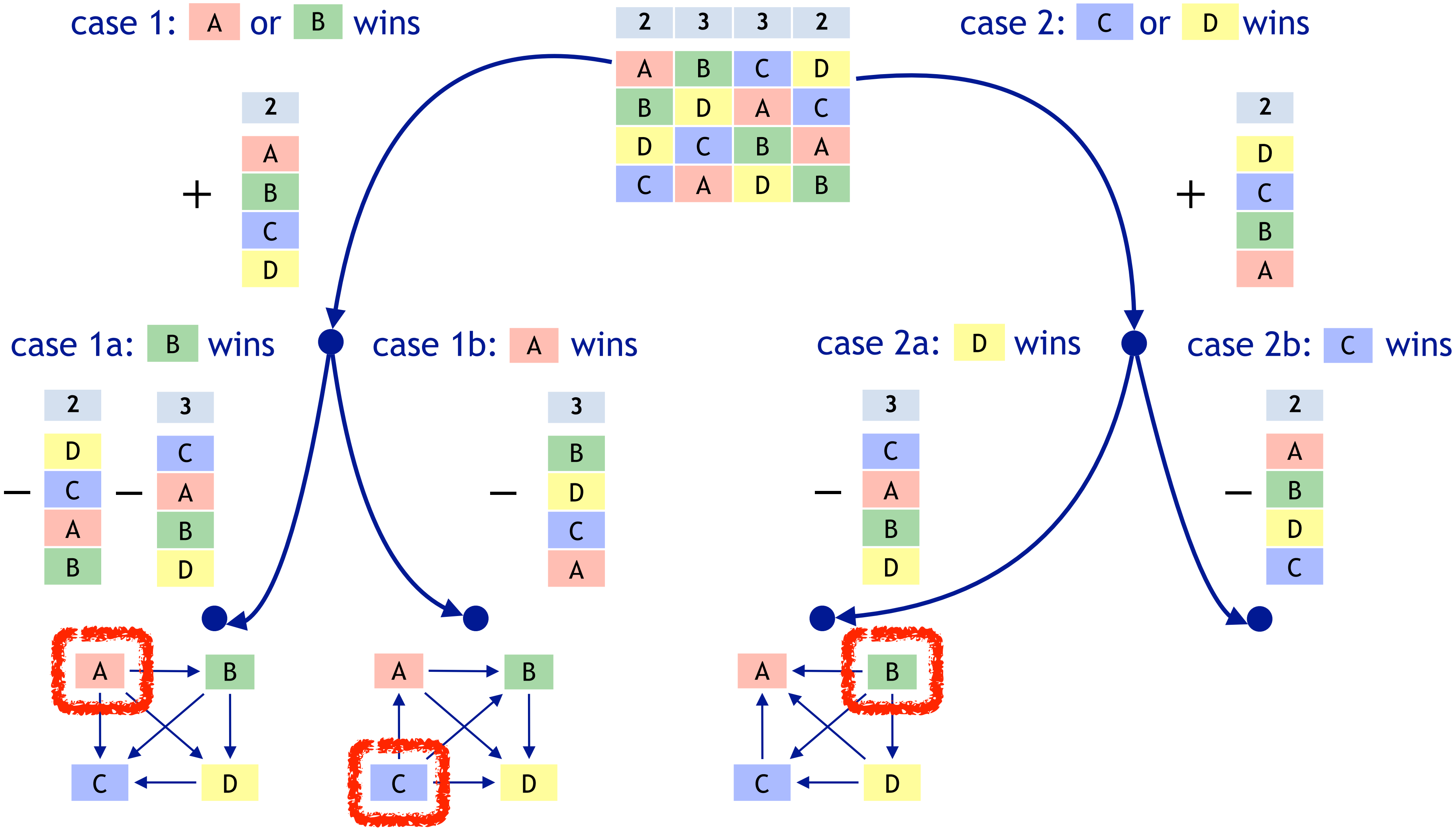
−

3
C
A
B
D

case 2b: C wins

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

case 2: C or D wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

+

2
A
B
C
D

+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

case 1b: A wins

−

3
B
D
C
A

case 2a: D wins

−

3
C
A
B
D

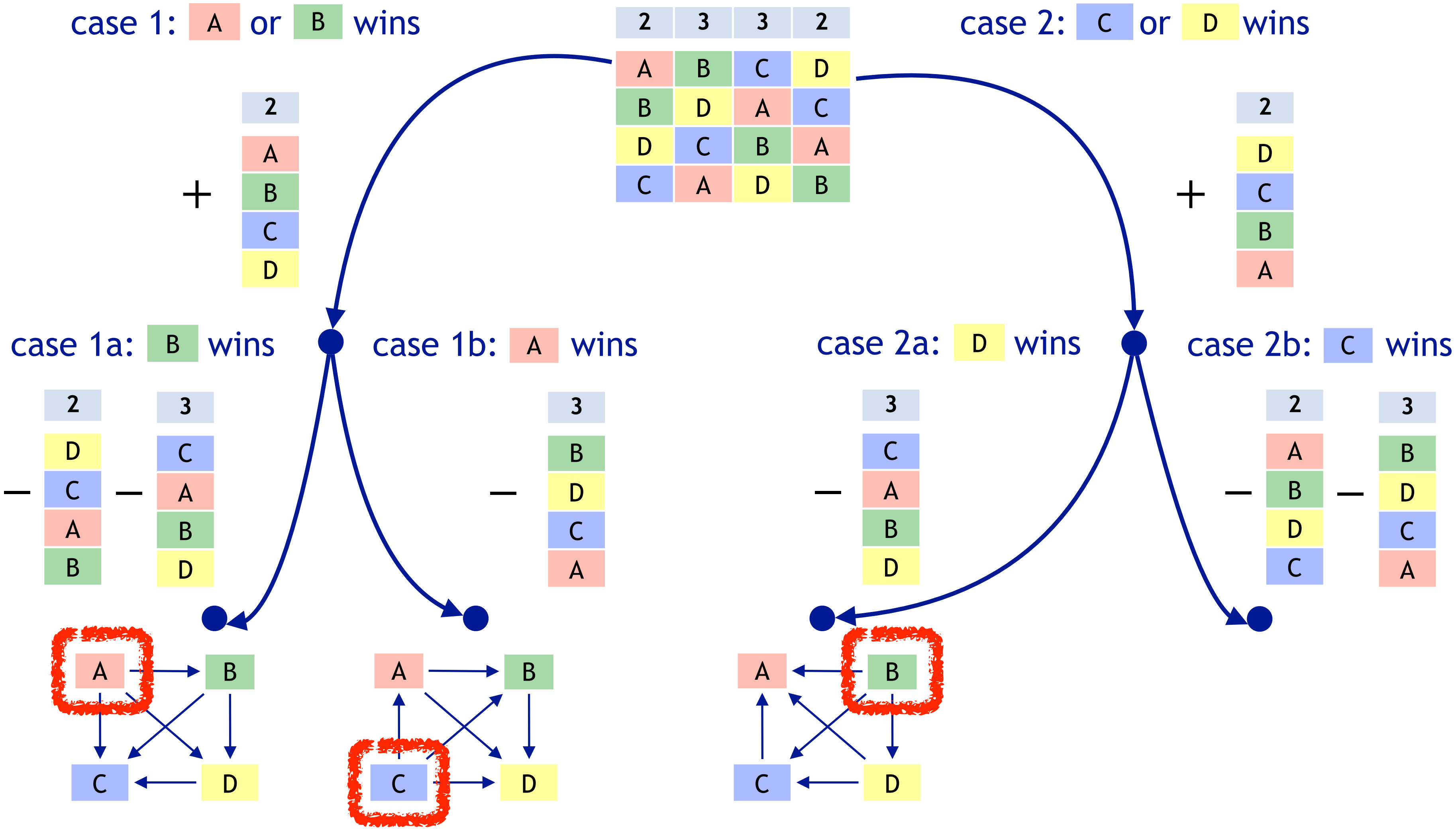
case 2b: C wins

−

2
A
B
D
C

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 1: A or B wins

+

2
A
B
C
D

case 2: C or D wins

+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

case 1b: A wins

−

3
B
D
C
A

case 2a: D wins

−

3
C
A
B
D

case 2b: C wins

−

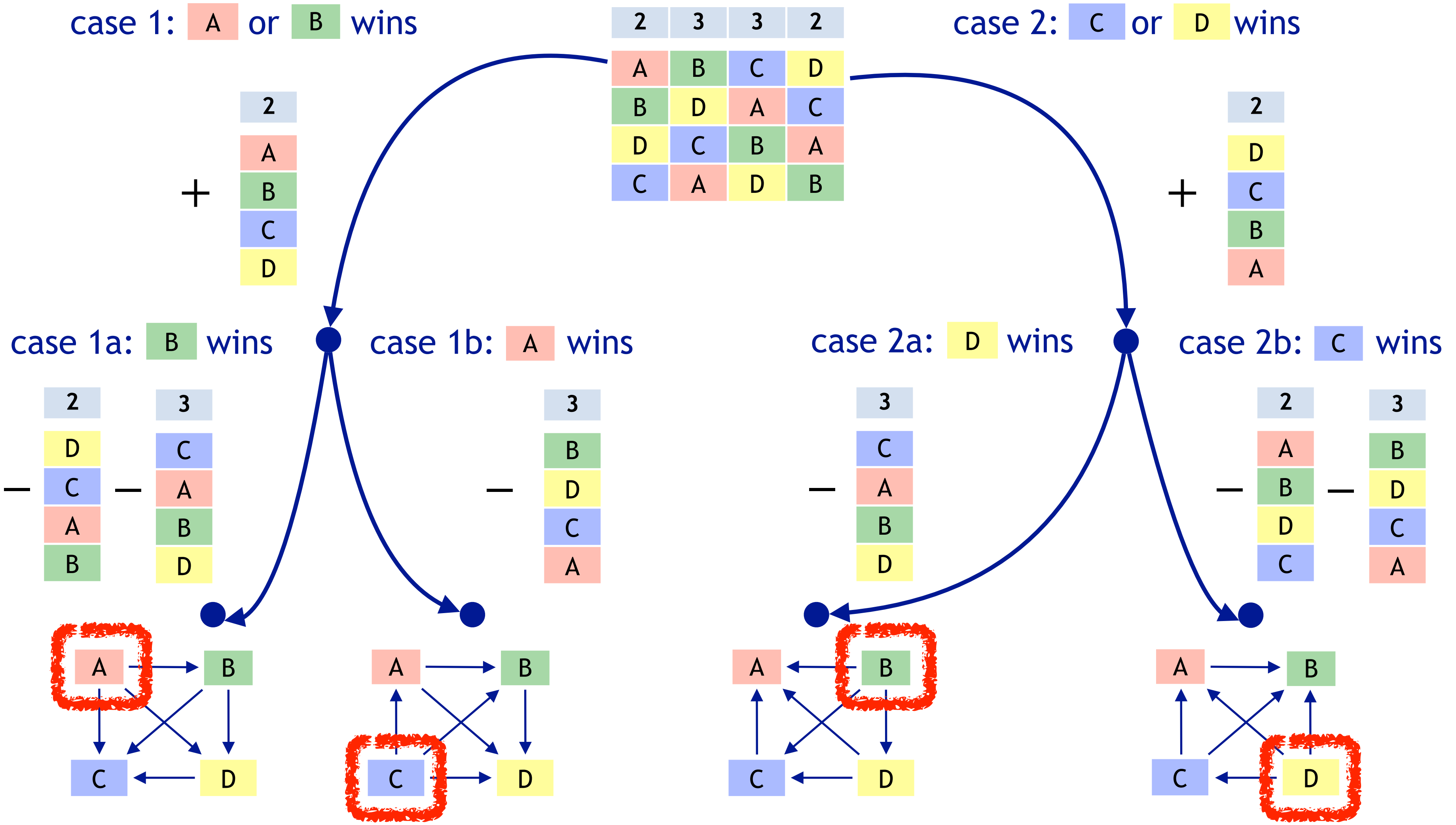
2
A
B
D
C

−

3
B
D
C
A

So which rule is the best?

Theorem: no Condorcet rule satisfies participation.



case 1: A or B wins

2	3	3	2
A	B	C	D
B	D	A	C
D	C	B	A
C	A	D	B

case 2: C or D wins

+

2
A
B
C
D

+

2
D
C
B
A

case 1a: B wins

−

2
D
C
A
B

−

3
C
A
B
D

case 1b: A wins

−

3
B
D
C
A

case 2a: D wins

−

3
C
A
B
D

case 2b: C wins

−

2
A
B
D
C

−

3
B
D
C
A

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

v1	v2	v3	v4	v5
A	A	C	A	B
B	C	B	B	C
C	B	A	C	A

v6	v7	v8	v9	V10
C	C	B	B	B
A	A	A	A	C
B	B	C	C	A

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

v1	v2	v3	v4	v5
A	A	C	A	B
B	C	B	B	C
C	B	A	C	A

A is a winner

v6	v7	v8	v9	V10
C	C	B	B	B
A	A	A	A	C
B	B	C	C	A

A is a winner

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

v1	v2	v3	v4	v5
A	A	C	A	B
B	C	B	B	C
C	B	A	C	A

A is a winner

v6	v7	v8	v9	V10
C	C	B	B	B
A	A	A	A	C
B	B	C	C	A

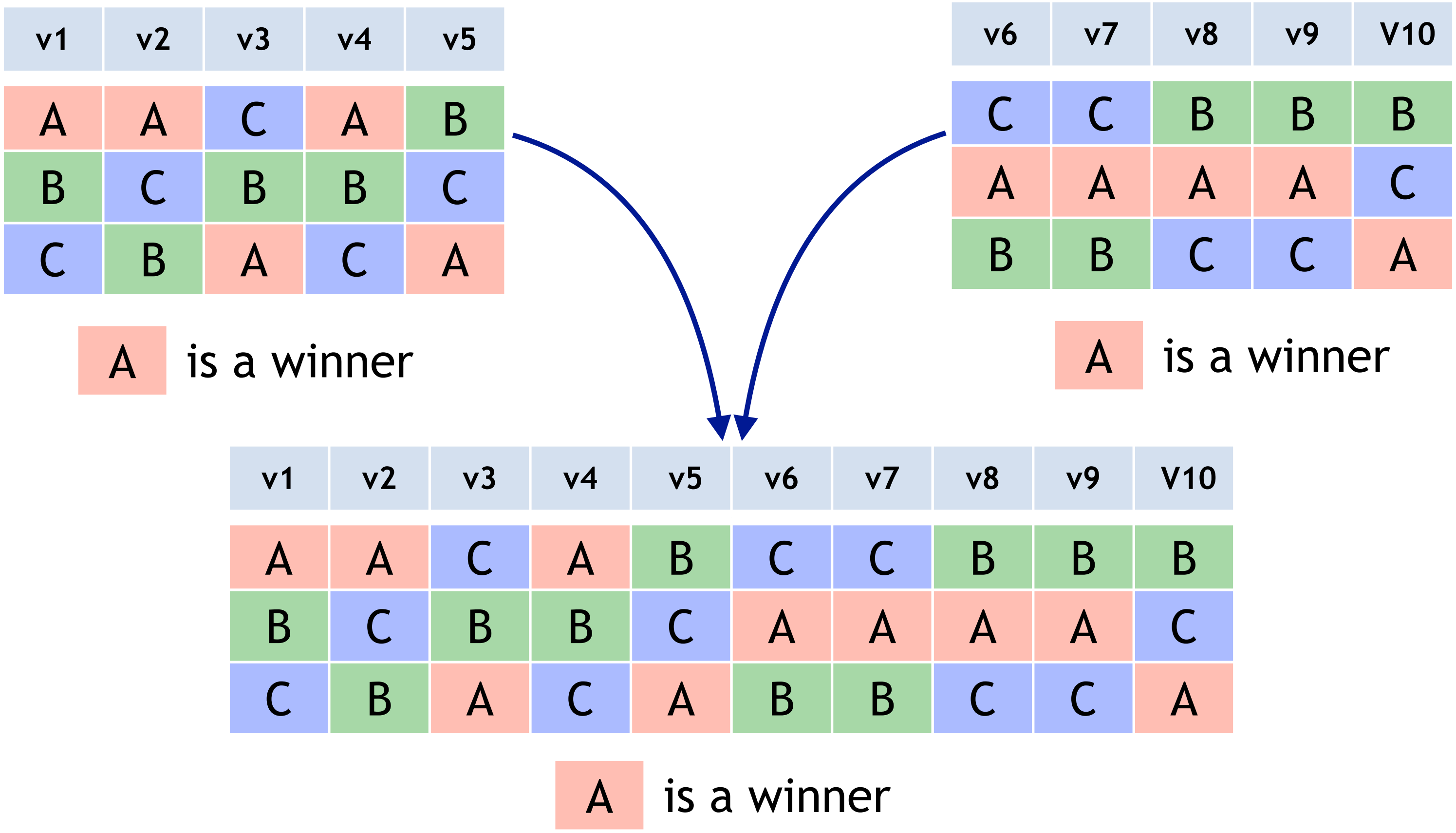
A is a winner

v1	v2	v3	v4	v5	v6	v7	v8	v9	V10
A	A	C	A	B	C	C	B	B	B
B	C	B	B	C	A	A	A	A	C
C	B	A	C	A	B	B	C	C	A

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$



v1	v2	v3	v4	v5
A	A	C	A	B
B	C	B	B	C
C	B	A	C	A

A is a winner

v6	v7	v8	v9	V10
C	C	B	B	B
A	A	A	A	C
B	B	C	C	A

A is a winner

v1	v2	v3	v4	v5	v6	v7	v8	v9	V10
A	A	C	A	B	C	C	B	B	B
B	C	B	B	C	A	A	A	A	C
C	B	A	C	A	B	B	C	C	A

A is a winner

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \emptyset$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

Theorem: Positional scoring rules are the only voting systems that satisfy consistency, anonymity and neutrality.

So which rule is the best?

A voting rule $\mathscr{R}$ satisfies **consistency** if for each two disjoint preference profiles, P_1 and P_2, such that $\mathscr{R}(P_1) \cap \mathscr{R}(P_2) \neq \varnothing$, it holds that:

$$\mathscr{R}(P_1 \cup P_2) = \mathscr{R}(P_1) \cap \mathscr{R}(P_2).$$

Theorem: Positional scoring rules are the only voting systems that satisfy consistency, anonymity and neutrality.

Beautiful, but very complex proof.

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	✗		✗		✗		✗	✗		✗
B		✗		✗		✗		✗	✗	
C	✗			✗			✗	✗		
D			✗		✗				✗	
E	✗		✗	✗						✗

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
A	✗		✗		✗		✗	✗		✗
B		✗		✗		✗		✗	✗	
C	✗			✗			✗	✗		
D			✗		✗				✗	
E	✗		✗	✗						✗

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	
A	✗		✗		✗		✗	✗		✗	6
B		✗		✗		✗		✗	✗		5
C	✗			✗			✗	✗			4
D			✗		✗				✗		3
E	✗		✗	✗						✗	4

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	
A	✗		✗		✗		✗	✗		✗	6
B		✗		✗		✗		✗	✗		5
C	✗			✗			✗	✗			4
D			✗		✗				✗		3
E	✗		✗	✗						✗	4

A is a winner

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

Advantages of approval voting:

1. A very simple ballot format! It is hard to rank more than several candidates

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

Advantages of approval voting:

1. A very **simple ballot format**! It is hard to rank more than several candidates (The Magical Number Seven, Plus or Minus Two).

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

Advantages of approval voting:

1. A very **simple ballot format**! It is hard to rank more than several candidates (The Magical Number Seven, Plus or Minus Two).
2. The experiments show that typically the **extremist candidates loose**, while the **centrist candidates benefit**.

Jean-François Laslier, Karine Vander Straeten. Approval Voting: An Experiment during the French 2002 Presidential Election. 2003.

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

Advantages of approval voting:

1. A very **simple ballot format**! It is hard to rank more than several candidates (The Magical Number Seven, Plus or Minus Two).
2. The experiments show that typically the **extremist candidates loose**, while the **centrist candidates benefit**.
3. A voter can **never loose by approving her top choice**.

Jean-François Laslier, Karine Vander Straeten. Approval Voting: An Experiment during the French 2002 Presidential Election. 2003.

Approval voting: a different approach

Instead of ranking, let each voter approve an arbitrary subset of candidates!

For each candidate we count the number of approvals, and pick the one that was approved by most voters.

Advantages of approval voting:

1. A very **simple ballot format**! It is hard to rank more than several candidates (The Magical Number Seven, Plus or Minus Two).
2. The experiments show that typically the **extremist candidates loose**, while the **centrist candidates benefit**.
3. A voter can **never loose by approving her top choice**.
4. **Reduces negative campaigning** (which might positively affect participation).

Jean-François Laslier, Karine Vander Straeten. Approval Voting: An Experiment during the French 2002 Presidential Election. 2003.

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

However, **the approval voting is typically a very good choice**, because of simplicity of ballots, reducing the effect of splitting votes, and positive nature of votes. **Go for approval voting as a default option!**

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

However, **the approval voting is typically a very good choice**, because of simplicity of ballots, reducing the effect of splitting votes, and positive nature of votes. **Go for approval voting as a default option!**

If **ranking candidates** is feasible (only if there are several candidates), this can **provide more information**.

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

However, **the approval voting is typically a very good choice**, because of simplicity of ballots, reducing the effect of splitting votes, and positive nature of votes. **Go for approval voting as a default option!**

If **ranking candidates** is feasible (only if there are several candidates), this can **provide more information**.

In practice **cloneprofness is a very important property**, and scoring rules are prone to cloning. **Therefore:**

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

However, **the approval voting is typically a very good choice**, because of simplicity of ballots, reducing the effect of splitting votes, and positive nature of votes. **Go for approval voting as a default option!**

If **ranking candidates** is feasible (only if there are several candidates), this can **provide more information**.

In practice **cloneprofness is a very important property**, and scoring rules are prone to cloning. **Therefore:**

1. **The Schulze method has very good properties**, and I would recommend it, if our goal is to find **consensus candidates**.

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

However, **the approval voting is typically a very good choice**, because of simplicity of ballots, reducing the effect of splitting votes, and positive nature of votes. **Go for approval voting as a default option!**

If **ranking candidates** is feasible (only if there are several candidates), this can **provide more information**.

In practice **cloneprofness is a very important property**, and scoring rules are prone to cloning. **Therefore:**

1. **The Schulze method has very good properties**, and I would recommend it, if our goal is to find **consensus candidates**.

2. If we want to put **more weight to the higher positions** in rankings, then **Single Transferrable Vote is a good choice**.

Summary: which rule to choose?

There is no single answer and each rule suffers from paradoxes.

In most cases you can choose one of these three options:

- **Approval Voting**
- **The Schulze method**
- **Single Transferrable Vote**

An interesting experiment

During 2010 VPP Workshop Assessing Alternative Voting Procedures, the participants used approval voting to vote on “What is the best voting rule that the city council of your town should use to elect the mayor?"

An interesting experiment

During 2010 VPP Workshop Assessing Alternative Voting Procedures, the participants used approval voting to vote on “What is the best voting rule that the city council of your town should use to elect the mayor?"

The results were:

- 68% approved Approval Voting,
- 45% approved STV,
- 40% approved Copeland,
- 36% approved Kemeny.

An interesting experiment

During 2010 VPP Workshop Assessing Alternative Voting Procedures, the participants used approval voting to vote on “What is the best voting rule that the city council of your town should use to elect the mayor?"

The results were:

- 68% approved Approval Voting,
- 45% approved STV,
- 40% approved Copeland,
- 36% approved Kemeny.

No one approved Plurality!

Resources

An application you can use to organise voting: https://whale5.noiraudes.net/

A simpler application for playing with voting rules: https://voting.ml/

The Wikipedia article about the Schulze method.

Further reading (for advanced):

Kenneth J. Arrow, Amartya K. Sen and Kotaro Suzumura. Handbook of Social Choice and Welfare. 2002.