

# Making Group Decisions

## Voting Procedures

# Contents

- Social Welfare Functions and Social Choice Functions
- Voting Procedures
- Desirable Properties for Voting Procedures
- Strategic Manipulation

# Introduction

- Class of protocols intended for making **group decisions – social choice theory – voting theory**

# Introduction

- The voting protocols have a strategic flavour, because the agents participating in the protocol will have preferences over the possible outcomes of voting, and they will take into account their own preferences, as well as those of others when making their decisions about how to vote, in order to try to bring about their most preferred outcome

# Social Welfare Functions and Social Choice Functions

- A set  $\mathbf{Ag} = \{1, \dots, n\}$  of agents – **voters**
- These are the participants in the group decision-making process
- $\mathbf{Ag}$  is finite
- $\mathbf{Ag}$  contains an odd number of voters – eliminate the possibility of ties in elections

# Social Welfare Functions and Social Choice Functions

- Voters will make group decisions with respect to a set  $\Omega = \{\omega_1, \omega_2, \dots\}$  of possible **outcomes** or **candidates**
- $\Omega$  is finite
- The voters will be aiming to **rank** or **order** these candidates
- Sometimes, they will be aiming **to choose one** (the top ranking candidate)

# Observation

- If  $|\Omega| = 2 \rightarrow$  **pairwise election**
- If  $|\Omega| > 2 \rightarrow$  **general voting scenario**
- Each voter will have preferences over  $\Omega$   
 $\rightarrow$  ordering over the set of possible outcomes  $\Omega$

# Example

- Agent  $i$ 's preference ordering is  $(\omega_2, \omega_3, \omega_1)$
- Agent  $i$  prefers  $\omega_2$  over  $\omega_3$  and  $\omega_3$  over  $\omega_1$
- $\varpi_1, \varpi_2, \dots, \varpi_n$  the preference orders of agents  $1, 2, \dots, n$
- $\omega \succ_i \omega'$  outcome  $\omega$  is ranked above outcome  $\omega'$  in agent  $i$ 's preference order  $\varpi_i$
- $\Pi(\Omega)$  the set of all preference orderings over outcomes  $\Omega$

# Example

- Given a collection of preference orders, one for each agent, how we combine these to derive a group decision ?

# Social Welfare Function

- A **social welfare function** takes the voter preferences and produces a **social preference order** – a ranking of the candidates, from most preferred down to least preferred
- A **social welfare function** is a function

$$f : \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n \text{ times}} \rightarrow \Pi(\Omega)$$

# Social Welfare Function

- $\succ^*$  the outcome of a social welfare function
- $\omega \succ^* \omega'$   $\omega$  is ranked above  $\omega'$  in the social outcome

# Social Choice Function

- Sometimes, we are not concerned with obtaining an entire ordering, but just one of the possible candidates
- A **social choice function** is such a mapping

$$f : \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n \text{ times}} \rightarrow \Omega$$

- Refer to social welfare functions and social choice functions as **voting procedures**

# Plurality

- **Plurality procedure** – used to select a single candidate, rather than produce a ranked list of candidates
- Every voter submits their preference order
- Count how many times each outcome is ranked first in a preference order
- The winner is the outcome that appears first in the preference orders the largest number of times

# Observation

- If we only have two outcomes to choose between, then plurality is just **simple majority voting** – in this case, we ask candidates to select one of the two outcomes, and the one that gets the majority of votes is the winner

# Example

- Suppose we have 3 outcomes and 3 voters
- $\Omega = \{\omega_1, \omega_2, \omega_3\}$
- $\mathbf{Ag} = \{1, 2, 3\}$
- Preferences
- $\omega_1 \succ_1 \omega_2 \succ_1 \omega_3$
- $\omega_3 \succ_2 \omega_1 \succ_2 \omega_2$
- $\omega_2 \succ_3 \omega_3 \succ_3 \omega_1$

# Example

- The result is tied with plurality voting – there is no winner, since each outcome is ranked first exactly once
- The situation is actually worse than this
- Consider the merits of each outcome in turn

# Example

- Suppose we select candidate  $\omega_1$
- $2/3$  of the electorate would prefer  $\omega_3$  over  $\omega_1$
- Motivated by this argument, we suggest choosing  $\omega_3$
- Again,  $2/3$  of the electorate would prefer  $\omega_2$  over  $\omega_3$
- If we suggest choosing  $\omega_2$ , again,  $2/3$  of the electorate would prefer  $\omega_1$  over  $\omega_2$

# Example

- For every possible candidate, there is another candidate who is preferred by  $2/3$  of the electorate
- This is known as **Condorcet's paradox**, which is one of the simplest and most profound difficulties with voting procedures

# Condorcet's Paradox

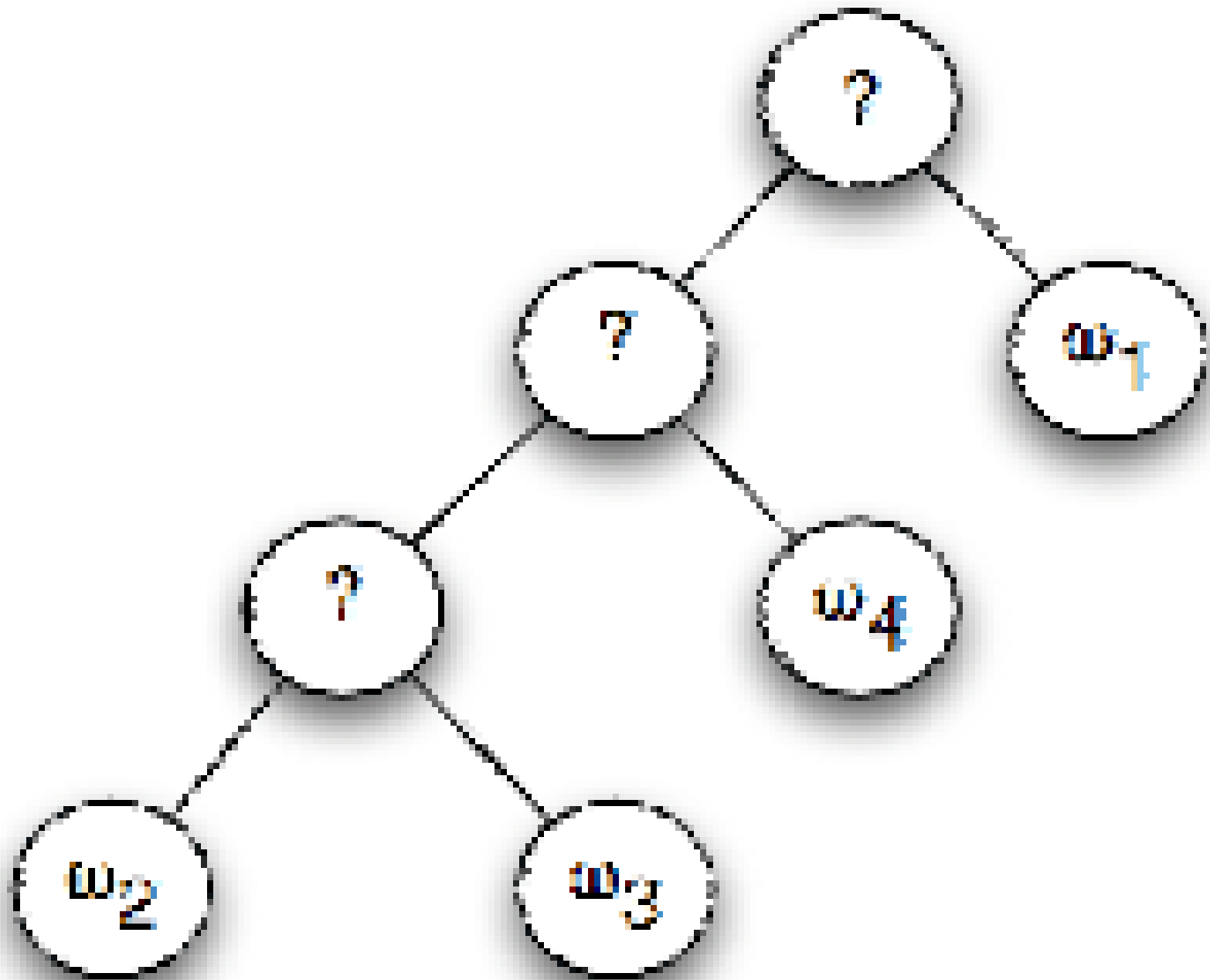
- While it seems intuitively inevitable that in a democracy we cannot hope to choose an outcome that keeps every voter happy, **Condorcet's paradox** tells us that there are scenarios in which no matter which outcome we choose, a majority of voters will be unhappy with the outcome chosen

# Sequential Majority Elections

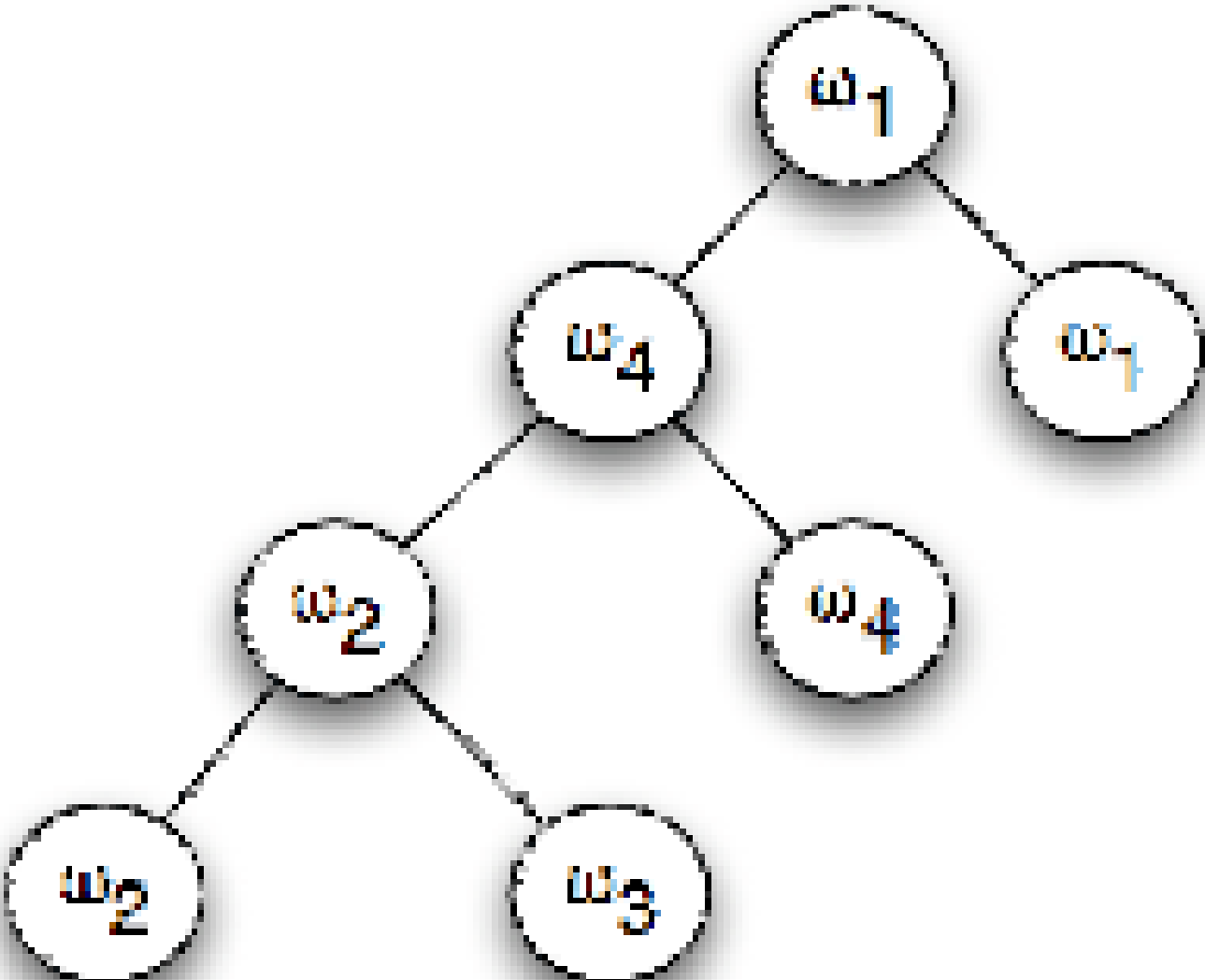
- Consider organizing a series of elections
- A pair of outcomes will face each other in a **pairwise** election, and the winner will go on the next election
- Suppose we are voting over 4 outcomes
- $\omega_1$ ,  $\omega_2$ ,  $\omega_3$  and  $\omega_4$
- Choose the order – **agenda** – for election
- $\omega_2$ ,  $\omega_3$ ,  $\omega_4$ ,  $\omega_1$

# Sequential Majority Elections

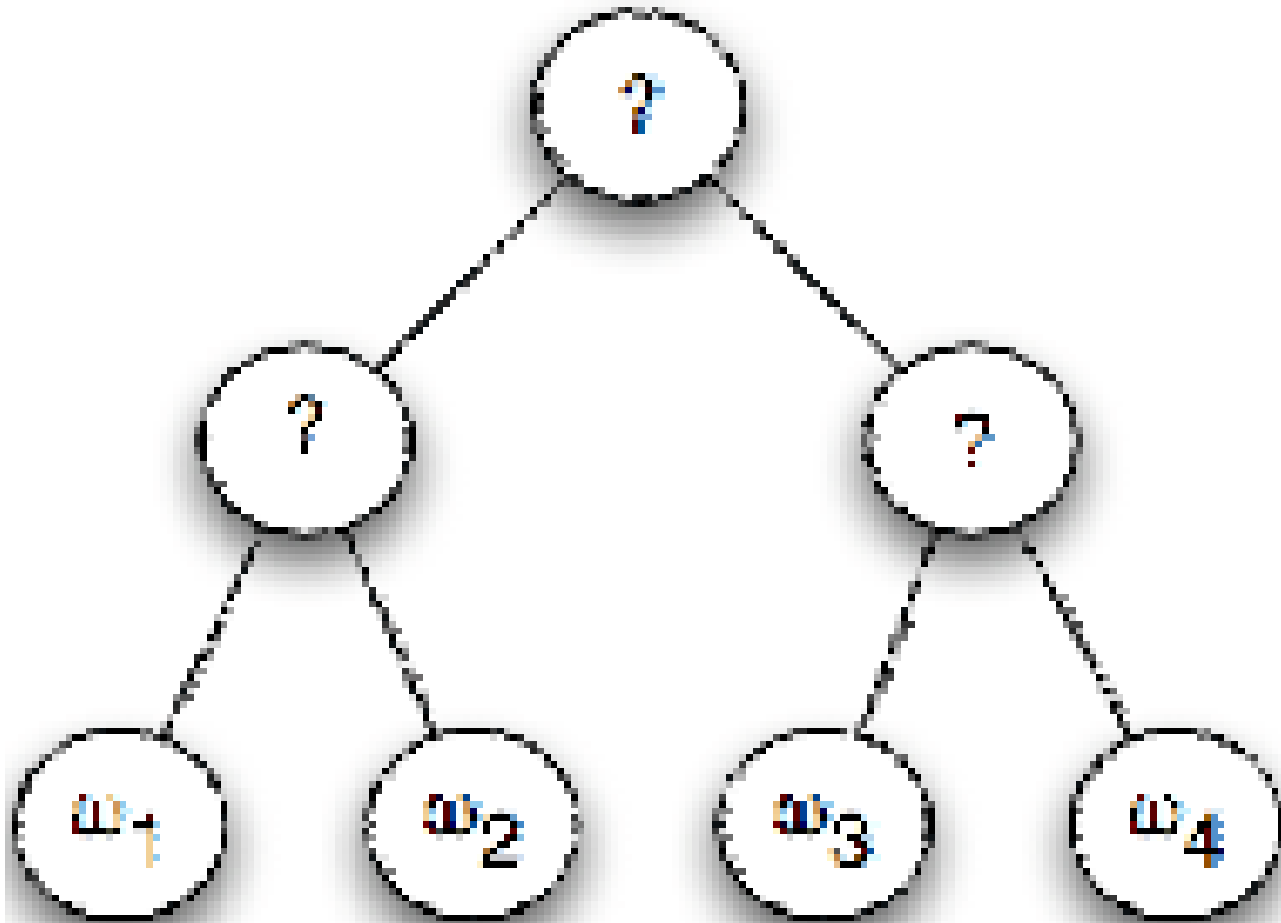
- The first election will be between  $\omega_2$  and  $\omega_3$
- The winner of this election will go on to face  $\omega_4$
- Then the winner of this election will face  $\omega_1$
- The winner of this final election is declared the overall winner



# Possible Outcome of Voting Order

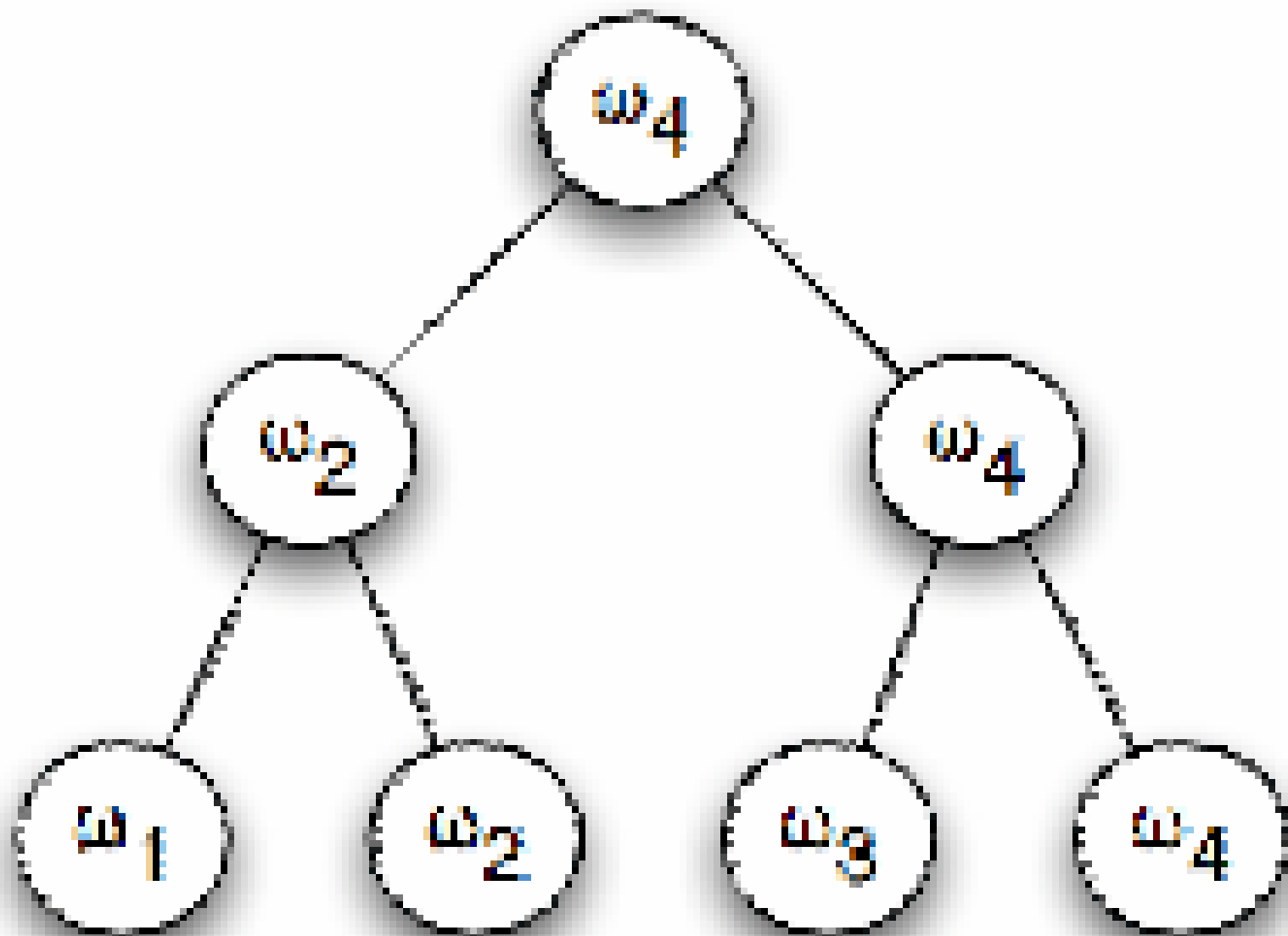


# Balanced Binary Tree of Elections



# Balanced Binary Tree of Elections

- The winner of the election between  $\omega_1$  and  $\omega_2$  will go on to face the winner of the election between  $\omega_3$  and  $\omega_4$ , and the winner of this third election is declared the overall winner



# Observation

- The final outcome selected may depend not just on voter preferences, but on the order in which the candidates come up for election – the election agenda
- If the agenda is selected at random, then randomness has some part to play in a democratic process, while if the agenda is chosen otherwise, then it opens up the possibility of the election being manipulated by an electioneer

# Majority Graph

- Create a directed graph, called a **majority graph**, from voter preferences
- The nodes in the graph will correspond to outcomes  $\Omega$ , and there will be an edge in the graph from outcome  $\omega$  to outcome  $\omega'$  if a majority of voters rank  $\omega$  above  $\omega'$  (if  $\omega$  would beat  $\omega'$  in a direct competition)
- Assume there are an odd number of voters, so that there are no ties

# Majority Graph

- The majority graph will have special properties:
- 1) **Complete** graph – for any two outcomes  $\omega_i$  and  $\omega_j$ , we must have either  $\omega_i$  defeats  $\omega_j$  or  $\omega_j$  defeats  $\omega_i$
- 2) **Asymmetric** graph – if  $\omega_i$  defeats  $\omega_j$ , then it cannot be the case that  $\omega_j$  defeats  $\omega_i$
- 3) **Irreflexive** graph – an outcome will never defeat itself

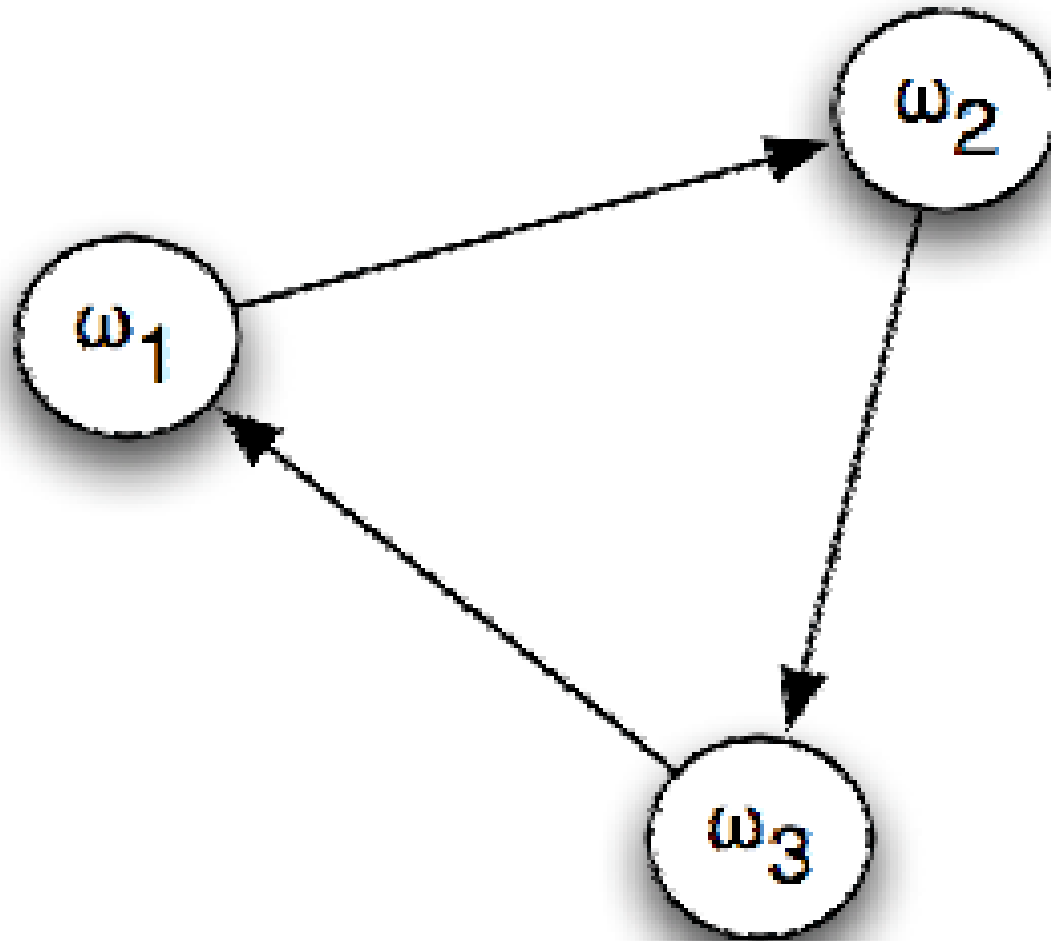
# Tournament

- A graph with these properties is called a **tournament**
- One way to think of the majority graph is as a **succinct representation of voter preferences**
- We can summarize a lot of the information contained in those preference orders in a majority graph

# Sequential Majority Elections

- Suppose we organize a linear sequential majority election for 3 candidates:  $\omega_1$ ,  $\omega_2$ , and  $\omega_3$ , with 99 voters, who have preferences as follows:
  - 33 voters have preferences  $\omega_1 \succ_i \omega_2 \succ_i \omega_3$
  - 33 voters have preferences  $\omega_3 \succ_i \omega_1 \succ_i \omega_2$
  - 33 voters have preferences  $\omega_2 \succ_i \omega_3 \succ_i \omega_1$

# Majority Graph



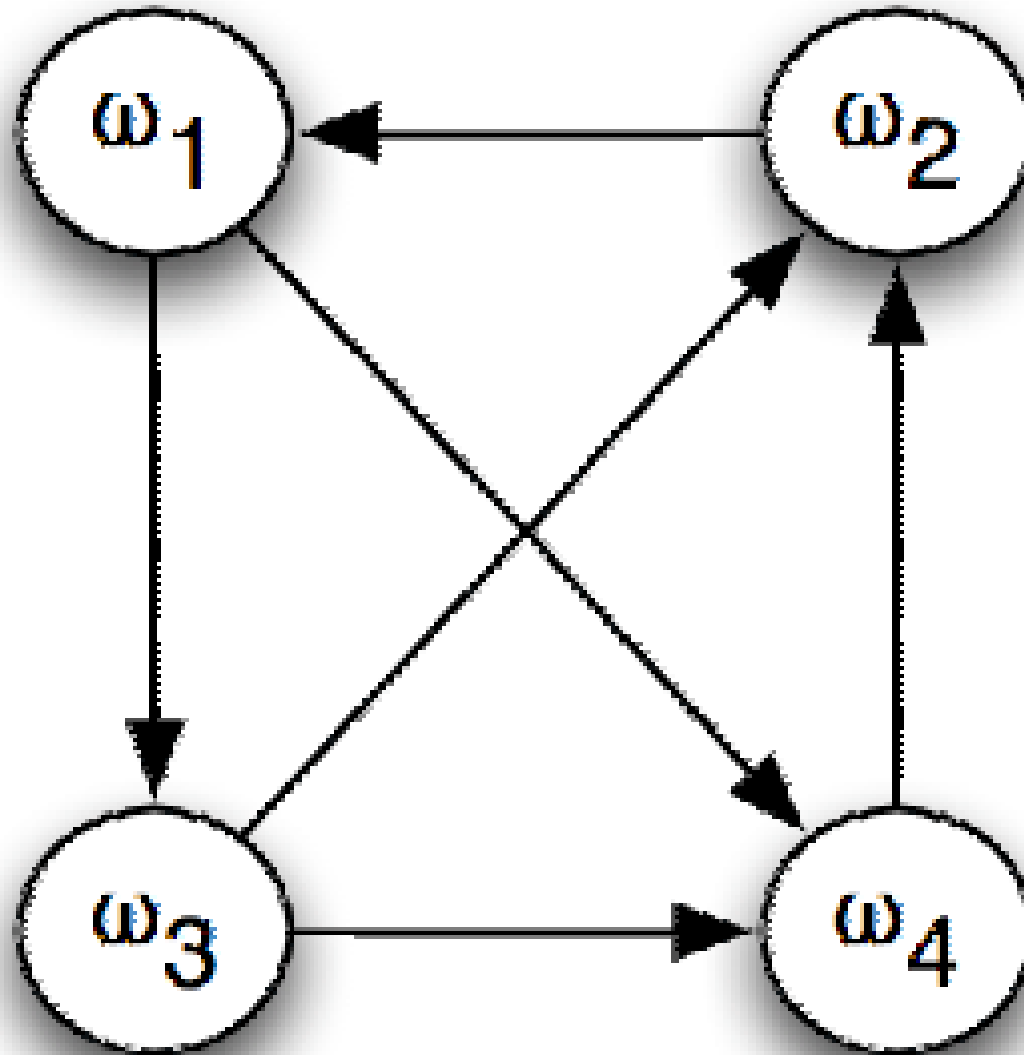
# Sequential Majority Elections

- The majority graph is easy to compute
- The graph is a simple cycle: starting at any node, there is a single path, which leads eventually back to the same node
- We can fix an election agenda so that any of the 3 candidates will win

# Example

- If we want candidate  $\omega_1$  to win, then the agenda  $\omega_3, \omega_2, \omega_1$  will do, since we know from the majority graph that  $\omega_2$  will beat  $\omega_3$ , and then  $\omega_1$  will beat  $\omega_2$
- If we want  $\omega_3$  to win, then the order  $\omega_2, \omega_1, \omega_3$  will do it:  $\omega_1$  will beat  $\omega_2$ , and then  $\omega_3$  will beat  $\omega_1$

# Majority Graph



# Observation

- Every outcome is a possible winner
- The agenda  $\omega_3, \omega_2, \omega_4, \omega_1$  will result in outcome  $\omega_1$  winning
- The agenda  $\omega_1, \omega_4, \omega_2, \omega_3$  will result in outcome  $\omega_3$  winning

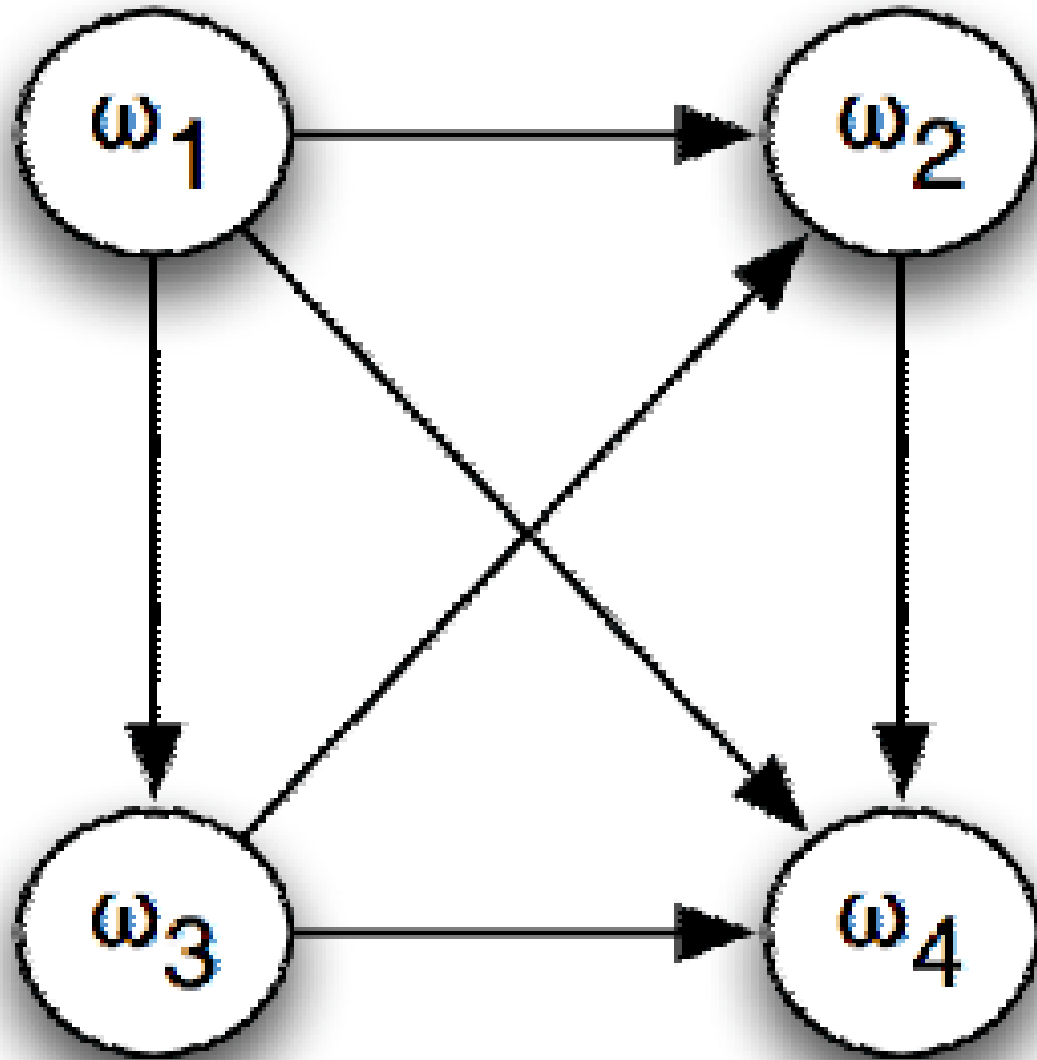
# Definitions

- An outcome is a **possible winner** if there is some agenda which would result in that outcome being the overall winner
- An outcome is called **Condorcet winner** if it is the overall winner for every possible agenda

# Example

- To determine whether an outcome  $\omega_i$  is a possible winner, we check whether, for every other outcome  $\omega_j$ , there is a path from  $\omega_i$  to  $\omega_j$  in the majority graph
- To check that  $\omega_i$  is a Condorcet winner involves checking whether there is an edge from  $\omega_i$  to every other node in the majority graph

$\omega_1$  is a Condorcet Winner



# Example

- Construct a voting order so as to give the best chance possible to an outcome  $\omega_i$
- Rank all the candidates in order, from most likely to defeat  $\omega_i$  down to least likely to defeat  $\omega_i$ , with  $\omega_i$  as the final candidate
- This heuristic works, because candidates appearing early in a linear voting order would have to face more elections in order to progress to a point where they face candidate  $\omega_i$

# Borda Count

- The **Borda count** takes into account all the information from a preference order
- Suppose we have **k** candidates
- $|\Omega| = k$
- For each of these possible candidates, we have numeric variable, counting the strength of opinion in favor of this candidate

# Borda Count

- Each voter preference order contributes to these counts
- If an outcome  $\omega_1$  appears first in the preference order, then we increment the count for  $\omega_1$  by  $k - 1$
- Then we increment the count for the next outcome in the preference order by  $k - 2$ , until the final outcome in the preference order has its total incremented by  $0$

# Borda Count

- We proceed with this process until all preference orders have been considered, and then order the outcomes  $\Omega$  by their count, largest first, down to smallest
- If we aim to select a single candidate, then the candidate with the largest count is chosen

# Slater Ranking

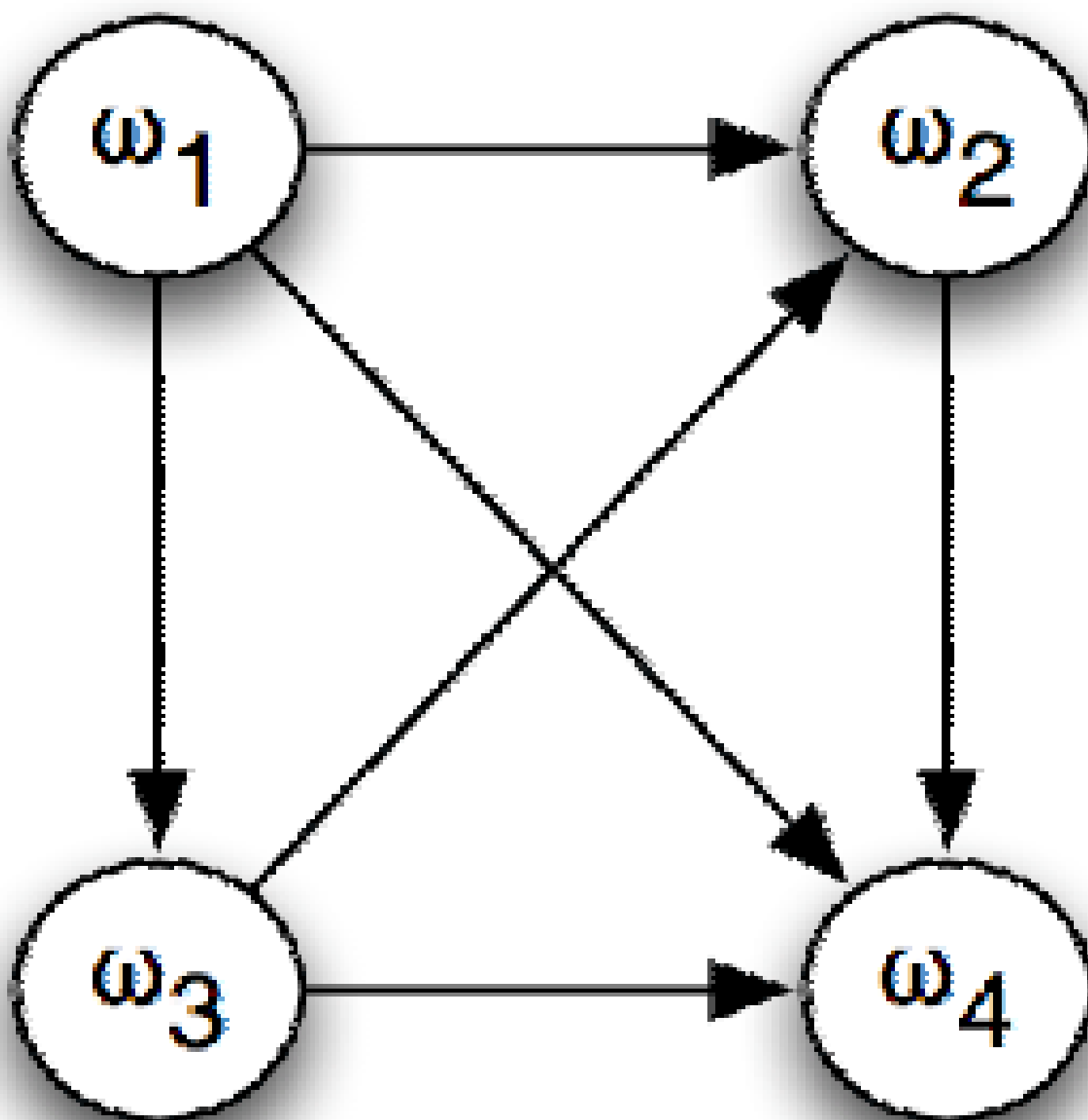
- Consider the question of which social ranking should be selected as being the question of trying to find a consistent ranking (one that does not contain cycles) that is as close to the majority graph as possible – to minimize the number of disagreements between **the majority graph and the social choice**

# Slater Ranking

- Consider all possible orderings of the candidates
- For each ordering, we measure how much that ordering would disagree with the information provided in the majority graph, that is, how many edges in the graph would have to be ‘flipped’ in order to make the corresponding ordering consistent with the majority graph

# Slater Ranking

- Which ordering minimizes the inconsistency measure ?
- If a majority graph has no cycles, then there is no difficulty in identifying a consistent ranking
- The majority graph in the figure has no cycles



# Slater Ranking

- The following social choice is acceptable:
- $\omega_1 >^* \omega_3 >^* \omega_2 >^* \omega_4$
- In this ranking, every candidate  $\omega_i$  that appears before a candidate  $\omega_j$  in the ordering would beat candidate  $\omega_j$  in a pairwise election according to the majority graph

# Slater Ranking

- The graph in the figure has cycles
- Whichever ordering we choose will disagree with the majority graph, in the sense that at least one candidate  $\omega_i$  will be ranked above a candidate  $\omega_j$  despite the fact that  $\omega_j$  would beat  $\omega_i$  in a pairwise election

# Example

- Consider the possible social choice ordering:
- $\omega_1 \succ^* \omega_2 \succ^* \omega_3 \succ^* \omega_4$
- Outcome  $\omega_1$  appears before  $\omega_4$ , despite the fact that  $\omega_4$  beats  $\omega_1$  according to the majority graph

# Example

- How inconsistent is this ordering ?
- To have no inconsistencies in the ordering, we have to flip the edge  $(\omega_4, \omega_1)$
- This is the only edge that we would have to flip, and so we will say the cost of this order is **1**

# Example

- Consider the following rank:
- $\omega_1 >^* \omega_2 >^* \omega_4 >^* \omega_3$
- We would still have to flip the edge  $(\omega_4, \omega_1)$ , but in addition, since  $\omega_3$  beats  $\omega_4$  according to the majority graph, we would also have to flip the edge  $(\omega_3, \omega_4)$
- No other edges would need to be flipped
- The cost of this order is **2**

# Slater Ranking

- To obtain the Slater ranking, we find out the cost of each ordering, and then choose one with the lower cost
- The order:
- $\omega_1 >^* \omega_2 >^* \omega_3 >^* \omega_4$
- would be selected by the Slater ranking for this example

# Slater Ranking

- The Slater ranking tries to come up with a solution that is 'optimal' in the sense that it is as close to the majority graph as possible
- Computing the Slater ranking is NP-hard

# Desirable Properties for Voting Procedures

- 1) Pareto condition
- 2) Condorcet winner condition
- 3) Independence of irrelevant alternatives
- 4) Dictatorship

# Pareto Condition

- An outcome  $\omega$  is said to be Pareto optimal if there is no other outcome that makes one agent better off, without making another agent worse off
- The Pareto condition for voting rules says that the preference order selected should be efficient – if every voter ranks  $\omega_i$  above a candidate  $\omega_j$ , then  $\omega_i \succ^* \omega_j$
- The Pareto condition is satisfied by plurality and Borda, but not by sequential majority election

# Condorcet Winner Condition

- An outcome is said to be a Condorcet winner if it would beat every other outcome in a pairwise election
- If  $\omega_i$  is a Condorcet winner, then  $\omega_i$  should be ranked first (and so, in the case of social choice functions, is the outcome selected)
- Only the sequential majority elections satisfies the Condorcet winner condition

# Independence of Irrelevant Alternatives

- Suppose there are a number of candidates, including  $\omega_i$  and  $\omega_j$ , and voter preferences are such that  $\omega_i \succ^* \omega_j$
- You decide to change your preferences, but not about the relative ranking of  $\omega_i$  and  $\omega_j$
- You rank these the same as before

# Independence of Irrelevant Alternatives

- In this case, we should still have  $\omega_i >^* \omega_j$
- The social ranking of  $\omega_i$  and  $\omega_j$  should depend only on the way that  $\omega_i$  and  $\omega_j$  are ranked in the preference orders – this is the only thing that should be taken into account when ranking  $\omega_i$  and  $\omega_j$
- Plurality, Borda, and sequential majority elections do not satisfy the independence of irrelevant alternatives

# Dictatorship

- A social welfare function  $f$  is a **dictatorship** if for some voter  $i$ , we have:
- $f(\varpi_1, \varpi_2, \dots, \varpi_n) = \varpi_i$
- The social welfare function takes as input a preference order for every agent and then produces as output the preference order of voter  $i$
- All other voters' preferences are ignored

# Dictatorship

- This does not seem like a very reasonable social welfare function
- Plurality, Borda, and the Slater ranking are not dictatorships
- Dictatorships satisfy the Pareto condition and the independence of irrelevant alternatives

# Arrow's Theorem

- Assume that we are not in a pairwise election setting, we have  $|\Omega| > 2$
- A voting procedure is 'good' if it satisfies the Pareto condition and the independence of irrelevant alternatives condition
- But the only voting procedure satisfying these conditions is dictatorship

# Strategic Manipulation

- A social choice function takes as input a preference order for each voter, and gives as output a selected candidate:

$$f : \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n \text{ times}} \rightarrow \Omega$$

- Each voter has its own ‘true’ preference profile, though this will be private information

# Strategic Manipulation

- A voter is free to declare any preference profile he likes
- I have preferences over the possible outcomes  $\Omega$ , and the actions or strategies that I have available to me are all the possible preference orders that I can declare

# Strategic Manipulation

- Given a social choice function  $f$ , we say that  $f$  is **manipulable** if, for some collection of voter preference profiles:
- $\varpi_1, \dots, \varpi_i, \dots, \varpi_n$  and a voter  $i$ , there exists some  $\varpi_i'$  such that:
- $f(\varpi_1, \dots, \varpi_i', \dots, \varpi_n) \succ_i f(\varpi_1, \dots, \varpi_i, \dots, \varpi_n)$

# Strategic Manipulation

- A voting procedure is manipulable if a voter can obtain a better outcome for himself by unilaterally changing his preference profile (by misreporting his preferences to the voting procedure)
- Manipulability is not desirable

# Complexity of Manipulation

- The Gibbard-Satterthwaite theorem establishes that strategic manipulation is always possible
- Make a distinction between a voting procedure being **easy to compute** and **easy to manipulate**

# Complexity of Manipulation

- Easy to compute – the function  $f$  can be implemented by an algorithm that runs in time polynomial in the number of voters and candidates
- The Slater ranking was the only example that was hard to compute
- The other voting procedures are easy to compute

# Complexity of Manipulation

- Easy to manipulate – if it is possible for a voter  $i$  to obtain a more preferred outcome by declaring a preference order  $\varpi_i'$  rather than its true preference order  $\varpi_i$ , then such a  $\varpi_i'$  can be computed in polynomial time

# Complexity of Manipulation

- Are there non-dictatorial voting procedures that are easy to compute and that satisfy the Pareto condition, but that are not easy to manipulate ?
- A voting procedure called **second-order Capeland** satisfies these requirements

# Complexity of Manipulation

- While manipulation of second-order Copeland is possible in principle, to actually do such manipulation in practice may be too computationally complex
- NP-completeness is a worst case result, that is the 'hard' instances do not occur in practice
- There might be heuristics for manipulation that work well on cases of practical interest