Online Social Networks and Media

Positive and Negative Relationships

Chapter 5, from D. Easley and J. Kleinberg book

Structural Balance

What about negative edges?

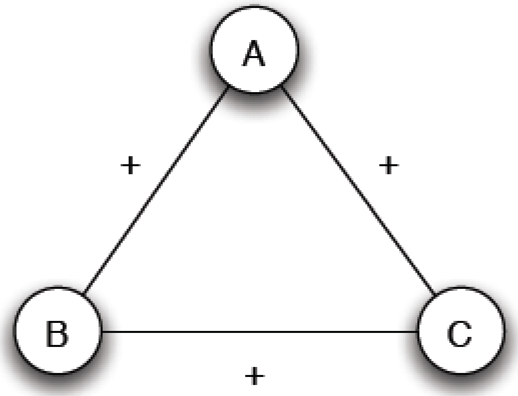
Initially, a complete graph (or clique): every edge either + or -

Let us first look at individual triangles

- Lets look at 3 people => 4 Cases
- See if all are equally possible (local property)

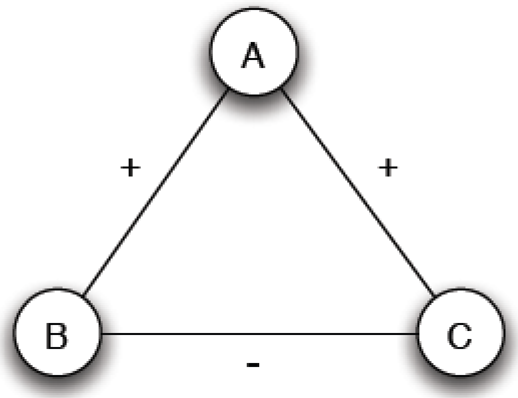
Structural Balance

Case (a): 3 +



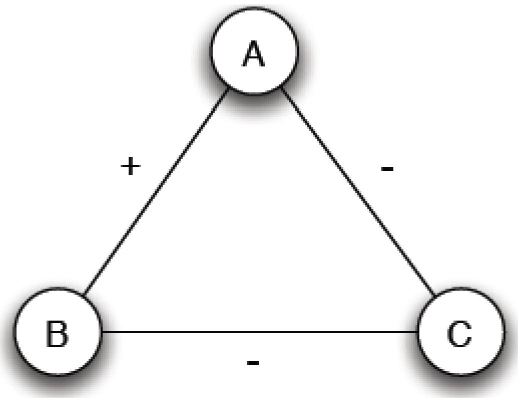
Mutual friends

Case (b): 2 +, 1 -



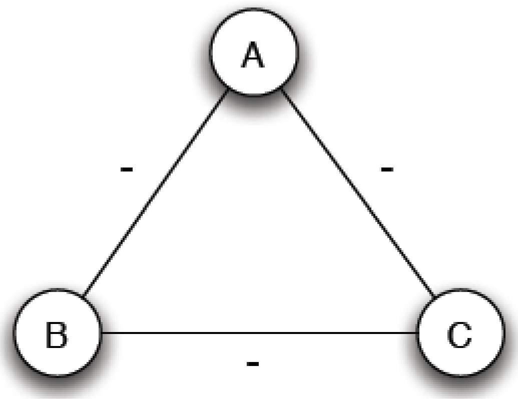
A is friend with B and C, but B and C do not get well together

Case (c): 1 +, 2 -



A and B are friends with a mutual enemy

Case (d): 3 -

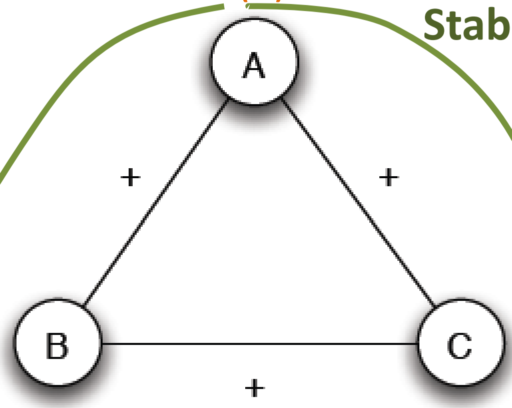


Mutual enemies

Structural Balance

Case (a): 3 +

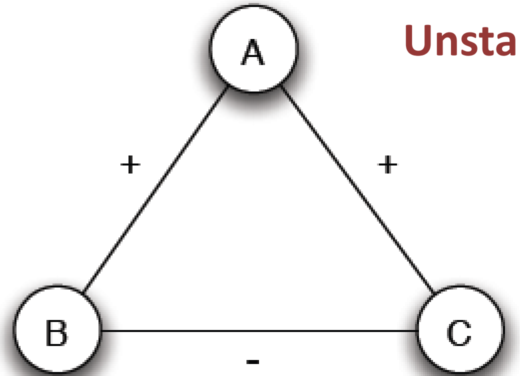
Stable or balanced



Mutual friends

Case (b): 2 +, 1 -

Unstable

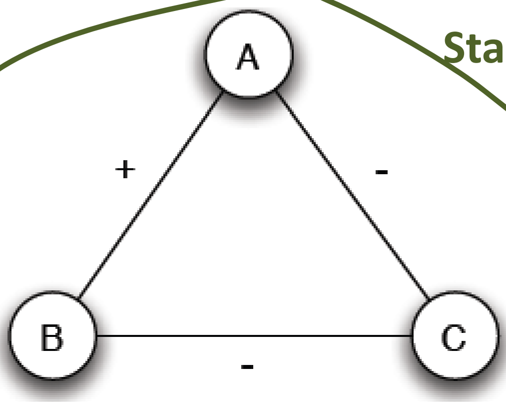


A is friend with B and C, but B and C do not get well together

Implicit force to make B and C friends (- => +) or turn one of the + to -

Case (c): 1 +, 2 -

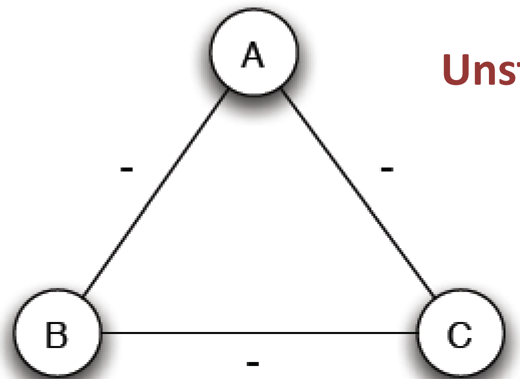
Stable or balanced



A and B are friends with a mutual enemy

Case (d): 3 -

Unstable



Mutual enemies

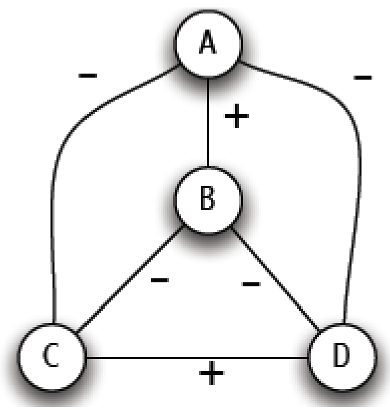
Forces to team up against the third (turn 1 – to +)

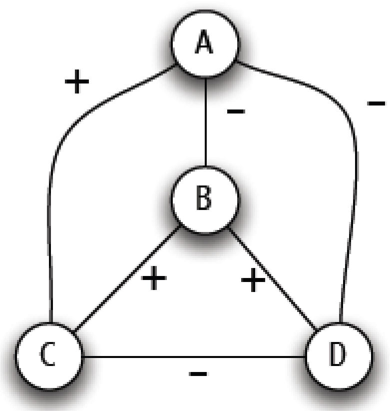
Structural Balance

Let us now define structural balance for the network

A labeled complete graph is balanced if every one of its triangles is balanced

Structural Balance Property: For every set of three nodes, if we consider the three edges connecting them, either all three of these are labeled +, or else exactly one of them is labeled -

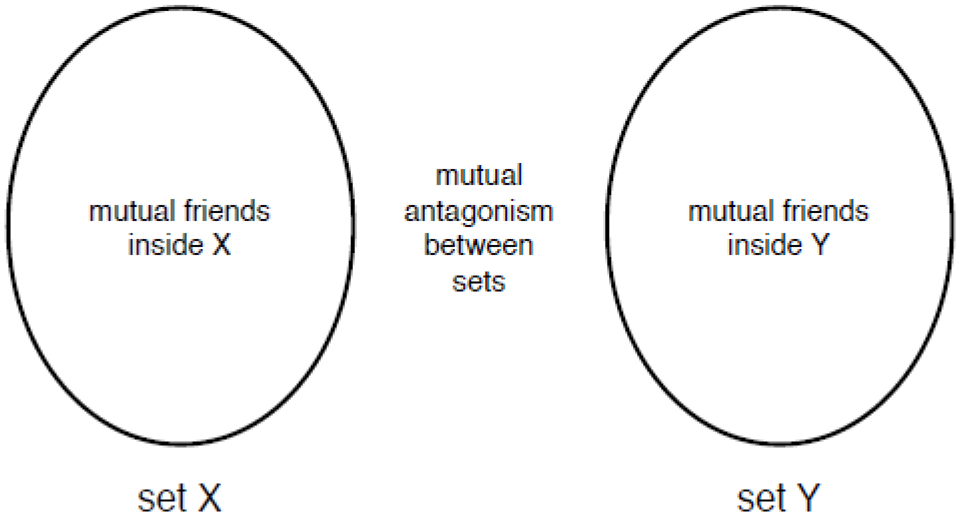




The Structure of Balanced Networks

What does a balanced network look like?

Is this balanced?

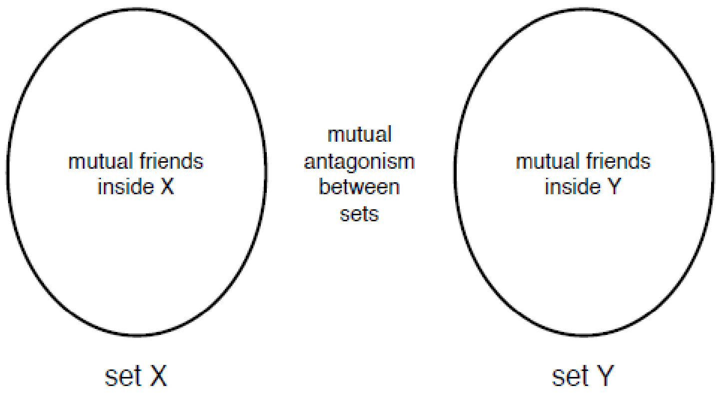


The Structure of Balanced Networks

Balance Theorem: If a labeled complete graph is balanced,

(a) all pairs of nodes are friends or,

(b) the nodes can be divided into two groups X and Y, such that every pair of nodes in X like each other, every pair of nodes in Y like each other, and every one in X is the enemy of every one in Y.

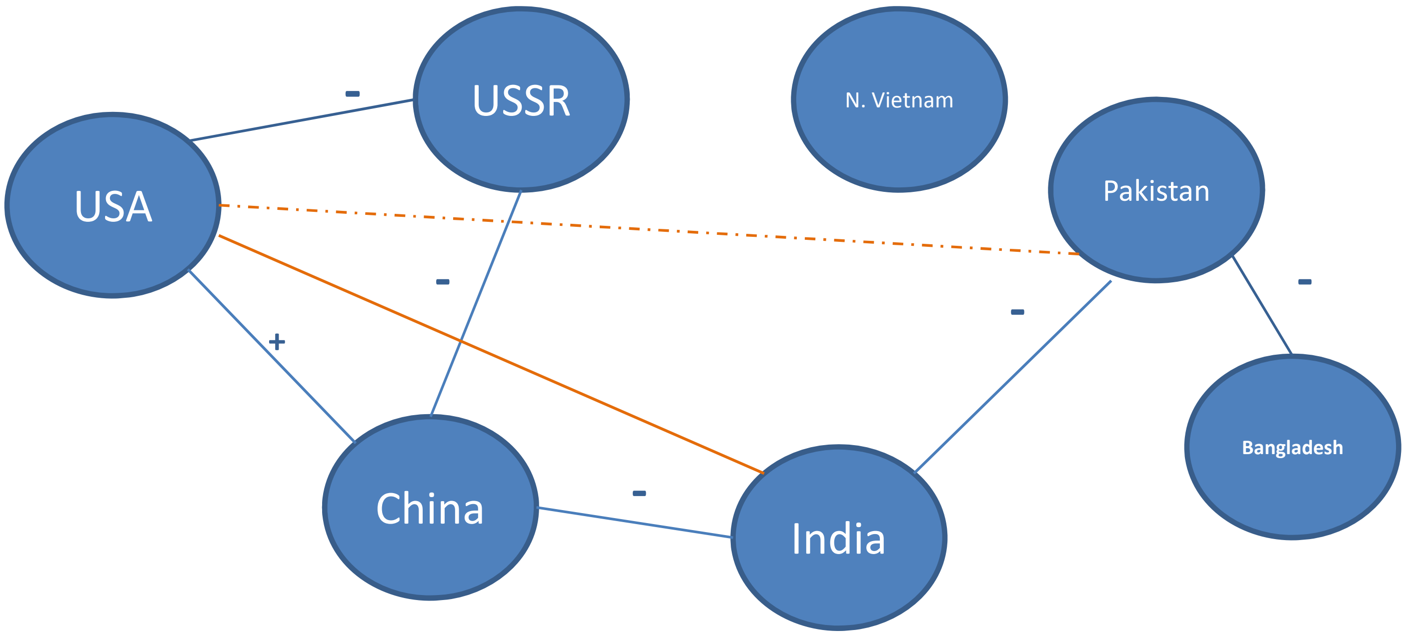


Proof ...

Applications of Structural Balance

✓International relationships (I)

The conflict of Bangladesh's separation from Pakistan in 1972 (1)

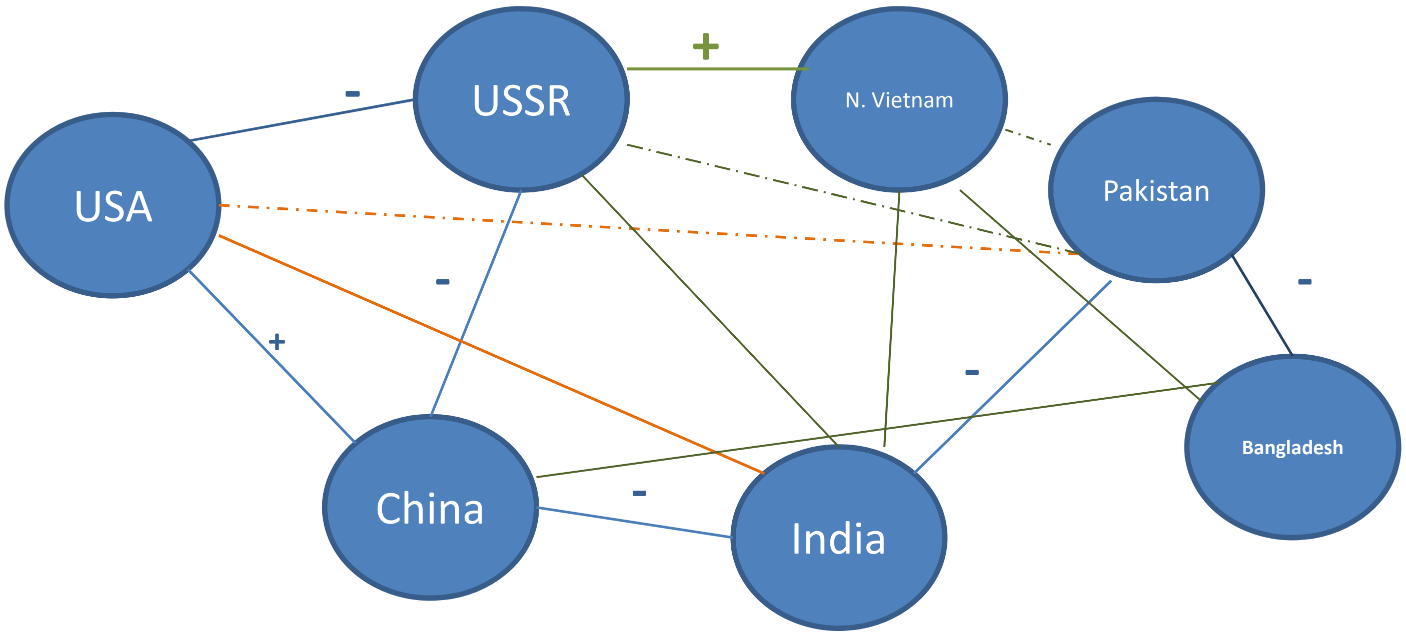


USA support to Pakistan?
China?

Applications of Structural Balance

✓International relationships (I)

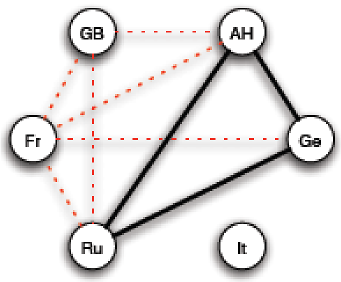
The conflict of Bangladesh's separation from Pakistan in 1972 (II)



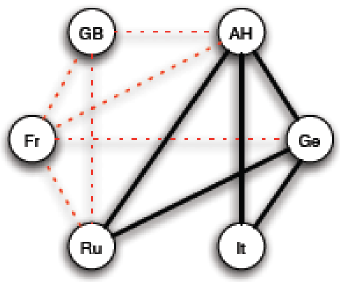
USA support to Pakistan?
China?

Applications of Structural Balance

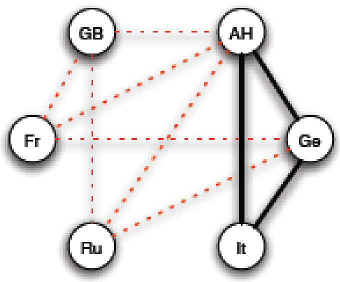
✓International relationships (II)



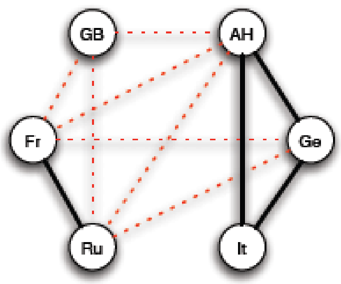
(a) *Three Emperors' League 1872–81*



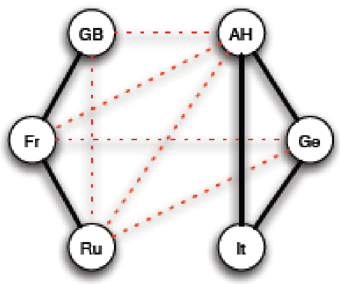
(b) *Triple Alliance 1882*



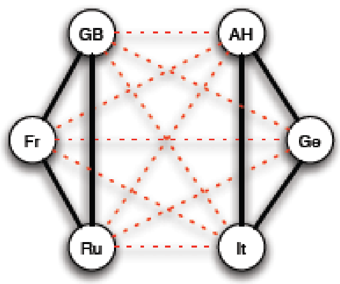
(c) *German-Russian Lapse 1890*



(d) *French-Russian Alliance 1891–94*



(e) *Entente Cordiale 1904*



(f) *British Russian Alliance 1907*

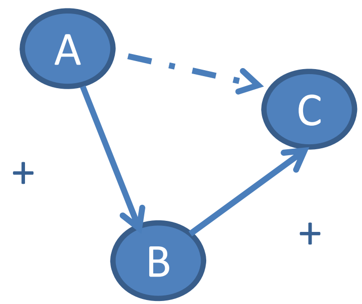
Figure 5.5: The evolution of alliances in Europe, 1872-1907 (the nations GB, Fr, Ru, It, Ge, and AH are Great Britain, France, Russia, Italy, Germany, and Austria-Hungary respectively). Solid dark edges indicate friendship while dotted red edges indicate enmity. Note how the network slides into a balanced labeling — and into World War I. This figure and example are from Antal, Krapivsky, and Redner [20].

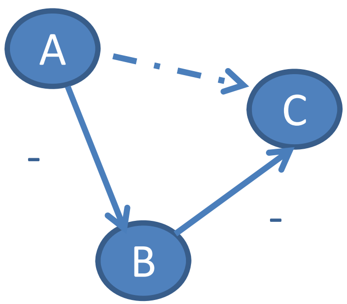
Applications of Structural Balance

✓Trust, Distrust and Online Rating

Evaluation of products and trust/distrust of other users

Directed Graphs



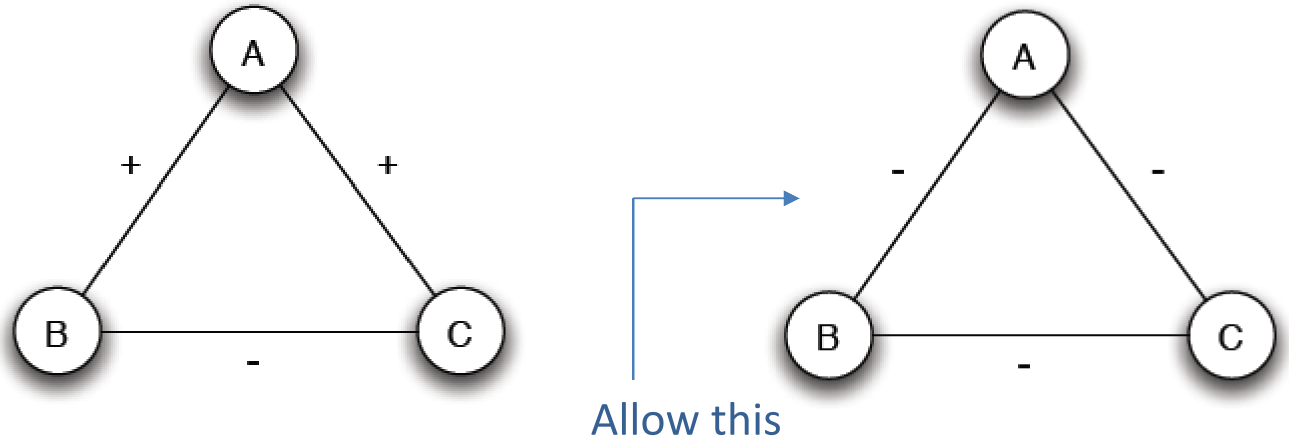


A trusts B, B trusts C, A ? C

A distrusts B, B distrusts C, A ? C
If distrust enemy relation, +
A distrusts means A is better than B, -

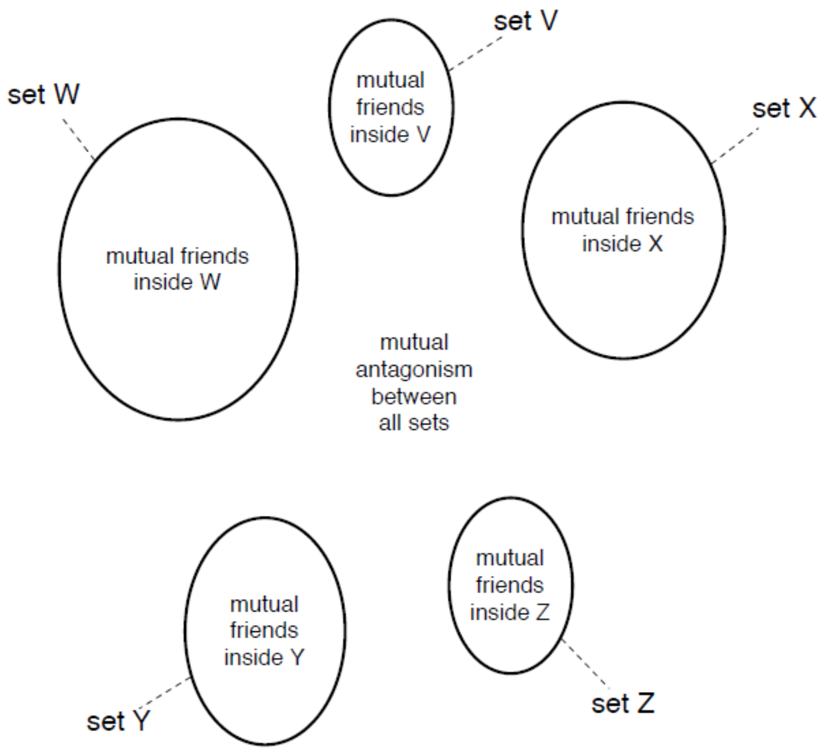
Depends on the application
Rating political books or
Consumer electronics products

A Weaker Form of Structural Balance



Weak Structural Balance Property: There is no set of three nodes such that the edges among them consist of exactly two positive edges and one negative edge

A Weaker Form of Structural Balance



Weakly Balance Theorem: If a labeled complete graph is weakly balanced, its nodes can be divided into groups in such a way that every two nodes belonging to the same group are friends, and every two nodes belonging to different groups are enemies

Proof …

A Weaker Form of Structural Balance

Two enemies of A can be either friends or enemies of each other

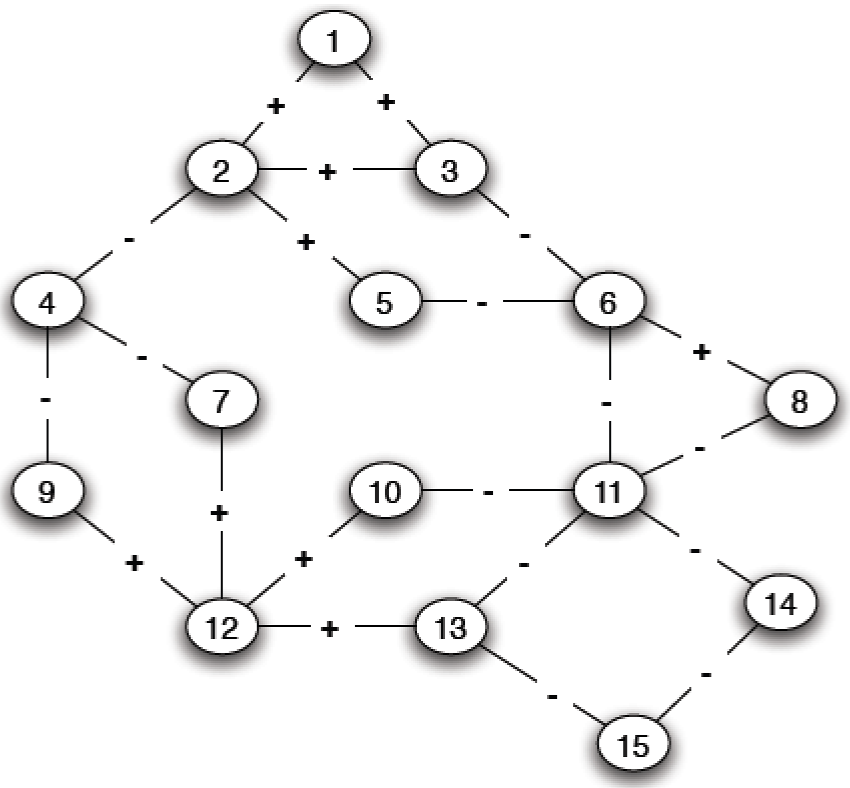
Generalizing

1. Non-complete graphs
2. Instead of all triangles, “most” triangles, approximately divide the graph

We shall use the original (“non-weak” definition of structural balance)

Structural Balance in Arbitrary Graphs

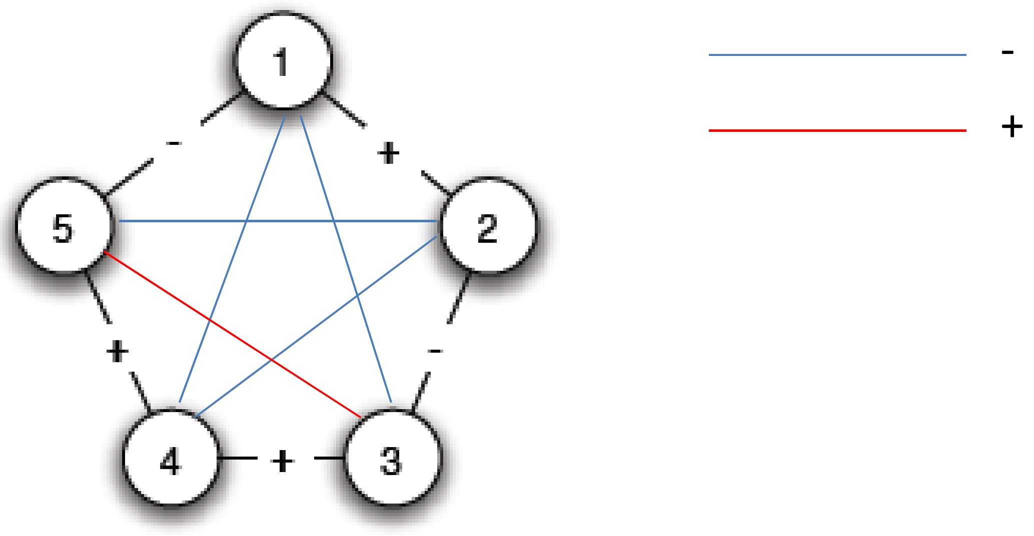
- Positive edge
- Negative edge
- Absence of an edge



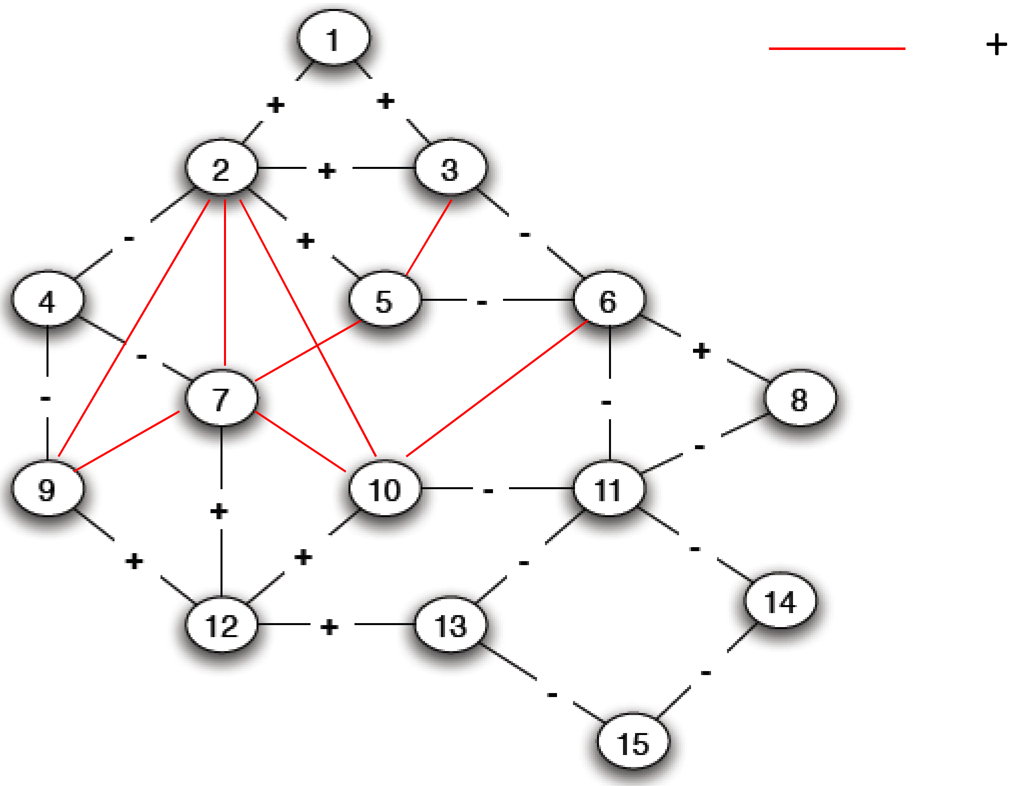
Balance Definition for General Graphs

1. Based on triangles (local view)
2. Division of the network (global view)

A (non-complete) graph is balanced if it can be completed by adding edges to form a signed complete graph that is balanced



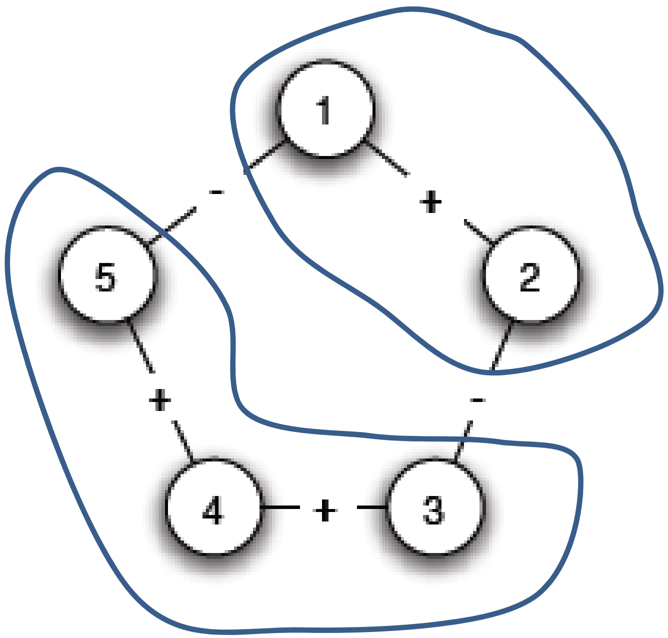
Balance Definition for General Graphs



Balance Definition for General Graphs

1. Based on triangles (local view)
2. Division of the network (global view)

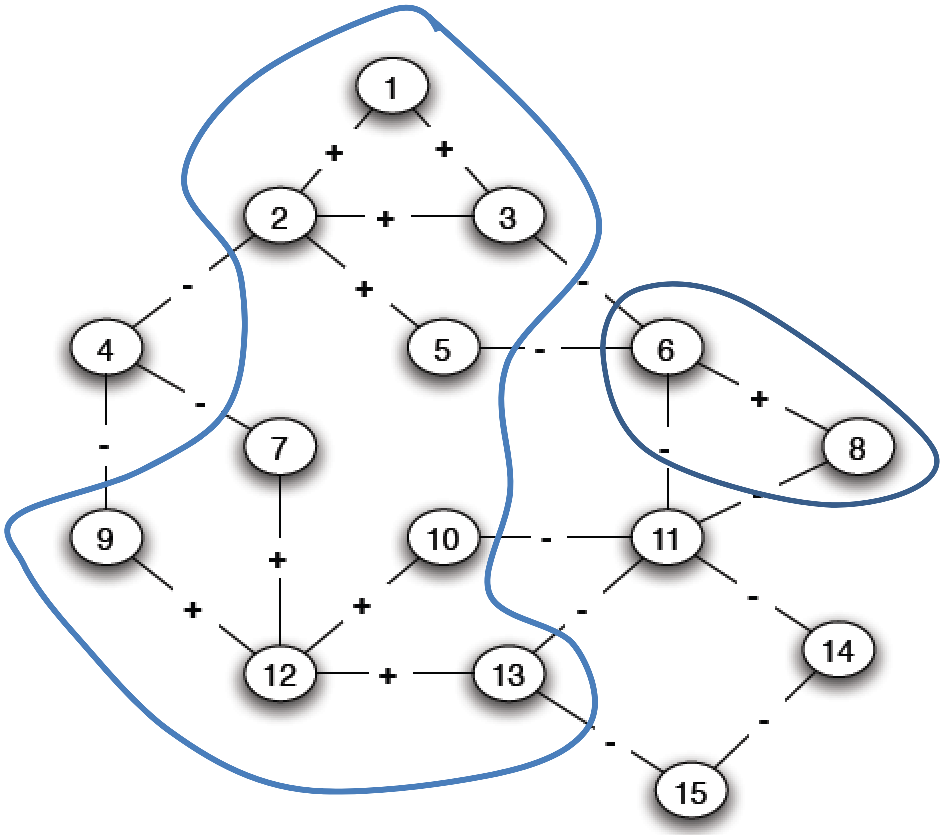
A (non-complete) graph is balanced if it possible to divide the nodes into two sets X and Y, such that any edge with both ends inside X or both ends inside Y is positive and any edge with one end in X and one end in Y is negative



The **two definition** are **equivalent**:
An arbitrary signed graph is balanced under the first definition, if and only if, it is balanced under the second definitions

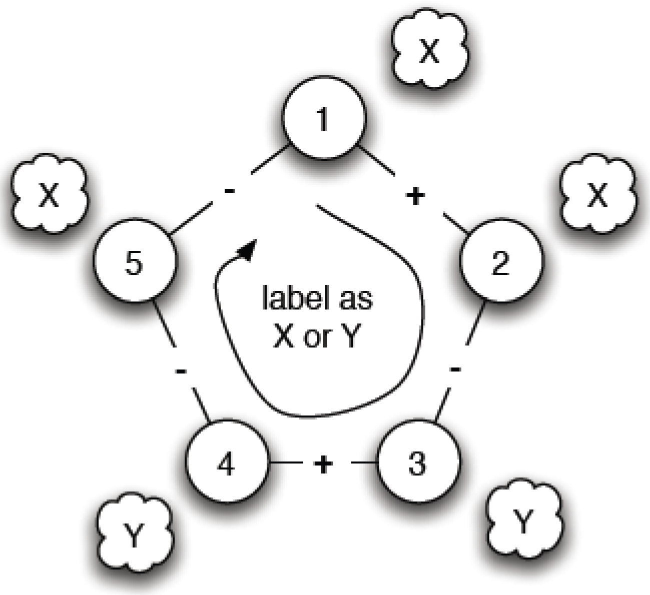
Balance Definition for General Graphs

Algorithm?



Balance Characterization

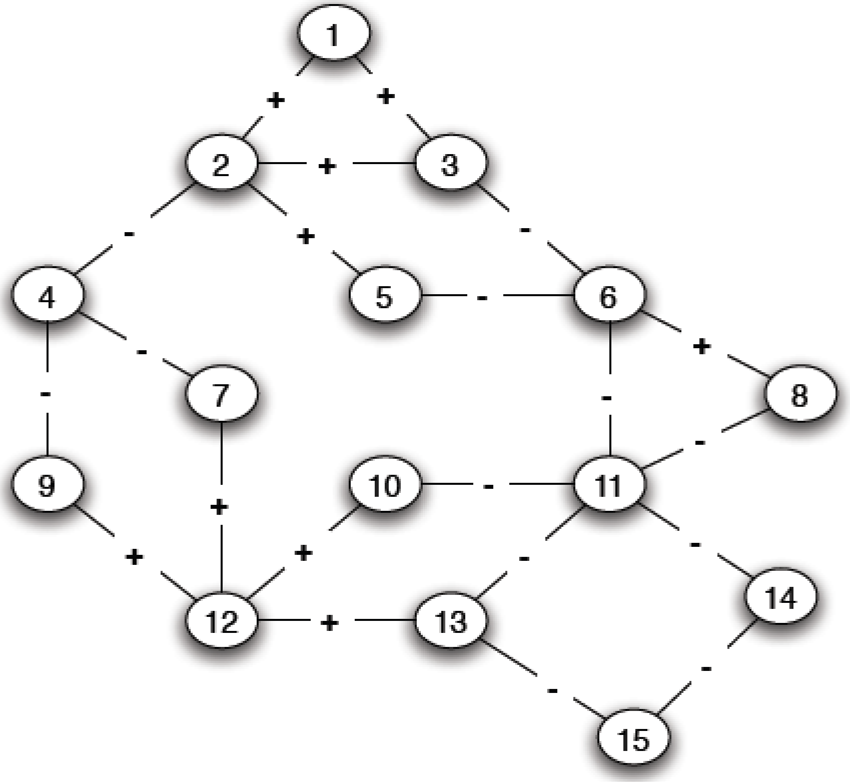
- Start from a node (1) and place nodes in X or Y
- Every time we cross a negative edge, change the set



Cycle with odd number of negative edges

Balance Definition for General Graphs

Is there such a cycle?



Balance Characterization

Claim: A signed graph is balanced, if and only if, it contains no cycles with an odd number of negative edges

(proof by construction)

Find *a balanced division:* partition into sets X and Y, all edges in X and Y positive, crossing edges negative

Either succeeds or Stops with a cycle containing an odd number of -

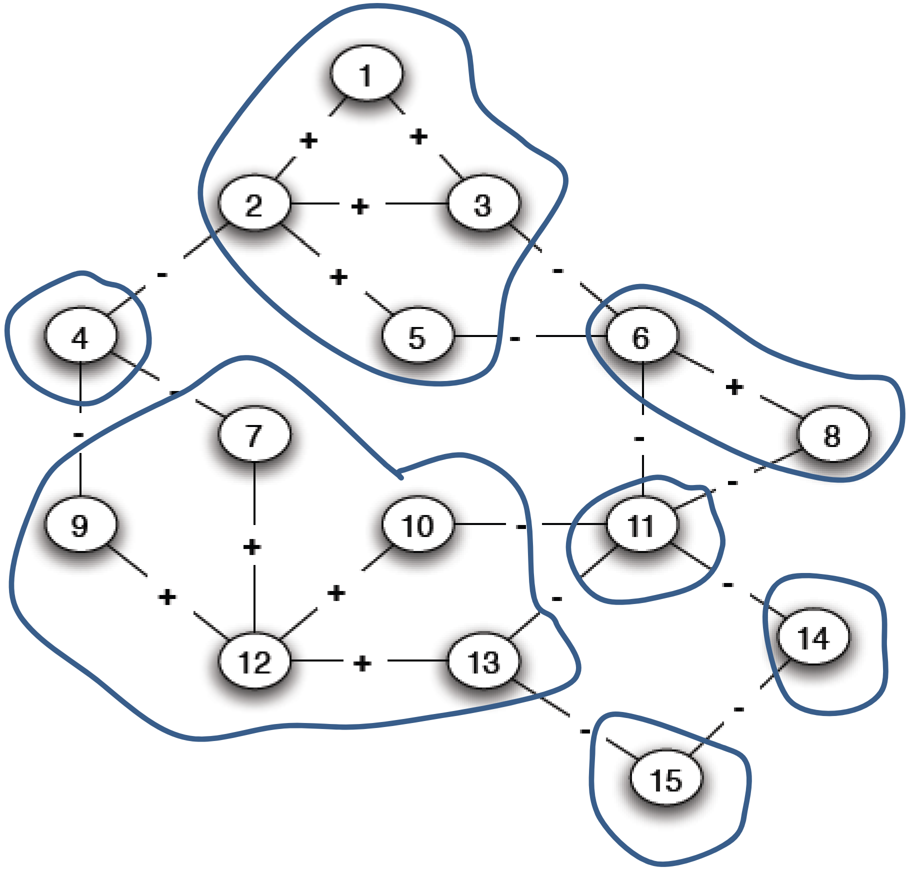
Two steps:

1. Convert the graph into a reduced one with only negative edges
2. Solve the problem in the reduced graph

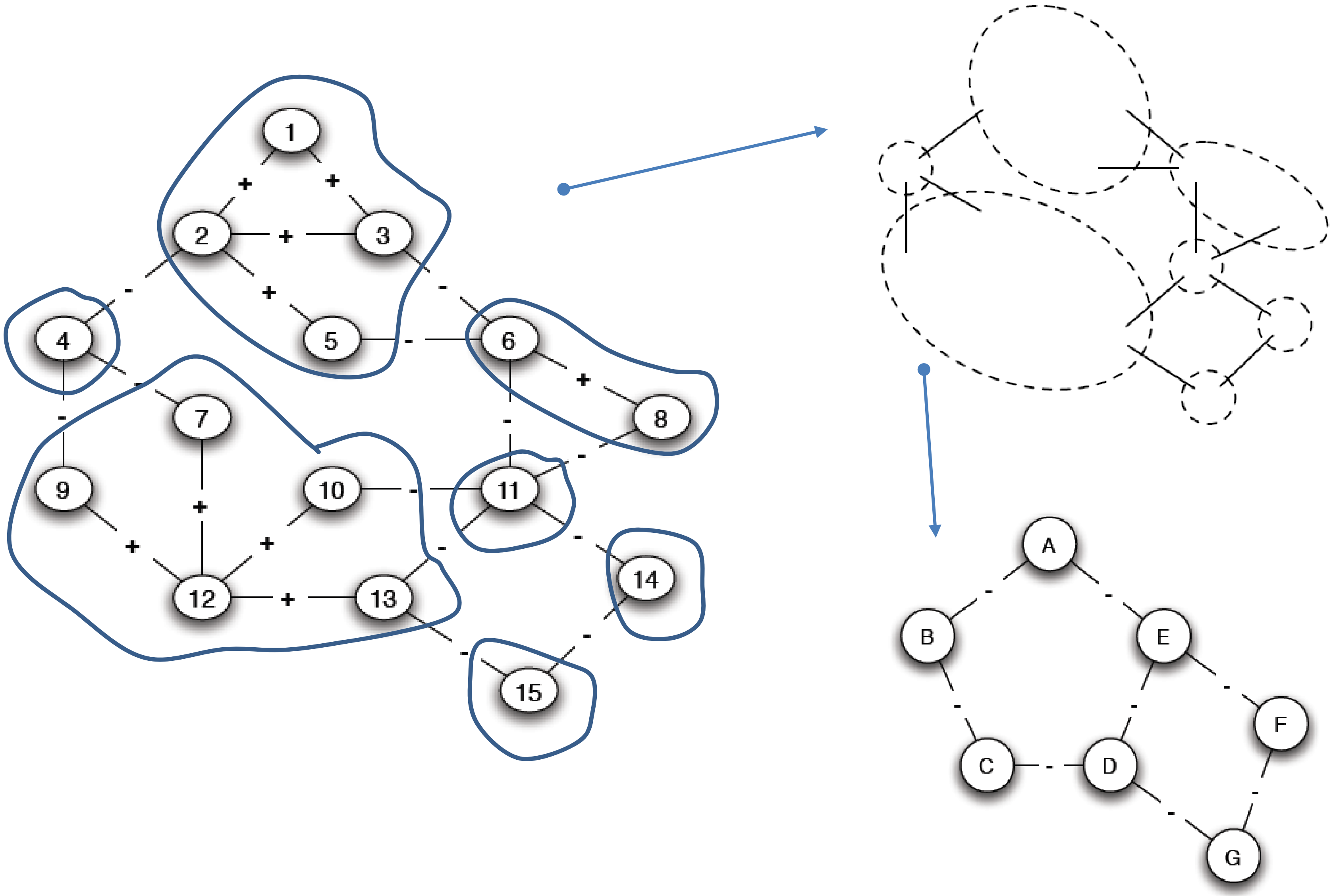
Balance Characterization

Step 1:

1. Find **connected components** by considering **only positive** edges, called supernodes
2. Supernodes contain a negative edge between any pair:
    - yes, odd cycle (1 negative)
    - no, each supernode either X or Y
3. Reduced problem: a node for each supernode, an edge between two supernodes if an edge in the original



Balance Characterization



Balance Characterization

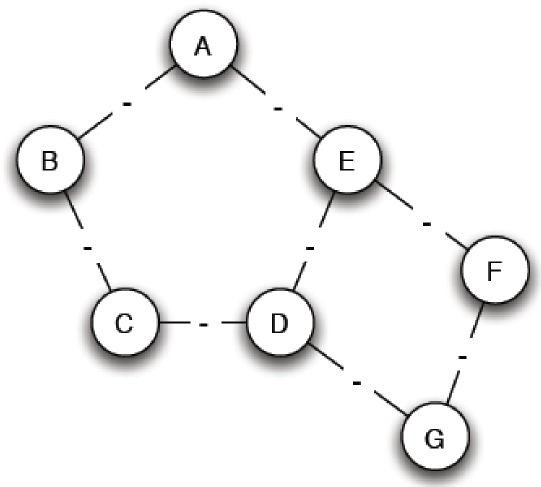
Step 2:

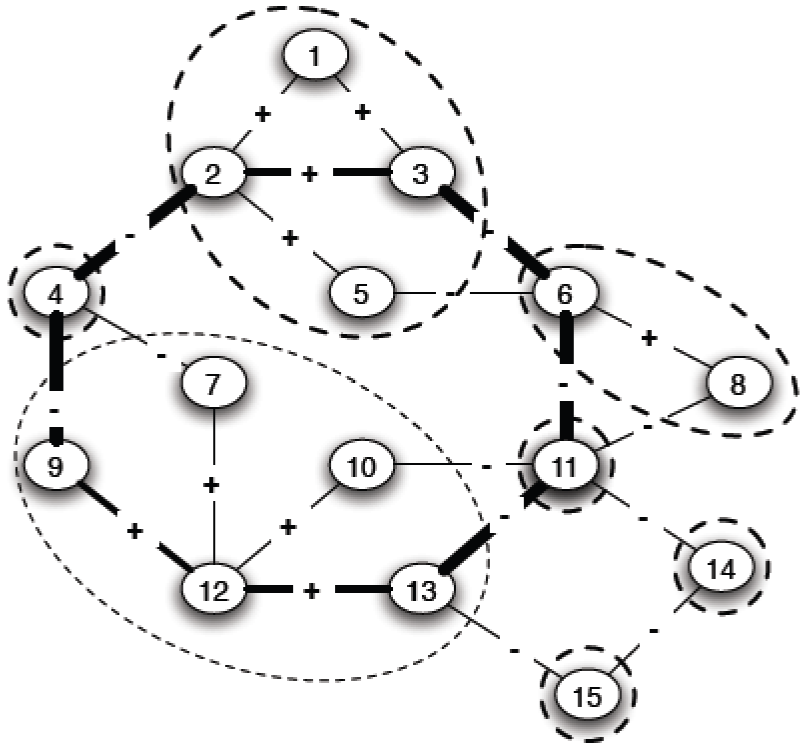
Note: Only negative edges among supernodes

Start labeling by either X and Y

If successful, then label the nodes of the supernode correspondingly

✓A cycle with an odd number, corresponds to a (possibly larger) odd cycle in the original





Balance Characterization

Step 2:

Determining whether the graph is bipartite

Use Breadth-First-Search (BFS)

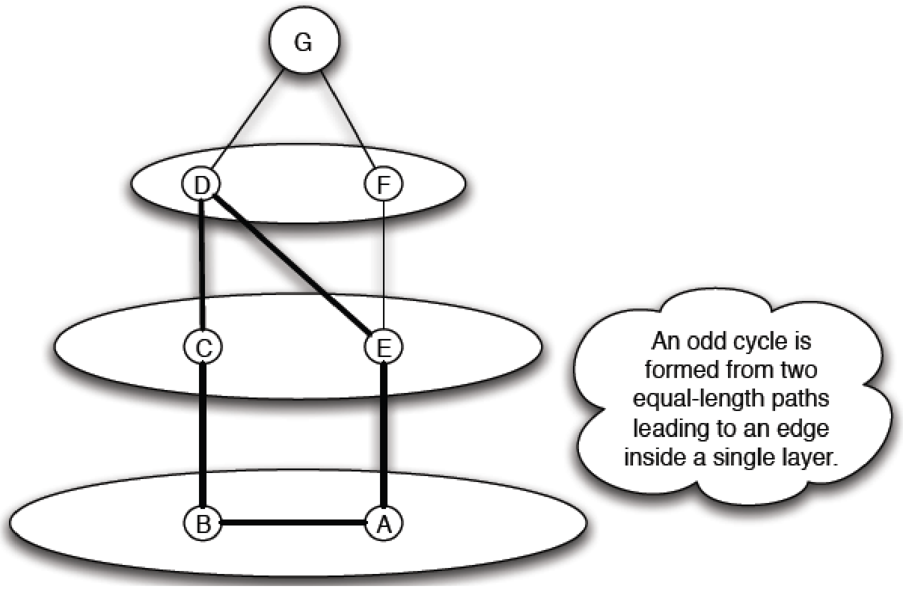
- Start the search at any node and give alternating labels to the vertices visited during the search. *That is, give label X to the starting node, Y to all its neighbors, X to those neighbors' neighbors, and so on.*

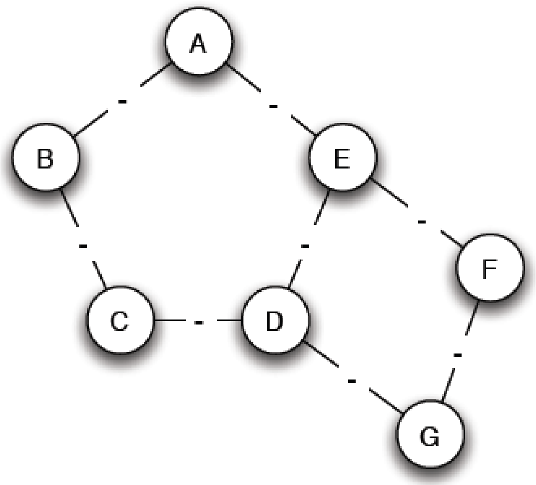
- If at any step a node has (visited) neighbors with the same label as itself, then the graph is not bipartite *(cross-level edge)*

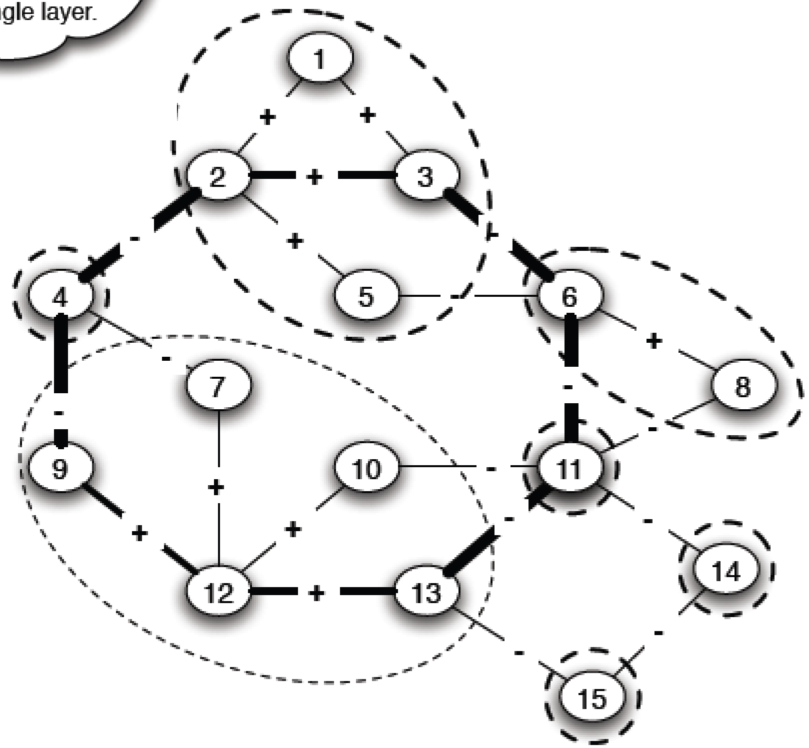
- If the search ends without such a situation occurring, then the graph is bipartite.

Why is this an “odd” cycle?

Balance Characterization







Generalizing

1. Non-complete graphs

2. Instead of all triangles, “most” triangles, approximately divide the graph

Approximately Balance Networks

a complete graph (or clique): every edge either + or -

Claim: If all triangles in a labeled complete graph are balanced, than either
(a) all pairs of nodes are friends or,
(b) the nodes can be divided into two groups X and Y, such that
 (i) every pair of nodes in X like each other,
 (ii) every pair of nodes in Y like each other, and
 (iii) every one in X is the enemy of every one in Y.

Not all, but most, triangles are balanced

Claim: If *at least 99.9%* of all triangles in a labeled compete graph are balanced, then either,
(a) There is a set consisting of *at least 90%* of the nodes in which *at least 90%* of all pairs are friends, or,
(b) the nodes can be divided into two groups X and Y, such that
 (i) *at least 90%* of the pairs in X like each other,
 (ii) *at least 90%* of the pairs in Y like each other, and
 (iii) *at least 90%* of the pairs with one end in X and one in Y are enemies

Approximately Balance Networks

Claim: If *at least 99.9%* of all triangles in a labeled complete graph are balanced, then either,

(a) There is a set consisting of *at least 90%* of the nodes in which *at least 90%* of all pairs are friends, or,

(b) the nodes can be divided into two groups X and Y, such that
   (i) *at least 90%* of the pairs in X like each other,
   (ii) *at least 90%* of the pairs in Y like each other, and
   (iii) *at least 90%* of the pairs with one end in X and one in Y are enemies

Claim: Let ε be any number, such that $0 \leq \varepsilon < 1/8$, If at *least $1 - \varepsilon$* in a labeled complete graph are balanced, then either

(a) There is a set consisting of *at least $1-\delta$* of the nodes in which *at least $1-\delta$* of all pairs are friends, or,

(b) the nodes can be divided into two groups X and Y, such that
   (i) *at least $1-\delta$* of the pairs in X like each other,
   (ii) *at least $1-\delta$* of the pairs in Y like each other, and
   (iii) *at least $1-\delta$* of the pairs with one end in X and one in Y are enemies

$$\delta = \sqrt[3]{\varepsilon}$$

Approximately Balance Networks

Basic idea – find a "good" node A (s.t., it does not belong to too many unbalanced triangles) to partition into X and Y

Pigeonhole principle: if *n* items are put into *m pigeonholes* with *n* > *m*, then at least one pigeonhole must contain more than one item

Counting argument based on pigeonhole: compute the average value of a set of objects and then argue that there must be at least one node that is equal to the average or below (or equal and above)

Approximately Balance Networks

Let a graph with N nodes

Number of edges? Number of triangles?

Weight of a node -> how many unbalanced triangles it is part of

Total weight? Average weight per node?

Choose a node whose weight is <= average

Approximately Balance Networks

Cases based on the relative size of X and Y

If X and Y too large -> case (a) else case (b)

End of Chapter 5

Balanced networks in the case of both positive and negative edges