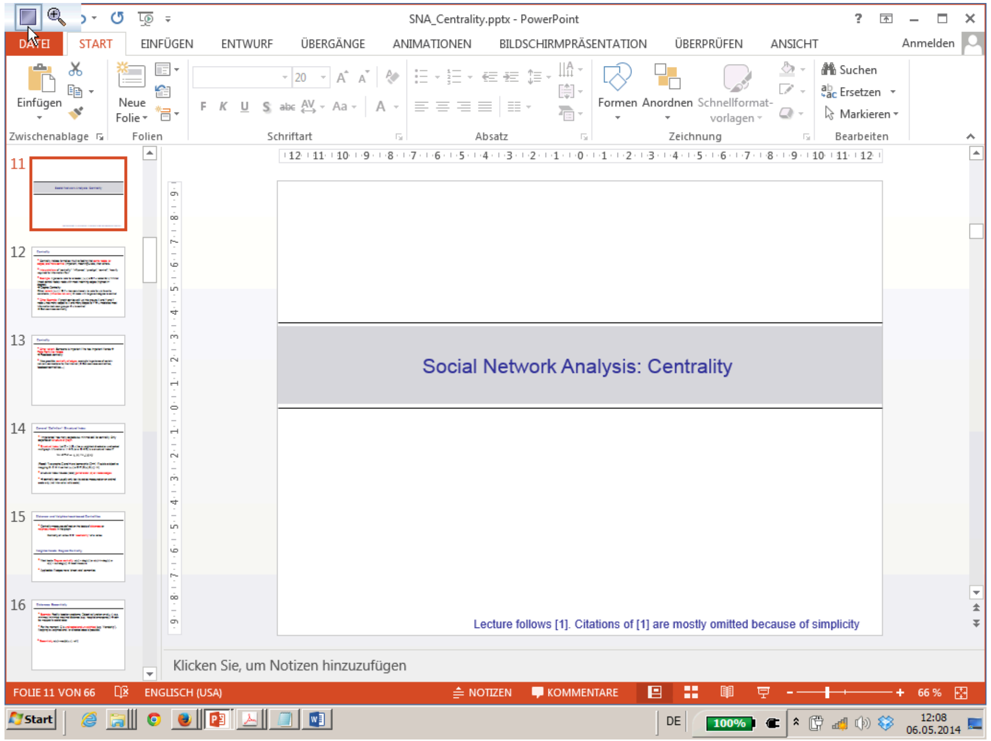
Script **generated by TTT**

Title: groh: profile1 (06.05.2014)

Date: Tue May 06 12:08:37 CEST 2014

Duration: 82:20 min

Pages: 68



General "Definition": Structural Index

- "Importance" has many aspects but minimal def. for centrality: Only depends on structure of graph:
- Structural Index: Let G = (V,E,w) be a weighted directed or undirected multigraph. A function s: V → ℝ (or s: E → ℝ) is a structural index iff

$$\forall x : G \simeq H \rightarrow s_G(x) = s_H(\phi(x))$$

(Recall: Two graphs G and H are isomorphic (G≃H) iff exists a bijective mapping Φ: G → H so that (u,v) ∈ G iff (Φ(u),Φ(v)) ∈H)

- structural index induces (total) partial-order (≤) on nodes/edges
- → centrality can usually only be viewed as measured on an ordinal scale only (not interval or ratio scale)

General "Definition": Structural Index

- "Importance" has many aspects but minimal def. for centrality: Only depends on structure of graph:
- Structural Index: Let G = (V,E,w) be a weighted directed or undirected multigraph. A function s: V → ℝ (or s: E → ℝ) is a structural index iff

$$\forall x : G \simeq H \rightarrow s_G(x) = s_H(\phi(x))$$

(Recall: Two graphs G and H are isomorphic (G≃H) iff exists a bijective mapping Φ: G → H so that (u,v) ∈ G iff (Φ(u),Φ(v)) ∈H)

- structural index induces (total) partial-order (≤) on nodes/edges
- → centrality can usually only be viewed as measured on an ordinal scale only (not interval or ratio scale)

General "Definition": Structural Index

- "Importance" has many aspects but minimal def. for centrality: Only depends on structure of graph:
- Structural Index: Let G = (V,E,w) be a weighted directed or undirected multigraph. A function s: V → ℝ (or s: E → ℝ) is a structural index iff

$$\forall x : G \simeq H \rightarrow s_G(x) = s_H(\phi(x))$$

(Recall: Two graphs G and H are isomorphic (G≃H) iff exists a bijective mapping Φ: G → H so that (u,v) ∈ G iff (Φ(u),Φ(v)) ∈H)

- structural index induces (total) partial-order (≤) on nodes/edges
- → centrality can usually only be viewed as measured on an ordinal scale only (not interval or ratio scale)

Distance- and Neighborhood-based Centralities

- Centrality-measures defined on the basis of distances or neighbourhoods in the graph:

Centrality of vertex ←→ "reachability" of a vertex

Neighborhoods: Degree Centrality

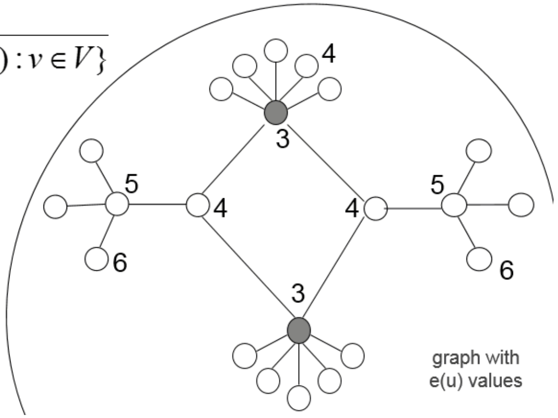
- Most basic: Degree centrality: c(u) = deg(u) (or c(u)=in-deg(u) or c(u) = out-deg(u)) → local measure
- Applicable: If edges have "direct vote" semantics

Distances: Eccentricity

- Eccentricity e(u)=max{d(u,v); v∈V}
- Center of a graph: Set of all nodes with minimum eccentricity
- Eccentricity based centrality measure:

$$c(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v) : v \in V\}}$$

- → nodes in the center of the graph have maximal centrality ☺



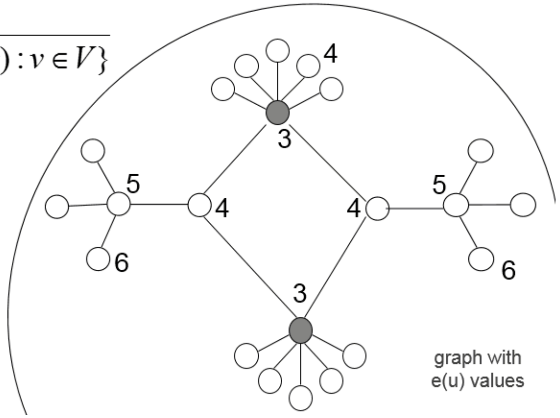
graph with e(u) values

Distances: Eccentricity

- Eccentricity e(u)=max{d(u,v); v∈V}
- Center of a graph: Set of all nodes with minimum eccentricity
- Eccentricity based centrality measure:

$$c(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v) : v \in V\}}$$

- → nodes in the center of the graph have maximal centrality ☺



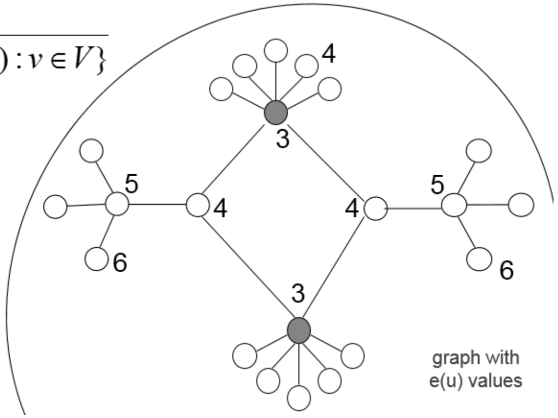
graph with e(u) values

Distances: Eccentricity

- Eccentricity e(u)=max{d(u,v); v∈V}
- Center of a graph: Set of all nodes with minimum eccentricity
- Eccentricity based centrality measure:

$$c(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v): v \in V\}}$$

- → nodes in the center of the graph have maximal centrality ☺

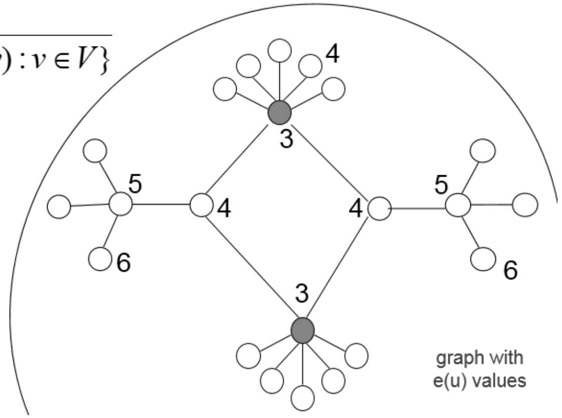


Distances: Eccentricity

- Eccentricity e(u)=max{d(u,v); v∈V}
- Center of a graph: Set of all nodes with minimum eccentricity
- Eccentricity based centrality measure:

$$c(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v): v \in V\}}$$

- → nodes in the center of the graph have maximal centrality ☺

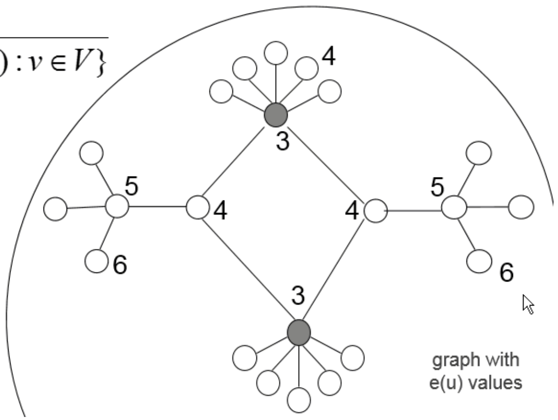


Distances: Eccentricity

- Eccentricity e(u)=max{d(u,v); v∈V}
- Center of a graph: Set of all nodes with minimum eccentricity
- Eccentricity based centrality measure:

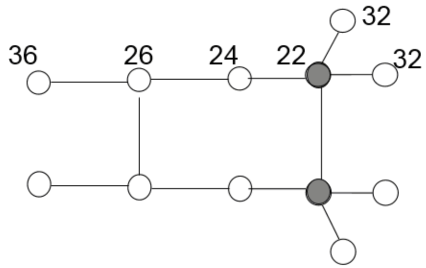
$$c(u) = \frac{1}{e(u)} = \frac{1}{\max\{d(u,v): v \in V\}}$$

- → nodes in the center of the graph have maximal centrality ☺



Distances: Closeness

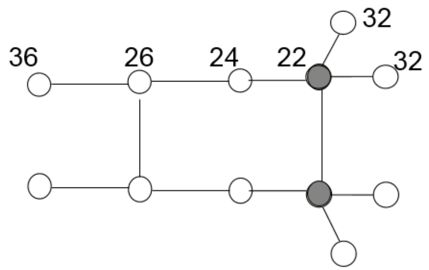
- Minisum problem: find nodes whose sum of distances to other nodes is minimal (→ service facility location problem): For all u minimize total sum of minimal distances $\sum_{v \in V} d(u,v)$
- Social analog: Determine central figure for coordination tasks
- Example:



graph with $\sum_{v \in V} d(u,v)$ values

Distances: Closeness

- Minisum problem: find nodes whose sum of distances to other nodes is minimal (→ service facility location problem): For all u minimize total sum of minimal distances $\sum_{v \in V} d(u,v)$
- Social analog: Determine central figure for coordination tasks
- Example:



graph with $\sum_{v \in V} d(u,v)$ values

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration", using out-distances „radiality" (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration", using out-distances „radiality" (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration", using out-distances „radiality" (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration“, using out-distances „radiality“ (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration“, using out-distances „radiality“ (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration“, using out-distances „radiality“ (see [6])

Distances: Closeness

- Possible resulting centrality index: closeness:

$$c(u) = \frac{1}{\sum_{v \in V} d(u,v)}$$

Only applicable to connected graphs; disconnected graph: all nodes will get the same centrality 1/∞

- Other possibility

$$c(u) = \frac{\sum_{v \in V} (\Delta_G + 1 - d(u,v))}{|V| - 1}$$

Δ_G is the diameter of the graph

  - if computed on directed graph: (set d(u,u) = 0 and set d(u,v) = 0 if u,v are unreachable via directed path → problematic !): using in-distances: „integration“, using out-distances „radiality“ (see [6])

Distances: Centroids

- Competitive objective: Given number of competitors: where to open a store (Customers will just choose store based on minimal distance)?
- Social Problem: Example: find “social ecological niche”
- Formalization: For u, v : $\gamma_u(v)$=number of vertices closer to u than to v; If one salesman selects u and competitor selects v as locations, the first will have

$$\gamma_u(v)+\frac{1}{2}(|V|-\gamma_u(v)-\gamma_v(u))=\frac{1}{2}|V|+\frac{1}{2}(\gamma_u(v)-\gamma_v(u))$$

customers

Distances: Centroids

- Competitive objective: Given number of competitors: where to open a store (Customers will just choose store based on minimal distance)?
- Social Problem: Example: find “social ecological niche”
- Formalization: For u, v : $\gamma_u(v)$=number of vertices closer to u than to v; If one salesman selects u and competitor selects v as locations, the first will have

$$\gamma_u(v)+\frac{1}{2}(|V|-\gamma_u(v)-\gamma_v(u))=\frac{1}{2}|V|+\frac{1}{2}(\gamma_u(v)-\gamma_v(u))$$

customers

Distances: Centroids

- Competitive objective: Given number of competitors: where to open a store (Customers will just choose store based on minimal distance)?
- Social Problem: Example: find “social ecological niche”
- Formalization: For u, v : $\gamma_u(v)$=number of vertices closer to u than to v; If one salesman selects u and competitor selects v as locations, the first will have

$$\gamma_u(v)+\frac{1}{2}(|V|-\gamma_u(v)-\gamma_v(u))=\frac{1}{2}|V|+\frac{1}{2}(\gamma_u(v)-\gamma_v(u))$$

customers

Distances: Centroids

- Competitive objective: Given number of competitors: where to open a store (Customers will just choose store based on minimal distance)?
- Social Problem: Example: find “social ecological niche”
- Formalization: For u, v : $\gamma_u(v)$=number of vertices closer to u than to v; If one salesman selects u and competitor selects v as locations, the first will have

$$\gamma_u(v)+\frac{1}{2}(|V|-\gamma_u(v)-\gamma_v(u))=\frac{1}{2}|V|+\frac{1}{2}(\gamma_u(v)-\gamma_v(u))$$

customers

Distances: Centroids

- Competitive objective: Given number of competitors: where to open a store (Customers will just choose store based on minimal distance)?
- Social Problem: Example: find “social ecological niche”
- Formalization: For u, v : $\gamma_u(v)$=number of vertices closer to u than to v; If one salesman selects u and competitor selects v as locations, the first will have

$$\gamma_u(v) + \frac{1}{2}(|V| - \gamma_u(v) - \gamma_v(u)) = \frac{1}{2}|V| + \frac{1}{2}(\gamma_u(v) - \gamma_v(u))$$

customers

Distances: Centroids

- → Competitor will want to minimize

$$f(u,v) = \gamma_u(v) - \gamma_v(u)$$

- → Possible centrality index: First salesman knows the strategy of the competitor and calculates for each location the worst case:

$$c(u) = \min_v \{f(u,v) : v \in V / \{u\}\}$$

- c(u) is called centroid value: measures the advantage of location u compared to other locations: Minimal loss of customers if he choses u and a competitor choses v

Distances: Centroids

- → Competitor will want to minimize

$$f(u,v) = \gamma_u(v) - \gamma_v(u)$$

- → Possible centrality index: First salesman knows the strategy of the competitor and calculates for each location the worst case:

$$c(u) = \min_v \{f(u,v) : v \in V / \{u\}\}$$

- c(u) is called centroid value: measures the advantage of location u compared to other locations: Minimal loss of customers if he choses u and a competitor choses v

Distances: Centroids

- → Competitor will want to minimize

$$f(u,v) = \gamma_u(v) - \gamma_v(u)$$

- → Possible centrality index: First salesman knows the strategy of the competitor and calculates for each location the worst case:

$$c(u) = \min_v \{f(u,v) : v \in V / \{u\}\}$$

- c(u) is called centroid value: measures the advantage of location u compared to other locations: Minimal loss of customers if he choses u and a competitor choses v

Distances: Centroids

- →Competitor will want to minimize

$$f(u,v) = \gamma_u(v) - \gamma_v(u)$$

- → Possible centrality index: First salesman knows the strategy of the competitor and calculates for each location the worst case:

$$c(u) = \min_v \{f(u,v) : v \in V / \{u\}\}$$

- c(u) is called centroid value: measures the advantage of location u compared to other locations: Minimal loss of customers if he choses u and a competitor choses v

Shortest Paths

- Indices of this section can be applied to weighted, unweighted, directed, undirected and multigraphs and to edges and vertices ("graph elements" x).
- Assume that set of all shortest paths APSP is known (e.g. by application of Floyd Warshall algorithm in $O(|V|^3)$ worst case time)
- Reminder:
  - BFS: SSSP; O(|V|+|E|) worst case time complexity, edge-weights==1
  - Djikstra: SSSP; O(|V| log|V| +|E|) with Fibonacci heap; edge-weights ≥ 0
  - Floyd Warshall: APSP, $O(|V|^3)$ worst case time, arbitrary weights, no negative cycles allowed (but can be detected via the alg.), dynamic programming;
  - Bellman Ford: SSSP; O(|V| |E|), arbitrary weights, no negative cycles allowed (but can be detected via the alg.)

Shortest Paths: Shortest Path Betweenness

- Again assume that communication (workflows etc.) happen along shortest paths only. Let

$$\delta_{ab}(v) = \frac{\sigma_{ab}(v)}{\sigma_{ab}}$$

with σ_{ab}: total number of shortest paths between nodes a and b.

Interpretation. Probability that v is involved in a communication between a and b

Shortest Paths: Shortest Path Betweenness

- Shortest Path Betweenness (SPB) centrality is then:

$$c(v) = \sum_{a \neq v} \sum_{b \neq v} \delta_{ab}(v)$$

- Interpretation: Control that v exerts on the communication in the graph
- Also applicable to disonnected graphs
- Algorithm by Ulrik Brandes computes SPB in $O(|V||E| + |V|^2 \log|V|)$ time

Shortest Paths: Shortest Path Betweenness

- Define c_SPB for edges analogously

$$c(e) = \sum_{a \in V} \sum_{b \in V} \delta_{ab}(e)$$

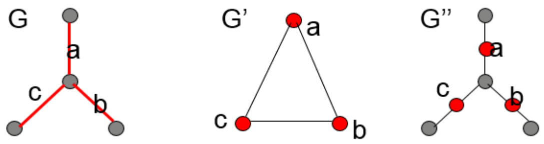
- Possible: Interpret quantity $\delta_{ab}(v)$ as general relative information flow through v (“rush”)
- Other variants: Instead of shortest paths between a and b regard
  - the set of all paths
  - the set of the k-shortest paths (interesting for social case; choose small k)
  - the set of the k-shortest node disjoint paths
  - the set of paths not longer than (1+ε)d(a,b)

k-shortest paths: paths not longer than k

Deriving edge centralities from vertex centralities

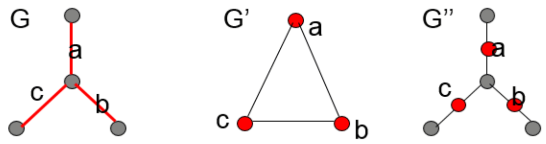
- What we have seen so far: Various centrality measures mostly for vertices (based on degree, closeness, betweenness)
- → Formal way to translate a given vertex centrality index to a corresponding edge centrality: Apply the given vertex centrality to a transformed version of G, the edge graph
- Given original G =(V,E) then the edge graph G’ = (E,K) is defined by taking original edges as vertices. Two original edges are connected in G’ if they are originally incident to the same original node.
- Size of G’ may be quadratic (w.r.t. number of nodes) compared to G

Deriving edge centralities from vertex centralities



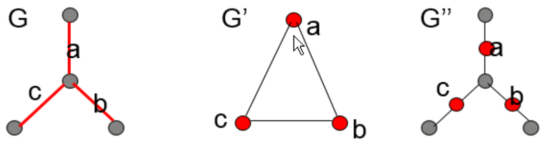
- Remember: Vertex stress centrality for node x: Number of shortest paths that use x; Straightforward version for edge e: Number of shortest paths that use e;
- → Upper Example: G: Stress centrality of edge a would be 3; But in edge graph G’ stress centrality of original edge a (now a node) is 0.
- → Formal translations of vertex centrality indices to edge centralities with edge graphs are not well suited for all purposes
- → Introduce incidence graph G’’: Each original edge is split by new “edge vertex” that represents the edge → Now vertex indices can be applied, preserving the intuition.

Deriving edge centralities from vertex centralities



- Remember: Vertex stress centrality for node x: Number of shortest paths that use x; Straightforward version for edge e: Number of shortest paths that use e;
- → Upper Example: G: Stress centrality of edge a would be 3; But in edge graph G’ stress centrality of original edge a (now a node) is 0.
- → Formal translations of vertex centrality indices to edge centralities with edge graphs are not well suited for all purposes
- → Introduce incidence graph G’’: Each original edge is split by new “edge vertex” that represents the edge → Now vertex indices can be applied, preserving the intuition.

Deriving edge centralities from vertex centralities



- Remember: Vertex stress centrality for node x: Number of shortest paths that use x; Straightforward version for edge e: Number of shortest paths that use e;
- → Upper Example: G: Stress centrality of edge a would be 3; But in edge graph G' stress centrality of original edge a (now a node) is 0.
- → Formal translations of vertex centrality indices to edge centralities with edge graphs are not well suited for all purposes
- → Introduce incidence graph G'': Each original edge is split by new "edge vertex" that represents the edge → Now vertex indices can be applied, preserving the intuition.

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})
- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})
- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})
- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})

- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})

- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Computing a flow f: E → $\mathbb{R}$ of maximum value (tweaking the local flows): $O(|V|\,|E|\,\log(|V|^2/|E|))$ (Algorithm by Goldberg & Tarjan (see [2]))

- Now define quality measure by e.g.:

$$q(G) = \sum_{s,t \in V} \max f(s,t)$$

- Social analog of flow: Workflow, Information-flow, “Doing favors flow” etc.

Vitality

- Computing a flow f: E → $\mathbb{R}$ of maximum value (tweaking the local flows): $O(|V|\,|E|\,\log(|V|^2/|E|))$ (Algorithm by Goldberg & Tarjan (see [2]))

- Now define quality measure by e.g.:

$$q(G) = \sum_{s,t \in V} \max f(s,t)$$

- Social analog of flow: Workflow, Information-flow, “Doing favors flow” etc.

Vitality

- Computing a flow f: E → ℝ of maximum value (tweaking the local flows): O(|V| |E| log(|V|²/|E|)) (Algorithm by Goldberg & Tarjan (see [2]))
- Now define quality measure by e.g.:

$$q(G) = \sum_{s,t \in V} \max f(s,t)$$

- Social analog of flow: Workflow, Information-flow, "Doing favors flow" etc.

Vitality

- Intuition: Measure importance of vertex (or edge) by the difference of a given quality measure q on G with or without the vertex (edge):
  - → Vitality v(x) of graph element x : v(x) = q(G) - q(G\{x})
- Example 1 for quality measure q: Flow:
  - Given directed graph G with positive edge weights w modeling capacities. The flow $f(s,t)$ from node s (source) to node t (sink) is defined as:

$$f(s,t) = \sum_{e \in \{Out-Edges\ of\ s\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ t\}} \tilde{f}(e)$$

where the local flows $\tilde{f}$ respect capacity contraints: $0 \le \tilde{f}(e) \le w(e)$ and balance conditions:

$$\forall v \in V \setminus \{s,t\}: \sum_{e \in \{Out-Edges\ of\ v\}} \tilde{f}(e) = \sum_{e \in \{In-Edges\ of\ v\}} \tilde{f}(e)$$

Vitality

- Possible Interpretation: Distance d(v,w) represents costs to send message from v to w
- If x is a cut-vertex or bridge-edge → Graph is disconnected after removal → centrality cannot be computed.

Stress Centrality as Vitality

- We had: stress centrality of v or e is equal to number of shortest paths through v or e

$$c_{stress}(v) = \sum_{a \in V; a \neq v} \sum_{b \in V; b \neq v} \sigma_{ab}(v) \qquad c_{stress}(e) = \sum_{a \in V} \sum_{b \in V} \sigma_{ab}(e)$$

- Intuition: $c_{stress}(v)$ seems to measure the number of shortest paths that would be lost if v wasn't avaliable any more
- Why can't we directly use c_{stress} as a graph quality index to construct a vitality index ?
- →Because actual number of shortest paths can INCREASE if e.g. edge is taken away

Stress Centrality as Vitality

- → In order to define a vitality-like version of stress: Consider only those shortest paths that haven't changed their length:

$$c_{vitality}(v,G) = c_{stress}(v,G) - c_{stress}(v, G \setminus \{v\})$$

with

$$c_{stress}(v, G \setminus \{v\}) = \sum_{a \in V; a \neq v} \sum_{b \in V; b \neq v} \sigma_{ab}[d_G(a,b) = d_{G \setminus \{v\}}(a,b)]$$

(Iverson notation)

Stress Centrality as Vitality

- → In order to define a vitality-like version of stress: Consider only those shortest paths that haven't changed their length:

$$c_{vitality}(v,G) = c_{stress}(v,G) - c_{stress}(v, G \setminus \{v\})$$

with

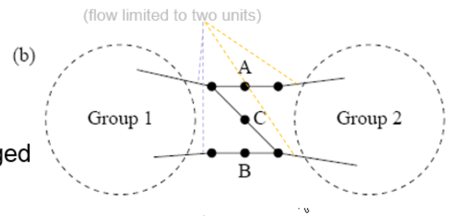
$$c_{stress}(v, G \setminus \{v\}) = \sum_{a \in V; a \neq v} \sum_{b \in V; b \neq v} \sigma_{ab}[d_G(a,b) = d_{G \setminus \{v\}}(a,b)]$$

(Iverson notation)

Critique on Betweeness Based Centralities

- major critique: Max-Flow betweenness centrality (suggested to counteract this drawback) may exhibit similar problems

- here: special Max-Flow betweenness centrality mfb:
  -- limit edge capacity to one
  -- mfb(i) := maximum possible flow through i over all possible solutions to the s-t-maximum flow problem, averaged over all s and t.



(b) In calculations of flow betweenness, vertices A and B in this configuration will get high scores while vertex C will not.

Source: [5]

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- random walk based centrality rwb: idea:
  rwb(i) := number of times that a random walk starting at s and ending at t passes through i along the way, averaged over all s and t

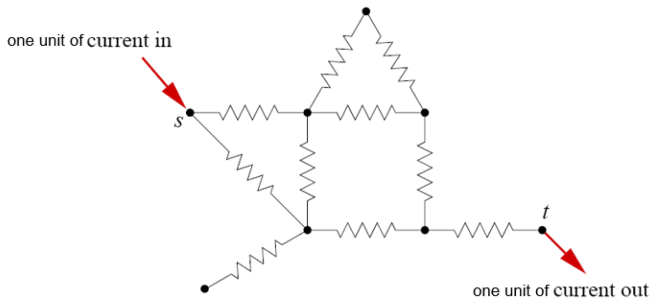
- rwb ↔ spb: opposite ends:
  - rwb: info has no idea where its going
  - spb: info knows exactly where its going

- compute for all i rwb(i): $O((m+n)n^2)$ worst case time complexity (comparable to spb)

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- flow of electric current in a resistor network;
V_i = voltage (potential) at vertex i

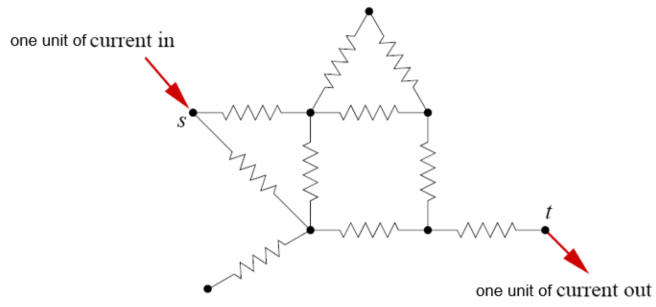
- ↔ Current Flow betweenness cfb centrality : cfb(i) := amount of current that flows through i in this setup, averaged over all s and t.



Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- flow of electric current in a resistor network;
V_i = voltage (potential) at vertex i

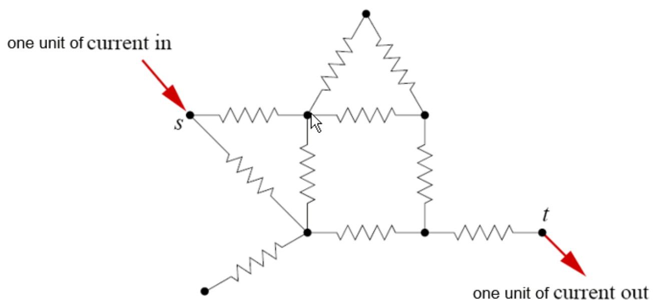
- ↔ Current Flow betweenness cfb centrality : cfb(i) := amount of current that flows through i in this setup, averaged over all s and t.



Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- flow of electric current in a resistor network;
V_i = voltage (potential) at vertex i

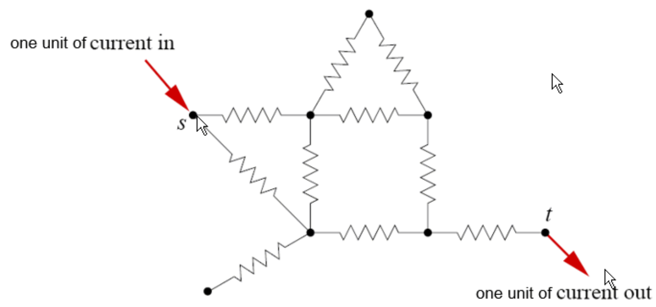
- ↔ Current Flow betweenness cfb centrality : cfb(i) := amount of current that flows through i in this setup, averaged over all s and t.



Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- flow of electric current in a resistor network;
V_i = voltage (potential) at vertex i

- ↔ Current Flow betweenness cfb centrality : cfb(i) := amount of current that flows through i in this setup, averaged over all s and t.



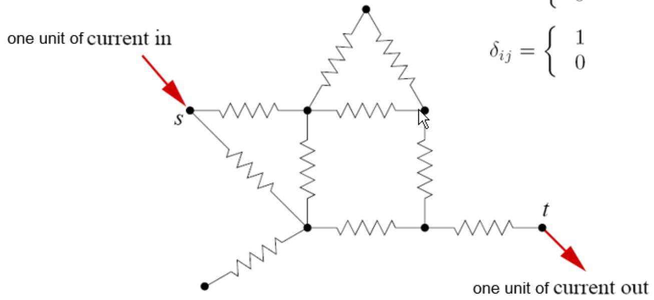
Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- Kirchhoffs point law (current conservation): total current flow in / out of node is zero:

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it},$$

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$



Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$\sum_j A_{ij} = k_i$, the degree of vertex i

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it} \longrightarrow \underbrace{(\mathbf{D} - \mathbf{A})}_{\text{"Graph Laplacian"}} \cdot \mathbf{V} = \mathbf{s}$$

$\mathbf{D}$ is the diagonal matrix with elements $D_{ii} = k_i$

source vector $\mathbf{s}$ $\quad s_i = \begin{cases} +1 & \text{for } i = s, \\ -1 & \text{for } i = t, \\ 0 & \text{otherwise.} \end{cases}$

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$\sum_j A_{ij} = k_i$, the degree of vertex i

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it} \longrightarrow \underbrace{(\mathbf{D} - \mathbf{A})}_{\text{"Graph Laplacian"}} \cdot \mathbf{V} = \mathbf{s}$$

$\mathbf{D}$ is the diagonal matrix with elements $D_{ii} = k_i$

source vector $\mathbf{s}$ $\quad s_i = \begin{cases} +1 & \text{for } i = s, \\ -1 & \text{for } i = t, \\ 0 & \text{otherwise.} \end{cases}$

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$\sum_j A_{ij} = k_i$, the degree of vertex i

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it} \longrightarrow \underbrace{(\mathbf{D} - \mathbf{A})}_{\text{"Graph Laplacian"}} \cdot \mathbf{V} = \mathbf{s}$$

$\mathbf{D}$ is the diagonal matrix with elements $D_{ii} = k_i$

source vector $\mathbf{s}$ $\quad s_i = \begin{cases} +1 & \text{for } i = s, \\ -1 & \text{for } i = t, \\ 0 & \text{otherwise.} \end{cases}$

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$\sum_j A_{ij} = k_i$, the degree of vertex i

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it} \longrightarrow \underbrace{(\mathbf{D}-\mathbf{A})}\cdot \mathbf{V} = \mathbf{s}$$

"Graph Laplacian"

$\mathbf{D}$ is the diagonal matrix with elements $D_{ii} = k_i$

source vector $\mathbf{s}$ $$s_i = \begin{cases} +1 & \text{for } i = s, \\ -1 & \text{for } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$$\underbrace{(\mathbf{D}-\mathbf{A})}\cdot \mathbf{V} = \mathbf{s}$$

Laplacian is not invertible, det = 0, because system of eq. is overdetermined → set one V_v=0 and measure voltages relative to v. → remove the v-th row and column (since V_v=0) → now invertible

$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$ (matrix inversion: $O(n^3)$)

let us now add a vth row and column back into $(\mathbf{D}_v - \mathbf{A}_v)^{-1}$ with values all equal to zero.
The resulting matrix we will denote $\mathbf{T}$.

$\longrightarrow$ $V_i^{(st)} = T_{is} - T_{it}$

$\longrightarrow$ current flow at node i: $I_i^{(st)} = \frac{1}{2} \sum_j A_{ij} |V_i^{(st)} - V_j^{(st)}|$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$$\underbrace{(\mathbf{D}-\mathbf{A})}\cdot \mathbf{V} = \mathbf{s}$$

Laplacian is not invertible, det = 0, because system of eq. is overdetermined → set one V_v=0 and measure voltages relative to v. → remove the v-th row and column (since V_v=0) → now invertible

$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$ (matrix inversion: $O(n^3)$)

let us now add a vth row and column back into $(\mathbf{D}_v - \mathbf{A}_v)^{-1}$ with values all equal to zero.
The resulting matrix we will denote $\mathbf{T}$.

$\longrightarrow$ $V_i^{(st)} = T_{is} - T_{it}$

$\longrightarrow$ current flow at node i: $I_i^{(st)} = \frac{1}{2} \sum_j A_{ij} |V_i^{(st)} - V_j^{(st)}|$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$\sum_j A_{ij} = k_i$, the degree of vertex i

$$\sum_j A_{ij}(V_i - V_j) = \delta_{is} - \delta_{it} \longrightarrow \underbrace{(\mathbf{D}-\mathbf{A})}\cdot \mathbf{V} = \mathbf{s}$$

"Graph Laplacian"

$\mathbf{D}$ is the diagonal matrix with elements $D_{ii} = k_i$

source vector $\mathbf{s}$ $$s_i = \begin{cases} +1 & \text{for } i = s, \\ -1 & \text{for } i = t, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{V} = (\mathbf{D}_v - \mathbf{A}_v)^{-1} \cdot \mathbf{s}$$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

$$\underbrace{(\mathbf{D}-\mathbf{A})}\cdot\mathbf{V}=\mathrm{s}$$

Laplacian is not invertible, det = 0, because system of eq. is overdetermined → set one V_v=0 and measure voltages relative to v. → remove the v-th row and column (since V_v=0) → now invertible

$$\mathbf{V}=(\mathbf{D}_v-\mathbf{A}_v)^{-1}\cdot\mathrm{s}$$ (matrix inversion: O(n^3))

let us now add a vth row and column back into $(\mathbf{D}_v-\mathbf{A}_v)^{-1}$ with values all equal to zero.
The resulting matrix we will denote $\mathbf{T}$.

$$\longrightarrow\ V_i^{(st)}=T_{is}-T_{it}$$

⟶ current flow at node i: $I_i^{(st)}=\frac{1}{2}\sum_j A_{ij}|V_i^{(st)}-V_j^{(st)}|$

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- current flow at node i:

$$I_i^{(st)}=\frac{1}{2}\sum_j A_{ij}|V_i^{(st)}-V_j^{(st)}|$$
$$=\frac{1}{2}\sum_j A_{ij}|T_{is}-T_{it}-T_{js}+T_{jt}|,\quad \text{for } i\neq s,t.$$

- unit current flow at nodes s and t:

$$I_s^{(st)}=1,\qquad I_t^{(st)}=1.$$

- cfb(i) (denoted as b_i) is then:

$$b_i=\frac{\sum_{s<t}I_i^{(st)}}{\frac{1}{2}n(n-1)}.$$

(takes O(m n^2) for all i) → (plus matrix inversion:) O((m+n) n^2) for everything

Random Walk Centrality == Current Flow Btw. Centrality (see [5])

- current flow at node i:

$$I_i^{(st)}=\frac{1}{2}\sum_j A_{ij}|V_i^{(st)}-V_j^{(st)}|$$
$$=\frac{1}{2}\sum_j A_{ij}|T_{is}-T_{it}-T_{js}+T_{jt}|,\quad \text{for } i\neq s,t.$$

- unit current flow at nodes s and t:

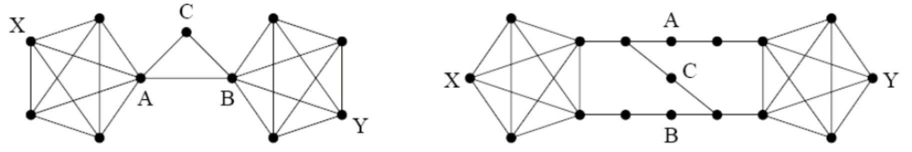
$$I_s^{(st)}=1,\qquad I_t^{(st)}=1.$$

- cfb(i) (denoted as b_i) is then:

$$b_i=\frac{\sum_{s<t}I_i^{(st)}}{\frac{1}{2}n(n-1)}.$$

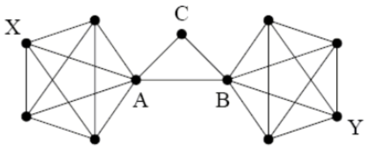
(takes O(m n^2) for all i) → (plus matrix inversion:) O((m+n) n^2) for everything

Example ([5])

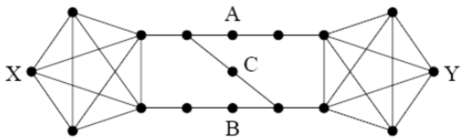


network		betweenness measure shortest-path	max-flow	random walk / current-flow
Network 1:	vertices A & B	0.636	0.631	0.670
	vertex C	0.200	0.282	0.333
	vertices X & Y	0.200	0.068	0.269
Network 2:	vertices A & B	0.265	0.269	0.321
	vertex C	0.243	0.004	0.267
	vertices X & Y	0.125	0.024	0.194

Example ([5])



Network 1



Network 2

network		betweenness measure		
		shortest-path	max-flow	random walk / current-flow
Network 1:	vertices A & B	0.636	0.631	0.670
	vertex C	0.200	0.282	0.333
	vertices X & Y	0.200	0.068	0.269
Network 2:	vertices A & B	0.265	0.269	0.321
	vertex C	0.243	0.004	0.267
	vertices X & Y	0.125	0.024	0.194