Better Approximation of Betweenness Centrality

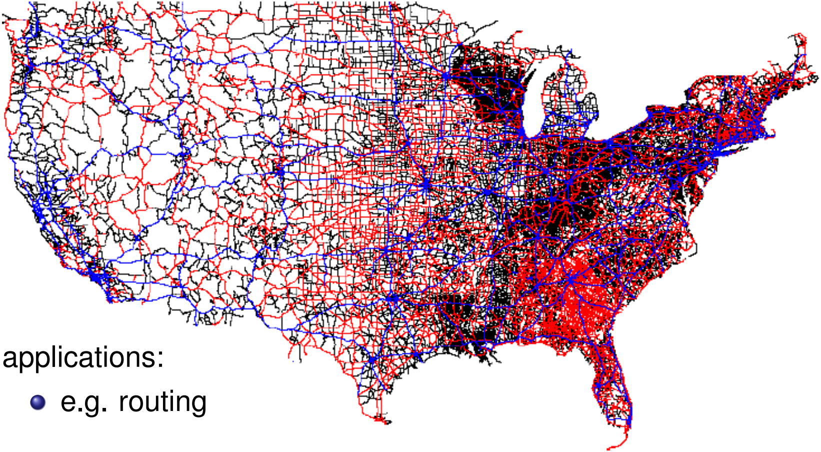
Robert Geisberger Peter Sanders Dominik Schultes

Workshop on Algorithm Engineering & Experiments, 2008

Motivation

Automatic analysis of networks requires fast computation of centrality indices.
The networks grow faster than the speed of our computers so fast approximation algorithms gain importance.

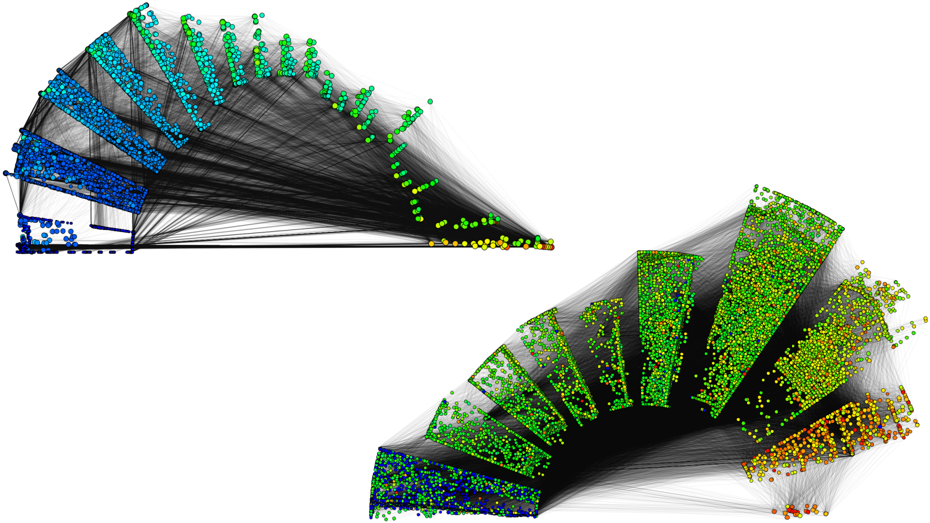
Transportation



applications:

- e.g. routing

Graph drawing



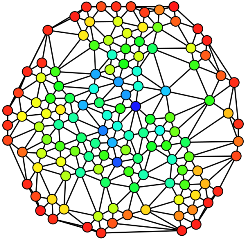
Definition Betweenness Centrality

Let

- $G = (V, E)$ be a weighted directed (multi)-graph,
- SP_{st} = set of shortest paths between source s and target t
- $SP_{st}(v)$ = set of shortest paths that have v in their interior.

Then the *betweenness centrality* for node v is

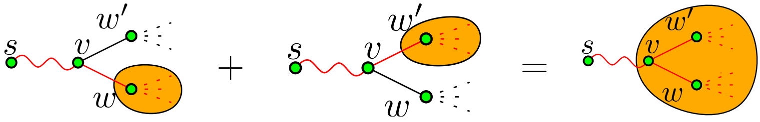
$$c(v) := \sum_{s,t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}, \text{ where } \sigma_{st} := |SP_{st}| \text{ and } \sigma_{st}(v) := |SP_{st}(v)| \quad .$$



Exact algorithm

Brandes [Brandes01] exact algorithm:

- solve single source shortest path problem (SSSP) from each node
- backward aggregation of counter values



Time requirements:

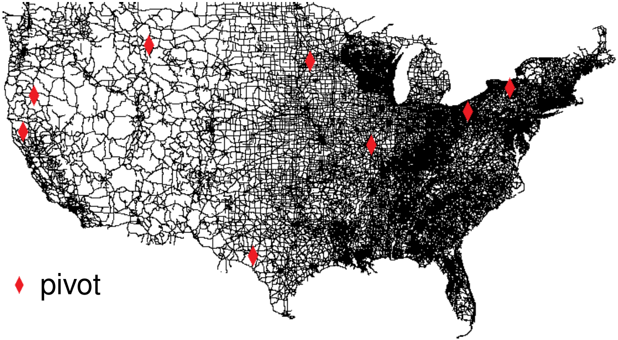
- $\Theta(nm)$ for unit distance, otherwise
- $\Theta\big(nm + n^2 \log(n)\big)$.

Approximation approach

Brandes and Pich [BrandesPich06] approximation algorithm:

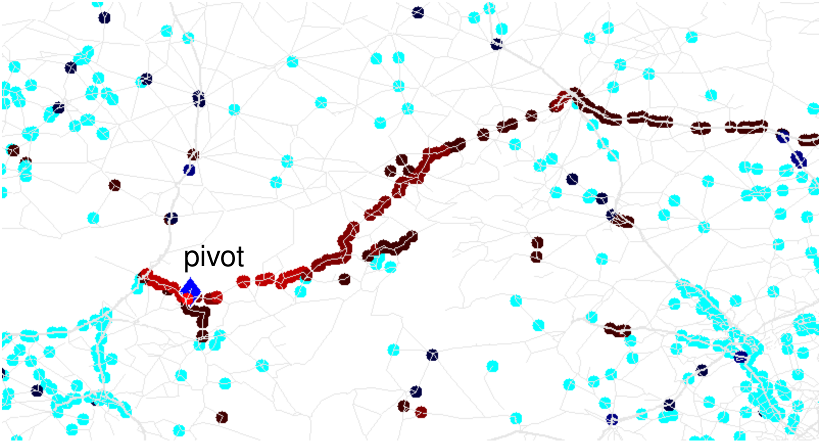
- choose subset k of starting nodes (*pivots*)
- solve only k single source shortest path problem (SSSP)
- *extrapolate* betweenness values

This yields an *unbiased* estimator for betweenness.



Deficiency of previous approach

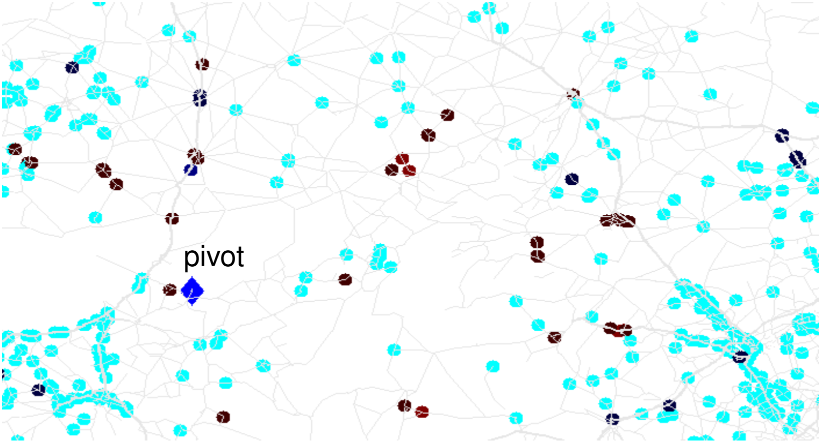
Overestimation of betweenness values of nodes near a pivot.



large overestimation   small overestimation   "false zero"

Main idea

Consider the *length* to the pivot to *scale* contributions.



● large overestimation ● small overestimation ● "false zero"

Generalized Framework

Parameters:

- *length function* ℓ on the edges
  For a path $P = \langle e_1, \ldots, e_k \rangle$ let $\ell(P) := \sum_{1 \le i \le k} \ell(e_i)$
- *scaling function* $f : [0, 1] \to [0, 1]$

Features:

- *unbiased estimator*
- focus on differences between approximation methods

Generalized Framework (continuation)

For each shortest path of the form

$$\langle \overbrace{s, \ldots, v}^{Q}, \ldots, t \rangle \quad \text{with } \underbrace{\langle s, \ldots, v, \ldots, t \rangle}_{P}$$

we define a *scaled contribution*

$$\delta_P(v) := \frac{f(\ell(Q)/\ell(P))}{\sigma_{st}}$$

Overall, v gets a contribution from a pivot *s*

$$\delta_s(v) := \sum_{t \in V} \sum \{\delta_P(v) : P \in SP_{st}(v)\}$$

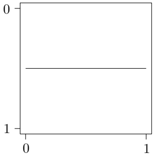
Proposed Parameters

Length function ℓ:

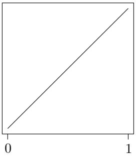
- edge weight function used for shortest-path calculation
- unit distance

Scaling function f:

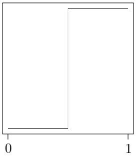
- Brandes and Pich — constant
- *linear scaling* — $f(x) = x$
- *bisection scaling* — $f(x) = \begin{cases} 0 & \text{for } x \in [0, 1/2) \\ 1 & \text{for } x \in [1/2, 1] \end{cases}$



Brandes and Pich



linear scaling



bisection scaling

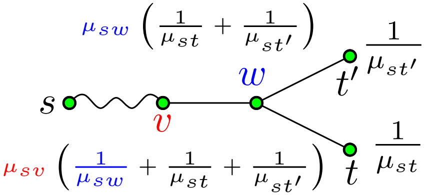
Linear Time Computation

Brandes [Brandes01]:

- compute σ_{st} on the fly during the shortest path calculation
- subsequent aggregation phase, like exact algorithm

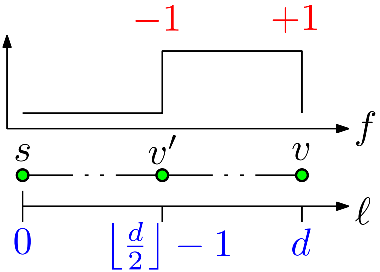
linear scaling:

- Let μ_{st} denote the shortest path distance from s to t, aggregate $1/\mu_{st}$ instead of 1, multiply with μ_{sv} at the end.



Linear Time Computation of bisection scaling

- use unit distance
- depth first traversal of shortest path DAG, keep an array storing the current path from *s*
- increment counter of current node *v* and decrement counter of middle node *v*′



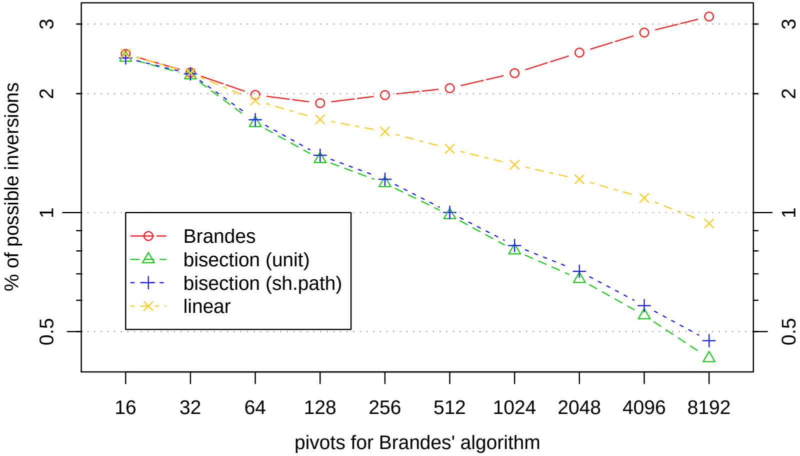
Comments:

- only efficient for $\sigma_{st} \in \{0, 1\}$
- for $\sigma_{st} \geq 2$ *sampling* of shortest paths required

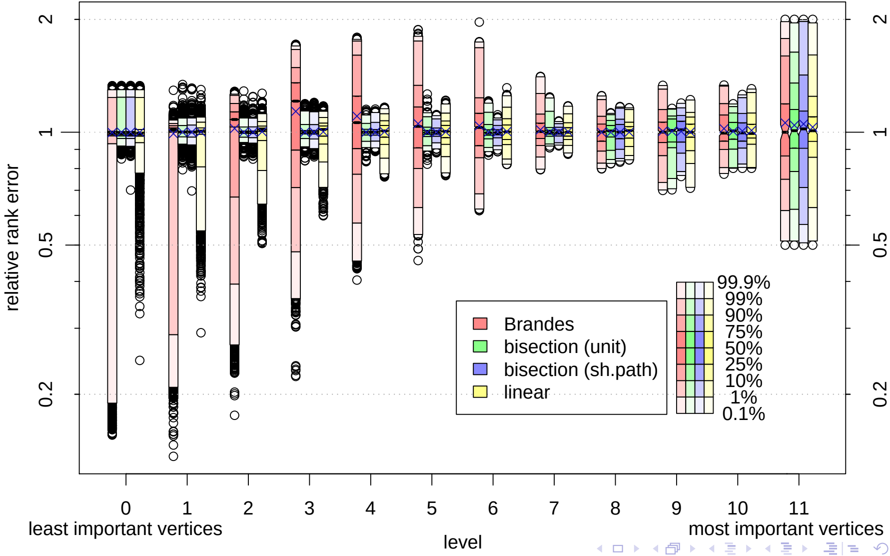
Overview of used graphs

graph	nodes	edges	source
Belgian road network	463 514	596 119	PTV AG
Belgian road network (unit dist.)	463 514	596 119	PTV AG
Actor co-starring network	392 400	16 557 451	[NotreD]
US patent network	3 774 769	16 518 947	[NBER]
World-Wide-Web graph	325 729	1 497 135	[NotreD]
CNR 2000 Webgraph	325 557	3 216 152	[LabWA]
CiteSeer undir. citation network	268 495	2 313 294	[Citeseer]
CiteSeer co-authorship network	227 320	1 628 268	[Citeseer]
CiteSeer co-paper network	434 102	32 073 440	[Citeseer]
DBLP co-authorship network	299 067	1 955 352	[DBLP]
DBLP co-paper network	540 486	30 491 458	[DBLP]

Belgium road network



Belgium road network

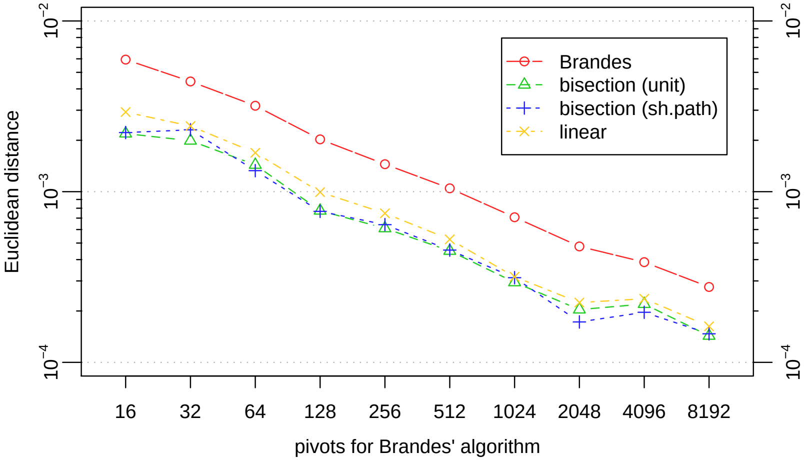


Summary

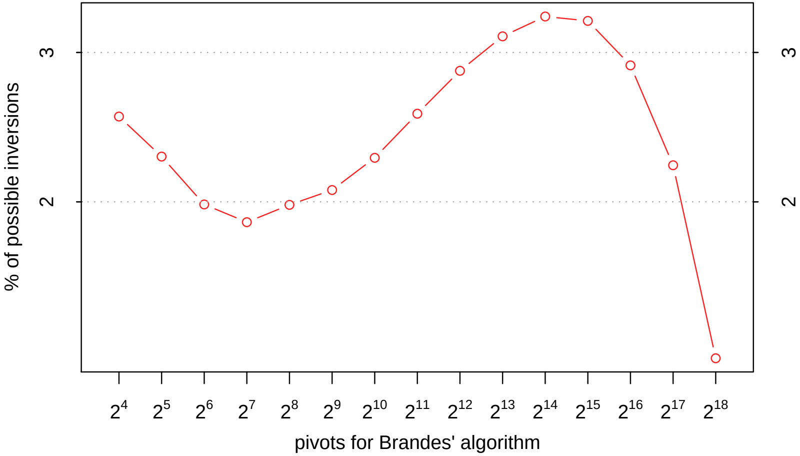
- The bisection scaling algorithm achieves the best results.

- Future work
  - efficient exact bisection scaling algorithm for $\sigma_{st} \geq 2$
  - local searches to eliminate "false zeros"

Belgian road network



Belgian road network (Brandes and Pich)



Additional networks