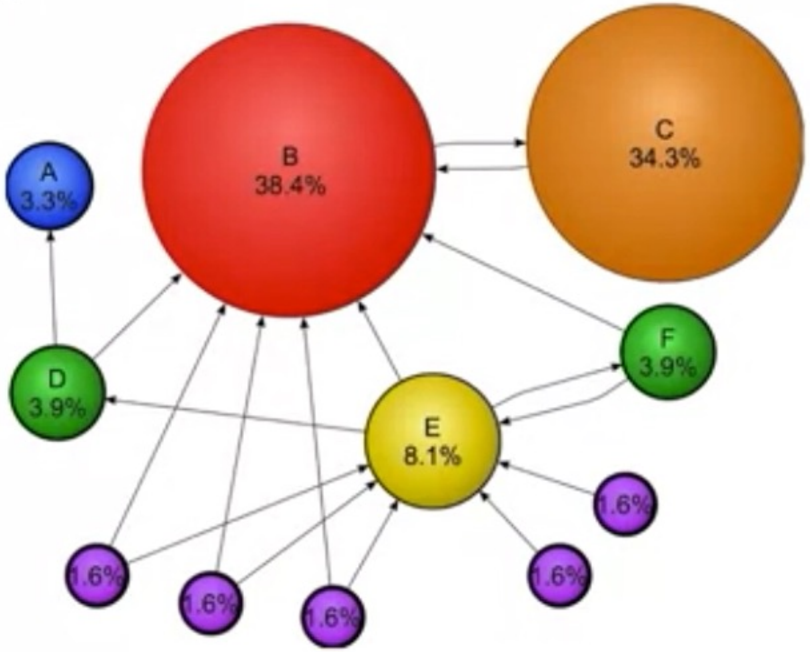
PageRank

Developed by Google founders to measure the importance of webpages from the hyperlink network structure.

PageRank assigns a score of importance to each node. Important nodes are those with many in-links from important pages.

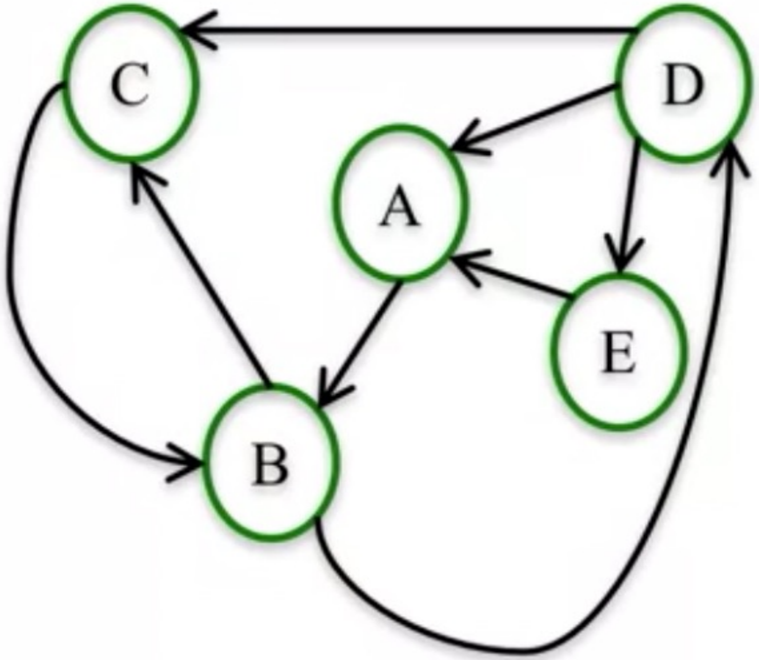
PageRank can be used for any type of network, but it is mainly useful for directed networks.



A node's PageRank depends on the PageRank of other nodes (Circular definition?).

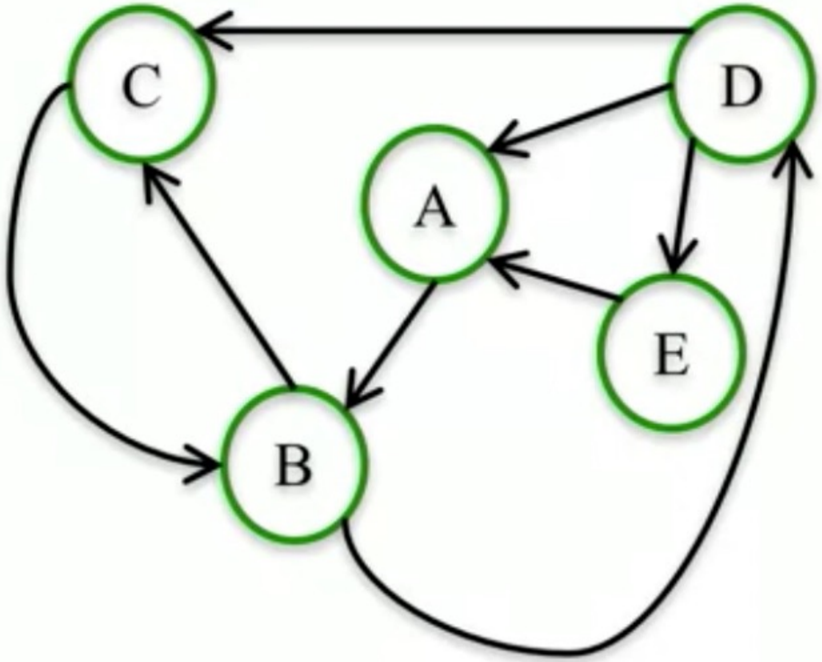
PageRank

Who should be the most "important" node in this network?



PageRank – Step 1

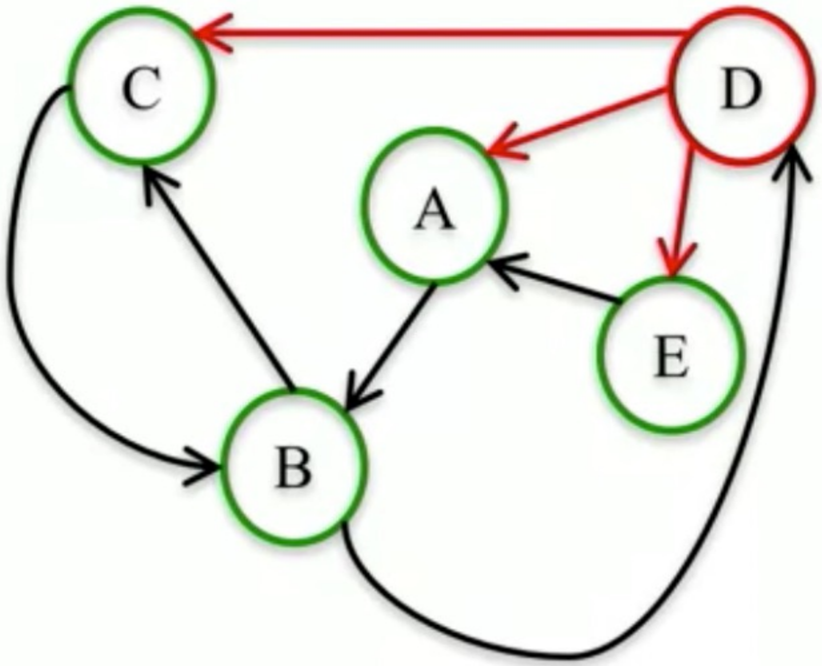
Page Rank (k = 1)					
	A	B	C	D	E
Old	1/5	1/5	1/5	1/5	1/5
New	4/15	2/5	1/6	1/10	1/15



A: (1/3)*(1/5) + 1/5 = 4/15
B: 1/5 + 1/5 = 2/5
C: (1/3)*(1/5) + (1/2)*(1/5) = 5/30 = 1/6
D: (1/2)*(1/5) = 1/10
E: (1/3)*(1/5) = 1/15

PageRank – Step 2

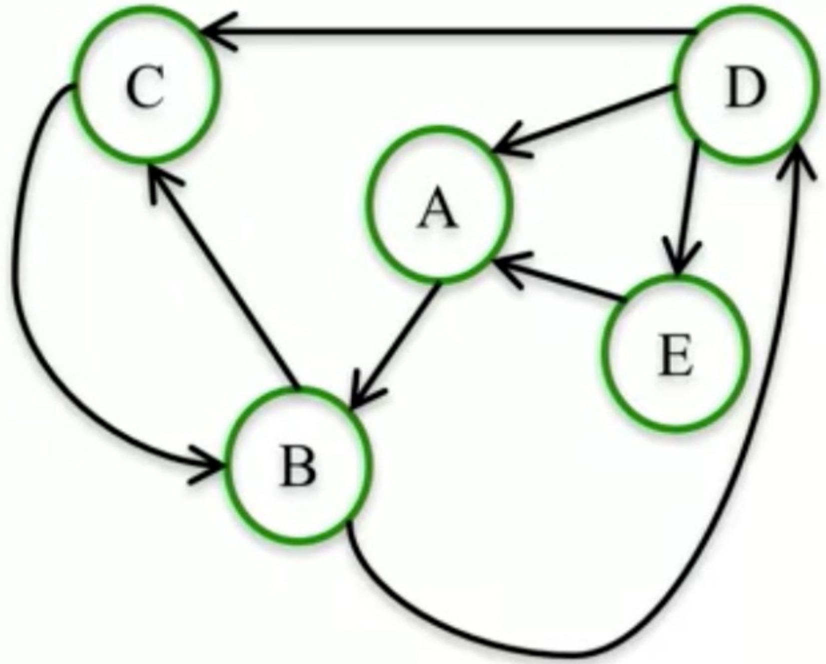
Page Rank (k = 2)					
	A	B	C	D	E
Old	4/15	2/5	1/6	1/10	1/15
New	1/10	13/30	7/30	2/10	



A: (1/3)*(1/10) + 1/15 = 1/10
B: 1/6 + 4/15 = 13/30
C: (1/3)*(1/10) + (1/2)*(2/5) = 7/30
D: (1/2)*(2/5) = 2/10
E: (1/3)*(1/10)

PageRank

	Page Rank				
	A	B	C	D	E
k=2	1/10	13/30	7/30	2/10	1/30
k=2	.1	.43	.23	.20	.03
k=3	.1	.33	.28	.22	.06
k=∞	.12	.38	.25	.19	.06



What if continue with k = 4,5,6,...?

Summary

Steps of Basic PageRank:

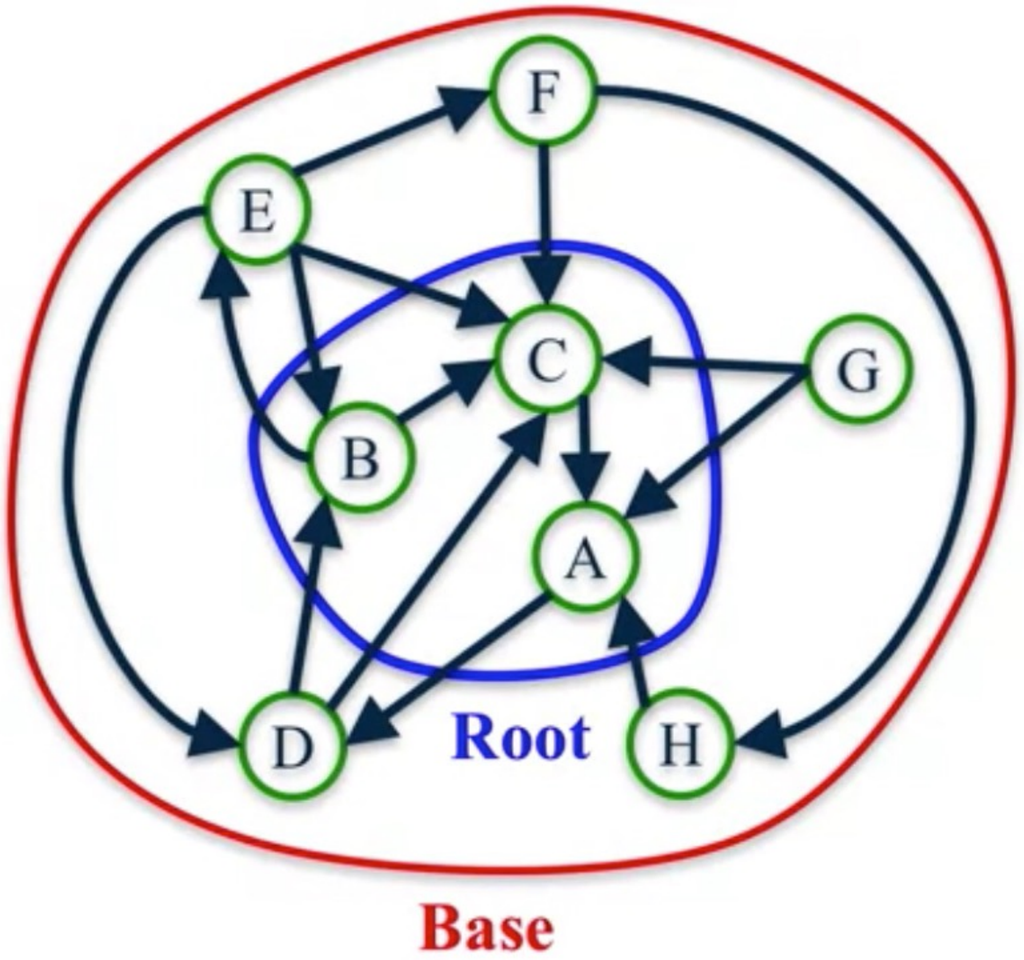
1. All nodes start with PageRank of $1/n$
2. Perform the *Basic PageRank Update Rule k* times:
   - **Basic PageRank Update Rule**: Each node gives an equal share of its current PageRank to all the nodes it links to.
   - The new PageRank of each node is the sum of all the PageRank it received from other nodes.

For most networks, PageRank values converge as k gets larger ($k \to \infty$)

Hubs and Authorities

Given a query to a search engine:

- **Root**: set of highly relevant web pages (e.g. pages that contain the query string) – potential *authorities*.
- Find all pages that link to a page in root – potential *hubs*.
- **Base**: root nodes and any node that links to a node in root.
- Consider all edges connecting nodes in the base set.



- Authority

$$x_i = \alpha \sum_j A_{ij} y_j$$

- Hub

$$y_i = \beta \sum_j A_{ji} x_j$$

- In matrix form

$$x = \alpha A y, \quad y = \beta A^T x$$

$$\begin{aligned} & \beta A^T x = \alpha\beta A^T A y, \quad \alpha A y = \alpha\beta A A^T x \\ \Rightarrow \quad & y = \alpha\beta A^T A y, \quad x = \alpha\beta A A^T x \\ \Rightarrow \quad & A^T A y = (\alpha\beta)^{-1} y \quad A A^T x = (\alpha\beta)^{-1} x \end{aligned}$$

- $A^T A$: the **cocitation matix**: authority centrality x is an eigenvector centrality of the cocitation network.

- $A^T A$: the **bibliographic coupling matrix**: Hub centrality y is an eigenvector centrality of bibliographic network.

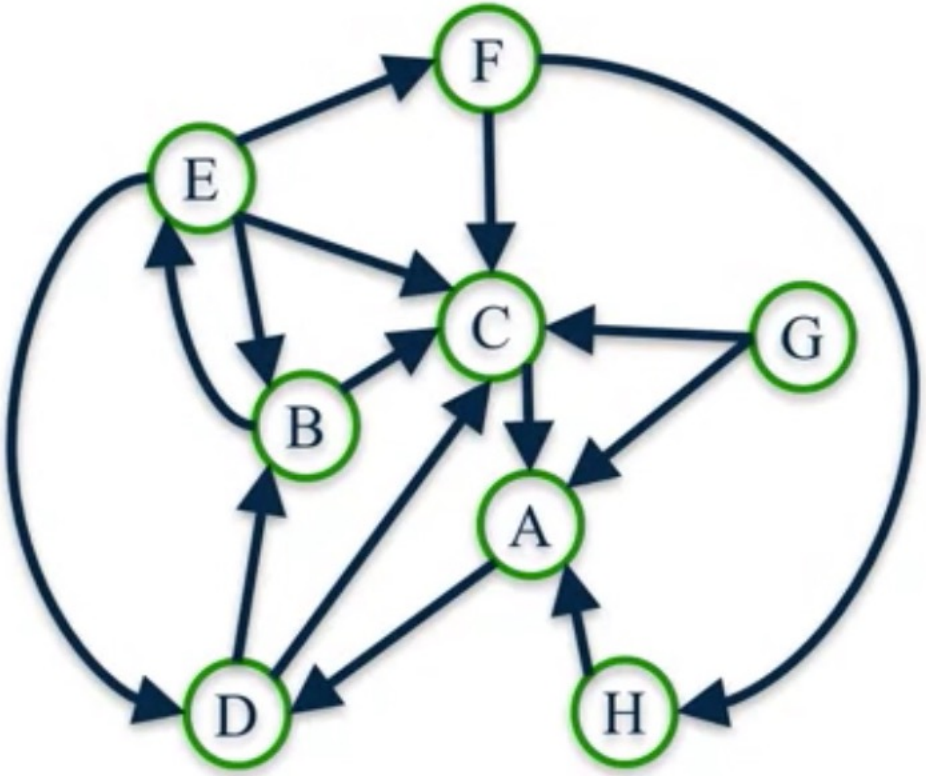
HITS Algorithm

Computing k iterations of the HITS algorithm to assign an *authority score* and *hub score* to each node.

1. Assign each node an authority and hub score of 1.
2. Apply the ***Authority Update Rule***: each node's *authority* score is the sum of *hub* scores of each node that *points to it*.
3. Apply the ***Hub Update Rule***: each node's hub score is the sum of authority scores of each node that *it points to*.
4. **Nomalize** Authority and Hub scores: $\text{auth}(j) = \frac{\text{auth}(j)}{\sum_{i \in N} \text{auth}(i)}$

HITS Algorithm Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3	1
B	1	1	2	2
C	1	1	5	1
D	1	1	2	2
E	1	1	1	4
F	1	1	1	2
G	1	1	0	2
H	1	1	1	1

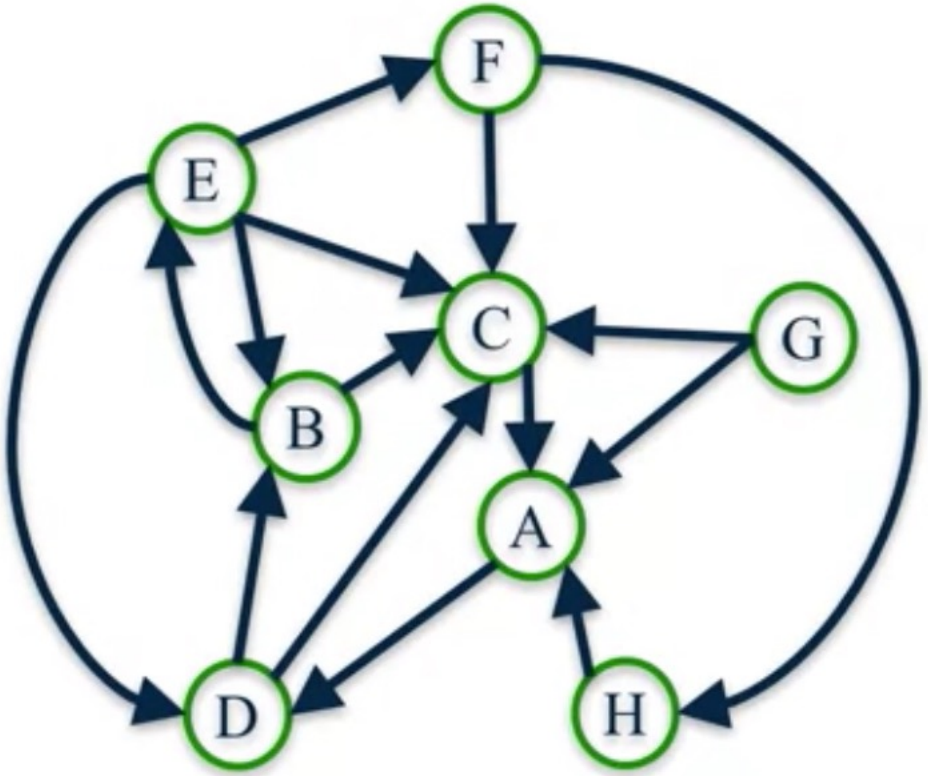


Normalize:

$\sum_{i \in N} \text{auth}(i) = 15 \quad \sum_{i \in N} \text{hub}(i) = 15$

HITS Algorithm Example

	Old Auth	Old Hub	New Auth	New Hub
A	1	1	3/15	1/15
B	1	1	2/15	2/15
C	1	1	5/15	1/15
D	1	1	2/15	2/15
E	1	1	1/15	4/15
F	1	1	1/15	2/15
G	1	1	0/15	2/15
H	1	1	1/15	1/15

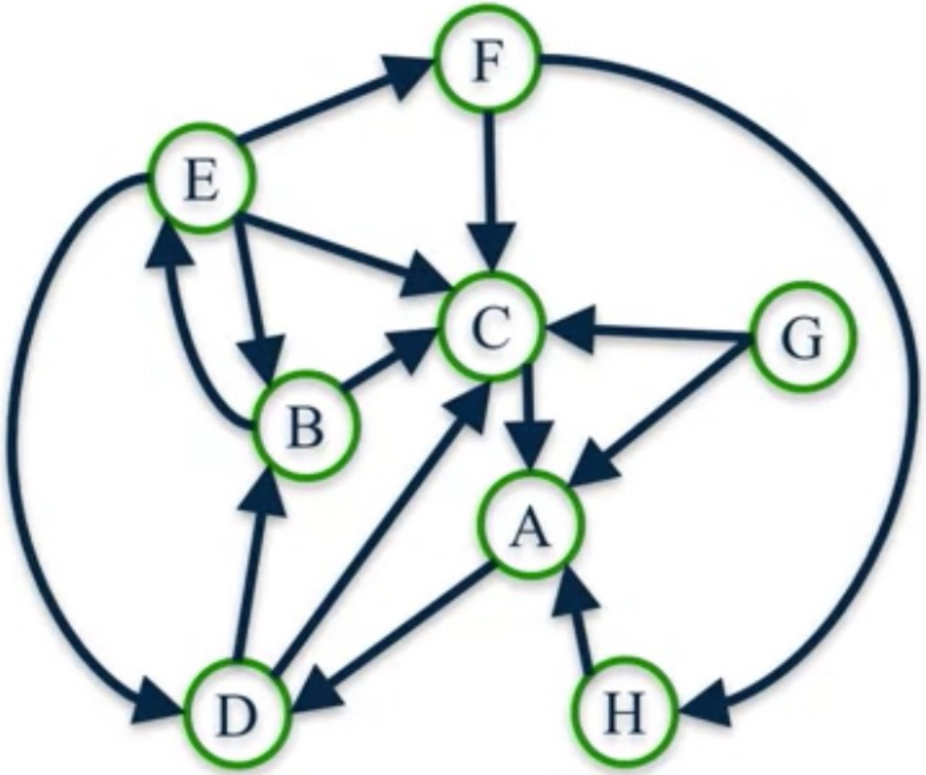


Normalize:

$\sum_{i \in N} \text{auth}(i) = 15 \quad \sum_{i \in N} \text{hub}(i) = 15$

HITS Algorithm Example

	Old Auth	Old Hub	New Auth	New Hub
A	1/5	1/15	4/15	2/15
B	2/15	2/15	6/15	2/5
C	1/3	1/15	12/15	1/5
D	2/15	2/15	1/3	7/15
E	1/15	4/15	2/15	2/3
F	1/15	2/15	4/15	2/5
G	0	2/15	0	8/15
H	1/15	1/15	2/15	1/5

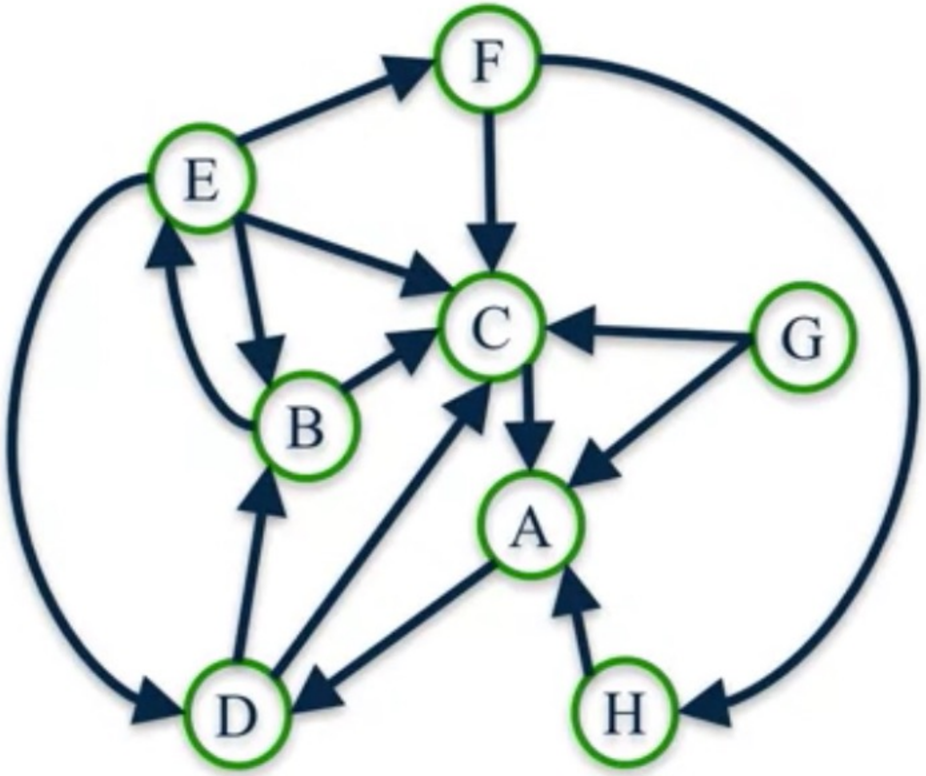


Normalize:

$$\sum_{i \in N} \text{auth}(i) = {}^{35}/_{15}$$

HITS Algorithm Example

	Old Auth	Old Hub	New Auth	New Hub
A	1/5	1/15	4/35	2/15
B	2/15	2/15	6/35	2/5
C	1/3	1/15	12/35	1/5
D	2/15	2/15	1/7	7/15
E	1/15	4/15	2/35	2/3
F	1/15	2/15	4/35	2/5
G	0	2/15	0	8/15
H	1/15	1/15	2/35	1/5

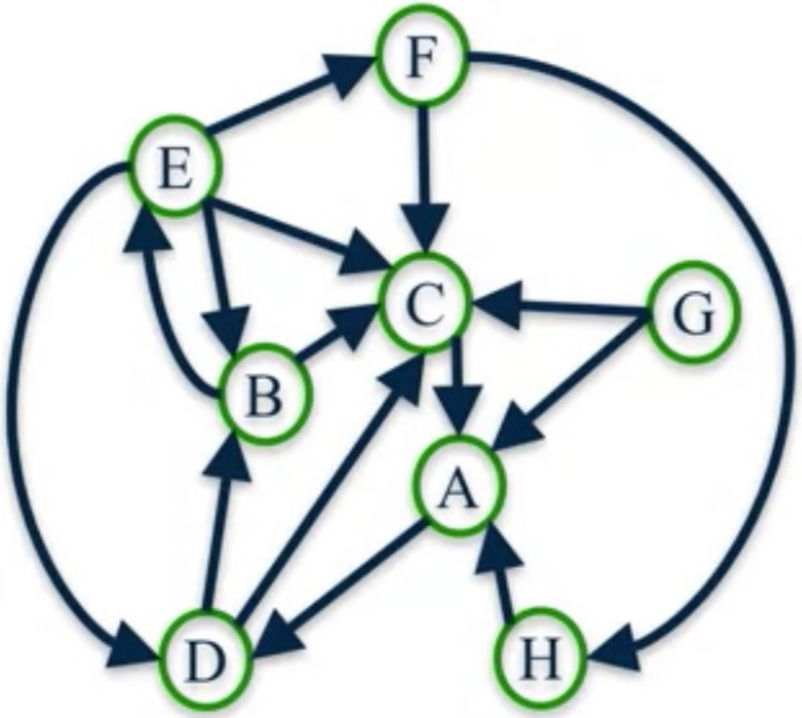


Normalize:

$$\sum_{i \in N} \text{hub}(i) = {}^{45}/_{15} = 3$$

HITS Algorithm Convergence

	k	A	B	C	D	E	F	G	H
Auth	**2**	.11	.17	.34	.14	.06	.11	0	.06
	4	.10	.18	.36	.13	.06	.11	0	.06
	6	.09	.19	.37	.13	.06	.11	0	.06
Hub	**2**	.04	.13	.07	.16	.22	.13	.18	.07
	4	.04	.14	.05	.18	.25	.14	.17	.04
	6	.04	.14	.04	.18	.26	.14	.16	.04

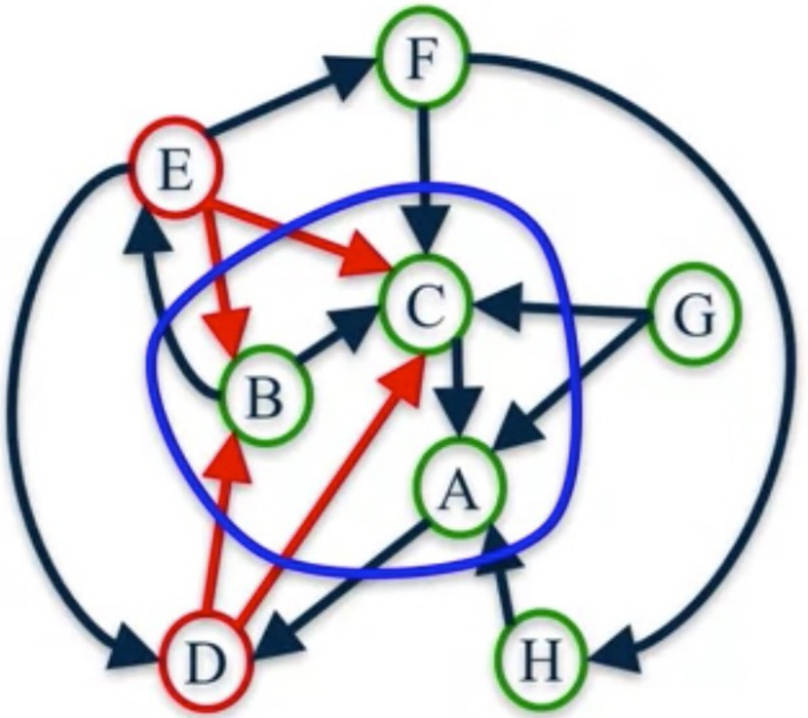


HITS Algorithm Convergence

For most networks, as k gets larger, authority and hub scores converge to a unique value.

As $k \longrightarrow \infty$ the hub and authority scores approach:

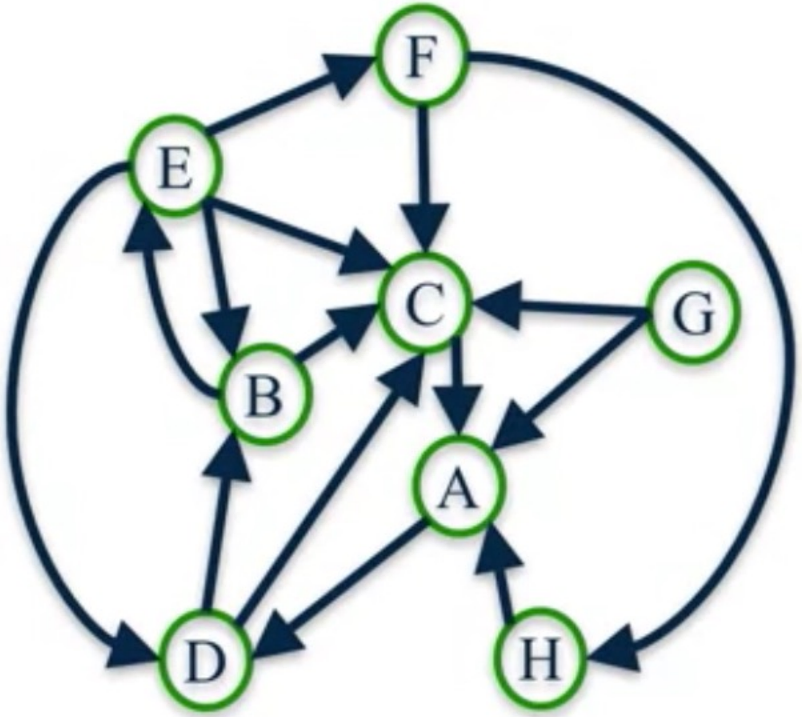
	A	B	C	D	E	F	G	H
Auth	.08	**.19**	**.40**	.13	.06	.11	0	.06
Hub	.04	.14	.03	**.19**	**.27**	.14	.15	.03



HITS Algorithm NetworkX

You can use NetworkX funtion hits(G) to compute the hub and authority scores of network G.

hits(G) outputs two dictionaries, keyed by node, with the hub and authority scores of the nodes.



6.1. Closeness Centrality

- Mean distance from vertex to other vertices
- Suppose d_{ij} is the length of geodesic path from i to j (number of edge from i to j in shortest path). Then

$$
\begin{aligned}
l_i &= \frac{1}{n}\sum_j d_{ij} \; : \text{ the geodesic distance from } i \\
&= \frac{1}{n-1}\sum_{j\neq i} d_{ij} \text{ (exclude } j = i)
\end{aligned}
$$

- Closeness centrality $C_i = \dfrac{1}{l_i}$.
- Problem: Difficult to distinguish between central and less central in some network.
- Redefine: $C_i' = \dfrac{1}{n-1}\sum_{j\neq i}\dfrac{1}{d_{ij}}$.

6.2. Betweenness Centrality

- Measure a vertex lies on paths between others.

$$\text{Let } n^i_{st} = \begin{cases} 1 \text{ if a vertex } i \text{ lies on the geodesic path from } s \text{ to } t \\ 0 \text{ otherwise} \end{cases}$$

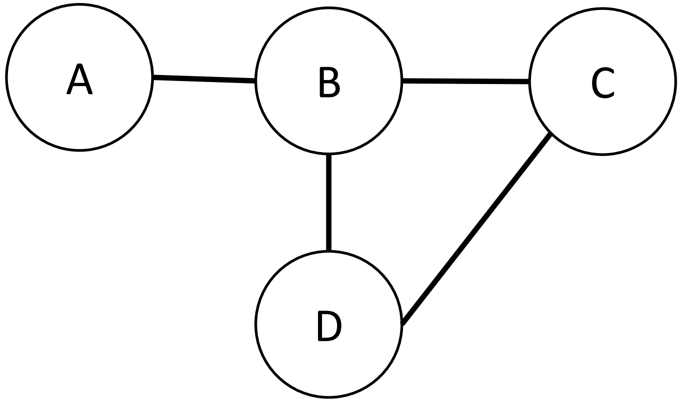
- Betweenness centrality X_i is given by

$$X_i = \sum_{s,t} \frac{n^i_{st}}{g_{st}}$$

where g_{st} = total number of geodesic path from s to t

6.2. Betweenness Centrality

Example



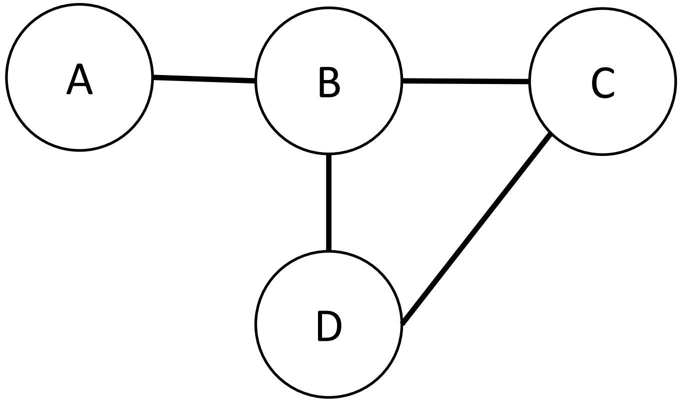
$$X_B = ?$$

$$X_B = \sum_{s,t} \frac{n_{st}^B}{g_{st}}$$

- Endpoint: We either include or exclude node i as nodes s & t in g_{st}

6.2. Betweenness Centrality

Example



$$X_B = ?$$

1. Exclude node B

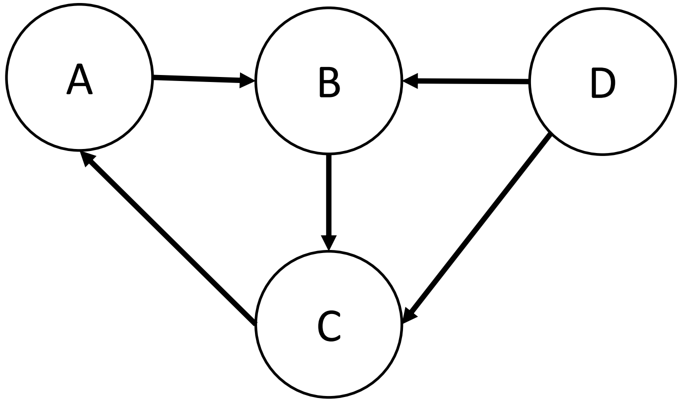
$$X_B = \frac{n^B_{AD}}{g_{AD}} + \frac{n^B_{AC}}{g_{AC}} + \frac{n^B_{CD}}{g_{CD}}$$
$$= \frac{1}{1} + \frac{1}{1} + \frac{0}{1} = 2$$

2. Include node B

$$X_B = \frac{n^B_{AB}}{g_{AB}} + \frac{n^B_{AC}}{g_{AC}} + \frac{n^B_{AD}}{g_{AD}} + \frac{n^B_{BC}}{g_{BC}} + \frac{n^B_{BD}}{g_{BD}} + \frac{n^B_{CD}}{g_{CD}}$$
$$= \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{1}{1} + \frac{0}{1} = 5$$

6.2. Betweenness Centrality

Example



$$X_C = ?$$

If we exclude node C as endpoint:

$$X_C = \frac{n_{AB}^C}{g_{AB}} + \frac{n_{BA}^C}{g_{BA}} + \frac{n_{DB}^C}{g_{DB}} + \frac{n_{BD}^C}{g_{BD}} + \frac{n_{AD}^C}{g_{AD}} + \frac{n_{DA}^C}{g_{DA}}$$

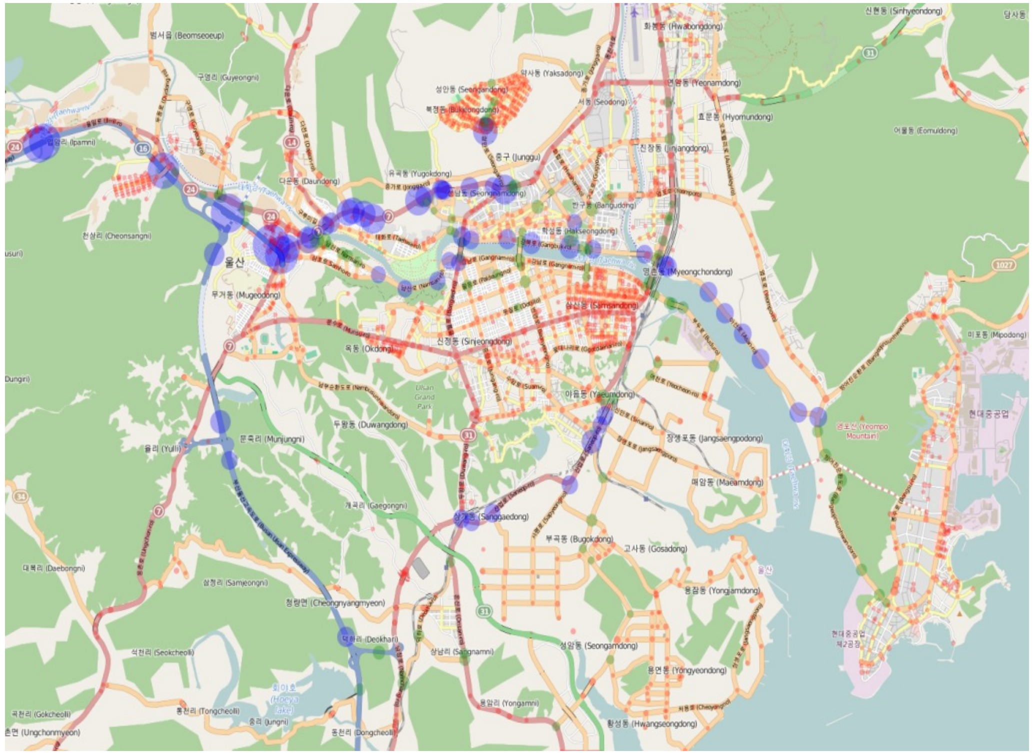
$$= \frac{0}{1} + \frac{1}{1} + \frac{0}{1} + 0 + 0 + \frac{1}{1} = 2 \; (g_{st} = 0 \implies \frac{n_{st}^i}{g_{st}} = 0)$$

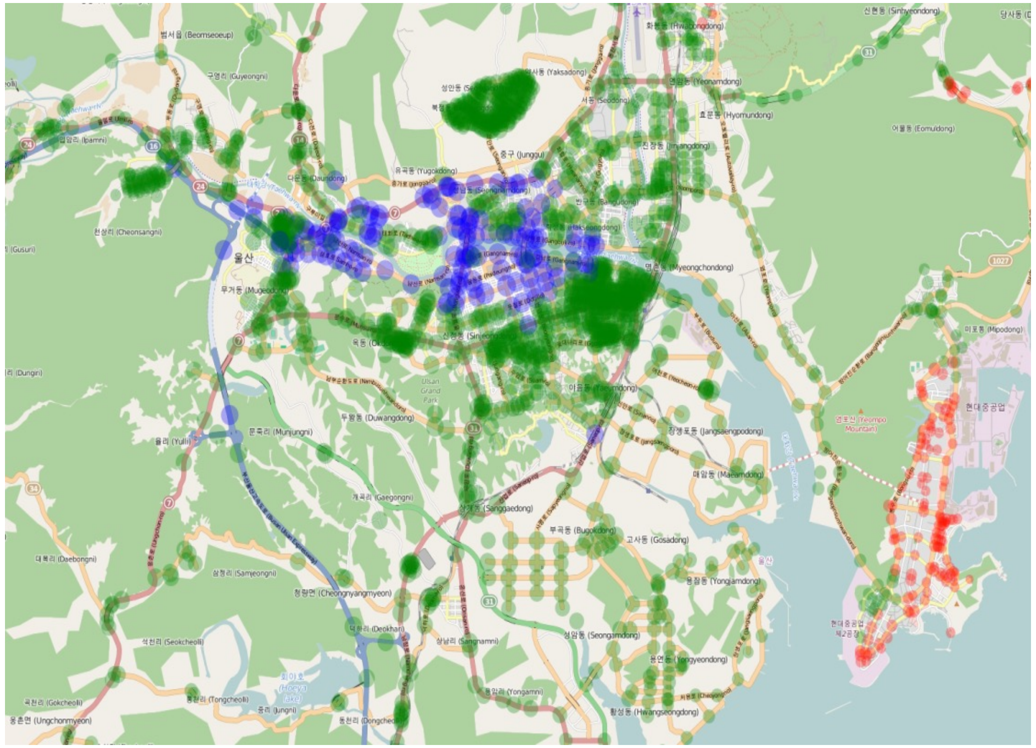
6.2. Betweenness Centrality

- Normalization: $X_i = \frac{1}{n^2} \sum_{s,t} \frac{n_{st}^i}{g_{st}}$ or

$$X_i = \begin{cases} \frac{1}{\binom{n-1}{2}} \sum_{s,t} \frac{n_{st}^i}{g_{st}} \text{ (undirected)} \\ \frac{1}{(n-1)(n-2)} \sum_{s,t} \frac{n_{st}^i}{g_{st}} \text{ (directed)} \end{cases}$$

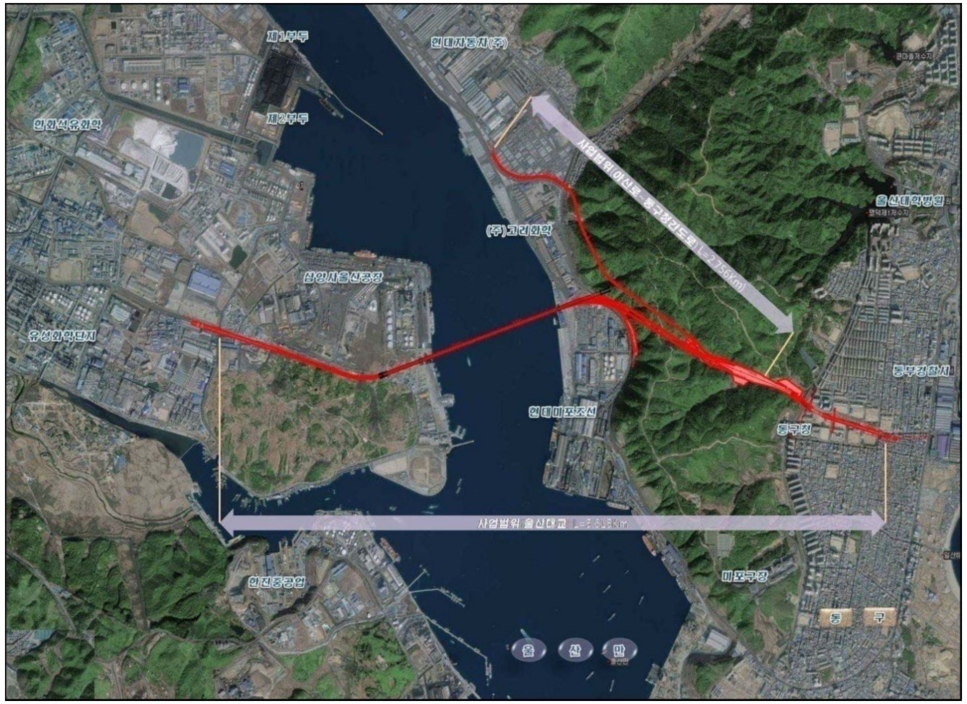
Centrality Analysis





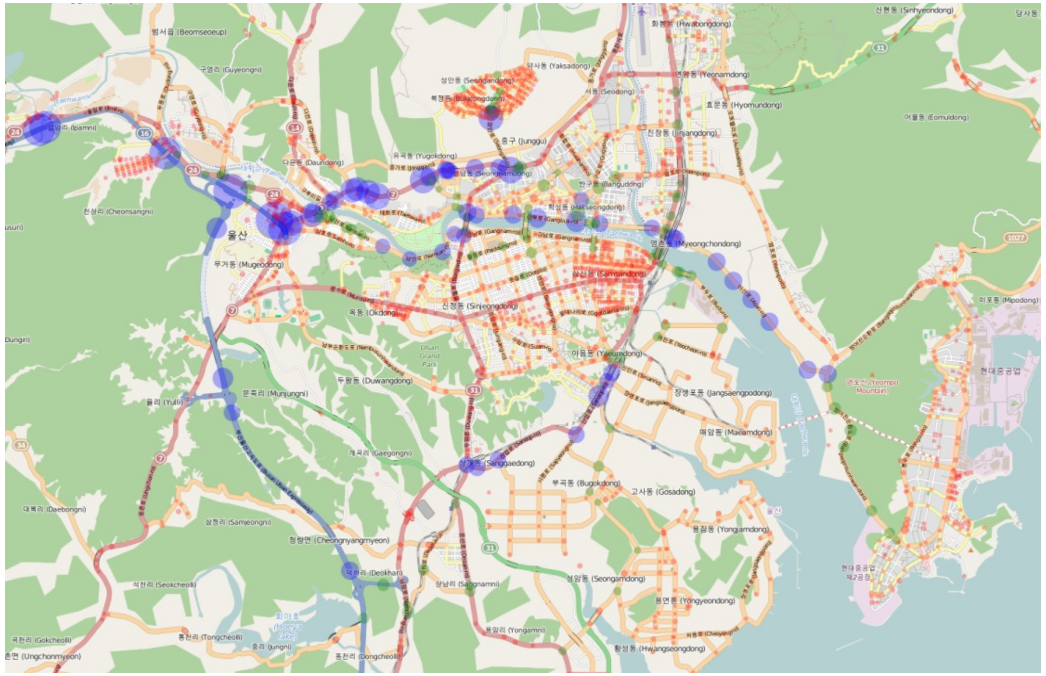
- BC: Line connecting nodes from a large BC to a small BC
  -> Ulsan Transportation Backbone
- CC: Area with large values of CC
  -> Center of Ulsan & Transportation benefit

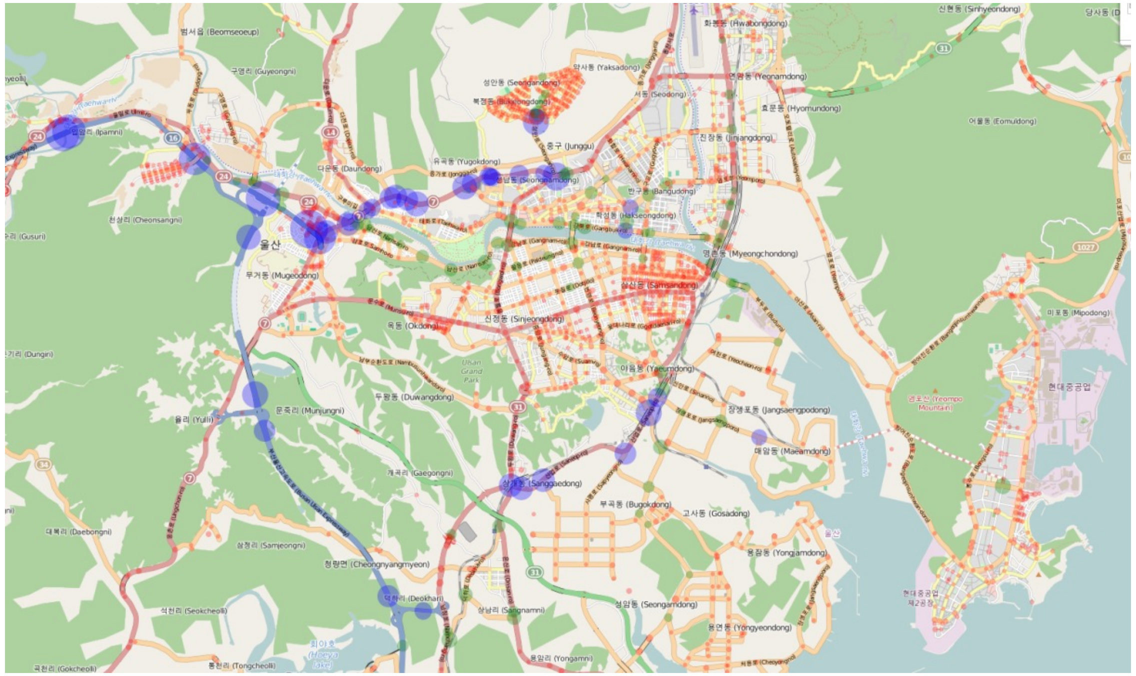
Ulsan Bridge



기간: 2010 - 2015년 5월 완공 예정
단경간 현수교(주탑과 주탑 사이의 거리인 경간이 하나로 연결된 현수교)
북구 염포동과 동구 화정동을 연결하는 다리
예산: 5398 억원/ 8.38 km(교량: 1.15km)

BC Centrality Analysis B/A





- **Change of transportation backbone**