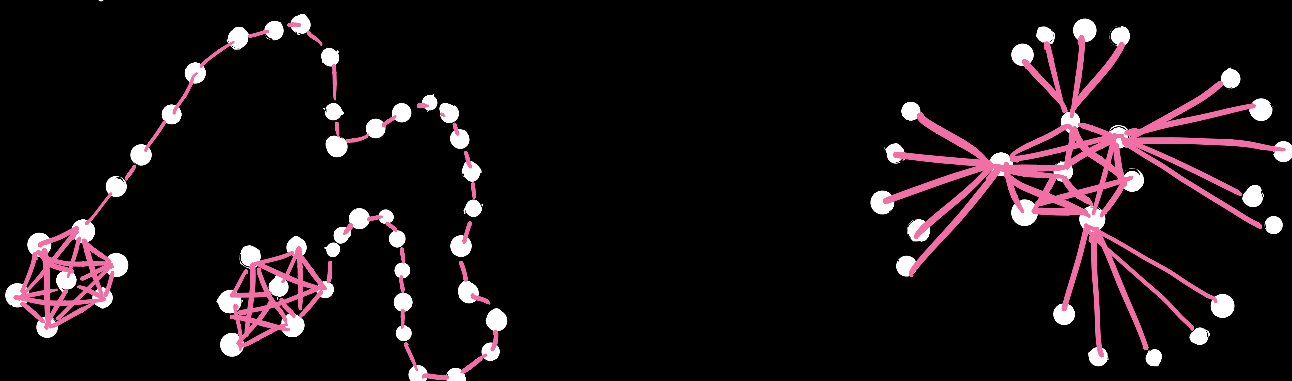
GraphT 6

1) Paths, diffusion, and navigation

- **1.1)** path lengths
  - ↓ shortest path length (Dijkstra's algo)
    - local efficiency
    - nodal efficiency
    - global efficiency
- **1.2)** diffusion processes on graphs → communicability
  - ↓ diffusion efficiency
    - search information
    - path transivity

> "* How the topological properties of graphs constrain the way in which information flows through the graph?
>
> * which specific graph properties can be used to gain insight into information processing by graph units (nodes / edges)?"

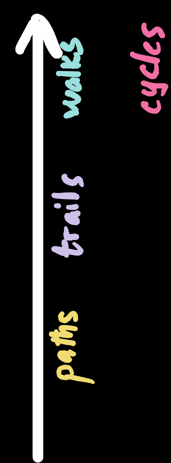
Example



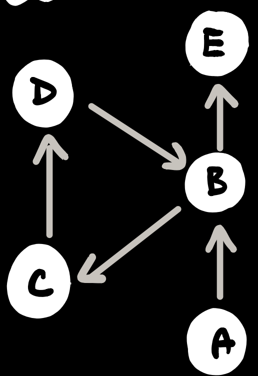
information flow Efficiency measures

walks, trails, paths, and cycles

- A walk is a sequence of edges that an entity can "walk" along, in a continuous manner, without having to ever "jump" from one node to another node and thereby break the contiguity of the sequence.
  A walk can visit the same node(s) or edge(s) more than once.
- A trail is a walk where all edges are unique.
- A path is a trail in which all edges and nodes are unique.
- A cycle is a closed walk where first and last nodes are the same.
  ↳ can be embedded within a larger walk.

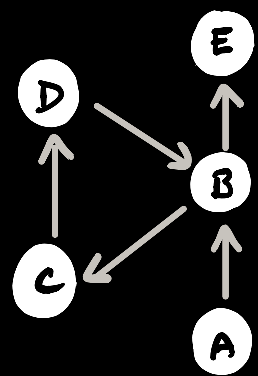


Example:

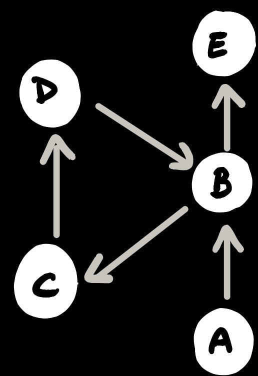


{A-B-C-D-B-C-D-B-E}

{B-C-D-B}



{A-B-C-D-B-E}



{A-B-E}

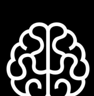
Remark

In some of the literature, the term "path" is used as a generic description of trails, walks, paths, and cycles.
In this case, a simple path means a path in which all nodes and edges are unique.

Shortest path routing

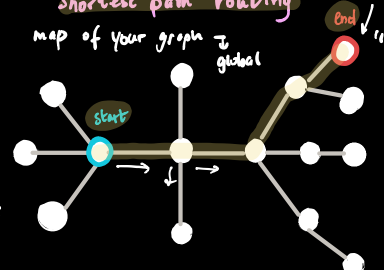
Diffusion of information flow

Shortest paths are fundamental to graph theory.
Many works focused on shortest paths as the principle routes along which information is communicated in a graph.



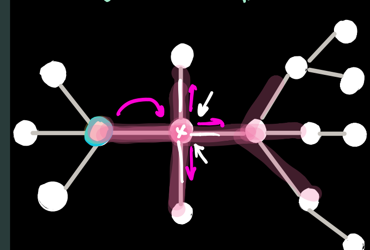
What about our brains?
↳ whether or not neural information is routed via the shortest paths of a nervous system nonetheless remains an unresolved question in neuroscience.

Shortest path routing



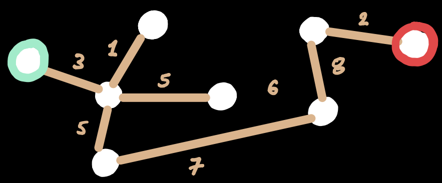
Rule: information flows along a single path to a single destination

Information diffusion



Rule: information flow is dispersive
→ propagates simultaneously on multiple "fronts" without being directed to a single destination.

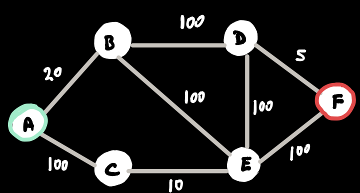
> " What exactly do we mean by a shortest path ?
> – Beware! Once you give it a definition, you build on it and act on it –



Definition

A shortest path, a geodesic, in a graph is an ordered set of edges linking two nodes in a graph for which the sum of the weights of its edges is minimal.

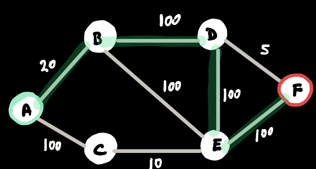
> " What if the edges with the highest weights mediate the strongest and most efficient flow of information in a graph ? "



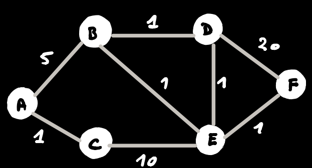
longest path ?

Idea → remap the longest path to the shortest path.
↳ similarity-to-distance remapping to the edge weights before computing the shortest paths.

$w_{ij} \leftarrow 1/w_{ij}$



Longest path = {A, B, D, E, F}
$\ell = 320$



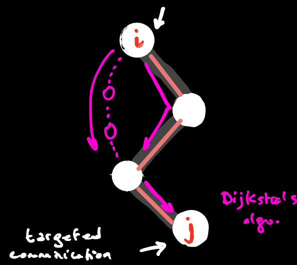
Shortest path = {A, , , F}
$\ell =$?

There is no theoretical justification for a remapping based on the reciprocal of the edge weights ($1/w_{ij}$), so we might consider other remapping functions.

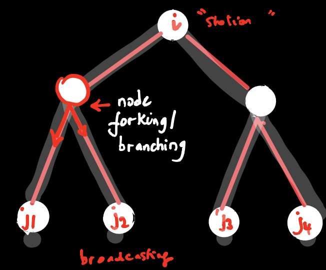
E.g., ① $w_{ij} \leftarrow 1/w_{ij}^{\beta}$, $\boxed{\beta}>0$
↓
Larger values of β accentuate the confidence we have in edges with ? weights representing ? connections.

② if $0 < w_{ij} < 1$

$$\begin{cases} w_{ij} \leftarrow -\log w_{ij} \\ w_{ij} \leftarrow 1 - w_{ij} \end{cases}$$



Under a routing model, information is routed along a single shortest path from node i to a single destination j.
→ sending a message

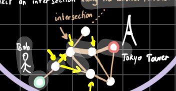


Under the diffusion model, information sent from a single node can reach any of a number of destinations via any of a number of paths.
→ broadcasting a message.

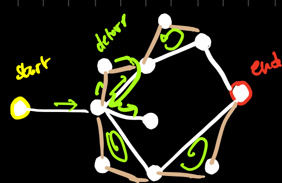
Analysis of diffusion on graphs

Scenario 2

A walker explores the roads without any road map.
E.g., Bob randomly chooses a direction whenever he arrives at an intersection.
☹ But soon he realizes that doing so has him walking in circles.
🙂 He biases his random choices so he is more likely to exit an intersection along the busiest/widest road.



{ Bob has no clear target
{ There is no apparent order to his journey.



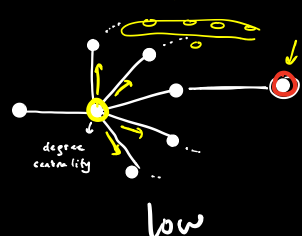
Analyse the behavior of a random walker on a graph.
↓ " Efficiency "

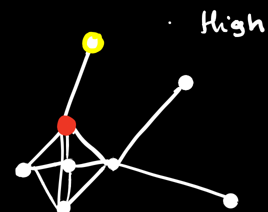
> What is the probability of the random walker reaching a specific destination by the shortest path ?
> the same one used for a deterministic walker who has a roadmap.

probability is high
🙂 shortest path is easily accessible
🙂 the efficiency of both random & deterministic walkers' paths will be similar.

probability is low
☹ shortest path is hidden
☹ the deterministic walker is more efficient than the random walker.

Sketch two graphs illustrating these two scenarios.





→ Diffusion efficiency is related to how easy to find the shortest path via a random walk.

how to quantify this ?

① Search information ② path transitivity

FINDING SHORTEST PATHS

- One of the classicals in finding shortest paths.

  Dijkstra conceived his algorithm in 1956, apparently while he was shopping with his fiancée and thinking about the shortest way to travel from Rotterdam to Groningen.

  - He published his algorithm in a two-page note 3 years later (Dijkstra, 1959).
  - The paper is in the top 1% of most cited papers ever published across all fields of sciences.

"Is it the first algorithm to compute shortest path lengths between two nodes in a graph?"

No! Matrix methods were developed in the early 1950s by Shimbel (1953) and Luce (1950) for binary graphs.

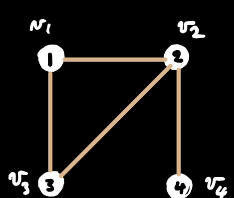
Matrix methods

Given the binary adjacency matrix A of a graph, then A^k characterizes all walks in a network that traverse exactly k edges (hops).

- For a graph with n nodes and with binary edges, the entries $n \times n$ adjacency matrix are defined by:

$$A = \begin{cases} A_{ii} = 0 \\ A_{ij} = 1 & \text{if there is an edge } e_{ij} \\ A_{ij} = 0 & \text{if there is no edge} \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

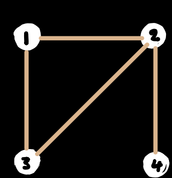


- The number of walks of length 2 between two nodes:

$$N_2(v_i, v_j) = A^2(i,j)$$

E.g.,

$$A^2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 3 & 1 & 0 \\ 1 & 1 & 2 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$



$N_2(v_2, v_3) = 1$; $N_2(v_2, v_2) = 3$

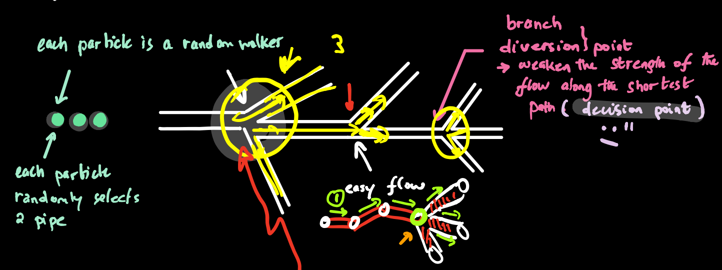
Theorem The number of walks of length k joining any two nodes v_i and v_j of a binary graph is given by the (i,j) entry of the matrix A^k:

$$N_k(v_i, v_j) = A^k(i,j)$$

Proof. This is easily proved by induction on k, as you are invited to do. □

Search information

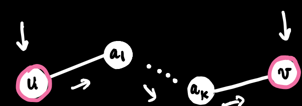
Imagine how a fluid flows through a network of pipes.



- Search information is the extent of the dispersion at the branch points.

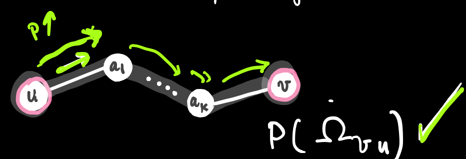
→ Paths associated with high search information are characterized by many branch points.

- Search information (Sneppen et al. 2005, Trusina et al. 2005) measures the $\boxed{\text{difficulty}}$ encountered by a random walker in identifying the shortest path between pairs of nodes as a matter of chance.



* Consider an ordered sequence $\Omega_{vu} = \{u, a_1, a_2, \dots, a_K, v\}$ that a walker traverses to go from node u to node v.

* The Search information S associated with Ω_{vu} is determined by the probability $P(\Omega_{vu})$ that a random walker beginning at node u reaches node v via the ordered sequence of nodes Ω_{vu}.



$$\Omega_{vu} = \{u, a_1, \dots, a_K, v\}$$

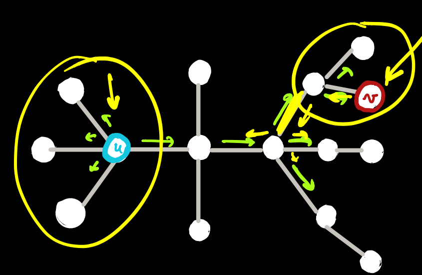
$$\rightarrow S(\Omega_{vu}) = -\log_2(P(\Omega_{vu}))$$

$[0-1]$

logarithmic mapping to ensure that the search information is high when the probability of a random walker traversing the path Ω_{vu} is low.

Is the search information symmetric $S(\Omega_{vu}) = S(\Omega_{uv})$?

$$P(\Omega_{vu}) \stackrel{?}{=} P(\Omega_{uv})$$



$$\underbrace{P(\Omega_{vu}) \neq P(\Omega_{uv})}$$

How to make it symmetric for undirected graphs?

DIJKSTRA'S ALGO

KEY RULES

- Dijkstra's algorithm is iterative.
- At each iteration, we update the path lengths from a user-defined initial node to all other nodes in the graph.
- Each node i is assigned a single value ℓ_{oi} that represents its path length from the initial node o.
- We aim to iteratively decrease the path length values assigned to each node.
- Initial node is assigned a path length of zero, while all other nodes are assigned ∞.
- We start by visiting the neighbors of the initial node. The algorithm stops when all nodes are visited.

Key steps:

0/ Initialize path length in each node.

① create a list to keep track of each visited nodes.

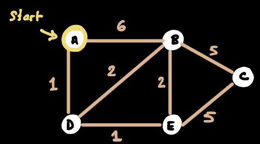
1/ From the starting node, $\boxed{\text{visit}}$ the node with the smallest known distance / cost.

② create a list to keep track of unvisited nodes.

2/ Once we've moved to the smallest-cost node, check each of its neighboring nodes.

3/ Calculate the distance/cost for the neighboring nodes by summing the cost of the edges leading from the start node.

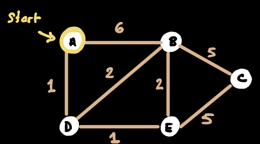
4/ If the distance/cost to a node we are checking is less than a known distance, update the shortest distance for that vertex.



node	shortest distance from A	previous node
A	0	
B	∞	
C	∞	
D	∞	
E	∞	

visited = [ ]

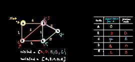
unvisited = [A, B, C, D, E]



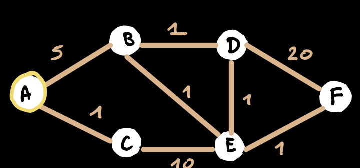
node	shortest distance from A	previous node
A	0	
B	∞	
C	∞	
D	∞	
E	∞	

visited = [ ]

unvisited = [A, B, C, D, E]



Exercise: run Dijkstra's algorithm step-by-step on the following graph.

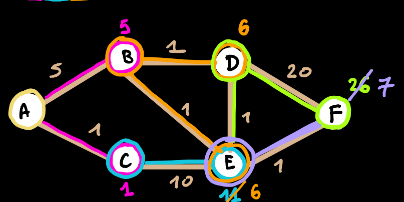


node	shortest distance	previous node
A	0	
B	∞	
C	∞	
D	∞	
E	∞	
F	∞	

visited = [ ]

unvisited = [A, B, C, D, E, F]

* A node is marked as visited when all its neighbors have been checked.
* Always visit the unvisited node with the smallest distance (in the table).



D or E

node	shortest distance	previous node
A	0	
B	$\boxed{5}$	A
C	$\boxed{1}$	A
D	$\boxed{6}$	B
E	~~11~~ $\boxed{6}$	~~C~~ B
F	~~26~~ 7	~~D~~ E

visited = [A, C, B, D, F]

unvisited = [ ]

* (Goñi et al. 2014) proposed a new search information definition as the average of $S(\Omega_{uv})$ and $S(\Omega_{vu})$:

$$S(\Omega_{vu}) = \frac{-log_2(P(\Omega_{vu})) - log_2(P(\Omega_{uv}))}{2}$$

* How to express the probability of a random walker beginning at u and reaching v by traversing Ω_{vu}?

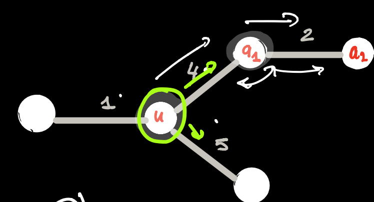
weight of the edge from u to a_1

$$P(\Omega_{vu}) = \frac{w_{a_1 u}}{s_u} \times \frac{w_{a_2 a_1}}{s_{a_1}} \times \cdots \times \frac{w_{v a_K}}{s_{a_K}}$$

strength of node u

$$s_j = \sum_{i \neq j} w_{ij}$$

probability that a random walker located at node u travels to node a_1 (directed graphs → outgoing strength)



$$P(\Omega_{a_1 u}) = \frac{4}{10}$$

$$P(\Omega_{a_2 u}) = \frac{4}{10} \times \frac{2}{6}$$

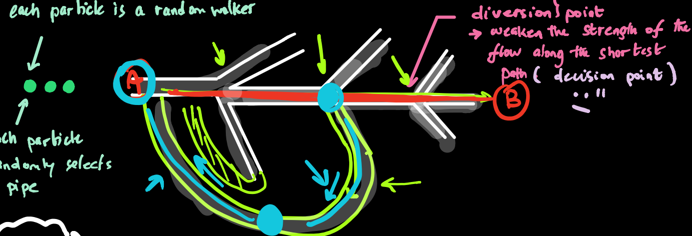
✓ or ✗

For graphs where higher weights entail better information flow:

- ✓ Edge weights should be remapped when computing shortest paths.
- ✓ Edge weights should not be remapped when computing the information search.

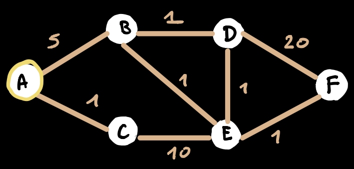
* Edge weights should not be remapped (i.e., when larger weights mean a stronger flow) when computing $P(\Omega_{vu})$ since the random walker will be biased towards choosing edges with low probability (lower weights).

Path transitivity (Goñi et al. 2014)



Fact

Although fluid can be diverted and weakened at each branch point, this diverted fluid may eventually return to the underlying shortest path at an upstream node.



node	shortest distance	previous node
A	0	
B	$\boxed{5}$	A
C	$\boxed{1}$	A
D	$\boxed{6}$	B
E	~~11~~ $\boxed{6}$	~~C~~ B
F	~~26~~ 7	~~D~~ E

D or E

what is the shortest path from A to F and its length?

Dijkstra's algorithm can handle weighted directed graphs but not negatively weighted edges!

Why is that?

How can we solve this? [Hint: Bellman-Ford algo] HW 2 :)

SHORTEST PATH ROUTING EFFICIENCY

1. Characteristic path length

The charecteristic shortest path length L is the average shortest path length between all possible pairs of nodes in a directed or undirected graph:

$$L = \frac{1}{N}\sum_i \boxed{l_i} = \underbrace{\frac{1}{N(N-1)}}_{\text{number of node pairs excluding self-pairing}} \sum_{i \neq j} \boxed{l_{ij}}$$

average shortest path from node i to all other nodes

shortest path length from node j to node i.

Intuition

A short characteristic path length L means that information can, on average, be routed between pairs of nodes using only a few connections.

" Can we compute L for fragmented graphs (with more than one connected component)?

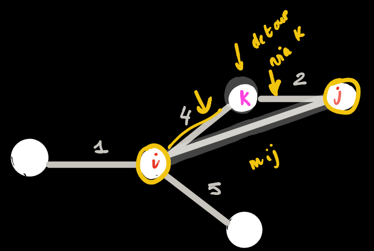
No since a path will not exist between at least one pair of nodes → $L = \infty$

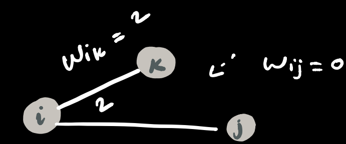
- Flow of information is restored.
- Branch point acts as a local detour.

The number of local detours is measured using path transivity.

* Path transitivity = number of local detours along a path.

* (Goñi et al. 2014) considered local detours that traverse two edges when measuring path transivity.





* For each possible pair of nodes i and j comprising the path under consideration, compute their matching index m_{ij}:

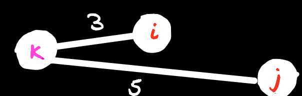
weight of edge between i and j

$$m_{ij} = \frac{\sum_{k\neq i,j}(w_{ik}+w_{jk})\mathbf{1}_{w_{ik}}\mathbf{1}_{w_{jk}}}{\underbrace{\sum_{k\neq j} w_{ik} + \sum_{k\neq i} w_{jk}}_{\text{normalizing constant} \Rightarrow m_{ij}\in[0,1]}}$$

$=1$ if $w_{ij} > 0$, otherwise 0

intermediate node along a detour between i and j

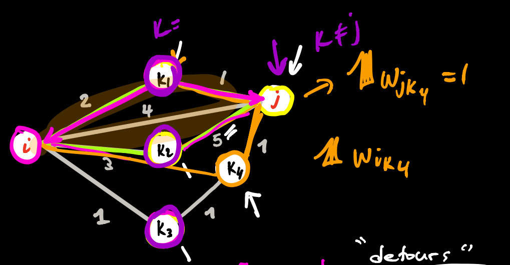
* For a detour via node k to exist, there must be an edge between nodes i and k as well as j and k.



$m_{ij} = ?$

if $e_{ik} = \emptyset \Rightarrow \mathbf{1}_{w_{ik}} = 0 \Rightarrow$ no edge $\Rightarrow m_{ij} = 0$

if $e_{jk} = \emptyset \Rightarrow \mathbf{1}_{w_{jk}} = 0 \Rightarrow$ no edge $\Rightarrow m_{ij} = 0$



$\mathbf{1}_{w_{ik_3}} = 1$

"detours" of 2-steps or hops

$$m_{ij} = \frac{\sum_{k\neq i,j}(w_{ik}+w_{jk})\mathbf{1}_{w_{ik}}\mathbf{1}_{w_{jk}}}{\sum_{k\neq j} w_{ik} + \sum_{k\neq i} w_{jk}}$$

$\mathbf{1}_{w_{jk_3}} = 0$

$$m_{ij} = \frac{(1+2)+(3+5)+0+0}{(\underbrace{2+3+1}_{6}) + (\underbrace{1+5+1}_{7})}$$

✓

$$m_{ij} = \frac{\overset{k_1}{(2+1)} + \overset{k_2}{(3+5)} + \overset{k_3}{0} + \overset{k_4}{0}}{6+7} = 0.84$$ ✓

Harmonic mean (Newman, 2003) estimates the average shortest path length between all possible node pairs in a fragmented graph.

$$\boxed{L'} = N(N-1)[\sum_{i\neq j} \boxed{\frac{1}{l_{ij}}}]^{-1}$$

harmonic mean is finite unless the graph has no edges at all!

reciprocal path is summed → if i & j belong to different components ⇒ $l_{ij} = \infty \Rightarrow \frac{1}{l_{ij}} = 0$

✓ or ✗

$$L' = N(N-1)[\sum_{i\neq j} \frac{1}{l_{ij}}]^{-1}$$

- [ ] Harmonic mean is sensitive to outliers (i.e, node pairs with exceptionally $\underbrace{\text{long shortest paths}}_{\text{their contribution is down-weighted}}$.

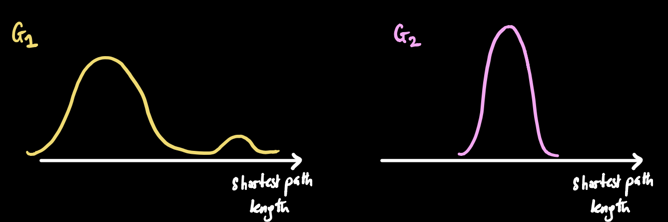
- [ ] Harmonic mean is a hub-centric measure of integration as it emphasizes the path lengths originating from hub regions and down-weights peripheral nodes.

E.g., shortest paths lengths in a graph: {2, 2, 3, 12}

↳ L = 4 and $\underbrace{L' = 2.1}$ more characteristic of the majority of path lengths in G.

The characteristic path length is a global measure of a graph's capacity to integrate information using shortest path routing. Using harmonic mean is desirable in fragmented graphs and is not unduly influenced by a small proportion of node pairs with exceptionally long shortest paths.

* Given the following distributions of shortest path lengths in two different graphs, specify which efficiency measure to use. Justify.



* Definition of the path transitivity of a path Ω_{vu}:

$\Omega = \{u, a_1, \dots, a_K, v\}$

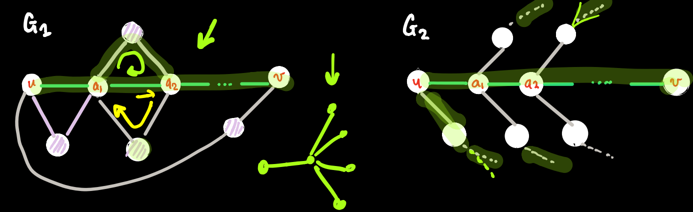
total number of node pairs in Ω_{vu}

$$M(\Omega_{vu}) = \frac{2}{|\Omega_{vu}||\Omega_{vu}-1|}\sum_{i>j\in\Omega_{vu}} m_{ij}$$

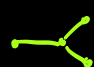
cardinality of set Ω_{vu}

Sum the matching index for every possible pair of nodes comprising Ω_{vu}

(1,2) (1,3)
(2,3)
(2,1) (3,1) (3,2)



1) any inf° that is diverted along a branch point returns to the underlying path at an upstream point.



2) Each branch does not return to the underlying path (at least not within two hops)
→ any information diverted along these branch points is unlikely to reach the desired destination.

* G1 or G2?

- [G2] Path with high search information and low transitivity
- [G1] Path with low search information and high transitivity.

→ G1 is more "accessible" in the absence of global knowledge of the graph structure.

→ The probability of a random walker from u to v to reach its destination is substantially small.

Recap

measure the ease with which a diffusion on G can find the shortest paths.

Path transitivity

$$M(\Omega_{vu}) = \frac{2}{|\Omega_{vu}||\Omega_{vu}-1|}\sum_{i>j\in\Omega_{vu}} m_{ij}$$

total number of node pairs in Ω_{vu}

cardinality of set Ω_{vu}

Sum the matching index for every possible pair of nodes comprising Ω_{vu}

Search information

$$S(\Omega_{vu}) = \frac{-\log_2(P(\Omega_{vu})) - \log_2(P(\Omega_{uv}))}{2}$$

Diffusion efficiency

* Goal → quantify the efficiency with which information can be communicated using a diffusion process.
* Given a random walker on a graph G, we will compute: two measures

Shortest path probability
↓
the probability of a random walker taking the shortest path to reach a particular destination.

Mean first passage time
↓
average number of hops that the random walker traverses to reach his destination.

2. Global efficiency

A graph's global efficiency, E_{glob}, is the reciprocal of the harmonic mean of its path lengths:

$$E_{glob} = \frac{1}{\underset{\text{harmonic mean}}{L'}} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{l_{ij}}$$

(Latora and Marchiori, 2011) contend that Eglob is the efficiency of information exchange in a parallel system in which nodes concurrently exchange information via shortest paths.

≠

whereas $\frac{1}{L}$ is a better measure of the efficiency of a sequential system in which information is processed serially.

3. Nodal efficiency

it is defined for each node j in G.

$$E_{nodal}(j) = \underbrace{\boxed{\frac{1}{N-1}}}_{\text{normalized}} \underbrace{\sum_i}_{\text{sum}} \boxed{\frac{1}{l_{ij}}}$$

reciprocal of the shortest path length from node j to node i

* Hub nodes tend to have the highest nodal efficiency, facilitating their role in efficiently integrating and distributing information.

"Any difference between closeness centrality and E_{nodal} of a node in a graph G?"

3. Nodal efficiency

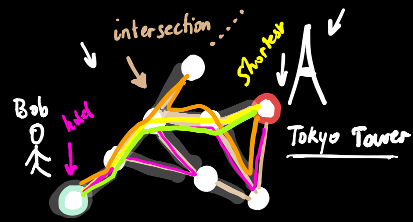
(Latora and Marchiori, 2011) define a node-specific measure:

$$E_{local}(i) = \frac{1}{N_{G_i}(N_{G_i}-1)} \underbrace{\sum_{j,h \in G_i} \frac{1}{l_{jh}}}$$

G_i: subgraph comprising all immediate neighbors of node i.

how efficiently the neighbors of a node can communicate when that node i is disrupted.

reflects the extent to which a node in integrated between its immediate neighbors ($\neq E_{glob}$)



* Diffusion efficiency definition (Goñi et al. 2013)

It is the number of times a random walker needs to commence a new random walk from his starting point to ensure he reaches his destination on at least one occasion by the shortest path.

* How to derive a measure of diffusion efficiency?

1) Define a matrix U:

$$U = WS^{-1}$$

W: $N \times N$ graph adjacency matrix (weighted or binary)

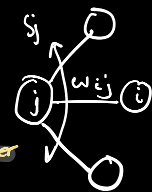
$S = \begin{bmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_N \end{bmatrix}$ — s_i: strength of the i^{th} node; strength of node

$U = \begin{bmatrix} & j & \\ i & \square & \\ & & \end{bmatrix}$

$$U_{ij} = w_{ij} / s_j \in \mathbb{R}$$

element (i,j) of matrix U

probability that a random walker goes from node j to node i



$$P(\Omega_{vu}) = \frac{w_{a_1 u}}{s_u} \times \frac{w_{a_2 a_1}}{s_{a_1}} \times \cdots \times \frac{w_{v a_K}}{s_{a_K}}$$

P: used in the search information $S(\Omega_{vu})$

$U_{a_1 u}$, $U_{a_2 a_1}$

$u \to a_1 \to a_2 \to \cdots \to a_K \to v$

$$U = WS^{-1}$$

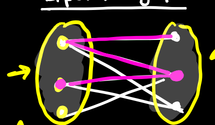
W: adjacency matrix; S: diagonal strength / degree matrix

- **Markov chain** → U is a transition matrix of the Markov chain with states corresponding to the network nodes.

- **Graph Laplacian**

  - Laplacian of a graph G is a transformation of its connectivity or adjacency matrix.
  - The eigenvalues of graph Laplacian provide clues about:
    - its connected components
    - community structure
    - bipartiness and more.

bipartite graph

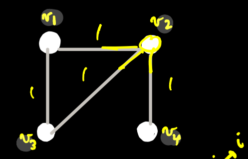


$$\Lambda = \underbrace{S - W}_{\text{reverses the polarity of the off-diagonal}} = \begin{cases} s_i / s_i = 1, & i = j \\ -w_{ij} / s_i, & i \neq j \end{cases}$$

Laplacian matrix $\in \mathbb{R}^{N \times N}$

and set the diagonal to the node degree/ strength.

$$\Lambda = \begin{array}{c c} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \end{array}$$



Normalized Laplacian

$$\boxed{\Lambda' = 1 - \underbrace{WS^{-1}}_{0}} = \begin{cases} 1, & i = j \\ \boxed{-w_{ij}/s_j}, & i \neq j \end{cases}$$

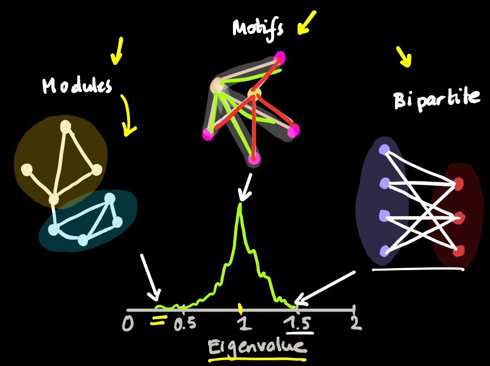
? ! $j \to i$

transition probability of a random walker stepping across the graph.

$$\Lambda' = \begin{array}{c c} & \begin{array}{cccc} j & & & \end{array} \\ i & \begin{bmatrix} 1 & -1/3 & -1 & 0 \\ -1/2 & 1 & -1 & -1 \\ -1/2 & -1/3 & 2 & 0 \\ 0 & -1/3 & 0 & 1 \end{bmatrix} \end{array}$$

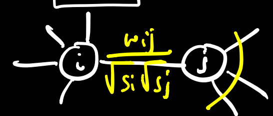
Note: in many text books: $\Lambda' = 1 - S^{-1}W$
- w_{ij} : go from i to j
- w_{ij}/s_i : probability of going from i to j.

- Eigenvalues of Λ' are between 0 and 2 (Chung, 1997).



- Symmetric normalized Laplacian matrix:

$$\Lambda'_{sym} = 1 - S^{-1/2} W S^{-1/2}$$



an element (ij) of this matrix:

$$w_{ij}/\sqrt{s_i}\sqrt{s_j}$$

1) Define a matrix U:

$$U = WS^{-1}$$

N×N graph adjacency matrix (weighted or binary)

strength of the i^{th} node

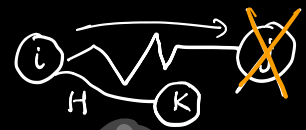
$$S = \begin{bmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_N \end{bmatrix}$$

2) Compute the shortest path probability to go from node i to j:

- $U = \begin{bmatrix} 1 & & \frac{w_{1n}}{s_n} \\ \vdots & \ddots & \vdots \\ \frac{w_{n1}}{s_1} & \cdots & 1 \end{bmatrix}$

- $U_j^H = \begin{bmatrix} 1 & & \frac{w_{1n}}{s_n} \\ 0 & \cdots & 0 \\ \frac{w_{n1}}{s_1} & \cdots & 1 \end{bmatrix}$ H, K, i, j

set to zero



a walker cannot reach node j from any node ($P_{\Omega jk} = 0$, $\forall k$)

* U_j^H : expresses the probability of a random walker travelling from one node to another in exactly H steps with no option to visit j.

$i \xrightarrow{H \text{ hops}} j$

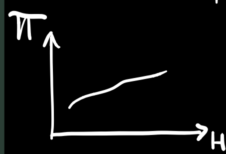
$$\pi_{ij} = 1 - \sum_{n=1}^{N} [U_j^H]_{ni}$$

probability that the walker is at node j in H steps

- Summing the (n,i) elements of U_j^H over all nodes $n = 1 \ldots N$

→ probability that the walker is at any node other than j after H steps.

3) Shortest path probability of the whole graph:



$$\boxed{\sum_{i \neq j} \frac{1}{N-1} \pi_{ij}^H = \Pi_H}$$

two measures

Shortest path probability	Mean first passage time
the probability of a random walker taking the shortest path to reach a particular destination.	average number of hops that the random walker traverses to reach his destination.

Average number of steps

- Let X_{ij} denote the number of steps/hops a random walker takes to go from node i to j (Wang and Pei, 2008):

$$< X_{ij} > = \sum_{t=0}^{\infty} P(X_{ij} > t)$$

$$= \sum_{t=0}^{\infty} \sum_{n=1}^{N} [U_j^t]_{ni}$$

$n = j$ //

$$\sum_{n=1}^{N} [(1 - U_j)^{-1}]_{ni}$$

probability of a random walker requiring more than t hops to arrive at j.

=

Σ of probabilities of the walker is still being at one of the other nodes than j after exactly t hops.

the sum includes node j but it is not problematic since $[U_j^t]_{ji} = 0$

geometric series

$$\begin{cases} \sum_{n=0}^{\infty} a r^n = \frac{a}{1-r} \\ -1 < r < 1 \end{cases}$$

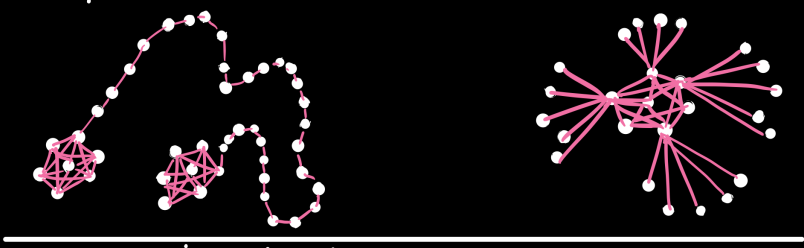
$(1-r)^{-1}$

Diffusion efficiency

- (Goñi et al., 2013) average the reciprocal of the mean first passage time over all node pairs to define diffusion efficiency:

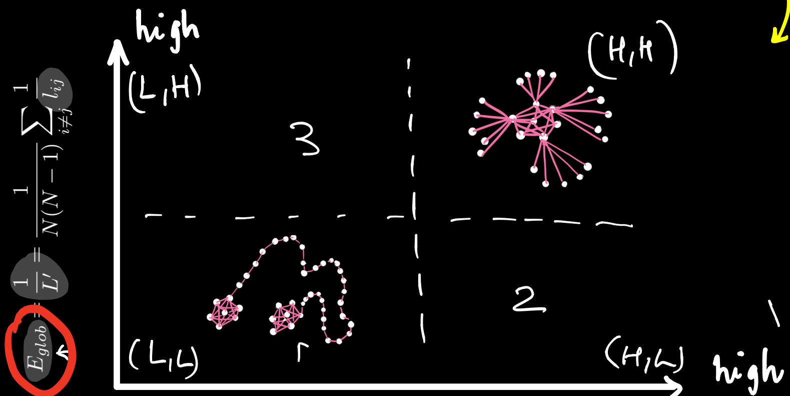
$$E_{diff} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{< X_{ij} >}$$

Example



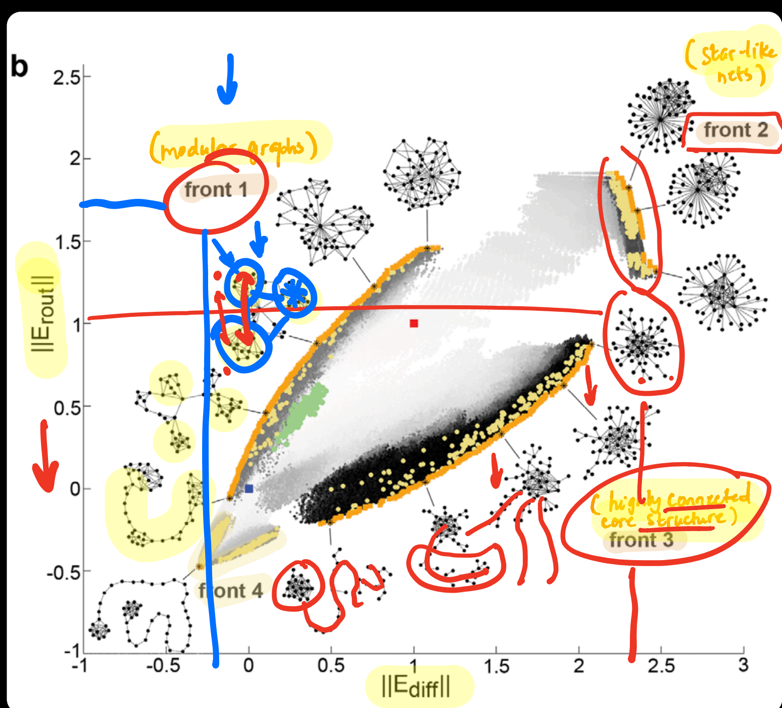
information flow Efficiency measures

place them where appropriate



$$E_{glob} = \frac{1}{L} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{l_{ij}}$$

$$E_{diff} = \frac{1}{N(N-1)} \sum_{i \neq j} \frac{1}{< X_{ij} >}$$



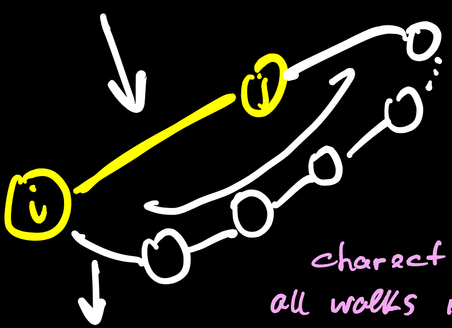
Source: Goñi et al, 2013 _ Plos ONE.
Exploring the morphospace of communication efficiency in complex networks.

COMMUNICABILITY

* A common measure of graph integration that is consistent with the diffusion model of information flow.

  - It accounts for the contribution of all possible walks between a pair of nodes.

  - If A is a binary adjacency matrix, communicability between nodes i and j is defined as (Estrada and Hatano, 2018):



total number of walks between i and j comprising n edges

$$Com_{ij} = \sum_{n=0}^{\infty} \frac{[A^n]_{ij}}{n!} = e^A$$

characterizes all walks in a graph that traverse exactly n hops

matrix exponential of A

Taylor expansion of e

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

  - Communicability between i and j is the weighted sum of the total number of walks between a node pair.

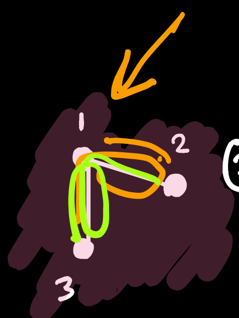
$1/n!$ weighting → longer walks ($n \uparrow$) contribute less to the sum, while direct connections contribute the most.

  - Example :

$$A = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & \boxed{0} \\ 1 & 0 & 0 \end{bmatrix}^{n=1}, \quad A^2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & \boxed{1} \\ 0 & 1 & 1 \end{bmatrix}^{n=2}, \quad A^3 = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 0 & \boxed{0} \\ 2 & 0 & 0 \end{bmatrix}^{n=3}, \quad A^4 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & \boxed{2} \\ 0 & 2 & 2 \end{bmatrix}$$

① * A^n_{23} $\boxed{=0}$ when $n = 1, 3, 5, \dots$ (odd number)

↳ this node pair can only be connected with a walk that traverses an even number of hops.

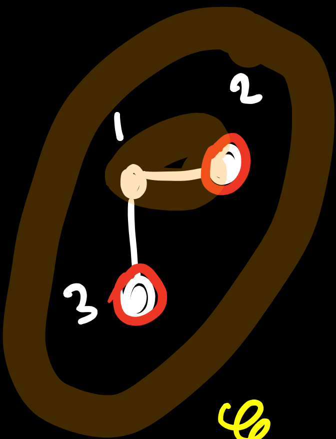


② * $A^4_{23} = \boxed{2}$ ⇒ two walks between nodes 2 and 3

walk 1 ⇒ 2 − 1 − 2 − 1 − 3

walk 2 ⇒ 2 − 1 − 3 − 1 − 3

③ what is the communicability between nodes 2 and 3?

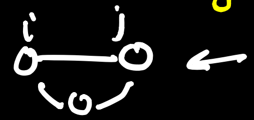


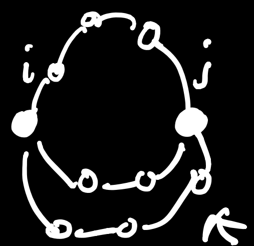
$$Com_{23} = \sum_{n=0}^{\infty} \frac{[A^n]_{23}}{n!} \approx 0.589$$

$n = 4$

$$Com_{23} = \underbrace{\frac{\boxed{?}}{1} + \frac{?}{2} + \frac{?}{6} + \frac{?}{24}}_{\text{only the first 4 terms}} \approx 0.583$$

what do you notice?





→ The number of walks that traverse back and forth along the same edge more than four times is minimal in this example due to the $1/n!$ weighting.

* Generalizing communicability to weighted graphs

- Replace the binary matrix A with a normalized version to regulate the influence of nodes with high strength:

$$A \longrightarrow \underbrace{S^{-1/2} W S^{-1/2}}_{\text{reduced adjacency matrix}}$$

element (i,j)

$$\boxed{w_{ij} / \sqrt{s_i}\sqrt{s_j}}$$

$$\boxed{Com_{ij} = \sum_{n=0}^{\infty} \frac{1}{n!} \times \underbrace{[(S^{-1/2} W S^{-1/2})^n]_{ij}}_{\text{related to the probability of a random walker reaching i from j in n steps}}}$$

reflects a graph's capacity for parallel transfer under the diffusion model of information flow