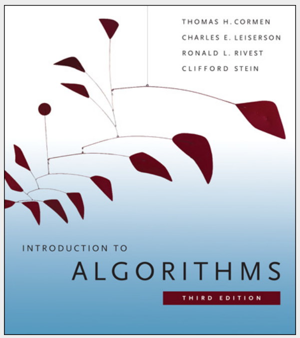
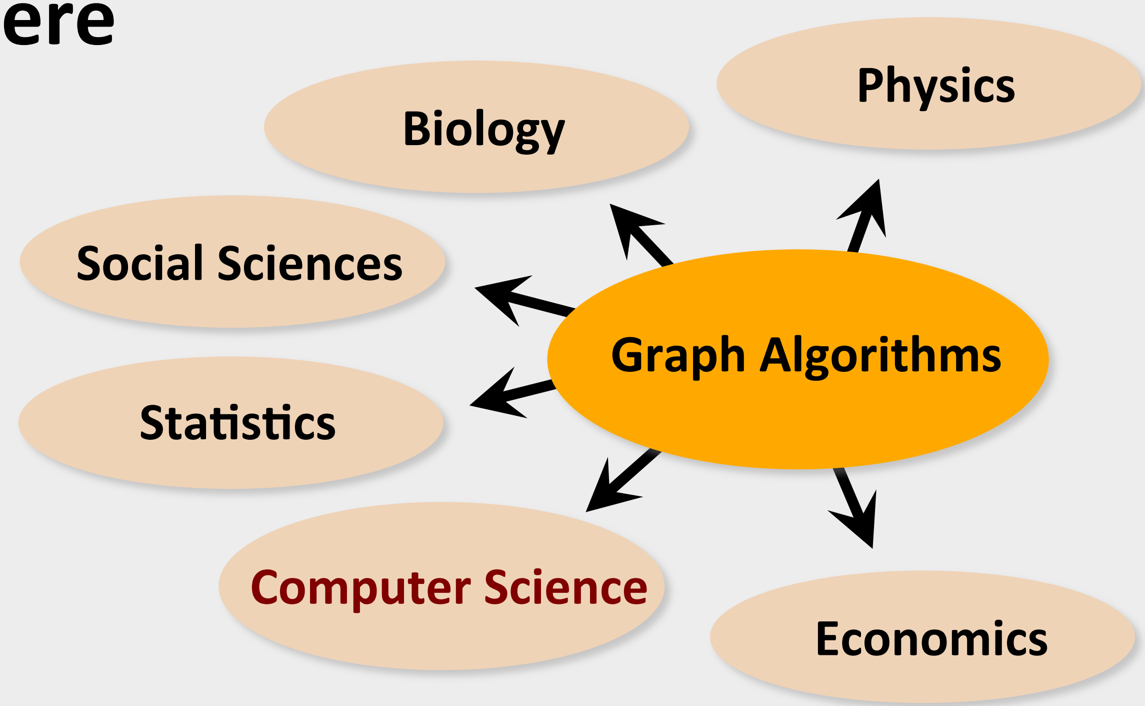
Electrical Flows, Laplacian Matrices, and New Approaches to the Maximum Flow Problem

Aleksander Mądry

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Graphs are **everywhere**



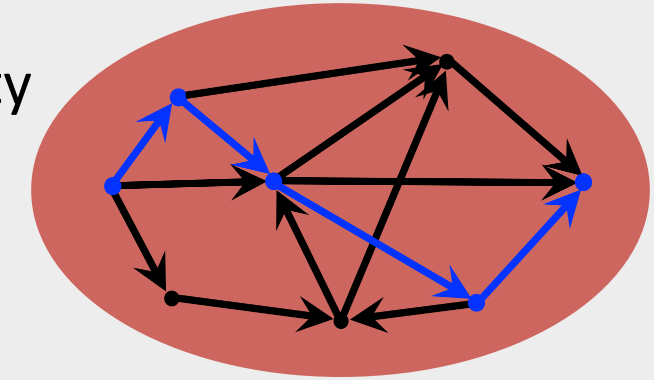


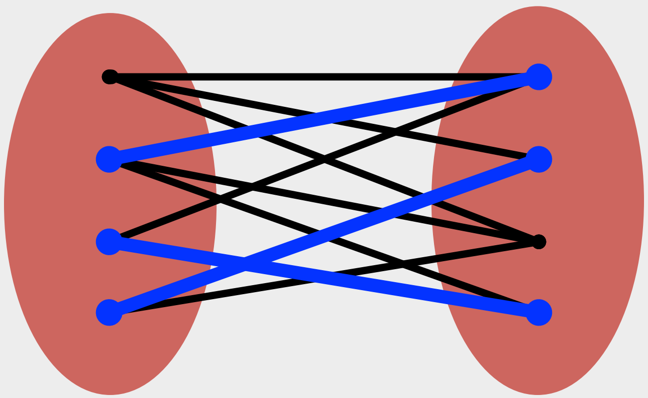
Algorithmic Graph Theory:
Shaping our understanding of algorithms since 1950s

But: Our graph toolkit is still far from being complete

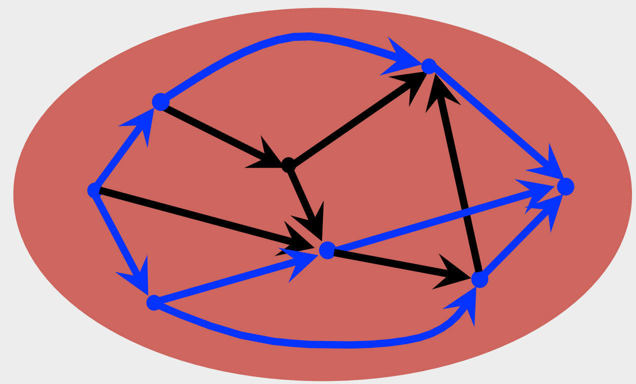
Challenge I: Tackling the complexity of core problems

Shortest path/Reachability problems





Matchings/Assignment tasks

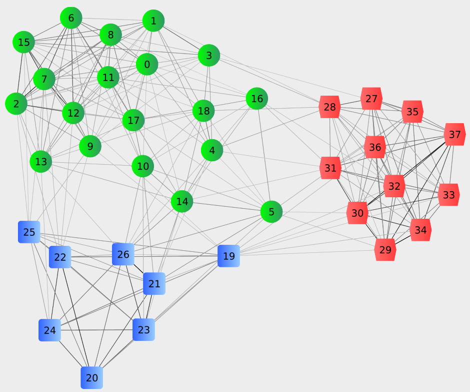


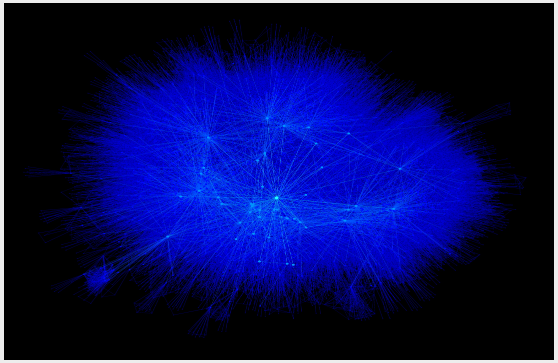
Network Flows

Many central questions is resisting progress for a long time

Possible reason: Our approaches to them did not change much over the last couple of decades

Challenge II : Dealing with massive graphs





"Big graph" regime:

→ **Asymptotics matters – ("O(n^2) won't cut it")**

Algorithms can be dirty, but have to be REALLY fast

→ **Parallelism/distributed aspects increasingly important**

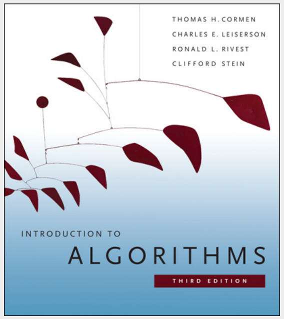
Emerging mindset: What can we do in **nearly-linear time**?

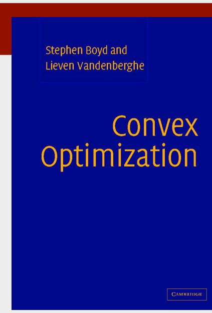
"**nearly-linear time**" = new **P**

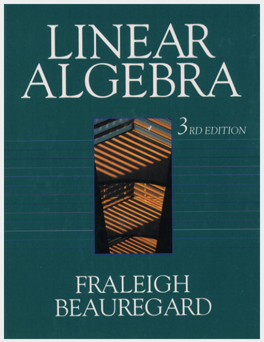
Problem: Traditional techniques seem inadequate here

→ New type of approaches needed

Grand goal: Forging the next generation of tools to speed up graph algorithms







Linear-algebraic tools
(eigenvalues, electrical flows, linear systems,...)

Combinatorial methods
(trees, paths, partitions, matchings, routings,...)

Convex opt. primitives
(gradient-descent, interior-point methods,...)

Underlying theme: Merging combinatorial and continuous methods

Interestingly: The same new tools apply to both classic and new challenges

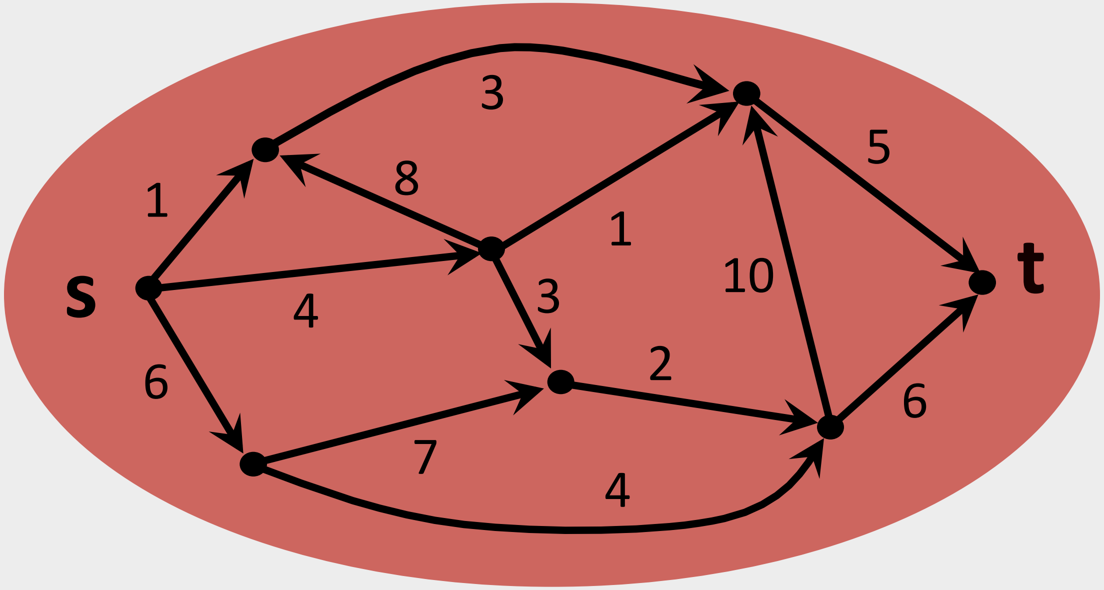
Our plan for this week:
Illustrate this theme on an example
of a single problem

Problem: Maximum flow

Underlying approach:
Relate combinatorial structure of a graph to
linear-algebraic properties of associated matrices

Maximum flow problem

Input: Directed graph **G**,
integer **capacities u_e**,
source s and **sink t**



Think: arcs = roads
capacities = # of lanes
s/t = origin/destination

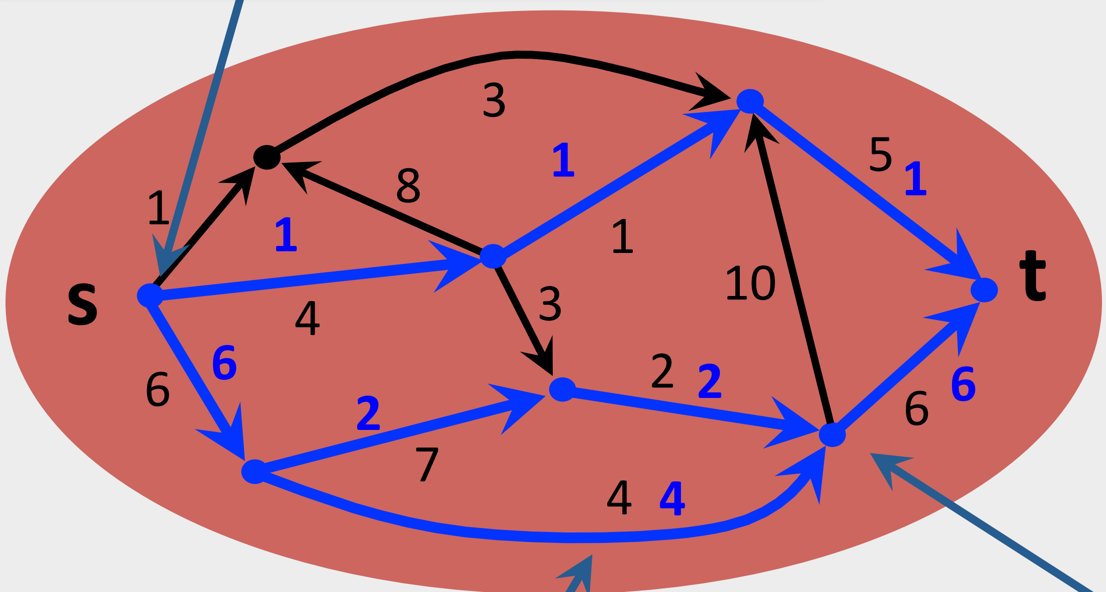
Task: Find a **feasible s-t flow** of **max value**

(**Think:** Estimate the **max** possible rate of traffic from **s** to **t**)

Maximum flow problem

Input: Directed graph **G**, integer **capacities u_e**, **source s** and **sink t**

value = net flow out of **s**



Think: arcs = roads
capacities = # of lanes
s/t = origin/destination

Max flow value
F*=10

no overflow on arcs:
$0 \leq f(e) \leq u(e)$

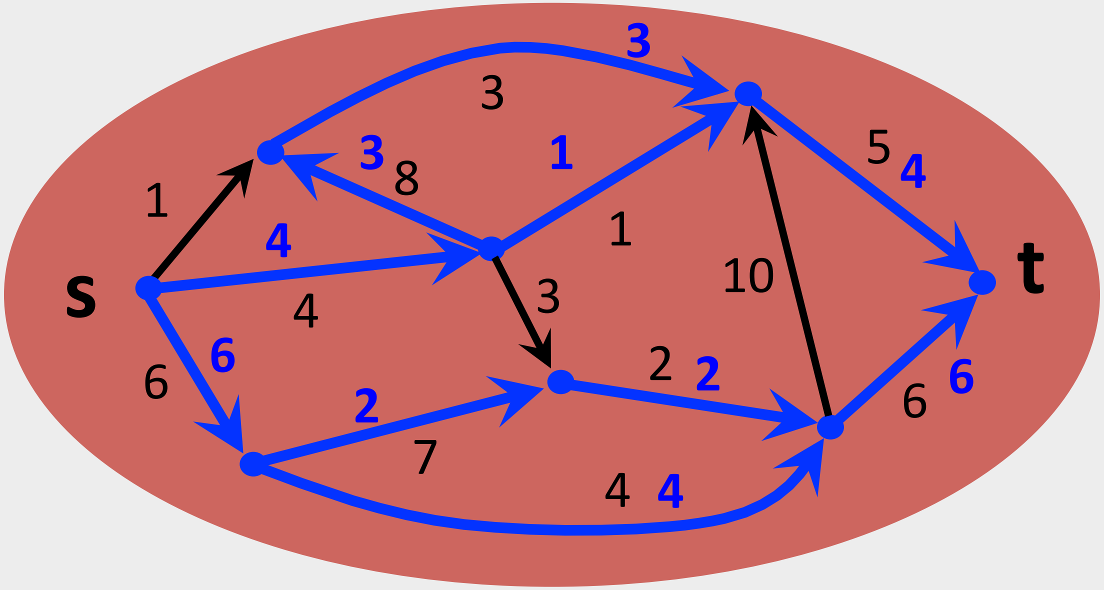
no leaks at all **v≠s,t**

Task: Find a **feasible s-t flow** of **max value**

(**Think:** Estimate the **max** possible rate of traffic from **s** to **t**)

Maximum flow problem

Input: Directed graph **G**, integer **capacities** u_e, **source s** and **sink t**



Think: arcs = roads
capacities = # of lanes
s/t = origin/destination

Max flow value
F*=10

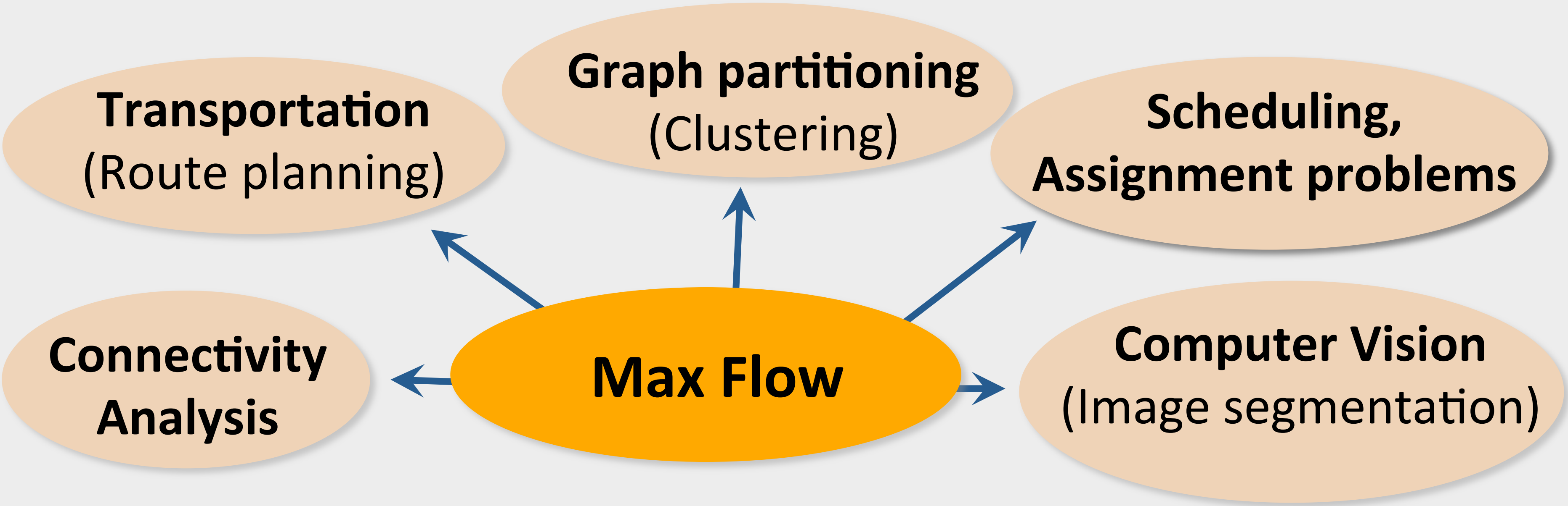
Task: Find a **feasible s-t flow** of **max value**

(**Think:** Estimate the **max** possible rate of traffic from **s** to **t**)

Why is this a good problem to study?

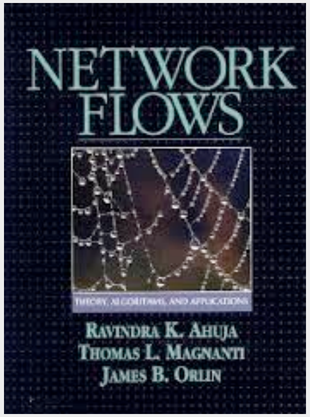
Max flow is a fundamental optimization problem

- **Extensively studied since 1930s** (classic 'textbook problem')
- **Surprisingly diverse set of applications**
- **Very influential in development of (graph) algorithms**



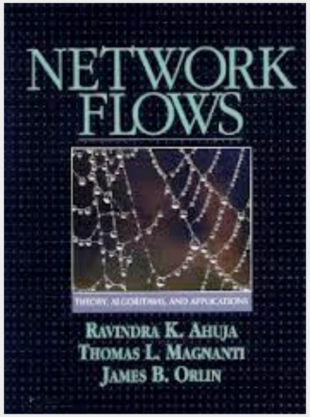
What is known about Max Flow?

A **LOT** of previous work



What is known about Max Flow?

A (very) rough history outline



[Dantzig '51]	$O(mn^2 U)$
[Ford Fulkerson '56]	$O(mn\ U)$
[Dinitz '70]	$O(mn^2)$
[Dinitz '70] [Edmonds Karp '72]	$O(m^2 n)$
[Dinitz '73] [Edmonds Karp '72]	$O(m^2 \log U)$
[Dinitz '73] [Gabow '85]	$O(mn \log U)$
[Goldber Rao '98]	$\tilde{O}(m \min(m^{1/2}, n^{2/3}) \log U)$

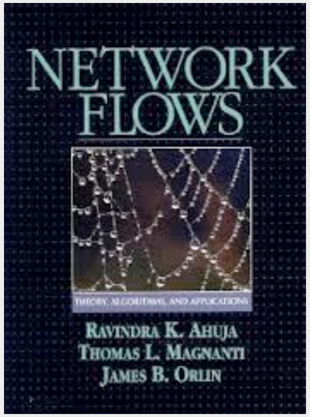
Our focus: Sparse graph (**m=O(n)**) and unit-capacity (**U=1**) regime

→ It is a good benchmark for combinatorial graph algorithms
→ Already captures, e.g., **bipartite matching** questions

(**n** = # of vertices, **m** = # of arcs, **U** = max capacity, **Õ()** hides polylogs)

What is known about Max Flow?

A (very) rough history outline



[Dantzig '51]	**$O(n^3)$**
[Ford Fulkerson '56]	**$O(n^2)$**
[Dinitz '70]	**$O(n^3)$**
[Dinitz '70] [Edmonds Karp '72]	**$O(n^3)$**
[Dinitz '73] [Edmonds Karp '72]	**$\tilde{O}(n^2)$**
[Dinitz '73] [Gabow '85]	**$\tilde{O}(n^2)$**
[Goldber Rao '98]	**$\tilde{O}(n^{3/2})$**

Our focus: Sparse graph (**m=O(n)**) and unit-capacity (**U=1**) regime

→ It is a good benchmark for combinatorial graph algorithms
→ Already captures, e.g., **bipartite matching** questions

(**n** = # of vertices, **m** = # of arcs, **U** = max capacity, **Õ()** hides polylogs)

What is known about Max Flow?

Emerging barrier: $O(n^{3/2})$

[Even Tarjan '75, Karzanov '73]: Achieved this bound for **U=1** long time ago

Last 40 years: Matching this bound in increasingly more general settings, but **no improvement**

This indicates a fundamental limitation of our techniques

Our goal: Show a new approach finally breaking this barrier

(**n** = # of vertices, **m** = # of arcs, **U** = max capacity, **Õ()** hides polylogs)

Breaking the $O(n^{3/2})$ barrier

Undirected graphs and approx. answers ($O(n^{3/2})$ barrier still holds here)

[M '10]: Crude approx. of max flow **value** in **close to linear** time

[CKMST '11]: (1-ε)-approx. to max flow in $\tilde{O}(n^{4/3}\varepsilon^{-3})$ time

[LSR '13, S '13, KLOS '14]: (1-ε)-approx. in **close to linear** time

But: What about the **directed** and **exact** setting?

[M '13]: Exact $\tilde{O}(n^{10/7})=\tilde{O}(n^{1.43})$-time alg.

This week

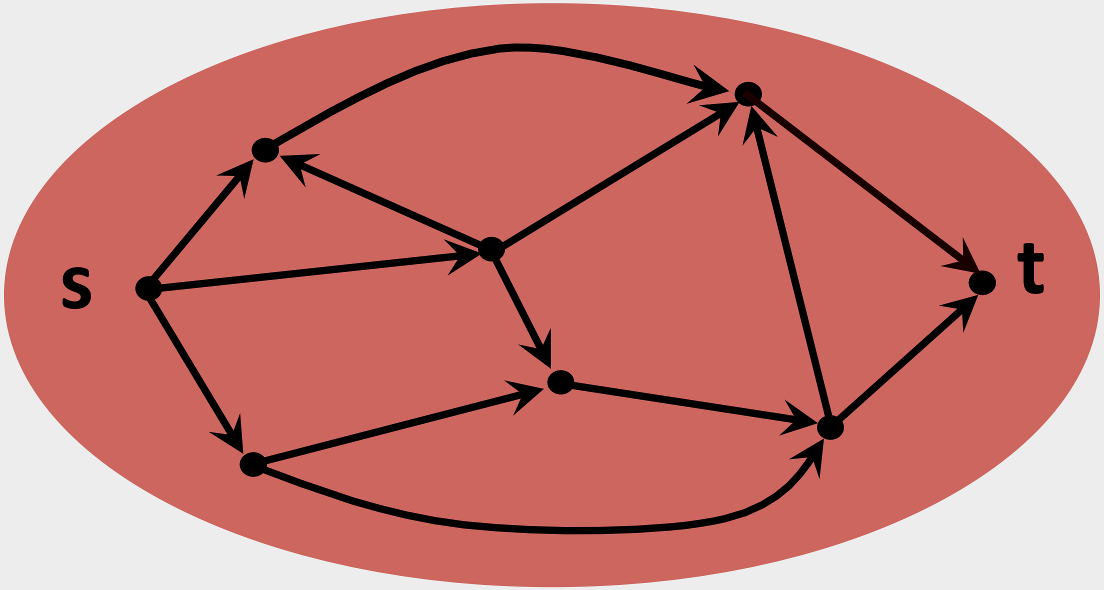
(**n** = # of vertices, $\tilde{O}()$ hides polylog factors)

Previous approach

Augmenting paths framework

[Ford Fulkerson '56]

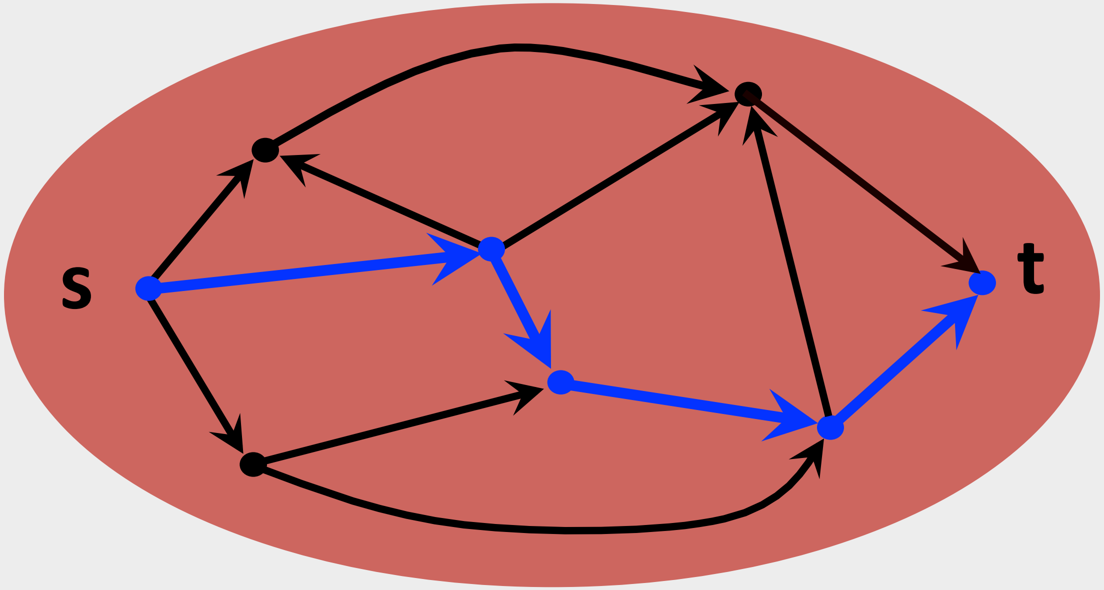
Basic idea: Repeatedly find **s-t paths** in the **residual graph**



Augmenting paths framework

[Ford Fulkerson '56]

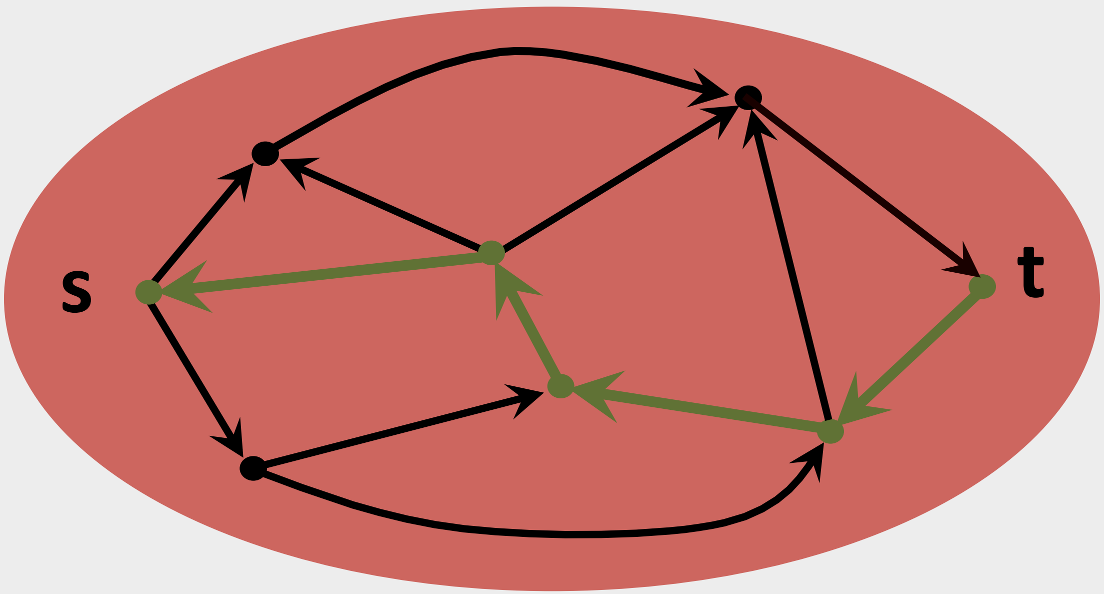
Basic idea: Repeatedly find **s-t paths** in the **residual graph**



Augmenting paths framework

[Ford Fulkerson '56]

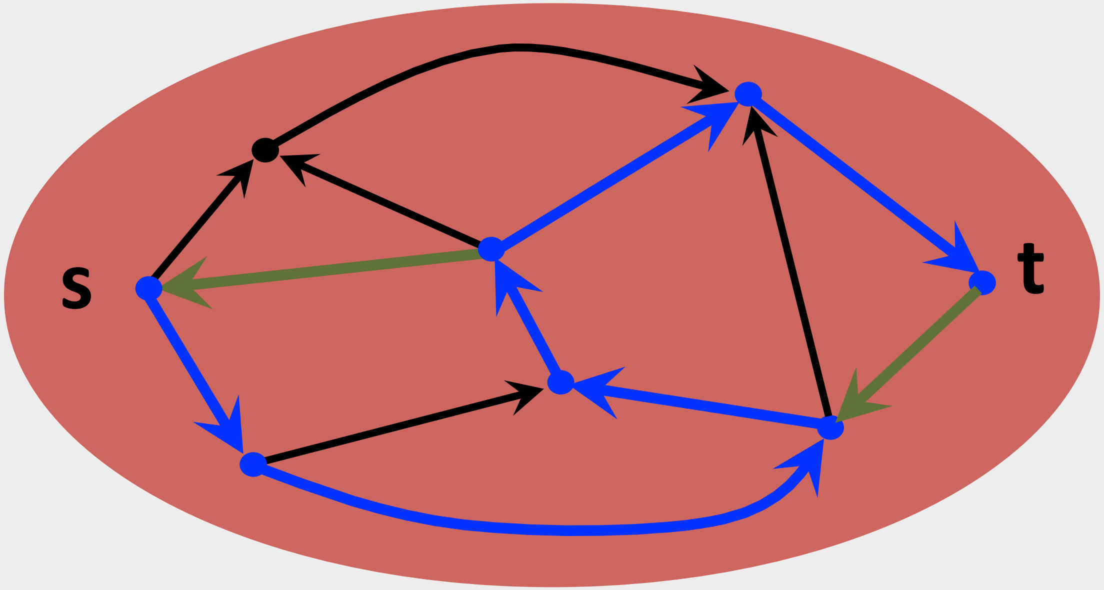
Basic idea: Repeatedly find **s-t paths** in the **residual graph**



Augmenting paths framework

[Ford Fulkerson '56]

Basic idea: Repeatedly find **s-t paths** in the **residual graph**

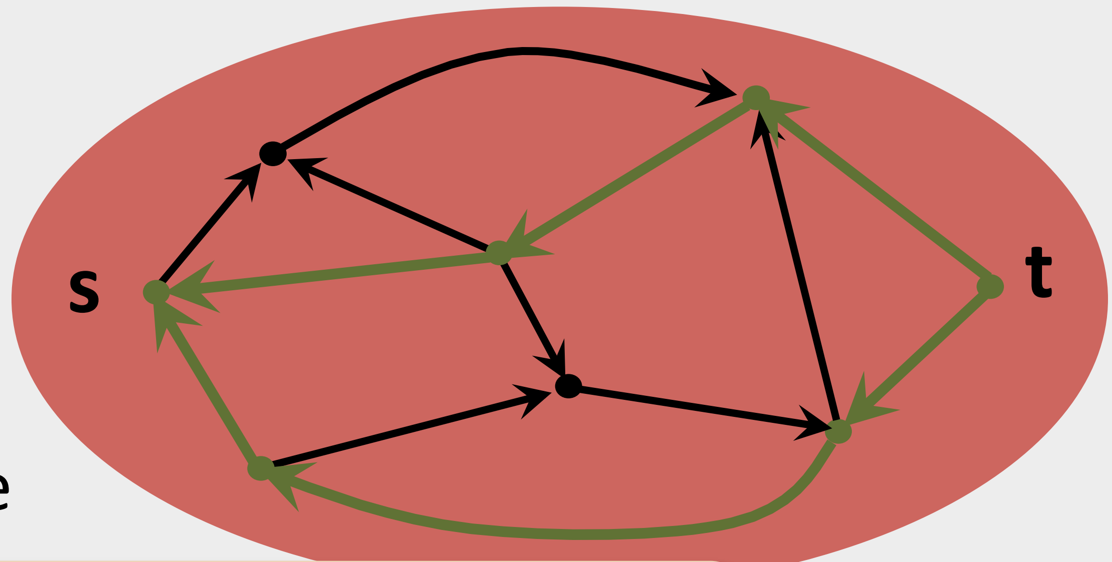


Augmenting paths framework

[Ford Fulkerson '56]

Basic idea: Repeatedly find **s-t paths** in the **residual graph**

Advantage: Simple, purely combinatorial and greedy (flow is built path-by-path)



Problem:
Very difficult to analyze

Naïve impl

(≤ **n** augme … path)

Unclear how to get
a further speed-up via this route

Sophisticated implementation and arguments:

$O(n^{3/2})$ time **[Karzanov '73] [Even Tarjan '75]**

Beyond augmenting paths

New approach:

Bring linear-algebraic techniques into play

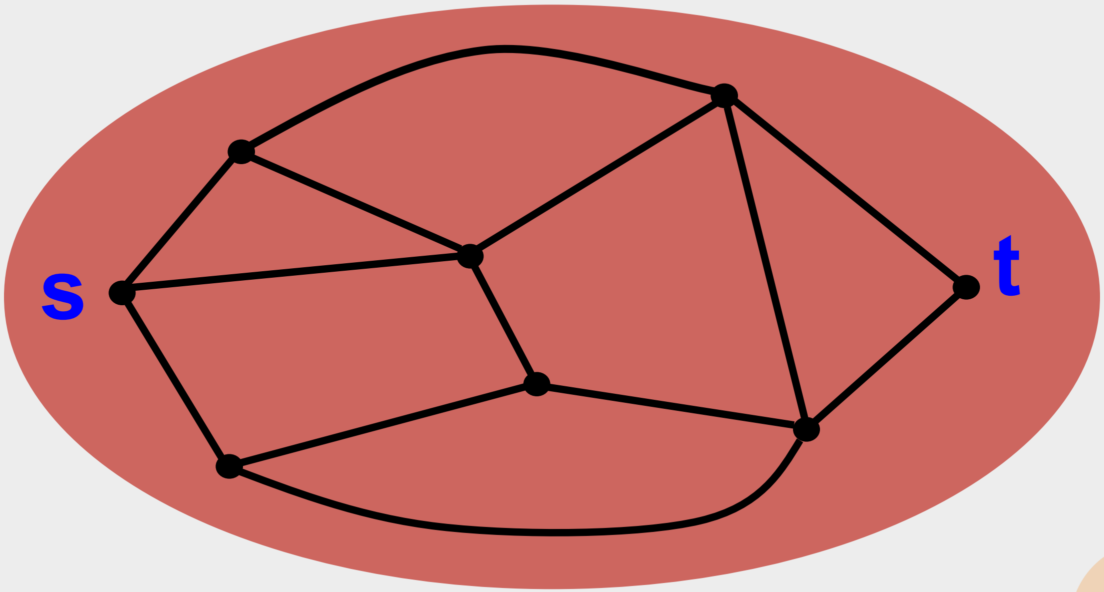
Idea: Probe the **global flow structure** of the graph by **solving linear systems**

How to relate **flow structure** to **linear algebra**?
(And why should it even help?)

Key object: Electrical flows

Electrical flows (Take I)

Input: **Undirected** graph **G**, **resistances** r_e, **source s** and **sink t**

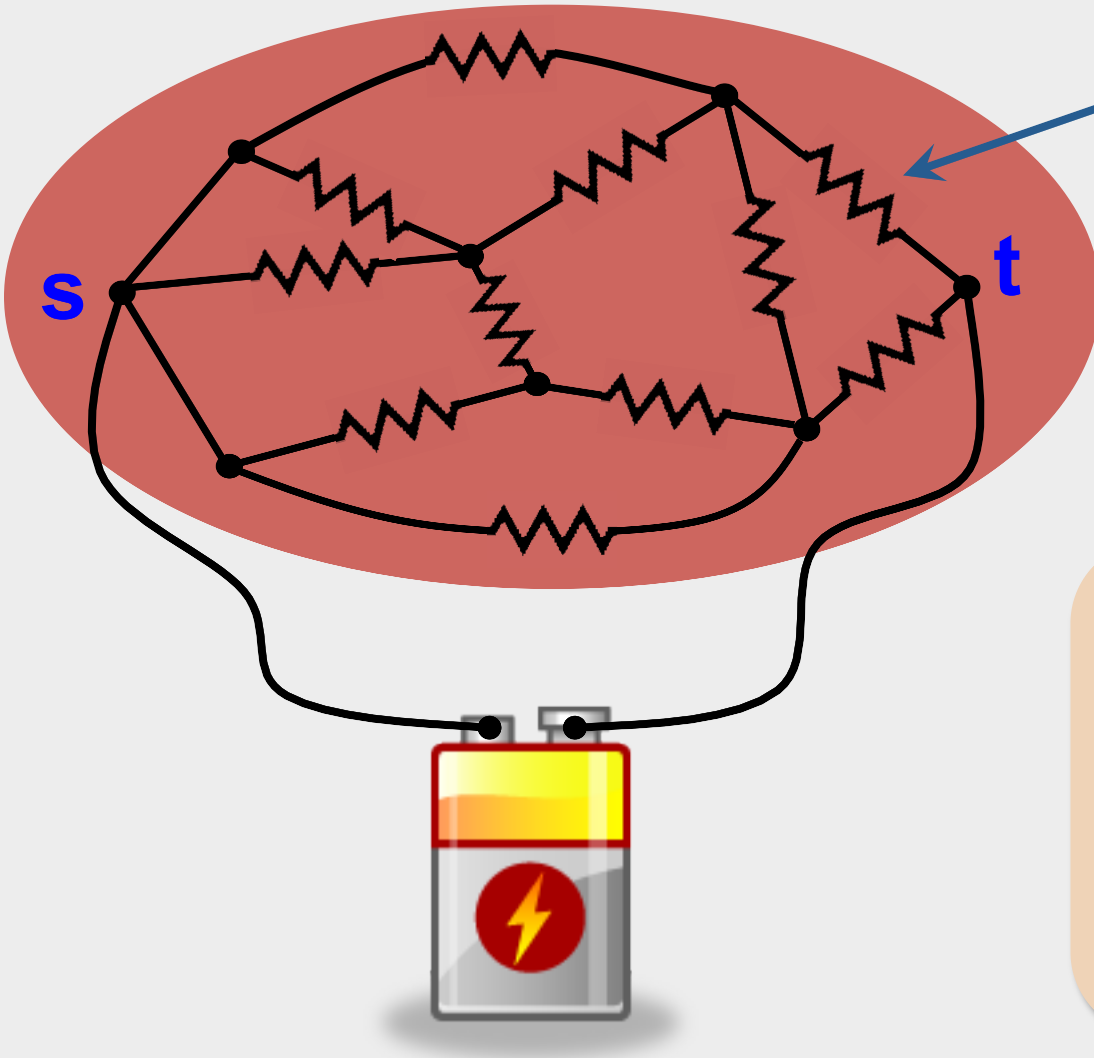


Recipe for elec. flow:

1) Treat edges as **resistors**

Electrical flows (Take I)

Input: **Undirected** graph **G**, **resistances r_e**, **source s** and **sink t**



resistance r_e

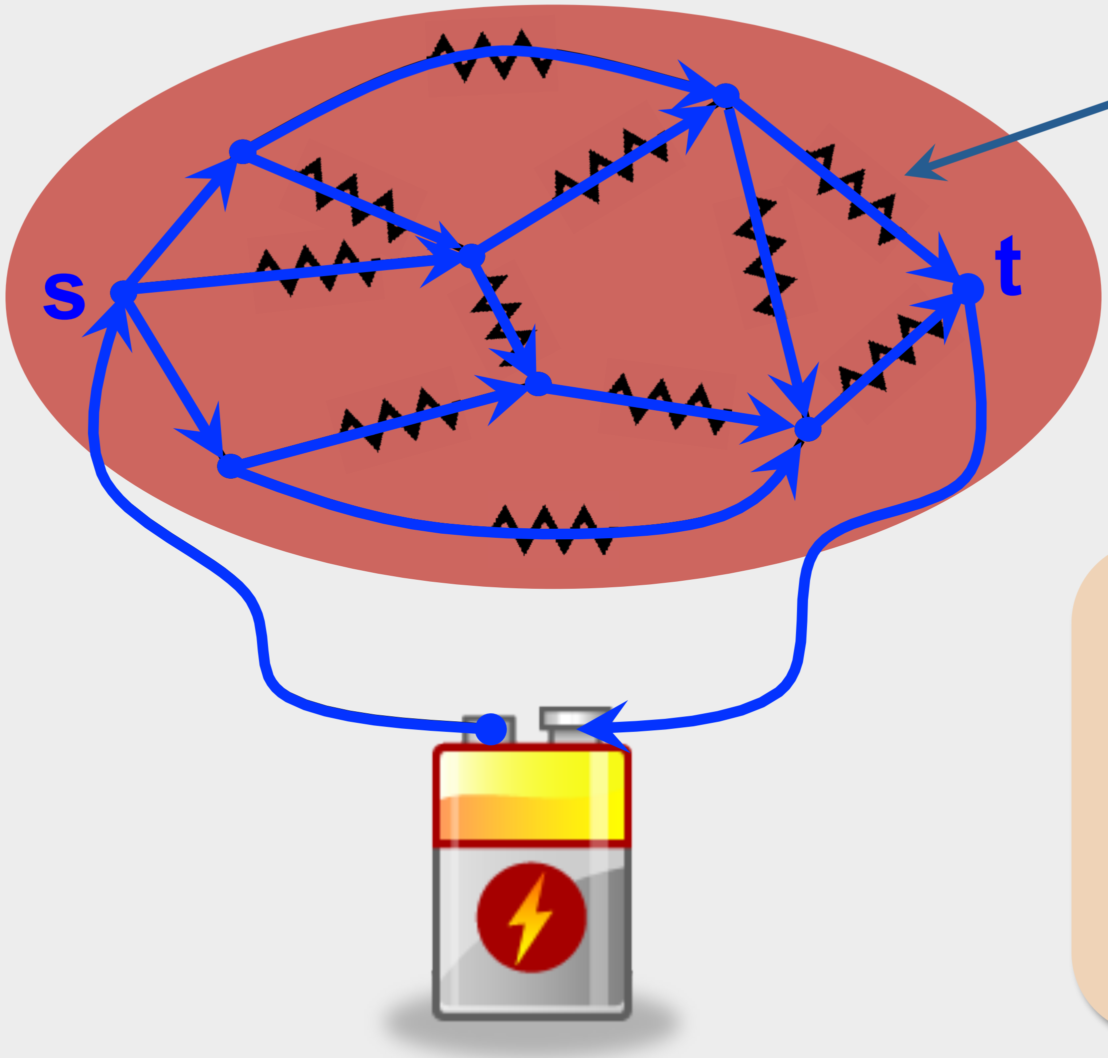
Recipe for elec. flow:

1) Treat edges as **resistors**

2) Connect a **battery** to **s** and **t**

Electrical flows (Take I)

Input: **Undirected** graph **G**,
resistances r_e,
source s and **sink t**



resistance r_e

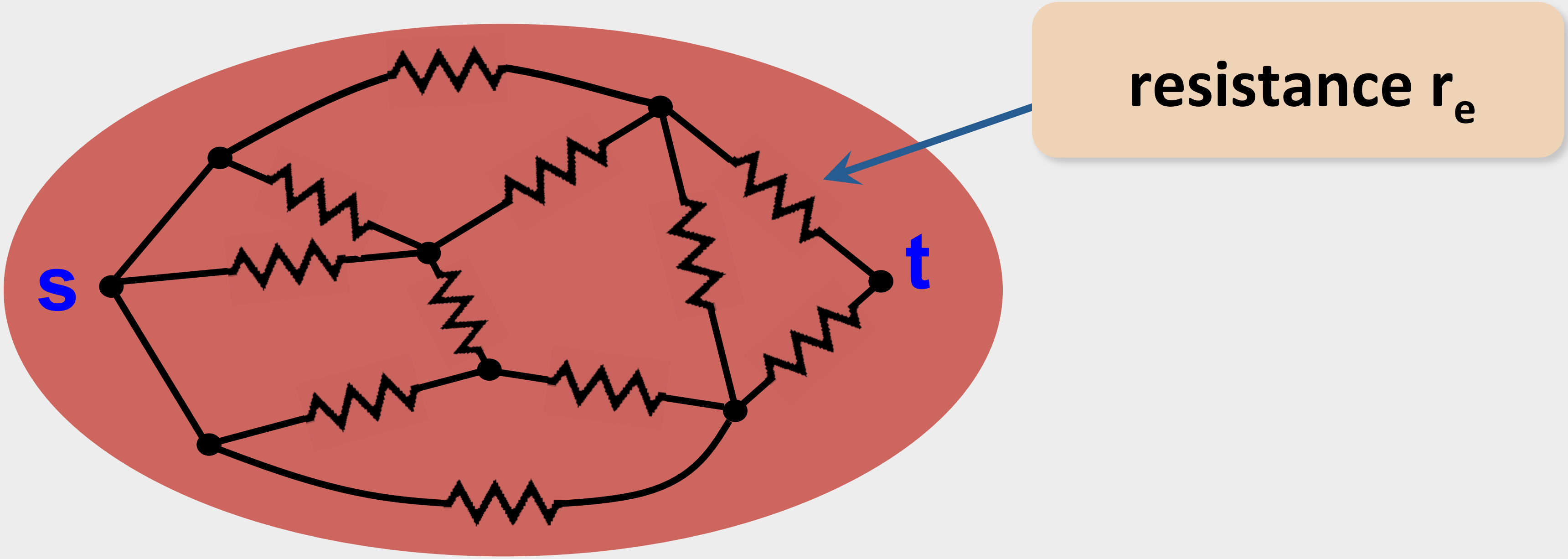
Recipe for elec. flow:

1) Treat edges as **resistors**

2) Connect a **battery** to **s** and **t**

Electrical flows (Take II)

Input: **Undirected** graph **G**, **resistances r_e**, **source s** and **sink t**

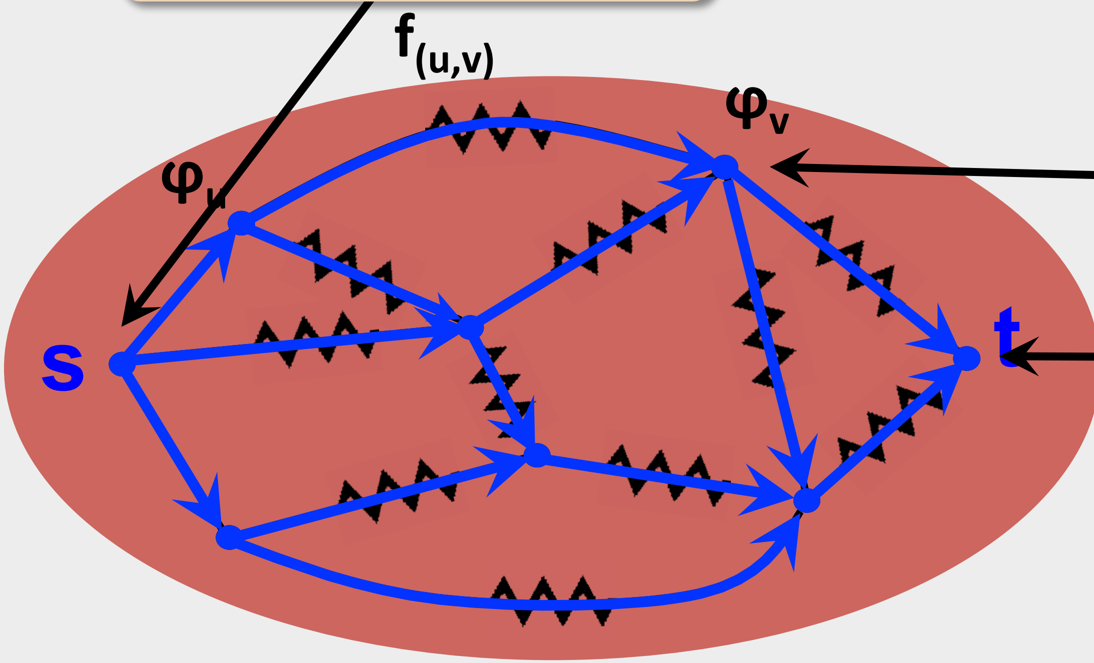


(Another) recipe for electrical flow (of value F):

Ele… (Take II)

Input: **Undirected** graph **G**,
resistances $\mathbf{r_e}$,
source s and **sink t**

deficit of **F** at **s**



no leaks at all **v≠s,t**

excess of **F** at **t**

(Another) recipe for electrical flow (of value F):

Find **vertex potentials** $\boldsymbol{\varphi_v}$ such that setting, for all **(u,v)**

$$f_{(u,v)} \leftarrow (\varphi_v - \varphi_u)/r_{(u,v)} \qquad \textbf{(Ohm's law)}$$

gives a **valid s-t flow** of **value F**

Electrical flows (Take III)

Input: **Undirected** graph **G**, **resistances r_e**, **source s** and **sink t**

Principle of least energy

Electrical flow of value F:
The unique minimizer of the **energy**

$$\mathbf{E}(f) = \Sigma_e\, r_e\, f(e)^2$$

among all **s-t** flows **f** of value **F**

Electrical flows = ℓ_2-minimization

How to compute an electrical flow?

Input: Graph **G=(V,E)**, **resistances r_e**, **source s** and **sink t**, **value F=1**

Solve a linear system!

Wlog as elect. flow are invariant under scaling

How to compute an electrical flow?

Input: Graph **G=(V,E)**, **resistances r_e**, **source s** and **sink t**, **value F=1**

Solve a linear system!

Observe: It suffices to compute **vertex potentials φ_v**

Ohm's law: If $\boldsymbol{\varphi}$ is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{B}^{T}\boldsymbol{\varphi}$$

is the corresponding flow

Here:

→ **f** is an **|E|**-dim vector with **$|f_e|$** giving the amount of flow on **e** and **sign(f_e)** encoding its direction (wrt edge orientation)

→ **R** is an **|E|×|E| diagonal** matrix with $\mathbf{R_{ee} = r_e}$

→ **B** is an **|V|×|E|** matrix with **e-th** column, for **e=(v,u)**, having **-1** (resp. **+1**) at its **v-th** (resp. **u-th**) coordinate and **0** everywhere else

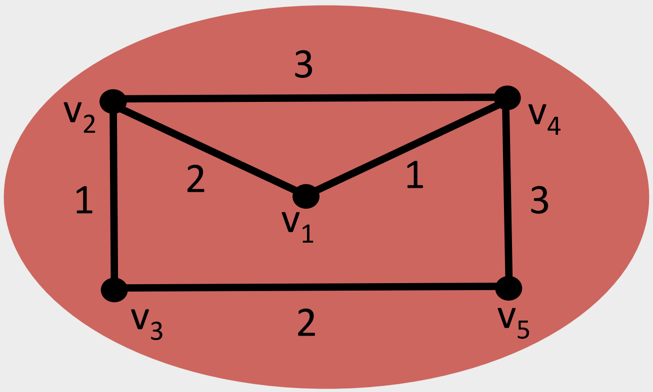
How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f = R^{-1}B^T \varphi}$$

is the corresponding flow

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **i<j**

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

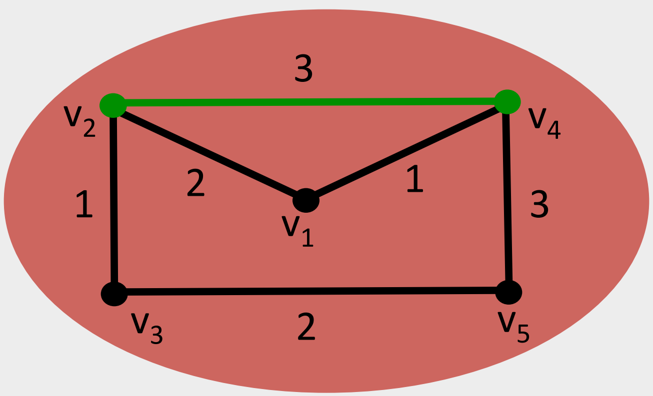
How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{B}^{T}\,\boldsymbol{\varphi}$$

is the corresponding flow

Example:



$$\mathbf{B} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **$i<j$**

$$\mathbf{R} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

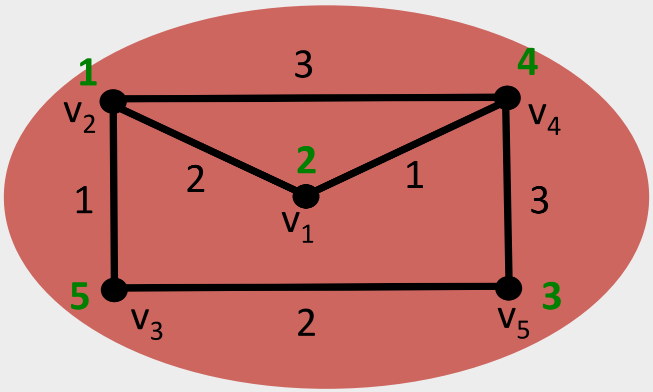
How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f = R^{-1}B^{T}\,\varphi}$$

is the corresponding flow

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **i<j**

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

$$\varphi = \begin{bmatrix} 2 \\ 1 \\ 5 \\ 4 \\ 3 \end{bmatrix}$$

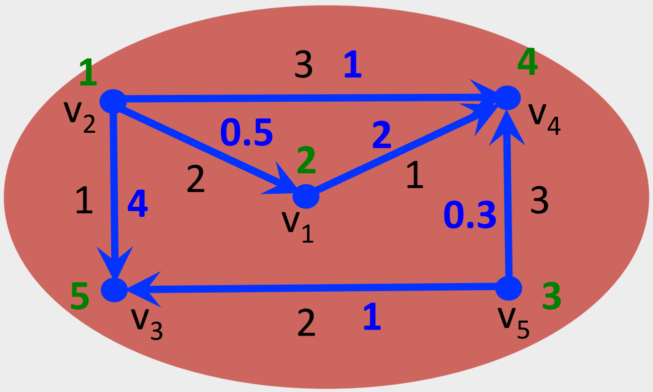
How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{B}^{T}\,\boldsymbol{\varphi}$$

is the corresponding flow

Example:



$$\mathbf{B} = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **i<j**

$$\boldsymbol{\varphi} = \begin{bmatrix} 2 \\ 1 \\ 5 \\ 4 \\ 3 \end{bmatrix} \xrightarrow{\mathbf{R}^{-1}\mathbf{B}^{T}} \mathbf{f} = \begin{bmatrix} -0.5 \\ 2 \\ 4 \\ 1 \\ -1 \\ -0.3 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f} = \mathbf{R}^{-1}\mathbf{B}^{T}\,\boldsymbol{\varphi}$$

is the corresponding flow

Recall: **φ** induces an electrical flow **f** iff

f is a valid **s-t** flow
(i.e., satisfies flow conservation constraints)

Equivalently: **φ** induces an electrical flow **f** iff

$$\mathbf{B}\,\mathbf{f} = \boldsymbol{\chi}_{s,t}$$

where $\boldsymbol{\chi}_{s,t}$ has a **1** at **t**, **-1** at **s** and **0**s everywhere else

Note: $(\mathbf{Bf})_v$ is the excess/deficit of **f** at **v**

How to compute an electrical flow?

Ohm's law: If **φ** is an (**|V|-dim**) vector of **vertex potentials** then

$$\mathbf{f} = R^{-1}B^{T}\varphi$$

is the corresponding flow

Recall: **φ** induces an electrical flow **f** iff

f is a valid **s-t** flow
(i.e., satisfies flow conservation constraints)

Equivalently: **φ** induces an electrical flow **f** iff

$$B\,\mathbf{f} = \chi_{s,t}$$

where $\chi_{s,t}$ has a **1** at **t**, **-1** at **s** and **0**s everywhere else

Note: $(B\mathbf{f})_v$ is the excess/deficit of **f** at **v**

Putting it together: **φ** induces an electrical flow iff

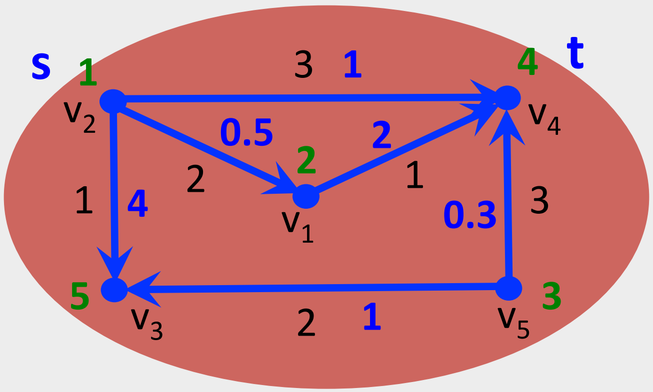
$$B\,R^{-1}B^{T}\varphi = \chi_{s,t}$$

How to compute an electrical flow?

Putting it together: φ induces an electrical flow iff

$$B\,R^{-1}B^T\varphi = \chi_{s,t}$$

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$|V|=5$, $|E|=6$, all edges oriented **(v_i,v_j)** with **$i<j$**

$$\varphi = \begin{bmatrix} 2 \\ 1 \\ 5 \\ 4 \\ 3 \end{bmatrix} \xrightarrow{R^{-1}B^T} f = \begin{bmatrix} -0.5 \\ 2 \\ 4 \\ 1 \\ -1 \\ -0.3 \end{bmatrix}$$

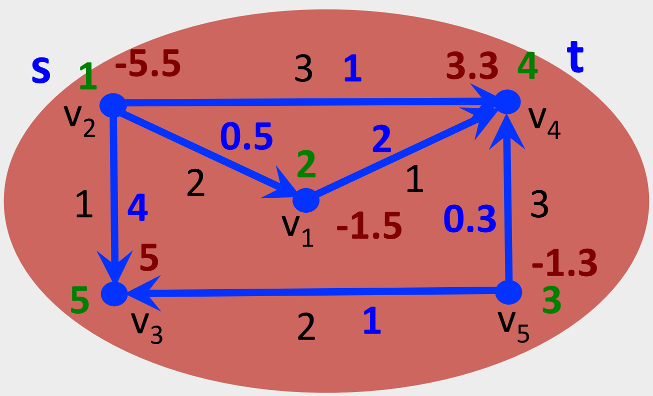
$$R = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

How to compute an electrical flow?

Putting it together: φ induces an electrical flow iff

$$B\,R^{-1}B^{T}\,\varphi = \chi_{s,t}$$

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **i<j**

$$\varphi = \begin{bmatrix} 2 \\ 1 \\ 5 \\ 4 \\ 3 \end{bmatrix} \xrightarrow{R^{-1}B^{T}} f = \begin{bmatrix} -0.5 \\ 2 \\ 4 \\ 1 \\ -1 \\ -0.3 \end{bmatrix} \xrightarrow{B} \begin{bmatrix} -1.5 \\ -5.5 \\ 5 \\ 3.3 \\ -1.3 \end{bmatrix} \neq \chi_{s,t} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

✗

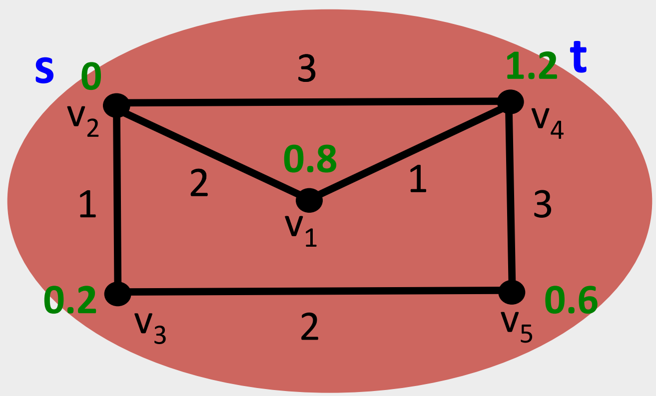
$$R = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

How to compute an electrical flow?

Putting it together: φ induces an electrical flow iff

$$\mathbf{B\, R^{-1} B^{T}\, \varphi = \chi_{s,t}}$$

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented $(\mathbf{v_i},\mathbf{v_j})$ with **i<j**

$$\varphi = \begin{bmatrix} 0.8 \\ 0 \\ 0.2 \\ 1.2 \\ 0.6 \end{bmatrix}$$

$$\chi_{s,t} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

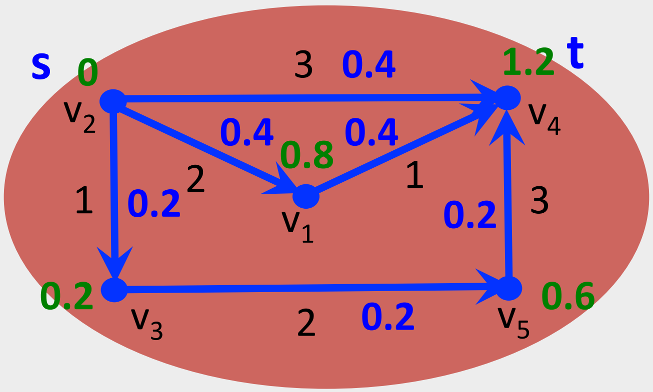
$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

How to compute an electrical flow?

Putting it together: φ induces an electrical flow iff

$$\mathbf{B\,R^{-1}B^{T}\,\varphi = \chi_{s,t}}$$

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented (v_i, v_j) with **i<j**

$$\varphi = \begin{bmatrix} 0.8 \\ 0 \\ 0.2 \\ 1.2 \\ 0.6 \end{bmatrix} \xrightarrow{R^{-1}B^{T}} f = \begin{bmatrix} -0.4 \\ 0.4 \\ 0.4 \\ 0.2 \\ 0.2 \\ -0.2 \end{bmatrix}$$

$$\chi_{s,t} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

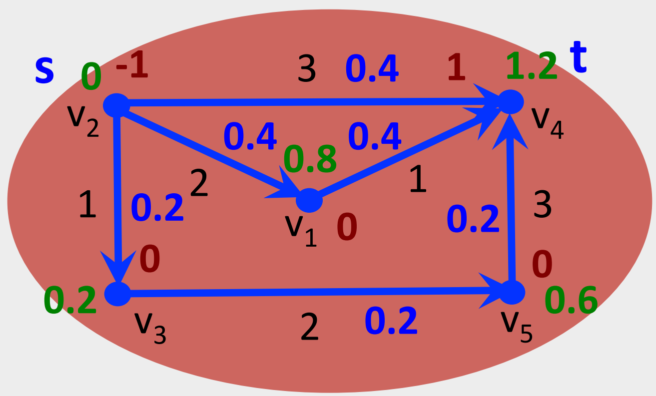
$$R = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

How to compute an electrical flow?

Putting it together: φ induces an electrical flow iff

$$B\,R^{-1}B^{T}\,\varphi = \chi_{s,t}$$

Example:



$$B = \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

|V|=5, |E|=6, all edges oriented **(v_i,v_j)** with **i<j**

$$\varphi = \begin{bmatrix} 0.8 \\ 0 \\ 0.2 \\ 1.2 \\ 0.6 \end{bmatrix} \xrightarrow{R^{-1}B^{T}} f = \begin{bmatrix} -0.4 \\ 0.4 \\ 0.4 \\ 0.2 \\ 0.2 \\ -0.2 \end{bmatrix} \xrightarrow{B} \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \chi_{s,t}$$

✔

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

How to compute an electrical flow?

Bottom line:

Electrical flow computation

$$BR^{-1}B^T \; \varphi = \chi_{s,t}$$

Solving a **linear system**

Bad news: Solving a linear system can take $\mathbf{O(n^{\omega})=O(n^{2.373})}$ (Prohibitive!)

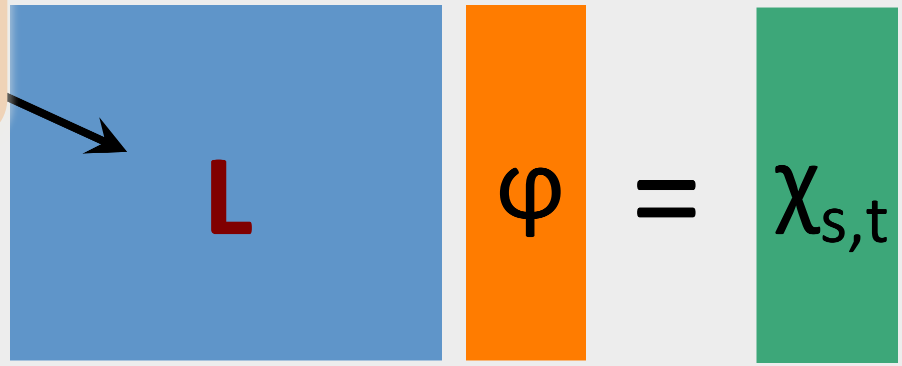
Key observation:
$\mathbf{BR^{-1}B^T}$ is the **Laplacian** matrix **L** of the underlying graph

How to compute an electrical flow?

Bot

Laplacian = key object of spectral graph theory (will get back to this)

Electrical flow computation



Solving a **Laplacian system**

Bad news: Solving a linear system can take $\mathbf{O(n^{\omega})=O(n^{2.373})}$

(Prohibitive!)

Key observation:
$\mathbf{BR^{-1}B^{T}}$ is the **Laplacian** matrix **L**
of the underlying graph

How to compute an electrical flow?

Bottom line:

Electrical flow computation

$$L \varphi = \chi_{s,t}$$

Solving a **Laplacian system**

Bad news: Solving a linear system can take $\mathbf{O(n^{\omega})=O(n^{2.373})}$

(Prohibitive!)

Key observation:
$\mathbf{BR^{-1}B^{T}}$ is the **Laplacian** matrix **L** of the underlying graph

How to utilize it?

Result: Electrical flow is a **nearly-linear time** primitive

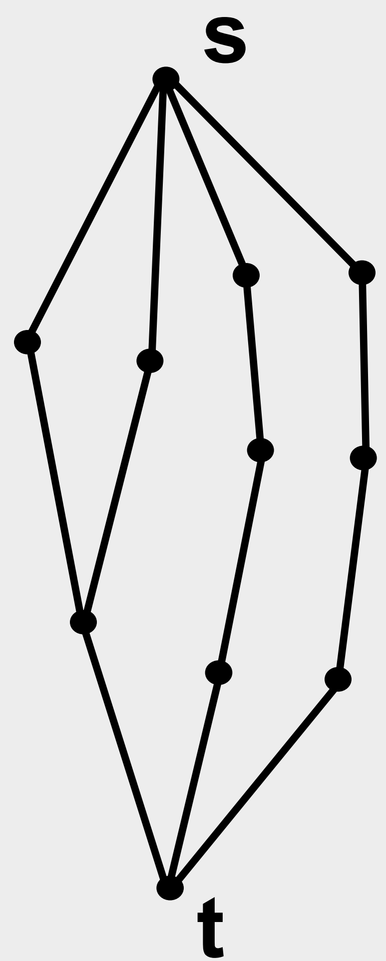
From electrical flows to undirected max flow

[CKMST ’11]

Approx. undirected max flow via electrical flows

Assume: $\mathbf{F}^*$ known (via binary search)

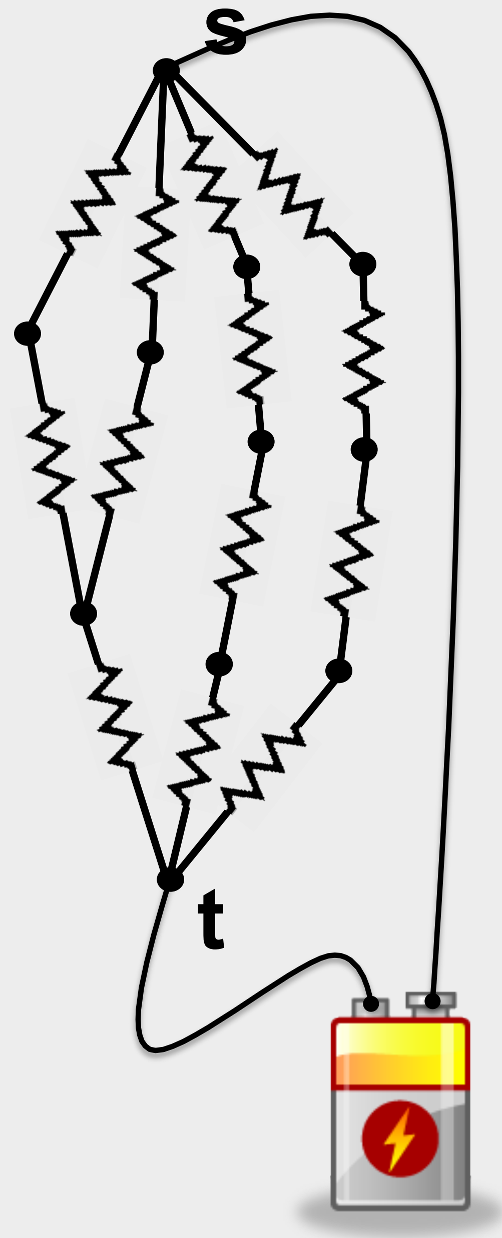
→ Treat edges as resistors of resistance **1**



Approx. undirected max flow via electrical flows

Assume: $\mathbf{F}^*$ known (via binary search)

→ Treat edges as resistors of resistance **1**
→ Compute electrical flow of value $\mathbf{F}^*$



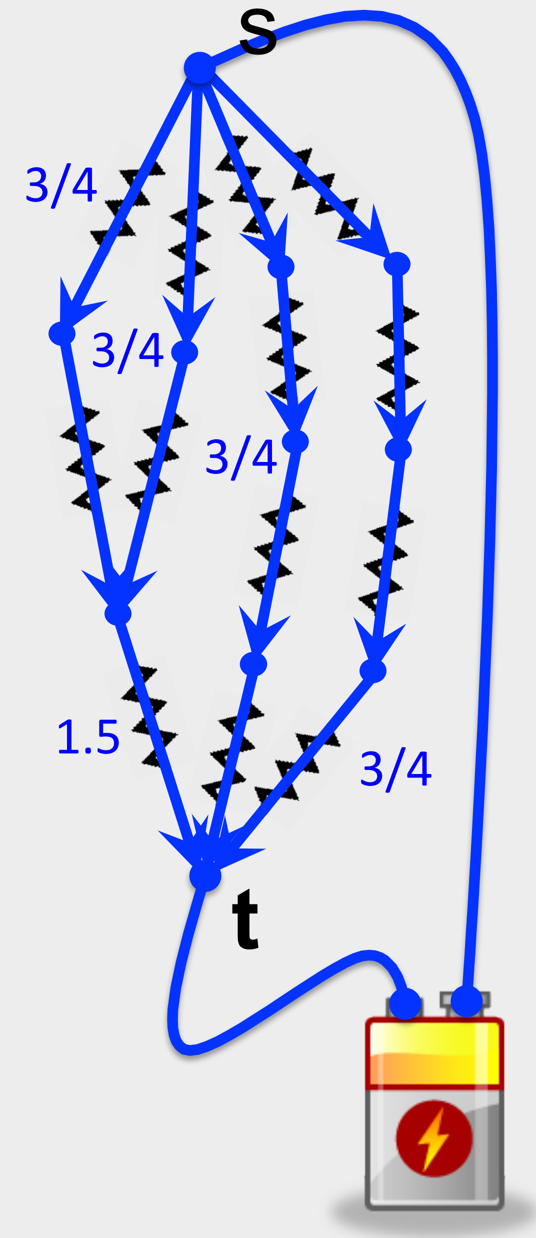
Approx. undirected max flow via electrical flows

Assume: F* known (via binary search)

→ Treat edges as resistors of resistance **1**

→ Compute electrical flow of value **F***

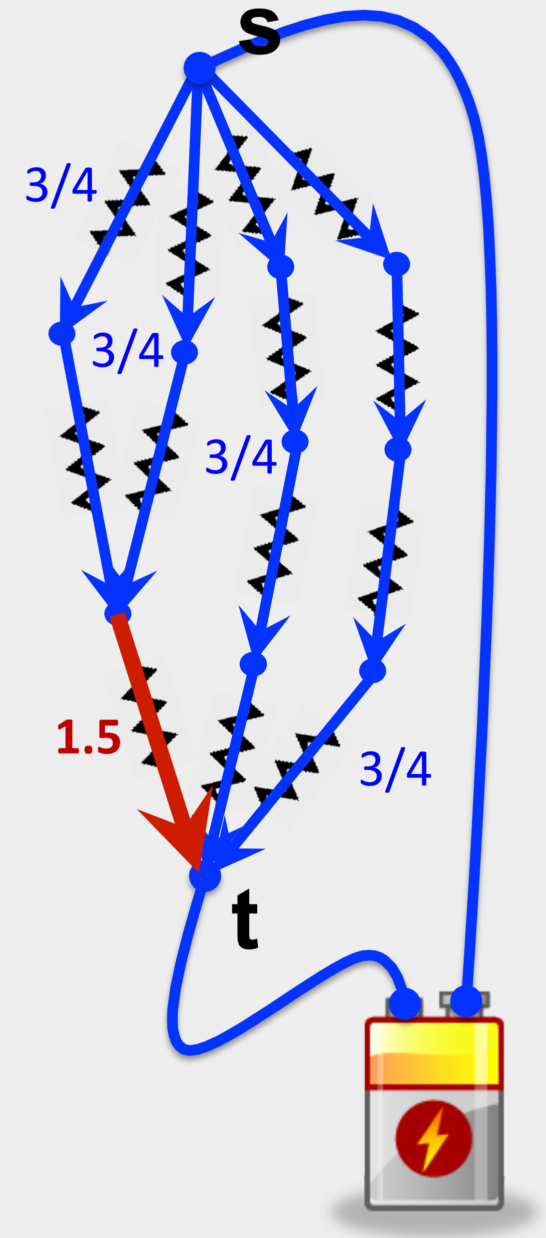
(This flow has **no leaks**, but **can overflow** some edges)



Approx. undirected max flow via electrical flows

Assume: F^* known (via binary search)

→ Treat edges as resistors of resistance **1**
→ Compute electrical flow of value $\mathbf{F^*}$
(This flow has **no leaks**, but **can overflow** some edges)
→ To fix that: **Increase resistances** on the overflowing edges
Repeat (**hope:** it doesn't happen too often)



Surprisingly: This approach can be made work!

Tomorrow: Will discuss how to fill the blanks

Addendum: A Glimpse of Spectral Graph Theory

Spectral graph theory: Understanding graphs via eigenvalues and eigenvectors of associated matrices

Central object: **Laplacian** matrix of a graph **G=(V,E,w)**

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

Diagonal **degree** matrix
$D_{vv} = \deg(v) = \Sigma_u w_{uv}$

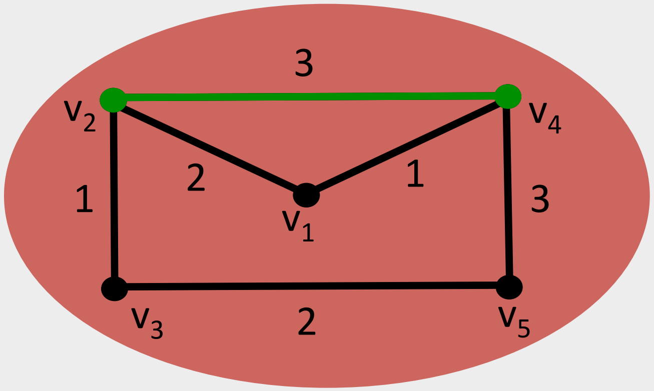
Adjacency matrix
$A_{uv} = w_{uv}$

Equivalently:

$$L_{uv} = \begin{cases} -w_{uv} & \text{if } (u,v) \text{ in } E \\ \deg(v) & \text{if } u=v \\ 0 & \text{otherwise} \end{cases}$$

Spectral graph theory: Understanding graphs via eigenvalues and eigenvectors of associated matrices

Example:



$$L = \begin{bmatrix} 3 & -2 & 0 & -1 & 0 \\ -2 & \boxed{6} & -1 & \boxed{-3} & 0 \\ 0 & -1 & 3 & 0 & -2 \\ -1 & \boxed{-3} & 0 & \boxed{7} & -3 \\ 0 & 0 & -2 & -3 & 5 \end{bmatrix}$$

Observe:

$$\mathbf{L = \Sigma_e\, w_e\, L^e}$$

Laplacian of a graph **(V, {e})**

Laplacian as a quadratic form:

$$x^T L\, x = \Sigma_e\, w_e\, x^T L^e x = \Sigma_e\, w_e\, (x_u - x_v)^2$$

Spectrum of a Laplacian

Laplacian is an **n×n symmetric** matrix.

→ It has as **n real eigenvalues** $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ with corresponding (orthogonal) **eigenvectors** $v^1, v^2, \ldots, v^n$ s.t.

$$L v^i = \lambda_i v^i$$

Can show: $\lambda_1 = 0$ and $v^1 = (1, \ldots, 1)$

These objects tell us a lot about the graph!
(And we can compute each λ_i and v^i in **nearly-linear time**)

Most important eigenvalue: λ_2

λ_2 and graph connectivity

Fact: λ_2**=0** iff **G** is disconnected

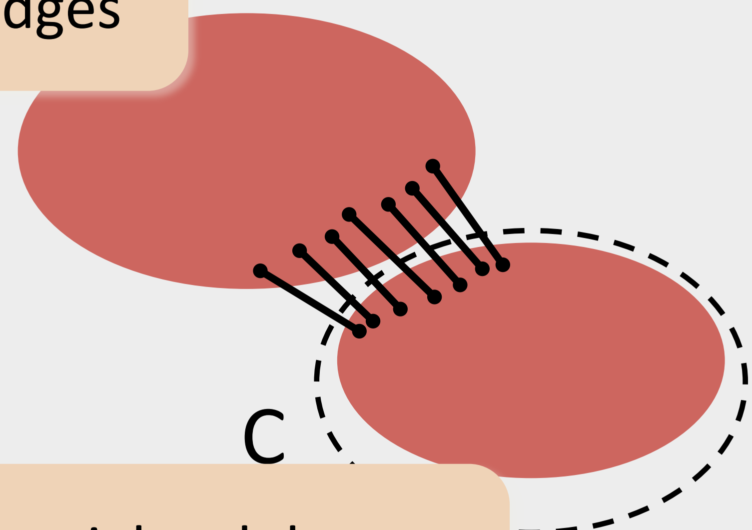
(**More generally:** λ_k**=0** iff **G** has at least **k** connected components)

Can we make this connect… e?

Cut conductance:

Total weight of cut edges

$$\Phi(C) = \frac{w(C)}{deg(C)}$$



Total weighted degree
(of the "smaller" side)

λ_2 and graph connectivity

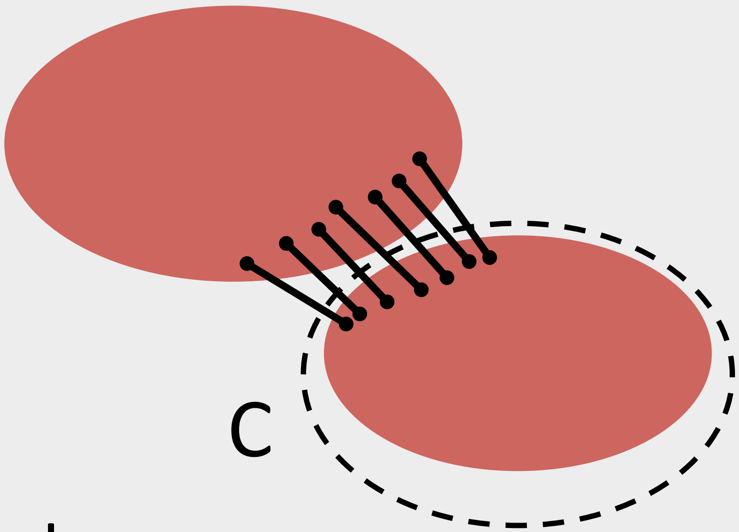
Fact: λ_2**=0** iff **G** is disconnected

(**More generally:** λ_k**=0** iff **G** has at least **k** connected components)

Can we make this connection quantitative?

Graph conductance:

$$\Phi_G = \min_C \frac{w(C)}{\deg(C)}$$



Φ_G **large** → **G** is well connected

Φ_G **small** → **G** has a "bottlenecking" cut

λ_2 and graph connectivity

For a **normalized** Laplacian $\mathcal{L}$ **= $\mathbf{D^{-1/2}\, L\, D^{-1/2}}$**

$$\lambda_2/2 \le \Phi_G \le 2\, \lambda_2^{1/2}$$

[Cheeger '70, Alon-Milman '85]

λ_2 and graph connectivity

For a **normalized** Laplacian $\mathcal{L}$ **= D$^{-½}$ L D$^{-½}$**

$$\lambda_2/2 \leq \Phi_G \leq 2\,\lambda_2^{½}$$

[Cheeger '70, Alon-Milman '85]

A cut **C** with **$\Phi(C) \leq 2\,\lambda_2^{½}$** can be found in **nearly-linear time**

→ Gives an **$O(\lambda_2^{-½})$-approx.** to **Φ_G**
(Computing **Φ_G** is **NP-hard**)

→ Great when **λ_2** is large, i.e., **G** is well-connected,
but pretty poor for small **λ_2**

λ_2 and graph connectivity

For a **normalized** Laplacian $\mathcal{L} = D^{-1/2} L D^{-1/2}$

$$\lambda_2/2 \le \Phi_G \le 2\,\lambda_2^{1/2}$$

[Cheeger '70, Alon-Milman '85]

A cut **C** with $\Phi(C) \le 2\,\lambda_2^{1/2}$ can be found in **nearly-linear time**

→ Gives an **$O(\lambda_2^{-1/2})$-approx.** to Φ_G
(Computing Φ_G is **NP-hard**)

→ Great when λ_2 is large, i.e., **G** is well-connected,
but pretty poor for small λ_2

Unfortunately: The $\lambda_2^{1/2}$ **vs.** λ_2 gap is unavoidable

λ_2 and random walks

Paint spilling process:

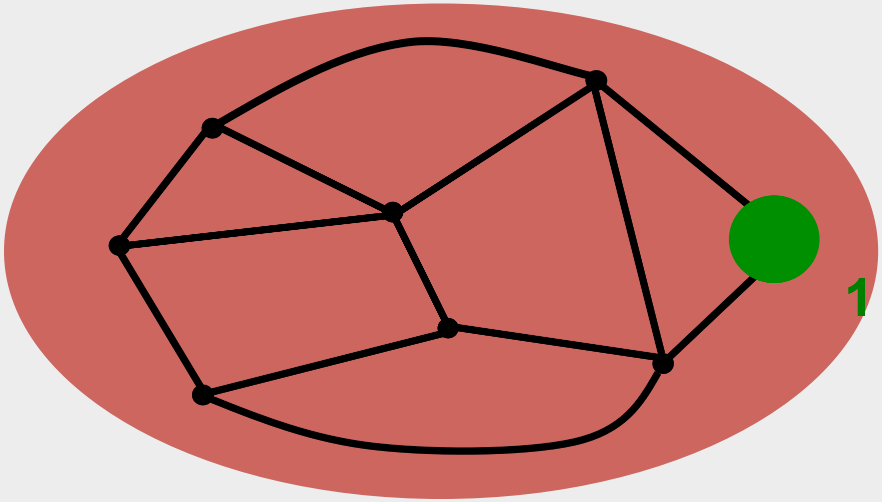
→ **Start with all paint at s**

→ **For each vertex:**

Split the paint in half:

– one half stays put

– distribute the rest (evenly) among the neighbors



λ_2 and random walks

Paint spilling process:

→ **Start with all paint at s**

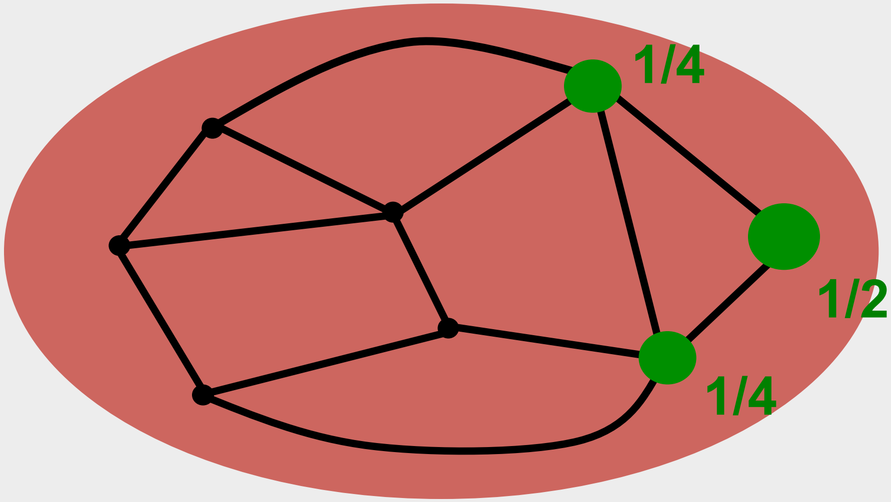
→ **For each vertex:**

Split the paint in half:

– one half stays put

– distribute the rest (evenly) among the neighbors

→ **Repeat**



λ_2 and random walks

Paint spilling process:

→ **Start with all paint at s**

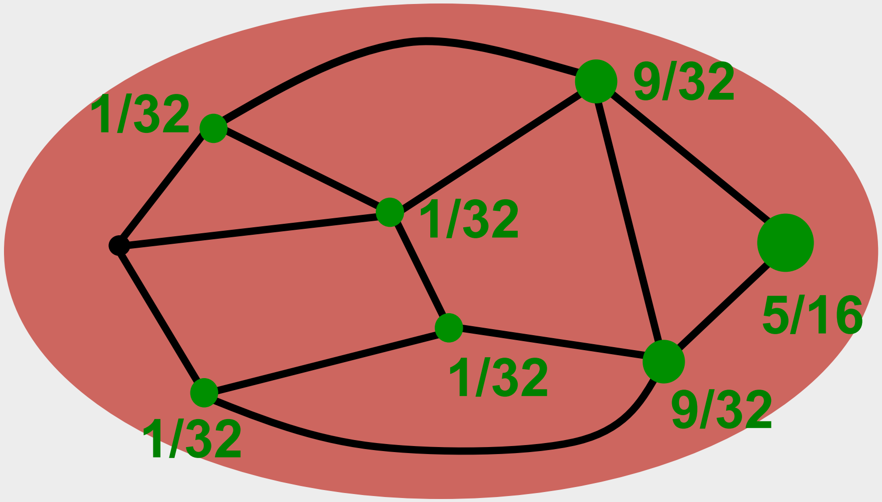
→ **For each vertex:**

Split the paint in half:

– one half stays put

– distribute the rest (evenly) among the neighbors

→ **Repeat**



This diffusive process corresponds to a **(lazy) random walk** and shows up everywhere

Fact: Paint distribution **always*** converges to a **stationary distribution π** with $\pi_v \sim \deg(v)$

But: How fast is this convergence?

Theorem: The convergence rate is $\Theta(\lambda_2^{-1})$

Beyond λ_2?

→ Looking at the higher order eigenvalues

For any $\mathbf{k \geq 2}$,

$$\Phi_G \leq O(k)\, \lambda_2/\lambda_k^{1/2}$$

[LOT'12, KLLOT'13]

→ **Electrical graph theory:** Using electrical flows

Key quantity: Effective resistance (between **s** and **t**)

$$R_{st} = \chi_{st}^T L^+ \chi_{st}$$

Vector with **1** at **t**, **-1** at **s** and **0**s everywhere else

Pseudo-inverse of the Laplacian

Beyond λ_2?

→ Looking at the higher order eigenvalues

For any **k ≥ 2**,

$$\Phi_G \leq O(k)\, \lambda_2/\lambda_k^{1/2}$$

[LOT'12, KLLOT'13]

→ **Electrical graph theory:** Using electrical flows

Key quantity: Effective resistance (between **s** and **t**)

$$R_{st} = \chi_{st}^T L^+ \chi_{st}$$

Note: Effective resistance depends on the **whole** spectrum of **L**

[SS '08]: We can (approx.) compute **all** resistances in **nearly-linear time**

Electrical flows show up in many contexts:

- → Behavior of random walks (commute time, PageRank,…)
- → Graph sparsification
- → Sampling random spanning trees
- → **Maximum flow problem**

Where else can we use them?

Thank you

Tomorrow: Computing an approx. max flow in undirected graphs using electrical flows