# Interior-point Methods and the Maximum Flow Problem

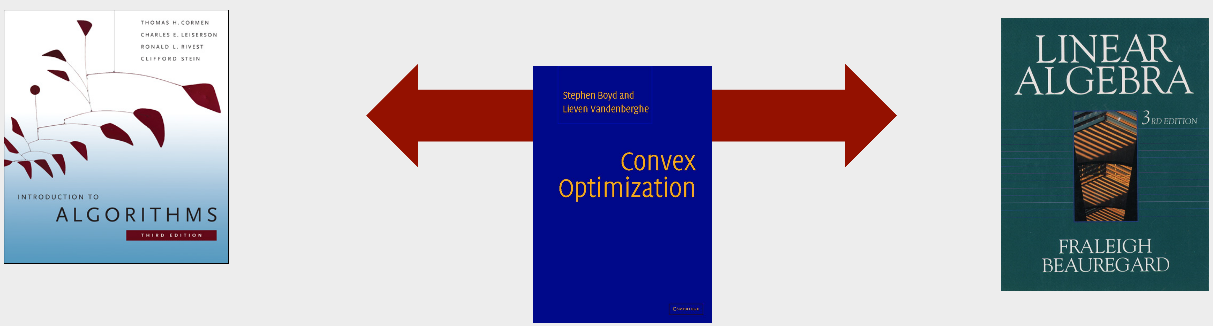

Aleksander Mądry

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

# What will this talk be about?

**At a first glance:** It is just a talk about recent progress on the maximum flow problem

**But also:** A "success story" of combining combinatorial alg., continuous optimization and linear-algebraic tools

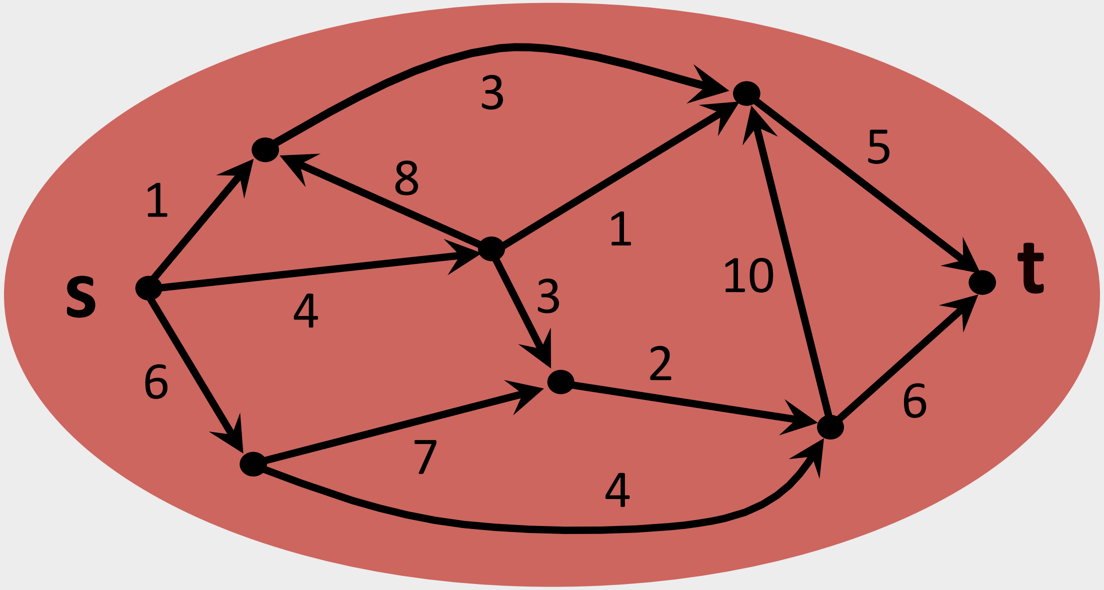



**Additionally:** An example where employing interior-point method (IPM) leads to very fast algorithms

**Bonus:** New(?) understanding of IPM's convergence

# Maximum flow problem

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

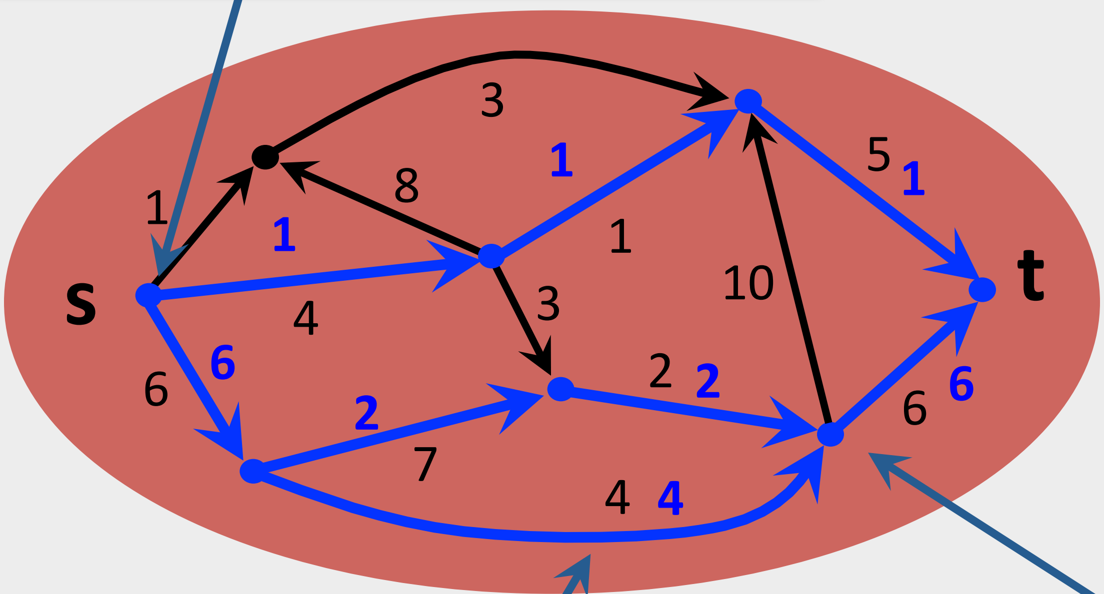



**Task:** Find a **feasible s-t flow** of **max value**

# Maximum flow problem

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

**value** = net flow out of **s**




Max flow value **F*=10**

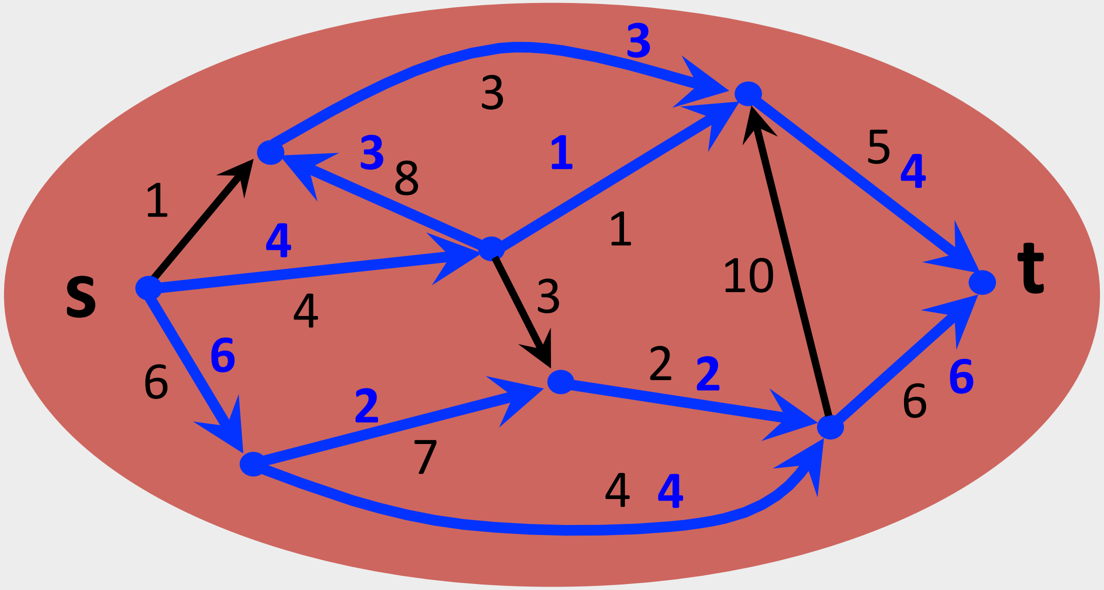
**no overflow** on arcs: **0 ≤ f(e) ≤ u(e)**

**no leaks** at all **v≠s,t**

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

# Maximum flow problem

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

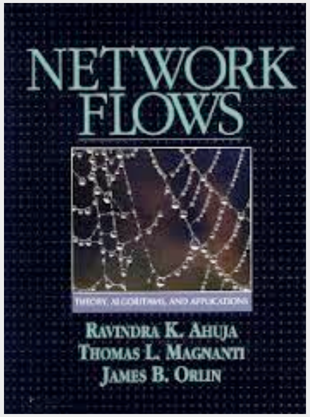



Max flow value
**F*=10**

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

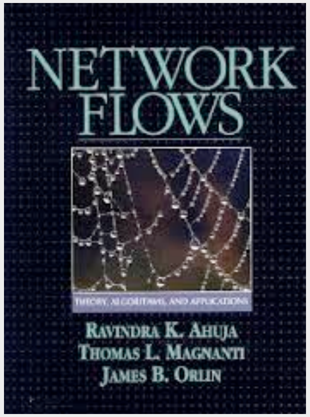
# What is known about Max Flow?

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]** | $O(mn \log U)$ |
| **[Goldberg Rao '98]** | $\tilde{O}(m \min(m^{1/2}, n^{2/3}) \log U)$ |
| **[Lee Sidford '14]** | $\tilde{O}(mn^{1/2} \log U)$ |

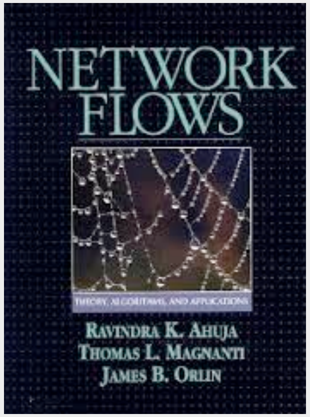
**Our focus:** Sparse graph (**$m=O(n)$**) and unit-capacity (**$U=1$**) regime

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

(**$n$** = # of vertices, **$m$** = # of arcs, **$U$** = max capacity, **$\tilde{O}()$** 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]** | $\tilde{O}(n^2)$ |
| **[Goldberg Rao '98]** | $\tilde{O}(n^{3/2})$ |
| **[Lee Sidford '14]** | $\tilde{O}(n^{3/2})$ |

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

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

(**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)

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

**[LSR '13, S '13, KLOS '14, P '14]: (1-ε)-approx.** in $\tilde{O}(n\varepsilon^{-2})$ time

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

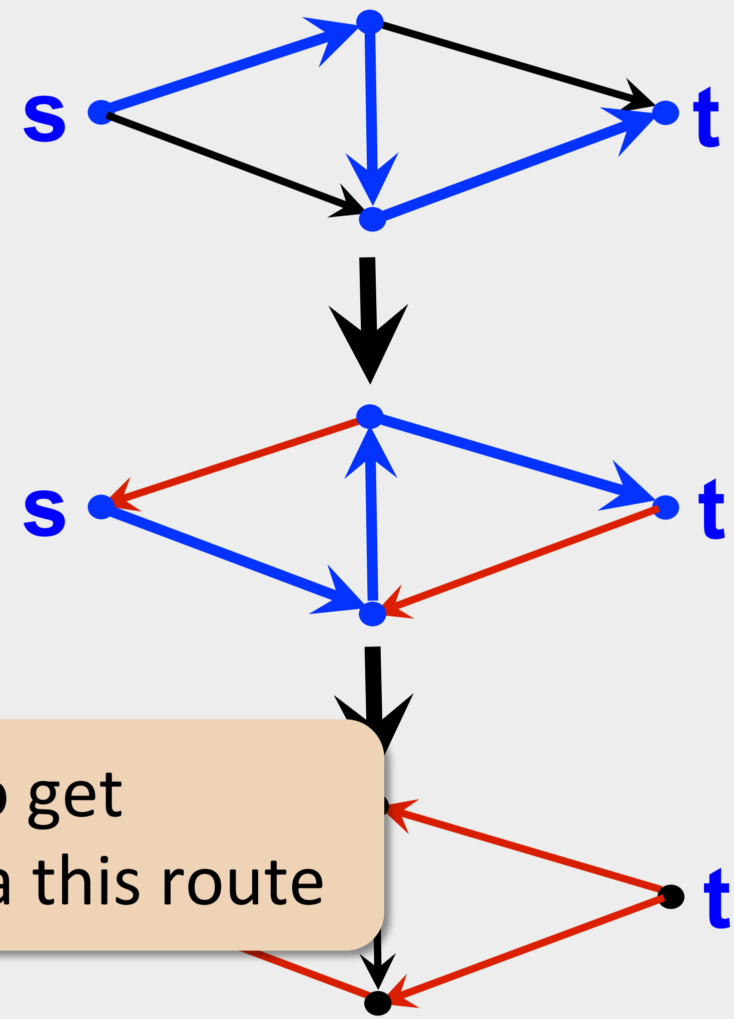
($n$ = # of vertices, $\tilde{O}()$ hides polylog factors)

# Previous approach

# Augmenting paths framework

[Ford Fulkerson '56]

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




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

**Problem:** Very difficult to analyze

**Naïve impl**

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

**Sophisticat**
**and arguments:** $O(n^{3/2})$ time

[Karzanov '73] [Even Tarjan '75]

# Beyond augmenting paths

**New approach:**

Bring linear-algebraic techniques into play

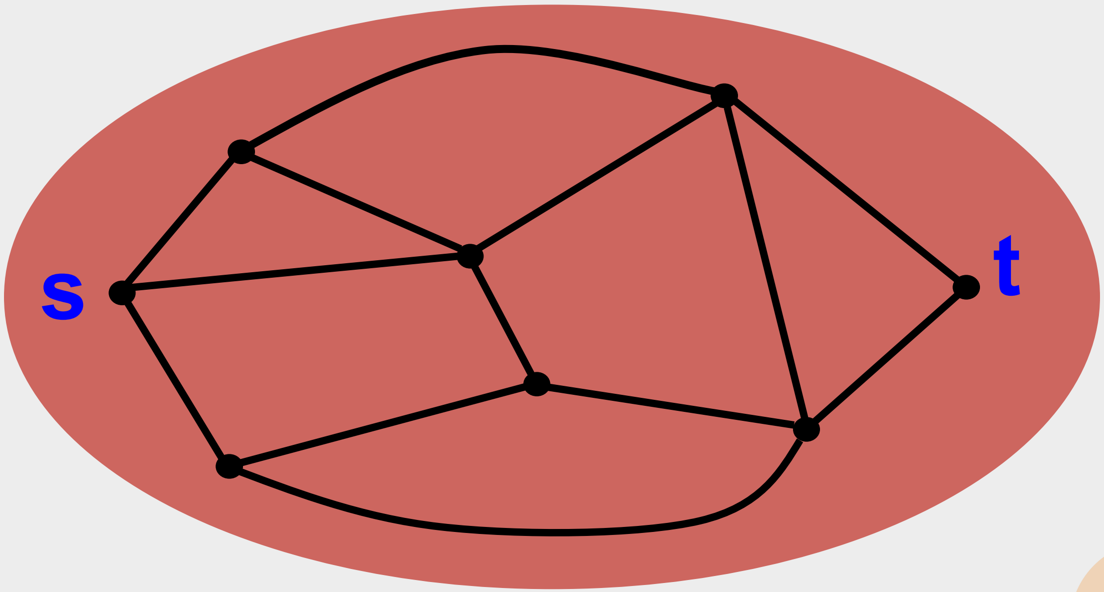
**Idea:** Probe the **global flow structure** of the graph by **solving linear systems**

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**

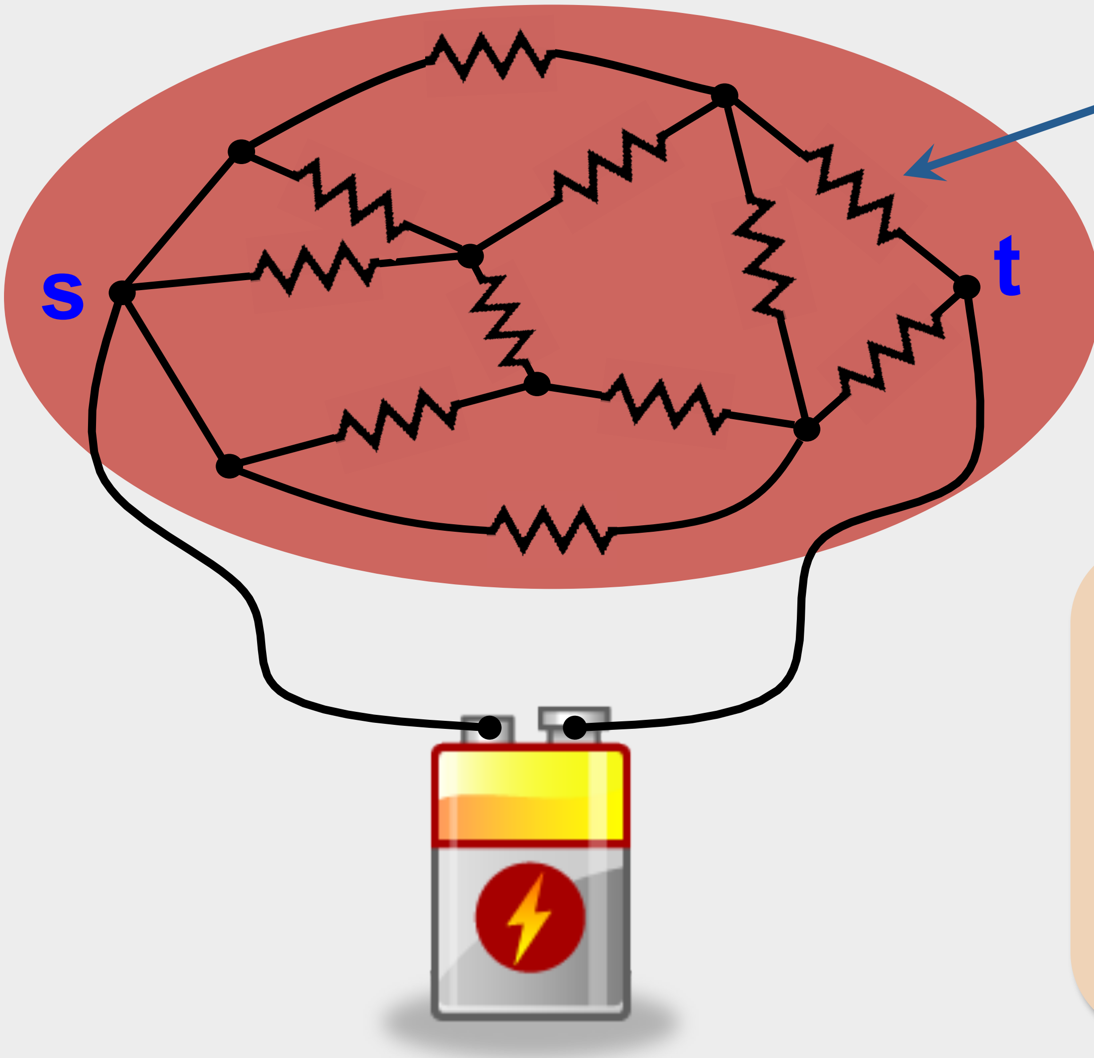



**Recipe for elec. flow:**

**1)** Treat edges as **resistors**

# Electrical flows (Take I)

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




**resistance** $\mathbf{r_e}$

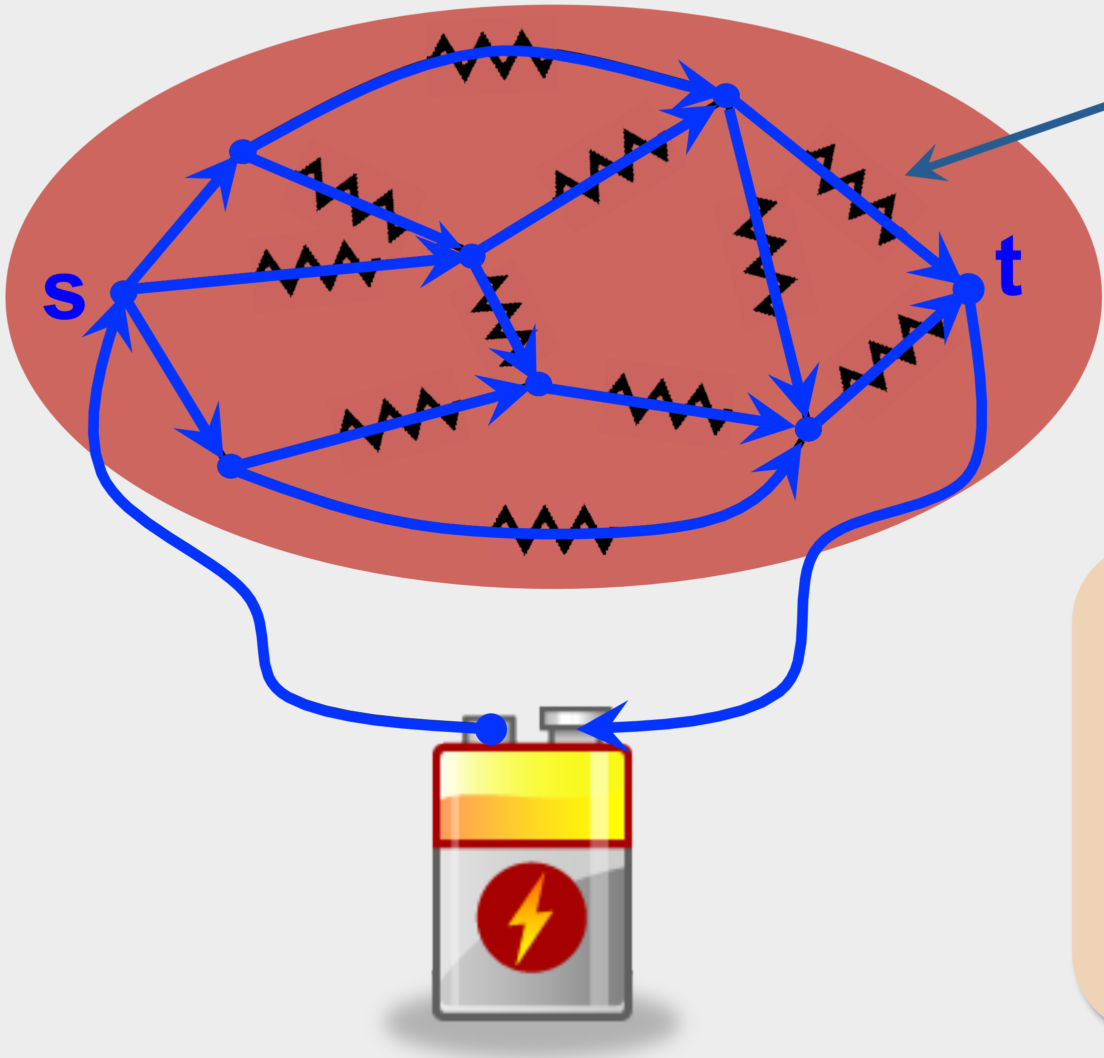
**Recipe for elec. flow:**

**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** $\mathbf{r_e}$, **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 = $\ell_2$-minimization

# How to compute an electrical flow?

Solve a linear system!

# How to compute an electrical flow?

Solve a **Laplacian** system!

$$L \, x = b$$

**Result:** Electrical flow is a **nearly-linear time** primitive
**[ST ’04, KMP ’10, KMP ’11, KOSZ ’13, LS ’13, CKPPR ‘14]**

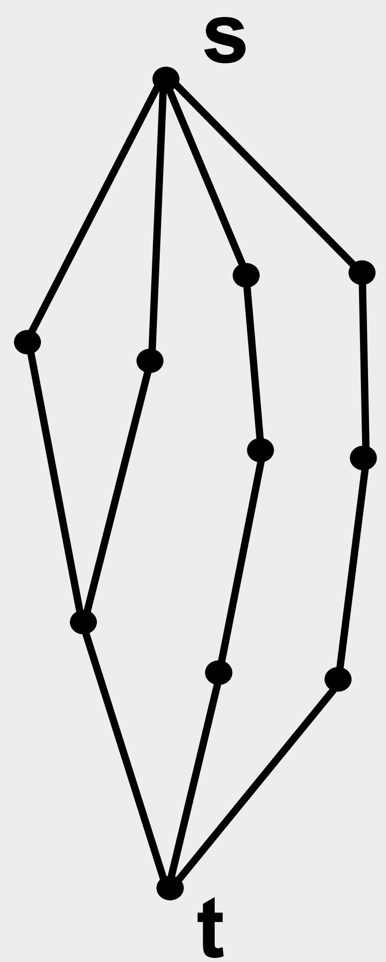
How to employ it?

# From electrical flows to undirected max flow

# Approx. undirected max flow via electrical flows

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

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

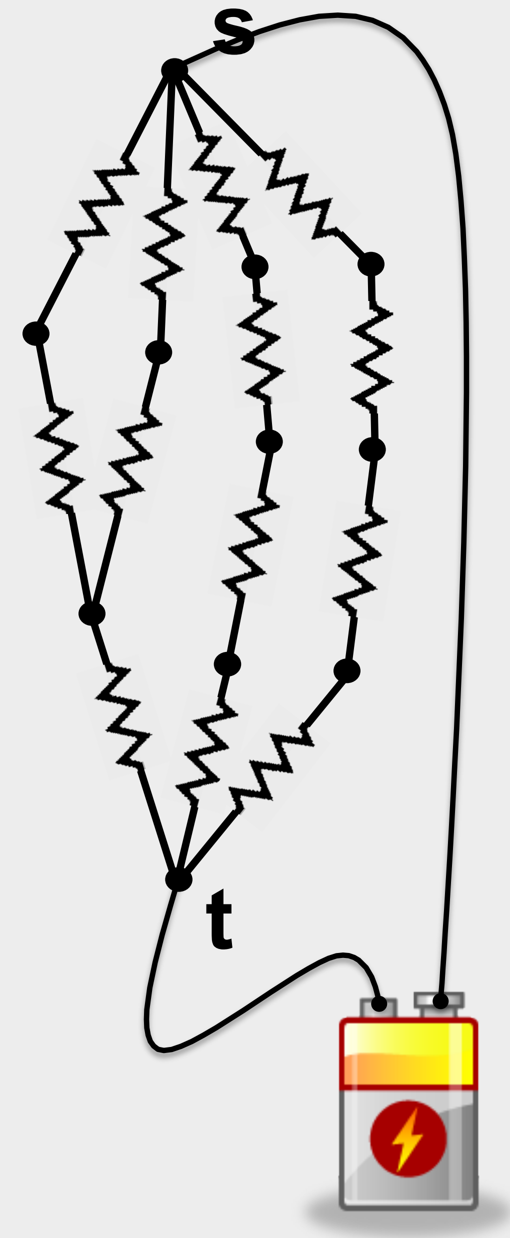


# Approx. undirected max flow via electrical flows

**Assume: $F^*$** known (via binary search)

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

→ Compute electrical flow of value $\mathbf{F^*}$



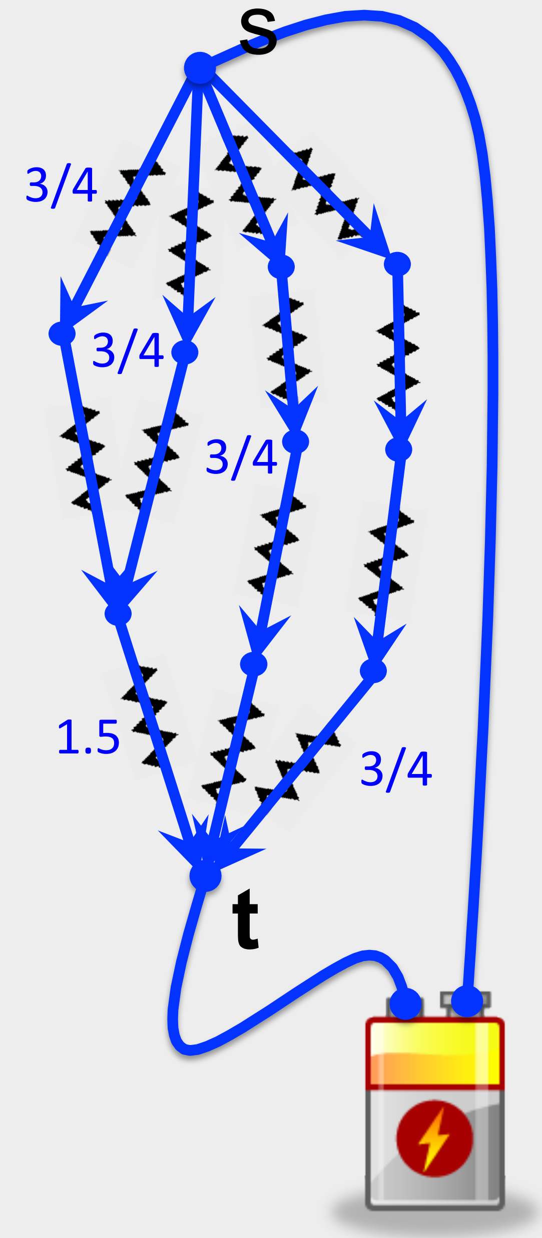
# Approx. undirected max flow via electrical flows

**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)

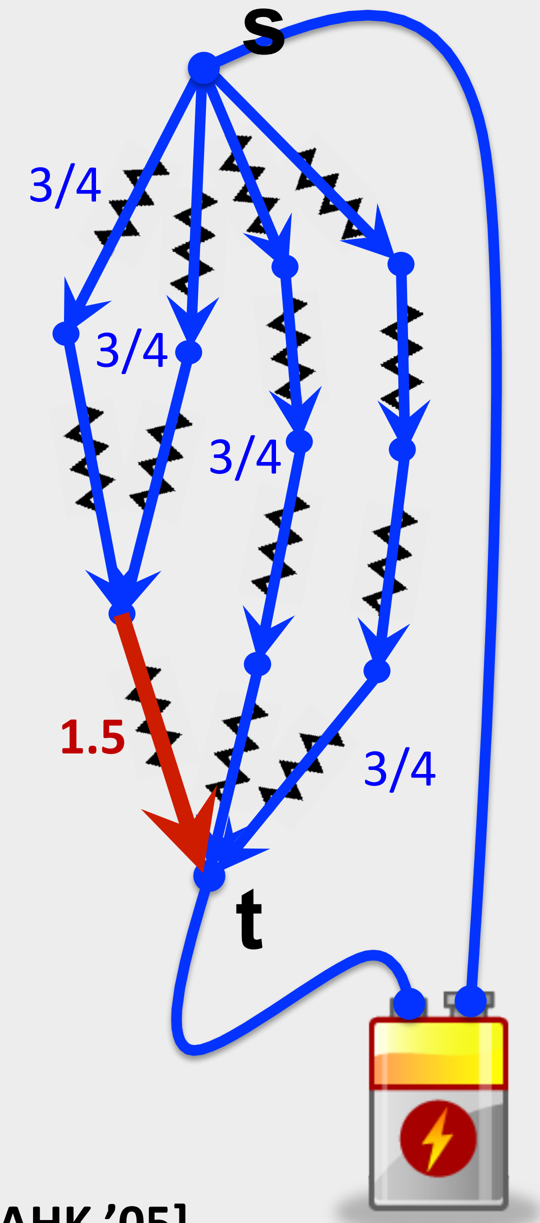


# Approx. undirected max flow via electrical flows

**Assume:** $\mathbf{F}^*$ 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
- → **At the end:** Take an **average** of all the flows as the final answer




**Evolution of resistances:**

Based on Multiplicative Weight Update method

**[FS ’97, PST ’95, AHK ’05]**

# Bounding the running time

→ Each iteration runs in **$\tilde{O}(n)$** time

→ How many iterations do we need?

**Can show:** # of iterations ≈ worst-case overflow **$\rho$**

**Think:** **$\rho$** measures the **electrical vs. max** flow difference

**Key question:** If **$f_E$ =** elect. flow of value **$F^*$** wrt all **$r_e$=1**

What is **$\rho = \max_e f_E(e)$**?

**Claim: $\rho \leq m^{1/2} = O(n^{1/2})$**

**Proof:** Suffices to show that

**$E(f_E) \leq m$**

# Bounding the running time

→ Each iteration runs in **$\tilde{O}(n)$** time

→ How many iterations do we need?

**Can show:** # of iterations ≈ worst-case overflow **$\rho$**

**Think: $\rho$** measures the **electrical vs. max** flow difference

**Key question:** If **$f_E$ =** elect. flow of value **$F^*$** wrt all **$r_e$=1**

What is **$\rho = \max_e f_E(e)$**?

**Claim: $\rho \leq m^{1/2} = O(n^{1/2})$**

**Proof:** Suffices to show that

$$E(f_E) = \Sigma_e\, r_e\, f_E(e)^2 = \Sigma_e\, f_E(e)^2 \leq m$$

**Note:** if **$f^*$** is the max flow (of value **$F^*$**) then

$$E(f^*) = \Sigma_e\, r_e\, f^*(e)^2 = \Sigma_e\, f^*(e)^2 \leq \Sigma_e 1 \leq m$$

# Bounding the running time

→ Each iteration runs in **$\tilde{O}(n)$** time

→ How many iterations do we need?

**Can show:** # of iterations ≈ worst-case overflow **$\rho$**

**Think:** **$\rho$** measures the **electrical vs. max** flow difference

**Key question:** If **$f_E$ =** elect. flow of value **$F^*$** wrt all **$r_e$=1**

What is **$\rho = \max_e f_E(e)$**?

**Claim: $\rho \le m^{1/2} = O(n^{1/2})$**

**Proof:** Suffices to show that

**$E(f_E) = \Sigma_e\, r_e\, f_E(e)^2 = \Sigma_e\, f_E(e)^2 \le m$**

**Note:** if **$f^*$** is the max flow (of value **$F^*$**) then

This gives an **$\tilde{O}(n\rho\varepsilon^{-3}) = \tilde{O}(n^{3/2}\,\varepsilon^{-3})$** time

**(1-ε)-approx** algorithm

# Breaking the $\Omega(n^{3/2})$ time bound

**Claim: $\rho \leq m^{1/2} = O(n^{1/2})$**

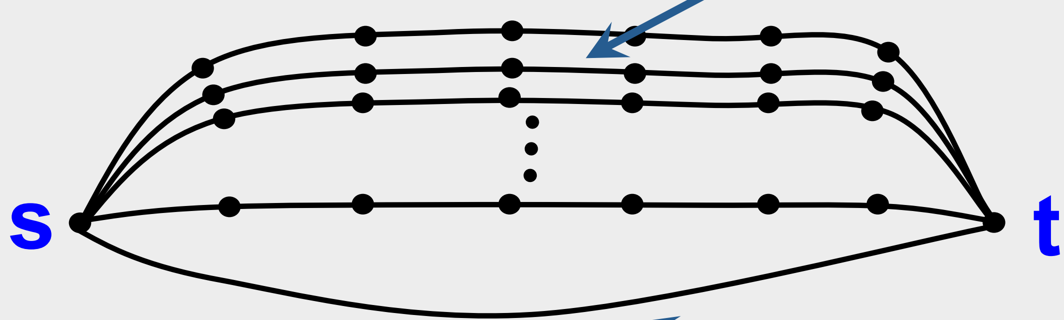
Is this bound tight?

Will be so **only** if there exists an edge that (single-handily) contributes **most of the energy** of $\mathbf{f_E}$

(**Recall:** We showed $\boldsymbol{\rho^2 = \max_e f_E(e)^2 \leq \Sigma_e f_E(e)^2 = E(f_E) \leq m}$)

Can this even happen?

$\approx$**n**$^{½}$ paths with $\approx$**n**$^{½}$ vertices each




one edge

# Breaking the $\Omega(n^{3/2})$ time bound

**Claim: $\rho \leq m^{1/2} = O(n^{1/2})$**

Is this bound tight?

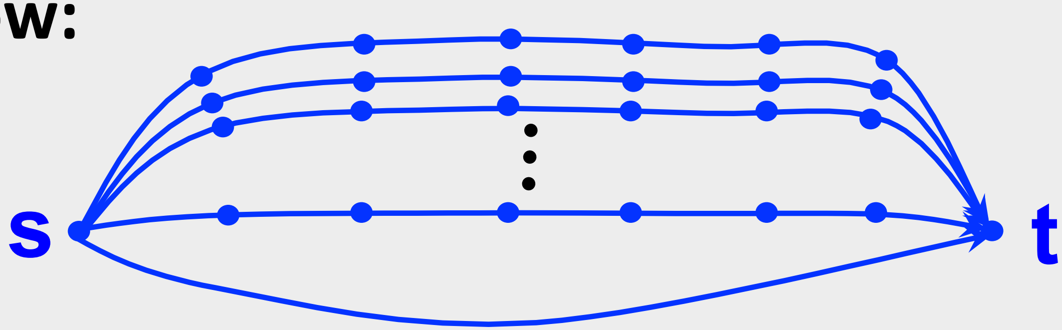
Will be so **only** if there exists an edge that (single-handily) contributes **most of the energy** of $\mathbf{f_E}$

(**Recall:** We showed $\boldsymbol{\rho^2 = \max_e f_E(e)^2 \leq \Sigma_e f_E(e)^2 = E(f_E) \leq m}$)

Can this even happen?

Unfortunately, yes

**Max flow:**




$\mathbf{F^* \approx n^{1/2}}$

# Breaking the $\Omega(n^{3/2})$ time bound

**Claim: $\rho \le m^{1/2} = O(n^{1/2})$**

Is this bound tight?

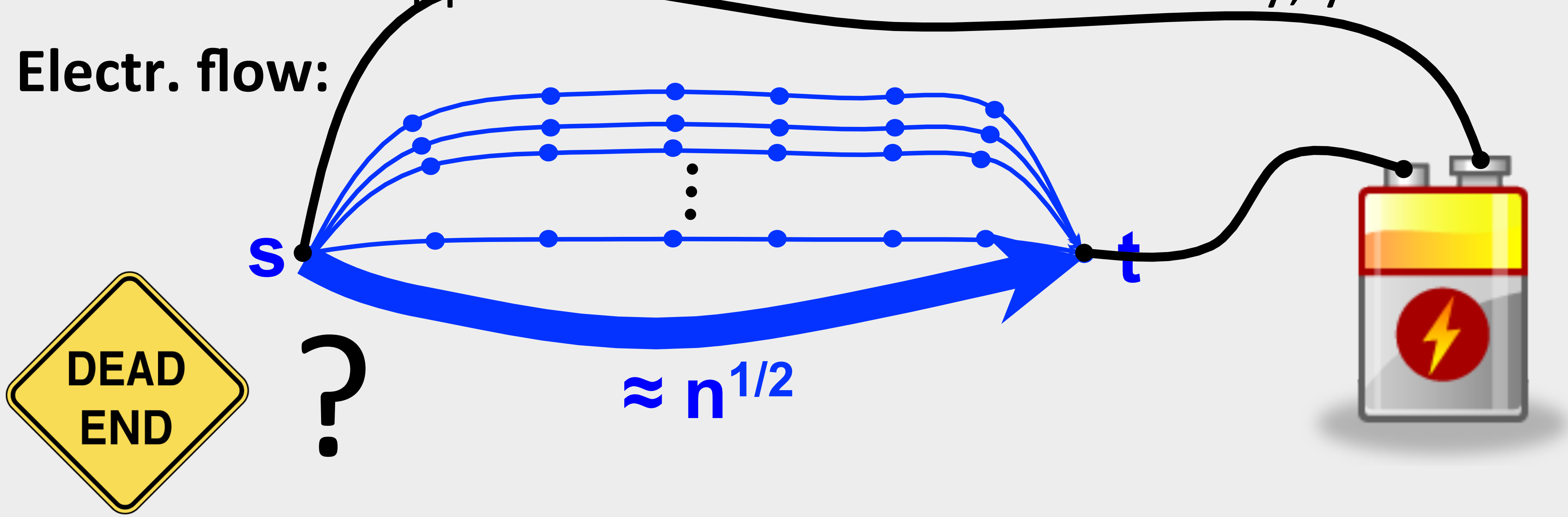
Will be so **only** if there exists an edge that (single-handily) contributes **most of the energy** of $\mathbf{f_E}$

(**Recall:** We showed $\rho^2 = \max_e f_E(e)^2 \le \Sigma_e f_E(e)^2 = E(f_E) \le m$)

Can this even happen?

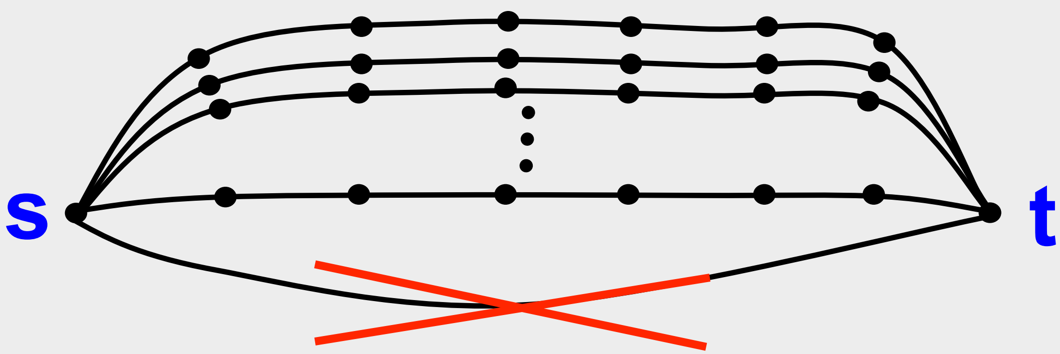
Unfortunately, yes

**Electr. flow:**



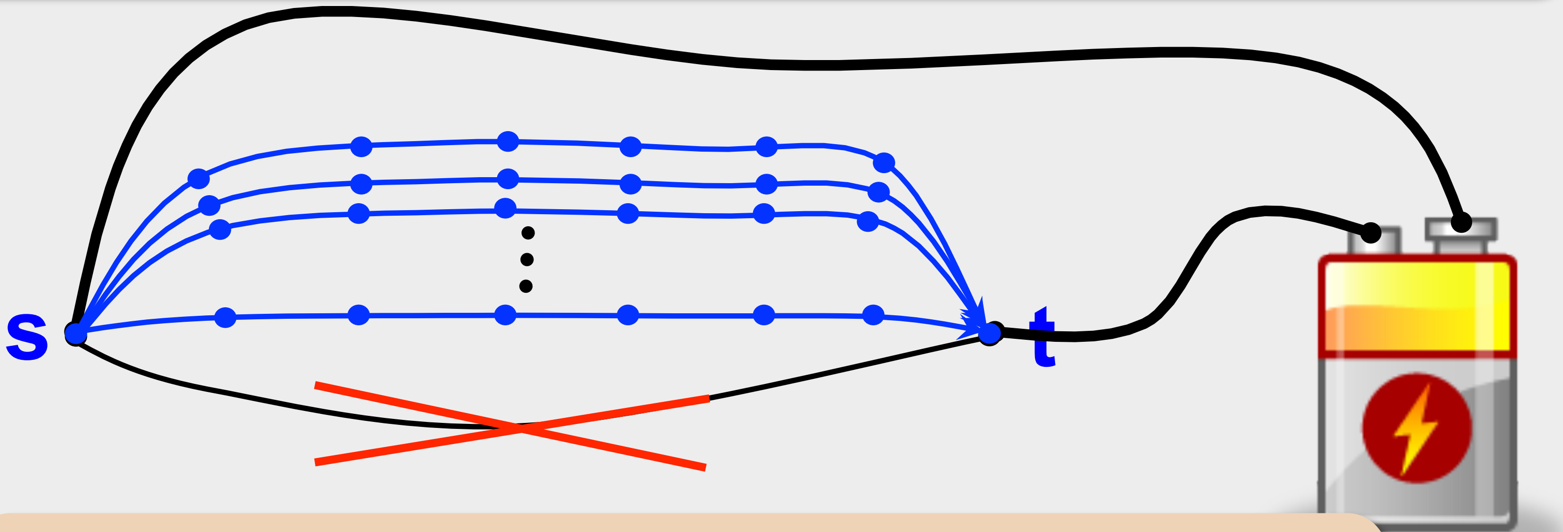
# Breaking the $\Omega(n^{3/2})$ time bound

**Key idea:** Perturb the graph by **removing such high-energy edges** whenever they emerge



# Breaking the $\Omega(n^{3/2})$ time bound

**Key idea:** Perturb the graph by **removing such high-energy edges** whenever they emerge




Careful energy-based argument gives the desired $\tilde{O}(n^{4/3}\,\varepsilon^{-3})$ time algorithm

**Later on:** **[LSR '13, S '13, KLOS '14, P'14]:** **(1-ε)-approx.** in $\tilde{O}(n\varepsilon^{-2})$ time via a version of an $\ell_\infty$-based gradient descent

# Directed Maximum Flow

Why the progress on **approx. undirected** max flow does not apply to the **directed** case?

**Key problem:** To solve **directed** max flow (even approx.), one needs to solve **exact undirected** max flow

First-order methods are inherently unable to deliver good enough accuracy here

We need a bigger hammer

# (Path-following) Interior-point method (IPM)

**[Dikin '67, Karmarkar '84, Renegar '88,…]**

A powerful framework for solving general LPs (and more)

**LP:** $\min c^T x$

s.t. $Ax = b$

$x \geq 0$

**Idea:** Take care of "hard" constraints by adding a "barrier" to the objective

**"easy"** constraints (use projection)

**"hard"** constraints

# (Path-following) Interior-point method (IPM)

[Dikin '67, Karmarkar '84, Renegar '88,…]

A powerful framework for solving general LPs (and more)

**LP($\mu$): min $c^T x$ - $\mu$ $\Sigma_i$ log $x_i$**

**s.t. $Ax = b$**

**$\cancel{x \geq 0}$**

**Idea:** Take care of "hard" constraints by adding a "barrier" to the objective

**Observe:** The barrier term enforces **$x \geq 0$** implicitly

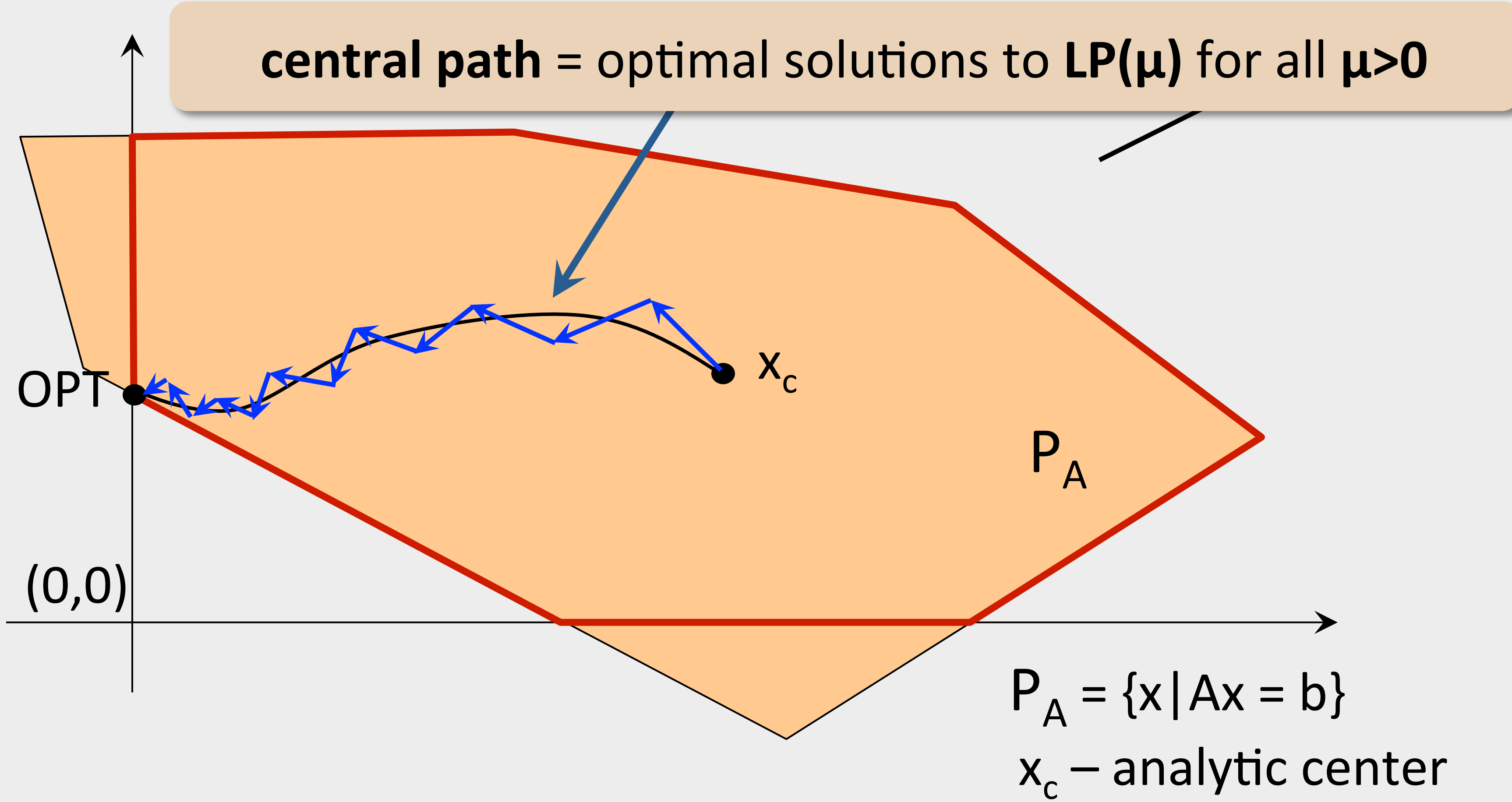
**Furthermore:** for large **$\mu$**, **LP($\mu$)** is easy to solve and

**LP($\mu$)** → original **LP**, as **$\mu \to 0^+$**

**Path-following routine:**

→ Start with (near-)optimal solution **$x(\mu)$** to **LP($\mu$)** for large **$\mu>0$**

→ Take an **improvement step** that gradually reduces **$\mu$** while maintaining the (near-)optimality of **$x(\mu)$** (wrt current **$\mu$**)




**Path-following routine:**

→ Start with (near-)optimal solution **x(μ)** to **LP(μ)** for large **μ>0**

→ Take an **improvement step** that gradually reduces **μ** while maintaining the (near-)optimality of **x(μ)** (wrt current **μ**)

# Can we use IPM to get a faster max flow alg.?

**Conventional wisdom:** This will be too slow!

→ Each **Newton's step** = solving a linear system $\mathbf{O(n^{\omega})=O(n^{2.373})}$ time (prohibitive!)

**But:** When solving **flow problems** – only $\mathbf{\tilde{O}(m)}$ time **[DS '08]**

**Fundamental question:** What is the number of iterations?

**[Renegar '88]:** $\mathbf{O(m^{1/2} \log \varepsilon^{-1})}$

**Unfortunately:** This gives only an $\mathbf{\tilde{O}(m^{3/2})}$-time algorithm

**Improve the $\mathbf{O(m^{1/2})}$ bound?**
Although believed to be **very** suboptimal,
its improvement is a major challenge


?

**[M '13]:** An improved $\mathbf{O(m^{3/7})}$ iterations bound for **unit-capacity max flow interior-point method**

**Observation:** IPM is solving max flow using electrical flows too!

**Result:** Better grasp of step size choice ($\ell_2$ **vs.** $\ell_4$ interplay)

- A simple energy-based argument recovers the $\mathbf{O(m^{1/2})}$ bound
- Lack of high-energy edges → better than $\mathbf{O(m^{1/2})}$ convergence

**Problem:** Removal of such high-energy edges is too drastic

**Instead:** Apply a **careful perturbation + preconditioning** of the **LP**
(This not only changes the current solution **but also the central path**)

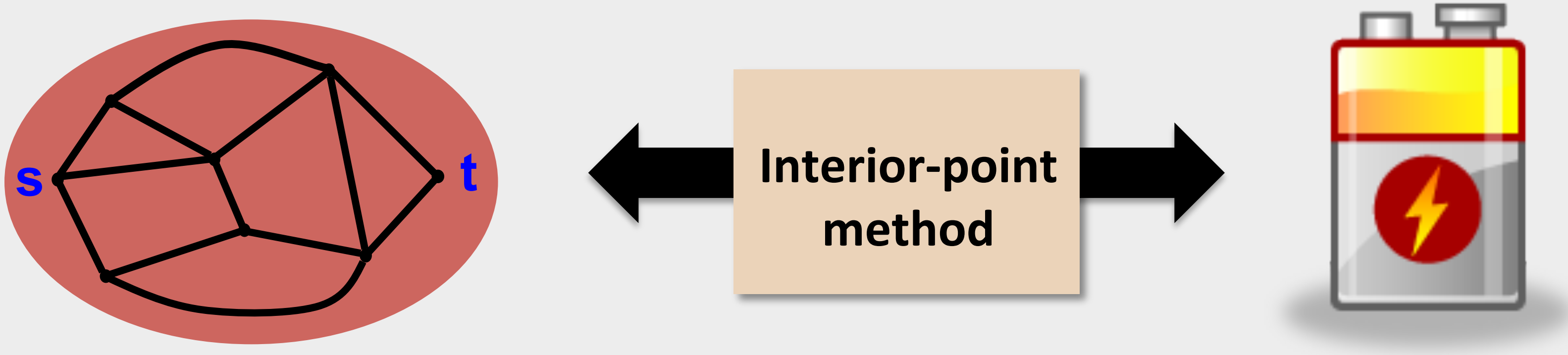
Use a new type of potential-based (**non-local**) convergence argument

Most of these elements seems broadly applicable (and new)

Will this lead to breaking the $\mathbf{\Omega(m^{1/2})}$ convergence barrier for **all** **LP**s?

# Conclusions
# and the Bigger Picture

# Maximum Flows and Electrical Flows




**Elect. flows + IPMs** → A powerful new approach to **max flow**

Can this lead to a **nearly-linear time** algorithm for the **exact directed** max flow?

We seem to have the "critical mass" of ideas

**Elect. flows =** next generation of "spectral" tools?

- Better "spectral" graph partitioning,
- Algorithmic grasp of random walks,
- …

# Max Flow and Interior-Point Methods

**Contributing back:** Max flow and electrical flows as a lens for analyzing general IPMs?

Our techniques can be lifted to the general LP setting

We can solve **any** LP within $\mathbf{\tilde{O}(m^{3/7}L)}$ iterations
**But:** this involves **perturbing** of this LP

Some (seemingly) new elements of our approach:

- Better grasp of $\ell_2$ vs. $\ell_4$ interplay wrt the step size $\boldsymbol{\delta}$
- Perturbing the central path when needed
- Usage of non-local convergence arguments

Can this lead to breaking the $\mathbf{\Omega(m^{1/2})}$ barrier for all LPs?

**[Lee Sidford '14]:** $\mathbf{\tilde{O}(n^{1/2})}$ iteration bound

# Bridging the Combinatorial and the Continuous

paths, trees, partitions,
routings, matchings,
data structures...

↔

matrices, eigenvalues,
linear systems, gradients,
convex sets...

**Powerful approach:** Exploiting the interplay of the two worlds

Some other early "success stories" of this approach:

- Spectral graph theory aka the "eigenvalue connection"
- Fast SDD/Laplacian system solvers
- Graph sparsification, random spanning tree generation
- Graph partitioning

...and this is just the beginning!

# Thank you

## Questions?