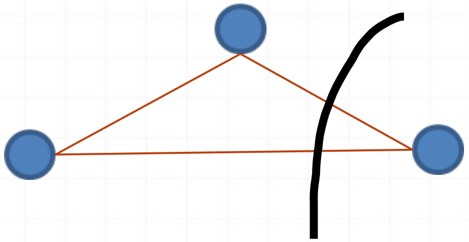
Local Flow Partitioning for Faster Edge Connectivity

Monika Henzinger, Satish Rao, Di Wang

University of Vienna, UC Berkeley, UC Berkeley

Edge Connectivity

- Edge-connectivity λ: least number of edges whose removal disconnects the graph.
- Minimum cut: set of edges of minimum size whose removal disconnects the graph.
  - Edge-connectivity = size of minimum cut in **unweighted** graphs



$\lambda=2$

Prior Work

Deterministic algorithm

Gabow'91	$O(\lambda m \log n)$	unweighted (multi-)graph
Kawarabayashi &Thorup'15	$O(m \log^{12} n)$	simple graph
Henzinger, Rao, W'17	$O(m \log^{2} n \log\log^{2} n)$	simple graph

Randomized algorithm

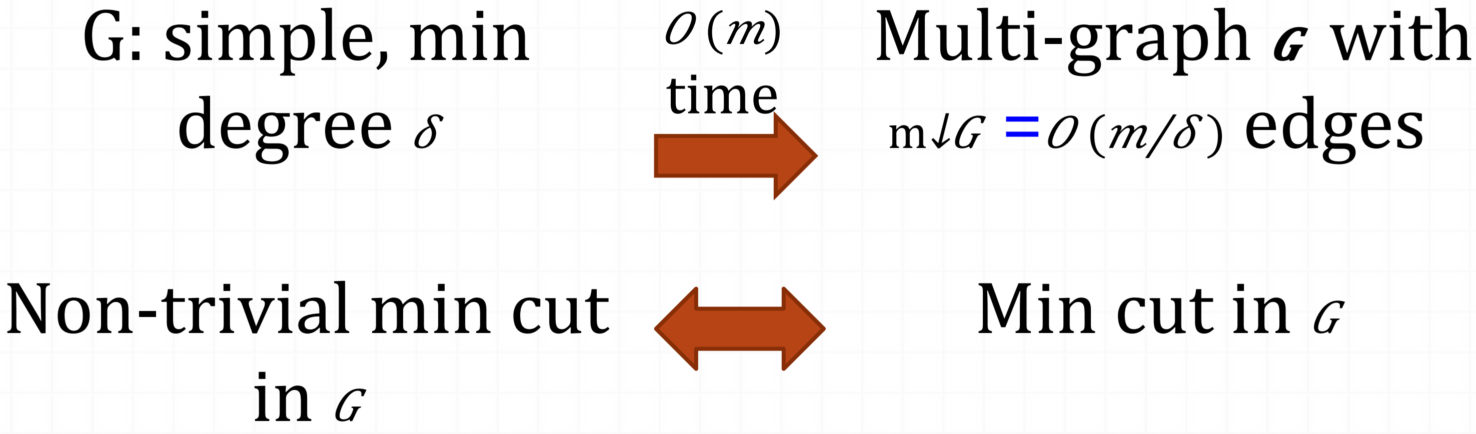
Karger'00	$O(m \log^{3} n)$	weighted graph

n nodes, m edges, min cut = λ

Simple graph: undirected, unweighted, no parallel edges
Multi-graph: can have parallel edges.

Kawarabayashi-Thorup(KT)

- **Theorem**



G: simple, min degree δ

$O(m)$ time

Multi-graph $\boldsymbol{G}$ with $m_G = O(m/\delta)$ edges

Non-trivial min cut in G

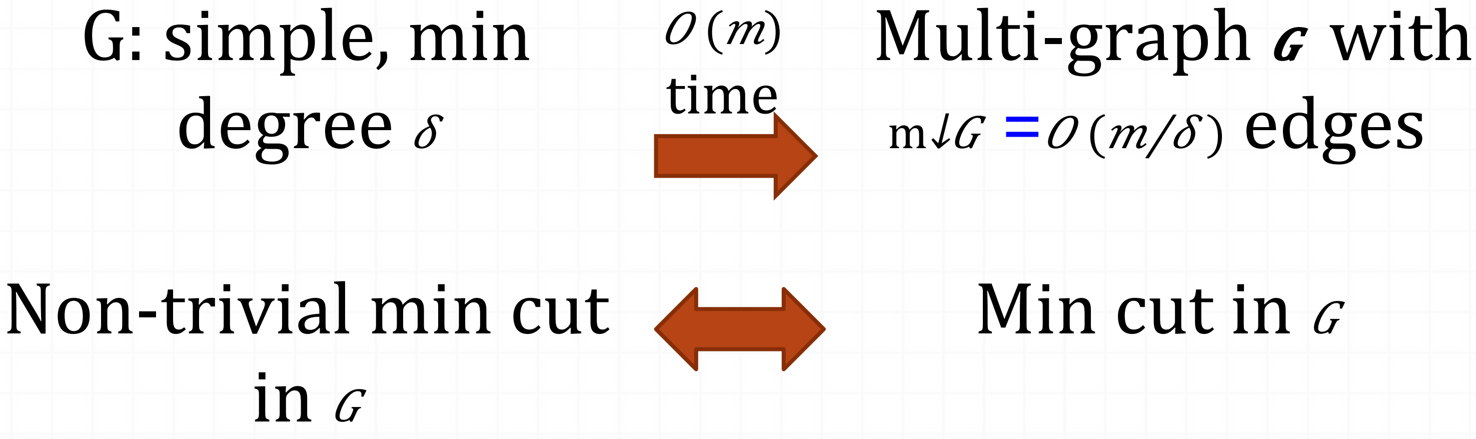
Min cut in G

- *Trivial cut*: only 1 node on one side of the cut.
- The min degree δ bounds the edge connectivity λ

$\boldsymbol{\lambda \leq \delta}$

Kawarabayashi-Thorup(KT)

- **Theorem**



G: simple, min degree δ

$O(m)$ time

Multi-graph $\boldsymbol{G}$ with $m_G = O(m/\delta)$ edges

Non-trivial min cut in G

Min cut in G

- Gabow's algorithm on G

$O(\lambda m_{\boldsymbol{G}} \log m) = O(\lambda m/\delta) = O(m)$

- Assume $\delta = \Omega(\log n)$

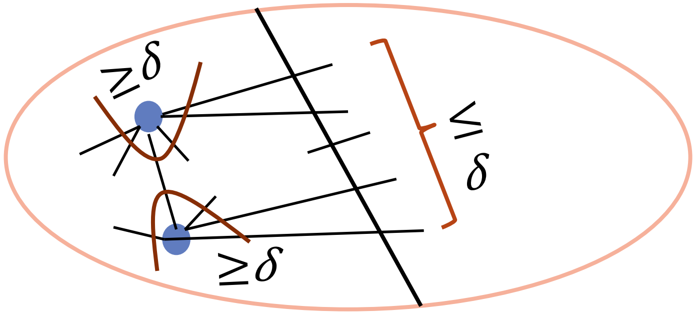
$\boldsymbol{\lambda \le \delta}$

Low Conductance Cut

Conductance: $\phi(A)=|E(A,A\ )|/\min\{vol(A),vol(A\ )\}$

$$vol(A)=\sum_{v\in A}\deg(v)$$

Non-trivial cut of size $\leq\delta$ has low conductance!



2 nodes: ≥ 2δ total degree

≤δ edges across the cut

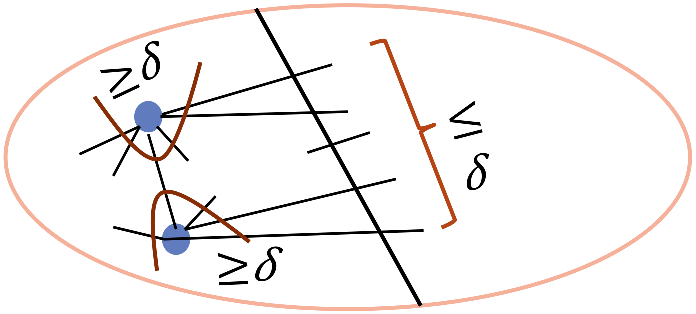
≥2 nodes ⇒ Ω(δ) nodes

Low Conductance Cut

Conductance: $\phi(A)=|E(A,A\ )|/\min\{vol(A),vol(A\ )\}$

$vol(A)=\sum_{v\in A} \deg(v)$

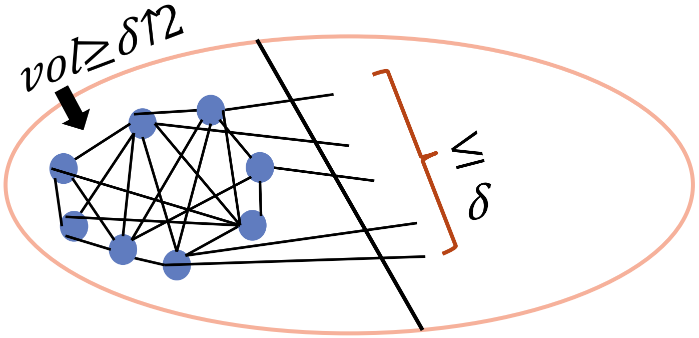
Non-trivial cut of size $\leq\delta$ has low conductance!



2 nodes: ≥ 2δ total degree

≤δ edges across the cut

≥2 nodes ⇒ Ω(δ) nodes

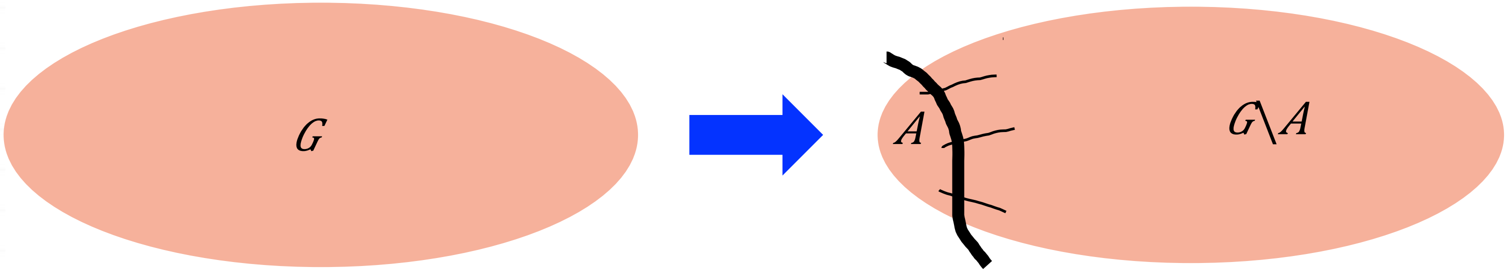


volume is $\Omega(\delta^2)$

$\Longrightarrow$ conductance $O(1/\delta)$

Local Graph Partitioning

Central tool in [KT'15], improved by us



Given G with m edges, find cut (A, A)

- Low conductance: $\phi(A) = O(1/\log m)$
- Local running time: $O(vol(A)\log^{c} m)$
  - Cannot afford $O(m)$ in recursive decomposition

PageRank/Diffusion [ACL’06]

Input: 1 unit of mass at a vertex v, rate of decay **α**

Maintains 2 vectors in n-dimensional space:

  - p =“settled mass” and r = “unsettled mass”

- **Initially:** $p = 0$, $r = 1$ at v and 0 everywhere else
- Repeat:
  - for every vertex u:
    - $p'(u) = p(u) + \alpha\, r(u)$ mass settles
    - $r'(u) = (1 - \alpha)\, r(u)/2$
    - For each neighbor v of u:
      $r'(v) = r(v) + (1- \alpha)r(u)/(2\deg(u))$ mass pushed to neighbors
  - $p = p'$, $r = r'$

PageRank/Diffusion [ACL'06]

- Input: starting distr., rate of decay **α**
- Settle fraction α of residual mass per round
- Spread half of the remaining evenly to neighbors
- ε-approx. of limiting distribution in time $O(1/(\alpha\varepsilon))$

PageRank/Diffusion [ACL'06]

Input: starting distr., rate of decay **α**

- Typical local partitioning result:

∃ conductance $O(\phi^{2}/\log m)$ cut

Find conductance ϕ cut A in time $O(vol(A)/\phi^{2})$

- Quadratic loss in cut quality and running time

PageRank/Diffusion [ACL’06]

Input: starting distr., rate of decay **α**

- Settle fraction α of residual mass per round
- Spread half of the remaining evenly to neighbors
- ε-approx. of limiting distribution in time $O(1/(\alpha\varepsilon))$
- Typical local partitioning result:

∃ conductance $\boldsymbol{O(\phi\uparrow 2 /\log m\ )}$ cut

Find conductance $\boldsymbol{\phi}$ cut A in time $\boldsymbol{O\,(vol(A)/\phi\uparrow 2\ )}$

- Quadratic loss in cut quality and running time

Flow-based Method

- Polylog loss in cut quality
- Difficult to make the running time local

Flow-based Method

- Polylog loss in cut quality
- Difficult to make the running time local

Two-level structure [HRW'17]

- *Unit-Flow*
  - Try to find low conductance cut
  - Running time "global" ~ size of instance
- *Excess Scaling*
  - Get running time local
  - Control instance size for Unit-Flow via value of unit.

Unit-Flow(G, Δ, ϕ)

Called repeatedly on "partial" flow problems

Input:

Graph G

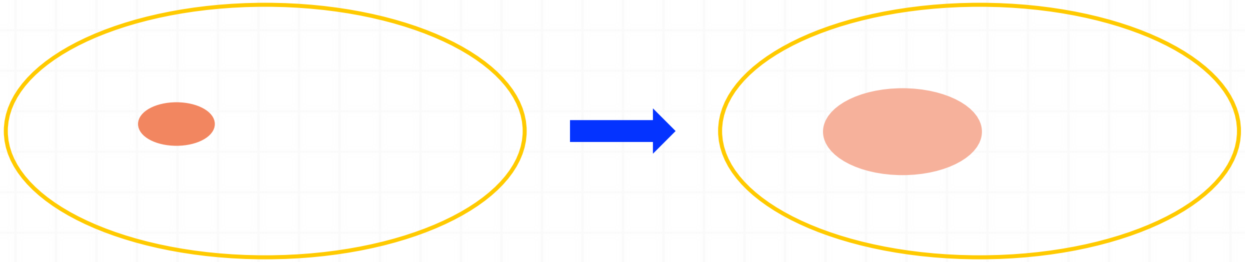
Source supply **Δ**: $\forall \boldsymbol{v}\, \Delta(\boldsymbol{v}) \leq \mathbf{2}\, deg(\boldsymbol{v})$ **units**

Parameters: target conductance **ϕ**

Flow Problem:

Each v has sink capacity $\deg(v)$ **units.**

Edge capacities $=1/\phi$ **units.**



Unit-Flow($G, \mathit{\Delta}, \phi$)

Variant of preflow push-relabel

Preflow $f: V \times V \rightarrow \mathbb{R}$

- Antisymmetry: $f(u,v) = -f(v,u)$
- Non-deficient flows: $\forall v,\ \sum u\uparrow ▒ f(v,u) \leq \Delta(v)$
- Respects edge capacities

$\boldsymbol{f}(\boldsymbol{v}) = \sum u\uparrow ▒ f(u,v) + \Delta(v)$

Unit-Flow(G, Δ, ϕ)

Variant of preflow push-relabel

Preflow $f\colon V\times V\to\mathbb{R}$

Antisymmetry: $f(u,v)=-f(v,u)$

Non-deficient flows: $\forall v,\ \sum u\uparrow ▒ f(v,u) \leq \Delta(v)$

Respects edge capacities

$\boldsymbol{f(v)}=\sum u\uparrow ▒ f(u,v)+\Delta(v)$

Push-relabel algorithm:

Each vertex has a height, starting at 0.

Repeatedly pick any v with excess (i.e. $f(v)>\mathrm{deg}(v)$)

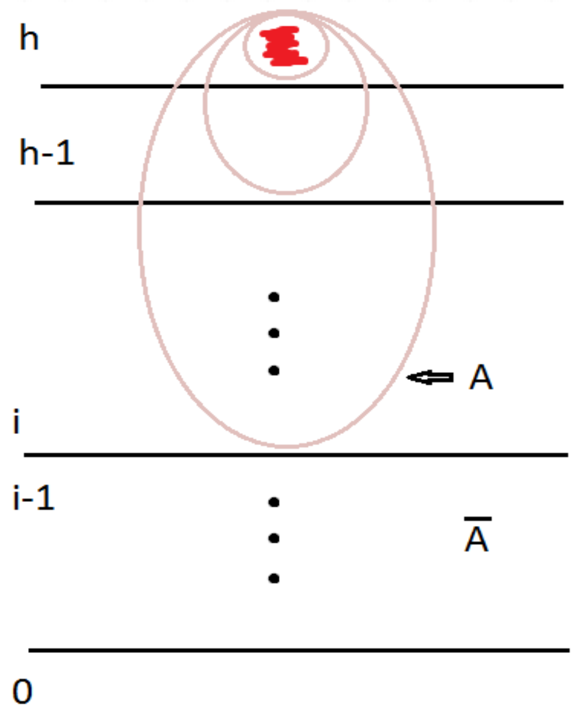
Push: send excess to lower neighbor along edges with residual capacity.

Relabel: if not possible, raise height of v by 1.

Unit-Flow(G, Δ, ϕ)

Key adaptations

- Upper-bound height by $\mathbf{h} = \boldsymbol{logm}$ /
  - Flow solution not guaranteed: Might not push all flow to sources



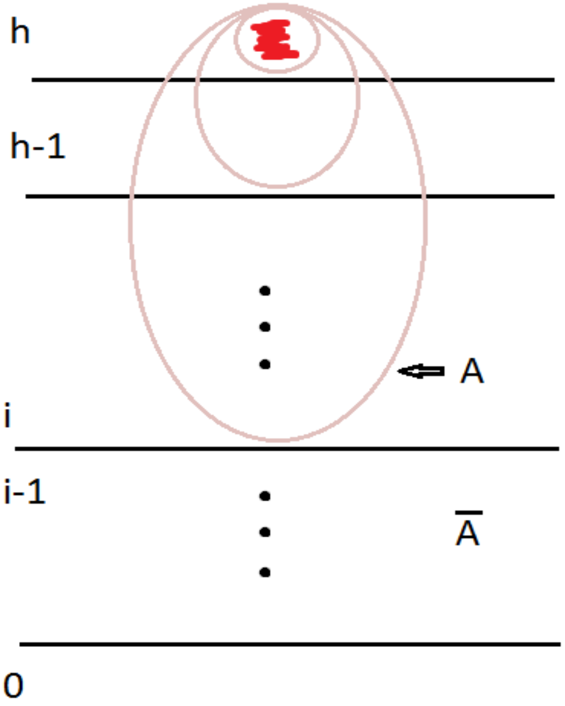
Unit-Flow(G, Δ, ϕ)

Key adaptations

- Upper-bound height by $\mathbf{h} = \boldsymbol{\log m / \phi}$,
  - Flow solution not guaranteed
  - But then ∃ conductance $O(\phi)$ "level cut"

Region growing argument

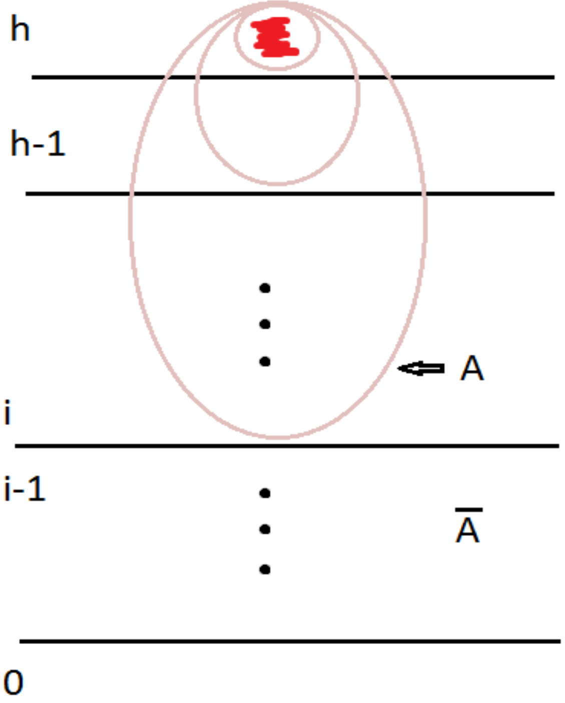
$(1+\phi)^{h} \gg m$



Unit-Flow(G, Δ, ϕ)

Key adaptations

- Upper-bound height by $\mathbf{h}=\boldsymbol{logm} /$
  - Flow solution not guaranteed
  - ∃ conductance $O(\phi)$ “level cut”



Region growing

$(1+\phi)\uparrow h \gg m$

- Upper-bound excess on vertex
  - Maintain $f(v)\leq 2\deg(v)$, assumed at start

⇒ Total excess $\leq \mathrm{vol}(A)$ at the end

Unit-Flow(G, Δ, ϕ)

$f(v)$= # units of supply on v at the end

- **Either** routes all source supply to sinks

$\forall v : f(v) \leq \deg(v)$

- **Or** finds conductance $O(\phi)$ cut $(A, A\ )$,

  and total excess bounded by $vol(A)$

total excess=$\sum v\uparrow$▒$\max(0, f(v) - \deg(v)\ )$

$$\leq vol(A)$$

- Explored subgraph volume $\approx \sum v\uparrow$▒$\Delta(v)$ = total units of flow

Unit-Flow(G, Δ, ϕ)

$f(v)=$ # units of supply on v at the end

- **Either** routes all source supply to sinks

$\forall v : f(v) \le \deg(v)$

- **Or** finds conductance $O(\phi)$ cut $(A, A\ )$,

  and total excess $\le vol(A)$

Running time: $O(|\Delta| \log m\ /\phi\ )$, $|\Delta|=\sum v\uparrow ▒\Delta(v)$, proportional to volume of explored subgraph

- **Running time:** $O(|\Delta|\log m/\phi)$, $|\Delta|=\sum_{v}\Delta(v)$
  But when cut (A,A) is returned we need time $O(vol(A)/\phi)$
- **Idea:**
  - Repeatedly run Unit-Flow for doubling values of $|\Delta|$ until Unit-Flow returns a cut (A,A) with excess $\geq\Omega(|\Delta|/\log n)$
    - ➢ $vol(A)\geq\Omega(|\Delta|/\log n)$
  - Can bound running time of all preceding calls to Unit-Flow by $O(vol(A)\log^{2} n/\phi)$
  - Done by Excess Scaling

Idea:

- Repeatedly run Unit-Flow for doubling values of $|\Delta|$ until Unit-Flow returns a cut $(A,\bar{A})$ with $vol(A) \geq \Omega(|\Delta|/\log n)$
- Can bound running time of all preceding calls to Unit-Flow by $O(vol(A)\log^2 n/\phi)$
- If never a “large enough” cut is returned then $\sum_j vol(A_j)$ is “small” and the (weighted) sum of the flows returned by all the Unit-Flow routes “almost all” flow

Excess Scaling

Input:

Graph G

Source supply Δ, $|\Delta| = \sum_{v} \Delta(v) = 2m$

Flow problem:

Each v sink of capacity $\mathbf{deg}(\boldsymbol{v})$

Sufficient edge capacity for all calls to Unit-Flow

Excess Scaling

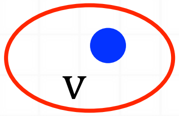
Source supply Δ, $/\Delta/=\sum v\uparrow ▒ \Delta(v) =2m$

Each v sink of capacity **deg**$(\boldsymbol{v})$

Divide into “growing” phases for Unit-Flow

- Start with large enough unit value $\mu=\max ┬ v\,\Delta(v)/2\ deg(v)$

$\Delta\downarrow 0 =\Delta/\mu \quad \rightarrow \quad \Delta\downarrow 0\ (v)\leq 2\deg(v)$,



Problem size: 2deg(v)

Excess Scaling

Source supply Δ, $/\Delta/=\sum_{v}\Delta(v) = 2m$

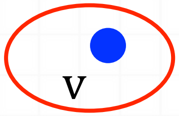
Each v sink of capacity **deg**$(\boldsymbol{v})$

Divide into "growing" phases for Unit-Flow

- Start with large enough unit value $\mu=\max_{v}\Delta(v)/2\,deg(v)$

  $\Delta_0 = \Delta/\mu \quad \rightarrow \quad \Delta_0(v) \leq 2\deg(v)$,

- Either returns low conductance cut *or* $\forall v\!: f(v) \leq \deg(v)$



Problem size: 2deg(v)

Excess Scaling

Source supply Δ, $/\Delta/=\sum v\uparrow ▒\Delta(v) = 2m$

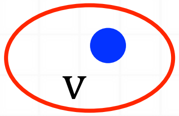
Each v sink of capacity $\boldsymbol{\deg(v)}$

Divide into “growing” phases for Unit-Flow

- Start with large enough unit value $\mu=\max ┬ v\, \Delta(v)/2\, deg(v)$

  $\Delta\downarrow 0 = \Delta/\mu \quad \rightarrow \quad \Delta\downarrow 0\ (v) \leq 2\deg(v)$,

- Either returns low conductance cut: STOP
  or $\forall v{:}\, f(v) \leq \deg(v)$: RESCALE and CALL Unit-Flow again



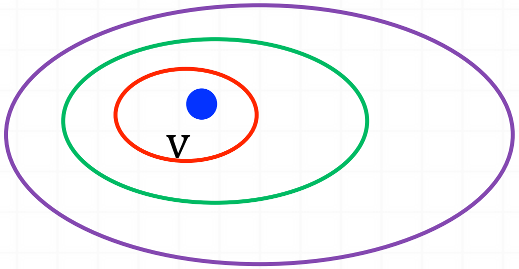
Excess Scaling

Source supply Δ, $|\Delta| = \sum_{v} \Delta(v) = 2m$

Each v sink of capacity $\mathbf{deg}(\boldsymbol{v})$

- Start with large enough unit value μ such that $\forall v$: $\Delta_0(v) = \Delta(v)/\mu \leq 2\deg(v)$
- Iteratively call Unit-Flow until low conductance cut with “large” volume is returned:
  - If Unit-Flow does not find such a cut, then $\forall v$: $f(v) \leq \deg(v)$: $Set\ \Delta_{j+1} \approx 2f_j$, i.e. $|\Delta|$ roughly doubles
    - Volume of explored subgraph, roughly doubles

Explored subgraph volume:
$2\deg(v) \rightarrow 4\deg(v)$
$\rightarrow 8\deg(v) \rightarrow 16\deg(v)$ …



Excess Scaling

Low conductance cut in local time

- Terminate when encounter “good cut” = Low conductance + large volume
  - j-th Unit-Flow: running time $O(|\Delta_j| \log m / \phi)$
  - Running time of **all previous** Unit-flow: $O(|\Delta_j| \log m / \phi)$

Excess Scaling

Low conductance cut in local time

- Terminate when encounter “good cut” = Low conductance + large volume
  - j-th Unit-Flow: running time $O(|\Delta_j| \log m / \phi)$
  - Running time of **all previous** Unit-flow: $O(|\Delta_j| \log m / \phi)$
  - Cut A_j returned by last Unit-Flow
    - Low conductance: $O(\phi)$
    - Large volume: $\mathbf{vol}(A)=\mathbf{\Omega}(|\Delta_j| / \log m)$
  - **Conductance ϕ cut A in time $O(vol(A) \log^2 m / \phi)$**

Excess Scaling

Low conductance cut in local time

- Terminate when encounter “good bottleneck”
  - j-th Unit-Flow: running time $O(|\Delta_j|\log m/\phi)$
  - Running time of **all previous** Unit-flow: $O(|\Delta_j|\log m/\phi)$
  - Cut A_j returned by last Unit-Flow
    - Low conductance: $O(\phi)$
    - Large volume: **vol**$(A)=\boldsymbol{\Omega}(|\Delta_j|/\log m)$
  - **Conductance $\boldsymbol{\phi}$ cut $\boldsymbol{A}$ in time $\boldsymbol{O(vol(A)\log^2 m/\phi)}$**
- Otherwise flow spread over G, almost all supply routed to sinks

Excess Scaling + Unit-Flow	**vs.**	**PageRank**
Spread “stuff” to find bottleneck		
Flow routing		Probability diffusion
Fail when no good enough “bottleneck”, so “stuff” spreads over entire graph		

Excess Scaling + Unit-Flow vs. PageRank

Spread "stuff" to find bottleneck

Excess Scaling + Unit-Flow	PageRank
Flow routing	Probability diffusion

Fail when no good enough "bottleneck", so "stuff" spreads over entire graph

Quality of cut vs. How easy to spread "stuff"

Excess Scaling + Unit-Flow	PageRank
$U=O(1/\phi)$	$\alpha=O(\phi\uparrow2)$

Excess Scaling + Unit-Flow vs. PageRank

Spread “stuff” to find bottleneck

Excess Scaling + Unit-Flow	PageRank
Flow routing	Probability diffusion

Fail when no good enough “bottleneck”, so “stuff” spreads over entire graph

Quality of cut vs. How easy to spread “stuff”

Excess Scaling + Unit-Flow	PageRank
$U=O(1/\phi)$	$\alpha=O(\phi^2)$

Quality of cut vs. Running time

Excess Scaling + Unit-Flow	PageRank
$O(vol(A)/\phi)$	$O(vol(A)/\phi^2)$

Wrap-up

Flow-based local low conductance method

- polylog loss versus quadratic loss of PageRank

+

Framework developed in [KT’15]

+

Appropriate interface

↓

Deterministic $O(m\log^2 m \log\log^2 m)$ algorithm for min cut in simple unweighted graphs

Open questions

Min cut in more general graphs:

- Determ. $o(mn)$ alg. for multi- or weighted graphs
- Directed graphs

Experimental evaluation

Further applications of flow-based local method:

- Local clustering (ICML'17)