Elements of Spectral Graph Theory II

A Toolkit for Fast Graph Algorithms

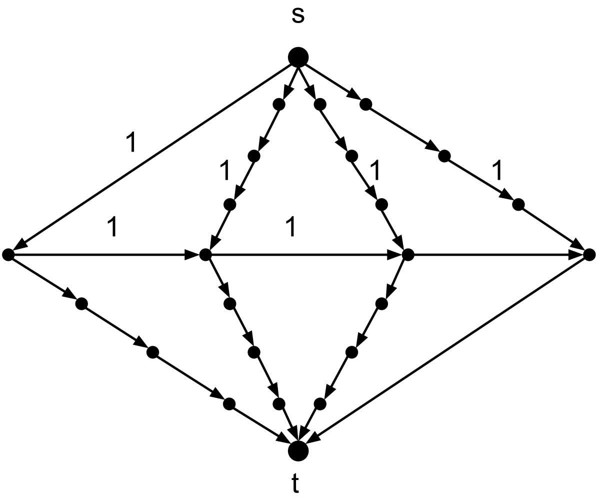
Aaron Schild

UC Berkeley EECS

March 27th, 2018

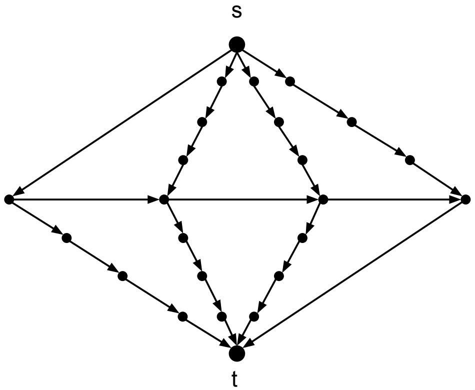
Motivation: Maximum Flow

Push as much flow from s to t without violating edge capacities.



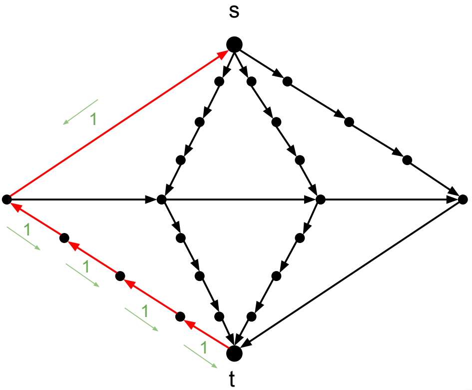
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



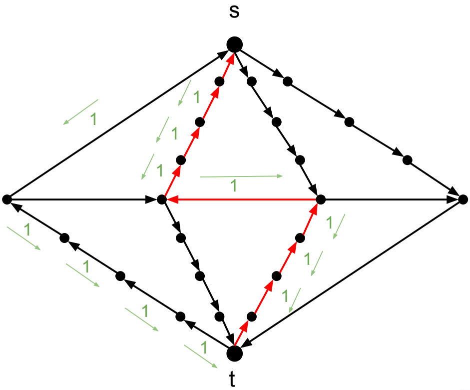
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



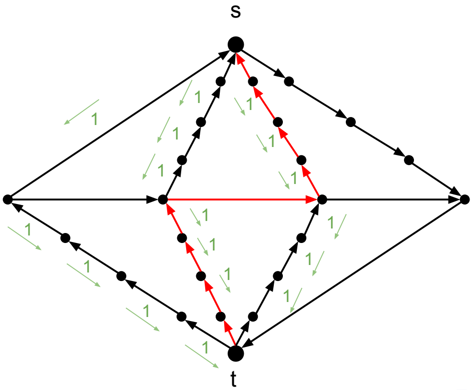
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



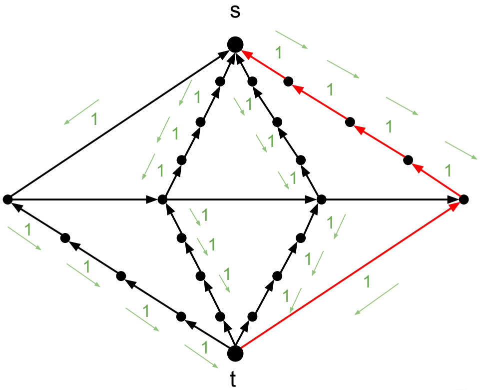
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



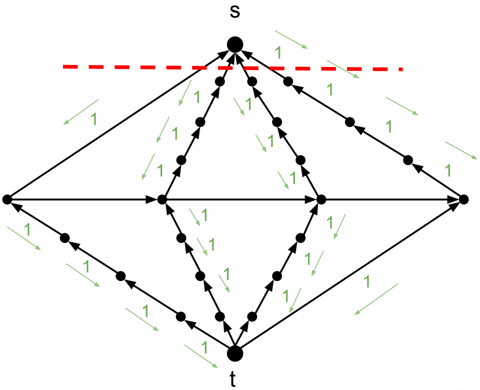
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



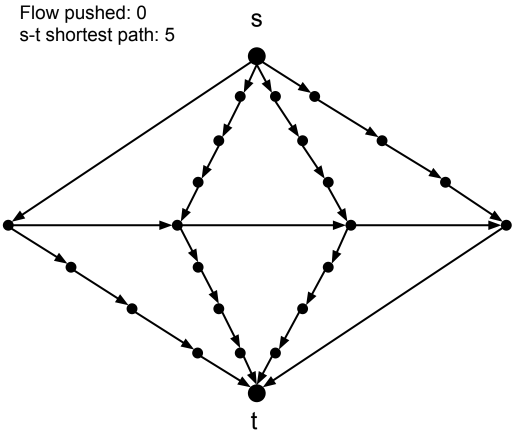
Classic technique: Ford-Fulkerson

- F: maximum flow value
- Push flow along paths one at a time
- $O(m)$ time to find each path, F paths
- $O(mF)$ total work



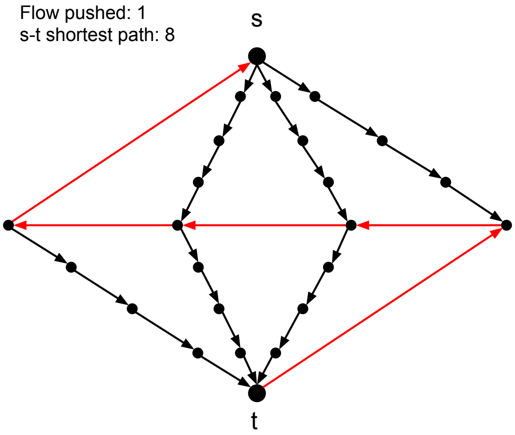
$O(m^{3/2})$ time: Blocking Flow

- Push flow along maximal collection of $s - t$ shortest paths
- $s - t$ distance increases after each augmentation



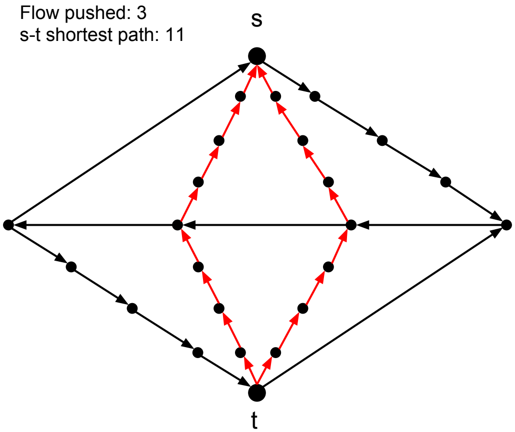
$O(m^{3/2})$ time: Blocking Flow

- Push flow along maximal collection of $s - t$ shortest paths
- $s - t$ distance increases after each augmentation



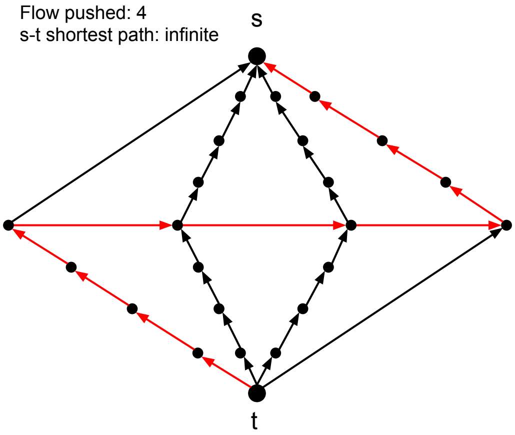
$O(m^{3/2})$ time: Blocking Flow

- Push flow along maximal collection of $s - t$ shortest paths
- $s - t$ distance increases after each augmentation



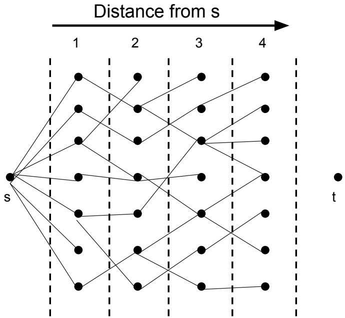
$O(m^{3/2})$ time: Blocking Flow

- Push flow along maximal collection of $s-t$ shortest paths
- $s-t$ distance increases after each augmentation



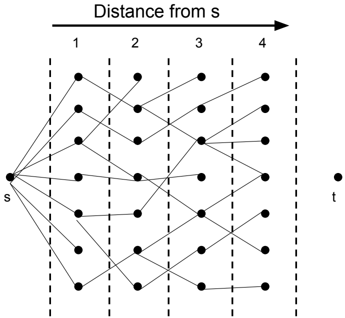
Analysis of Blocking Flow

- $s - t$ distance increases after each augmentation



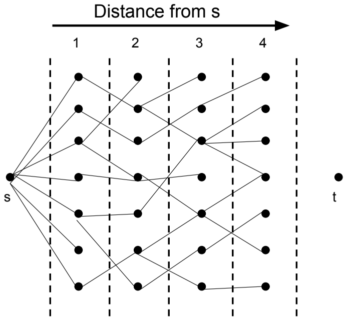
Analysis of Blocking Flow

- $s - t$ distance increases after each augmentation
- **If distance $> D$, there must be a cut with size at most m/D**



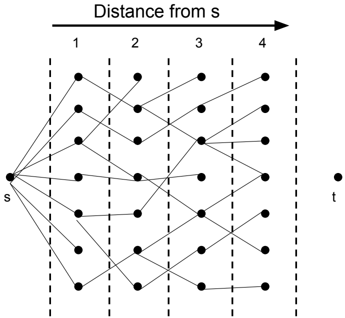
Analysis of Blocking Flow

- $s - t$ distance increases after each augmentation
- **If distance $> D$, there must be a cut with size at most m/D**
- Only m/D more paths needed at this point



Analysis of Blocking Flow

- $s - t$ distance increases after each augmentation
- **If distance $> D$, there must be a cut with size at most** m/D
- Only m/D more paths needed at this point
- Runtime: $O(m(D + m/D)))$ for any $D > 0$, minimized with $D = m^{1/2} \rightarrow O(m^{3/2})$ time

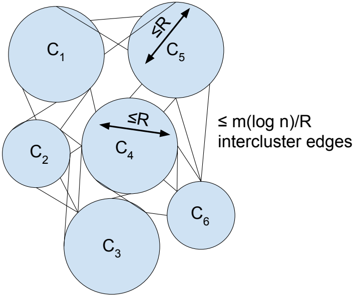


Graph Partitioning and Region Growing

Theorem ([LR99])

For any value $R > 0$, can partition a graph into clusters $C_1, C_2, \ldots, C_k$ with two properties:

- *each cluster has diameter at most R*
- *the number of edges between clusters is at most $\tilde{O}(m/R)$*

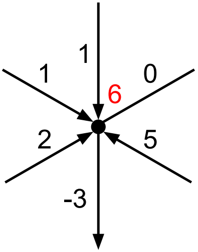


Maximum Flow as Norm Minimization

- suppose graph has unit capacities, m edges, and n vertices

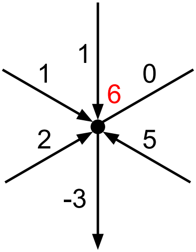
Maximum Flow as Norm Minimization

- suppose graph has unit capacities, m edges, and n vertices
- M: $n \times m$ matrix that maps a vector of flow to the induced vertex demand



Maximum Flow as Norm Minimization

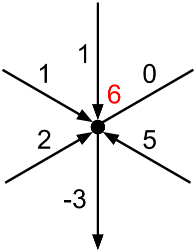
- suppose graph has unit capacities, m edges, and n vertices
- M: $n \times m$ matrix that maps a vector of flow to the induced vertex demand
- χ_{st}: n-dimensional vector with $\chi_{st}(s) = 1$, $\chi_{st}(t) = -1$, and $\chi_{st}(v) = 0$ for all other vertices v



Maximum Flow as Norm Minimization

- suppose graph has unit capacities, m edges, and n vertices
- M: $n \times m$ matrix that maps a vector of flow to the induced vertex demand
- χ_{st}: n-dimensional vector with $\chi_{st}(s) = 1$, $\chi_{st}(t) = -1$, and $\chi_{st}(v) = 0$ for all other vertices v

Maximum flow problem: find maximum value of α for which there is a vector $f \in \mathbb{R}^m$ with $Mf = \alpha\chi_{st}$ with $||f||_\infty \leq 1$.

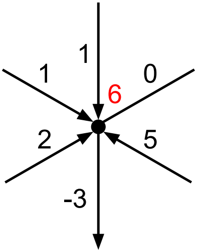


Maximum Flow as Norm Minimization

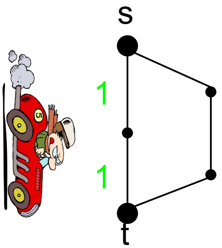
- suppose graph has unit capacities, m edges, and n vertices
- M: $n \times m$ matrix that maps a vector of flow to the induced vertex demand
- χ_{st}: n-dimensional vector with $\chi_{st}(s) = 1$, $\chi_{st}(t) = -1$, and $\chi_{st}(v) = 0$ for all other vertices v

Maximum flow problem: find maximum value of α for which there is a vector $f \in \mathbb{R}^m$ with $Mf = \alpha\chi_{st}$ with $||f||_\infty \leq 1$.

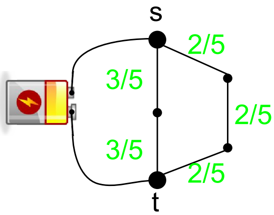
Equivalently up to scaling: $\min_f ||f||_\infty$ subject to $Mf = \chi_{st}$



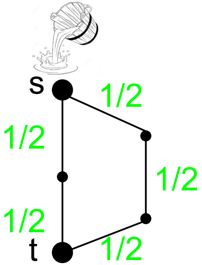
Other Norm-Minimizing Flows



(a) ℓ_1 (shortest path)

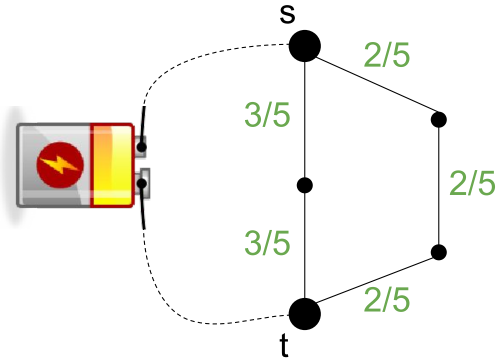


(b) ℓ_2



(c) ℓ_∞ (max flow)

Electrical Flows

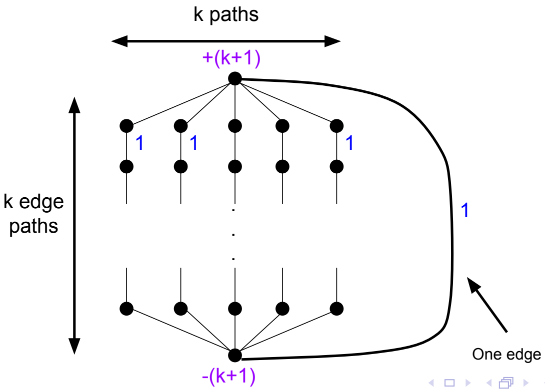


Can be computed in almost linear time by solving a Laplacian linear system! [ST003]

Finding maximum flows using electrical flows

Algorithm (similar to [CKM$^+$10]):

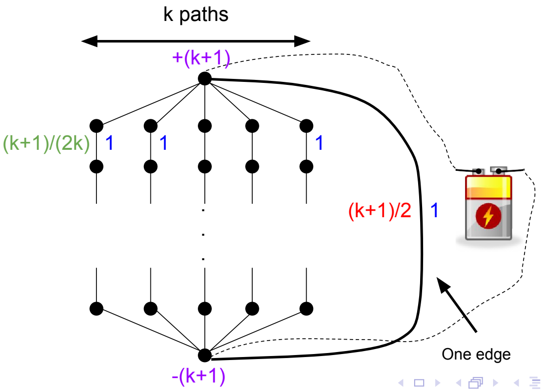
- Arbitrarily initialize resistances $\{\mathbf{r}_e\}_e$
- While there is some edge e with $\mathbf{f}_e > \mathbf{c}_e$
  - Let $\mathbf{f}$ be the $s - t$ electrical flow with resistances $\mathbf{r}$
  - Increase the resistance of edges with $\mathbf{f}_e > \mathbf{c}_e$



Finding maximum flows using electrical flows

Algorithm (similar to [CKM$^+$10]):

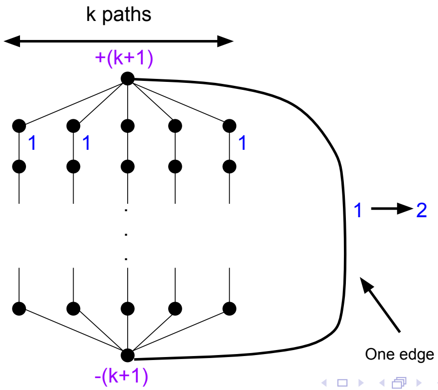
- Arbitrarily initialize resistances $\{\mathbf{r}_e\}_e$
- While there is some edge e with $\mathbf{f}_e > \mathbf{c}_e$
  - Let $\mathbf{f}$ be the $s - t$ electrical flow with resistances $\mathbf{r}$
  - Increase the resistance of edges with $\mathbf{f}_e > \mathbf{c}_e$



Finding maximum flows using electrical flows

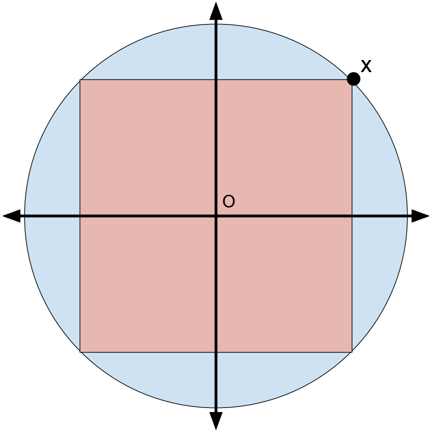
Algorithm (similar to [CKM$^+$10]):

- Arbitrarily initialize resistances $\{\mathbf{r}_e\}_e$
- While there is some edge e with $\mathbf{f}_e > \mathbf{c}_e$
  - Let $\mathbf{f}$ be the $s-t$ electrical flow with resistances $\mathbf{r}$
  - Increase the resistance of edges with $\mathbf{f}_e > \mathbf{c}_e$



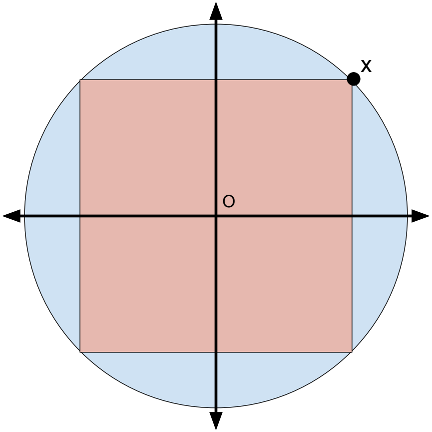
Geometry and its relationship to iteration count

- Can find an approximate max flow in $\tilde{O}(\sqrt{m}\text{poly}(1/\epsilon))$ iterations



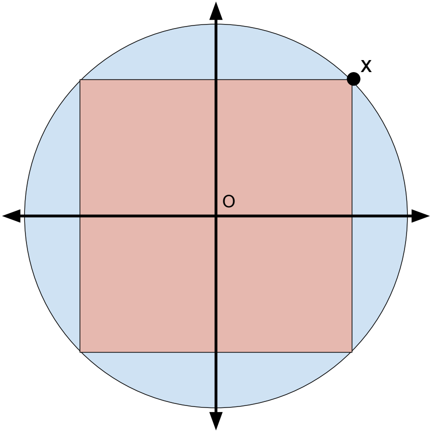
Geometry and its relationship to iteration count

- Can find an approximate max flow in $\tilde{O}(\sqrt{m}\text{poly}(1/\epsilon))$ iterations
- $\sqrt{m}$ comes from gap between ℓ_2 and ℓ_∞ norm in $\mathbb{R}^m$



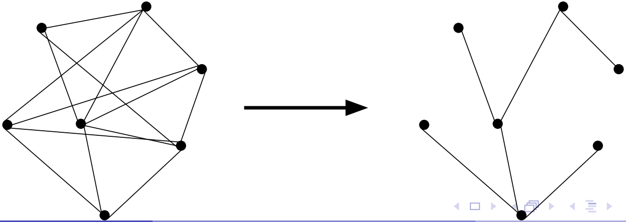
Geometry and its relationship to iteration count

- Can find an approximate max flow in $\tilde{O}(\sqrt{m}\text{poly}(1/\epsilon))$ iterations
- $\sqrt{m}$ comes from gap between ℓ_2 and ℓ_∞ norm in $\mathbb{R}^m$
- $\tilde{O}(m)$ time per iteration $\rightarrow \tilde{O}(m^{3/2}\text{poly}(1/\epsilon))$ time



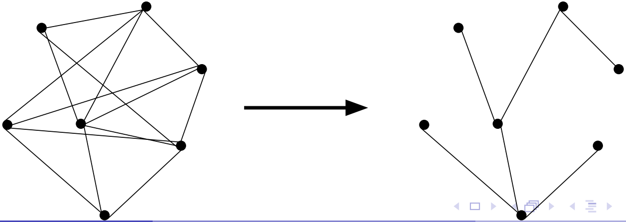
Better geometry with graph embeddings

- Embed graph into a simpler graph while preserving cuts/distances



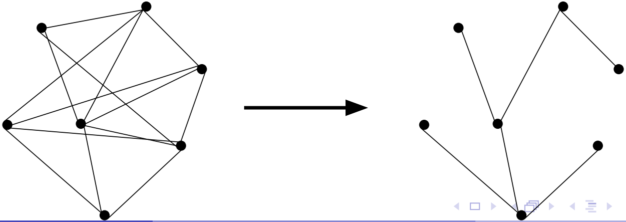
Better geometry with graph embeddings

- Embed graph into a simpler graph while preserving cuts/distances
- Simplest possible graph: a tree



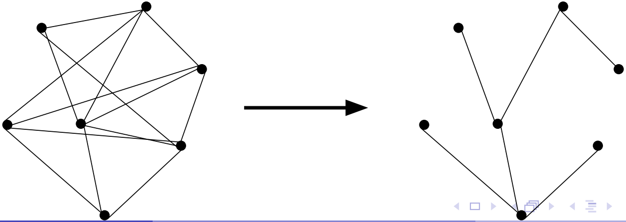
Better geometry with graph embeddings

- Embed graph into a simpler graph while preserving cuts/distances
- Simplest possible graph: a tree
- Unfortunately, can't do well with a single tree, but can do well with a distribution over trees!



Better geometry with graph embeddings

- Embed graph into a simpler graph while preserving cuts/distances
- Simplest possible graph: a tree
- Unfortunately, can't do well with a single tree, but can do well with a distribution over trees!
- Related embedding technique can be used to find an $m^{o(1)}$-approximate ℓ_∞ projection, which yields an $(1+\epsilon)$-approximate max flow in $O(m\text{polylog}(n)/\epsilon)$ time [KLOS14, She13, She17, Pen16]



Summary

- Graph partitioning (diameter v.s. cutsize)
- Writing flow problems as norm minimization
- Graph embedding

Bibliography

Paul Christiano, Jonathan A. Kelner, Aleksander Madry, Daniel A. Spielman, and Shang-Hua Teng.
Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs.
CoRR, abs/1010.2921, 2010.

Jonathan A. Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford.
An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations.
In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014, pages 217–226, 2014.

Frank Thomson Leighton and Satish Rao.
Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms.