Max flow min cut in higher dimensions

Art Duval[1] Caroline Klivans[2] Jeremy Martin[3]

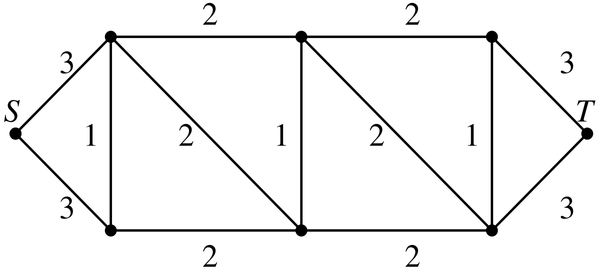
[1]University of Texas at El Paso

[2]Brown University

[3]University of Kansas

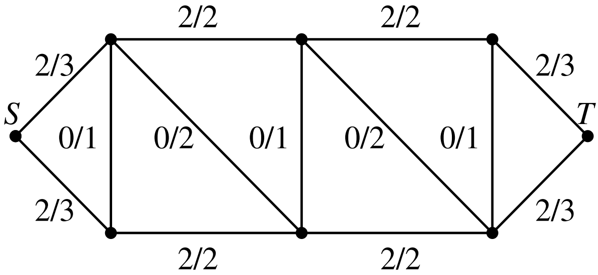
12th Joint UTEP/NMSU Workshop on Mathematics, Computer Science, and Computational Sciences
University of Texas at El Paso
October 27, 2012

Max flow



Given a graph G with source S, sink T, and edge capacities κ.

Max flow



Given a graph G with source S, sink T, and edge capacities κ.

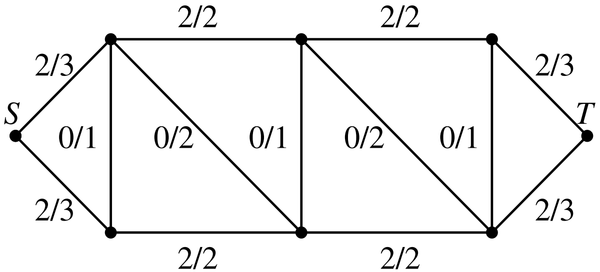
Definition

Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- net flow at each vertex, except S and T, is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is outflow(S) = inflow(T).

Max flow



Given a graph G with source S, sink T, and edge capacities κ.

Definition

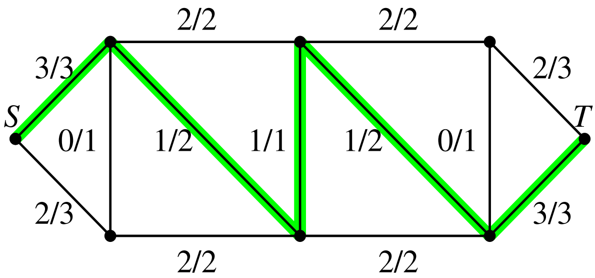
Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- net flow at each vertex, except S and T, is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is outflow(S) = inflow(T).

What is maximum value of flow on G? How can we be sure?

Max flow



Given a graph G with source S, sink T, and edge capacities κ.

Definition

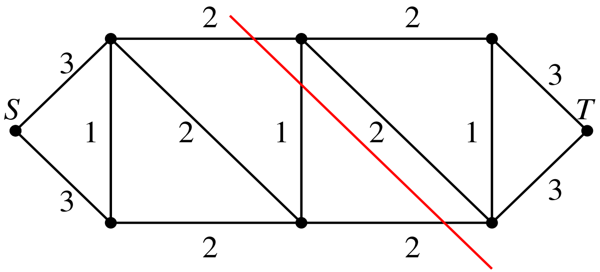
Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- net flow at each vertex, except S and T, is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is outflow(S) = inflow(T).

What is maximum value of flow on G? How can we be sure?

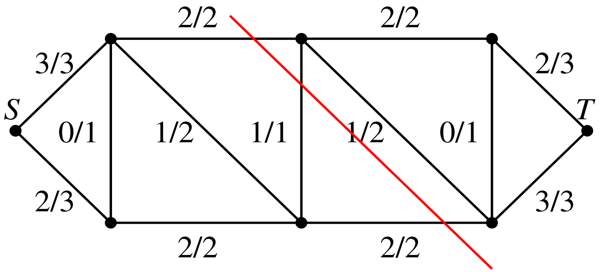
Min cut



Definition

Cut is minimal set of edges whose removal disconnects S from T.
Value of cut is $\sum_{e \in \text{cut}} \kappa_e$.

Min cut



Definition

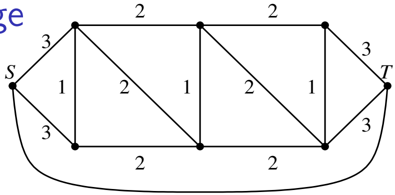
Cut is minimal set of edges whose removal disconnects S from T.
Value of cut is $\sum_{e \in \text{cut}} \kappa_e$.

Clearly, value(flow) $\leq$ value(cut), so max flow $\leq$ min cut.

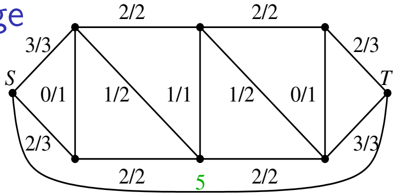
Theorem (Classic max flow min cut)

Max flow = min cut.

Add an extra edge



Add an extra edge



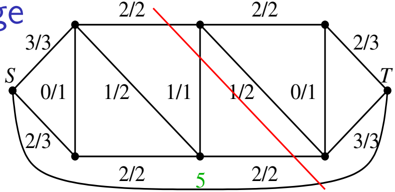
Definition

Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- net flow at each vertex is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is x_0.

Add an extra edge



Definition

Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

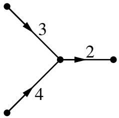
- net flow at each vertex is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is x_0.

Definition

Cut is minimal set of edges, including e_0, whose removal disconnects G. Value of cut is $\sum_{e \in \text{cut} \setminus e_0} \kappa_e$.

Flows and boundary

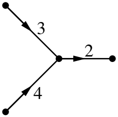


$$\begin{pmatrix}1 & 1 & -1\end{pmatrix}\begin{pmatrix}3\\4\\2\end{pmatrix} = 5.$$

Assign orientation to each edge (flow going “backwards” gets negative value)

$$\mathrm{netflow}(v) = \sum_{v=e^+} x_e - \sum_{v=e^-} x_e = \sum_{v\in e}(-1)^{\varepsilon(e,v)} x_e = (\partial x)_v$$

Flows and boundary



$$\begin{pmatrix}1 & 1 & -1\end{pmatrix}\begin{pmatrix}3\\4\\2\end{pmatrix} = 5.$$

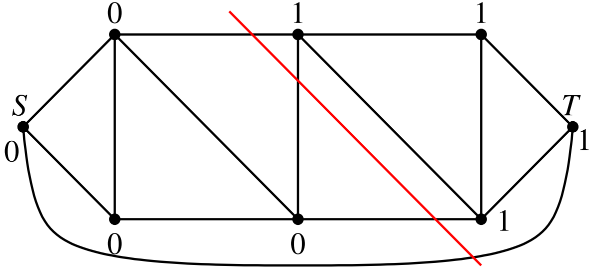
Assign orientation to each edge (flow going “backwards” gets negative value)

$$\text{netflow}(v) = \sum_{v=e^+} x_e - \sum_{v=e^-} x_e = \sum_{v\in e} (-1)^{\varepsilon(e,v)} x_e = (\partial x)_v$$

So net flow condition is $\partial x = 0$.

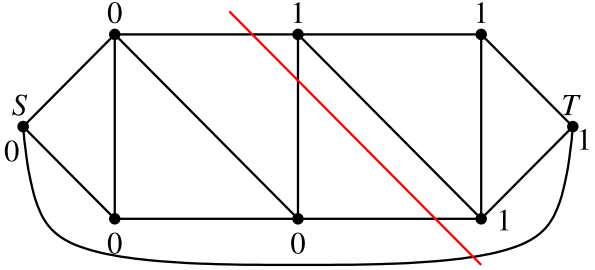
Cuts and coboundary

Assign 1 to every vertex in connected component with T, 0 to others. Let y_v be value at v. Edges in cut are those that have both 0 and 1 endpoints.



Cuts and coboundary

Assign 1 to every vertex in connected component with T, 0 to others. Let y_v be value at v. Edges in cut are those that have both 0 and 1 endpoints.



Coboundary will do this: $\partial^T y$ (linear combination of rows of ∂) gives characteristic vector of cut.

Linear programming

Flow is now a linear program

- Find vector x (in edge space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_e \leq x_e \leq \kappa_e$ (can omit e_0)
- max x_0

Linear programming

Flow is now a linear program

- Find vector x (in edge space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_e \leq x_e \leq \kappa_e$ (can omit e_0)
- $\max x_0$

The dual program is (can easily be reworked to say):

- Find vector y (in vertex space)
- Let $u = \partial^T y$ (in cut space)
- $u_0 = 1$
- $\min \sum_e \kappa_e |u_e|$

Linear programming

Flow is now a linear program

- Find vector x (in edge space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_e \leq x_e \leq \kappa_e$ (can omit e_0)
- $\max x_0$

The dual program is (can easily be reworked to say):

- Find vector y (in vertex space)
- Let $u = \partial^T y$ (in cut space)
- $u_0 = 1$
- $\min \sum_e \kappa_e |u_e|$

Linear programming says the solutions are equal; with some effort we can show the solution to the dual LP is the min cut problem.

Max flow in higher dimensions

Example: 2-dimensional complex; ∂ maps 2-dimensional cells (polygons) to edges.

Max flow in higher dimensions

Example: 2-dimensional complex; ∂ maps 2-dimensional cells (polygons) to edges.

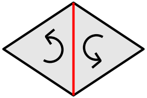
- Find vector x (in polygon space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_p \leq x_p \leq \kappa_p$ (can omit p_0)
- identify designated polygon p_0; max x_0

Max flow in higher dimensions

Example: 2-dimensional complex; ∂ maps 2-dimensional cells (polygons) to edges.

- Find vector x (in polygon space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_p \leq x_p \leq \kappa_p$ (can omit p_0)
- identify designated polygon p_0; max x_0

Find a 1-dimensional cycle on the complex, and attach a polygon face filling that cycle. We are trying to maximize circulation on that designated polygon (around that cycle), while making all circulation balance on each edge.



Min cut in higher dimensions

The dual program is (can easily be reworked to say):

- Find vector y (in edge space)
- Let $u = \partial^T y$ (in cut space)
- $u_0 = 1$
- min $\sum_p \kappa_p |u_p|$

Linear programming says the solutions are equal; with some effort we can show the solution to the dual LP is the min cut problem.

Min cut in higher dimensions

The dual program is (can easily be reworked to say):

- Find vector y (in edge space)
- Let $u = \partial^T y$ (in cut space)
- $u_0 = 1$
- min $\sum_p \kappa_p |u_p|$

Linear programming says the solutions are equal; with some effort we can show the solution to the dual LP is the min cut problem. In particular, the support on this cut is a minimal set of polygons whose removal introduces codimension-1 homology, e.g., 1-dimensional "circular" hole in 2-dimensional complex

Summary

Theorem (DKM)

The max circulation around a codimension-1 cycle (e.g., 1-dimensional cycle in 2-dimensional complex) equals the value of a minimum cut containing the added face that fills in the cycle (e.g., polygon filling in 1-dimensional cycle).

Fine print:

- cut is minimal set of faces whose removal increases codimension-1 homology
- cut vector is in span of row space of boundary matrix
- normalize cut vector by specifying its value is 1 on p_0, the added filling-in face
- cut vector might not be all 1's and 0's
- value of cut is inner product of capacities with cut vector