Max flow min cut in higher dimensions

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

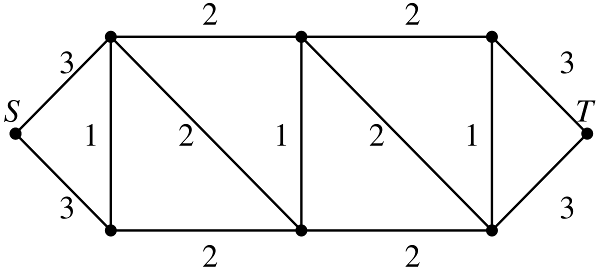
[1]University of Texas at El Paso

[2]Brown University

[3]University of Kansas

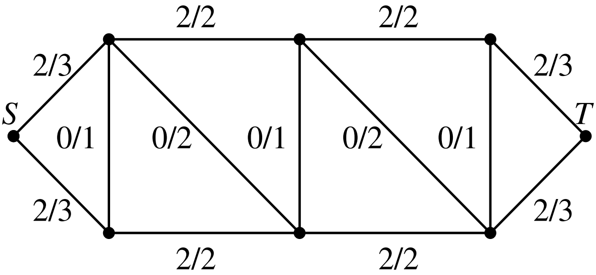
Joint Mathematics Meeting
AMS Special Session on Topological Combinatorics
San Diego, CA
January 12, 2013

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.

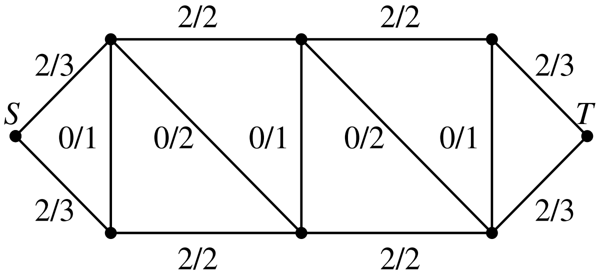
Definition

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

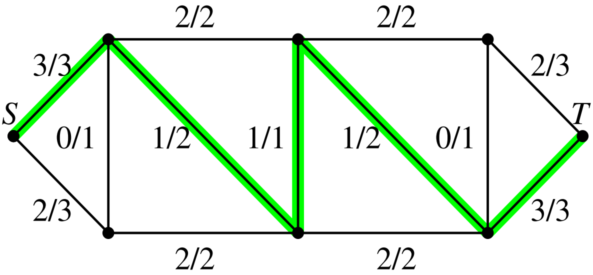
Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- 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

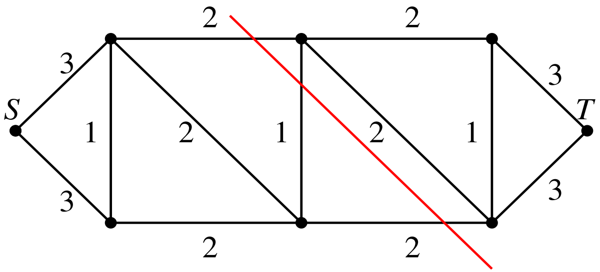
Flow on G is an assignment of flow x_e (non-negative number, and direction) to each edge such that:

- 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?

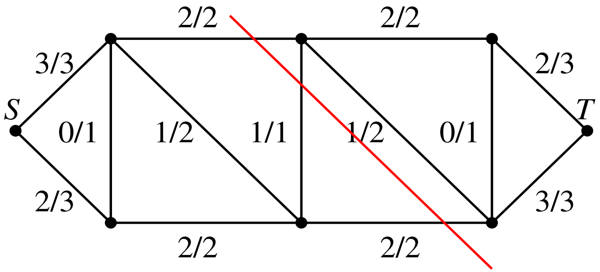
Min cut



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

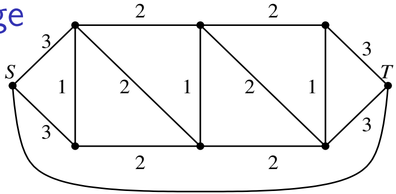
Cut is minimal set of edges whose removal disconnects S from T.
Value of cut is $\sum_{e \in \text{cut}} \kappa_e$.

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

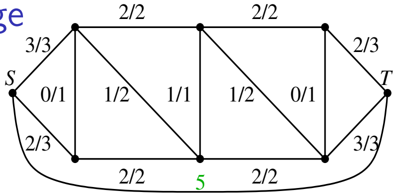
Theorem (Classic max flow min cut)

Max flow = min cut.

Add an extra edge



Add an extra edge



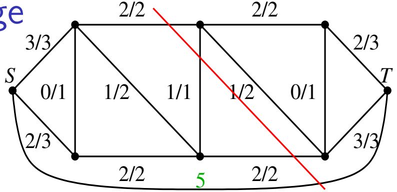
Definition

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:

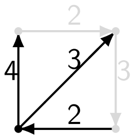
- net flow at each vertex is zero; and
- $|x_e| \leq \kappa_e$.

Value of flow is x_0.

Definition

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

Flows and boundary

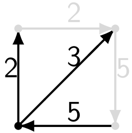


$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 4 \\ 3 \\ 3 \\ 2 \end{pmatrix} = -4 - 3 + 2 = -5$$

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



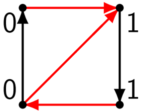
$$\begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 3 \\ 5 \\ 5 \end{pmatrix} = -2 - 3 + 5 = \quad 0$$

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

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

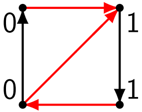
So net flow condition is $\partial x = 0$.

Cuts and coboundary



Assign 1 to every vertex on one side of the cut, 0 to every vertex on the other side.

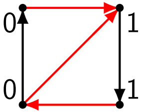
Cuts and coboundary



Assign 1 to every vertex on one side of the cut, 0 to every vertex on the other side. Let y_v be value at v. Compute coboundary,

$$\partial^T y = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}^T \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & -1 \end{pmatrix}$$

Cuts and coboundary



Assign 1 to every vertex on one side of the cut, 0 to every vertex on the other side. Let y_v be value at v. Compute coboundary,

$$\partial^T y = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}^T \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 0 & -1 \end{pmatrix}$$

Edges in cut are those that have both 0 and 1 endpoints.

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 $\sum_e \kappa_e |u_e|$

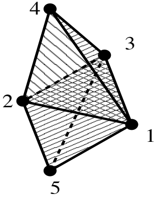
Linear programming says the solutions are equal; with some effort we can show the solution to the dual LP is the min cut problem.

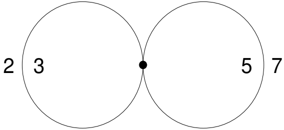
Cell complexes

Definition

A cell complex X is a finite CW-complex (i.e., collection of cells of different dimensions), with say n facets and p ridges, and a $p \times n$ cellular boundary matrix $\partial \in \mathbb{Z}^{p \times n}$.

Example





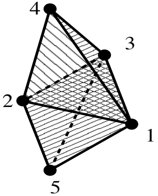
$$\begin{matrix} 2 & 3 & 0 & 0 \\ 0 & 0 & 5 & 7 \end{matrix}$$

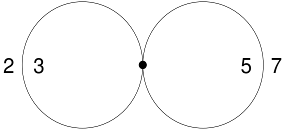
Cell complexes

Definition

A cell complex X is a finite CW-complex (i.e., collection of cells of different dimensions), with say n facets and p ridges, and a $p \times n$ cellular boundary matrix $\partial \in \mathbb{Z}^{p \times n}$.

Example





$$\begin{matrix} 2 & 3 & 0 & 0 \\ 0 & 0 & 5 & 7 \end{matrix}$$

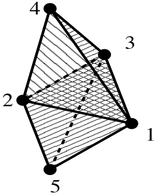
Remark

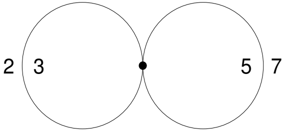
Any $\mathbb{Z}$ matrix can be the boundary matrix of a cell complex.

Cellular matroids

- Matroid whose elements are columns of boundary matrix
- Dependent sets are the supports of the kernel of the boundary matrix

Example





$$\begin{matrix} 2 & 3 & 0 & 0 \\ 0 & 0 & 5 & 7 \end{matrix}$$

Flow space and circuits

Definition

i-dimensional flow space of cell complex X is

$$\mathrm{Flow}_i(X) = \ker(\partial_i : C_{i-1}(X, \mathbb{R}) \to C_{i-1}(X, \mathbb{R})).$$

Flow space and circuits

Definition

i-dimensional flow space of cell complex X is

$$\mathrm{Flow}_i(X) = \ker(\partial_i : C_{i-1}(X, \mathbb{R}) \to C_{i-1}(X, \mathbb{R})).$$

A circuit of X is a minimal set of i-faces that support non-0 vector of $Flow_i(X)$

Remark

Circuits are the circuits (minimal dependent sets) of cellular matroid.

Flow space and circuits

Definition

i-dimensional flow space of cell complex X is

$$\mathrm{Flow}_i(X) = \ker(\partial_i : C_{i-1}(X, \mathbb{R}) \to C_{i-1}(X, \mathbb{R})).$$

A circuit of X is a minimal set of i-faces that support non-0 vector of $Flow_i(X)$

Remark

Circuits are the circuits (minimal dependent sets) of cellular matroid.

Example

Bipyramid

Cut space and cocircuits

Definition

i-dimensional cut space of cell complex X is

$$\mathrm{Cut}_i(X) = \mathrm{im}(\partial_i^* : C_{i-1}(X, \mathbb{R}) \to C_i(X, \mathbb{R})).$$

Remark

Cut space is the rowspace of the boundary matrix; cut space and flow space are orthogonal complements.

Cut space and cocircuits

Definition

i-dimensional cut space of cell complex X is

$$\mathrm{Cut}_i(X) = \mathrm{im}(\partial_i^* : C_{i-1}(X, \mathbb{R}) \to C_i(X, \mathbb{R})).$$

A cocircuit of X is a minimal set of i-faces that support non-0 vector of $\mathrm{Cut}_i(X)$

Remark

Cut space is the rowspace of the boundary matrix; cut space and flow space are orthogonal complements.

Remark

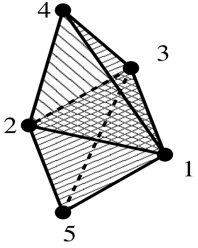
Cocircuits are the cocircuits of cellular matroid

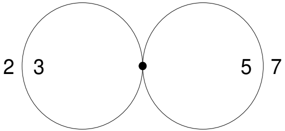
Topological interpretation of cocircuits

Remark

Cocircuits are minimal for increasing $(i-1)$-dimensional homology instead of decreasing i-dimensional homology

Examples





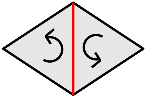
Max flow in higher dimensions

- Find vector x (in facet space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_f \leq x_f \leq \kappa_f$ (can omit f_0)
- identify designated facet f_0; max f_0

Max flow in higher dimensions

- Find vector x (in facet space)
- $\partial x = 0$ (x is in flow space)
- $-\kappa_f \leq x_f \leq \kappa_f$ (can omit f_0)
- identify designated facet f_0; max f_0

Find a codimension-1 cycle on the complex, and attach a facet f_0 filling that cycle. We are trying to maximize circulation on that designated facet (around that cycle), while making all circulation balance on each codimension-1 face (ridge).



Min cut in higher dimensions

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

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

Min cut in higher dimensions

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

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

Linear programming says this min value equals the max flow, and it is (by construction) in cut space. But is the solution u supported on a cocircuit?

Min cut in higher dimensions

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

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

Linear programming says this min value equals the max flow, and it is (by construction) in cut space. But is the solution u supported on a cocircuit?

Theorem (DKM)

The max flow equals the value of some solution to the dual LP whose support is a cocircuit.

Summary

Theorem (DKM)

The max flow around a codimension-1 cycle equals the "capacity" of a min cut containing the added face that fills in the cycle.

Summary

Theorem (DKM)

The max flow around a codimension-1 cycle equals the "capacity" of a min cut containing the added face that fills in the cycle.

Fine print:

- normalize cut vector by specifying its value is 1 on f_0, the added filling-in facet
- cut vector might not be all 1's and 0's
- capacity of cut is inner product of facet capacities with cut vector

Summary

Theorem (DKM)

The max flow around a codimension-1 cycle equals the "capacity" of a min cut containing the added face that fills in the cycle.

Fine print:

- normalize cut vector by specifying its value is 1 on f_0, the added filling-in facet
- cut vector might not be all 1's and 0's
- capacity of cut is inner product of facet capacities with cut vector

Questions:

- Is there an analogue to Ford-Fulkerson? That is, a combinatorial algorithm that would construct the "min cut", without relying on linear programming?
- What happens when we restrict to integers?