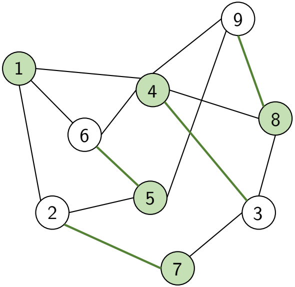
CE 205A
Transportation Logistics

Lecture 14

Cuts for TSPs – Part II

Previously on Transportation Logistics

Consider an undirected graph $G = (V, E)$. A *matching* $M \subseteq E$ is a set of disjoint edges (edges that do not have a node in common). A *node cover* is a set $N \subseteq V$ such that every edge has at least one end point in N.



Formulate the maximum cardinality matching and minimum cardinality cover problems using the set cover/packing/partitioning framework.

Previously on Transportation Logistics

The DFJ formulation using E-δ notation can be written as

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e & \\ \text{s.t.} \quad & x(\delta(u)) = 2 & \forall\, u \in V \\ & x(E(S)) \leq |S| - 1 & \forall\, S \subset V, S \neq \emptyset \\ & x_e \in \{0, 1\} & \forall\, e \in E \end{aligned}$$

The formulation with the alternate SEC constraints take the form

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e x_e & \\ \text{s.t.} \quad & x(\delta(u)) = 2 & \forall\, u \in V \\ & x(\delta(S)) \geq 2 & \forall\, S \subset V, S \neq \emptyset \\ & x_e \in \{0, 1\} & \forall\, e \in E \end{aligned}$$

We can further restrict $3 \leq |S| \leq |V|/2$. (Why?)

Previously on Transportation Logistics

This method generalizes the previous two approaches on connectivity and 2-connectedness and can provide constraints that result in maximum violations of SECs.

The idea is to simply find the global min-cut between any s-t pair in the support graph, i.e., find $S \subset V$ such that $\min x^{\mathsf{LP}}(\delta(S))$. If this is 2, we can generate a cut.

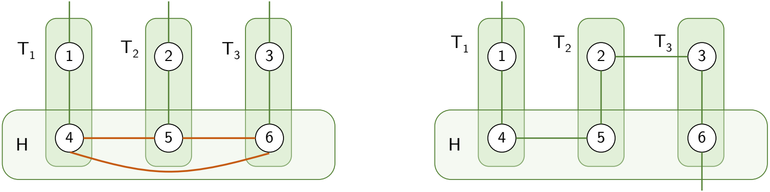
For instance, solving this would give 0 for disconnected graphs and an objective less than or equal to 1 for support graphs that are not 2-connected (e.g., $\{12, 13, 14, 15, 16\}$ in the instance with LP solution $= 682.5$).

Previously on Transportation Logistics

Consider a set of subsets $H, T_1, T_2, \ldots, T_k$ of V satisfying the following conditions.

- $|T_i| = 2$ and each T_i has a vertex in H and one in H^c.
- $T_1, T_2, \ldots, T_k$ are pairwise disjoint.
- $k \geq 3$ and is odd.

In this example, $T_1 = \{1, 4\}$, $T_2 = \{2, 5\}$, $T_3 = \{3, 6\}$, and $H = \{4, 5, 6\}$. The T sets are also called *teeth* and H is called the *handle*.



Previously on Transportation Logistics

Subsets which satisfy these conditions are called blossoms and can be used to generate Blossom inequalities or 2-matching inequalities of the form

$$x\left(\delta(H)\right) + \sum_{i=1}^{k} x\left(\delta(T_i)\right) \geq 3k + 1$$

Using the degree constraints $x(\delta(S)) = 2|S| - 2x(E(S)) \, \forall \, S \subseteq V$ (Why?). Hence, blossom inequalities can also be written as

$$x(E(H)) + \sum_{i=1}^{k} x(E(T_i)) \leq |H| + \sum_{i=1}^{k} |T_i| - \frac{3k+1}{2}$$

Lecture Outline

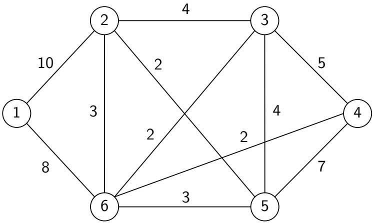
1. Gomory-Hu Trees
2. Odd Min-cut Sets
3. Exact Separation of Blossoms

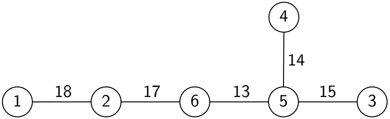
Lecture Outline

Gomory-Hu Trees

The global min-cut problem for an undirected graph has an indirect role to play in exact separation of Blossom inequalities.

Consider the problem of finding min-cuts on the network in the left panel where the edge weights are capacities. Do you notice any connection to the tree on the right?





The min-cut between every pair of nodes in the graph is same as that in the tree!

Gomory-Hu Trees

Introduction

Finding the min-cuts in a tree is easy (Why?) What is the global min-cut in a tree? How many steps are involved?

The following questions are worth exploring.

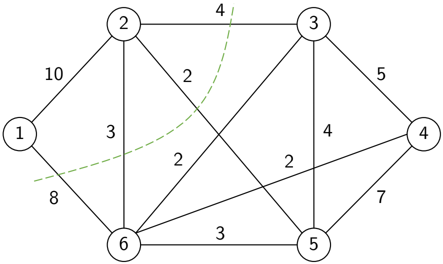
- Is it always possible to construct such a tree?
- If yes, can such a tree be discovered using an algorithm? What is its complexity?

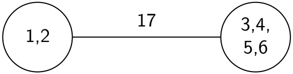
Let us first try to create such a tree.

Gomory-Hu Trees

Iteration 1

Suppose we find the min cut between two arbitrary nodes, say 2 and 6. The min cut is $\{1, 2\}$ and $\{3, 4, 5, 6\}$.



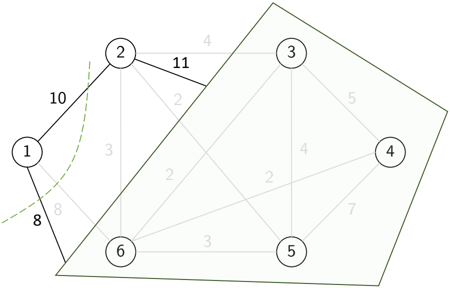


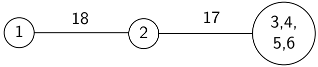
Let's now try to split the partitions. Suppose we start with nodes 1 and 2.

Gomory-Hu Trees

Iteration 2

We contract the rest of the nodes and find the min-cut between 1 and 2.

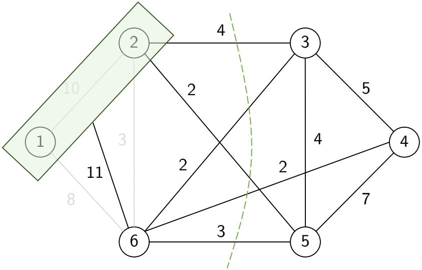


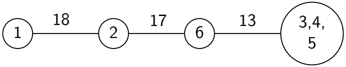


The min-cut capacity is 18 and we expand the contracted node 1,2 to create two nodes 1 and 2 in the tree.

Let us now try to split the contracted node 3,4,5,6. Suppose we select nodes 3 and 6.

We contract everything connected to the branches of the contracted node 3,4,5,6 and find a min-cut between 3 and 6.





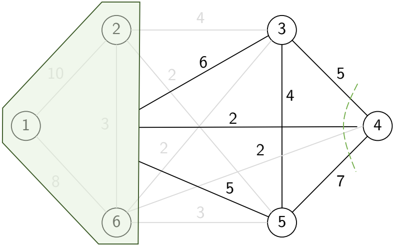
The min-cut capacity is 13 and we expand the contracted node 3,4,5,6 to create two nodes 6 and 3,4,5 in the tree.

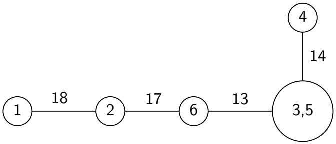
Gomory-Hu Trees

Iteration 4

Let us now try to split the contracted node 3,4,5. Suppose we select nodes 4 and 5.

We contract everything connected to the branches of the contracted node 3,4,5 and find a min-cut between 4 and 5.





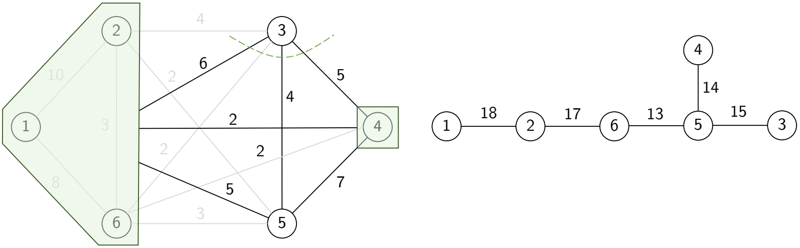
The min-cut capacity is 14 and we expand the contracted node 3,4,5 to create two nodes 4 and 3,5 in the tree.

Gomory-Hu Trees

Iteration 5

Finally, we split the contracted node 3,5.

We contract everything connected to the branches of the contracted node 3, 5, i.e., 1,2,6 and 4.



The min-cut capacity is 15 and we expand the contracted node 3,5 to create two nodes 3 and 5 in the tree. Note that the algorithm converges in $n - 1$ iterations.

Gomory-Hu Trees

Existence

It turns out that Gomory-Hu trees always exist because the capacity function is sub-modular.

More specifically, given a graph $G = (V, E)$ and capacities $c : E \to \mathbb{R}^+$, define $f : 2^V \to \mathbb{R}^+$ using the capacities as $f(S) = c(\delta(S))$.

This function turns out to be sub-modular. A function $f : 2^S \to \mathbb{R}$ is submodular if for $A, B \in 2^S$, $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.

Lecture Outline

Odd Min-Cut Sets

Odd Min-Cut Sets

Introduction

An exact separation algorithm for finding violated blossom inequalities was proposed by Padberg and Rao (1982).

This method uses a modified global-min cut problem called as the *odd minimum-cut set problem*.

For this version, assume that vertices in the graph have labels *odd* and *even*. Suppose, that the total number of *odd* nodes in the graph is even.

Odd Min-Cut Sets

Definition

The odd minimum-cut set problem is to find a global min-cut that separates the graph into two partitions with an odd number of *odd* nodes.

Define a label $\lambda(S)$, where $S \subseteq V$, to be *odd* if S contains an odd number of *odd* labelled nodes.

Mathematically, this problem can be written as

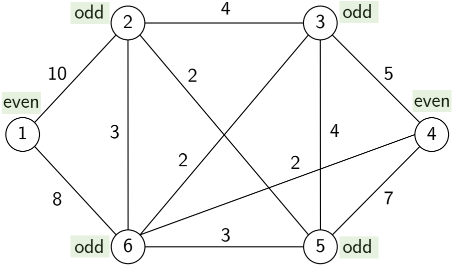
$$c(R, R^c) = \min\{c(S, S^c) : S \subseteq V, \lambda(S) \text{ is } \textit{odd}\}$$

where $c(S, S^c)$ is the capacity of the cut set (S, S^c).

Theorem

Given a graph $G = (V, E)$, let (S, S^c) be the min-cut partitioning. Then, there exists an odd minimum cut-set (R, R^c) such that $R \subset S$ or $R \subset S^c$.

What is the global-min cut in this network? Odd minimum-cut set?



The consequence of the earlier theorem is that we can search for the odd min cut-set separately on shrunk graphs obtained from S and S^c.

Odd Min-Cut Sets

Algorithm

The following observations help in designing an algorithm for finding the odd min-cut set. Recall that $\lambda(V)$ is *even*.

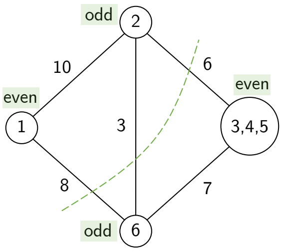
1. If the global min-cut (S, S^c) is such that $\lambda(S)$ is *odd*, then we terminate and declare (S, S^c) as the odd min-cut solution.
2. Else, we solve the global min cut problem recursively on contracted graphs derived from S and S^c. The odd min-cut must belong to one of these two graphs.
3. These graphs have *even* labels but contain at least two less *odd* nodes than the original graph. If the global min-cut in these shrunk graphs violates Condition 1, we repeat this process recursively.

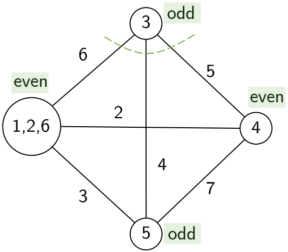
Odd Min-Cut Sets

Example

The global min-cut is 13 and the partition of vertices is $\{1, 2, 6\}$ and $\{3, 4, 5\}$.

Define two new graphs $G^1 = (V^1, E^1)$ and $G^2 = (V^2, E^2)$, where $V^1 = \{1, 2, 6, \{3, 4, 5\}\}$ and $V^2 = \{3, 4, 5, \{1, 2, 6\}\}$ and edges as shown below.





The odd min-cut is therefore $\min\{17, 15\}$. Both these graphs yield odd cut sets and hence we can terminate.

Lecture Outline

Exact Separation of Blossoms

Exact Separation of Blossoms

How does this help identify a violated blossom inequality? Recall that Blossom inequality is of the form

$$x\left(\delta(H)\right) + \sum_{i=1}^{k} x\left(\delta(T_i)\right) \geq 3k + 1$$

Since $x(\delta(S)) = 2|S| - 2x(E(S)) \,\forall\, S \subseteq V$,

$$x(\delta(T_i)) = 4 - 2x(E(T_i))$$

Suppose T represents the set of edges in the teeth, then the blossom inequality can be written as

$$\begin{aligned} & x\left(\delta(H)\right) + \sum_{i=1}^{k} \Big(4 - 2x(E(T_i))\Big) \geq 3k + 1 \\ \Rightarrow & x\left(\delta(H)\right) + k - 2x(T) \geq 1 \\ \Rightarrow & x\left(\delta(H) \setminus T\right) + \left(|T| - x(T)\right) \geq 1 \end{aligned}$$

Exact Separation of Blossoms

The left-hand side does not look like the edges of a cut.

$$\sum_{e \in \delta(H) \setminus T} x_e + \sum_{e \in T} (1 - x_e) \geq 1$$

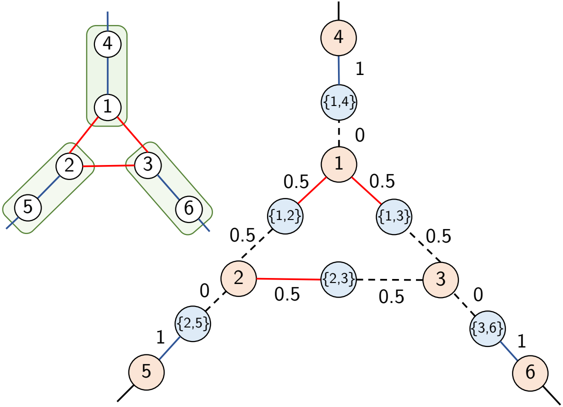
But can we define a new graph and make it look like the cut capacity, particularly the odd min-cut capacity?

If the min-cut capacity is less than 1, then we would have discovered a violated Blossom inequality.

Note that the first term has x_e and the second one has $1 - x_e$. Hence, we would have to create edges whose capacity is $1 - x_e$.

Suppose the current solution is $\bar{x}$. Let $E(\bar{x})$ be the edges of the support graph. Let's create a new graph $G(\bar{x}) = (W, F \cup \bar{F})$ as follows.

Let the vertices W include the original vertices V and a new vertex for every edge in $E(\bar{x})$. That is, $W = V \cup E(\bar{x})$.



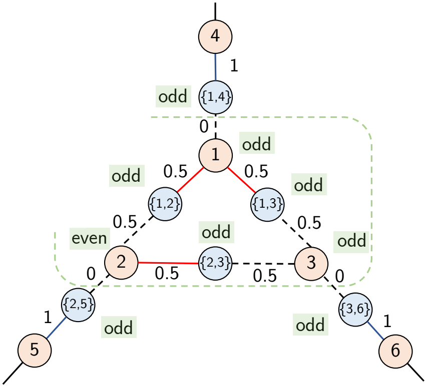
Each edge $e = \{u, v\}$ in $E(\bar{x})$ is split into two edges $\{u, e\}$ and $\{e, v\}$.

Edge $\{u, e\}$ is assigned a capacity x_e and $\{e, v\}$ is assigned a capacity $1 - x_e$. Call these edge sets F and $\bar{F}$, respectively.

Exact Separation of Blossoms

Introduction

Mark a node odd if it meets an odd number of nodes in $\bar{F}$, even otherwise.



A min-odd cut in this graph identifies the handle and the cut edges identify the tooth. Note in this example, the cut represents the $\sum_{e \in T}(1 - x_e)$ term and the other term is 0.

Exact Separation of Blossoms

Introduction

Theorem

Let $H \subset V$ and $T \subset E(\bar{x})$ be such that

$$x\left(\delta(H) \setminus T\right) + \left(|T| - x(T)\right) < 1$$

Then there exists an odd cut set (S, S^c) in $G(\bar{x})$ such that

$$x\left(\delta(H) \setminus T\right) + \left(|T| - x(T)\right) = c(S, S^c)$$

We can formally obtain the handle $H = S \cap V$ and the teeth using the following construction

$$T = \left\{ e \in E(\bar{x}) \mid \exists e' \in (S, S^c) \text{ such that } e' = \{e, v\}, v \in V, c_{e'} = 1 - \bar{x}_e \right\}$$

Here (S, S^c) is estimated with respect to the original graph G.

Exact Separation of Blossoms

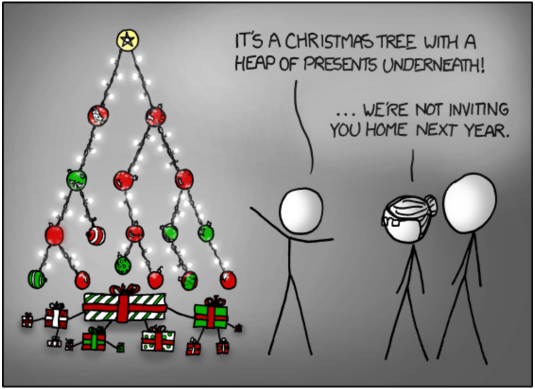
Generalization

Odd cuts are also referred to as T-cuts.

Blossom inequalities are also valid for general matching problems called *b-matching* in which every vertex v can be matched to at most $b(v)$ other vertices.

The TSP problem without the SEC constraints is a special case of b-matching and is also called a perfect 2-matching. In addition it has upper bounds of 1 on the edge variables.

Your Moment of Zen



Source: xkcd