

Combinatorial optimization: Max-Cut, Min UnCut and Sparsest Cut Problems

Marina Danilova, Nikita Puchkin, Alena Shilova

SKOLKOVO INSTITUTE OF SCIENCE AND TECHNOLOGY

March 13, 2017

Outline

1 Combinatorial Optimization

- Max-Cut
- Min UnCut
- Sparsest Cut

2 Approaches

- Naive Algorithm
- SDP relaxations

Max-Cut Problem

Given an undirected weighted graph $G = (V, E, W)$, where

$V = \{1, \dots, n\}$ – set of vertices

$E \subseteq V \times V$ – set of edges

$W : E \rightarrow \mathbb{R}$ – weights

One wants to find a partition $f : V \rightarrow \{0, 1\}$ in order to maximize the sum of edges in the cut:

$$\sum_{(i,j) \in E: f(i) \neq f(j)} w_{ij} \rightarrow \max,$$

where w_{ij} stands for weight of the edge (i, j) .

Min UnCut Problem

Given an undirected weighted graph $G = (V, E, W)$, one wants to find a partition $f : \{0, 1\}$ in order to minimize the sum of edges out of the cut:

$$\sum_{(i,j) \in E: f(i) \neq f(j)} w_{ij} \rightarrow \min$$

REMARK. Let $Opt(MUC)$ and $Opt(MC)$ stand for optimal solutions of Min UnCut and Max-Cut problems respectively. Then it holds

$$Opt(MUC) + Opt(MC) = \sum_{(i,j) \in E} w_{ij}$$

Sparsest Cut Problem

Given an undirected weighted graph $G = (V, E)$ and a capacity function $c : E \rightarrow \mathbb{R}_+$. Also given a set of demand pairs $(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)$ and demand values d_1, d_2, \dots, d_k . One wants to find a set $E' \subseteq E$ minimizing

$$\frac{c(E')}{D(E')} \rightarrow \min,$$

where

$$c(f) = \sum_{(i,j) \in E'} c_{ij}$$
$$D(f) = \sum_{i: (s_i, t_i) \text{ are separated by } E'} d_i$$

Outline

1 Combinatorial Optimization

- Max-Cut
- Min UnCut
- Sparsest Cut

2 Approaches

- Naive Algorithm
- SDP relaxations

Max-Cut, Min UnCut and Sparsest Cut problems are NP-hard

Max-Cut, Min UnCut and Sparsest Cut problems are NP-hard

Approaches:

- Use naive algorithm of discrete optimization
- Use convex relaxations

Greedy algorithm

Idea:

- On the k -th iteration choose a point x_{k+1} from neighbourhood of the current position x_k , such that

$$\text{Obj}(x_{k+1}) = \text{Obj}(x_k)$$

- If there is no such point x_{k+1} stop and return x_k

Problems:

- How to choose the neighbourhood?
- How far will be the result from the solution?

Semi-Definite-Programming

The following type of optimisation problems is considered to be the SDP problems:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^T x \\ \text{s.t.} \quad & A_0 - x_1 A_1 - \dots - x_n A_n \succeq 0 \end{aligned}$$

SDP relaxation

Suppose we have the initial problem:

$$\min_{x \in X} f(x),$$

where X is a feasible region. If we construct the SDP problem with the X' feasible region, s.t. $X \subseteq X'$, then this SDP problem is considered to be an SDP relaxation for the initial problem.

SDP relaxation with triangle constraints

In order to improve the SDP relaxation one can add triangle constraints like:

$$\begin{aligned}x_{ij} + x_{jk} + x_{ki} &\leq 2, \\ x_{ij} + x_{jk} &\geq x_{ki}.\end{aligned}$$

Such constraints are appropriate for the cut problems on graphs.

Interior Point Method

- One can find solutions of SDP-relaxations via interior-point method
- Use log-det barrier function

Thank you for attention!