

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
- LP relaxations
- First order methods

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
- LP relaxations
- First order methods

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 optimization 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 for MaxCut

The following optimization problem represents the SDP relaxation for MaxCut

$$\begin{aligned} \min_X \quad & \text{tr}(W^T X), \\ \text{s.t.} \quad & X \succeq 0, \\ & X_{ii} = 1 \quad \forall i. \end{aligned}$$

Here W is a matrix of weights.

SDP relaxation with triangle constraints

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

$$x_{ij} + x_{jk} + x_{ki} \leq 2,$$

$$x_{ij} + x_{jk} \geq x_{ki}.$$

Such constraints are appropriate for the cut problems on graphs. In these cases x_{ij} could be:

$$x_{ij} = \begin{cases} 1 & \text{if } (i, j) \text{ edge is in the cut,} \\ 0 & \text{otherwise.} \end{cases}$$

Interior Point Method

One can apply interior-point method to the problem

$$\begin{cases} \operatorname{tr}(W^T X) \rightarrow \min_{X \in \mathbb{R}^{n \times n}} \\ X \succeq 0 \\ X_{ii} = 1, \quad i = \overline{1, n} \end{cases}$$

Use log-det barrier function and solve the problem

$$\begin{cases} \operatorname{tr}(W^T X) - \mu \log \det X \rightarrow \min_{X \in \mathbb{R}^{n \times n}} \\ X_{ii} = 1, \quad i = \overline{1, n} \end{cases}$$

The duality gap is equal to μn

LP relaxations

We want to implement **LP relaxations** for Maximal Cut, Min UnCut and Sparsest cut problems.

For example:

MAXCUT can be phrased as the following integer program.

$$\begin{aligned} \max \quad & \sum_{(u,v) \in E} e_{uv} \\ & x_u \in \{0, 1\} \quad \forall u \in V \\ & e_{uv} \in \{0, 1\} \quad \forall (u, v) \in E \\ & e_{uv} \leq \begin{cases} x_u + x_v \\ 2 - (x_u + x_v) \end{cases} \quad \forall (u, v) \in E \end{aligned}$$

LP relaxation for MAXCUT

We relax $e_{uv} \in \{0, 1\}$ to $0 \leq e_{uv} \leq 1$ and $x_{u,v} \in \{0, 1\}$ to $0 \leq x_{u,v} \leq 1$ to obtain the following LP relation.

$$\begin{aligned} \max \quad & \sum_{(u,v) \in E} e_{uv} \\ & x_u \in [0, 1] \quad \forall u \in V \\ & e_{uv} \in [0, 1] \quad \forall (u, v) \in E \end{aligned}$$

$$e_{uv} \leq \begin{cases} x_u + x_v \\ 2 - (x_u + x_v) \end{cases} \quad \forall (u, v) \in E$$

First order methods

We want to implement the following first order methods for SDP relaxations:

- 1 gradient descent;
- 2 ADMM.

Alternating direction method of multipliers

ADMM problem form (with f and g convex):

$$\begin{aligned} \min_{x,z} \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax + Bz = c \end{aligned}$$

Augmented Lagrangian:

$$L_r(x, y, z) = f(z) + g(z) + y^\top (Ax + Bz - c) + \frac{r}{2} \|Ax + Bz - c\|_2^2$$

ADMM:

$$\begin{aligned} x^{k+1} &= \underset{x}{\operatorname{argmin}} L_r(x, z^k, y^k), \text{ } x \text{ - update,} \\ z^{k+1} &= \underset{z}{\operatorname{argmin}} L_r(x^{k+1}, z, y^k), \text{ } z \text{ - update,} \\ y^{k+1} &= y^k + r(Ax^{k+1} + Bz^{k+1} - c), \text{ dual update} \end{aligned}$$

Thank you for attention!

SDP relaxation for Sparsest cut Problem

Let $x_e = \mathbb{I}\{e \in E'\}$ and y_i represents whether or not the pair (s_i, t_i) should be separated, then we have the following SDP relaxation:

min t

s.t.

$$\begin{pmatrix} t & 1 \\ c^T x & d^T y \end{pmatrix} \succeq 0$$

$$\sum_{e \in p} x_e \geq y_i,$$

$$p \in \mathcal{P}_{s_i t_i}$$

$$1 \geq y_i \geq 0$$

$$i \in \overline{1, k}$$

$$1 \geq x_e \geq 0$$

$$e \in E$$