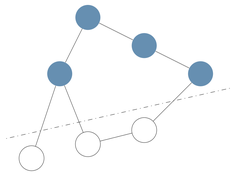
Memory-efficient approximation algorithms for MaxCut and Max-k-Cut

Nimita Shinde [1] Vishnu Narayanan [2] James Saunderson [3]

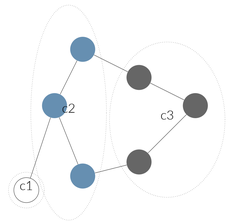
[1]IITB-Monash Research Academy [2]Indian Institute of Technology Bombay [3]Monash University

MaxCut and Max-k-Cut

MaxCut Partition nodes into two sets so that the weights of the edges crossing the partition is maximized



Max-k-Cut Partition nodes into k sets so that the weights of the edges crossing the partition is maximized



SDP Relaxation:

$$\begin{aligned} \max \quad & \langle cL_G, X\rangle \\ \text{s.t. } & \operatorname{diag}(X) = \mathbb{1}, \quad X \succeq 0 \end{aligned}$$

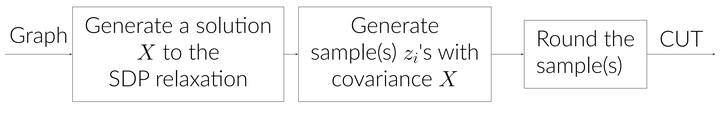
SDP relaxation:

$$\begin{aligned} \max \quad & \langle c_k L_G, X\rangle \\ \text{s.t.} \quad & X_{ij} \geq -\frac{1}{k-1} \quad i \neq j, \\ & \operatorname{diag}(X) = \mathbb{1}, X \succeq 0 \end{aligned}$$

Decision variable X uses n^2 storage!
Too large for large-scale problems

First aim: *Provide a low-memory representation of the decision variable*

SDP-based rounding method



Generation of a cut only requires $z \sim \mathcal{N}(0, X)$!

If opt is the optimal cut value

Goemans-Williamson (GW) rounding: $\mathbb{E}[\text{CUT}] \geq \alpha_{GW}\text{opt}$ [2]

Frieze-Jerrum (FJ) rounding: $\mathbb{E}[\text{k-CUT}] \geq \alpha_k\text{opt}$ [1]

Second aim: *Provide a low-memory implementation of the GW and FJ rounding that preserves approximation guarantees*

Gaussian Sampling: A way to achieve our first aim

Nonnegative weighted sum of samples $\iff$ Nonnegative weighted sum of matrices

- Let $z_t \sim \mathcal{N}(0, X_t)$ and $h_t \sim \mathcal{N}(0, H_t)$
- If $z_{t+1} = \sqrt{1-\gamma_t}\, z_t + \sqrt{\gamma_t \alpha}\, \zeta\, h_t$ for $\zeta \sim \mathcal{N}(0,1)$, then

$$\mathbb{E}\left[z_{t+1} z_{t+1}^T\right] = (1-\gamma_t)X_t + \gamma_t H_t = X_{t+1}$$

Frank-Wolfe with Gausssian Sampling (FW-GS) [3]

- Frank-Wolfe update step

$$X_{t+1} = (1-\gamma)X_t + \gamma H_t$$

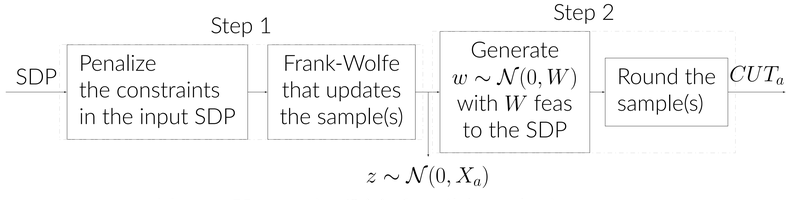
- Rank of update direction H_t at most one for the problem

$$\max_{X:\operatorname{Tr}(X)\leq\alpha, X\succeq 0} \quad g([\langle B_i, X\rangle]_{i=1}^d)$$

Not necessary to compute X_{t+1} explicitly if we use Gaussian sampling!

Gaussian sample z***:*** *a low-memory representation of* X

Low-memory implementation of GW rounding: Our second aim (Part 1)



Note: $X_a \succeq 0$ is $\mathcal{O}(\epsilon)$-feasible solution to input SDP with d linear constraints

Key features: Uses $\mathcal{O}(n+d)$ memory, preserves convergence rate of Frank-Wolfe

MaxCut using FW-GS:
$\mathbb{E}[\text{CUT}_a] \geq \alpha_{GW}(1-2\epsilon)\text{opt}$

What about Max-k-Cut? SDP relaxation of Max-k-Cut has n^2 constraints. FW-GS uses $\mathcal{O}(n^2)$ memory!

Low-memory implementation of FJ rounding: Our second aim (Part 2)

A new relaxation of Max-k-Cut

$$\begin{aligned} \max \quad & \langle c_k L_G, X\rangle \\ \text{s.t.} \quad & X_{ij} \geq -\frac{1}{k-1} \quad (i,j) \in E, i < j, \\ & \operatorname{diag}(X) = \mathbb{1}, X \succeq 0 \end{aligned}$$

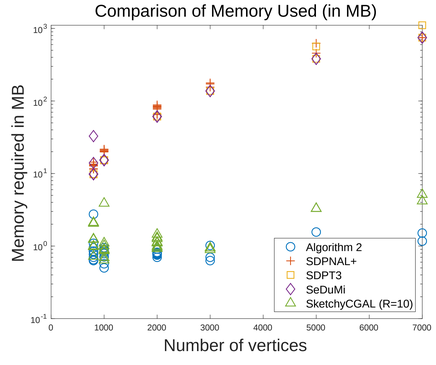
Key observation: If the rounding is applied to the new relaxation

$\mathbb{E}[\text{k-CUT}] \geq \alpha_k\text{opt}$

Max-k-Cut using FW-GS:
$\mathbb{E}[\text{CUT}_a] \geq \alpha_k(1-4\epsilon)\text{opt}$

Implementing FW-GS for MaxCut with $\epsilon = 0.1$

Specifications: Used a machine with 8GB RAM and 4 cores



- Simple to implement
- Offers scope for improvement in computational time
- Memory used is linear in n and significantly lower than standard SDP solvers

Future Directions

- What if the input graph for Max-k-Cut is dense? Sparsification of the graph to reduce the number of edges
- Are there other problems which can be solved using FW-GS?

References

[1] Alan Frieze and Mark Jerrum. Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67--81, 1997.

[2] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115--1145, 1995.

[3] Nimita Shinde, Vishnu Narayanan, and James Saunderson. Memory-efficient structured convex optimization via extreme point sampling. *arXiv preprint arXiv:2006.10945*, 2020.