

This lecture:

‘Approximation algorithms  
based on convex optimization’

We will cover:

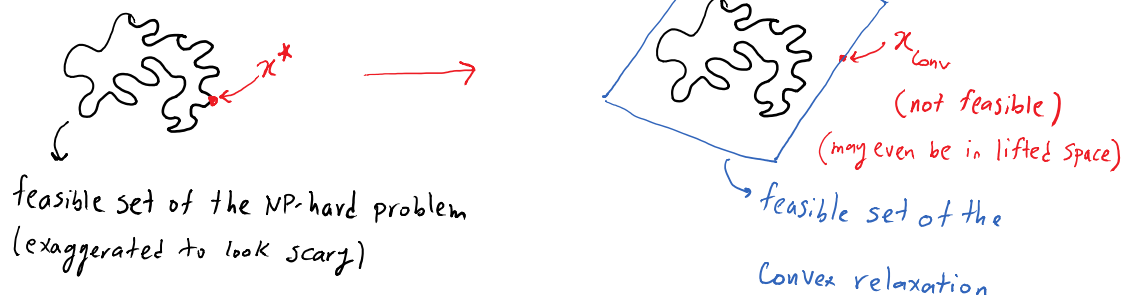
- A 2-approx. alg. for Vertex Cover based on LP (easy and as warmup)
- A .878-approx. alg. for MaxCut based on SDP  
(breakthrough result of Goemans and Williamson [GW95])
- Since we know that finding the optimal solution to an NP-hard problem in polynomial time is impossible (unless  $P=NP$ ), it is natural to ask if we can find (in poly time) a solution whose objective value is guaranteed to be within some multiplicative factor of the optimal value. This is what approx. algs. do.
- For a minimization problem with optimal value  $f^*$ , we say that algorithm  $\mathcal{A}$  is an  $\alpha$ -approximation algorithm, if it runs in polynomial time and produces a solution with objective value  $\hat{f}$ , such that  $\underline{f^* \leq \hat{f} \leq \alpha f^*}$  (where  $\alpha > 1$ ).
- For a maximization problem with optimal value  $f^*$ , we say that algorithm  $\mathcal{A}$  is an  $\alpha$ -approximation algorithm, if it runs in polynomial time and produces a solution with objective value  $\hat{f}$ , such that  $\underline{\alpha f^* \leq \hat{f} \leq f^*}$  (where  $0 < \alpha < 1$ ).

## Lec18p2, ORF523

In both cases, we want  $\alpha$  to be as close to 1 as possible. In our definitions, we also allow for "randomized algorithms". The bounds then need to hold in expectation.

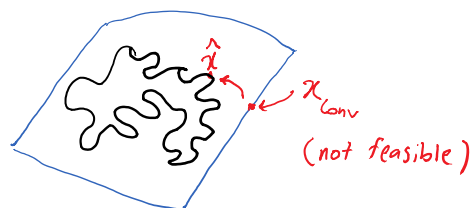
o General outline of convex optimization based approximation algorithms.

Relax:



$$f_{\text{conv}} := f(x_{\text{conv}}) \leq f^* := f(x^*) \quad (\text{for a minimization problem})$$

Round:



$\hat{x}$ : rounded solution, feasible.

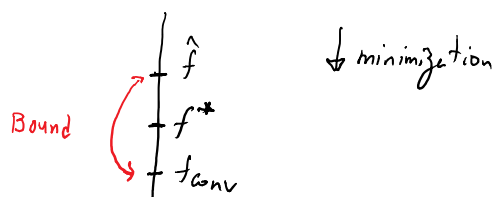
$$\text{Let } \hat{f} := f(\hat{x}).$$

Bound:

• We know  $f^* \leq \hat{f}$  (just b/c  $\hat{x}$  is feasible).

• Want to bound the gap between  $f^*$  and  $\hat{f}$ , but we have no idea what's  $f^*$ .

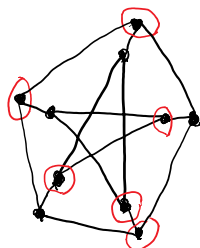
• But we know  $f_{\text{conv}} \leq f^* \Rightarrow$  Let's instead bound the gap between  $f_{\text{conv}}$  and  $\hat{f}$ . This would also be a valid bound on the ratio of  $\hat{f}$  and  $f^*$ .



## Lec18p3, ORF523

### Vertex Cover

Given an undirected unweighted graph  $G(V, E)$ , find a set of vertices of minimum size that each edge gets touched.



• Valid vertex cover b/c each edge touches at least one red node.

• In fact of minimum size.

• Finding a minimum vertex cover is NP-hard. Here's why:

• Let  $n := |V|$ ,  $\alpha(G) :=$  stability number  $VC(G) :=$  size of minimum vertex cover.

Then:  $VC(G) = n - \alpha(G)$

— why? A set of nodes  $S$  is a vertex cover  $\Leftrightarrow V \setminus S$  is a stable set  
 $\uparrow$   
convince yourself.

• We have already proved that finding  $\alpha(G)$  is NP-hard.

Vertex Cover as an integer program:

$$f^* := VC(G) = \min_x \sum_{i=1}^n x_i$$
$$x_i + x_j \geq 1 \quad \forall (i, j) \in E$$
$$x_i \in \{0, 1\} \quad i=1, \dots, n$$

## Lec18p4, ORF523

LP relaxation:

$$f_{LP} := \min \sum_{i=1}^n x_i$$
$$x_i + x_j \geq 1, \text{ if } (i,j) \in E$$
$$0 \leq x_i \leq 1 \quad i=1, \dots, n$$

Obviously  $f_{LP} \leq f^*$ . Denote the optimal solution by  $x_{LP}$ .

Rounding: Set  $\hat{x}_i = \begin{cases} 1, & \text{if } x_{LP,i} \geq 1/2. \\ 0 & \text{otherwise} \end{cases}$

•  $\hat{x}$  gives a valid vertex cover b/c  $\forall$  edge, one of the two end nodes in the LP solution must be  $\geq 1/2$ .

• So  $f^* \leq \hat{f} := \sum_i \hat{x}_i$

Bounding:

$$\hat{f} \leq 2 f_{LP}$$

b/c in worst case, we are changing a bunch of " $1/2$ 's" to "1's".

$$\Rightarrow \hat{f} \leq 2 f^*$$

$$\text{b/c } f_{LP} \leq f^*$$

Overall:

$$f^* \leq \hat{f} \leq 2 f^*$$

This is the best approximation ratio known to date!

## Max Cut

Given an undirected graph  $G(V, E)$  with nonnegative edge weights  $w_{ij}$ , find a partition of the nodes into two disjoint sets  $V_1$  and  $V_2$  ( $V_1 \cap V_2 = \emptyset$ ,  $V_1 \cup V_2 = V$ ) such that the sum of the edge weights going from  $V_1$  to  $V_2$  is maximized.

- Finding the Max Cut value of a graph is NP-hard (e.g., there's a relatively straight forward reduction from 3SAT).
- Contrast this with Min Cut, which we argued can be solved in poly-time by linear programming.
- We will now produce a (randomized) solution for Max Cut (in poly-time), which in expectation is 87% optimal!
- Denote the Max Cut value of your graph by  $f^*$ :

$$f^* = \max \frac{1}{4} \sum_{i,j} w_{ij} (1 - x_i x_j) = \frac{1}{4} \sum_{i,j} w_{ij} - \frac{1}{4} \underbrace{\left[ \min_{\substack{\sum_{i,j} w_{ij} x_i x_j \\ \text{s.t. } x_i^2 = 1}} \right]}_{:= f_2^*}$$

Define a matrix  $Q \in \mathbb{S}^{n \times n}$  (where  $n = |V|$ ) as  $Q_{ij} = \begin{cases} 0 & i=j \\ w_{ij} & i \neq j \end{cases}$

## Lec18p6, ORF523

$$\text{Then, } f_2^* = \min x^T Q x \\ \text{s.t. } x_i^2 = 1.$$

Here's the standard SDP relaxation for this problem:

$$f_{2\text{SDP}} := \min_{X \in S^{n \times n}} \text{Tr}(QX) \\ X_{ii} = 1 \quad (\text{with a constraint } \text{rank}(X) = 1, \text{ this} \\ X \succeq 0 \quad \text{would be an equivalent formulation})$$

Clearly,  $f_{2\text{SDP}} \leq f_2^*$ .

Rounding Step.

- o If the optimal solution of the SDP is rank-1, you are happy and you go home.
- o If not, take the Cholesky factorization of the optimal solution  $X$ :

$$X = \begin{matrix} V^T \\ n \times n \end{matrix} \begin{matrix} V \\ n \times r \end{matrix} \begin{matrix} \\ r \times n \end{matrix}, \quad \text{where } r = \text{rank}(X).$$

o Denote the columns of  $V$  by  $v_i \in \mathbb{R}^r$ :  $V = [v_1, \dots, v_n]$

o Observe that  $X_{ij} = v_i^T v_j$

o So  $\|v_i\| = 1 \quad \forall i$  (b/c  $X_{ii} = 1$  must hold).

o So we have  $n$  points  $v_1, \dots, v_n$  on the unit sphere  $S^{r-1}$  in  $\mathbb{R}^r$ .

o Generate a point  $p \in S^{r-1}$  uniformly at random (e.g.,  $p = \text{randn}(r, 1)$ ;  $p = p / \text{norm}(p, 2)$ );

o Set  $x_i = \begin{cases} 1 & \text{if } p^T v_i \geq 0 \\ -1 & \text{if } p^T v_i < 0 \end{cases} \quad i = 1, \dots, n.$

o That's it.

# Lec18p7, ORF523

Bounding:

Consider the hyperplane  $\mathcal{P} := \{x \in \mathbb{R}^n \mid p^T x = 0\}$

Let  $\hat{f}_2$  denote the expected value of the objective value of our rounded solution:

$$\hat{f}_2 = E \left[ \sum_{i,j} w_{ij} x_i x_j \right] = \sum_{i,j} w_{ij} E[x_i x_j]$$

$$E[x_i x_j] = 1 - \Pr[v_i, v_j \text{ on same side of } \mathcal{P}] - \underbrace{1 \cdot \Pr[v_i, v_j \text{ on different sides of } \mathcal{P}]}_{\frac{\theta_{ij}}{\pi} = \frac{1}{\pi} \arccos(v_i^T v_j)}$$

(i ≠ j)

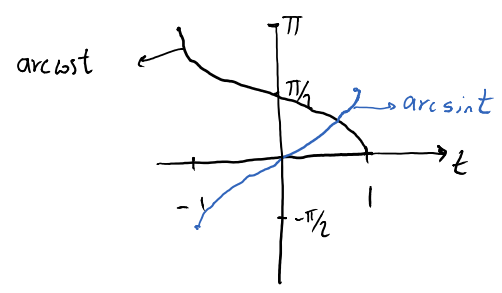
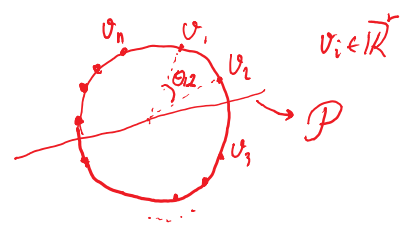
$$= 1 - \frac{\theta_{ij}}{\pi} - \frac{\theta_{ij}}{\pi}$$

$$= 1 - \frac{2}{\pi} \arccos v_i^T v_j$$

Well-defined b/c  $X_{ij} \leq 1$  (why?)

$$= \frac{2}{\pi} \arcsin v_i^T v_j$$

$$\arcsin t + \arccos t = \frac{\pi}{2}$$



$$\Rightarrow \hat{f}_2 = \frac{2}{\pi} \sum_{i,j} w_{ij} \arcsin X_{ij}$$

o Recall that  $f^* = \frac{1}{4} \left( \sum_{i,j} w_{ij} - f_2^* \right)$

o Let  $\hat{f} := \frac{1}{4} \left( \sum_{i,j} w_{ij} - \hat{f}_2 \right) = \frac{1}{4} \left( \sum_{i,j} w_{ij} - \frac{2}{\pi} \sum_{i,j} w_{ij} \arcsin X_{ij} \right)$

$$= \frac{1}{4} \sum_{i,j} w_{ij} \left[ 1 - \frac{2}{\pi} \arcsin X_{ij} \right] = \frac{1}{4} \cdot \frac{2}{\pi} \sum_{i,j} w_{ij} \arccos X_{ij}$$


---

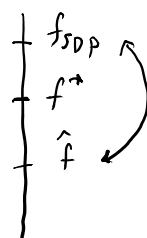
# Lec18p8, ORF523

We want to relate this to the optimal value of the SDP:

$$f_{SDP} := \frac{1}{4} \left( \sum_{ij} w_{ij} - f_{2SDP} \right)$$

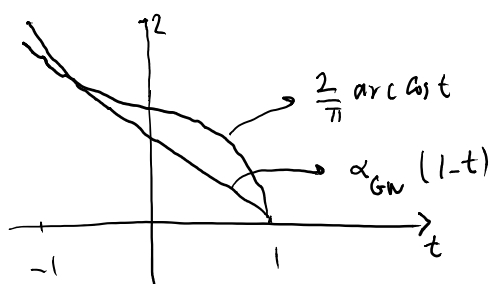
$$= \frac{1}{4} \sum_{ij} w_{ij} - \frac{1}{4} \sum_{ij} w_{ij} X_{ij} = \frac{1}{4} \sum_{ij} w_{ij} (1 - X_{ij})$$

• Want to argue:  $\alpha f_{SDP} \leq \hat{f}$   
for  $\alpha$  as large as possible.



• We will bound term by term (since  $w_{ij} \geq 0$ ). So we need the largest  $\alpha$  for which:

$$\alpha(1-t) \leq \frac{2}{\pi} \arccos t \quad \forall t \in [0, 1]$$



Optimal  $\alpha$ :  $\alpha_{GW} \approx 0.878$



his car  
(before the  
algorithm) →  
True  
story!





## Lec18p9, ORF523

### Notes

Further reading for this lecture can include Chapter 7 of [LV12] and Chapter 3 of [BN01].

### References

- [GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maxcut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 1995.
- [Pa14] P.A. Parrilo. *Lecture notes on Algebraic Techniques and Semidefinite Optimization*, MIT, 2014.
- [BN01] A. Ben-Tal and A. Nemirovski. *Lecture Notes on Modern Convex Optimization*. MPS/SIAM Series on Optimization, 2001.
- [LV12] M. Laurent and F. Vallentin. *Lecture Notes on Semidefinite Optimization* 2012.