Randomized Rounding

- Introductory Example: Set cover
- Using Randomized Rounding for a proof
- A famous example: Maximum Cut

- MAX-SAT was presented in the exercises.

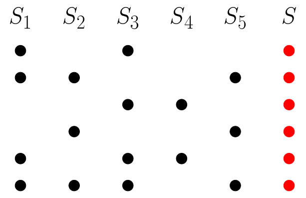
Approximation Algorithms

- Natural way to approach computationally hard problems
- Problem: Finding the optimum OPT is too hard
- Approximation algorithm: finds a (suboptimal) feasible solution Z.
- An algorithm is called ϵ-approximation algorithm, iff
  - $Z \geq \epsilon OPT$ for a maximization problem ($\epsilon \leq 1$).
  - $Z \leq \epsilon OPT$ for a minimization problem ($\epsilon \geq 1$).

Set Cover

Input: A set $(S_i)_{i \in I}$ of subsets of a ground set S.

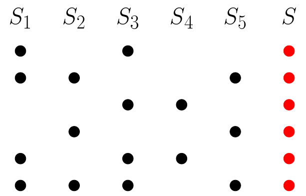
Output: A subset $I' \subseteq I$ such that for each $j \in S$, there is $i \in I'$ with $j \in S_i$.



Set Cover Integer Linear Program

We have a binary variable x_i for every set S_i indicating whether we chose S_i in our set cover.

$$\begin{array}{rlcll} \min & \sum_{i \in I} c_i x_i & & & \\ s.t. & \sum_{i \in I \mid j \in S_i} x_i & \geq & 1 & \forall j \in S \\ & x_i & \in & \{0,1\} & \end{array}$$



Rounding Algorithm

- Let x^* be the optimal solution of the Linear Program, i.e. the ILP with $x_i \in \{0,1\}$ replaced by $0 \le x_i \le 1$.
- Include S_i in the set cover with probability x_i^*, i.e. $x_i = 1$ with probability x_i^* and $x_i = 0$ otherwise.
- Expected cost: $E[\sum_{i\in I} c_i x_i] = \sum_{i \in I} c_i E[x_i] = \sum_{i\in I} c_i x_i^*$, i.e. the LP solution.
- Probability of not covering $j \in S$ is at most $(1-1/n)^n \le 1/e$.

Repeated Rounding

Idea: Perform k rounds of rounding and include all sets which were selected in at least one round.

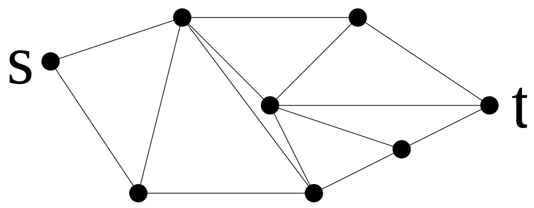
Expected Cost: k times the LP-solution.

Number of Rounds: until all elements are covered?

- After $O(\log n)$ rounds, the probability of covering the first element is at least $1/(2n)$.
- Thus, the probability of covering all elements is at least $n/2n = 1/2$

Thus we have a $O(\log n)$ approximation algorithm for set cover.

s-t Cuts in Graphs



Input: An undirected weighted graph $G = (V, E, c)$ and two nodes s, t of G with $c > 0$.

Output: The minimal $s - t$ cut, i.e. the subset $\{s\} \subseteq S \subseteq V \setminus \{t\}$ which minimizes $\sum_{uv \in E | u \in S, v \notin S} c_{uv}$.

Integer Linear Program

We have a binary variable x_v for every node $v \in V$, with $v \in S$, iff $x_v = 0$ and
a binary variable z_{uv} for every edge $uv \in E$, indicating whether uv crosses the cut.

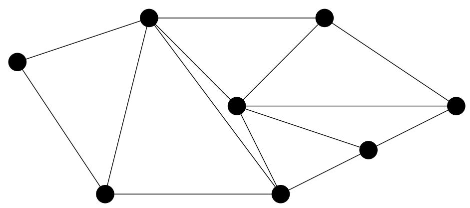
$$
\begin{array}{lrcll}
\min & \multicolumn{4}{l}{\sum_{uv \in E} c_{uv} z_{uv}} \\
s.t. & z_{uv} & \ge & x_u - x_v & \forall uv \in E \\
 & z_{uv} & \ge & x_v - x_u & \forall uv \in E \\
 & x_s & = & 0 & x_t = 1 \\
 & x_u, z_{uv} & \in & \{0,1\} &
\end{array}
$$

Note that in the optimal LP solution $z_{uv} = |x_u - x_v|$.

A Rounding Procedure

- Let (x^*, z^*) be the optimal LP-solution.
- Chose a random $y \in [0, 1)$.
- Let $v \in S$, iff $x_v^* \leq y$, i.e. $x_v = 0$, if $x_v^* \leq y$ and $x_v = 1$ otherwise.
- Let $z_{uv} = 1$, iff $x_u \neq x_v$.

A Famous Example: Max-Cut



Input: An undirected weighted complete graph $G = (V, E, c)$ with $c \geq 0$.

Output: The maximal cut, i.e. the subset $S \subseteq V$ which maximizes $\sum_{uv \in E | u \in S, v \in T} c_{uv}$.

Assumption: The nodes are ordered.

Analysis

- Note that an edge uv crosses the cut iff $y \in [\min\{x_u^*, x_v^*\}, \max\{x_u^*, x_v^*\})$.
- Expected cost: $E[\sum_{uv \in E} c_{uv} z_{uv}] = \sum_{uv \in E} c_{uv} E(z_{uv}) = \sum_{uv \in E} c_{uv} z_{uv}^*$
- Thus the expected cost of the rounded solution has the same value as the optimal LP solution.
- Since the LP solution has at most the cost of any rounded solution, all rounded solutions must have the same value, namely the cost of the LP solution.

Quadratic Programming Formulation

For every node $v \in V$ we introduce a variable y_v with value -1 if $v \in S$ and 1 if $v \notin S$.

$$\begin{array}{llll} \max & \frac{1}{2} \sum_{u<v} c_{uv}(1 - y_u y_v) & & \\ s.t. & y_u \in & \{-1, 1\} & \forall u \in V \end{array}$$

A Relaxation (typical for Semidefinite Programming)

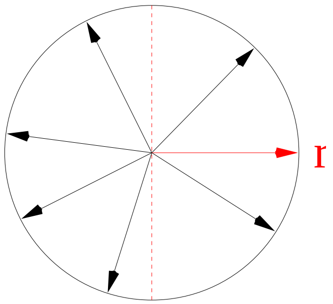
Quadratic Program:

$$\max \quad \frac{1}{2}\sum_{u<v} c_{uv}(1 - y_u y_v)$$
$$s.t. \quad y_u \in \{-1,1\} \quad \forall u \in V$$

We replace every variable y_u in the Quadratic Program by a vector $x_u \in \mathbb{R}^n$. S_n denotes the unit-sphere in $\mathbb{R}^n$.

$$\max \quad \frac{1}{2}\sum_{u<v} c_{uv}(1 - x_u \cdot x_v)$$
$$s.t. \quad x_u \in S_n \quad \forall u \in V$$

A Rounding Procedure



- Solve the relaxation.
- Chose a random $r \in S_n$ (details ignored).
- Set $S = \{v \in V \mid y_v \cdot r \geq 0\}$.

On the relaxation

$$\max \quad \frac{1}{2}\sum_{u<v} c_{uv}(1 - x_u \cdot x_v)$$
$$s.t. \quad x_u \in S_n \quad \forall u \in V$$

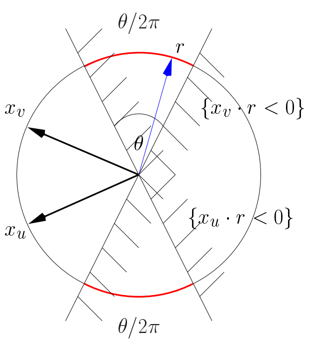
- It is a relaxation
  - Let $u \in V$.
  - Let y^* be a solution of the Quadratic Program.
  - Define $x_v = y^*_v e_u$.
  - x is feasible and has the same objective function value.
- The relaxation can be solved.
  - The program is equivalent to a semidefinite program.
  - We will see this later.

Analysis

- Let W denote the value of the cut we produced.
- We will show $E[W] = \sum_{u<v} c_{uv} \arccos(x_u \cdot x_v)/\pi$.
- Let $\alpha = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi}\frac{\theta}{1-\cos\theta}$.
- Thus we have: $E[W] \leq \alpha \frac{1}{2}\sum_{u<v} c_{uv}(1 - x_u \cdot x_v)$.
- We can show $\alpha > 0.87856$.

$E[W] = \sum_{u<v} c_{uv} \arccos(x_u \cdot x_v)/\pi$

- $E[W] = \sum_{u<v} c_{uv} P[sgn(x_u \cdot r) \neq sgn(x_v \cdot r)]$.
- Let $\theta = \arccos(x_u, x_v)$.
- The intersection of $\{x_u \cdot r \geq 0, x_v \cdot r < 0\}$ with the unit sphere covers exactly $\theta/2\pi$ of the area.



The Semidefinite Program

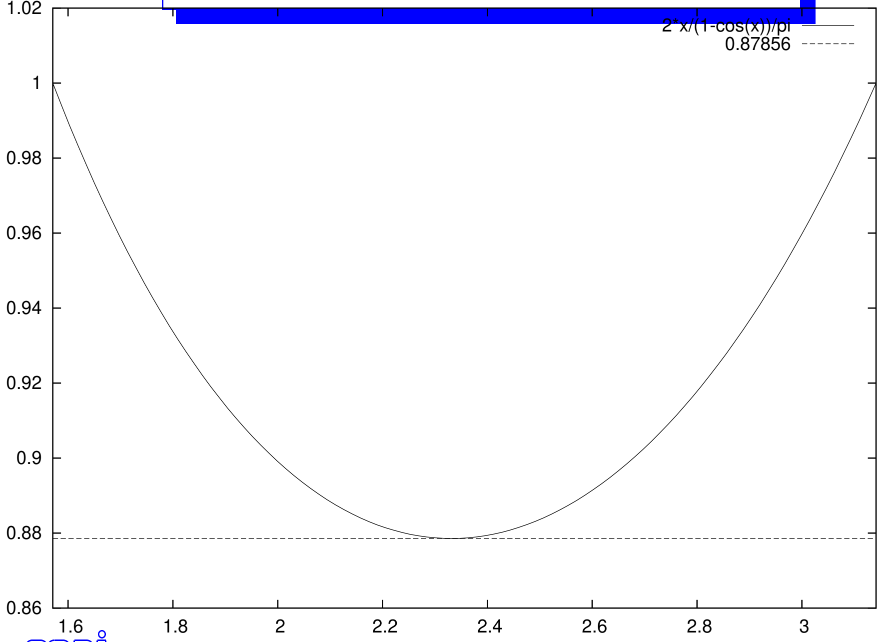
$$\max \quad \frac{1}{2}\sum_{u<v} c_{uv}(1 - x_u \cdot x_v)$$
$$s.t. \quad x_u \in S_n \quad \forall u \in V$$

- For every $x_u \cdot x_v$ we introduce a variable y_{uv}.
- Let Y be symmetric. From algebra:
  Y is positive semidefinite $\Leftrightarrow$ Y can be written as $Y = B^T B$
- Also: If $y_{uu} = 1$ for all u then the columns of B have unit length.

$$\max \quad \frac{1}{2}\sum_{uv \in E} c_{uv}(1 - y_{uv})$$
$$s.t. \quad y_{uu} = 1 \quad \forall u \in V$$

Y symmetric positive semidefinite.

$\alpha = \min_{0 \leq \theta \leq \pi} \frac{2}{\pi} \frac{\theta}{1-\cos\theta} > 0.87856$



Summary

Theorem:

A Semidefinite Program can be approximated within $(1-\epsilon)$ in time polynomial in the input size and $\log(1/\epsilon)$.

Theorem:

There is a polynomial time algorithm to find an 0.87856-approximation of the maximum cut.

Note that $0.87859 < \alpha$ and thus

1) An approximate solution of the semidefinite program suffices.
2) A solution close to the expected value can be found.