**Algo.** Assign each vertex independently and uniformly at random to one of the two sides.

**Theorem 5.2** The algo. above is a 1/2-approximation for the Max Cut problem.

**Proof:** $\Pr(\boxed{\text{edge } e \text{ in cut}}) = 1/2$ $\quad \Pr(i \in U \,\&\, j \in W) + \Pr(i \in W \,\&\, j \in U) = \frac{1}{2}\cdot\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{2}.$

Let $Z$ be the total # of edges in the cut found by the algo.

Then $E(Z) = \sum_{e \in E} \Pr(e \text{ in cut}) = \frac{1}{2}|E| \geq OPT/2$

For the weighted version: let $W$ be the total weight of the edges.

$$E(Z) = \sum_e w_e/2 = W/2 \geq OPT/2$$

## 5.2. Derandomization

Max SAT (unweighted)

Idea: Assign the variables in the order $x_1, x_2, \cdots, x_n$, each time choose the value (T/F) which maximizes the expected value when all remaining variables are assigned at random.

Denote:
- $S_i$: the assignment for $x_1, \cdots, x_i$
- $Z$: # of satisfied clauses
- $E(Z|S_i)$: expected value of $Z$ when $x_{i+1}, \cdots x_n$ are assigned at random and given the assignment $S_i$.

Then, $E(Z|S_i) = \frac{1}{2}E(Z|S_i \,\&\, x_{i+1} = T) + \frac{1}{2}E(Z|S_i \,\&\, x_{i+1} = F)$.

**Algo.** For $i = 0, \cdots, n-1$:

  if $E(Z|S_i \text{ and } x_{i+1} = T) \geq E(Z|S_i \,\&\, x_{i+1} = F)$:

    $x_{i+1} = T$

  else $x_{i+1} = F$

**Theorem 5.3** The algo. above is a deterministic 1/2-approximation.

**Proof** 1. Polynomial: ✓. since the conditional expectations can be computed in polynomial time

2. Feasible: ✓

3. Expected Ratio: $E(Z|S_i) = \boxed{\frac{1}{2}E(Z|S_i \,\&\, x_{i+1} = T)} + \boxed{\frac{1}{2}E(Z|S_i \,\&\, x_{i+1} = F)}$

$\Rightarrow E(Z|S_{i+1}) = \max\{E_1, E_2\} \geq E(Z|S_i)$

$E(Z|S_n) \geq E(Z|S_{n-1}) \geq \cdots \geq E(Z|S_1) \geq E(Z) \geq OPT/2$

Max Cut (unweighted).

Idea: similar to Max SAT.: Assign the vertices one by one.

$E(Z|S_i) = \frac{1}{2}E(Z|S_i \,\&\, v_{i+1} \in U) + \frac{1}{2}E(Z|S_i \,\&\, v_{i+1} \in W\}$.

**Algo.** For $i = 0, \cdots, n-1$:

  if $E(Z|S_i \,\&\, v_{i+1} \in U) \geq E(Z|S_i \,\&\, v_{i+1} \in W\}$:

    $v_{i+1} \to U$

  else: $v_{i+1} \to W$

**Theorem 5.4** The algo. above is a 1/2-approximation for Max Cut.

Skip the proof since similar to Max SAT.

## 5.3 Randomized rounding for Max SAT.

(ILP) $\max \quad Z = \sum_{j=1}^{m} w_j z_j$.

$$\text{s.t.} \quad \sum_{i \in P_j} y_i + \sum_{i \in N_j} (1 - y_i) \geq z_j \quad \forall j.$$

$y_i \in \{0, 1\} \quad \forall i$ $\quad$ 1: $X_i$ = TRUE

$z_j \in \{0, 1\} \quad \forall j$. $\quad$ 1: clause $j$ is satisfied.

For relaxation, replace the last two constraints by $0 \leq y_i \leq 1$ & $0 \leq z_j \leq 1$.

**Algo.**: Step 1: Solve LP $\to y^*, z^*, Z^*_{LP}$

Step 2: Assign at random: $\Pr(X_i = \text{TRUE}) = \boxed{y_i^*}$

**Theorem 5.5** The algo. above is a $(1 - \frac{1}{e})$-approximation. $(1 - \frac{1}{e} \approx 0.63)$

**Proof:** Consider an arbitrary clause $C_j$ and let $l_j$ be the # of literals

$\Pr(\boxed{C_j \text{ is not satisfied}})$

$= \prod_{i \in P_j}(1 - y_i^*) \prod_{i \in N_j} y_i^*$ $\quad$ $P_j / N_j$ = positive/negative literals of clause $j$

$\overset{①}{\leq} \left[\frac{1}{l_j}\left(\sum_{i \in P_j}(1 - y_i^*) + \sum_{i \in N_j} y_i^*\right)\right]^{l_j}$ $\quad$ ① $\left(\prod_{i=1}^{k} a_i\right)^{1/k} \leq \frac{1}{k}\sum_{i=1}^{k} a_i \Rightarrow \prod_{i=1}^{k} a_i \leq \left(\frac{1}{k}\sum_{i=1}^{k} a_i\right)^k$

$\overset{②}{=} \left[\boxed{1 -} \frac{1}{l_j}\boxed{\left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j}(1 - y_i^*)\right)}\right]^{l_j}$ $\quad$ ② rearranging terms, $|P_j| + |N_j| = l_j$

$\overset{③}{\leq} \left[1 - \frac{1}{l_j} z_j^*\right]^{l_j}$ $\quad$ ③ LP - ~~constrai~~ inequality

Thus, $\Pr(\boxed{C_j \text{ is SAT}}) \geq 1 - \left[1 - \frac{1}{l_j} z_j^*\right]^{l_j}$ 括号内

Observe that function $f(z) = 1-(1-\frac{1}{l_j}z)^{l_j}$ is concave on $[0,1]$.

Thus $f(z) \geq f(0) + (f(1)-f(0))z = (1-[1-\frac{1}{l_j}]^{l_j})z$

$$\Rightarrow \Pr(C_j \text{ is SAT}) \geq (1-[1-\frac{1}{l_j}]^{l_j}) z_j^* \geq (1-\frac{1}{e}) z_j^* \text{ for } l_j \geq 1$$

括号外

Therefore

$$E(W) = \sum_{j=1}^m w_j \Pr(C_j \text{ is SAT}) \geq \sum_{j=1}^m w_j (1-\frac{1}{e}) z_j^* = (1-\frac{1}{e}) Z^*_{LP} \geq (1-\frac{1}{e}) OPT.$$

## 5.4. Choosing the better of two solutions.

Let $W_1$ and $W_2$ be the weight of the solution for the algo. of ~~section~~ section 5.1 and 5.3.

**Theorem 5.6** $E[\max(W_1, W_2)] \geq 3/4$ OPT.

**Proof** $E[\max(W_1, W_2)] \geq \frac{1}{2}E(W_1) + \frac{1}{2}E(W_2)$

$$\geq \frac{1}{2}\sum_{j=1}^m (1-2^{-l_j}) w_j + \frac{1}{2}\sum_{j=1}^m (1-(1-\frac{1}{l_j})^{l_j}) w_j z_j^*$$

$$\geq \frac{1}{2}\sum_{j=1}^m \left((1-2^{-l_j}) + 1 - (1-\frac{1}{l_j})^{l_j}\right) w_j z_j^*$$

The last inequality above follows from $\boxed{z_j^* \leq 1}$.

Note,

$$(1-2^{-l_j}) + 1 - (1-\frac{1}{l_j})^{l_j} \geq \begin{cases} = 3/2 & \text{for } l_j = 1 \text{ or } 2 \\ > \frac{7}{8} + 1 - 1/e > \frac{3}{2} & \text{for } l_j \geq 3 \end{cases}$$

Thus, $E[\max(W_1, W_2)] \geq \frac{1}{2}\sum_{j=1}^m \frac{3}{2} w_j z_j^* = \frac{3}{4}\sum_{j=1}^m w_j z_j^* = \frac{3}{4} Z^*_{LP} \geq \frac{3}{4}$ OPT.

## 5.5. Non-linear randomized rounding

Let $1-4^{-y} \leq f(y) \leq 4^{y-1}$

**Algo.** Step 1: Solve LP $\rightarrow y^*, z^*, Z^*_{LP}$.

Step 2: Assign at random : $\Pr(x_i = \text{TRUE}) = \boxed{f(y_i^*)}$

**Theorem 5.7** The algo. above is a 3/4 − approximation algo.

**Proof** $\Pr(C_j \text{ is not satisfied}) = \prod_{i \in P_j}(1-f(y_i^*)) \prod_{i \in N_j} f(y_i^*)$

$$\leq \prod_{i \in P_j} 4^{-y_i^*} \prod_{i \in N_j} 4^{y_i^*-1}$$

$$\leq 4^{-z_j^*}$$

Thus, $\Pr(C_j \text{ is SAT}) \geq 1 - 4^{-z_j^*}$.

Similar to 5.3. $g(z) = 1-4^{-z}$ is concave function on $[0,1]$.

Thus. $g(z) \geq g(0) + (g(1)-g(0))z = (1-\frac{1}{4})z = \frac{3}{4}z$

$$\Rightarrow E(W) = \sum_{j=1}^m w_j \Pr(C_j \text{ is SAT}) \geq \sum_{j=1}^m w_j \frac{3}{4} z_j^* = \frac{3}{4} Z^*_{LP} \geq \frac{3}{4} OPT.$$

## 5.6 Prize-collecting Steiner tree.

We take $\alpha \in [\gamma, 1]$ uniformly at random. The value $\gamma$ is chosen appropriately later.

**Algo.** Step 1: solve LP $\rightarrow x^*, y^*, Z^*_{LP}$

Step 2: let $U = \{i \mid y_i^* \geq \alpha\}$, construct a Steiner tree $T$ on $U$.

**Lemma 1.** The expected connection cost for the Steiner tree $T$ is

$$E(\sum_{e \in T} c_e) \leq \frac{2}{1-\gamma} \ln\frac{1}{\gamma} \sum_{e \in E} c_e x_e^*$$

**Proof** From 4.3, the expected connection cost is at most

$$E(\frac{2}{\alpha}\sum_{e \in E} c_e x_e^*) = E(\frac{2}{\alpha})\sum_{e \in E} c_e x_e^*$$

$$E(\frac{2}{\alpha}) = \frac{1}{1-\gamma}\int_{\alpha=\gamma}^{\alpha=1} \frac{2}{\alpha}\, d\alpha = \frac{2}{1-\gamma}\ln\frac{1}{\gamma}$$

**Lemma 2.** The expected total penalty cost is at most

$$\frac{1}{1-\gamma}\sum_{i \in V} \pi_i (1-y_i^*)$$

**Proof** Note that $\Pr(i \notin U) = \begin{cases} 1 & \text{if } y_i^* \leq \gamma \\ \frac{1-y_i^*}{1-\gamma} & \text{if } y_i^* \geq \gamma. \end{cases}$

In both cases, the probability is at most $(1-y_i^*)/(1-\gamma)$.

The expected penalty cost is at most

$$\sum_{i \in V} \pi_i \Pr(i \notin U) \leq \sum_{i \in V} \pi_i \frac{1-y_i^*}{1-\gamma} = \frac{1}{1-\gamma}\sum_{i \in V} \pi_i (1-y_i^*)$$

**Theorem. 5.8** The algo. above with $\gamma = e^{-\frac{1}{2}}$ is a $\frac{1}{1-e^{-1/2}}$-approximation algo. for Prize-collecting Steiner tree.

**Proof** The total cost expected of the solution is at most

$$\max\left\{\frac{2}{1-\gamma}\ln\frac{1}{\gamma}, \frac{1}{1-\gamma}\right\} Z^*_{LP}$$

The maximum above is $\frac{1}{1-e^{-1/2}} \approx 2.54$ for $\gamma = e^{-\frac{1}{2}}$.

## Derandomization

Assume (w.l.o.g.) that $y_1^* \le y_2^* \le \cdots \le y_n^*$

Observation: when increasing $\alpha$ from $\gamma$ to 1, the solution only changes at points $y_i^*$. Hence, there are at most $n+1$ different outcomes possible. It is enough to try only $n+1$ different values of $\alpha$ and take the best solution found.

Algo. Step 1: Solve LP $\to x^*, y^*, Z^*_{LP}$.

Step 2: For all $\alpha \in \{0, y_1^*, \cdots, y_n^*, 1\}$ do the following:

Let $U = \{i \mid y_i^* \ge \alpha\}$. Construct a Steiner tree $T$ on $U$

Step 3: Return the best solution found.

## 5.7 Uncapacitated facility location.

Algo. For $k=1,2,\cdots$ until all clients are assigned do:

Step 1: Among the unconnected clients choose client $j_k$ with smallest $v_{j_k}^* + C_{j_k}^*$ [Define the fractional connection cost of client $j$ as $C_{j_k}^* = \sum_{i\in F} c_{ij} x_{ij}^*$]

Step 2: Choose facility $i_k \in N(j_k)$ at random, where $P_r(i=i_k) = x_{ij}^*$

Step 3: Connect all still unconnected clients in $N^2(j_k)$ to facility $i_k$.

Theorem 5.9 Algo. above is a randomized 3-approximation algo.

Proof The expected opening cost for facility opened in iteration $k$ is

$$\sum_{i\in N(j_k)} f_i x_{ij_k} \le \sum_{i\in N(j_k)} f_i y_i^*$$

The total expected opening cost is

$$\sum_k \sum_{i\in N(j_k)} f_i y_i^* \le \sum_{i\in F} f_i y_i^*$$

For the expected connection cost, consider an arbitrary iteration $k$. The distance $C_{hj_k} + C_{hl}$ are not random variables. However, the distance $C_{i_k j_k}$ is a random variable since the facility $i_k$ is chosen at random.

$$E(C_{i_k j_k}) = \sum_{i\in N(j_k)} x_{ij_k}^* c_{ij_k} = \sum_{i\in F} x_{ij_k}^* c_{ij_k} = C_{j_k}^*.$$

The algo. assigns client $l$ to facility $i_k$. The expected connection cost for client $l$ is

$$\begin{aligned} E(C_{i_k l}) &\overset{①}{\le} E(C_{i_k j_k} + C_{hj_k} + C_{hl}) \\ &\overset{②}{=} E(C_{i_k j_k}) + C_{hj_k} + C_{hl} \\ &= C_{j_k}^* + C_{hj_k} + C_{hl} \\ &\overset{③}{\le} C_{j_k}^* + v_{j_k}^* + v_l^* \\ &\overset{④}{\le} C_l^* + v_l^* + v_l^* \end{aligned}$$

① Triangle inequality.

② Only the first is a random variable.

③ Complementary slackness

④ From Step 1. Note $j_k$ and $l$ were both unassigned before $j_k$ was chosen.

The total expected value

$$\sum_{i\in F} f_i y_i^* + \sum_{l\in D} C_l^* + 2\sum_{l\in D} v_l^*$$

$$= \left(\sum_{i\in F} f_i y_i^* + \sum_{l\in D}\sum_{i\in F} x_{il}^* c_{il}\right) + 2\sum_{l\in D} v_l^*$$

$$= Z^*_{LP} + 2Z^*_D \le 3Z^*_{LP} \le 3OPT.$$

# Chapter 6. Random Rounding of Semidefinite Programs.

## 6.1 Semidefinite Programming

The general form ~~with~~ of an SDP with $m$ linear constraints is

$$\begin{aligned} \text{min or max} \quad & \sum_{ij} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{ij} a_{ij_k} x_{ij} = b_k \quad \text{for } k=1,\cdots,m \\ & x_{ij} = x_{ji} \quad \forall i,j \\ & X = (x_{ij}) \succeq 0 \end{aligned}$$

The general form of a vector program is

$$\begin{aligned} \text{min or max} \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} \cdot v_i \cdot v_j \\ \text{s.t.} \quad & \sum_{i=1}^n \sum_{j=1}^n a_{ij_k} \cdot v_i \cdot v_j = b_k \quad \text{for } k=1,\cdots,m \\ & v_i \in \mathbb{R}^n, \quad \text{for } i=1,\cdots,n. \end{aligned}$$