Algorithms – Ex6 solution

MAX-CUT

1) Define random variables $X_1, \dots X_{|E|}$ as follows:

for each $i \in E$:

$X_i = 1$ if i is an edge between *A* and *B*.
$X_i = 0$ if i is an edge whose two vertices are in the same set.

The expectation of each X_i is:

$$(X_i) = 1 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}$$

The size of a cut is therefore defined: $q(O) = \sum_{i \in E} x_i$, where x_i is the value of X_i .

$$[q(O)] = \ (\sum_{i \in E} x_i) = \sum_{i \in E} \ (X_i) = \sum_{E} \frac{1}{2} = \frac{|E|}{2}$$

where the second equation is due to the expectation's linearity.

2) 2-approximation algorithm for MAX-CUT:

In order to remove the randomness we replace the exponential sample space with a polynomial sample space.

As proven in class – it is possible to create a *k*-wise independent sample space of $\{0,1\}^n$ of size $O(n^k)$ in a polynomial time. In our case *k*=2.

Once we have this sample space, we search it until we find a cut of size ¿ $\frac{|E|}{2}$. The sample space is of polynomial size, which means the search will take a polynomial time, and we are assured to find such a cut, as the expectation is $\frac{|E|}{2}$, as proven in section a.

$q(Opt) \le |E|$ and therefore this is a 2-approximation solution.

3) A greedy 2-approximation algorithm for MAX_CUT:

i. **Initialization:** Order the vertices arbitrarily and insert v_1 into *A*.

ii. **Greedy step:** For each vertex in the list insert it to the set which will add more edges to the cut: If v_i has more edges with vertices in *A* than with vertices in *B*– add it to *B* and vice versa.

Claim: This algorithm is 2-approxiamtion.

Proof:

Let q_i be the number of edges added to the cut in step i of the algorithm and let q'_i be the number of edges we "lost" in step *i*, i.e., the number of edges that stayed in the same set in that step.

As the algorithm is greedy, $\forall i \quad q_i \geq q'_i$ and therefore $q(O) = \sum_{i=1}^{n} q_i \geq \sum_{i=1}^{n} q'_i$.

Since $|E| = \sum_{i=1}^{n} q_i + \sum_{i=1}^{n} q'_i$ we get that always $q(O) \geq \frac{|E|}{2}$ and therefore

$[q(O)] \geq \frac{|E|}{2}$

Polynomial algorithm for *M*-approximation for MIN-HITTING-SET:

We construct the 0-1 LP problem that is equivalent to the MIN-HITTING-SET problem:

Define for $j=1,\ldots,n$:

$$x_j = 1 \quad \text{if} \quad j \in H$$
$$x_j = 0 \quad \text{if} \quad j \notin H$$

and thus $H = \{j : x_j = 1\}$ and $|H| = \sum_{j=1}^{n} x_j$.

We're looking for a solution that <u>minimizes</u> $\sum_{j=1}^{n} x_j$, subject to the following <u>constraints</u>:

4) $x_j \in \{0,1\}$

5) For each *i* we have $j \in S_i$ such that $j \in H$, i.e.: $\sum_{j \in S_i} x_j \geq 1$.

We now <u>relax</u> the problem by changing the first constraint to: $x'_j \in [0,1]$.

Assume <u>there's a polynomial solution</u> for the relaxed problem. We <u>round</u> this solution to get back to the integral 0-1 problem:
$x_j = 1$ if in the solution $x'_j \geq \frac{1}{M}$, $x_j = 0$ otherwise, i.e. $H = \{j : x'_j \geq \frac{1}{M}\}$ where *M* is the size of the maximal set from $S_1, \ldots, S_m$.

Note that now we meet the constraints of the integral 0-1 problem since in any S_i there must be at least one such *j* that $x'_j \geq \frac{1}{M}$ in order to satisfy the second constraint of the relaxed problem.

This algorithm is <u>*M*-approximation</u>:

$$|H| = \sum_{j \in H} x_j \leq_{(1)} \text{¿} \sum_{j \in H} (M \cdot x'_j) = M \sum_{j \in H} x'_j \leq_{(2)} M \sum_{j=1}^{n} x'_j \leq_{(3)} M \sum_{j=1}^{n} x_{j_{Opt}} = M \cdot q(Opt)$$
$$\text{¿}$$

where (1) is because to $x'_j \geq \frac{1}{M}$; (2) is because we sum on all *j* and (3) is due to the way we rounded the solution.

$$\Rightarrow \frac{|H|}{q(Opt)} \leq M$$