Max Wisniewski
Paul Podlech

Dozent : Panos Giannopoulos

Exercise 1 (Sub-Set Ratio Sum Problem)

We didn't find any satisfying solution.

Exercise 2 (MAX-CUT greedy algorithm)

In the tutorial we analyzed a randomized algorithm for MAX-CUT. The result was that $E[Z] \geq \frac{1}{2}OPT$ where $E[Z]$ was the expected weight of the result returned. For every $i \in V$ let $M_i \in \{A, B\}$ be the choice of the subset to put i in. By the law of total expectation we know

$$\begin{array}{rcl} E[Z] & = & E[Z|1 \in A]Pr[M_1 = A] + E[Z|1 \in B]Pr[M_1 = B] \\ & = & \frac{1}{2}(E[Z|1 \in A] + E[Z|1 \in B]) \end{array}$$

and

$$\begin{array}{rcl} E[Z|1 \in M_1, \cdots, i-1 \in M_{i-1}] & = & E[Z|1 \in M_1, \cdots, i \in A]Pr[M_i = A] \\ & + & E[Z|1 \in M_1, \cdots, i \in B]Pr[M_i = B] \\ & = & \frac{1}{2}(E[Z|1 \in M_1, \cdots, i \in A] \\ & + & E[Z|1 \in M_1, \cdots, i \in B]) \end{array}$$

If we choose M_i deterministically as $M_1 := A$ and M_i (recursively) to maximize $E[Z|1 \in M_1, \cdots, i \in M_i]$ we also know $E[Z] = E[Z|1 \in M_1]$ and $E[Z|1 \in M_1, \cdots, i \in M_i] \geq E[Z|1 \in M_1, \cdots, i-1 \in M_{i-1}]$. This implies $E[Z|1 \in M_1, \cdots, n \in M_n] \geq E[Z]$ by induction.
One can now easily observe that the described greedy algorithm chooses the M_i exactly as described. In the k-th iteration the edges already in the cut and not yet in the cut (determined by previous choices), except those between F and the vertex k, add to both considered expectations. Hence the choice is only dependant on the edges between F and k.
This concludes the greedy algorithm to be a $\frac{1}{2}$-approximation for MAX-CUT.

Exercise 3 (Max-Cut Local Search)

Consider the Local Search Algorithm for *Max-Cut* from the lecture and consider the following variant: a vertex changes sides only when the improvement in the total weight of the resulting cut is at least $\frac{2n}{\varepsilon}w(A, B)$, when $w(A, B)$ was the weight of the original cut.

Proof that this is a $(\frac{1}{2} - \varepsilon)$ approximation with $0 < \varepsilon < \frac{1}{2}$. How many iterations does this algorithm make?

Solution:
We didn't erased the part we concluded so far, but we believe there is a mistake in the question.

If we consider the change should be at least $\frac{2n}{\varepsilon}$ like in in the exercise, for $\lim_{\varepsilon\to 0}\frac{2n}{\varepsilon}$ we would literaly allow no changes, so this algorithm would return any feasable solution in the first step.

Solution assumed $\frac{2\varepsilon}{n}$

Lemma 1. *The variant of Local Search Max - Cut is a $\frac{1}{2}-\varepsilon$ approximation.*

Proof lemma 1 :
After the algorithm terminated, we know taht for every vertix holds, if we swap the partition of the vertex, the result will have a new weight, less than $\frac{2\varepsilon}{n}w(A,B)$.

$$\begin{array}{rlcl} (I) & \forall a\in A: & w(A\setminus\{a\},B\cup\{a\}) \leq & \frac{2\varepsilon}{n}w(A,B) \\ (II) & \forall b\in B: & w(A\cup\{b\},B\setminus\{b\}) \leq & \frac{2\varepsilon}{n}w(A,B) \end{array}$$

We will now use the trick from the lecture and sum the equation up, for all $a\in A$ (afterwards in B).

$$\begin{array}{rrcl} & \sum_{a\in A}\sum_{i\in A\setminus\{a\}}\sum_{j\in B\cup\{a\}} w_{ij} & \leq & \sum_{a\in A}\frac{2\varepsilon}{n}w(A,B) \\ \Leftrightarrow & \sum_{a\in A}\left(w(A,B)-\sum_{i\in B}w_{aj}+\sum_{i\in A}w_{ai}\right) & \leq & \sum_{a\in A}\frac{2\varepsilon}{n}w(A,B) \\ \Leftrightarrow & |A|w(A,B)-\sum_{a\in A}\sum_{i\in B}w_{ti}+\sum_{a\in A}\sum_{i\in A}w_{ai} & \leq & |A|\frac{2\varepsilon}{n}w(A,B) \end{array}$$

Now we sum both lines up.

$$\begin{array}{rrcl} \overset{+}{\Rightarrow} & n\cdot w(A,B)-\sum_{a\in A}\sum_{i\in B}w_{ai}-\sum_{b\in B}\sum_{i\in A}w_{bi}+\sum_{a\in A}\sum_{i\in A}w_{ai}+\sum_{b\in B}\sum_{i\in B}w_{bi} & \leq & \frac{2}{\varepsilon}w(A,B) \\ \Leftrightarrow & (n-2)w(A,B)+\sum_{\{i,j\}\in A}w_{ij}+\sum_{\{i,j\}\in B}w_{ij} & \leq & \frac{2}{\varepsilon}w(A,B) \\ \Leftrightarrow & (n-3)w(A,B)+W & \leq & \frac{2}{\varepsilon}w(A,B) \\ \Leftrightarrow & W & \leq & \left(\frac{2}{\varepsilon}-n+3\right)w(A,B) \end{array}$$

For $n>3$ now $OPT\leq W\leq(\frac{2}{\varepsilon}-n+3)w(A,B)\leq(\frac{2}{\varepsilon}+1)w(A,B)$ and we conclude $w(A,B)\geq(\frac{1}{2}-\varepsilon)OPT$

Lemma 2. *The number of iterations for this algorithm is $\frac{n}{\varepsilon}\log W$*

Proof lemma 2 :
For our case we now, that in each step the algorithm will take the change is at least $\frac{2\varepsilon}{n}$. So for each step in the iteration $w(A_k,B_k)\geq\frac{2\varepsilon}{n}w(A_{k-1},B_{k-1})$. We can start with the initial Partition and get the result:

$$w(A_k,B_k)=\left(\frac{2\varepsilon}{n}\right)^k w(A_0,B_0)$$

The algorithm will terminate, if we reach a point, were there is no such step possible:

$$\begin{array}{rrcl} & w(A_{k+1},B_{k+1} & \leq & \frac{2\varepsilon}{n}w(A_k,B_k) \\ \Leftrightarrow & (\frac{2\varepsilon}{n})^{k+1}w(A_0,B_0) & \leq & (\frac{2\varepsilon}{n})^k w(A_0,B_0) \\ \Leftrightarrow & \left(\frac{2\varepsilon}{n}\right)^{k+1} & \leq & \left(\frac{2\varepsilon}{n}\right)^k W \\ \Leftrightarrow & \left(\frac{2\varepsilon}{n}\right)^{k+1} & \leq & W \\ \Leftrightarrow & (k+1) & \geq & \log_{\frac{2\varepsilon}{n}} W \\ \Leftrightarrow & k+1 & \geq & \frac{\log W}{\log\frac{2\varepsilon}{n}} \\ \Leftrightarrow & k & \geq & \frac{n}{\varepsilon}\log W \end{array}$$

The last step for sure isn't right, but we saw the error in the exercise 2h before the tutorium, so there was no real time.
THIS SOLUTION WAS CREATED BEFORE WE SAW THE ERROR
Note that this is not a correct solution. This is the last revision of our work, but it doesn't look right.

Lemma 3. *The variant of Local Search Max - Cut is a $\frac{1}{2} - \varepsilon$ approximation.*

Proof Lemma 3:
After the algorithm terminated, we know taht for every vertix holds, if we swap the partition the vertix is in, the resulting change of the weight of the cut is less than $\frac{2n}{\varepsilon} w(A, B)$.

$$\begin{array}{rlrcl} (I) & \forall a \in A : & w(A \setminus \{a\}, B \cup \{a\}) & \leq & \frac{2n}{\varepsilon} w(A, B) \\ (II) & \forall b \in B : & w(A \cup \{b\}, B \setminus \{b\}) & \leq & \frac{2n}{\varepsilon} w(A, B) \end{array}$$

Next we will sum up this equotation for all $a \in A$ or $b \in B$.

$$\begin{array}{rrcl} & \sum\limits_{a \in A} \sum\limits_{i \in A \setminus \{a\}} \sum\limits_{j \in B \cup \{a\}} w_{ij} & \leq & \sum\limits_{a \in A} \frac{2n}{\varepsilon} w(A, B) \\ \Leftrightarrow & \sum\limits_{a \in A} \left(w(A, B) - \sum\limits_{i \in B} w_{aj} + \sum\limits_{i \in A} w_{ai} \right) & \leq & \sum\limits_{a \in A} \frac{2n}{\varepsilon} w(A, B) \\ \Leftrightarrow & |A| w(A, B) - \sum\limits_{a \in A} \sum\limits_{i \in B} w_{ti} + \sum\limits_{a \in A} \sum\limits_{i \in A} w_{ai} & \leq & |A| \frac{2n}{\varepsilon} w(A, B) \end{array}$$

We will now apply the same transformation to (II) and add the equations up.

$$\begin{array}{lrcl} \overset{+}{\Rightarrow} & n \cdot w(A, B) - \sum\limits_{a \in A} \sum\limits_{i \in B} w_{ai} - \sum\limits_{b \in B} \sum\limits_{i \in A} w_{bi} + \sum\limits_{a \in A} \sum\limits_{i \in A} w_{ai} + \sum\limits_{b \in B} \sum\limits_{i \in B} w_{bi} & \leq & \frac{2n^2}{\varepsilon} w(A, B) \\ \Leftrightarrow & (n-2) w(A, B) + \sum\limits_{\{i,j\} \in A} w_{ij} + \sum\limits_{\{i,j\} \in B} w_{ij} & \leq & \frac{2n^2}{\varepsilon} w(A, B) \\ \Leftrightarrow & (n-3) w(A, B) + W & \leq & \frac{2n^2}{\varepsilon} w(A, B) \\ \Leftrightarrow & W & \leq & \left(\frac{2n^2}{\varepsilon} - n + 3 \right) w(A, B) \end{array}$$

Further we know from the lecture, that $OPT \leq W$ holds.

$$\begin{array}{lrcl} & OPT \leq W & \leq & \left(\frac{2n^2}{\varepsilon} - n + 3 \right) w(A, B) \\ \Leftrightarrow & W & \geq & \frac{1}{\frac{2n^2}{\varepsilon} - n + 3} OPT \\ & & = & \frac{varepsilon}{2n^2 \varepsilon n + 3\varepsilon} OPT \\ & & = & \left(\frac{1}{2n^2 - \varepsilon n + 3\varepsilon} - \frac{1 - \varepsilon}{2n^2 - \varepsilon n + 3\varepsilon} \right) OPT \\ & & \leq & \left(\frac{1}{2} - \frac{1 - \varepsilon}{2 + 3\varepsilon} \right) OPT \\ & & \leq & \left(\frac{1}{2} - \varepsilon \right) OPT \end{array}$$

The step to the $\frac{1}{2}$ holds, because if we choose n and ε to be small, the result will be only greater than the other values. Hence we can drop them and the result will be only bigger. The last step will only hold for $\varepsilon < \frac{1}{4}$ approximatly. Therefore we are shure there is a mistake in our solution.

Lemma 4. *The amount of iterations this algorithm will make is at most $\frac{n}{\varepsilon} \log W$.*

Proof lemma 4: