Approximation Algorithms
Exercise sheet 3
Due: 24/5

Dozent: Panos Giannopoulos

May 11, 2012

Note: Try to solve at least one exercise, e.g., the one you like most! For any questions you can email me, or ask Herr Mulzer in the next lecture.

Ex. 1 — (Vazirani 8.3) Consider the Subset-Sum ratio problem: Given n positive integers $a_1 < \ldots < a_n$, find two disjoint nonempty subsets $S_1, S_2 \subseteq \{1, \ldots, n\}$ with $\sum_{i \in S_1} a_i \geq \sum_{i \in S_2} a_i$, such that the ratio

$$\frac{\sum_{i \in S_1} a_i}{\sum_{i \in S_2} a_i}$$

is minimized. Give an FPTAS for the problem. (Hint: First obtain a pseudopolynomial time algorithm and then scale and round.)

Ex. 2 — (Williamson & Smoys 5.2)
Consider the following greedy algorithm for the cardinality MAX-CUT: Let $G(V, E)$ be the input graph with $V = \{1, \ldots, n\}$, and weight $w_{ij} = 1$ for each edge $(i, j) \in E$. In the first iteration, we place vertex 1 in A. In the kth iteration we place vertex k in either A or B. (Recall that (A, B) is a partition of V.) The choice on where to place k will depend on all the edges F with k as one endpoint and some vertex in $\{1, \ldots, k-1\}$ as the other endpoint, i.e., $F = \{(j, k) \in E : 1 \leq j \leq k-1\}$. Vertex k is put in A or B depending on which of these two choices maximizes the number of edges in the cut. Prove that this is a 1/2-approximation algorithm.

Ex. 3 — Consider the local search algorithm for general (weighted) MAX-CUT that we did in the tutorial: The algorithm starts with any partition (A, B) and in each step it moves a vertex v from its current side of the partition (A or B) to the other side (B or A) when the total weight of the edges from v to vertices on the other side is smaller than the total weight of edges from v to vertices on its side. When no such vertex exists, the algorithms returns the current (last) partition. We have shown that this partition gives a cut whose weight is at least 1/2 times the weight of an optimal cut. However, the algorithm is pseudopolynomial since the number of iterations can be $\Theta(W)$, where $W = \sum_{(i,j)\in E} w_{ij}$.

Consider now the following variant of this algorithm: a vertex changes sides (resulting to a new partition) only when the improvement in the total weight of the cut is at least $(2n/\epsilon)w(A, B)$, where $w(A, B)$ is the weight of the cut resulting from the current partition (A, B) and ϵ is given with $0 < \epsilon < 1/2$. (That is, we don't care about 'small' changes.) The algorithm terminates when no such 'big' change is possible. Show that this is a $(1/2 - \epsilon)$-approximation algorithm. How many iterations does the algorithm make?