Advanced Algorithms – Exercise 1

Deadline: 28/11/2005

1. Recall the LP for SET-COVER we saw in class:

   Minimize: $\Sigma_{s \in S} c(s) x_s$ *Subject to:*

   - For each element $e \in U$: $\Sigma_{s|e \in s} x_s \geq 1$
   - for each $s \in S$: $x_s \geq 0$

   Define the *frequency* of an element e, denoted by f_e, to be the number of sets the element e is in. Let $f = \max_{e \in U} f_e$. In this question we will devise an f-approximation algorithm for SET-COVER.

   (a) Show that in any solution of the LP, for each element e there is at least one set s that covers it, such that $x_s \geq \frac{1}{f_e}$.

   (b) Argue that the following set, $COVER = \{s | s \in S, x_s \geq \frac{1}{f}\}$ is a valid set cover.

   (c) Prove that the cost of the sets in $COVER$ is at most f times the size of the optimal cover. Conclude that there is an f-approximation algorithm for SET-COVER.

   (d) Show a 2-approximation algorithm for the *weighted* vertex cover problem.

2. In the previous question we used the optimal fractional solution as a lower bound to the value of the optimal solution. We proved that for every fractional solution of cost c, there is an integral solution of cost no more than $c \cdot f$. We will now show that there are SET-COVER instances of which the cost of the optimal solution is almost f times the cost of the optimal fractional solution. That is, the *integrality gap* for this relaxation of SET-COVER is exactly f. Consequently, no algorithm that uses the optimal fractional solution as a lower bound can guarantee more than f approximation.

   (a) We first start with proving an integrality gap for the cardinality version of VERTEX-COVER. Consider the Complete graph with n vertices K_n. What is the optimal integral vertex cover of K_n? What is the optimal fractional vertex cover of K_n? Show that the integrality gap of VERTEX-COVER is at least $2 - \frac{1}{n}$.

   (b) An *hypergraph* is a pair $G = (V, E)$ where V is a set of vertices and E, and E is a set of *hyperedges*. Each hyperedge is a set of vertices[1]. A set S of vertices is said to be a vertex cover of a hypergraph, if for each hypergraph $e \in E$, there exists a vertex $v \in V$ such that $v \in e$.

      i. Show that the problem of finding a minimal vertex cover in a hypergraph is equivalent to SET-COVER.

      ii. Show that the integrality gap of SET-COVER relaxation is at least $f - o(1)$.

[1] In graphs, each such set has size of exactly 2

3. Consider the following two problems:

   PARTITION: we are given n positive numbers $a_1, ..., a_n$. Our goal is to decide if there is a partition of the numbers into two sets S_1 and S_2 such that $\Sigma_{a_i \in S_1} a_i = \Sigma_{a_i \in S_2} a_i = \Sigma_{i=1}^{n} \frac{a_i}{2}$.

   BIN-PACKING: we are given n positive numbers $a_1, ..., a_n$, where for each i, $0 < a_i \leq 1$. Our goal is to find a packing of the items in unit-sized bins that minimizes the number of bins used[2].

   It is known that PARTITION is NP-hard. We will now reduce PARTITION to BIN-PACKING and show that for any $\epsilon > 0$, approximating BIN-PACKING to within a factor of $\frac{3}{2} - \epsilon$ is NP-hard too.

   (a) Reduce PARTITION to BIN-PACKING.

   Remark: This shows that BIN-PACKING is NP-hard, but we are interested in a stronger result.

   (b) If there was a partition of the items into two sets as required, what is the size of the optimal bin-packing?

   (c) If there was no partition of the items into two sets as required, what is the size of the optimal bin-packing?

   (d) Conclude that for any $\epsilon > 0$, approximating BIN-PACKING to within a factor of $\frac{3}{2} - \epsilon$ is NP-hard.

4. In class we saw an FPTAS for knapsack. However, its running time was prohibitive. The goal of this question is to develop a faster algorithm, in the cost of achieving only 2-approximation for the optimal solution.

   (a) Order the items in a decreasing profit to size ratio. Greedily pick objects in that order. Show that the performance of this heuristic is unbounded.

   (b) We will show that the following modification of the greedy algorithm is a 2-approximation: Greedily choose (in decreasing profit to size ratio) the first k items that their size exceeds the knapsack size B. Pick the most profitable set: $\{a_1, ..., a_{k-1}\}$, or $\{a_k\}$.

      i. Prove the following lemma:

      Lemma: $\Sigma_{i=1}^{k} \text{profit}(a_i) \geq OPT$.

      Hint: Compare the average value per item of items in $\{a_1, ..., a_k\}$ that are not in the optimal solution, to the average value per item of items in the optimal solution that are not in $\{a_1, ..., a_k\}$. You will also need to compare the sizes of these two sets.

      ii. Devise a 2-approximation algorithm from the lemma.

   (c) What is the running time of the 2-approximation algorithm?

[2] We say that a packing is legal if the sum of items packed in each bin is less than 1.