6.854J / 18.415J Advanced Algorithms
Fall 2008

18.415/6.854 Advanced Algorithms

Problem Set 5

1. In the bin packing problem, we are given n items, item i being of length a_i ($0 < a_i \leq 1$), and we need to find the minimum number of bins of length 1 so that no bin contains items whose total length exceeds 1. This problem is NP-hard. Consider the following heuristic, called "First Fit" (FF): Consider the items in any order and place each item into the first bin that still has room for it. Let L^* denote the minimum number of bins needed and let L_{FF} be the number of bins obtained by using First Fit.

   (a) Show that $L_{FF} \leq 2L^* - 1$ for any instance.

   (b) Show that $L_{FF} \leq \alpha L^* + \beta$ for some $\alpha < 2$. The best possible answer is $\alpha = 1.7$ and $\beta = 2$, but this is somewhat tricky to show (or *supposedly* tricky: you might have an easy argument).

   Hint to get $\alpha = 1.75$ in case you don't have any other idea. Consider three types of bins in the packing obtained by FF. B_1 consists of the bins containing items of total length greater than $2/3$, B_2 consists of the bins not in B_1 containing one item of length greater than 0.5 (and possibly other items) and B_3 consists of the remaining bins. Show first that $|B_3| \leq 2$.

2. Consider the following problem. Given a collection $\mathcal{F}$ of subsets of $\{1, \ldots, n\}$ and an integer k, find k sets $S_1, \ldots, S_k$ in $\mathcal{F}$ such that $|S_1 \cup S_2 \cup \ldots \cup S_k|$ is maximum. This problem is NP-hard. The greedy algorithm first chooses S_1 to be the largest set, and then having constructed $S_1, \ldots, S_{i-1}$ chooses S_i to be the set that maximizes

$$\left|S_i \setminus \cup_{j=1}^{i-1} S_j\right|.$$

   Show that the greedy algorithm is a $1 - \left(1 - \frac{1}{k}\right)^k$-approximation algorithm.

   (Hint: You may want to show that, for any j, the union of the first j sets given by the greedy algorithm have a cardinality at least

$$1 - \left(1 - \frac{1}{k}\right)^j OPT,$$

   where OPT denotes the maximum cardinality of the union of k sets.)

3. In MAX 2SAT, we are given a collection $C_1, \ldots, C_k$ of boolean clauses with at most two literals per clause. Each clause is thus either a literal or the disjunction of two literals drawn from a set of variables $\{x_1, x_2, \ldots, x_n\}$. A *literal* is either

a variable x or its negation $\bar{x}$. The goal is to find an assignment of truth values to the variables $x_1, \ldots, x_n$ that maximizes the number of satisfied clauses.

(a) Show that the algorithm which independently sets every x_i to true with probability 0.5 is a randomized 0.5-approximation algorithm. (As usual, compute the expected number of satisfied clauses.)

(b) Consider the following linear program:

$$
(LP) \quad
\begin{array}{lll}
\text{Max} & \displaystyle\sum_{j=1}^{k} z_j & \\
\text{subject to:} & & \\
& \displaystyle\sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i) \geq z_j & j = 1, \ldots, k \\
& 0 \leq y_i \leq 1 & 1 \leq i \leq n \\
& 0 \leq z_j \leq 1 & j = 1, \ldots, k,
\end{array}
$$

where I_j^+ (resp. I_j^-) denotes the set of variables appearing unnegated (resp. negated) in C_j. For example, the clause $x_3 \vee \bar{x}_5$ would give rise to the constraint $y_3 + 1 - y_5 \geq z_j$.

i. Show that the optimum value of this linear program is an upper bound on the optimum value of MAX 2SAT.

ii. Let y^*, z^* denote the optimum solution of this linear program. Show that the algorithm which independently sets every x_i to true with probability y_i^* is a randomized 0.75-approximation algorithm.

(c) Consider now an approach similar to the one described in class for MAX CUT. Define a unit vector v_0 corresponding to "true" and also a unit vector v_i for each variable x_i. Define the "value" of the clause or literal x_i as $v(x_i) = \frac{1+v_0 \cdot v_i}{2}$ and the value of $\bar{x}_i$ as $v(\bar{x}_i) = \frac{1-v_0 \cdot v_i}{2}$. Observe that $v(x_i)$ is 1 if $v_0 = v_i$, 0 if $v_0 = -v_i$, and between 0 and 1 otherwise. For a clause with two literals, say $C = x_1 \vee x_2$, define $v(C)$ as $(3+v_0 \cdot v_1 + v_0 \cdot v_2 - v_1 \cdot v_2)/4$. The value of other clauses with two literals are similarly defined. Consider now the nonlinear program:

$$
(NLP) \quad
\begin{array}{lll}
\text{Maximize} & \displaystyle\sum_{j=1}^{k} v(C_j) & \\
\text{subject to:} & & \\
& v_i \in S_n & i = 0, 1 \ldots, n.
\end{array}
$$

i. Show that the optimum value of this nonlinear program is an upper bound on the optimum value of MAX 2SAT.

ii. Consider the algorithm which first solves this nonlinear program optimally, then generates a uniformly selected point r on the unit sphere S_n, and sets x_i to be true if $(v_0 \cdot r)(v_i \cdot r) \geq 0$. Using the analysis of the MAX CUT algorithm seen in class, show that this algorithm is a randomized 0.878-approximation algorithm for MAX 2SAT.

(d) Can you do better than 0.878?