Master on Foundations of Computer Science (1st year)
Approximation Algorithms

Mid-Term Exam

November 23rd, 2004

Exercise 1: Traveler Salesman Problem with edge weights 1 or 2

Let $\text{TSP}_{\{1,2\}}$ denote the restriction of TSP to instances where the edge weight function takes its values in $\{1, 2\}$ only.

Question 1. *Under the assumption $P \neq NP$, show that, for all $\epsilon > 0$, there does not exist any $(1 + \frac{1}{n} - \epsilon)$-approximation for $TSP_{\{1,2\}}$.*

Exercise 2: Easy Steiner

Question 2. *Show that if the vertices used in an optimal Steiner tree are given with the instance, then the Steiner Tree problem can be solved in polynomial time.*

Exercise 3: MAX-DIRECTED-CUT

Given a *directed* graph $G = (V, E)$ and a non-negative weight function on the arcs, $w : E \to \mathbb{R}^+$, find a maximum weight directed cut, i.e., a subset of vertices, $S \subseteq V$, that maximizes the total weight of its *outgoing* arcs (*i.e.*, the total weight of the arcs $(u \to v)$ such that $u \in S$ and $v \notin S$).

Question 3. *Give (and analyze) a randomized $\frac{1}{4}$-approximation for this problem. Give a family of tight instances.*

Question 4. *Using the method seen during the lectures, derandomize your algorithm to obtain a deterministic $\frac{1}{4}$-approximation. What is its time complexity? Give a family of tight instances.*

Exercise 4: Cycles and tournois

A *tournois* is a complete graph $G = (V, E)$ whose edge have been directed (i.e., such that, for all pair of vertices $u, v \in V$, either $(u \to v) \in E$ or $(v \to u) \in E$). Given a tournois G, the *acyclic vertex set problem* consists in finding a maximum size subset of vertices, $S \subseteq V$, such that the subgraph induced by S in G is acyclic.

Question 5. *Show that a tournois without cycle of length 3 is acyclic.*

We admit that there exist a f-approximation for the restricton of SET-COVER to the instances where each element of the universe belongs to at most f sets.

Question 6. *Give (and analyze) a 3-approximation for the acyclic vertex set problem.*

Exercise 5: Traveler Salesman Path Problem

Given a undirected complete graph with a metric edge weight function, find a minimum length *simple path*, *i.e.*, a path that contains each vertex exactly once.

There exists three variants of the problem, depending on whether 0, 1 or 2 endpoints of the path are fixed by the input.

Question 7. *Give (and analyze) a 2-approximation for the variants where 0 or 1 endpoint of the path are given in the input (i.e., where (a) the traveler can go from and arrive to any vertex or (b) where the traveler has to start from a given vertex but can arrive to any other vertex).*

Give a family of tight instances.

Question 8. *Give (and analyze) a 3/2-approximation for the variant where none of the endpoints are given in the input.*

Give a family of tight instances.

Exercise 6: Derandomization by the Method of Conditional Expectation

Question 9. *Let X and B be two random variables, such that B takes its values in a finite set $\mathcal{B}$. Show that:*

$$\max_{b\in\mathcal{B}} \mathbb{E}(X \mid B = b) \geq \mathbb{E}(X).$$

We will now use this fact to derandomize a Monte-Carlo algorithm by replacing every random bit draw by the "best random choice" for this bit (*i.e.*, the b that maximizes $\mathbb{E}(X \mid B = b)$ in the question above).

Let $\mathcal{A}(I, \omega)$ be a randomized α-approximation for a given maximization problem Π ($\alpha < 1$), where I denotes the problem instance and $\omega = \omega_1\omega_2\cdots$ denotes the chain of random bits. Assume that algorithm $\mathcal{A}(I, \omega)$ is Monte-Carlo and that its computation time on every instance I is (uniformly) bounded by a polynomial $p(|I|)$ (independently of the chain of random bits).

Let I be a fixed instance. Let X be the random variable for the value of the (random) solution computed by algorithm $\mathcal{A}$ on instance I.

Assume that, for all q and all binary sequences $b_1 \dots b_q$, one can compute *exactly in polynomial time* the conditional expectation:

$$\mathbb{E}(X \mid \omega_q = b_q, \dots, \omega_1 = b_1)$$

i.e. the conditional expectation of the value of the solution returned by algorithm $\mathcal{A}$, given that the values of the first q random bits are $b_1, \dots, b_q$.

Given $b_1, \dots, b_q \in \{0, 1\}$ constant, let $b_{q+1} \in \{0, 1\}$ be the bit that maximizes:

$$\mathbb{E}(X \mid \omega_{q+1} = b_{q+1}, \omega_q = b_q, \dots, \omega_1 = b_1).$$

Note that, given $b_1, \dots, b_q$ fixed, one can compute b_{q+1} in polynomial time.

Question 10. *Show that:*

$$\mathbb{E}(X|\omega_{q+1} = b_{q+1}, \omega_q = b_q, \dots, \omega_1 = b_1) \geq \mathbb{E}(X).$$

Hint: proceed by induction.

The main idea of the method of conditional expectation is to execute the algorithm and build greedily the random bits one after the other when the algorithm requests them. When the algorithm requests the qth bit, we make greedily the "best random possible choice" for b_q, given the choices already made for $b_1, \dots, b_{q-1}$.

Question 11. *Describe precisely the* determistic *greedy algorithm obtained by this method. Show that the obtained greedy algorithm is a (determistic) α-approximation if $\mathcal{A}$ is a (randomized) α-approximation for Π.*

We now apply the method of conditional expectation to MAX-CUT.

Given a graph G with a non-negative weight function of edges, we consider the following randomized approximation for MAX-CUT (seen in the lectures):

```
S_0 ← ∅
Let V = {v_1, ..., v_n}
For i = 1 to n do
    If ω_i = 1 then S_i ← S_{i-1} ∪ {v_i} else S_i ← S_{i-1}
EndFor
Return the cut (S_n, \overline{S_n})
```

We use the same notation as above, applied to this particular algorithm.

Question 12. *Given $b_1, \ldots, b_q \in \{0, 1\}$ constant, show that one can compute in polynomial time:*

$$\mathbb{E}(X \mid \omega_q = b_q, \ldots, \omega_1 = b_1).$$

Hint: Rewrite $\mathbb{E}(X \mid \omega_q = b_q, \ldots, \omega_1 = b_1)$ as a function of the relationships of the edges with the two sides of the cut already built.

Question 13. *Deduce (and analyze) a greedy deterministic $1/2$-approximation for MAX-CUT. Rephrase this algorithm in a natural way.*

♣