CMPUT 675 - Assignment #2

Fall 2014, University of Alberta
Due October 8 in class.

Pages: 3

This assignment is to be completed individually. I understand that you may want to discuss the assignment with other students, a good guide for understanding my expectations is that you should not take notes or work out precise details in your discussions (keep it high-level). Mention any discussions and cite any resources you used on the writeup you hand in.

To be clear, whenever you are asked to give an approximation algorithm for a problem, it is expected that you will both describe the algorithm and prove the claimed approximation guarantee. If you can only think of an algorithm with a worse approximation guarantee than I am asking for, then describe it anyway. You may get partial marks (though, it cannot be entirely trivial).

Problem 1)

Marks: 3

Give an FPTAS for MINIMIZING MAKESPAN ON IDENTICAL MACHINES when there are only 2 machines (this version is still NP-hard).

Problem 2)

Marks: 2

A problem is said to be *strongly NP-hard* if it remains NP-hard even when all of its numerical parameters are integers bounded by a polynomial in the input size.

For example, MINIMIZING MAKESPAN ON IDENTICAL PARALLEL MACHINES is known to be strongly NP-hard. In particular, the problem remains NP-hard even when all processing times p_j are integers in the range $[0, 2^{16} \cdot n^4]$ where n is the number of jobs[1].

Using this fact, show that if there is an FPTAS for MINIMIZING MAKESPAN ON IDENTICAL PARALLEL MACHINES then P = NP.

Note: One can prove more generally that if a problem is strongly NP-hard then there is no FPTAS for that problem unless P = NP. Solving this exercise should be enough to convince you that this general statement is true.

[1] A reference to this fact will be provided in the solutions.

Problem 3)

Marks: 4

In the MINIMUM COST BOUNDED LENGTH PATH problem, you are given a **directed acyclic** graph $G = (V, E)$ where each edge $e \in E$ has a *length* $\ell(e) \geq 0$ and a *cost* $c(e) \geq 0$. Additionally, you are given two different vertices $s, t \in V$ and a *length bound* $L \geq 0$. The goal is to find the cheapest $s - t$ path that has length $\leq L$.

Give an FPTAS for this problem. The length (w.r.t. ℓ) of the path your algorithm finds must be at most L and the cost (w.r.t. c) should be at most $(1 + \epsilon) \cdot OPT$ where OPT is the minimum cost path among all paths with length at most L.

Bonus: Give an FPTAS for general directed graphs (i.e. without the acyclic assumption) for a small +0.5 bonus.

Problem 4)

Marks: 5

In the MAXIMUM COVERAGE problem, we are given a ground set of items X and a collection of subsets $\mathcal{S}$ of X. Additionally, we are given an integer k. Note that this is almost the same input as in the SET COVER problem, except there are no set costs.

The goal is to select k subsets $S_1, \ldots, S_k \in \mathcal{S}$ to maximize the number of items $i \in S$ that lie in some chosen S_j. That is, these subsets should maximize the size of $\bigcup_{j=1}^{k} S_j$.

- Give a greedy $\left(1 - \frac{1}{e}\right)$-approximation for this problem where $e = 2.718\ldots$ is the base of the natural logarithm. [3 marks]

  Hint: $\left(1 - \frac{1}{k}\right)^k \leq e^{-1}$ for every $k \geq 1$.

- Recall that for any constant $c < 1$ that there is no $c \cdot \ln(n)$-approximation for SET COVER unless P = NP. This holds even when $c(S) = 1$ for every set $S \in \mathcal{S}$. Use this to prove that there is no c'-approximation for MAXIMUM COVERAGE for any constant $c' > 1 - \frac{1}{e}$ unless P = NP. [2 marks]

Problem 5)

Marks: 4

Consider an instance $(X, \mathcal{S})$ of SET COVER with set costs $c(S) \geq 0, S \in \mathcal{S}$. For each $i \in X$, let $\mathcal{S}_i = \{S \in \mathcal{S} : i \in S\}$ be the sets that cover i. Suppose that b is a bound such that $|\mathcal{S}_i| \leq b$ for each $i \in I$.

For example, the MINIMUM WEIGHT VERTEX COVER problem is really just an instance of the SET COVER problem with $b = 2$.

Algorithm 1 SET COVER approximation

> $\mathcal{C} \leftarrow \emptyset$
> $z_i \leftarrow 0$ for each $i \in I$
> **while** $\mathcal{C}$ does not cover X **do**
>   Pick some item $i \in X$ not covered by $\mathcal{C}$
> $\alpha \leftarrow \min\{c(S) - \sum_{j \in S} z_j : S \in \mathcal{S}_i\}$
> $z_i \leftarrow \alpha$
> $\mathcal{C} \leftarrow \mathcal{C} \cup \{S \in \mathcal{S}_i : c(S) = \sum_{j \in S} z_j\}$
> **end while**
> **return** $\mathcal{C}$

Answer the following questions about Algorithm 1.

- Show that when the algorithm terminates, $\sum_{i \in X} z_i \leq OPT$. [2 marks]
- Show that the cost of $\mathcal{C}$ is at most $b \cdot \sum_{i \in X} z_i$. Conclude that this is a b-approximation. [2 marks]

Problem 6)

Marks: 3

Return to the KNAPSACK problem, in which we have items I, weights $w_i \geq 0$ and values $v_i \geq 0$ for each item $i \in I$, and a knapsack capacity of C. We saw a $(1 - \epsilon)$-approximation in class with running time $O(n^3/\epsilon)$. Here, you will reduce the running time to $O(n^2/\epsilon)$.

- Show that if we already know of values α and β such that $OPT/\beta \leq \alpha \leq OPT$ then the FPTAS from the lecture can be modified to run in time $O(\beta \cdot n^2/\epsilon)$. [1 mark]
- Assume $w_i \leq C$ for each $i \in I$. Sort the items $i \in I$ so that $v_1/w_1 \geq v_1/w_2 \geq \ldots \geq v_n/w_n$. Let k^* be the largest integer such that $\sum_{i=1}^{k^*} w_i \leq C$. Finally, let $i' \in I$ be an item with maximum value. Show that one of $\{i'\}$ and $\{1, 2, \ldots, k^*\}$ has value at least $OPT/2$. Conclude that there is an FPTAS for KNAPSACK with running time $O(n^2/\epsilon)$. [2 marks]