# Advanced Algorithms – Exam

Deadline: 26/2/2006, Ross closing time.

**General Guidelines:**

1. Solve the questions by yourself. In particular, you may not discuss the questions with your classmates.
2. You may consult written material such as books, papers, etc. In this case, clearly cite the sources you have used.
3. The exam consists of 5 questions, each one worth 25 points. You may solve all of them. There is also one bonus question which worth 10 points. Each question will be graded as a whole, and not each item separately.
4. The grade can not exceed 100.

**Questions:**

1. The MAX-k-COVERAGE problem is defined as follows:

   **Input:** A universe $U$, a collection of subsets of $U$, $S = \{S_1, ..., S_n\}$, an integer $k$.

   **Output:** A subcollection $T$ of $S$ of size $k$ such that the number of elements (in $U$) covered by $T$ is maximized. That is, we are interested in maximizing $|\cup_{J \in T} J|$.

   (a) Describe the ILP of this problem. Obtain an LP for this problem.

   (b) Show a randomized-rounding based $(1-(1-\frac{1}{k})^k) \approx (1-\frac{1}{e})$-approximation algorithm for MAX-k-COVERAGE.

2. The GENERALIZED-k-COVERAGE problem is defined as follows:

   **Input:** A universe $U$, $r$ collections of subsets of $U$, $S_1 = \{S^1_1, ..., S^1_{n_1}\}, ..., S_r = \{S^r_1, ..., S^r_{n_r}\}$, an integer $k$.

   **Output:** A collection $T$ of size $k$ that can be constructed by choosing a single subset from each $S_i$. We are interested in maximizing the total number of elements covered by $T$.

   Observe that the MAX-k-COVERAGE is a special case of GENERALIZED-k-COVERAGE (where all $S_i$'s are identical.)

   Show that the following simple greedy algorithm is a 2-approximation for the GENERALIZED-k-COVERAGE problem.

   **The Greedy Algorithm:**

(a) For $i = 1...r$:

  i. Let $J \in S_j$ be the set that maximizes $(U \cap J)$.

  ii. Add $J$ to $T$.

  iii. Let $U = U \backslash J$.

(b) Output $T$.

**Hint:** Suppose that $S_1^1$ is the subset selected from $S_1$ in the optimal solution. What is the maximal loss (comparing to the optimal solution) if the algorithm chooses some other subset $S_k^1$ instead? Use induction to prove the approximation ratio.

3. In class we saw a deterministic $(2 - \frac{1}{B})$-competitive algorithm for the online ski problem. In this question we will use randomness to get a better guarantee. However, our algorithm will be *barely random*. I.e., the algorithm may flip only *one* unbiased random coin.

   Recall that a randomized algorithm can be seen as a probability distribution over deterministic algorithms. We know that in the ski problem there is a one-to-one correspondence between deterministic algorithms and the positive integers: each positive integer $d$ defines a deterministic algorithm – buy on day $d$. Therefore, a barely-random algorithm for the ski problem will buy on day $a$ with probability $\frac{1}{2}$, and on day $b$ with probability $\frac{1}{2}$. We assume that $b \geq a$.

   (a) Prove that in order to minimize the competitive ratio, $a \leq B$.

   (b) Fix some $a \leq B$. Show that in order to minimize the competitive ratio $b = B$.

   (c) Find the optimal value of $a$. Find the competitive ratio of the optimal barely-random algorithm.

   **Remark:** We are only interested in this question in the asymptotic competitive ratio. E.g, the asymptotic competitive ratio of the deterministic algorithm is 2.

   **Bonus (10 points):** Suppose we allow the algorithm to buy on day $a$ with some probability $p$, and on day $b$ with probability $1 - p$ (notice that $p$ might be different than $\frac{1}{2}$.) What is the best competitive ratio you can get?

4. Let $G = (V, E)$ be an unweighted graph. In this question we have $n$ players, where each player $i$ is interested in getting a path that will connect his known pair $(s_i, t_i)$. Each player has a *private* utility $v_i$ for getting a path from $s_i$ to $t_i$. Any edge $e \in E$ can serve at most $k = \Theta(\log |E|)$ players. Our goal is to maximize the sum of the utilities of the players that were able to connect their pair. Observe that the setting is a generalization of the multi-commodity flow problem we saw in class, but here we have different values for transferring each "commodity" from its source to its sink. In the multi-commodity flow problem we obtained a $(1 + \epsilon)$-approximation to the optimal solution. We will now see an $O(k)$-truthful approximation mechanism for our generalized problem. For simplicity we assume that the minimal non-zero valuation is 1, and that $v_{max} = \max_i v_i$ is known. Furthermore, let us assume that $v_{max} = poly(|E|)$.

   **The Mechanism:**

   (a) Let $\rho = (4k|E|v_{max})^{\frac{1}{k}}$

   (b) For each $e \in E$ let $p_e = \frac{1}{2k|E|}$.

   (c) For $i = 1...n$:

      i. Let $P_i$ be the most profitable path for bidder $i$. I.e., $P_i$ is the path that maximizes $v_i - \Sigma_{e \in P_i} p_e$.

ii. Assign the path $P_i$ to player $i$. Let player $i$ pay $\Sigma_{e \in P_i} p_e$.

iii. For each $e \in P_i$: let $p_e = p_e \cdot \rho$.

(a) Prove that the algorithm can be implemented in polynomial time, and that the solution obtained is valid.

(b) Prove that the mechanism is incentive compatible.

(c) Prove that the solution provided by the mechanism is an $O(k)$ approximation to the optimal solution.

**Hint:** Let $p_j^0$ denote the value $p_j$ was initiated to. Let $p_j^*$ denote the value of $p_j$ at the end of the algorithm. Let ALG denote the value of the solution produced by the algorithm, and let OPT denote the value of the optimal solution. Let REV denote the total amount paid by the players to the mechanism. Prove the following bounds on ALG:

- $ALG \geq OPT - k\Sigma_{e \in E}\ p_e^*$ (hint to the hint: start with considering each player $i$ alone.)
- $ALG \geq REV \geq (\Sigma_{e \in E}\ p_e^* - \Sigma_{e \in E}\ p_e^0)/\rho$

The approximation ratio follows by taking into account that $k\Sigma_{e \in E}\ p_e^0 \leq \frac{OPT}{2}$ (prove it!)

5. Consider the sparsest cut problem we saw in class. Let $f$ denote the max flow in a graph, and let $SC$ denote the sparsity of the sparsest cut $S$ (I.e., $SC = \min_{S \subseteq V} \frac{C(S)}{\Sigma_{i|(s_i,t_i) \cap S = 1} dem(i)}$). We have proved that $\frac{SC}{O(\log n)} \leq f \leq SC$. We will now see an example which will prove that our analysis was tight.

A graph $G = (V, E)$ will be called an *expander* if for each set of vertices $S \subseteq V$, $C(S) \geq c \cdot \min(|S|, |\overline{S}|)$, for some constant $c$. We will also require that our graph will be 3-regular. That is, each vertex has exactly 3 neighbors. Such graphs do exist – even a random construction will produce such graphs with high probability.

We build our instance as follows: given an expander with $n = |V|$ nodes, define the set of commodities to be all pairs of different vertices. We let all capacities be 1, and also let $dem(i) = 1$, for each commodity $i$.

(a) Show that the minimal cut in the expander is of size $\Theta(\frac{1}{n})$.

(b) Show that in order to sustain a flow $f$ the network capacity should be $\Theta(f \cdot n^2 \log n)$.

**Hint:** Fix a vertex $v \in V$. Call a vertex $v' \neq v$ *far* from $v$ if the shortest path between $v$ and $v'$ is of length $\Theta(\log n)$. Show that for some choice of the constants inside the $\Theta$ at least half of the vertices in $V$ are far from $v$.

(c) Prove that in our instance the gap between the max flow and the sparsest cut is $\Theta(\log n)$.

(d) Claim that there are metrics such that any embedding into $l_1$ has a distortion of $\Omega(\log n)$. I.e., Bourgain theorem is tight. What can you say about embedding into $l_2$?