HW3 Can't get enough Algorithms

MPRI 2.11.1

16.10.2014 - **Due on Thursday 23.10 before 15:00**

You are asked to complete the exercise marked with a [★] and to send me your solutions at:
nicolas.schabanel@liafa.fr
(or drop it in my mail box on the 4th floor) on **Thursday 23.10 before 15:00**.

■ **Exercise 1 (A PTAS for the Knapsack problem).** Given a knapsack of capacity C and n objects of sizes $0 \leqslant s_1, \dots, s_n \leqslant C$ and values $v_1, \dots, v_n > 0$, the knapsack problem consists in finding a subset of objects fitting in the knapsack and of maximum value, i.e. a set $S \subseteq \{1, \dots, n\}$ such that $\mathrm{size}(S) = \sum_{i \in S} s_i \leqslant C$ and that $\mathrm{value}(S) = \sum_{i \in S} v_i$ is maximum. This problem is NP-complete and we want to design a PTAS for it, i.e. a $(1 - \epsilon)$-approximation for all $\epsilon > 0$.

In order to do so, we first try to solve efficiently a simpler problem:

► ***Question* 1.1*)*** *Assume that the values of the objects are positive integers in $\{1, \dots, V\}$ for some integer V. Propose an algorithm that solves exactly the knapsack problem in time $O(n^2 V)$ using $O(nV)$ memory.*

We proceed as follows to reduce the $(1 - \epsilon)$-approximation of the general knapsack problem to this simpler setting. Let $V = \max_i v_i$ and $K = \epsilon V / n$. We associate to each object its *rounded value*: $\bar{v}_i = \lfloor v_i / K \rfloor \in \{1, \dots, \lfloor n/\epsilon \rfloor\}$. We denote by $\overline{\mathrm{value}}(S) = \sum_{i \in S} \bar{v}_i$ the rounded value of a given set of objects S. We denote by $\mathrm{OPT}' = \max\{\overline{\mathrm{value}}(S) : S \subseteq \{1, \dots, n\}, \mathrm{size}(S) \leqslant C\}$ the optimum rounded value of a subset of objects fitting in the knapsack.

Let S^* be the optimal solution computed by the algorithm of question 1.1 for the rounded instance with sizes $s_1, \dots, s_n$ and values $\bar{v}_1, \dots, \bar{v}_n$.

► ***Question* 1.2*)*** *Show that* $\mathrm{value}(S^*) \geqslant (1 - \epsilon) \cdot \mathrm{OPT}$. *What is the overall time complexity of the resulting PTAS in n and ϵ?*
▷ Hint. *Show that* $\mathrm{value}(S^*) \geqslant \mathrm{value}(O) - nK$ *for all optimal solution O of the initial instance.*

■ **Exercise 2 (Constant Time Approximation Scheme).** (★) A *matching* in an undirected graph $G = (V, E)$ is a subset $M \subseteq E$ of edges with no extremities in common, i.e.: $\forall (e, e') \in M^2,\ e \neq e' \Rightarrow e \cap e' = \varnothing$. A matching is *maximal* if it is maximal for the inclusion (no more edge can be added to it without violating the matching property).

We consider the following problem: Given an undirected graph $G = (V, E)$ with maximum degree δ, determine the size of one of its maximal matchings.

Assume that we assign to each edge e a real number r_e chosen independently and uniformly in $[0, 1]$. Consider the following greedy algorithm: Start with an empty matching M ; Consider each edge e in order of increasing r_e, and add e to M if e does not intersect any edge already in M.

► ***Question* 2.1*)*** *Show that the greedy algorithm computes a maximal matching that depends only on r (almost surely, i.e. with probability 1). We will denote this matching $M(r)$.*

▶ ***Question* 2.2*)*** *Show that an edge e belongs to $M(r)$ if and only if none of its incident edges e' with $r_{e'} < r_e$ belongs to $M(r)$.*

Consider the following recursive procedure $\texttt{MagicTest}(e)$ for testing if e belongs to $M(r)$: output $\texttt{true}$ if for all edge e' incident to e with $r_{e'} < r_e$, we have $\texttt{MagicTest}(e')$ = $\texttt{false}$; and output $\texttt{false}$ otherwise. (Note that $\texttt{MagicTest}(e)$ outputs $\texttt{true}$ (the neutral element for $\texttt{and}$) if e has no incident edge e' with $r_{e'} < r_e$).

▶ ***Question* 2.3*)*** *Show that* $\texttt{MagicTest}$*(e) determines correctly and after a* finite *number of steps whether e belongs to $M(r)$ or not.*

▶ ***Question* 2.4*)*** *Show that for any given edge e with $r_e = \rho$, the expected number of recursive calls made by* $\texttt{MagicTest}$*(e) (over all possible values of r conditioned to $r_e = \rho$) is upper bounded by the function $Q(\rho)$ verifying:*

$$Q(0) = 1 \text{ and } Q(\rho) = 1 + (2\delta - 2) \int_0^\rho Q(x)dx.$$

Solve this system to prove that $Q(\rho) = e^{(2\delta-2)\rho}$. Conclude that the expected number of recursive calls made by $\texttt{MagicTest}$*(e) is upper bounded by $e^{2\delta-2}$ for all edge e.*

Consider the following algorithm for estimating the size of a maximal matching: First, draw the random values (r_e); then repeat t times: draw an edge $e \in E$ at uniformly and independently at random and run $\texttt{MagicTest}(e)$; Let s be the number of times $\texttt{MagicTest}(e)$ answered true and output the value $\vartheta = \frac{m \cdot s}{t}$.

▶ ***Question* 2.5*)*** *Show that for fixed r, the expectation of the output value ϑ is $|M(r)|$. Show moreover that if $t \geqslant c \cdot \frac{\delta^2}{\epsilon^2}$ for some suitably chosen constant c, then:*

$$\Pr\Big\{ |\vartheta - M(r)| \leqslant \epsilon \cdot |M(r)| \Big\} \geqslant \frac{2}{3}.$$

Hint. *Show that for all r, $\frac{m}{2\delta-1} \leqslant |M(r)| \leqslant m$ and use Hoeffding's inequalities.*

We conclude that with probability $\geqslant 2/3$, our algorithm returns an estimate of the size of $M(r)$ with a relative error $\leqslant \epsilon$. We call such an algorithm a ϵ-estimator of $|M(r)|$.

▶ ***Question* 2.6*)*** *How many values of r_e does our algorithm use in expectation? How to avoid drawing all the r_e beforehand? Conclude with an ϵ-estimator running in constant time $O(\frac{\delta^2 \cdot e^{2\delta-2}}{\epsilon^2})$ in expectation for the problem of determining the size of a maximal matching in a graph G with maximum degree δ.*

Note. Estimating the size of a *maximal* matching is not really interesting. However, with the same technics, one can modify the $\texttt{MagicTest}$ procedure to construct an ϵ-estimator running in constant expected time for the size of the *maximum* matching in a graph with maximum degree δ.