TD3 Approximation Algorithms

Wed. Oct. 10, 2012 - **Due on Mon. Oct. 29, 2012**

You are asked to complete the exercises and send me your solutions by email at:
nicolas.schabanel@liafa.univ-paris-diderot.fr
(or drop it in my mail box on the 7th floor) on or before **Monday Oct. 29**.

■ **Exercise 1 (FPT algorithm for spotting k disjoint triangles).** Given $G = (V, E)$ an undirected graph ($n = |V|$ and $m = |E|$) and k an integer, we are looking for k vertex-disjoint triangles in G. We are looking for an algorithm of time complexity $O(f(k)n^c m^{c'})$ where the exponents c and c' are constant, independent of k. Consider the following randomized algorithm

---

Algorithm 1 FPT randomized algorithm for k disjoint triangles

---

- Choose independently for each vertex u a color $c_u \in \{1, \dots, 3k\}$ uniformly at random.
- **return** "Yes" if there is a *colorful* solution, i.e. a set of k triangles whose $3k$ vertices use exactly once each color; **return** "I don't know" otherwise.

▶ ***Question* 1.1)** *Show that if G contains k disjoint triangles, then the probability that the algorithm answer "Yes" is at least e^{-3k}.*
▷ Hint. *use that : $k! \geqslant (k/e)^k$ for all k.*

▶ ***Question* 1.2)** *How many times should you run this algorithm to improve success probability to $1/2$?*

In order to check whether a colorful solution exists, we propose to try all permutations π on $\{1, \dots, 3k\}$ and check if there is are triangles of colors $(\pi_1, \pi_2, \pi_3), \dots, (\pi_{3k-2}, \pi_{3k-1}, \pi_{3k})$.

▶ ***Question* 1.3)** *Describe an algorithm that decides if such a collection of triangles exists. What is the overall expected time complexity in k, n and m, of the algorithm that uses this method to return k disjoint triangles with probability at least $1/2$ if they exists in G? What is the time complexity if k is fixed?*

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

▶ ***Question* 2.3**) *Show that* $\mathtt{MagicTest}(e)$ *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* $\mathtt{MagicTest}(e)$ *(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 $\mathtt{MagicTest}(e)$ *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 $\mathtt{MagicTest}(e)$; Let s be the number of times $\mathtt{MagicTest}(e)$ 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 $\mathtt{MagicTest}$ 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.