Prof. Friedrich Eisenbrand
Dr. Yuri Faenza

Discussions from: November 19, 2013

Combinatorial Optimization

Fall 2013

Assignment Sheet 5

Exercises marked with a $\star$ can be handed in for bonus points. Due date is December 3.

Recall that in class we saw that the randomized contraction (Karger's) algorithm provides a minimum cut with probability at least n^{-2} in time $O(n^2)$. Hence in $O(n^4)$ time (i.e. after n^2 repetitions of randomized contraction), the probability that we obtain a min cut is at least $1-(1-n^{-2})^{n^2} \geq 1-e^{-1} = 1-1/e$, hence a strictly positive constant.

Exercise 1

Consider the following algorithm to compute a min-cut in a graph $G(V,E)$. While the graph has at least 3 vertices, take uniformly at random a pair of distinct vertices $u, v \in V$, and contract them. Then output the only cut of the remaining graph. Is this a good algorithm?

Exercise 2

Consider the following algorithm for computing a min-weight cut in a graph.

Iterative-contraction

Input: a weighted graph G
Output: a cut of G

1. **If** $|V| \leq 6$, then **return** a minimum weighted cut of G by enumeration.
2. **Else**
   a) Set $t = 1 + \lceil |V|/\sqrt{2} \rceil$.
   b) Apply twice the random contraction algorithm to the graph G, and each time stop when your graph has t vertices. Call the resulting graphs H_1 and H_2, respectively.
   c) Compute Iterative-contraction(H_1) and Iterative-contraction(H_2), and **return** the cut with smallest weight among the two.

(i) Show that Iterative-contraction runs in $O(n^2 \log n)$ time.

(ii) [$\star$] Show that Iterative-contraction provides a min cut with probability $\Omega(1/\log n)$. (Hint: What is the probability that random contraction does not "kill" a minimum cut after $n-t$ iterations?)

(iii) [$\star$] Compare Iterative-contraction with the randomized contraction algorithm.

Exercise 3
Consider the following modification of the random contraction algorithm: Apply the algorithm until the graph is reduced to a graph G' with t vertices. Then, compute a minimum cut on G' (this requires $O(|V(G')|^3)$. Show that t can be tuned so that this routine can be repeated as to provide an algorithm with running time $\theta(n^{8/3})$ that computes a minimum cut with probability at least 1/2. Show that this is tight (i.e., we cannot choose t as to achieve a better running time, and at the same time have a probability of success of at least 1/2).

Exercise 4
Given a weighted graph $G(V,E)$, you aim at finding a collection $\mathscr{C}$ of cuts of G such that, for each $s,t \in V$, there exists a min (s,t)-cut in G that is contained in $\mathscr{C}$. How small can $|\mathscr{C}|$ be? Show examples proving that your bound is tight. Can you compute such a minimum $\mathscr{C}$ in polynomial time?

Exercise 5
Given a graph $G(V,E)$ and any two nodes s,t, show that a minimum cut in G is either a minimum (s,t)-cut, or a minimum cut in the graph obtained from G by contracting s and t.

Exercise 6 ($\star$)
An ordering $v_1,\dots,v_n$ of the vertices of a graph is called *good* if each v_i realizes $\max_{v\notin\{v_1,\dots,v_{i-1}\}} |\{uv : u \in \{v_1,\dots,v_{i-1}\}\}|$. Consider the following algorithm to compute a min-cut in a graph $G(V,E)$.

Input: an unweighted graph G
Output: the cardinality of a minimum cut of G

1. Set $G^n = G(V,E)$.
2. **For** $k = n, n-1, \dots, 2$:
   a) Compute a good ordering $v_1,\dots,v_{|V(G^k)|}$ of G^k, and let x^k be the degree of $v_{|V(G^k)|}$ in G^k.
   b) Define G^{k-1} from G^k by contracting $v_{|V(G^k)|}$ and $v_{|V(G^k)|-1}$.
3. Output the minimum of $x^2,\dots,x^n$.

(i) Prove that the algorithm is correct. (Hint: use the previous exercise).

(ii) Can one deduce from the output the set of vertices in a minimum cut?

Exercise 7
Prove or disprove the following: in each graph $G(V,E)$ with minimum degree d, there exists a pair of nodes s,t with d edge-disjoint paths between them in G.