Succinct Graph Structures and Their Applications	**Spring 2018**

Exercise 4: June 07

Lecturer: Merav Parter

Low-Diameter Decomposition

A *low-diameter* decomposition of a graph $G = (V, E)$ and a parameter D is a randomized partitioning of the vertices V into $V_1, \ldots, V_t$ such that:

(P1) the *weak-diameter*[1] of each $G[V_i]$ is at most D, and

(P2) for every $u, v \in V$, $Pr(u \in V_i \text{ and } v \in V_{j \neq i}) = \alpha \cdot \texttt{dist}_G(u, v)$ for $\alpha = O(\log n/D)$.

Exercise 1. In this exercise, we consider a candidate algorithm for computing a low-diameter decomposition. For a vertex v and integer r, let $B(u, r) = \{v \in V \mid \texttt{dist}_G(u, v) \leq r\}$ be the r-radius ball of u in G.

Algorithm Decomp(G, D)

1. Pick a radius $\delta \in [D/8, D/4]$ at random.
2. Consider the vertices in an *arbitrary* order π.
3. The i^{th} set V_i is the set of all vertices in $B(\pi(i), \delta) \setminus \bigcup_{j<i} B(\pi(j), \delta)$

Figure 4.1: A Low-Diameter Decomposition Algorithm?

Prove or disprove: the sets $G[V_1], \ldots, G[V_n]$ satisfy the properties (P1) and (P2) w.h.p. In case you think the algorithm is incorrect, suggest how to fix it (along with a proof that your fix works).

Exercise 2. We now turn to consider a deterministic procedure from computing low-diameter decomposition. The benefit of this procedure is that it also handles multi-graphs (where a given edge might have several copies in the graph). In addition, the clusters computed by this procedure will have small *strong-diameter*[2]. Let $c(e)$ be the number of copies of an edge $e \in G$ and for a subset of edges $F \subseteq E(G)$, let $C(F) = \sum_{e \in F} c(F)$. Let $E(u, r) = \{(x, y) \in E(G) \mid x, y \in B(u, r)\}$ be the G-edges connecting vertices in $B(u, r)$.

The input to the decomposition algorithm DetDecomp (see Fig. 4.2) consists of (1) a multi-graph $G = (V, E, c)$ (where each edge $e \in E$ has $c(e)$ copies in G), and (2) a desired diameter parameter D. Prove that the following lemma holds.

Lemma. *Consider an unweighted undirected graph $G = (V, E)$ with $C = C(E)$ and let $\alpha = 4\ln(C)/D$. Then Alg.* DetDecomp(G, D, α) *returns subsets $V_1, \ldots, V_k$ s.t:*

(Q1) The strong diameter of each subgraph $G[V_i]$ is at most D.

(Q2) There are at most $\alpha \cdot C(E)$ inter-cluster edges (i.e., edges connecting $u \in V_i$ and $v \in V_{j \neq i}$). (This is the deterministic *equivalent of property (P2) in Exercise 1).*

[1] The weak diameter of a subgraph $G' \subseteq G$ with respect to G is $\max_{u,v \in G'} \texttt{dist}(u, v, G)$.
[2] The strong diameter of a subgraph $G' \subseteq G$ is $\max_{u,v \in G'} \texttt{dist}_{G'}(u, v)$.

Algorithm DetDecomp($G = (V, E, c), D, \alpha$)

1. Set $\ell \leftarrow 1$.
2. While G is nonempty do:
   (a) Pick a vertex v in G.
   (b) Let r_v be the smallest r satisfying that $C(E(v, r+1)) \leq (1+\alpha)C(E(v, r))$.
   (c) $V_\ell \leftarrow B(v, r_v)$.
   (d) $\ell \leftarrow \ell + 1$.
   (e) Remove all vertices of V_ℓ from G (along with their edges).
3. Return $V_1, \ldots, V_k$.

Figure 4.2: Deterministic low-diameter decomposition algorithm

Trees with Small Average Stretch

We showed in classes 06 and 07, the construction of distribution over trees such that the expected stretch of each pair u, v (when sampling a tree from the distribution) is bounded by α. A dual problem considers the construction of a *single* tree (either a subgraph of G or not) that has a small *average* stretch over all edges (u, v) in G. Formally, given an unweighted graph $G = (V, E)$ and a tree T with $V(G) \subseteq V(T)$, define the average stretch of T by: $1/|E(G)| \cdot \sum_{(u,v)\in E} \texttt{dist}_T(u, v)$.

Exercise 3. (a) Given n even, let W_n be the wheel graph consisting of n vertex ring C_n together with chords joining antipodal points on the ring. Find a tree $T \subseteq W_n$ with average stretch at most 8/3. (b) Show that the 2-dimensional $\sqrt{n} \cdot \sqrt{n}$ grid has a spanning tree with average stretch $O(\log n)$.

Exercise 4. In class 06, we showed a randomized construction of a tree T such that $V(G) \subseteq V(T)$ and $\texttt{dist}_G(u, v) \leq Exp(\texttt{dist}_T(u, v)) \leq \alpha \cdot \texttt{dist}_G(u, v)$ for every $u, v \in V(G)$. Adapt this construction to provide a tree T (where $V(G) \subseteq V(T)$) with *average stretch* at most α.

Applications of Tree Embedding

Low-stretch tree embeddings have many applications in approximation algorithms. The general recipe is to solve the first the problem of interest on a ree T of G that is sampled from the low-stretch tree distribution, and then to "translate" the solution back to G. We will now see one such example.

Exercise 5. In the k-median problem, we are given a graph $G = (V, E)$ and parameter $k \geq 1$, the goal is to find a subset $C \subseteq V$ of k vertices that minimizes $\sum_{v \in V} \texttt{dist}_G(v, C)$ where $\texttt{dist}_G(v, C) = \min_{c\in C} \texttt{dist}_G(v, c)$. Whereas the k-median problem is NP-hard for general graphs, one can compute an exact solution (hint: dynamic programming) for the problem in time $\mathsf{poly}(n, k)$ when the given graph is a *tree*.

Provide a randomized[3] polynomial time algorithm for the problem that computes a set $C' \subseteq V$ of size k such that $Exp(\sum_{v\in V} \texttt{dist}_G(v, C')) \leq \alpha \cdot OPT$ where $OPT = \sum_{v\in V} \texttt{dist}_G(v, C^*)$ is the cost of the optimal algorithm. You can use the algorithms of classes 06 and 07 as black-box. Do we have to use an α-stretch spanning tree T which is *subgraph* of G? or would it be sufficient that $V(G) \subseteq V(T)$?

[3] The randomization part comes from the randomized algorithm for constructing the tree.