Introduction

Kelner and Madry, FOCS 2009

Algorithm for dense graphs: Colbourn, Myrvold and Neufeld'96 gave an $O(n^{2.376})$ algorithm.

Algorithm for sparse graphs: An $O(mn)$ algorithm based on the following theorem, proved independently by Broder'89 and Aldous'90.

Theorem 1 (Theorem 1)**.** *Suppose you simulate a random walk in $G = (V, E)$ starting from an arbitrary vertex s until all vertices has been visited. For each $v \in V \backslash \{s\}$, let e_v be the edge through which v was visited for the first time. Then $T = \{e_v\}$ is a uniformly random spanning tree of G.*

Today: I will give an $O(m\sqrt{n} \log 1/\delta)$ algorithm to produce a δ-random spanning tree of G. Meaning that the probability of generating a tree T is between $\frac{1-\delta}{|\mathcal{T}(G)|}$ and $\frac{1+\delta}{|\mathcal{T}(G)|}$. I will first present an $O(m^2 \log 1/\delta/\sqrt{n})$ algorithm.

Plan:

1. Building good decompositions
2. Computing the shortcutting probabilities
3. Bounding the expected simulation time

1 - Strong (ϕ,γ)-decompositions

Let $S \subseteq V(G)$ and $D_1,\dots,D_k$ be the connected components of $G-S$. Let $C = E(G)\setminus \cup E(D_i)$. For $X \subseteq V(G)$, let $P(X)$ be the set of vertices of X that have a neighbour outside. Then $(D_1,\dots,D_k,S)$ is a (ϕ,γ)-*decomposition* if

1. $|C| \le \phi|E(G)|$,
2. $|P(S)| \le \phi|V(G)|$.
3. $\forall i$ the diameter of $D_i \le \gamma$,
4. $\forall i$ $|\delta(D_i)| \le |E(D_i)|$.

Lemma 2 (Lemma 13)**.** *For any G and any $\phi = o(1)$, a strong $(\phi, \widetilde{O}(1/\phi))$-decomposition of G can be computed in time $\widetilde{O}(|E(G)|)$.*

Proof.

- $B(v,j)$ be the ball of radius j around v,
- $R(v,j)$ be the sphere of radius j around v,
- $R^+(v,j) = E(B(v,j+1))\setminus E(B(v,j))$,
- $R^-(v,j) = E(B(v,j))\setminus E(B(v,j-1))$,
- $t = \phi/(1-\phi)$.

Run the following algorithm

```
while G is nonempty do
   choose an arbitrary vertex v and set j = 0
   while |R(v, j+1)| > t |V(B(v, j))| OR |R+(v, j+1)| > t |E(B(v,j))|
   OR |R-(v, j+1)| > t |E(B(v,j))| do
      j = j + 1
   Suppose that you stop at j
   Add R(v, j+1) to S and the ball B(v, j) as a new component Di
   Delete S, Di and all the incident edges from G
```

1. For each i, the number edges added to C is at most t times the edges in $E(D_i)$, so $|C| \le \phi|E(G)|$.
2. For each i, the number of vertices added to S is at most t times the vertices in $V(D_i)$, so $|P(S)| \le |S| \le \phi|V(G)|$.
3. For each i, we claim the diameter of D_i is $\le 6(1+\ln|E(G)|)/(\ln(1+t)) = O(\frac{\ln m}{-\ln(1-\phi)}) = O(\ln m/\phi)$. Assume not. Then for some i, a particular one of the conditions has been triggered more than $j/3 = 1+\ln|E(G)|/\ln(1+t)$ times. If it was the first condition, the ball $B(v,j)$ would have more than $(1+t)^{j/3} \ge |E(G)|$ vertices! If it was the second (or third) one, then the ball $B(v,j)$ would have more than $(1+t)^{j/3-1} \ge |E(G)|$ edges!
4. If there is a D_i with $|\delta(D_i)|$, then add $V(D_i)$ to S. The size of C becomes at most twice, and the size of $P(S)$ do not change.

$\square$

2 - The Shortcutting Probabilities

For $v \in D_i$ and $e \in \delta(D_i)$, let $P_v(e)$ be the probability of the random walk leaving D_i through e after entering D_i through vertex D_i.

Lemma 3 (Lemma 9)**.** *Given a (ϕ, γ)-decomposition of G, we can compute multiplicative $(1+\epsilon)$-approximations of all of the $P_v(e)$ in time $\widetilde{O}\left(\phi m^2 \log 1/\epsilon\right)$.*

Proof. Fix a $D = D_i$ and $e = (u, u') \in \delta(D)$ with $u \in V(D)$. Build graph D' as follows. Add vertex u' and a dummy vertex u^* to D, then for each $(w, w') \in \delta(D) \backslash \{e\}$, $w \in V(D)$, add an edge (w, u^*). Notice that for any $v \in V(D)$, $P_v(e)$ is exactly the probability that a random walk in D' started at v will hit u' before it hits u^*.

Now, if we treat D' as an electrical circuit with unit resistance on each edge, in which we impose voltage $+1$ at u' and 0 at u^*, then the voltage achieved at v is equal to $P_v(e)$. We can compute a $(1+\epsilon)$-approximation of all such voltages in time $\widetilde{O}\left(|E(D')| \log 1/\epsilon\right)$ using the linear system solver of Spielman and Tang. (an $n \times n$ Laplacian matrix, which multiplied by the voltages vector gives the external current in each vertex by Kirchoff's node law)

The number of such e's is $|C|$, so the total running time is

$$\widetilde{O}\left(|C| \sum |E(D_i')| \log 1/\epsilon\right) = \widetilde{O}\left(|C| \sum |E(D_i)| \log 1/\epsilon\right) = \widetilde{O}\left((\phi m) m \log 1/\epsilon\right).$$

□

Lemma 4 (Lemma 10)**.** *To get a δ-random spanning tree, one may choose $\epsilon \leq \delta / mn$.*

3 - Bounding the Expected Simulation Time

Lemma 5 (Fact 5)**.** *The expected number of steps that we the walk moves on edges in C is ϕmn .*

Proof. Recall that the cover time of G in the original random walk is $O(mn)$. Now, note that $|C| \leq \phi m$. ◻

Lemma 6 (Lemma 6)**.** *The expected number of steps that the walk moves on edges of D_i before covering it is $\widetilde{O}\left(|E(D_i)| diam(D_i)\right)$.*

Proof. The cover time of (usual random walk in) a graph G is $\widetilde{O}\left(m \log n diam(G)\right)$. The proof is nontrivial. Here it is important that $|\delta(D_i)| \leq |E(D_i)|$. ◻

Lemma 7 (Lemma 8)**.** *The expected simulation time is $\widetilde{O}\left(m\gamma + \phi mn\right)$.*

Obtaining a Faster Algorithm

For a given i, some $v \in V(D_i)$, and $u \in S$, define $Q_v(u)$ to be the probability that u is the first vertex out of $V(D_i)$ that is reached by a random walk that starts at v.

Lemma 8 (Lemma 14)**.** *All* $Q_v(u)$ *can be computed in time* $\widetilde{O}(\phi mn \log 1/\epsilon)$.

Proof. Fix $D = D_i$ and $u \in P(S)$. Consider a new graph D' that we obtain from $G[D \cup P(S)]$ by merging all vertices in $S \backslash \{u\}$ into one vertex u^*. Then $Q_v(u)$ is the probability that a random walk in D' starting from v will hit u before u^*. We can find all such probabilities in time $\widetilde{O}(|E(D')| \log 1/\epsilon)$ by treating the graph as an electrical network and using the linear system solver of Spielman and Teng. The running time is bounded by $\widetilde{O}(|P(S)| \sum |E(D_i)| \log 1/\epsilon) = \widetilde{O}(\phi nm \log 1/\epsilon)$. $\square$

But e_v may be undefined for some $v \in P(S)$. To solve this, we remove all edges e_v with $v \in P(S)$ defined. Then contract the connected components. We will get a graph with $\leq \phi n$ vertices. Build a random spanning tree using the algorithm of Colbourn et al. in time $O((\phi n)^{2.376})$.

The total running time is $O(m\sqrt{n} \log 1/\delta)$.