15-750: Graduate Algorithms February 13, 2017

Lecture 12: Low Diameter Decomposition using Exponential Delays

Lecturer: Gary Miller *Scribe: Andrew Chung*

1 Introduction

1.1 Exploring a Graph Using BFS

There are at least three methods to explore a graph:

1. DFS (earlier lectures)
2. BFS (today)
3. Random Walks

1.2 Applications of Low Diameter Decomposition

1. Spanners: Distance preserving sparse graphs.
2. Hop Set: Added set of extra edges to a graph to decrease number of edges used in shortest paths.
3. Low Stretch Spanning Tree (LSST): Preserve distances on average. Applications of LSSTs include fast algorithms for:
   (a) Linear solvers
   (b) Max flow
   (c) Image processing

2 Definitions

1. Undirected and unweighted graph $G = (V, E)$.
2. $n \equiv |V|$
3. $m \equiv |E|$
4. $d(v) \equiv$ degree of $v \in V$

Definition 2.1. $Vol(W) = \sum_{v \in W} d(v)$, where $W \subseteq V$

Note: $Vol(V) = 2m$, since each edge in the graph is counted twice.

Definition 2.2. $Boundary(W) \equiv \partial W = \{(x, y) \mid x \in W, \text{and} y \notin W, (x, y) \in E\}$, where $W \subseteq V$.

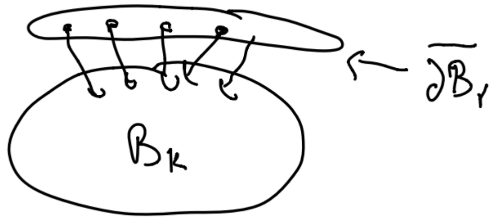
Intuitively, ∂W is the set of outgoing edges from the cluster that connect to vertices in W.

Definition 2.3. The Isoperimetric Number of W $\equiv \Phi(W) = \frac{|\partial W|}{Vol(W)}$

The Isoperimetric Number of W is the fraction of outgoing edges to the double-counted edges remaining within the cluster W.

Definition 2.4. The distance $dist(v, w)$ is the minimal distance between two vertices $v, w \in G$.

Figure 1: An illustration of $\overline{\partial B_r}$.



3 Low Diameter Decomposition

3.1 Problem Statement

Given $G = (V, E)$, $x \in V$, and $0 < \beta < 1$, **want to find:**

$$x \in W \subseteq V \text{ of nearby points such that } \Phi(W) \leq \beta$$

3.2 Ball Growing

Definition 3.1. $B(x, r) = \{y \in V \mid dist(x, y) \leq r\}$

We can think of $B(x, r)$ as a "ball" of radius r, centered at x. The cluster (or ball) consists of vertices that are no further than a distance of r from x.

Algorithm 1 Ball Growing

function GrowBall($G = (V, E), x, \beta$)
  $r \leftarrow 1$
  while $\Phi(B(x, r)) > \beta$ **do**
    $r \leftarrow r + 1$
  end while
  return $B_r = B(x, r)$, $R = r$
end function

Claim 3.2. $R = O(\frac{\log(m)}{\beta})$, *where R is the largest radius returned from GrowBall.*

Note: If $r < R$, then $|\partial B_r| \geq \beta \cdot Vol(B_r)$

Definition 3.3. $\overline{\partial B_r} = \{y \in V \mid (x, y) \in E, x \in B_r, y \notin B_r\}$

Intuitively, $\overline{\partial B_r}$ is the set of vertices that are "neighbors" of the cluster B_r. They are the vertices that are in consideration to be added in at the next increment of r. For an illustration, please refer to Figure 1.

Note: $Vol(\overline{\partial B_r}) \geq \beta \cdot Vol(B_r)$.
We trivially get this from $|\partial B_r| \geq \beta \cdot Vol(B_r)$
Thus: $Vol(B_{r+1}) \geq (1 + \beta) \cdot Vol(B_r)$

Going back to proving Claim 3.2:

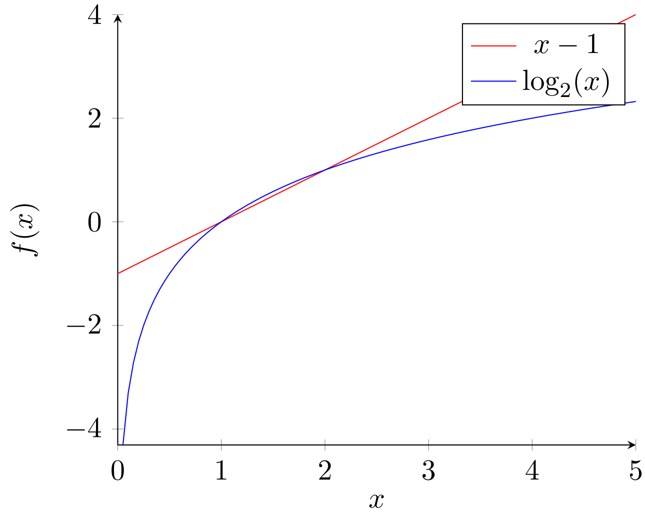


Figure 2: $\log_2(1+\beta) \geq \beta$ for $0 \leq \beta \leq 1$

Proof.

$$\begin{aligned}
(1+\beta)^r &\leq Vol(B_r) \leq 2m \\
\text{Taking logs} &\implies r \cdot \log_2(1+\beta) \leq \log_2(2m) \\
\text{Provided that } \log_2(1+\beta) \geq \beta \text{ for } 0 \leq \beta \leq 1 &\implies r \cdot \beta \leq \log_2(m) + 1 \\
&\implies r \leq \frac{\log_2(m)+1}{\beta}
\end{aligned}$$

□

3.3 Low Diameter Decomposition Through Ball Decomposition

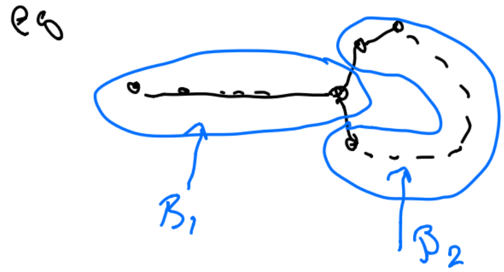
We now introduce a simple sequential algorithm *BallDecomp* that uses *GrowBall* to get a partition of V.

Algorithm 2 Ball Decomposition

function BALLDECOMP($G = (V, E), \beta$)
  $balls \leftarrow \emptyset$
  while $V \neq \emptyset$ **do**
    Pick $x \in V$ ▷ Pick a arbitrary vertex as the center for creating a new ball
    $B_r \leftarrow GrowBall(G, x, \beta)$ ▷ Creates a ball with x as the center
    $balls \leftarrow \cup\{B_r\}$ ▷ Add the new ball to the set of balls
    Remove B_r and ∂B_r from G ▷ Remove components of the ball from the graph
  end while
  return $balls$
end function

Note: $dist_G(V, W) \ll dist_B(V, W)$ for $V, W \in B \equiv$ Ball. Essentially, the distance between two vertices within a ball may be much greater than their distance in the whole graph. For an illustration, please refer to Figure 3.

Figure 3: An illustration of distance between balls in a graph.



3.4 Ball Growing Using Exponential Decay

In this section, we will introduce a ball-growing technique using the properties of exponential distributions.

3.4.1 Algorithm definition

Algorithm 3 Ball Growing Using Exponential Delay

function EXPONENTIALDELAY($G = (V, E), \beta$)
    for each $v \in V$ draw $X_v \sim Exp(\beta)$
    $X_{max} \leftarrow \max_{v \in V} X_v$
    for each $v \in V$ compute $S_v \leftarrow X_{max} - X_v$
    $t \leftarrow 0$
    while True **do**
        for each $v \in V$ **where** $S_v = t$ **do**
            if v is not owned at time S_v **then**
                v owns v, start BFS from v
            else
                v is owned by first arrival vertex, do nothing
            end if
        end for
        if All $v \in V$ are owned **then**
            break
        end if
        $t \leftarrow t + 1$
    end while
end function

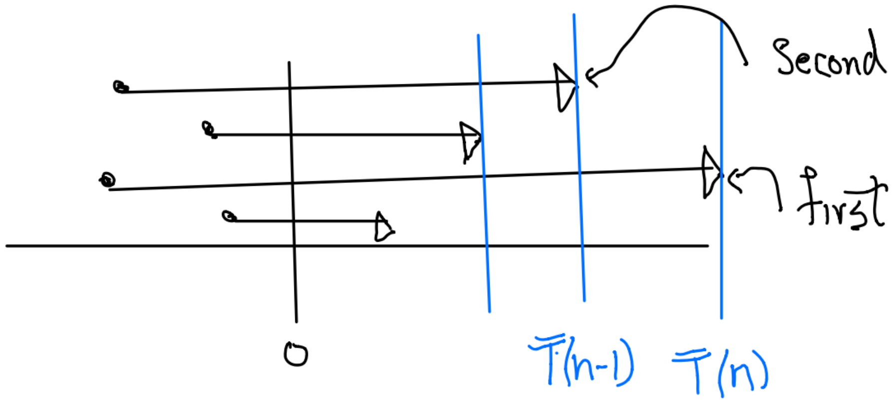
Note: At each time step each of the active BFSs move outward a distance of 1.

Definition 3.4. $u \in cluster(v)$ if

1. $v = \arg\min_v \{dist(u, v) + S_v\}$ **or**
2. $v = \arg\max_v \{X_v - dist(u, v)\}$,

where $u, v \in V$ and S_v and X_v are defined as in Algorithm 3.

Figure 4: An illustration of the horse race and photo finish framing of the intercluster edges problem.



We can think of S_v as an additional distance that v has to travel to u, and u will belong to the cluster centered at v whose total distance (including S_v) from v to u is the smallest.

3.4.2 What is the maximum cluster radius?

Lemma 3.5. *At time X_{max}, all of the nodes will be owned.*

Proof. Intuitively, at time X_{max} each node must have either been owned by another vertex or has started its own BFS. □

Corollary 3.6. *Max cluster radius $\leq X_{max} \leq \frac{2 \ln n}{\beta}$ with probability $\geq 1 - \frac{1}{n}$, where $n = |V|$.*

This follows from Lemma 3.5 and $X_{max} \sim Exp(\beta)$.

Note: Remember that the max of n $Exp(\beta)$'s is at most $\frac{2 \log n}{\beta}$ with high probability.

3.4.3 What is the probability that an edge is intercluster?

In other words, what is the probability that an edge is cut?

To answer this question, let e be some edge and c be the midpoint of edge e. We think of each vertex doing a BFS of G starting at time $S_{v_i} = S_i$.

Definition 3.7. The arrival time at c will be a random variable

$$T_i = X_{max} - X_i + dist(v_i, c) = S_i + dist(v_i, c)$$

$$\textbf{Early arrival}: \overline{T_i} = X_{max} - T_i = X_i - dist(v_i, c)$$

The probability that an edge is intercluster is bounded above by the probability that the difference between two arrival times is less than a unit of time.

A way to frame this problem is to think of it as a horse race and photo finish, where $dist(v_i, c)$ can be thought of as the handicap and X_i can be thought of as the speed. Figure 4 serves as an illustration.

Definition 3.8. $Gap_1 = \overline{T}(n) - \overline{T}(n-1)$

By the memoryless property of exponential distributions, $Gap_1 \sim Exp(\beta)$, and thus $Pr(Gap_1 < 1) = 1 - e^{-\beta}$.

Claim 3.9. $1 - e^{-\beta} < \beta$

Proof.

$$\begin{aligned} e^{-\beta} &= 1 - \beta + \frac{\beta^2}{2!} - \frac{\beta^3}{3!} + \dots \\ \implies 1 - e^{-\beta} &= \beta - \frac{\beta^2}{2!} + \frac{\beta^3}{3!} - \dots \\ \implies 1 - e^{-\beta} &< \beta, \text{by Taylor's Theorem} \end{aligned}$$

□

Hence, we show that the probability that an edge is intercluster is $< \beta$.

4 Exponential Delay

Theorem 4.1. *Exponential delay generates a clustering such that*

1. *Max radius in expectation is* $\frac{\ln(n)}{\beta}$
2. *Max radius is* $\frac{2\ln(n)}{\beta}$ *with probability 1 -* $\frac{1}{n}$
3. *The expected number of intercluster edges is* $m \cdot \beta$
4. *Run time is* $O(m+n)$
5. ***Strong Diameter Property****: If* $w \in Cluster_v$*, then shortest path from* v *to* w *is in cluster* v*, where* v *is the center of the cluster and* $v, w \in V$