Random Rates for 0-Extension and Low-Diameter Decompositions

Anupam Gupta
Carnegie Mellon University

Kunal Talwar
Microsoft Research

Follow this and additional works at: http://repository.cmu.edu/compsci
Part of the Computer Sciences Commons
Random Rates for 0-Extension and Low-Diameter Decompositions

Anupam Gupta∗ Kunal Talwar†

Abstract

Consider the problem of partitioning an arbitrary metric space into pieces of diameter at most \( \Delta \), such that every pair of points is separated with relatively low probability. We propose a rate-based algorithm inspired by multiplicatively-weighted Voronoi diagrams, and prove it has optimal trade-offs. This also gives us another algorithm for the 0-extension problem.

1 Introduction

We consider partitioning problems of the following form: given a metric \((V, d)\), how should we decompose it into “small” pieces so as to cut “few” edges. There are many variants of this general form, and in this note we consider two of them: terminal partitioning/0-extension and low-diameter decompositions.

In the low-diameter decomposition problem, we are given a metric \((V, d)\) and a diameter bound \( \Delta \), and the goal is to (randomly) partition the set \( V \) into pieces each of diameter at most \( \Delta \) so that for any pair \( x, y \in V \),

\[
\Pr[x, y \text{ separated}] \leq \beta \cdot \frac{d(x, y)}{\Delta}.
\]

It is known that \( \beta = O(\log n) \) is possible for any \( n \)-point metric, and there are metrics for which no better is possible. Such decompositions have been widely studied, e.g., works by Awerbuch [Awe85], Linial and Saks [LS93], Leighton and Rao [LR99], Garg, Vazirani, and Yannakakis [GVY96], and Seymour [Sey93] studied an equivalent deterministic version of this problem, and Bartal [Bar96], Calinescu, Karloff, and Rabani [CKR05], Fakcharoenphol, Rao, and Talwar [FRT04], and Abraham, Bartal, and Neiman [ABN06] studied randomized versions. (This is almost certainly an incomplete list — though some other pertinent references follow.) Many of these results study more nuanced parameters and give bounds that improve on \( O(\log n) \) for special cases, but we omit discussions of these for sake of brevity.

The terminal partitioning problem can be thought of as a multi-scale version of low-diameter decomposition. This name is not standard (we coin it here), but it arises in solving the 0-extension problem. In terminal partitioning, instead of a diameter bound, we are given a set of terminals \( T \subseteq V \) and \(|T| = k\), and we want a (random) partition \( V_1, V_2, \ldots, V_k \), such that the \( i^{th} \) terminal \( t_i \in V_i \), and for any \( x, y \in V \),

\[
\Pr[x, y, \text{ separated}] \leq \alpha \cdot \frac{d(x, y)}{\min\{d(x, T), d(y, T)\}}.
\]

In other words, edges whose endpoints are far away from the terminal set should be cut with smaller probability than edges whose endpoints are close to terminals, a natural enough requirement. Again, it is known that \( O(\log k) \) is possible for any metric [CKR05]; however, this is not the best possible in this case [FHRT03].

∗Department of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213, and Microsoft Research SVC, Mountain View, CA 94043. Research was partly supported by NSF awards CCF-0964474 and CCF-1016799, and by a grant from the CMU-Microsoft Center for Computational Thinking.

†Microsoft Research SVC, Mountain View, CA 94043.
The writing of this note was prompted by two elegant recent results. The first is a paper of Buchbinder, Naor, and Schwartz \[BNS13\] that studies the multiway cut problem, which is a special case of 0-extension. They give a rounding based on exponential clocks. (An identical rounding was earlier, though independently, also given by Ge et al. \[GHYZ11\].) The second is a paper of Miller, Peng, and Xu \[MPX13\], who study low-diameter decompositions and give an algorithm with \(\beta = O(\log n)\) based on exponential clocks. Their algorithm is easily parallelizable, and it substantially improves and cleans up a previous sub-optimal algorithm in the parallel setting due to Blelloch et al. \[BGK+13\].

1.1 Our Results

In this note we give an algorithm for the terminal partitioning problem, which has \(\alpha = O(\log k)\). This immediately gives an \(O(\log k)\) approximation for the 0-extension problem. While this ratio is not optimal, we find the algorithm appealing due to its simplicity: for each terminal \(t \in T\), we pick a random rate \(\rho_t\) from a certain (shifted, truncated exponential) probability distribution. Then for each non-terminal \(v \in V\), we assign it to the terminal

\[
\arg\min_{t \in T} \left\{ \frac{d(x, t)}{\rho_t} \right\}
\]

breaking ties arbitrarily. (This is very similar in spirit to the \[BNS13, GHYZ11\] geometric rounding for multiway cut simplex linear program.)

A side-effect of our algorithm for terminal partitioning is a certain “proximity” condition: it only assigns each vertex to “close-by” terminals. We show that terminal partitionings that satisfy this kind of proximity condition also give us low-diameter decompositions, merely by choosing an \(O(\Delta)\)-net of the metric as the terminal set and then running the terminal partitioning algorithm. This immediately gives a low-diameter decomposition with \(\beta = O(\log n)\), which is best possible. Details appear in Section 4.

A word about the relationship of this note to the work of Miller, Peng, and Xu \[MPX13\]: in their algorithm each vertex \(v \in V\) first picks a random value \(X_v \sim \text{Exp}(\ln n/\Delta)\), and say \(X_{\text{max}} := \max_v X_v\). Their algorithm builds BFS trees at unit rate from a set of terminals, where we start off with the terminal set being empty, and each vertex \(v\) enters the terminal set (and hence starts building its BFS tree) at time \(X_{\text{max}} - X_v\). Each vertex is assigned to the first BFS tree it belongs to. We can think of this as building \textit{additively weighted Voronoi diagrams}. In contrast, we choose a set of terminals that are fixed over time, but our BFS trees grow at random rates — this is more akin to \textit{multiplicatively weighted} Voronoi diagrams. Their algorithm is parallelizable, and also gives strong-diameter decompositions, whereas we only give weak-diameter decompositions. On the other hand, our algorithm is naturally scale-free and hence lends itself more naturally to terminal partitioning and 0-extension, whereas the \[MPX13\] algorithm is scale-based and more natural for low-diameter decompositions.

2 The Terminal Partitioning Problem

Input: given a metric \((V, d)\) and terminals \(T \subseteq V\), where \(n := |V|\) and \(k := |T|\).

Output: a (random) map \(f : V \to T\) such that

(i) (retraction) \(f(t) = t\) for all \(t \in T\),
(ii) (separation) for all \(u, v \in V\), we have

\[
\Pr[f(u) \neq f(v)] \leq \alpha \cdot \frac{d(u, v)}{\min(A_u, A_v)},
\]

where \(A_u := d(u, T)\) is the distance from \(u\) to its closest terminal in \(T\).

\footnote{The random variable \(\rho_t \sim 1 + \text{Exp}(\ln k)\) conditioned on being at most 2; details follow in Section 3.}
such a (random) map $f$ is called a terminal partitioning with stretch $\alpha$. There is an optional property
that will be useful:

(iii) Let $B(x, r) := \{ y \in V \mid d(x, y) \leq r \}$ be the radius-$r$ ball around $x$ in the metric $(V, d)$. For $c > 0$, the map $f$ is $c$-proximate if for all $u \in V$,

$$\Pr[f(u) \in B(u, c \cdot A_u)] = 1.$$ 

Note that if a mapping satisfies the proximity property (iii), it also satisfies the retraction property (i), simply because each terminal $t$ has $A_t = 0$, hence $f(t) \in B(t, 0) \implies f(t) = t$.

An $\alpha$-stretch algorithm for terminal partitioning immediately implies an $\alpha$-approximation for the 0-extension problem (which we do not define here); for details, see the original paper of Calinescu et al. [CKR05].

3 An Algorithm for Terminal Partitioning

We now give the algorithm for terminal partitioning. We first define the truncated exponential distribution. Given parameters $\lambda$ and $\gamma > 0$, the distribution $\text{TExp}(\lambda, \gamma)$ is simply the exponential distribution $\text{Exp}(\lambda)$ conditioned on being at most $\gamma$. Formally it is supported on $[0, \gamma]$ and has density at $x \in [0, \gamma]$ equal to $p(x) = Z(\lambda, \gamma) \cdot \lambda \exp(-\lambda x)$. Here $Z(\lambda, \gamma) = (1 - \exp(-\lambda \gamma))^{-1}$ is a normalization term. Some useful properties of this distribution, which we use in the following analysis, can be found in Section 5.

3.1 The Random-Rates Algorithm

Let $K \geq 3$ be a parameter such that for every vertex $x$, $|T \cap B(x, 2A_x)| \leq K$. Clearly $K \leq \max(3, |T|) = \max(3, k)$.

Algorithm Random-Rates

(a) For each terminal $t$, independently set $\nu_t \sim \text{TExp}(\ln K, 1)$.
(b) For each terminal $t$, set its “rate” $\rho_t \leftarrow 1 + \nu_t$.
(c) Imagine growing “Voronoi” regions at rate $\rho_t$ around each terminal $t$ to capture vertices. Formally, define the retraction $f$ as

$$f(x) = \text{argmin}_{t \in T} \left\{ \frac{d(x, t)}{\rho_t} \right\}$$

(3.2)

We break ties arbitrarily.

The main theorem of this section is the following:

**Theorem 3.1** The random map $f$ defined by Algorithm Random-Rates is a terminal partitioning with stretch $\alpha = O(\log K)$, and is 2-proximate.

The proof appears in the next section. Moreover, the paper [FHT03] shows that for any map that satisfies the 2-proximity condition, the stretch of $O(\log k)$ is best possible. In Section 4 we will see another proof of this optimality.

3.2 Proof of Theorem 3.1

It is easy to see the 2-proximity. Indeed, by definition, each $\rho_t \in [1, 2]$. If $t_x$ is the terminal closest to $x$, then the definition of $f$ ensures that

$$\frac{d(x, f(x))}{2} \leq \frac{d(x, f(x))}{\rho_{f(x)}} \leq \frac{d(x, t_x)}{\rho_{t_x}} \leq d(x, t_x).$$
It follows that \( d(x, f(x)) \leq 2A_u \), which proves the map \( f \) is 2-proximate.

To prove the stretch bound, we will show a stronger padding property. For any \( u \in V \), and any \( r \geq 0 \), we say that the ball \( B(u, r) \) is cut (by the mapping \( f \)) if there exists \( v \in B(u, r) \) such that \( f(u) \neq f(v) \). We say that a terminal \( t \) captures \( u \) if \( f(u) = t \), and that \( t \) cuts \( B(u, r) \) if \( t \) captures \( u \) and \( B(u, r) \) is cut.

**Lemma 3.2**  
For any \( u \in V \) and any radius \( r \leq A_u/4 \),

\[
\Pr[\text{B}(u, r) \text{ is cut}] \leq O(\log K) \cdot \frac{r}{A_u}. \tag{3.3}
\]

**Proof.**  Fix a terminal \( t^* \). We first upper bound \( \Pr[\text{B}(u, r) \text{ is cut by } t^*] \). Note that by the 2-proximity condition, it suffices to consider \( t^* \) such that \( d(u, t^*) \in [A_u, 2A_u] \). Condition on the rates \( \hat{\rho}_t \) for all other terminals \( t \neq t^* \), and define the “critical threshold” for \( x \in V \) to be

\[
\rho^c_t(x) := d(x, t^*) \cdot \arg\max_{v: t \neq t^*} \left\{ \frac{\hat{\rho}_t}{d(x, t)} \right\}.
\tag{3.4}
\]

for all \( x \in V \). Note that if \( \rho^c_t > \rho^c_t(x) \), then \( f(x) = t^* \). We first prove a simple lemma.

**Lemma 3.3**  
Let \( v \in B(u, r) \) for \( r \leq A_u/4 \), and let \( t \) be such that \( d(u, t) \leq 2A_u \). Then

\[
\rho^c_t(v) - \rho^c_t(u) \leq \frac{12r}{A_u}. \tag{3.5}
\]

**Proof.**  First observe that for any \( t' \),

\[
\frac{d(v, t)}{d(v, t')} - \frac{d(u, t)}{d(u, t')} \leq \frac{d(u, t) + r - d(u, t')}{d(u, t')} \leq \frac{(d(u, t) + r)(1 + \frac{2r}{d(u, t')})}{d(u, t')} \leq \frac{r + (d(u, t) + r)(\frac{2r}{A_u})}{A_u} \leq \frac{r + 5\frac{d(u, t)}{4}(\frac{2r}{A_u})}{A_u} \leq \frac{r + 5r}{A_u}.
\]

Thus \( \frac{d(v, t)\hat{\rho}_t}{d(v, t')\hat{\rho}_t} - \frac{d(u, t)\hat{\rho}_t}{d(u, t')\hat{\rho}_t} \leq \frac{12r}{A_u} \). The claim follows by definition of \( \rho^c_t \) and Lipschitz-ness of max.  

The rest of the proof is relatively simple: when the threshold is far from \( \gamma \), the truncation has little effect, and the memorylessness property of the exponential suffices to show that the probability of cutting \( B(u, r) \), conditioned on capturing \( u \) is small for \( t^* \). When the threshold is closer to \( \gamma \), this conditional probability can be large. However, for such large thresholds, the unconditional probability is small enough that we can afford to add these probabilities over the \( K \) terminals. We formalize this next.

Let \( \delta := 12r/A_u \) be the upper bound in (3.5), and let \( \lambda := \ln K \), the parameter for the truncated exponential. It follows that if \( \rho^c_t \geq \rho^c_t(u) + \delta \), then \( t^* \) captures all of \( B(u, r) \). Recall that the definition of \( t^* \) cutting \( B(u, r) \) is that \( t^* \) must capture \( u \) but not all of \( B(u, r) \). Hence,

\[
\Pr[t^* \text{ cuts } B(u, r)] \leq \Pr[\rho^c_t \in [\rho^c_t(u), \rho^c_t(u) + \delta]].
\]
Observe that if \( a \leq 1 - \frac{1}{\lambda} \), then \( e^{-\lambda a} - e^{-\lambda} = e^{-\lambda a}(1 - e^{\lambda(a-1)}) \geq \frac{e^{-\lambda a}}{\lambda} \). Thus if \( \rho_t^* (u) \leq 2 - \frac{1}{\lambda} \), then recall that \( \rho_t^* - 1 \) is a truncated exponential, and use Proposition 5.1(c) to get

\[
\Pr \left[ t^* \text{ cuts } B(u, r) \mid (t^* \text{ captures } u) \land (\rho_t^* (u) \leq 2 - \frac{1}{\lambda}) \right] \\
\leq \Pr \left[ \rho_t^* \leq \rho_t^* (u) + \delta \mid (\rho_t^* \geq \rho_t^* (u)) \land (\rho_t^* (u) \leq 2 - \frac{1}{\lambda}) \right] \\
\leq \delta \lambda \cdot \frac{\exp(-\lambda \rho_t^* (u))}{\exp(-\lambda \rho_t^* (u)) - \exp(-\lambda)} \leq 2\delta\lambda.
\]

On the other hand, if \( \rho_t^* (u) > 2 - \frac{1}{\lambda} \), then by Proposition 5.1(b),

\[
\Pr \left[ t^* \text{ cuts } B(u, r) \mid \rho_t^* (u) > 2 - \frac{1}{\lambda} \right] = \Pr \left[ \rho_t^* \in [\rho_t^* (u), \rho_t^* (u) + \delta] \mid \rho_t^* (u) > 2 - \frac{1}{\lambda} \right] \\
\leq 2\delta\lambda e^{-\lambda(1-1/\lambda)} = 2\delta\lambda e^{1-\lambda} \leq 2e\delta\lambda/K.
\]

It follows that

\[
\Pr[t^* \text{ cuts } B(u, r)] \leq \Pr[t^* \text{ captures } u] \cdot 2\delta\lambda + 2e\delta\lambda/K
\]

Since there are \( K \) possible terminals that can capture \( u \), and exactly one captures \( u \), it follows that

\[
\Pr[B(u, r) \text{ gets cut}] \leq \left( \sum_{t^*} \Pr[t^* \text{ captures } u] \right) \cdot 2\delta\lambda + K \cdot 2e\delta\lambda/K \\
\leq 2(1 + e)\delta\lambda.
\]

Since \( \delta = O(r/A_u) \), and \( \lambda = \ln K \), the claim follows.

Finally, to show that the padding property of Lemma 3.2 implies the separation probability (2.1) is standard: we give it here for completeness. If \( d(u, v) \geq A_u/4 \), then \( O(\log K) \cdot \frac{d(u, v)}{A_u} \geq 1 \) for a large enough constant in the big-Oh, so (2.1) is trivially satisfied. Else, \( v \in B(u, r^*) \) for \( r^* = d(u, v) \leq A_u/4 \), and \( B(u, r^*) \) not being cut implies that \( u, v \) are not separated; by Lemma 3.2 this happens with probability

\[
O(\log K) \cdot \frac{r^*}{A_u} = O(\log K) \cdot \frac{d(u, v)}{A_u} \leq O(\log K) \cdot \frac{d(u, v)}{\min(A_u, A_v)}.
\]

This completes the proof of Theorem 3.3.

### 4 An Algorithm for Low-Diameter Decompositions

We can get an algorithm for low-diameter decompositions (LDDs) using a similar random rates idea. Recall that in the LDD problem, we are given a metric \((V, d)\) and parameter \(\Delta\), we want a random partition \(V_1, V_2, \ldots, V_q\) of the point set \(V\) such that:

(i) The clusters have diameter at most \(\Delta\); i.e., \(\max_{i} \max_{x,y \in V_i} d(x, y) \leq \Delta\), and

(ii) The probability

\[
\Pr[x, y \text{ not in same cluster }] \leq \beta \cdot \frac{d(x, y)}{\Delta}.
\]

(4.6)

Recall that an \(\varepsilon\)-net of a metric \((V, d)\) is a set \(N \subseteq V\) such that (a) for all \(v \in V\), the distance to the nearest net point is at most \(\varepsilon\) (i.e., \(d(v, N) \leq \varepsilon\)), and (b) two net points are \(\varepsilon\) apart (i.e., \(d(t_1, t_2) \geq \varepsilon\) for \(t_1, t_2 \in N\) such that \(t_1 \neq t_2\)). A greedy algorithm gives us such a net; near-linear time algorithms are also known to find nets [HPM03].

Our LDD procedure is the following simple reduction:
Algorithm Random-Rates-LDD: Let $T$ be a $\Delta/10$-net of $(V, d)$. Use a 2-proximate terminal partitioning algorithm to define the clusters in the natural way: the vertices that map to the same terminal in $T$ are in the same cluster.

Lemma 4.1  A 2-proximate terminal partitioning $f$ with stretch $\alpha$ gives us a $\Delta$-LDD with $\beta = O(\alpha)$.

Proof. Consider $x, y$ such that $d(x, y) > \Delta$, we claim that $f(x) \neq f(y)$. Indeed, since we found a $\Delta/10$-net, the closest terminal to each node is at distance at most $\Delta/10$ from it. By the proximity property, each node is assigned to a terminal at distance at most $\Delta/5$ from it, and since $d(x, y) > \Delta$, we must have $f(x) \neq f(y)$ by the triangle inequality. Hence we have the low-diameter property.

Now for the probability of separation for some pair $x, y$. For $x, y$ which are “far apart”, say, $d(x, y) > \Delta/100$, the probability that $x, y$ are separated is trivially at most 1, which is at most $100 \cdot d(x, y)/\Delta$, so $\beta \geq 100$ suffices for them.

So assume $d(x, y) \leq \Delta/100$. Let $t_x, t_y$ be the closest terminals to $x, y$ respectively, and so $A_x = d(x, t_x)$ and $A_y = d(y, t_y)$. There are two cases:

- Both $A_x, A_y \geq \Delta/100$. Then by (3.2), we have the probability of $x, y$ separated (or equivalently $f(x) \neq f(y)$) is at most
  \[
  \alpha \cdot \frac{d(x, y)}{\min(A_x, A_y)} \leq 100 \alpha \cdot \frac{d(x, y)}{\Delta}.
  \]

- At least one of $A_x, A_y \leq \Delta/100$, say $A_x \leq A_y$. Then $A_y \leq A_x + d(x, y) \leq \Delta/100 + \Delta/100 = \Delta/50$. Since we also have $d(t_x, t_y) \leq d(t_x, x) + d(x, y) + d(y, t_y) = A_x + A_y + d(x, y) \leq \Delta/25$. By the packing property of a $\Delta/10$-net, we know that if $t_x \neq t_y$ then $d(t_x, t_y) \geq \Delta/10$, which implies that $t_x = t_y$.

Moreover, consider any other terminal $t$ within $B(x, 2A_x) \cup B(y, 2A_y)$, then $d(t_x, t) \leq 3A_x$ or $d(t_x, t) \leq A_x + d(x, y) + 2A_y$. In either case, this would mean $d(t_x, t) \leq 6\Delta/100$, and hence again $t_x = t$. In other words, the only terminal within distance $2A_x$ of $x$ (and within $2A_y$ of $y$) is $t_x = t_y$.

Now by the proximity condition, $f(x) = f(y)$ with probability 1.

This shows that the LDD procedure above satisfies $\beta \leq 100\alpha$. \hfill $\blacksquare$

Since the size of the net is at most $n$, this implies $\beta = O(\log n)$. Moreover, recall that a metric has doubling dimension $\dim$ if for all $u \in V$ and $r \geq 0$, any set of diameter $2r$ can be covered by $2^{\dim}$ sets of diameter at most $r$. It is a standard fact that for metrics of doubling dimension $\dim$, any net $T$ has the property that for every $u \in V$, $|B(u, 2A_u) \cap T| \leq 2^{O(\dim)}$. Thus $K$ is $2^{O(\dim)}$, and we get an LDD with parameter $\beta = O(\dim)$, matching known results $[3KL03]$. We summarize these results below.

Corollary 4.2  Algorithm Random-Rates-LDD, using the random map $f$ from Section 3 has parameter $\beta = O(\log n)$. Moreover, for metrics of constant doubling dimension, the parameter $\beta = O(1)$.

It is known that for LDDs on general metrics, $\beta = \Omega(\log n)$ is best possible, e.g., for large girth expanders (see, e.g., $[Bar96]$). The above reduction gives another proof that for $O(1)$-proximate terminal partitionings, we cannot achieve $\alpha = o(\log k)$.

5  Properties of the Truncated Exponential Distribution

Here are some properties of the truncated exponential that were useful in our analysis.

Proposition 5.1  Let $\nu \sim T \exp(\lambda, \gamma)$, and $a, b > 0$ be such that $(a + b) \leq \gamma$. Suppose $\gamma > 1/\lambda$. Then

(a) $Z(\lambda, \gamma) = (1 - \exp(-\lambda\gamma))^{-1} \leq 2$.  

(b) \( \Pr[\nu \in (a, a+b)] \leq 2 \exp(-\lambda(1 - \exp(-b\lambda)) \leq 2b\lambda \exp(-\lambda a) \).

(c) \( \Pr[\nu \leq (a+b) \mid \nu \geq a] = \frac{\exp(-\lambda a) - \exp(-\lambda (a+b))}{\exp(-\lambda a) - \exp(-\lambda b)} \leq b\lambda \cdot \frac{\exp(-\lambda a) - \exp(-\lambda b)}{\exp(-\lambda a) - \exp(-\lambda b)} \).

**Proof.** Part (a) follows from \( \gamma \lambda > 1 \) and hence \( Z(\lambda, \gamma) = (1 - \exp(-\lambda \gamma))^{-1} \leq (1 - \exp(1))^{-1} = e \leq 2 \).

For part (b), we have

\[
\Pr[\nu \in (a, a+b)] = Z(\lambda, \gamma) \cdot \lambda \cdot \int_{x=a}^{a+b} e^{-\lambda x} dx
\]

\[
= Z(\lambda, \gamma) \cdot (e^{-\lambda a} - e^{-\lambda(a+b)})
\]

\[
\leq 2e^{-\lambda a} (1 - e^{-\lambda b}) \leq 2b\lambda e^{-\lambda a}.
\]

The last step uses part (a), and that \( 1 + y \leq e^y \) for all \( y \in \mathbb{R} \). For part (c), we use similar calculations.

**Acknowledgments**

This work was done when A. Gupta was visiting Microsoft Research SVC in 2006; he thanks them for their hospitality. We also thank T.-H. Hubert Chan and Satish Rao for useful discussions.

**References**


