Optimal Euclidean spanners: really short, thin and lanky

Michael Elkin and Shay Solomon
Department of Computer Science
Ben-Gurion University of the Negev
P.O. Box 653, Beer-Sheva 84105, ISRAEL
E-mail: {elkinm, shayso}@cs.bgu.ac.il
Technical Report CS-12-04

April 4, 2012

Optimal Euclidean spanners: really short, thin and lanky

Michael Elkin * Shay Solomon *†

Abstract

The degree, the (hop-)diameter, and the weight are the most basic and well-studied parameters of geometric spanners. In a seminal STOC'95 paper, titled "Euclidean spanners: short, thin and lanky", Arya et al. [4] devised a construction of Euclidean $(1+\epsilon)$-spanners that achieves constant degree, diameter $O(\log n)$, and weight $O(\log^2 n) \cdot \omega(MST)$. This construction applies to n-point constant-dimensional Euclidean spaces. Moreover, Arya et al. conjectured that the weight bound can be improved by a logarithmic factor, without increasing the degree and the diameter of the spanner.

This conjecture of Arya et al. became one of the most central and fundamental open problems in the area of Euclidean spanners. Nevertheless, the only progress since 1995 towards its resolution was reported in the lower bounds front: Any spanner with diameter $O(\log n)$ must incur weight $\Omega(\log n) \cdot \omega(MST)$, and this lower bound holds regardless of the stretch or the degree of the spanner (see the SODA'05 paper of Aggarwal et al. [2], and the FOCS'08 paper of Dinitz et al. [21]).

In this paper we resolve the long-standing conjecture of Arya et al. in the affirmative. Specifically, we present a construction of spanners with the same stretch, degree and diameter, as in Arya et al.'s result, but with *optimal weight* $O(\log n) \cdot \omega(MST)$. So our spanners are as thin and lanky as those of Arya et al., but they are *really* short!

Moreover, our result is more general in two ways. First, we demonstrate that the conjecture holds true not only in constant-dimensional Euclidean spaces, but also in *metrics of constant doubling dimension* (namely, doubling metrics). Second, we provide a general tradeoff between the three involved parameters, which is *tight in the entire range*. Specifically, we prove that for any n-point doubling metric, any $\epsilon > 0$, and any parameter $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner with degree $O(\rho)$, diameter $O(\log_\rho n)$, and weight $O(\rho \log_\rho n) \cdot \omega(MST)$.

To prove this result, we develop a general technique for transforming any given (possibly heavy-weight) spanner H into a *light-weight spanner* H' that has (essentially) the same number of edges, stretch, degree and diameter as those of H. Our technique is not restricted to Euclidean or doubling metrics, but rather applies to general metrics.

*Department of Computer Science, Ben-Gurion University of the Negev, POB 653, Beer-Sheva 84105, Israel.
E-mail: {elkinm,shayso}@cs.bgu.ac.il
Both authors are supported by the BSF grant No. 2008430 and by the ISF grant No. 87209011. In addition, both authors are partially supported by the Lynn and William Frankel Center for Computer Sciences.
†This research has been supported by the Clore Fellowship grant No. 81265410.

1 Introduction

1.1 Euclidean Metrics. Consider a set P of n points in $\mathbb{R}^d$, $d \geq 2$, and a real number $t \geq 1$. A graph $G = (P, E, \omega)$ in which the weight $\omega(p,q)$ of each edge $e = (p,q) \in E$ is equal to the Euclidean distance $\|p - q\|$ between p and q is called a *Euclidean graph.* We say that the Euclidean graph G is a *t-spanner* for P if for every pair $p, q \in P$ of distinct points, there exists a path $\Pi(p,q)$ in G between p and q whose weight (i.e., the sum of all edge weights in it) is at most $t \cdot \|p - q\|$. The parameter t is called the *stretch* of the spanner. The path $\Pi(p,q)$ is said to be a *t-spanner path* between p and q. In this paper we focus on the regime $t = 1 + \epsilon$, for $\epsilon > 0$ being an arbitrarily small constant. We will also concentrate on spanners with $|E| = O(n)$, or nearly $O(n)$, edges. Euclidean spanners were introduced by Chew [13] in 1986. The first constructions of $(1+\epsilon)$-spanners with $O(n)$ edges were devised soon afterwards [14, 33], and the running time of such constructions was improved to $O(n \log n)$ a few years later [46, 41].

Euclidean spanners turned out to be a fundamental geometric construct, with numerous applications. In particular, they were found useful in geometric approximation algorithms [39, 27, 28], geometric distance oracles [27, 29, 28] and Network Design [32, 37]. Various properties of Euclidean spanners are a subject of intensive ongoing research effort [34, 11, 3, 18, 19, 4, 20, 39, 26, 2, 9, 21]. See also the book by Narasimhan and Smid [38], and the references therein. This book is titled "Geometric Spanner Networks", and it is devoted almost exclusively to Euclidean spanners and their various applications.

In addition to stretch ($t = 1+\epsilon$) and sparsity ($|E| = O(n)$), other fundamental properties of Euclidean spanners include their *maximum degree*, their *(hop-)diameter*, and their *lightness.* The *(maximum) degree* $\Delta(G)$ of a spanner G is the maximum degree of a vertex in G. The *diameter* $\Lambda(G)$ of a $(1+\epsilon)$-spanner G is the smallest number Λ such that for every pair of points $p, q \in P$ there exists a $(1+\epsilon)$-spanner path between p and q in G that consists of at most Λ edges (or *hops*). The *lightness* $\Psi(G)$ of a spanner G is defined as the ratio between the *weight* $\omega(G) = \sum_{e \in E} \omega(e)$ of G and the weight $\omega(MST(P))$ of the minimum spanning tree $MST(P)$ for the point set P.

Arya and Smid [6] devised a construction of $(1+\epsilon)$-spanners with constant degree. (In this section we may write "spanner" as a shortcut for a "$(1+\epsilon)$-spanner with $O(n)$ edges".) In FOCS'94, Arya, Mount and Smid [5] devised a construction of spanners with logarithmic diameter. The diameter was improved to $O(\alpha(n))$, where $\alpha(n)$ is the inverse-Ackermann function, by Arya et al. [4] in STOC'95. (Further work on the tradeoff between the diameter and number of edges in spanners can be found in [9, 38, 43].)

Also, in the beginning of the nineties researchers started to systematically investigate spanners that *combine* several additional parameters (among degree, diameter and lightness). Das and Narasimhan [19] devised a construction of spanners with constant degree and lightness. Arya et al. [4] devised a construction of spanners with logarithmic diameter and logarithmic lightness. (This combination was shown to be optimal by Dinitz et al. [21] in FOCS'08; see also [36, 2] for previous lower bounds on this problem.) This construction of [4] may have, however, an arbitrarily large degree. On the other hand, Arya et al. [4] devised also a construction of spanners with constant degree, logarithmic diameter and lightness $O(\log^2 n)$. In the end of their seminal work Arya et al. [4] conjectured that one can obtain a spanner with constant degree, logarithmic diameter and logarithmic lightness. Specifically, they wrote:

Conjecture 1 ([4]) *For any $t > 1$, and any dimension k, there is a t-spanner, constructible in $O(n \log n)$ time, with bounded degree, $O(\log n)$ diameter, and weight $O(\omega(MST) \log n)$.*

In this paper we prove the conjecture of Arya et al. [4], and devise a construction of $(1+\epsilon)$-spanners with bounded degree, and with logarithmic diameter and lightness. The running time of our construction is $O(n \log n)$, which is optimal in the algebraic computation-tree model [12]. (We remark that regardless of the running time, prior to our work it was unknown whether $(1+\epsilon)$-spanners with constant degree, and logarithmic diameter and lightness, exist.)

In fact, our result is far more general than this. Specifically, we provide a tradeoff parameterized by a degree parameter $\rho \geq 2$, and show that there exist $(1+\epsilon)$-spanners with $O(n)$ edges, degree $O(\rho)$, diameter $O(\log_\rho n + \alpha(\rho))$ and lightness $O(\rho \cdot \log_\rho n)$. Due to lower bounds by [9, 21], this tradeoff is *optimal in the entire range* of the parameter ρ. The running time of our construction is $O(n \log n)$.

1.2 Doubling Metrics. Our result extends in another direction as well. Specifically, it applies to any *doubling metric*.[1] Doubling metrics, implicit in the works of Assoud [7] and Clarkson [15], were explicitly defined by Gupta et al. [30]. They were subject of intensive research since then [35, 45, 31, 9, 1, 8].

Spanners for doubling metrics were also intensively studied [22, 10, 31, 40, 24, 25, 42]. They were also found useful for Approximation Algorithms [8], and for Machine Learning [23]. In SODA'05 Chan et al. [10] showed that for any doubling metric there exists a $(1+\epsilon)$-spanner with constant degree. In SODA'06, Chan and Gupta [9] devised a construction of $(1+\epsilon)$-spanners with $O(n)$ edges and diameter $O(\alpha(n))$. Smid [42] showed that in doubling metrics a greedy construction produces spanners with logarithmic lightness. Gottlieb et al. [23] devised a construction of $(1+\epsilon)$-spanners with constant degree and logarithmic diameter, with $O(n \log n)$ running time. To the best of our knowledge, there is no known construction of $(1+\epsilon)$-spanners with $O(n)$ edges for doubling metrics that provides logarithmic diameter and lightness simultaneously.[2]

In this paper we devise a construction of $(1+\epsilon)$-spanners for doubling metrics with constant degree, logarithmic diameter and logarithmic lightness. Furthermore, for any parameter $\rho \geq 2$, we devise a construction of $(1+\epsilon)$-spanners with degree $O(\rho)$, diameter $O(\log_\rho n)$ and lightness $O(\rho \cdot \log_\rho n)$. The running time of our construction is $O(n \log n)$. This matches exactly our results for Euclidean metrics.

1.3 Our and Previous Techniques. Our starting point is the paper of Chandra et al. [11] from SoCG'92. In this paper the authors devised a general transformation: given a "black-box" construction of spanners with certain stretch and number of edges their transformation returns a construction with roughly the same stretch and number of edges, but with a logarithmic lightness. The drawback of their transformation is that it blows up the degree and the diameter of the original spanner.

In this paper we devise a much more refined transformation. Our transformation enjoys all the useful properties of the transformation of [11], but, in addition, it preserves (up to constant factors) the degree and the diameter of the original "black-box" construction. We then compose our refined transformation on top of known constructions of spanners with constant degree and logarithmic diameter (due to Arya et al. [4] in the Euclidean case, and due to Gottlieb et al. [23] in the case of doubling metrics). As a result we obtain a construction of spanners with constant degree, logarithmic diameter and logarithmic lightness. The latter proves the conjecture of Arya et al. [4].

We remark that our transformation can be applied not only for Euclidean or doubling metrics, but in much more general scenarios. In fact, we have already obtained some improved results for spanners on general graphs that are based on a variant of this transformation. These results are, however, outside the scope of the current paper.

Next, we provide a (short) schematic overview of the two transformations (the one due to [11], and our refined one). The transformation of [11] starts with constructing an MST T of the input metric. Then it constructs the preorder traversal path $\mathcal{L}$ of T. The path $\mathcal{L}$ is then partitioned into (roughly) n intervals of length $\frac{|\mathcal{L}|}{n}$ each. This is the bottom-most level $\mathcal{F}_1$ of the hierarchy $\mathcal{F}$ of intervals that the transformation constructs. Pairs of consecutive intervals are grouped together; this gives rise to $n/2$ intervals of length $2 \cdot \frac{|\mathcal{L}|}{n}$ each. The hierarchy $\mathcal{F}$ consists of $\ell = \log n$ levels, with just one single interval of length $|\mathcal{L}|$ on the last level $\mathcal{F}_\ell$.

On each level $j \in [\ell]$ of the hierarchy each non-empty interval is represented by a point of the original metric (henceforth, its *representative*). Let Q_j denote the set of j-level representatives. The transformation then invokes its input black-box construction of spanners on each point set Q_j separately. Each of those ℓ auxiliary spanners is then pruned, i.e., "long" edges are removed from it. The remaining edges in all the auxiliary spanners, together with the MST T, form the output spanner.

Intuitively, the pruning step ensures that the resulting spanner is reasonably light. The stretch remains

[1]The *doubling dimension* of a metric is the smallest value d such that every ball B in the metric can be covered by at most 2^d balls of half the radius of B. A metric is called *doubling* if its doubling dimension is constant.

[2]On the other hand, as was mentioned in Section 1.1, for Euclidean metrics such a construction was devised by Arya et al. [4]. However, the degree in the latter construction is unbounded.

roughly intact, because each distance is taken care "on its own scale". The number of edges does not grow by much, because the sequence $|Q_1|, |Q_2|, \ldots, |Q_\ell|$ decays geometrically. However, the diameter is blown up, because within each interval the MST-paths (which may contain many edges) are used to reach points that do not serve as representatives. Also, the degree is blown up because the same point may serve as a representative on many different levels.

Our first idea is to use a construction of 1-dimensional spanners to shortcut the traversal path $\mathcal{L}$. We remark that $(1+\epsilon)$-spanners with $O(n)$ edges, constant degree, logarithmic diameter and logarithmic lightness for sets of n points on a line (1-dimensional case) were devised already by Arya et al. [4]. Plugging[3] this 1-dimensional spanner construction into the transformation of Chandra et al. [11] gives rise to an improved transformation that keeps the diameter in check, but still blows up the degree.

Our second idea is to distribute the degree load evenly between "nearby" points along $\mathcal{L}$. Unfortunately, this turns out to be impossible, at least if one sticks with the original hierarchy $\mathcal{F}$ of partitions of $\mathcal{L}$ into intervals. The problem is that the same point may well be the only eligible representative for many different levels of the hierarchy. Overcoming this hurdle is the heart of our paper. Instead of intervals we divide the point set into a different hierarchy $\hat{\mathcal{F}}$ of sets, which we call *bags*. On the lowest level of the hierarchy the bags and the intervals coincide. This, however, soon changes. As our algorithm proceeds it carefully moves points between bags so as to guarantee that no point will ever be too loaded. At the same time, we try not to move points too far from their initial position on $\mathcal{L}$. Indeed, if remote points (with respect to their initial position on $\mathcal{L}$) end up in the same bag, then the auxiliary spanners for the sets of representatives, as well as the 1-dimensional spanner for $\mathcal{L}$, cease being useful for providing short (in terms of both weighted distance and hop-distance) $(1+\epsilon)$-spanner paths for the original point set. On the other hand, degree constraints may force our algorithm to relocate points arbitrarily far away from their initial position on $\mathcal{L}$. Coping with this delicate issue is a major challenge.

1.4 Related Work. Most of the related work was already discussed above. One more relevant result is the ESA'10 paper [44] by the authors of the current paper. There we devised a construction of spanners that trades between the degree, diameter and lightness. That construction, however, could only match the previous suboptimal bounds of Arya et al. [4], but not improve them. In particular, the lightness of the construction of [44] is $\Omega(\log^2 n)$, regardless of the other parameters.

1.5 Structure of the Paper. In Section 2 we describe our algorithm. We analyze it in Section 3. Appendix C contains the description of two subroutines of our algorithm. Appendix D is devoted to a few parts of the analysis that we could not provide in the main part of the paper due to lack of space. The bibliography appears at the very end of the submission.

1.6 Preliminaries. We will use the following two results as a black-box.

Theorem 1.1 ([4, 44]) *For any n-point 1-dimensional space M and any integer $\rho \geq 2$, there exists a 1-spanner H with $|H| = O(n)$, $\Delta(H) = O(\rho)$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$ and $\Psi = O(\rho \cdot \log_\rho n)$. The running time of this construction is $O(n)$.*

Theorem 1.2 ([4, 25, 44]) *For any n-point doubling metric $M = (P, \delta)$, any $\epsilon > 0$ and any integer $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner H with $|H| = O(n)$, $\Delta(H) = O(\rho)$ and $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$. (Note that there is no guarantee on the lightness.) The running time of this construction is $O(n \log n)$.*

For the sake of completeness we provide a proof of Theorem 1.2 in Appendix B.

Our transformation theorem is formulated below.

Theorem 1.3 *Let $M = (P, \delta)$ be an arbitrary metric. Let $\rho \geq 2$ be an arbitrary integer parameter, and $t \geq 1$ be an additional parameter. Suppose that for any subset $Q \subseteq P$, $|Q| = n$, there exists an algorithm*

[3] In fact, we use our own more recent construction [44] of 1-spanners for 1-dimensional spaces with the above properties. Having stretch 1 instead of $(1+\epsilon)$ simplifies the analysis.

(henceforth, Algorithm BasicSp) which builds a t-spanner H for the sub-metric $M[Q]$ of M induced by the point set Q, so that $|H| \le SpSz(n)$, $\Delta(H) \le \Delta(n)$, $\Lambda(H) \le \Lambda(n)$. Moreover, Algorithm BasicSp requires at most $SpTm(n)$ time. Suppose also that all the functions $SpSz(n), \Delta(n), \Lambda(n)$ and $SpTm(n)$ are monotone non-decreasing, while the functions $SpSz(n)$ and $SpTm(n)$ are also convex.

Then there is an algorithm (henceforth, Algorithm LightSp) which builds, for every subset $Q \subseteq P$, $|Q| = n$, and any $\epsilon > 0$, a $(t+\epsilon)$-spanner H' for $M[Q]$ with $|H'| = O(SpSz(n) \cdot \log_\rho(t/\epsilon))$, $\Delta(H') = O(\Delta(n) \cdot \log_\rho(t/\epsilon) + \rho)$, $\Lambda(H') = O(\Lambda(n) + \log_\rho n + \alpha(\rho))$, $\Psi(H') = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot (t^3/\epsilon))$. The running time of Algorithm LightSp is $O(SpTm(n) \cdot \log_\rho(t/\epsilon) + SpSz(n) \cdot \log n)$.

Given this theorem we derive our main result by instantiating the algorithm from Theorem 1.2 as Algorithm *BasicSp* in Theorem 1.3. As a result we obtain a construction of $(1+\epsilon)$-spanner H for doubling metrics with $|H| = O(n)$, $\Delta(H) = O(\rho)$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$, $\Psi(H) = O(\rho \cdot \log_\rho n)$, in time $O(n \log n)$. (We substituted $t = 1+\epsilon$, and $\epsilon > 0$ is a constant.) See Appendix A for the more general statement of our result, which applies to general (not necessarily constant) ϵ and doubling dimension d.

In the sequel we will use a simple graph procedure, which we call Procedure *Attach*. It accepts as input an n-vertex graph $G = (V, E)$, whose vertices are labeled by either *safe* or *risky*. The procedure returns a *star forest*, i.e., a collection of vertex disjoint stars Γ, that satisfies the following two conditions.
(1) $\bigcup_{S\in\Gamma}^{h} V(S)$ contains the set $R \subseteq V$ of vertices which are not isolated in G, and labeled as risky.
(2) Each star $S \in \Gamma$ contains a *center* $s \in V$, which is labeled as either safe or risky, and one or more leaves $z_1, \ldots, z_k \in V$ labeled as risky. The edge set $E(S)$ of a star S is given by $E(S) = \{(z_i, s) \mid i \in [k]\}$.

See Appendix C.1 for the implementation and analysis of the Procedure *Attach*.

For a pair of non-negative integers $i, j, i \le j$, we denote $[i, j] = \{i, i+1, \ldots, j\}$, $[i] = \{1, 2, \ldots, i\}$.

2 Algorithm *LightSp*

Let $M = (P, \delta)$ be an arbitrary metric, and let $Q \subseteq P$ be a subset of n points from P.

Algorithm *LightSp* starts with computing an MST, or an approximate MST, T, for the metric $M[Q]$. In Euclidean and doubling metrics an $O(1)$-approximate MST can be computed in $O(n \log n)$ time. This can be done by running Prim's MST Algorithm over any $O(1)$-spanner of the metric with $O(n)$ edges, such as the spanner of Theorem 1.2. The case of more general metrics is addressed in Appendix D.2.

Let $\mathcal{L}$ be the Hamiltonian path of $M[Q]$ obtained by taking the preorder traversal of T. Define $L = \omega(\mathcal{L})$; it is well known ([17], ch. 36) that $L \le 2 \cdot \omega(T)$, and so $L = O(\omega(MST(M[Q])))$. Write $\mathcal{L} = (q_1, q_2, \ldots, q_n)$, and let $M_{\mathcal{L}} = (Q, \delta_{\mathcal{L}})$ be the 1-dimensional space induced by the path $\mathcal{L}$, where $\delta_{\mathcal{L}}$ is the distance in $\mathcal{L}$ (henceforth, *path distance*), i.e., $\delta_{\mathcal{L}}(v_k, v_{k'}) = \sum_{i=k}^{k'-1} \delta(v_i, v_{i+1})$, for every pair k, k' of indices, $1 \le k < k' \le n$. We employ Theorem 1.1 to build in $O(n)$ time a 1-spanner $\tilde{H}$ for $M_{\mathcal{L}}$ with $|\tilde{H}| = O(n)$, $\Delta(\tilde{H}) = O(\rho)$, $\Lambda(\tilde{H}) = O(\log_\rho n + \alpha(\rho))$ and $\Psi(\tilde{H}) = O(\rho \cdot \log_\rho n)$. Let $H = (Q, E_H)$ be the graph obtained from $\tilde{H}$ by assigning weight $\delta(p, q)$ to each edge $(p, q) \in \tilde{H}$. Since edge weights in H are no greater than the corresponding edge weights in $\tilde{H}$, we have (i) $\omega(H) \le \omega(\tilde{H}) = O(\rho \cdot \log_\rho n) \cdot L$, and (ii) for any pair $p, q \in Q$ of points, there is a path $\Pi_H(p, q)$ in H that has weight at most $\delta_{\mathcal{L}}(p, q)$ and $O(\log_\rho n + \alpha(\rho))$ edges. We henceforth call H the *path-spanner*. We also define an order relation $\prec_{\mathcal{L}}$ for the points of Q. Specifically, $q_i \prec_{\mathcal{L}} q_j$ (respectively, $q_i \preceq_{\mathcal{L}} q_j$) if and only if $i < j$ (resp., $i \le j$).

Let $\ell = \lceil \log_\rho n \rceil$. Define $Q_0 = Q$, let $n_0 = |Q_0| = n$, and define the *0-level threshold* $\tau_0 = 2 \cdot \frac{L}{n} \cdot t \cdot (1 + \frac{1}{c})$, where $c = \lceil \frac{4\cdot(t+1)}{\epsilon} \rceil = \Theta(t/\epsilon)$ is a constant that depends on t and ϵ. For $j \in [\ell]$, we define $\xi_j = \rho^{j-1} \cdot \frac{L}{n}$. Divide the path $\mathcal{L}$ into $n_j = \frac{c \cdot L}{\xi_j} = \frac{c \cdot n}{\rho^{j-1}}$ intervals of length $\mu_j = \frac{\xi_j}{c}$ each. Define also the *j-level threshold* $\tau_j = 2\mu_j \cdot \rho \cdot t \cdot (c+1)$. These intervals induce a partition of the point set Q in the obvious way; denote these intervals and the corresponding point sets by $I_j^{(1)}, I_j^{(2)}, \ldots, I_j^{(n_j)}$ and $Q_j^{(1)}, Q_j^{(2)}, \ldots, Q_j^{(n_j)}$, respectively.

We define $\mathcal{I}_j = \{I_j^{(1)}, \ldots, I_j^{(n_j)}\}$, and $\mathcal{I} = \bigcup_{j=1}^{\ell} \mathcal{I}_j$. Note that, for each $j \in [2, \ell]$, every j-level interval I is a union of ρ consecutive $(j-1)$-level intervals. The interval I is called the *parent* of these $(j-1)$-level intervals, and they are called its *children*. This nested hierarchy of intervals defines in a natural way a forest $\mathcal{F}$ of ρ-ary trees, whose vertices (henceforth, *bags*) are intervals from $\mathcal{I}$. Each of these trees is rooted at an ℓ-level interval. Thus, the number of trees in $\mathcal{F}$ is equal to the number $|\mathcal{I}_\ell| = n_\ell$ of ℓ-level intervals. Specifically, $n_\ell = \frac{c \cdot n}{\rho^{\ell-1}} = \frac{c \cdot n}{\rho^{\log_\rho n - 1}} = c \cdot \rho$. Denote the interval that corresponds to a bag v of $\mathcal{F}$ by $I(v)$, and denote the point set of $I(v)$ by $\bar{Q}(v)$. We call the point set $\bar{Q}(v)$ the *native point set* of v. For an inner bag v in $\mathcal{F}$ with ρ children $c_1(v), \ldots, c_\rho(v)$, we have $I(v) = \bigcup_{i=1}^{\rho} I(c_i(v))$, and $\bar{Q}(v) = \bigcup_{i=1}^{\rho} \bar{Q}(c_i(v))$. Denote by $\mathcal{F}_j$ the set of j-level bags in $\mathcal{F}$, for each $j \in [\ell]$; note that $\bigcup_{v \in \mathcal{F}_j} I(v) = [q_1, q_n]$, and $\bigcup_{v \in \mathcal{F}_j} \bar{Q}(v) = Q$. Also, for any pair of distinct bags $u, v \in \mathcal{F}_j$, $I(u) \cap I(v) = \bar{Q}(u) \cap \bar{Q}(v) = \emptyset$.

In Algorithm *LightSp* we (implicitly) maintain another forest $\mathcal{F}'$ over the same bag set $\mathcal{I}$. Specifically, a j-level bag v, for some index $j \in [\ell - 1]$, may get attached by the algorithm to some $(j+1)$-level bag u, other than the parent $\pi(v)$ of v in $\mathcal{F}$. If this happens we say that u becomes a *step-parent* of v in $\mathcal{F}$ (and u is a parent of v in $\mathcal{F}'$), and v becomes a *step-child* of u in $\mathcal{F}$ (and v is a child of u in $\mathcal{F}'$).

2.1 Point Sets

In addition to the native point set $\bar{Q}(v)$, the algorithm will also maintain three more point sets: the *base point set* $B(v)$, the *kernel set* $K(v)$, and the *point set* $Q(v)$. These sets will satisfy $B(v) \subseteq K(v) \subseteq Q(v)$. It will also hold that $B(v) \subseteq \bar{Q}(v)$. A bag v is called empty if $Q(v) = \emptyset$.

Algorithm *LightSp* processes the forest $\mathcal{F}$ bottom-up. In other words, it starts with processing bags of $\mathcal{F}_1$, then it proceeds to processing bags of $\mathcal{F}_2$, and so on. At the last iteration the algorithm processes bags of $\mathcal{F}_\ell$. We refer to the processing of bags of $\mathcal{F}_j$ as *j-level processing*, for each index $j \in [\ell]$. The algorithm maintains the point sets $B(v), K(v)$ and $Q(v)$ of all bags $v \in \mathcal{F}_j$ during the j-level processing in the following way. For a bag $v \in \mathcal{F}_1$, we set $B(v) = K(v) = Q(v) = \bar{Q}(v)$.

A non-empty $(j-1)$-level bag z, $j \in [2, \ell]$ may get attached by the algorithm to some j-level bag v, other than the parent $\pi(z)$ of z in $\mathcal{F}$. If this happens, we say that z *is disintegrated from* $\pi(z)$, and also that z *is integrated into* v. Denote by $\mathcal{J}(v)$ the bags z that were integrated into the bag v. They will be referred to as the *joining step-children* (or shortly, *step-children*) of v. Denote also by $\mathcal{S}(v)$ the set of *surviving children* of v, i.e., the non-empty bags z with $v = \pi(z)$ that were not integrated into some other j-level bag v', $v' \neq v$ ($v' \in \mathcal{F}_j$). Let $\chi(v) = \mathcal{S}(v) \cup \mathcal{J}(v)$ be the set of *extended children* of v. Observe that $\chi(v) \subseteq \mathcal{F}_{j-1}$, and that all bags in $\chi(v)$ are non-empty.

The *base point set* $B(v)$ (respectively, *point set* $Q(v)$) of a bag $v \in \mathcal{F}_j, j \in [2, \ell]$ is defined as the union of the base point sets (resp., point sets) of its surviving (resp., extended) children, i.e., $B(v) = \bigcup_{z \in \mathcal{S}(v)} B(z)$, $Q(v) = \bigcup_{z \in \chi(v)} Q(z)$. The *kernel set* $K(v)$ of v is an intermediate set, in the sense that $B(v) \subseteq K(v) \subseteq Q(v)$. We will soon specify which of the points of $Q(v) \setminus B(v)$ are included into $K(v)$. Intuitively, all points of $K(v)$ will always be "pretty close" to the base point set $B(v)$, both in terms of the metric distance in M, and in terms of the hop-distance. They will be used to alleviate the degree load from the points of $B(v)$.

The algorithm will assign to every bag v a representative point $r(v)$. As was discussed in the introduction, if one selects representatives only from the native point set $\bar{Q}(v)$, then large maximum degree of the resulting spanner may be *inevitable*, regardless of the specific way in which representatives are selected. This may happen, for example, if there is a point p which is far away in the path metric $M_{\mathcal{L}}$ from any other point of M, but close to many points of M in the original metric. This point may be the only point in the point set of some bag $v = v^{(0)}$, as well as in the point sets of many of its ancestors $v^{(1)} = \pi(v), v^{(2)} = \pi(\pi(v)), \ldots$ in $\mathcal{F}$. In this case p will necessarily serve as a representative of all these bags, and will accumulate a large degree. Instead, we will pick $r(v)$ from the kernel set $K(v)$.

The kernel set $K(v)$ of a bag $v \in \mathcal{F}_j, j \in [2, \ell]$ is defined as follows. The *surviving point set* $Q'(v)$ and *surviving kernel set* $K'(v)$ are given by $Q'(v) = \bigcup_{z \in \mathcal{S}(v)} Q(z)$. and $K'(v) = \bigcup_{z \in \mathcal{S}(v)} K(z)$, respectively. If $|Q'(v)| \geq \ell$ then the kernel set of v is set to be equal to its surviving kernel, i.e., $K(v) = K'(v)$. Otherwise (if $|Q'(v)| < \ell$), we set $K(v) = K'(v) \cup \bigcup_{z \in \mathcal{J}(v)} K(z) = \bigcup_{z \in \chi(v)} K(z)$.

The intuition behind increasing the kernel set $K(v)$ beyond its surviving kernel $K'(v)$ (i.e., setting $K(v) = K'(v) \cup \bigcup_{z \in \mathcal{J}(v)} K(z)$) in the case that $|Q'(v)| < \ell$ is that in this case the surviving kernel is "too small". Hence one needs to insert into it more points to distribute the degree load.

In the complementary case ($|Q'(v)| \geq \ell$), the surviving kernel set $K'(v)$ also satisfies $|K'(v)| \geq \ell$. Thus we can distribute the load of the $O(\ell)$ auxiliary spanners that Algorithm *LightSp* constructs among the points of $K'(v)(= K(v))$ in such a way that no kernel point is overloaded.

We say that a bag v is *small* if $|Q(v)| < \ell$, and that it is *large* otherwise.

Observation 2.1 *Fix an arbitrary index $j \in [\ell]$, and let v be a j-level bag. (1) If v is small, then $K(v) = Q(v)$. (2) If v is large, then $|K(v)| \geq \ell$.*

Note that for every index $j \in [\ell]$, $Q = \bigcup_{v \in \mathcal{F}_j} Q(v)$. Also, for any pair u, v of distinct j-level bags, $Q(u) \cap Q(v) = \emptyset$. Moreover, $Q(v) = \emptyset$ iff $B(v) = \emptyset$. These assertions can be readily verified.

The algorithm will also maintain a set of edges $\mathcal{B}$, which we call the *base edge set* of the spanner.

For each non-empty bag $v \in \mathcal{F}$, the base edge set $\mathcal{B}$ will connect the base point set $B(v)$ of v via a simple path $P(v)$. That is, if we denote the points of $B(v)$ from left to right (w.r.t. the order relation $\prec_{\mathcal{L}}$) by $p_1, \ldots, p_k$, then $P(v) = (p_1, \ldots, p_k)$. In Appendix C.2 we show that $\Delta(\mathcal{B}) \leq 2$ and $\Psi(\mathcal{B}) = O(\ell)$.

2.2 Zombies and Incubators

Algorithm *LightSp* starts with computing the path-spanner H and the base edge set $\mathcal{B}$. Next, it invokes Algorithm *BasicSp* to build a t-spanner $G'_0 = (Q_0, E'_0)$ for the sub-metric $M[Q_0]$ of M induced by $Q = Q_0$. Define $\tilde{E}_0$ to be the edge set obtained by *pruning* E'_0, i.e., removing all edges of weight greater than the *0-level threshold* τ_0. The corresponding graph $\tilde{G}_0 = (Q_0, \tilde{E}_0)$ is called the *0-level auxiliary spanner*. In a similar way (see Section 2.5), the algorithm builds an auxiliary j-level spanner $\tilde{G}_j$, for each $j \in [\ell]$. The union $\mathcal{B} \cup H \cup \bigcup_{j=0}^{\ell} \tilde{G}_j$ is the ultimate spanner $\tilde{G} = (Q, \tilde{E})$ that Algorithm *LightSp* returns.

As was discussed above, during the algorithm a j-level bag z may get integrated into a $(j+1)$-level bag v, $v \neq \pi(z)$. We now take a closer look on the process of integration.

Each bag $v \in \mathcal{F}$ may hold a *label* of exactly one of two types, a *zombie* and an *incubator*. Initially, all bags are unlabeled. As Algorithm *LightSp* proceeds, some bags may be assigned with labels.

It may happen that an i-level bag v is *abandoned* by its parent $\pi(v)$, and is *attached* to an i-level bag u. It must hold that $\pi(v) \neq \pi(u)$. We also say that v is *adopted* by $\pi(u) \in \mathcal{F}_{i+1}$. We denote the *attachment* of v to u by $\mathcal{A}(u, v)$. We also call it an *adoption* of v by $\pi(u)$. However, the attachment and adoption come with a suspension period, henceforth, *incubation period*. Specifically, there is a positive integer constant γ, to which we refer to as the *window*, which determines the length of the incubation period. The $(i + \gamma - 1)$-level ancestor v' of v will actually be *disintegrated* from its parent $\pi(v')$, and integrated into the $(i+\gamma)$-level ancestor u' of u. This bag is referred to as the *actual adopter*. It will be shown later (see Corollary D.4 in Appendix D.1) that adoption rules (which we still did not finish to specify) imply that $\pi(v') \neq u'$. We remark that adoptions occur only for $i \leq \ell - \gamma$. The sets $B(u'), K(u'), Q(u')$ and $B(\pi(v')), K(\pi(v')), Q(\pi(v'))$ are computed according to the rules specified in Section 2.2.

The $\gamma - 1$ immediate ancestors of $v = v^{(0)}$, namely, the bags $v^{(1)} = \pi(v), v^{(2)} = \pi(v^{(1)}), \ldots, v^{(\gamma-1)} = \pi(v^{(\gamma-2)}) = v'$, change their status as a a result of this attachment. They will be now labeled as *zombies*. The bag v' is called a *disappearing zombie*, because it gets integrated into u' rather than into its original parent $\pi(v')$. We will refer to v as an *attached bag*. Similarly, the $\gamma - 1$ immediate ancestors of $u = u^{(0)}$, namely, the bags $u^{(1)} = \pi(u), u^{(2)} = \pi(u^{(1)}), \ldots, u^{(\gamma-1)} = \pi(u^{(\gamma-2)})$, change their status as well. The bags $u^{(1)}, u^{(2)}, \ldots, u^{(\gamma-1)}$ will be labeled as *incubators*. The $(i+\gamma)$-level bag $u' = u^{(\gamma)}$ is *not* labeled as an incubator. This bag is called the *actual adopter*. We remark that the same bag may become an adopter (and incubator) of several different descendants. Note also that, since $i \geq 1$, for a j-level bag $u' \in \mathcal{F}_j$ to be an adopter, it must hold that $j = i + \gamma \geq \gamma + 1$. The i-level bag $u = u^{(0)}$ will be referred to as the *initiator* of the attachment $\mathcal{A}(u, v)$. (See Figure 1 for an illustration.)

2.3 Representatives

For a point $p \in Q$ and an index $j \in [\ell]$, denote by $v_j(p)$ the j-level *host bag* of p, i.e., the unique bag $v_j(p)$ that satisfies $p \in Q(v_j(p))$, $v_j(p) \in \mathcal{F}_j$.

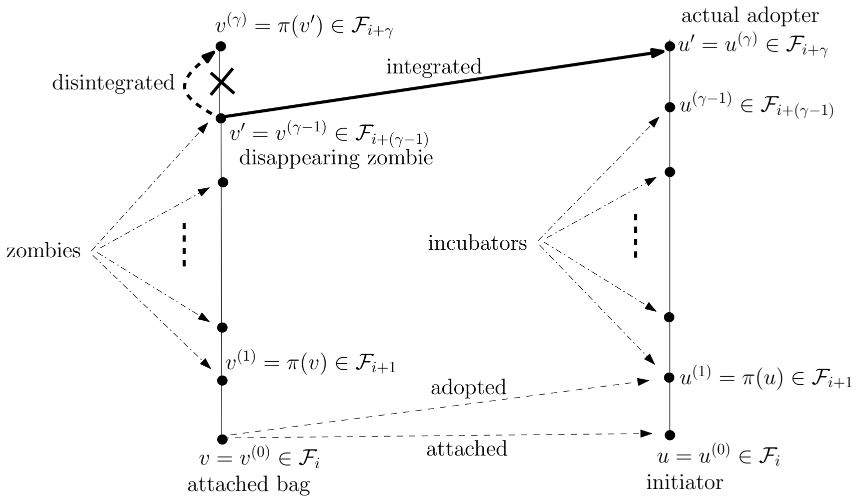

Figure 1: An illustration of an attachment $\mathcal{A}(u, v)$.

The algorithm maintains a few load indicators and counters for every point $p \in Q$. For each index $j \in [\ell]$, the load indicator $load_j(p)$ is equal to 1 if the point p is not isolated in the j-level auxiliary spanner $\tilde{G}_j$. Otherwise, $load_j(p)$ is set to 0. The *load counter* $load\_ctr_j(p)$ is defined by $load\_ctr_j(p) = \sum_{i=1}^{j} load_i(p)$. Algorithm *LightSp* also maintains three more refined load counters for every point p. Specifically, the *small counter* $ctr_j(p)$ (respectively, *large counter* $CTR_j(p)$) is the number of indices i, $1 \le i \le j$, such that the point p is not isolated in $\tilde{G}_i$ *and* its host bag $v_i(p)$ is small (resp., large). Note that $load\_ctr_j(p) = ctr_j(p) + CTR_j(p)$. The algorithm also counts the number of indices $i, 1 \le i \le j$, such that the point p is not isolated in $\tilde{G}_i$ *and* its host bag $v_i(p)$ satisfies $Q(v_i(p)) = \{p\}$. This counter is referred to as the *single counter* of p, and is denoted $single\_ctr_j(p)$. It also maintains the complementary counter $plain\_ctr_j(p) = ctr_j(p) - single\_ctr_j(p)$, which is referred to as the *plain counter* of p. (For convenience, all counters with index 0 are set as 0, i.e., $CTR_0(p) = ctr_0(p) = plain\_ctr_0(p) = single\_ctr_0(p) = 0$.)

A point $p \in Q$ may have edges incident on it in the j-level auxiliary spanner $\tilde{G}_j$ only if it is a representative of a j-level bag $v \in \mathcal{F}_j$. Hence we generally make an effort to select a representative with as small counter as possible. The specific way in which Algorithm *LightSp* selects representatives at the beginning of the j-level processing, $j \in [\ell]$, is the following one.

The representative $r(v)$ of a non-empty 1-level bag $v \in \mathcal{F}_1$ is selected arbitrarily from $K(v) = Q(v)$.

Next, consider a non-empty j-level bag, $j \in [2, \ell]$. The bag v is said to be a *growing bag* if $|\chi(v)| \ge 2$, i.e., if v is obtained as a result of integration of two or more non-empty $(j-1)$-level bags. Otherwise, the bag v is called *stagnating.* If v is a stagnating bag, then necessarily $\mathcal{J}(v) = \emptyset$, and $|\mathcal{S}(v)| = 1$.

If v is large (i.e., $|Q(v)| \ge \ell$) then Algorithm *LightSp* appoints a point $p \in K(v)$ with the smallest large counter $CTR_{j-1}(p)$ as its representative $r(v)$.

If v is small (i.e., $1 \le |Q(v)| < \ell$) then the algorithm checks whether it is a growing bag or a stagnating one. If v is a stagnating bag then $\mathcal{S}(v) = \{w\}$, for some $(j-1)$-level bag w. In this case Algorithm *LightSp* sets the representative $r(v)$ of v to be equal to the representative $r(w)$ of w, i.e., $r(v) = r(w)$. Otherwise, v is a growing small bag. In this case Algorithm *LightSp* appoints a point $p \in K(v)$ with the smallest plain counter $plain\_ctr_{j-1}(p)$ as its representative $r(v)$.

2.4 j-level processing

The routine that performs j-level processing (henceforth, Procedure $Process_j$), for $j \in [\ell]$, accepts as input the forest $\mathcal{F}$ that was processed by $Process_1, Process_2, \ldots, Process_{j-1}$. For each j-level bag w (i.e., $w \in \mathcal{F}_j$), Procedure $Process_j$ accepts as input the sets $B(w), K(w)$ and $Q(w)$. Also, for each bag $w \in \mathcal{F}_j$, the procedure accepts as input its representative $r(w) \in K(w) \subseteq Q$. It is also known to the procedure if this bag is (labeled as) a zombie, a disappearing zombie, or an incubator.

Denote by $Q_j = \{r(w) \mid w \in \mathcal{F}_j, Q(w) \neq \emptyset\}$ the set of representatives of the non-empty j-level bags. Observe that $|Q_j| \leq \min\{n, n_j\} = \min\left\{n, \frac{c \cdot n}{\rho^{j-1}}\right\}$.

Procedure $Process_j$ consists of three parts. *Part I* of Procedure $Process_j$ invokes Algorithm *BasicSp* for the metric $M[Q_j]$. The algorithm constructs a t-spanner $G'_j = (Q_j, E'_j)$. It then prunes G'_j, i.e., it removes from it all edges e with $\omega(e) > \tau_j$. Denote the resulting pruned graph $G^*_j = (Q_j, E^*_j)$. The edge set E^*_j is inserted into the spanner $\tilde{G}$. This completes the description of the Part I of Procedure $Process_j$.

While the spanner G'_j is connected, some points of Q_j may be isolated in G^*_j. Denote by Q^*_j the subset of Q_j of all points $q \in Q_j$ which are not isolated in G^*_j.

If $j \leq \ell - \gamma$ then Procedure $Process_j$ enters *Part II*, which is the main ingredient of Procedure $Process_j$. (Otherwise, Part II is skipped.) We need to introduce some more definitions before proceeding.

A bag v is called *useless* if it is either empty or a zombie. Otherwise it is called *useful*.

For a bag $v \in \mathcal{F}_i$, $i \leq \ell - \gamma$, its $(i+\gamma)$-level ancestor $v^{(\gamma)}$ is called the *cage-ancestor* of v. The set of all i-level descendants of $v^{(\gamma)}$, denoted $\mathcal{C}(v)$, is called the *cage* of v. If v is the only useful bag in its cage, it is called a *lonely* bag; otherwise it is called a *crowded* bag.

A bag v is called *safe* if it satisfies at least one of the following four conditions: (1) v is large, (2) v is crowded, (3) v is an incubator, (4) v is a zombie. Otherwise v is called *risky*. Note that for v to be risky it must be small (i.e., $|Q(v)| < \ell$), lonely, and neither an incubator nor a zombie. A representative $r(v)$ of a safe (respectively, risky; useful; zombie) bag v is called *safe* (resp., *risky*; *useful*; *zombie*) as well.

Part II of Procedure $Process_j$ starts with marking each bag $w \in \mathcal{F}_j$ (and its representative $r(w)$) as either useful or useless, and as either safe or risky. Denote by $\hat{Q}_j$ the subset of Q^*_j which contains only useful representatives. (Note that $\hat{Q}_j$ contains all points of Q^*_j, except for zombie representatives.) Then it invokes Algorithm *BasicSp*, this time with input $M[\hat{Q}_j]$. As a result, a graph $\check{G}_j = (\hat{Q}_j, \check{E}_j)$ is constructed. Next, it prunes $\check{G}_j$, i.e., it removes from it all edges e with $\omega(e) > \tau_j$. Denote by $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$ the resulting pruned graph. The edge set $\hat{E}_j$ is also inserted into the output spanner $\tilde{G}$. Let the *j-level auxiliary spanner* $\tilde{G}_j = (Q_j, \tilde{E}_j)$ denote the graph obtained as a union of the graphs $G^*_j = (Q_j, E^*_j)$ and $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$. (In case $j > \ell - \gamma$, we take the j-level auxiliary spanner $\tilde{G}_j$ to be G^*_j.)

Next, Part II of Procedure $Process_j$ constructs the (j-level) *attachment graph* $G_j = (\hat{Q}_j, \mathcal{E}_j)$ in the following way. It inserts into $\mathcal{E}_j$ all edges of $\hat{E}_j$, and also the edges of E^*_j restricted to the point set $\hat{Q}_j$. In other words, $\mathcal{E}_j = \hat{E}_j \cup E^*_j(\hat{Q}_j)$. Note that all vertices of G_j are labeled as either safe or risky.

Part II of Procedure $Process_j$ now invokes Procedure *Attach* on the graph $G_j = (\hat{Q}_j, \mathcal{E}_j)$. By Corollary C.1 (see Appendix C.1) this procedure returns a star forest Γ_j that satisfies that $R_j \subseteq \bigcup_{S \in \Gamma_j} V(S)$, where $R_j \subseteq \hat{Q}_j$ is the set of all risky points in $\hat{Q}_j$, which are not isolated in G_j. Also, each star $S \in \Gamma_j$ is centered at a center $s \in \hat{Q}_j$, which is either safe or risky. The star S also contains one or more leaves $q_1, \ldots, q_k \in \hat{Q}_j$, $k \geq 1$, which are all risky.

Next, Part II of Procedure $Process_j$ performs adoptions. Specifically, for each star $S \in \Gamma_j$ with center s and leaves $q_1, \ldots, q_k$, the host bags $v(q_1), \ldots, v(q_k)$ of $q_1, \ldots, q_k$, respectively, are attached to the host bag $v(s)$ of s. As a result the parent bag $\pi(v(s)) = v^{(1)}(s)$ of $v(s)$ adopts the bags $v(q_1), \ldots, v(q_k)$. In other words, $\gamma - 1$ of the immediate ancestors of $v(s)$ (in $\mathcal{F}$) $v^{(1)}(s) = \pi(v(s)), \ldots, v^{(\gamma-1)}(s)$ are labeled as incubators, and for each $i \in [k]$, $\gamma - 1$ of the immediate ancestors of $v(q_i)$ (in $\mathcal{F}$) $v^{(1)}(q_i), \ldots, v^{(\gamma-1)}(q_i)$ are labeled as zombies. Note that $v^{(\gamma-1)}(q_i)$ is a disappearing zombie, and $v^{(\gamma)}(s)$ is the actual adopter. We say that $v(s)$ *performs k attachments* $\mathcal{A}(v(s), v(q_1)), \mathcal{A}(v(s), v(q_2)), \ldots, \mathcal{A}(v(s), v(q_k))$. The bag $v(s)$ is the initiator of all these k attachments. The edges $\{(s, q_i) \mid i \in [k]\}$ that connect the center s of the star S with the leaves q_i of S, $i \in [k]$, belong to the attachment graph $G_j = (\hat{Q}_j, \mathcal{E}_j)$, and they are inserted into the spanner $\tilde{G}$. For each $i \in [k]$, we say that the spanner edge $(s, q_i) = (r(v(s)), r(v(q_i)))$ is a *representing edge* of the attachment $\mathcal{A}(v(s), v(q_i))$. This completes the description of Part II of Procedure $Process_j$.

Next, if $j \in [\ell - 1]$, Procedure $Process_j$ moves to the *Part III* of Procedure $Process_j$. (If $j = \ell$, part III is skipped.) Specifically, it computes the sets $\mathcal{S}(v)$ and $\mathcal{J}(v)$ of surviving children and joining step-children, respectively, for every bag $v \in \mathcal{F}_{j+1}$. This is done according to the set of adoptions which were computed in previous levels. In particular, a child w of v which is integrated into some other $(j+1)$-level vertex $u, u \neq v$, is excluded from $\mathcal{S}(v)$. Such a bag w is a disappearing zombie, and a step-child of u. Similarly, a bag $z \in \mathcal{F}_j$ with $\pi(z) \neq v$, which is a step-child of v (i.e., it is integrated into v), joins the set $\mathcal{J}(v)$. Given the sets $\mathcal{S}(v)$ and $\mathcal{J}(v)$, Procedure $Process_j$ computes the sets $B(v), K(v), Q(v)$, and the representative $r(v)$ of v. This completes the description of Part III (the last part) of Procedure $Process_j$.

Observe that for $j \in [\ell - \gamma]$, all three parts of Procedure $Process_j$ are executed. Also, for $j \in [\ell - \gamma + 1, \ell - 1]$, just Parts I and III of Procedure $Process_j$ are executed, and Part II is skipped. Finally, Procedure $Process_\ell$ (i.e., the ℓ-level processing) executes just Part I, and skips Parts II and III.

3 Analysis

This section is devoted to the analysis of the spanner $\tilde{G}$ constructed by Algorithm *LightSp*.

The suspension window γ is set as $\gamma = c_0 \cdot (\lceil \log_\rho t \rceil + \lceil \log_\rho c \rceil + 1)$, for a large constant c_0. (Note that $c = \Theta(t/\epsilon)$, hence $\gamma = \Theta(\log_\rho(t/\epsilon))$.) The following lemma is central in our analysis. It is used not only for bounding the stretch and the diameter of the spanner, but also for bounding its maximum degree.

Lemma 3.1 *Fix any index $j \in [\ell]$, and let $v \in \mathcal{F}_j, Q(v) \neq \emptyset$. Then for every $p \in Q(v)$, there is a (simple) path $\Pi_j(p)$ in the spanner $\tilde{G}$ that leads to a point $b_j(p)$ in the base point set $B(v)$ of v, having weight at most $\frac{1}{2} \cdot \mu_j$ and at most 3ℓ edges. Moreover, if $p \in K(v)$, then $\Pi_j(p)$ consists of at most 2ℓ edges. All points of $\Pi_j(p)$ belong to the point set $Q(v)$ of v. (The point $b_j(p)$ is called the* base point *of p.)*

Proof: The proof is by induction on j. The basis $j = 1$ is immediate.
Induction Step: Let $j \geq 2$, and $p \in Q(v)$. Also, let $u \in \chi(v) \subseteq \mathcal{F}_{j-1}$ be the $(j-1)$-level host bag of p.

Suppose first that $u \in \mathcal{S}(v)$, i.e., u is a non-empty surviving child of v in $\mathcal{F}$. In this case, $B(u) \subseteq B(v), K(u) \subseteq K(v), Q(u) \subseteq Q(v)$. Consider the path $\Pi_{j-1}(p)$ between p and its base point $b_{j-1}(p) \in B(u) \subseteq B(v)$ guaranteed by the induction hypothesis for u. Its weight is at most $\frac{1}{2} \cdot \mu_{j-1} = \frac{1}{2} \cdot \frac{\mu_j}{\rho} < \frac{1}{2} \cdot \mu_j$ and it consists of at most 3ℓ edges. Also, all points of $\Pi_{j-1}(p)$ belong to $Q(u) \subseteq Q(v)$. Moreover, suppose now that $p \in K(v)$. Recall that u is the unique $(j-1)$-level bag such that $p \in Q(u)$. Since $K(v) \subseteq \bigcup_{z \in \chi(v)} K(z)$ and each kernel set $K(z)$ is contained in $Q(z)$, it follows that $p \in K(u)$. By the induction hypothesis, $\Pi_{j-1}(p)$ consists of at most 2ℓ edges. Thus, we set $\Pi_j(p) = \Pi_{j-1}(p), b_j(p) = b_{j-1}(p)$.

We henceforth assume that u is a disappearing zombie, i.e., $u \in \mathcal{J}(v)$ is a joining step-child of v. In this case, since $v \in \mathcal{F}_j$ is an actual adopter, it must hold that $j \geq \gamma + 1$. For each index $i \in [0, \gamma - 1]$, let $y^{(i)}$ denote the $(j - 1 - (\gamma - 1) + i) = (j - \gamma + i)$-level *copy* of u. (That is, for each of these identical copies $y^{(i)}$, we have $B(y^{(i)}) = B(u), K(y^{(i)}) = K(u), Q(y^{(i)}) = Q(u)$. See Lemma D.5 in Appendix D.1.) In particular, $u = y^{(\gamma-1)}$ is a disappearing zombie, and $y^{(0)} = y$ is an attached bag. Observe that an attachment $\mathcal{A}(x, y)$, for some $(j - \gamma)$-level bag $x = x^{(0)}$, occurs during the $(j - \gamma)$-level processing. As a result of this attachment, $y = y^{(0)}$ became an attached bag. The initiator bag x of this attachment is a descendant of the actual adopter v. The bags $x^{(1)} = \pi(x^{(0)}), x^{(2)} = \pi(x^{(1)}), \ldots, x^{(\gamma-1)} = \pi(x^{(\gamma-2)})$ are labeled as a result of this attachment as incubators. Observe that $v = x^{(\gamma)} = \pi(x^{(\gamma-1)})$. Recall that the attachment $\mathcal{A}(x, y)$ is represented by the edge $(r(x), r(y))$ in the spanner $\tilde{G}$.

We will use the following claim to prove Lemma 3.1. (Its proof can be found in Appendix D.3.1.)

Claim 3.2 *Define $k = j - \gamma$. There is a simple path $\Pi(p, r(y))$ in $\tilde{G}$ between p and $r(y)$ that has weight at most $2 \cdot \mu_k$ and at most $\ell - 2$ edges. Also, all points of $\Pi(p, r(y))$ belong to $Q(y) = Q(u) \subseteq Q(v)$.*

Observe that at this point we have built a "good path" from p to $r(x)$, via $\Pi(p, r(y))$ and $(r(x), r(y))$. We now need to "connect" $r(x)$ to a base point $b_j(p) \in B(v)$. Since x is a descendant of v, it holds that $B(x) \subseteq B(v), K(x) \subseteq K(v), Q(x) \subseteq Q(v)$. As a representative of a bag must belong to its kernel, we have $r(x) \in K(x)$. By the induction hypothesis for x, there exists a path $\Pi_k(r(x))$ between $r(x)$ and its base point $b_k(r(x)) \in B(x) \subseteq B(v)$ in the spanner $\tilde{G}$. Moreover, all points of this path belong to $Q(x) \subseteq Q(v)$. In addition, the weight of this path is at most $\frac{1}{2} \cdot \mu_k = \frac{1}{2} \cdot \frac{\mu_j}{\rho^\gamma}$, and since $r_k(x) \in K(x)$, it consists of at most 2ℓ edges. We take $b_j(p)$ to be $b_k(r(x)) \in B(v)$, and take $\Pi_j(p)$ to be the path that is obtained as the concatenation of the path $\Pi(p, r(y))$ (guaranteed by Claim 3.2), the attachment edge $(r(x), r(y))$, and the path $\Pi_k(r(x))$. It is easy to see that $\Pi_j(p)$ is a path between p and its base point $b_j(p) = b_k(r(x))$, and that all points of $\Pi_j(p)$ belong to $Q(v)$. Notice that $\omega(r(x), r(y)) \leq \tau_k$. Therefore, the total weight $\omega(\Pi_j(p))$ of the path $\Pi_j(p) = \Pi(p, r(y)) \circ r(x), r(y) \circ \Pi_k(r(x))$ satisfies (for sufficiently large c_0)

$$\omega(\Pi_j(p)) \;\leq\; \mu_j \cdot \frac{\left(1 + 3 \cdot \frac{1}{2} + 2 \cdot (c+1) \cdot \rho \cdot t\right)}{(c \cdot \rho \cdot t)^{c_0}} \;<\; \frac{1}{2} \cdot \mu_j.$$

Also, it holds that $|\Pi_j(p)| \;=\; |\Pi(p, r(y))| + 1 + |\Pi_k(r(x))| \;\leq\; \ell - 2 + 1 + 2\ell \;\leq\; 3\ell$.

Suppose now that $p \in K(v)$. We argue that in this case x must be a small bag. Suppose for contradiction otherwise, and consider the $(j-1)$-level ancestor $x^{(\gamma-1)}$ of x, which is a surviving child of $v = x^{(\gamma)}$. Observe that $Q'(v) \;=\; \bigcup_{z \in \mathcal{S}(v)} Q(z) \;\supseteq\; Q(x^{(\gamma-1)}) \;\supseteq\; Q(x)$, and so $|Q'(v)| \geq |Q(x)| \geq \ell$. By construction, $K(v) = K'(v) = \bigcup_{z \in \mathcal{S}(v)} K(z)$. Hence the kernel set $K(v)$ of v contains only points from the kernel sets of its surviving children, and contains no points from its joining step-children. However, $p \in Q(u)$, and u is a joining step-child of v. Hence $p \;\notin\; K(v)$, a contradiction.

Therefore x is a small bag. Since all points of $\Pi_k(r(x))$ belong to x, it follows that this path consists of at most $\ell - 2$ edges (rather than at most 2ℓ edges as in the general case). Consequently, $|\Pi_j(p)| \;=\; |\Pi(p, r(y))| + 1 + |\Pi_k(r(x))| \;\leq\; \ell - 2 + 1 + \ell - 2 \;\leq\; 2\ell$. ▮

Lemma 3.1 implies the following corollary. (The proof is deferred to Appendix D.3.2.)

Corollary 3.3 *Fix an arbitrary index $j \in [\ell]$, and let v be an arbitrary non-empty j-level bag. There is a path in the spanner $\tilde{G}$ between every pair of points in $Q(v)$, having weight at most $2 \cdot \mu_j$ and at most $O(\log_\rho n + \alpha(\rho))$ edges. In particular, the metric distance between any two points in $Q(v)$ is at most $2 \cdot \mu_j$.*

The next lemma implies that $\tilde{G}$ is a $(t + \epsilon)$-spanner for $M[Q]$ with diameter $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$.

Lemma 3.4 *For any $p, q \in Q$, there is a $(t + \epsilon)$-spanner path in $\tilde{G}$ with $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$ edges.*

Proof: We start the proof of the lemma with the following observation (see Appendix D.3.3 for the proof).

Observation 3.5 *Fix any index $j \in [0, \ell]$. For any pair $u, v \in \mathcal{F}_j$ of non-empty j-level bags, such that $\delta(r(u), r(v)) \leq \frac{\tau_j}{t}$, there is a t-spanner path in G^*_j between $r(u)$ and $r(v)$ with at most $\Lambda(n)$ edges.*

Let $p, q \in Q$. Suppose first that $\delta(p, q) \leq \frac{L}{n} < \frac{\tau_0}{t}$. Note that the graph $\tilde{G}_0 = G^*_0$ belongs to $\tilde{G}$. By Observation 3.5 for $j = 0$, there is a t-spanner path in $\tilde{G}_0$ between p and q with at most $\Lambda(n)$ edges.

We henceforth assume that $\delta(p, q) > \frac{L}{n}$. Let $j \in [\ell]$ be the index such that $\rho^{j-1} \cdot \frac{L}{n} < \delta(p, q) \leq \rho^j \cdot \frac{L}{n}$, i.e., $\xi_j < \delta(p, q) \leq \rho \cdot \xi_j$. Let $u = v_j(p)$ (respectively, $w = v_j(q)$) be the j-level host bag of p (resp., q). By Corollary 3.3, the metric distance between every pair of points in the same j-level bag is at most $2 \cdot \mu_j < \xi_j$. Since $\delta(p, q) > \xi_j$, it follows that $u \neq w$. Consider the representative $r(u) \in Q_j$ (respectively, $r(w) \in Q_j$) of u (resp., w); by Corollary 3.3, $\delta(p, r(u)), \delta(q, r(w)) \leq 2 \cdot \mu_j = 2 \cdot \frac{\xi_j}{c}$. It follows that

$$\delta(r(u), r(w)) \;\leq\; \delta(p, r(u)) + \delta(p, q) + \delta(q, r(w)) \;\leq\; \rho \cdot \xi_j + 4 \cdot \frac{\xi_j}{c} \;\leq\; 2\rho^j \cdot \frac{L}{n}\left(1 + \frac{1}{c}\right) \;=\; \frac{\tau_j}{t}. \quad (1)$$

By Observation 3.5, there is a t-spanner path between $r(u)$ and $r(w)$ in G^*_j (and thus in $\tilde{G}$) with at most $\Lambda(n)$ edges; denote this path by $\Pi^*(r(u), r(w))$, and observe that $\omega(\Pi^*(r(u), r(w))) \leq t \cdot \delta(r(u), r(w))$. Also, by Corollary 3.3, the spanner $\tilde{G}$ contains a path $\Pi(p, r(u))$ (respectively, $\Pi(q, r(w))$) between p and $r(u)$ (resp., between q and $r(w)$) that has weight at most $2 \cdot \mu_j = 2 \cdot \frac{\xi_j}{c}$ and $O(\log_\rho n + \alpha(\rho))$ edges.

Let $\Pi(p, q) \;=\; \Pi(p, r(u)) \circ \Pi^*(r(u), r(w)) \circ \Pi(q, r(w))$. Note that $\Pi(p, q)$ is a path in $\tilde{G}$ between p and q that has weight $\omega(\Pi(p, q))$ at most $t \cdot \delta(r(u), r(w)) + 4 \cdot \frac{\xi_j}{c}$ and $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$ edges. By Equation (1), $t \cdot \delta(r(u), r(w)) \;\leq\; t \cdot (\delta(p, q) + 4 \cdot \frac{\xi_j}{c})$. Also, recall that $c = \lceil \frac{4 \cdot (t+1)}{\epsilon} \rceil$. It follows that

$$\omega(\Pi(p, q)) \;\leq\; t \cdot \left(\delta(p, q) + 4 \cdot \frac{\xi_j}{c}\right) + 4 \cdot \frac{\xi_j}{c} \;\leq\; \left(t + \frac{4 \cdot (t+1)}{c}\right) \cdot \delta(p, q) \;\leq\; (t + \epsilon) \cdot \delta(p, q). \quad ▮$$

To analyze the number of edges and weight in $\tilde{G} = (Q, \tilde{E})$, note that, roughly speaking, $\tilde{G}$ is a union of $\ell + 1$ auxiliary spanners $\tilde{G}_j = (Q_j, \tilde{E}_j), j \in [0, \ell]$. (It also contains the base edge set $\mathcal{B}$ and the path-spanner H. However, their contribution can be neglected.) For the case of Euclidean and doubling metrics, $|\tilde{E}_j| = O(|Q_j|)$, for each $j \in [0, \ell]$. Hence $|\tilde{E}| \approx \sum_{j=0}^{\ell} |\tilde{E}_j| = O(\sum_{j=0}^{\ell} |Q_j|)$. Since the sequence $|Q| = |Q_0|, |Q_1|, \ldots, |Q_\ell|$ decays geometrically, we have $|\tilde{E}| = O(|Q_0|) = O(n)$. (Formally, the sequence starts to decay from the $O(\log_\rho(t/\epsilon))$th element.) For the weight analysis, recall that each auxiliary spanner $\tilde{G}_j$ is pruned according to the weight threshold τ_j. These thresholds grow geometrically, at the same rate as the cardinalities of the sets Q_j decay. That is, $|Q_j| \leq \frac{O(|Q_0|)}{\rho^{j-1}}$, and $\tau_j = \rho^j \cdot \tau_0$. Hence, roughly speaking, $\omega(\tilde{E}) \approx \sum_{j=0}^{\ell} \omega(\tilde{E}_j) = \sum_{j=0}^{\ell} O(|Q_j| \cdot \tau_j) = O(\rho \cdot \log_\rho n) \cdot \omega(MST(M[Q]))$. For general metrics the analysis is very similar to the above. The analysis of the running time is similar to the analysis of $|\tilde{E}|$. See Appendix D.2 for a rigorous analysis of the number of edges, weight and running time.

The degree analysis, however, is far more involved. Together with the analysis of the stretch and the (hop-)diameter of $\tilde{G}$ which was provided above, the degree analysis constitutes the most complex part of this paper. We next sketch the intuition behind it.

If a j-level bag v is large, i.e., $|Q(v)| \geq \ell$, then all its $\ell - j$ ancestors are large as well. Each point $p \in Q(v)$ gets loaded by one of the auxiliary spanners $\tilde{G}_j, \tilde{G}_{j+1}, \ldots, \tilde{G}_\ell$ only if it is a representative of v or of one of its ancestors. However, we have at least ℓ points to "represent v" in at most ℓ auxiliary spanners. (One for v, and one for each of its ancestors.) Hence it is not hard to share the load in such a way that each point $p \in Q(v)$ will be loaded by $O(1)$ auxiliary spanners. Consequently, the maximum degree of points that belong to large bags are small. (In fact, a point p may, of course, belong to a small bag, and later join a large bag. However, for the sake of this discussion one can imagine that p duplicates itself into p^{large} and p^{small}, where p^{large} (respectively, p^{small}) belongs only to large (resp., small) bags.) On the other hand, for a small bag v, its representative $r(v)$ is loaded by a j-level auxiliary spanner only if $r(v)$ is not isolated in G_j^* (see Section 2.5). It means that there exists another j-level representative $r(u)$, such that $\delta(r(v), r(u)) \leq \tau_j$; in other words, $r(u)$ is close to $r(v)$. Intuitively, we will want the bags v and u to merge, as this would increase the pool of eligible representatives. We cannot merge them right away, however, because this would blow up the weighted diameters of the $(j+1)$-level bags. Instead we wait for $\gamma = O(1)$ levels, and then merge v into the $(j+\gamma)$-level ancestor u' of u. (Or the other way around, merge u into the $(j+\gamma)$-level ancestor v' of v.) The weighted diameters of the j-level bags, are, roughly speaking, proportional to the length μ_j of the j-level intervals, i.e., they grow geometrically with the level j. Hence when v is merged into u', it contributes only an $\left(\frac{1}{2}\right)^{O(\gamma)}$-fraction to the weighted diameter of the $(j+\gamma)$-level bag u'. In this way we keep the weighted diameters of bags in check, while always maintaining sufficiently large pools of eligible representatives. During the γ levels $j, j+1, \ldots, j+\gamma-1$, points of v do accumulate some extra degree; however, since $\gamma = O(1)$, they are overloaded by at most a constant factor. Rigorous analysis of the degree of the spanner $\tilde{G}$ can be found in Appendix D.4.

The next theorem summarizes our main result. (See also Appendix A.)

Theorem 3.6 *For any n-point doubling metric M, any $\epsilon > 0$ and any integer parameter $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner with $O(n)$ edges, degree $O(\rho)$, diameter $O(\log_\rho n + \alpha(\rho))$ and lightness $O(\rho \cdot \log_\rho n)$. The running time of this construction is $O(n \log n)$.*

As was mentioned in Section 1.6, this theorem follows from Theorem 1.3 by instantiating the algorithm from Theorem 1.2 as Algorithm *BasicSp*. Theorem 3.6 implies Conjecture 1 of Arya et al. [4] by setting $\rho = O(1)$ and observing that any Euclidean metric of constant dimension is a doubling metric.

Acknowledgments. The second-named author is indebted to Michiel Smid, for many helpful and timely comments, and for his constant support and willingness to help. Also, we wish to thank Adi Gottlieb, for referring us to [25], and for many helpful discussions.

Appendix

A The General Result

Next we explicate the dependence on ϵ and the doubling dimension in Theorem 3.6.

Theorem A.1 *For any n-point metric M with an arbitrary (not necessarily constant) doubling dimension $dim(M)$, any $\epsilon > 0$ and any integer parameter $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner with $n \cdot \epsilon^{-O(dim(M))}$ edges, degree $\rho \cdot \epsilon^{-O(dim(M))}$, diameter $O(\log_\rho n + \alpha(\rho))$ and lightness $(\rho \cdot \log_\rho n) \cdot \epsilon^{-O(dim(M))}$. The running time of this construction is $(n \log n) \cdot \epsilon^{-O(dim(M))}$.*

B Proof of Theorem 1.2

This appendix is devoted to the proof of Theorem 1.2. For Euclidean metrics Arya et al. [4] proved this theorem for the case $\rho = 2$, and the authors of the current paper generalized it in [44] to the entire range of the degree parameter ρ. For doubling metrics the proof of this theorem is based on the works of [25] and [44]. We provide it here for the sake of completeness.

Let $M = (P, \delta)$ be an n-point doubling metric. A $(1+\epsilon)$-spanner H for M is called a *tree-like spanner*, if it contains a tree T that satisfies the following conditions:

1. Each vertex v of T is assigned a representative point $r(v) \in P$.
2. There is a 1-1 correspondence between the points of P and the representatives of the leaves of T.
3. Each internal vertex is assigned a unique representative. (Thus, each point of P will be the representative of at most two vertices of T.) In particular, there are at most $2n$ vertices in T.
4. For any two points $p, q \in P$, there is a $(1+\epsilon)$-spanner path in H between p and q that is composed of three consecutive parts: (a) a path ascending the edges of T, (b) a single edge, and (c) a path descending the edges of T. (Each edge $e = (u, v)$ in T is translated into an edge $(r(u), r(v))$ in H.)

We say that such a tree T is a *tree-skeleton* of the spanner H.

Gottlieb and Roditty [25] proved the following theorem. (See also [22, 10, 16, 40, 24] for a number of earlier related works.)

Theorem B.1 ([25]) *For any n-point doubling metric $M = (P, \delta)$ and any $\epsilon > 0$, one can build in $O(n \log n)$ time a $(1+\epsilon)$-spanner H and a tree-skeleton T for H, such that both H and T have constant degree.*

The spanner of Gottlieb and Roditty [25] may have a large diameter. To reduce the diameter, we employ the following tree-shortcutting theorem from [44].

Theorem B.2 (Theorem 3 in [44]) *Let T be an arbitrary n-vertex tree, and denote by M_T the tree metric induced by T. One can build in $O(n \log_\rho n)$ time, for any integer $\rho \geq 2$, a 1-spanner G_ρ for M_T with $|G_\rho| = O(n)$, $\Delta(G_\rho) \leq \Delta(T) + 2\rho$, and $\Lambda(G_\rho) = O(\log_\rho n + \alpha(\rho))$.*

Next, we describe a spanner construction H^* that satisfies all conditions of Theorem 1.2.

We start by building the spanner H and its tree-skeleton T that are guaranteed by Theorem B.1. Note that T contains at most $2n = O(n)$ vertices. Next, we build the 1-spanner G_ρ for the tree metric $M_T = (P, \delta_T)$ induced by T that is guaranteed by Theorem B.2. Notice that the edge weights of G_ρ are assigned according to the distance function δ_T of the tree metric M_T. The 1-spanner G_ρ is converted into a graph G^*_ρ over the point set P in the following way. Each edge (u, v) of G_ρ, for a pair u, v of vertices in T, is translated into the edge $(r(u), r(v))$ between their respective representatives. Finally, let H^* be the spanner obtained from the union of the graphs H and G^*_ρ.

It is easy to see that the graph H^* satisfies all conditions of Theorem 1.2.

C Procedure *Attach* and Base Edges

C.1 Procedure *Attach*

In this appendix we describe Procedure *Attach*, which is used as a building block for Algorithm *LightSp*.

Recall that Procedure *Attach* accepts as input an n-vertex graph $G = (V, E)$, whose vertices are labeled by either safe or risky. The procedure returns as output a star forest that satisfies some desired conditions (see Corollary C.1 below).

Intuitively, Procedure *Attach* attaches each risky vertex to some other vertex. Each star of Γ will eventually be merged into a single super-vertex in a certain supergraph in our algorithm. This will be, roughly speaking, our way to "get rid" of risky vertices.

Procedure *Attach* starts with forming the attachment digraph $\mathcal{G}$ as follows: for every vertex $z \in R$, we pick an arbitrary neighbor $x \in V$ of z in G, and insert the arc $\langle z, x\rangle$ into $\mathcal{G}$.

It is easy to see that each vertex $z \in R$ has out-degree one in $\mathcal{G}$, and each vertex $s \in V \setminus R$ (in particular, each vertex labeled as safe) has out-degree zero in $\mathcal{G}$.

Procedure *Attach* proceeds in two stages.

The first stage is carried out iteratively. At each iteration the procedure picks an arbitrary non-isolated vertex z in $\mathcal{G}$ with in-degree zero, and handles it as follows. Since z is non-isolated, it must have an outgoing neighbor s; thus z must be labeled as risky. The procedure removes the edge $\langle z, s\rangle$ from $\mathcal{G}$.

Next, suppose that s is the center of some existing star S' in Γ. In this case the procedure adds the vertex z as well as the recently removed edge $\langle z, s\rangle$ into S'. The vertex z is designated as a leaf of S'.

Otherwise, s does not belong yet to any star in Γ. In this case the procedure forms a new star S and adds it to Γ. It then adds the vertices z and s as well as the recently removed edge $\langle z, s\rangle$ into S. The vertex s is designated as the center of S and the vertex z is designated as a leaf of S. Moreover, if s has an outgoing neighbor s' in $\mathcal{G}$, the procedure removes the edge $\langle s, s'\rangle$ from $\mathcal{G}$. (By removing the edge $\langle s, s'\rangle$ from $\mathcal{G}$, we guarantee that s will not be added to any other star in subsequent iterations.)

The first stage terminates when all the non-isolated vertices in $\mathcal{G}$ have in-degree at least one. Let V' be the set of non-isolated vertices in $\mathcal{G}$ at the end of the first stage, and denote by $\mathcal{G}' = \mathcal{G}[V']$ the subgraph of $\mathcal{G}$ induced by the vertex set V'. If $\mathcal{G}'$ is empty, then the procedure *Attach* terminates. Otherwise, the second stage of the procedure starts.

Notice that all vertices of $\mathcal{G}'$ have in-degree at least one, and so $|\mathcal{G}'| \geq |V'|$. Also, the out-degree of each vertex in $\mathcal{G}'$ is at most one, and so $|\mathcal{G}'| = |V'|$. It follows that both the in-degree and the out-degree of each vertex of $\mathcal{G}'$ must be equal to one. This, in turn, means that all the vertices of $\mathcal{G}'$ are labeled as risky (i.e., $V' \subseteq R$). Moreover, the graph $\mathcal{G}'$ is comprised of a collection $\mathcal{C}$ of directed vertex disjoint cycles. Consider a cycle $C = (v_0, \ldots, v_{g-1}, v_g = v_0) \in \mathcal{C}$, for some positive integer $g \geq 2$. If g is even then the procedure forms $\frac{g}{2}$ stars $\{\langle v_0, v_1\rangle\}, \ldots, \{\langle v_{g-2}, v_{g-1}\rangle\}$, each containing a single arc. Otherwise (if g is odd), the procedure forms $\frac{g-1}{2}$ stars $\{\langle v_0, v_1\rangle\}, \ldots, \{\langle v_{g-3}, v_{g-2}\rangle, \langle v_{g-1}, v_{g-2}\rangle\}$. (Each of these stars except for the last one contains one arc, and the last contains two arcs. Note that the orientation of the arc $\langle v_{g-2}, v_{g-1}\rangle$ gets inverted.) In both cases each of these $\lfloor \frac{g}{2} \rfloor$ stars is added to Γ.

This completes the description of the procedure *Attach*.

It is easy to verify that the graph Γ constructed by the procedure *Attach* is a star forest that satisfies the two conditions listed above. Moreover, it is straightforward to implement this procedure in time $O(|V|)$. We summarize this appendix in the following corollary.

Corollary C.1 *Procedure Attach, given a graph $G = (V, E)$ whose vertices are labeled by either safe or risky, produces a star forest that satisfies properties 1 and 2 (listed in the beginning of this appendix). The running time of this procedure is $O(|V|)$.*

C.2 The Base Edge Set $\mathcal{B}$

In this appendix we describe how the base edge set $\mathcal{B}$ is formed, and analyze its properties.

Fix an index $j, j \in [\ell]$. For each non-empty bag $v \in \mathcal{F}_j$, let $x(v)$ (respectively, $y(v)$) denote the leftmost (resp., rightmost) (with respect to $\prec_{\mathcal{L}}$) point in the base point set $B(v)$ of v. The next observation, which follows easily from the construction, implies that the order relation $\prec_{\mathcal{L}}$ can be used in the obvious way to define a total order on the non-empty bags of $\mathcal{F}_j$.

Observation C.2 *For any pair u, v of distinct non-empty bags in $\mathcal{F}_j$, either $x(u) \preceq_{\mathcal{L}} y(u) \prec_{\mathcal{L}} x(v) \preceq_{\mathcal{L}} y(v)$ or $x(v) \preceq_{\mathcal{L}} y(v) \prec_{\mathcal{L}} x(u) \preceq_{\mathcal{L}} y(u)$ must hold. With a slight abuse of notation, we will write $u \prec_{\mathcal{L}} v$ in the former case and $v \prec_{\mathcal{L}} u$ in the latter.*

We may henceforth assume without loss of generality that, for each bag $v \in \mathcal{F}_j$, with $j \geq 2$, its surviving children $c^{(1)}(v), c^{(2)}(v), \ldots, c^{(h)}(v)$ are ordered such that $c^{(1)}(v) \prec_{\mathcal{L}} c^{(2)}(v) \prec_{\mathcal{L}} \ldots \prec_{\mathcal{L}} c^{(h)}(v)$.

Next, we turn to describe how the base edge set $\mathcal{B}$ is formed.

On the bottom-most level ($j = 1$), for each bag $v \in \mathcal{F}_1$, we order all points of $B(v) = \bar{Q}(v)$ from left to right, according to their respective order in $\mathcal{L}$. In other words, write $B(v) = (p_1, p_2, \ldots, p_{|B(v)|})$, where $p_1 \prec_{\mathcal{L}} p_2 \prec_{\mathcal{L}} \ldots \prec_{\mathcal{L}} p_{|B(v)|}$. The $(|B(v)| - 1)$ edges $(p_1, p_2), \ldots, (p_{|B(v)|-1}, p_{|B(v)|})$ form the *base edge set* $\mathcal{B}(v)$ of the bag v. The union $\mathcal{B}_1 = \bigcup_{v \in \mathcal{F}_1} \mathcal{B}(v)$ is the *1-level base edge set*.

For $j \geq 2$, the *base edge set* $\mathcal{B}(v)$ of a j-level bag v is formed in the following way. Recall that $c^{(1)}(v), c^{(2)}(v), \ldots, c^{(h)}(v)$ denote the surviving children of v from left to right (w.r.t. $\prec_{\mathcal{L}}$), and denote by $x^{(i)}(v)$ (respectively, $y^{(i)}(v)$) the left-most (resp., right-most) point in the base point set $B(c^{(i)}(v))$ of $c^{(i)}(v)$, for each index $i \in [h]$. Then the *base edge set* $\mathcal{B}(v)$ of v will be the edge set $\mathcal{B}(v) = \{(y^{(1)}(v), x^{(2)}(v)), (y^{(2)}(v), x^{(3)}(v)), \ldots, (y^{(h-1)}(v), x^{(h)}(v))\}$. Given the base edge sets of all j-level bags $v \in \mathcal{F}_j$, the *j-level base edge set* $\mathcal{B}_j$ is formed as their union, i.e., $\mathcal{B}_j = \bigcup_{v \in \mathcal{F}_j} \mathcal{B}(v)$. Finally, the *base edge set* $\mathcal{B}$ is formed as the union $\mathcal{B} = \bigcup_{j=1}^{\ell} \mathcal{B}_j$.

We also define the *recursive base edge set* $\hat{\mathcal{B}}(v)$ of a j-level bag v in the following way. For $j = 1$, $\hat{\mathcal{B}}(v) = \mathcal{B}(v)$. For $j \in [2, \ell]$, the recursive base edge set $\hat{\mathcal{B}}(v)$ of v is defined as the union of the recursive base edge sets $\hat{\mathcal{B}}(c^{(1)}(v)), \ldots, \hat{\mathcal{B}}(c^{(h)}(v))$ of its surviving children $c^{(1)}(v), \ldots, c^{(h)}(v)$, respectively, union with the base edge set $B(v)$ of v. In other words, $\hat{\mathcal{B}}(v) = B(v) \cup \bigcup_{i=1}^{h} \hat{\mathcal{B}}(c^{(i)}(v))$. The following lemma follows from the construction by a straightforward induction.

Lemma C.3 *Fix an arbitrary index $j \in [\ell]$, and let v be an arbitrary non-empty j-level bag. Let $B(v) = (p_1, \ldots, p_{|B(v)|})$ be the base point set of v, ordered according to $\prec_{\mathcal{L}}$. (In other words, $p_1 \prec_{\mathcal{L}} p_2 \prec_{\mathcal{L}} \ldots \prec_{\mathcal{L}} p_{|B(v)|}$.) Then the recursive base edge set $\hat{\mathcal{B}}(v)$ is the edge set given by $\hat{\mathcal{B}}(v) = \{(p_1, p_2), \ldots, (p_{|B(v)|-1}, p_{|B(v)|})\}$.*

Consider the path $P(v) = ((p_1, p_2), \ldots, (p_{|B(v)|-1}, p_{|B(v)|}))$. By Lemma C.3, the edge set of the path $P(v)$ is equal to the recursive base edge set $\hat{\mathcal{B}}(v)$ of v. (The base point set $B(v)$ is equal to $\{p_1, p_2, \ldots, p_{|B(v)|}\}$, with $p_1 \prec_{\mathcal{L}} p_2 \prec_{\mathcal{L}} \ldots \prec_{\mathcal{L}} p_{|B(v)|}$.)

For a point p and an index $j \in [\ell]$, we say that a bag $v \in \mathcal{F}_j$ (if exists) is the *j-level base bag* of p if $p \in B(v)$. Recall that for a pair $u, v \in \mathcal{F}_j$ of distinct bags, $B(u) \cap B(v) = \emptyset$. Hence for any point p, there is at most one j-level base bag. Moreover, for any point p there exists a 1-level base bag. However, on subsequent levels the base bag of p may not exist; this happens when the base bag v of p becomes a step-child of some other bag u different from its parent $\pi(v)$ in $\mathcal{F}$. In other words, for any point p, there exists an index $j = j(p) \in [\ell]$ such that there exist i-level base bags for p, for all indices $1 \leq i \leq j$, and there are no i-level base bags for p, for all indices $j + 1 \leq i \leq \ell$. We will say that the base bags of p in iterations $1, \ldots, j - 1$ are *surviving*, and the base bag of p in iteration j is *disappearing*.

Next, we argue that the maximum degree $\Delta(\mathcal{B})$ of the base edge set $\mathcal{B}$ is at most 2, and that its lightness $\Psi(\mathcal{B})$ is $O(\ell)$.

We start with analyzing $\Delta(\mathcal{B})$. For each point $p \in Q$ and any index $j \in [\ell]$, we say that a point $q \in Q$ is a *left neighbor* (respectively, *right neighbor*) of p in $\mathcal{B}_j$ if the edge (p, q) belongs to $\mathcal{B}_j$ and $q \prec_{\mathcal{L}} p$ (resp., $p \prec_{\mathcal{L}} q$). In addition, we will say that q is a left neighbor (respectively, right neighbor) of p in $\mathcal{B}$, if there exists an index $j \in [\ell]$, such that q is is a left neighbor (resp., right neighbor) of p in $\mathcal{B}_j$. The *left degree* (resp., *right degree*) of p in $\mathcal{B}$, denoted $leftdeg_{\mathcal{B}}(p)$ (resp., $rightdeg_{\mathcal{B}}(p)$) is the number of left neighbors (resp., right neighbors) q of p in $\mathcal{B}$.

Next, we argue that for every point $p \in Q$, $rightdeg_{\mathcal{B}}(p) \leq 1$. Symmetrically, it also holds that $leftdeg_{\mathcal{B}}(p) \leq 1$. We will conclude that $deg_{\mathcal{B}}(p) = leftdeg_{\mathcal{B}}(p) + rightdeg_{\mathcal{B}}(p) \leq 2$, and thus $\Delta(\mathcal{B}) \leq 2$.

Lemma C.4 *For every point $p \in Q$, $rightdeg_{\mathcal{B}}(p) \leq 1$.*

Proof: Suppose first that p is not the rightmost point of a base point set $B(v)$, for some 1-level bag $v \in F_1$. Denote by p' the right neighbor of p in $\mathcal{B}_1$. In this case, by construction, for every bag $v \in \mathcal{F}$ such that $p \in B(v)$, it also holds that $p' \in B(v)$. Therefore, p will not be the rightmost point of $B(v)$, for any bag $v \in \mathcal{F}$. Hence p will not have any right neighbor in $\bigcup_{j=2}^{\ell} \mathcal{B}_j$, and so $rightdeg_{\mathcal{B}}(p) = 1$.

Suppose now that p is the right-most point of a base point set $B(v)$, for a 1-level bag $v \in \mathcal{F}_1$. Let $h, h \in [\ell]$, denote the maximum level such that p is the right-most point of a base point set $B(v)$, for an h-level bag $v \in \mathcal{F}_h$. By construction, p will not have any right neighbor in $\bigcup_{j=1}^{h} \mathcal{B}_j$. Denote by $v_i \in \mathcal{F}$ the base bag of p on level i (if exists), for each index $i \in [\ell]$. (In other words, $p \in B(v_i)$, for each index i as above.) Recall that there exists an index $j \in [\ell]$ such that the bags $v_1, v_2, \ldots, v_{j-1}$ are surviving, but the bag v_j is disappearing. It holds, however, that $h \leq j$. If the bag v_h is disappearing (i.e., if $h = j$), then the point p acquires no right degree on levels $h+1, h+2, \ldots, \ell$. Hence, in this case $rightdeg_{\mathcal{B}}(p) = 0$. Otherwise the point p acquires exactly one right neighbor on level $h+1$. From that moment on, however, p will no longer be the rightmost point of the base point sets $B(v_i)$ of its host bags. Hence it acquires no additional right neighbors on subsequent levels. In this case $rightdeg_{\mathcal{B}}(p) = 1$. ∎

Corollary C.5 $\Delta(\mathcal{B}) \leq 2$.

Corollary C.5 implies that there are at most n base edges in $\mathcal{B}$.

Corollary C.6 $|\mathcal{B}| \leq n$.

Next we analyze the lightness of the base edge set $\mathcal{B}$.

Lemma C.7 *Fix an arbitrary index $j \in [\ell]$, and let $e = (p, q), e' = (p', q')$ be a pair of distinct edges in the j-level base edge set $\mathcal{B}_j$, such that $p \prec_{\mathcal{L}} q$ and $p' \prec_{\mathcal{L}} q'$. Then either $p \prec_{\mathcal{L}} q \prec_{\mathcal{L}} p' \prec_{\mathcal{L}} q'$ or $p' \prec_{\mathcal{L}} q' \prec_{\mathcal{L}} p \prec_{\mathcal{L}} q$ must hold.*

Proof: Suppose first that e and e' belong to the base edge set $\mathcal{B}(v)$ of the same j-level bag v, and write $\mathcal{B}(v) = \{(y^{(1)}(v), x^{(2)}(v)), (y^{(2)}(v), x^{(3)}(v)), \ldots, (y^{(h-1)}(v), x^{(h)}(v))\}$. In this case $e = (y^{(i)}(v), x^{(i+1)}(v))$ and $e' = (y^{(i')}(v), x^{(i'+1)}(v))$, for two distinct indices $i, i' \in [h-1]$. By Observation C.2, $y^{(i)}(v) \prec_{\mathcal{L}} x^{(i+1)}(v)$ and $y^{(i')}(v) \prec_{\mathcal{L}} x^{(i'+1)}(v)$, which implies that $p = y^{(i)}(v), q = x^{(i+1)}(v), p' = y^{(i')}(v), q' = x^{(i'+1)}(v)$. Applying Observation C.2 again, we get that $q \prec_{\mathcal{L}} p'$ holds in the case $i < i'$, whereas $q' \prec_{\mathcal{L}} p$ holds in the complementary case $i' < i$. Thus either $p \prec_{\mathcal{L}} q \prec_{\mathcal{L}} p' \prec_{\mathcal{L}} q'$ or $p' \prec_{\mathcal{L}} q' \prec_{\mathcal{L}} p \prec_{\mathcal{L}} q$ must hold.

Otherwise, e belongs to some base edge set $\mathcal{B}(v)$ and e' belong to another base edge set $\mathcal{B}(w)$, where v and w are two distinct bags in $\mathcal{F}_j$. By construction, both endpoints p and q of e (respectively, p' and q' of e') belong to the base point set $B(v)$ of v (resp., $B(w)$ of w). By Observation C.2, we get that $q \prec_{\mathcal{L}} p'$ holds in the case $v \prec_{\mathcal{L}} w$, whereas $q' \prec_{\mathcal{L}} p$ holds in the complementary case $w \prec_{\mathcal{L}} v$. Thus either $p \prec_{\mathcal{L}} q \prec_{\mathcal{L}} p' \prec_{\mathcal{L}} q'$ or $p' \prec_{\mathcal{L}} q' \prec_{\mathcal{L}} p \prec_{\mathcal{L}} q$ must hold, and we are done. ∎

Lemma C.7 and the triangle inequality imply that the weight $\omega(\mathcal{B}_j)$ of the j-level base edge set $\mathcal{B}_j$ is bounded above by the weight $L = \omega(\mathcal{L}) = O(\omega(MST(M[Q])))$ of the Hamiltonian path $\mathcal{L}$, for each index $j \in [\ell]$. We conclude that the weight $\omega(\mathcal{B})$ of the base edge set $\mathcal{B} = \bigcup_{j=1}^{\ell} \mathcal{B}_j$ satisfies $\omega(\mathcal{B}) = \omega(\bigcup_{j=1}^{\ell} \mathcal{B}_j) \leq \sum_{j=1}^{\ell} \omega(\mathcal{B}_j) \leq \ell \cdot O(\omega(MST(M[Q]))) = O(\ell) \cdot \omega(MST(M[Q]))$.

Corollary C.8 $\Psi(\mathcal{B}) = O(\ell)$.

D Analysis

D.1 Zombies and Adopters

In this appendix we prove a few basic properties of labels (zombies and incubators) used in our algorithm.

We say that a j-level bag v is an *attached bag* if it is unlabeled, and is adopted during the execution of Procedure $Process_j$. Observe that an attached bag v must be lonely, i.e., the cage $\mathcal{C}(v)$ does not contain any useful bags. In other words, all the non-empty bags in that cage are labeled as zombies.

When Procedure $Process_j$ "creates" an attached bag v, it labels $\gamma - 1$ of its immediate ancestors in $\mathcal{F}$ as zombies. We remark, however, that Procedure $Process_j$ does not label v itself as a zombie. Moreover, for v to become an attached bag, it must be unlabeled at the beginning of the j-level processing. Hence an attached bag v is *never* labeled by the algorithm. Thus for any zombie, there is (at least one) path in $\mathcal{F}$ of hop-distance at most $\gamma - 1$ leading down to an attached bag. (It will be shown in Lemma D.3 that there exists exactly one such path.)

Lemma D.1 *Fix an arbitrary index $j \in [\ell]$, and let v be a non-empty j-level bag. Then:*
(1) If v is not labeled as a zombie, then there is a path Υ_v of non-empty bags which are not labeled as zombies, leading down from v to some 1-level bag in $\mathcal{F}$.
(2) v cannot be labeled as both a zombie and an incubator.

Remark: The second assertion of this lemma implies that the distinction between useful and useless bags is well-defined.

Proof: The proof of both assertions of the lemma is by induction on j. The basis $j = 1$ is trivial.
Induction Step: Assume the correctness for all smaller values of $j, j \geq 2$, and prove it for j.

First, we prove the first assertion. Let v be a non-empty j-level bag which is not labeled as a zombie.

Suppose for contradiction that all non-empty children of v in $\mathcal{F}$ are labeled as zombies. Since v is not labeled as a zombie, it follows that all its zombie children are, in fact, disappearing zombies. By construction, these disappearing zombies become step-children of other j-level bags $u, u \neq v$. Moreover, by the second assertion of the induction hypothesis, none of these disappearing zombies can be labeled as an incubator. Hence, by construction, v cannot be a step-parent of any $(j-1)$-level bag. It follows that v is empty, a contradiction.

Therefore, there must be a non-empty child z of v that is not labeled as a zombie. By the first assertion of the induction hypothesis, there is a path $\Upsilon_z = (z = v_1, \ldots, v_k), k \geq 1$, of non-empty bags which are not labeled as zombies, leading down from z to some 1-level bag v_k in $\mathcal{F}$. The path $\Upsilon_v = (v = v_0, z = v_1, \ldots, v_k) = (v) \circ \Upsilon_z$ obtained by concatenating the singleton path (v) with Υ_z satisfies the conditions of the first assertion of the lemma.

Next, we prove the second assertion. Suppose that v is labeled as a zombie, and consider any path that leads down to a j'-level attached bag v'. Note that v' is a lonely bag, hence all the non-empty j'-level bags in the cage $\mathcal{C}(v')$ are useless (i.e., they are all labeled as zombies).

Suppose for contradiction that v has a non-empty child z that is not labeled as a zombie. Consider the path Υ_z that is guaranteed by the first assertion of the induction hypothesis. This path contains a j'-level non-empty bag z' which is not labeled as a zombie. However, z' belongs to $\mathcal{C}(v')$, yielding a contradiction. Therefore, all the non-empty children of v must be labeled as zombies, and by the induction hypothesis, they cannot be labeled as incubators. By construction (by the label assignment rules), no child u of v may become the initiator of any attachment (since an attachment initiator cannot be labeled as a zombie). Thus, v cannot be labeled as an incubator, and we are done. ∎

We use the next claim to prove Lemma D.3.

Claim D.2 *Fix an arbitrary index $j \in [\ell]$, and let v be a non-empty j-level bag. Then there is a useful $(j - \gamma + 1)$-level descendant u for v in $\mathcal{F}$.*

Proof: First, we argue that v has a useful j'-level descendant v' in $\mathcal{F}$, for some index $j-\gamma+1 \le j' \le j$. If v is useful, then we can simply take $v' = v$, $j' = j$. We henceforth assume that v is a zombie, and consider the path leading down to an appearing j'-level zombie v'. Since the hop-distance of this path is at most $\gamma - 1$, it follows that $j' \ge j - \gamma + 1$. Also, by construction, to become an attached bag the bag v' must be useful, as required. Consequently, Lemma D.1 implies that there is a path $\Upsilon_{v'}$ of useful bags, leading down from v' to some 1-level bag in $\mathcal{F}$. The claim follows. ∎

In the next lemma we show that a zombie cannot have "brothers" or "step-brothers".

Lemma D.3 *Fix an arbitrary index $j \in [\ell - 1]$, and let v be a non-empty $(j+1)$-level bag. If v has a zombie child z, then all its other children are empty and it has no step-children, i.e., $\mathcal{S}(v) = \{z\}, \mathcal{J}(v) = \emptyset$.*

Proof: Suppose for contradiction that v has a non-empty child u in addition to its zombie child z. Both z and u are j-level bags. Set $j' = j - \gamma + 1$. Let z' (respectively, u') be a useful j'-level descendant of z (resp., u) in $\mathcal{F}$ that is guaranteed by Claim D.2. Observe that the cage-ancestor of z' and u' is v, and so z' and u' belong to the same cage $\mathcal{C}(z') = \mathcal{C}(u')$. It follows that z' and u' are not lonely, and so they are safe and do not get adopted. Hence they do not become attached bags during the j'-level processing. More generally, note that the least common ancestor of z' and u' in $\mathcal{F}$ is v. Hence, for each index $i = j', j'+1, \ldots, j$, the i-level ancestors of z' and u' in $\mathcal{F}$ belong to the same cage, and so they are safe and do not get adopted, and do not become attached bags. By construction, any other useful i-level descendant of v must be safe as well, for each index $i = j', j'+1, \ldots, j$, hence it does not get adopted, and does not become an attached bag. However, again by construction, we conclude that the j-level ancestor z of z' in $\mathcal{F}$ will not become a zombie, a contradiction.

For the bag v to have a step-child, at least one of the children of v in $\mathcal{F}$ must be an adopter. However, we have showed that all children of v besides the zombie z are empty. Hence $\mathcal{S}(v) = \{z\}, \mathcal{J}(v) = \emptyset$. ∎

Lemma D.3 implies the following corollary.

Corollary D.4 *Fix an arbitrary index $j \in [\ell - 1]$, and let v be a (non-empty) j-level bag which is a disappearing zombie. Then the parent $\pi(v)$ of v in $\mathcal{F}$ is empty, and therefore is different than its step-parent v' (in other words, the bag that adopts v), i.e., $\pi(v) \neq v'$.*

Let w be a bag, and w' be an ancestor of w in $\mathcal{F}$. We say that w and w' are *identical bags* if $Q(w) = Q(w')$. (We remark that identical bags w and w' also satisfy $K(w) = K(w')$ and $B(w) = B(w')$.)

Lemma D.5 *Let $w \in \mathcal{F}_j$ be a disappearing zombie. (Hence $j \ge \gamma$.) Then there exists a unique useful descendant $\tilde{w} \in \mathcal{F}_{j-(\gamma-1)}$ of w, and it is an attached bag. The disappearing zombie $w = \tilde{w}^{(\gamma-1)}$ is identical to the attached bag $\tilde{w} = \tilde{w}^{(0)}$. More generally, each of the γ zombie bags $\tilde{w}^{(i)} \in \mathcal{F}_{j-(\gamma-1)+i}$ along the path between $\tilde{w} = \tilde{w}^{(0)}$ and $w = \tilde{w}^{(\gamma-1)}$, $i \in [0, \gamma-1]$, is identical to $\tilde{w} = \tilde{w}^{(0)}$. (All these γ bags are identical.)*

Proof: Since w is a disappearing zombie, there exists an attached bag $\tilde{w} \in \mathcal{F}_{j-(\gamma-1)}$, such that $w = \tilde{w}^{(\gamma-1)}$, i.e., $\tilde{w}$ is a $(j-(\gamma-1))$-level descendant of w in $\mathcal{F}$. For the bag $\tilde{w}$ to become an attached bag, it must be risky, and therefore lonely in its cage $C(\tilde{w})$. Hence all other $(j-(\gamma-1))$-level descendants of $\tilde{w}^{(\gamma)} = \pi(w)$ (and therefore, of w) are useless.

Next, we prove by induction on the index $i, i \in [0, \gamma-1]$, that each bag $\tilde{w}^{(i)}$ is identical to the attached bag $\tilde{w} = \tilde{w}^{(0)}$. The basis $i = 0$ is obvious, as $\tilde{w}$ is identical to itself.

Induction Step: Assume the correctness of the statement for all smaller values of $i, i \ge 1$, and prove it for i. By the induction hypothesis, the bag $\tilde{w}^{(i-1)}$ is identical to $\tilde{w}$, i.e., $Q(\tilde{w}^{(i-1)}) = Q(\tilde{w})$. Also, since the bag $\tilde{w}^{(i-1)}$ is a zombie child of $\tilde{w}^{(i)}$, Lemma D.3 yields $\mathcal{S}(\tilde{w}^{(i)}) = \{\tilde{w}^{(i-1)}\}, \mathcal{J}(\tilde{w}^{(i)}) = \emptyset$. By construction,

$$Q(\tilde{w}^{(i)}) \;=\; \bigcup_{z \in (\mathcal{S}(\tilde{w}^{(i)}) \cup \mathcal{J}(\tilde{w}^{(i)}))} Q(z) \;=\; Q(\tilde{w}^{(i-1)}) \;=\; Q(\tilde{w}).$$

We conclude that the bags $\tilde{w}^{(i)}$ and $\tilde{w}$ are identical. ∎

For a disappearing zombie $w \in \mathcal{F}_j, j \geq \gamma$, and an index i, such that $j - (\gamma - 1) \leq i \leq j$, we refer to the i-level descendant of w (which is, by Lemma D.5, identical to w) as the *i-level copy* of w. We also call it the *i-level copy* of $\tilde{w}$, where $\tilde{w} \in \mathcal{F}_{j-(\gamma-1)}$ is the unique non-empty $(j - (\gamma - 1))$-level descendant of w.

D.2 Number of Edges, Weight, and Running Time

We provide here complete proofs for the unproved statements from Section 3, regarding the number of edges and weight of the spanner $\tilde{G}$ built by Algorithm *LightSp*, and the running time of the algorithm.

Next, we provide a few auxiliary lemmas. They will be used in the analysis of the number of edges, weight and running time of our construction. We start with making the following observation.

Observation D.6 *Let f be a monotone non-decreasing convex functions, and let $n'_1, n'_2, \ldots, n'_\ell$ be a sequence of positive numbers that satisfy that $n'_j \leq \min\{n, n_j\}$, for each index $j \in [\ell]$. Then for each index $j \in [\ell]$, $f(n'_j) \leq \frac{c}{\rho^{j-1}} \cdot f(n)$. Moreover, for each index $1 \leq j < \log_\rho c + 1$, $f(n'_j) \leq f(n) < \frac{c}{\rho^{j-1}} \cdot f(n)$.*

Proof: Suppose first that $1 \leq j < \log_\rho c + 1$; in this case, we have $\frac{c}{\rho^{j-1}} > 1$, and so $n < \frac{c \cdot n}{\rho^{j-1}}$. It follows that $|n'_j| \leq n$, which yields $f(n'_j) \leq f(n) < \frac{c}{\rho^{j-1}} \cdot f(n)$. We henceforth assume that $\log_\rho c + 1 \leq j \leq \ell$. In this case, we have $\frac{c}{\rho^{j-1}} \leq 1$, and so $n \geq \frac{c \cdot n}{\rho^{j-1}}$. It follows that $n'_j \leq \frac{c \cdot n}{\rho^{j-1}}$. Also, the assumption that f is convex imply that $f(\frac{c \cdot n}{\rho^{j-1}}) \leq \frac{c}{\rho^{j-1}} \cdot f(n)$. We conclude that $f(n'_j) \leq f(\frac{c \cdot n}{\rho^{j-1}}) \leq \frac{c}{\rho^{j-1}} \cdot f(n)$. ∎

Observation D.6 implies the following corollary.

Corollary D.7 *For any monotone non-decreasing convex function f and any sequence $n'_1, n'_2, \ldots, n'_\ell$ as above: (1) $\sum_{j=1}^{\ell} f(n'_j) = O(f(n) \cdot \log_\rho(t/\epsilon))$, and (2) $\sum_{j=1}^{\ell} f(n'_j) \cdot \tau_j = O(\frac{f(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon) \cdot L$.*

Proof: We start proving the first assertion. Recall that $c = O(t/\epsilon)$. Hence

$$\begin{aligned}
\sum_{j=1}^{\ell} f(n'_j) &= \sum_{1 \leq j < \log_\rho c+1, j \in \mathbb{N}} f(n'_j) + \sum_{\log_\rho c+1 \leq j < \ell, j \in \mathbb{N}} f(n'_j) \\
&\leq (\log_\rho c + 1) \cdot f(n) + \sum_{\log_\rho c+1 \leq j < \log_\rho n, j \in \mathbb{N}} \frac{c}{\rho^{j-1}} \cdot f(n) \\
&\leq (\log_\rho c + 1) \cdot f(n) + \sum_{j=0}^{\infty} \frac{1}{\rho^j} \cdot f(n) \ \leq \ O(f(n) \cdot \log_\rho(t/\epsilon)).
\end{aligned}$$

Next, we prove the second assertion. For each $j \in [\ell]$, $\tau_j = 2 \cdot \rho^j \cdot \frac{L}{n} \cdot t \cdot (1 + \frac{1}{c}) = O(\rho^j \cdot \frac{L}{n} \cdot t)$. Hence,

$$\sum_{j=1}^{\ell} f(n'_j) \cdot \tau_j \ \leq \ \sum_{j=1}^{\ell} \frac{c}{\rho^{j-1}} \cdot f(n) \cdot O\left(\rho^j \cdot \frac{L}{n} \cdot t\right) \ = \ O\left(\frac{f(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon\right) \cdot L. \quad ∎$$

Recall that $|Q_j| \ \leq \ \min\{n, n_j\} = \min\left\{n, \frac{c \cdot n}{\rho^{j-1}}\right\}$. Hence, Corollary D.7 yields:

Corollary D.8 *For any monotone non-decreasing convex function f, we have (1) $\sum_{j=1}^{\ell} f(|Q_j|) = O(f(n) \cdot \log_\rho(t/\epsilon))$, and (2) $\sum_{j=1}^{\ell} f(|Q_j|) \cdot \tau_j = O(\frac{f(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon) \cdot L$.*

Next, we bound the number of edges in the spanner $\tilde{G} = (Q, \tilde{E})$ computed by Algorithm *LightSp*.

Lemma D.9 $|\tilde{E}| = O(SpSz(n) \cdot \log_\rho(t/\epsilon))$.

Proof: By construction, the edge set $\tilde{E}$ of $\tilde{G}$ is obtained as the union of the path-spanner $H = (Q, E_H)$, the base edge set $\mathcal{B}$, and all the j-level auxiliary spanners $\tilde{G}_j = (Q_j, \tilde{E}_j)$, $j \in [0, \ell]$, i.e., $\tilde{E} = E_H \cup \mathcal{B} \cup \bigcup_{j=0}^{\ell} \tilde{E}_j$.

The path-spanner H contains at most $O(n)$ edges, i.e., $|E_H| = O(n)$, and the graph $\tilde{G}_0 = (Q_0 = Q, \tilde{E}_0)$ contains at most $SpSz(n)$ edges, i.e., $|\tilde{E}_0| \le SpSz(|Q_0|) = SpSz(n)$. Also, as shown in Appendix C.2 (see Corollary C.6), the base edge set $\mathcal{B}$ contains at most n edges.

For each $j \in [\ell - \gamma]$, we have $\tilde{G}_j = G^*_j \cup \hat{G}_j$, where $G^*_j = (Q_j, E^*_j)$ and $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$. Also, for each $j \in [\ell - \gamma + 1, \ell]$, we have $\tilde{G}_j = G^*_j = (Q_j, E^*_j)$. Observe that $|E^*_j| \le SpSz(|Q_j|)$, for every $j \in [\ell]$. We also have $|\hat{E}_j| \le SpSz(|\hat{Q}_j|) \le SpSz(|Q_j|)$, for every $j \in [\ell - \gamma]$. It follows that $|\tilde{E}_j| = |E^*_j \cup \hat{E}_j| \le |E^*_j| + |\hat{E}_j| \le 2 \cdot SpSz(|Q_j|)$, for every index $j \in [\ell - \gamma]$, and $|\tilde{E}_j| = |E^*_j| \le SpSz(|Q_j|)$, for every index $j \in [\ell - \gamma + 1, \ell]$. Finally, recall that $SpSz(\cdot)$ is a monotone non-decreasing convex function. Consequently,

$$
\begin{aligned}
|\tilde{E}| &= |E_H| + |\mathcal{B}| + \sum_{j=0}^{\ell} |\tilde{E}_j| \;=\; |E_H| + |\mathcal{B}| + |\tilde{E}_0| + \sum_{j=1}^{\ell-\gamma} |E^*_j \cup \hat{E}_j| + \sum_{j=\ell-\gamma+1}^{\ell} |E^*_j| \\
&\le O(n) + SpSz(n) + 2 \cdot \sum_{j=1}^{\ell} SpSz(|Q_j|) \;\le\; O(n) + SpSz(n) + O(SpSz(n) \cdot \log_\rho(t/\epsilon)) \\
&= O(SpSz(n) \cdot \log_\rho(t/\epsilon)).
\end{aligned}
$$

(The last inequality follows from the first assertion of Corollary D.8.) ∎

The next lemma bounds the weight of $\tilde{G}$.

Lemma D.10 $\omega(\tilde{G}) = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^3/\epsilon) \cdot \omega(MST(M[Q]))$.

Proof: First, note that the weight $\omega(H)$ of the path-spanner H satisfies $\omega(H) = O(\rho \cdot \log_\rho n) \cdot L$. As shown in Section C.2 (see Corollary C.8), the weight $\omega(\mathcal{B})$ of the base edge set $\mathcal{B}$ satisfies $\omega(\mathcal{B}) = O(\log_\rho n) \cdot L$. Also, observe that the maximum edge weight in the graph $\tilde{G}_0$ is at most τ_0, and so

$$
\omega(\tilde{G}_0) \;\le\; |\tilde{E}_0| \cdot \tau_0 \;\le\; SpSz(|Q_0|) \cdot \tau_0 \;=\; SpSz(n) \cdot 2 \cdot \frac{L}{n} \cdot t \cdot \left(1 + \frac{1}{c}\right) \;=\; SpSz(n) \cdot O\left(\frac{L}{n} \cdot t\right).
$$

Next, observe that the maximum edge weight in the graph G^*_j (for every index $j \in [\ell]$) and the graph $\hat{G}_j$ (for every index $j \in [\ell - \gamma]$) is bounded above by j-level threshold τ_j. In other words, for every index $j \in [\ell]$, the maximum edge weight in the graph $\tilde{G}_j$ is bounded above by τ_j, and so

$$
\omega(\tilde{G}_j) \;\le\; |\tilde{E}_j| \cdot \tau_j \;\le\; 2 \cdot SpSz(|Q_j|) \cdot \tau_j.
$$

Finally, we have $\omega(\tilde{G}) \;=\; \omega(H) + \omega(\mathcal{B}) + \omega(\tilde{G}_0) + \sum_{j=1}^{\ell} \omega(\tilde{G}_j)$. It follows that

$$
\begin{aligned}
\omega(\tilde{G}) &\le O(\rho \cdot \log_\rho n) \cdot L + O(\log_\rho n) \cdot L + SpSz(n) \cdot O\left(\frac{L}{n} \cdot t\right) + \sum_{j=1}^{\ell} 2 \cdot SpSz(|Q_j|) \cdot \tau_j \\
&\le O(\rho \cdot \log_\rho n) \cdot L + SpSz(n) \cdot O\left(\frac{L}{n} \cdot t\right) + 4 \cdot O\left(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon\right) \cdot L \\
&= O\left(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon\right) \cdot L.
\end{aligned}
$$

(The last inequality follows from the second assertion of Corollary D.8.)

Finally, we bound the weight L of the path $\mathcal{L}$. Recall that Algorithm *LightSp* starts with computing an approximate MST T for $M[Q]$. The tree T is built in the following way.

First, we use Algorithm *BasicSp* to build a t-spanner $\mathcal{H}$ for $M[Q]$. Then we run Prim's Algorithm for $\mathcal{H}$ to get a t-approximate MST for $M[Q]$. It follows that $\omega(T) \leq t \cdot \omega(MST(M[Q]))$.

Recall that $L \leq 2 \cdot \omega(T)$, which implies that $L = O(t) \cdot \omega(MST(M[Q]))$, and we are done. ▮

Finally, we bound the running time of Algorithm *LightSp*.

Lemma D.11 *Algorithm LightSp can be implemented in time $O(SpTm(n) \cdot \log_\rho(t/\epsilon) + SpSz(n) \cdot \log n)$.*

Proof: Algorithm *LightSp* starts by building a t-approximate MST T for $M[Q]$. To this end, the algorithm first invokes Algorithm *BasicSp* to build a t-spanner $\mathcal{H}$ for $M[Q]$ with $SpSz(n)$ edges in time $O(SpTm(n))$. Then it runs Prim's Algorithm over $\mathcal{H}$ to get T in another time $O(SpSz(n) \cdot \log n)$.

The Hamiltonian path $\mathcal{L}$ is built in $O(n)$ time by computing the preorder traversal of T. The 1-spanner $\tilde{H}$ can be built in $O(n)$ time. The path-spanner H can be obtained from $\tilde{H}$ in another $O(n)$ time. Also, it is easy to see that the 0-level auxiliary spanner $\tilde{G}_0$ can be built within $O(SpTm(n))$ time.

In each iteration $j = 1, \ldots, \ell$, we spend $SpTm(|Q_j|)$ time to build the t-spanner G'_j. Then Algorithm *LightSp* prunes G'_j to obtain the graph G^*_j. Since there are at most $SpSz(|Q_j|)$ edges in G'_j and $SpSz(|Q_j|) \leq SpTm(|Q_j|)$, the graph G^*_j can be obtained from G'_j in $O(SpSz(|Q_j|)) = O(SpTm(|Q_j|))$ time (see Part I of Procedure $Process_j$ in Section 2.5). By similar considerations, the graph $\hat{G}_j$ can be built in time $O(SpTm(|\hat{Q}_j|)) = O(SpTm(|Q_j|))$; also, the attachment graph G_j can be built within another amount of $O(SpSz(|Q_j|)) = O(SpTm(|Q_j|))$ time (see Part II of Procedure $Process_j$). Updating the load indicators and counters of representatives requires another $O(SpSz(|Q_j|)) = O(SpTm(|Q_j|))$ time. By Corollary C.1 (see Appendix C.1), Procedure *Attach* runs in time that is linear in the number of vertices of the attachment graph G_j, i.e., in time $O(|Q_j|) = O(SpTm(|Q_j|))$. The number of non-empty bags in the forest $\mathcal{F}$ is $O(n)$. Each time one of these bags is processed, at most $O(\gamma)$ bags are labeled as zombies or incubators. Hence the total time required for labeling bags is $O(n \cdot \gamma) = O(n \cdot \log_\rho(t/\epsilon))$. It follows that the time needed to build all 2ℓ graphs $G^*_1, \hat{G}_1, \ldots, G^*_\ell, \hat{G}_\ell$ (and updating the load indicators and counters of the involved representatives accordingly), as well as executing Procedure *Attach* and labeling bags throughout all ℓ iterations is at most $\sum_{j=1}^{\ell} O(SpTm(|Q_j|)) + O(n \cdot \log_\rho(t/\epsilon)) \leq O(SpTm(n) \cdot \log_\rho(t/\epsilon))$. (Recall that $SpTm(\cdot)$ is a monotone non-decreasing convex function, and see the first assertion of Corollary D.8.)

On level j, determining which bags are crowded requires $O(|Q_j|)$ time. Hence, by Corollary D.8, in all ℓ levels altogether this step requires $\sum_{i=1}^{\ell} O(|Q_j|) = O(n \cdot \log_\rho(t/\epsilon))$ time.

Altogether, Algorithm *LightSp* takes time $O(SpTm(n) \cdot \log_\rho(t/\epsilon) + SpSz(n) \cdot \log n)$. ▮

D.3 Stretch and Diameter

The stretch and diameter of $\tilde{G}$ were analyzed rigorously in Section 3. Below we provide a few minor statements that were left unproved there.

D.3.1 Proof of Claim 3.2

Recall that $p \in Q(u)$, and y is a $(j-\gamma)$-level copy of u. Hence both points p and $r(y)$ belong to the k-level bag y, i.e., $p, r(y) \in Q(y)$. Consider the paths $\Pi_k(p)$ and $\Pi_k(r(y))$ in $\tilde{G}$ that are guaranteed by the induction hypothesis for y, having weight at most $\frac{1}{2} \cdot \mu_k = \frac{1}{2} \cdot \frac{\mu_j}{\rho^\gamma}$; all points of these two paths belong to $Q(y)$. The path $\Pi_k(p)$ (respectively, $\Pi_k(r(y))$) leads to a point $b_k(p)$ (resp., $b_k(r(y))$) in the base point set $B(y)$ of y. Recall that the spanner $\tilde{G}$ contains a path $P(y)$ which connects the base point set $B(y)$ via a simple path. Denote by $\Pi(b_k(p), b_k(r(y)))$ the sub-path of $P(y)$ between $b_k(p)$ and $b_k(r(y))$; by the triangle inequality, the weight of this path is at most $\delta_{\mathcal{L}}(b_k(p), b_k(r(y))) \leq \mu_k = \frac{\mu_j}{\rho^\gamma}$. We take $\Pi(p, r(y))$ to be the path that is obtained as the concatenation of the path $\Pi_k(p)$, the path $\Pi(b_k(p), b_k(r(y)))$, and the path $\Pi_k(r(y))$. (We assume that $\Pi(p, r(y))$ is a simple path. Otherwise we transform it into such.) It is easy to see that $\Pi(p, r(y))$ is a path between p and $r(y)$ in the spanner $\tilde{G}$ that has weight at most $2 \cdot \frac{\mu_j}{\rho^\gamma}$.

Moreover, all points of $\Pi(p, r(y))$ belong to $Q(y) = Q(u) \subseteq Q(v)$. Note that an attached bag $y \in \mathcal{F}_k$ was necessarily marked as risky by Procedure $Process_k$. Therefore, y is a small bag. Hence $|Q(y)| \leq \ell - 1$. It follows that $\Pi(p, r(y))$ consists of at most $\ell - 2$ edges, which completes the proof of Claim 3.2.

D.3.2 Proof of Corollary 3.3

Consider an arbitrary pair p, q of points in $Q(v)$, and let $\Pi_j(p)$ and $\Pi_j(q)$ be the paths in $\tilde{G}$ that are guaranteed by Lemma 3.1, having weight at most $\frac{1}{2} \cdot \mu_j$ and at most $3\ell = 3 \log_\rho n$ edges each. The path $\Pi_j(p)$ (respectively, $\Pi_j(q)$) leads to a point $b_j(p)$ (resp., $b_j(q)$) in the base point set $B(v)$ of v. The spanner $\tilde{G}$ contains the path-spanner H. Recall that for any pair $x, y \in Q$ of points, there is a path $\Pi_H(x, y)$ in the path-spanner H that has weight at most $\delta_{\mathcal{L}}(x, y)$ and $O(\log_\rho n + \alpha(\rho))$ edges. In particular, H contains a path $\Pi_H(b_j(p), b_j(q))$ between $b_j(p)$ and $b_j(q)$, having weight at most $\delta_{\mathcal{L}}(b_j(p), b_j(q)) \leq \mu_j$ and $O(\log_\rho n + \alpha(\rho))$ edges. Consider the path $\Pi(p, q) = \Pi_j(p) \circ \Pi_H(b_j(p), b_j(q)) \circ \Pi_j(q)$, obtained as the concatenation of the path $\Pi_j(p)$, the path $\Pi_H(b_j(p), b_j(q))$ and the path $\Pi_j(q)$. Note that $\Pi(p, q)$ is a path between p and q in the spanner $\tilde{G}$ that has weight at most $\frac{1}{2} \cdot \mu_j + \mu_j + \frac{1}{2} \cdot \mu_j = 2 \cdot \mu_j$, and at most $3 \log_\rho n + O(\log_\rho n + \alpha(\rho)) + 3 \log_\rho n = O(\log_\rho n + \alpha(\rho))$ edges.

D.3.3 Proof of Observation 3.5

Recall that $\Lambda(n)$ is an upper bound on the diameter of the auxiliary spanner, produced by Algorithm *BasicSp*. (See the statement of Theorem 1.3.)

Since u and v are non-empty j-level bags, it holds that $r(u), r(v) \in Q_j$. In addition, since G'_j is a t-spanner for Q_j with diameter at most $\Lambda(n)$, there is a t-spanner path Π in G'_j between $r(u)$ and $r(v)$ that consists of at most $\Lambda(n)$ edges. The fact that Π is a t-spanner path between $r(u)$ and $r(v)$ implies that the weight $\omega(\Pi)$ of Π satisfies $\omega(\Pi) \leq t \cdot \delta(r(u), r(v)) \leq \tau_j$. Clearly, the weight of each edge of Π is bounded above by $\omega(\Pi) \leq \tau_j$. By construction (see Section 2.2), G^*_j contains all the edges of G'_j with weight at most τ_j. It follows that all edges of Π belong to G^*_j, and we are done.

D.4 Degree Analysis

In this appendix we bound the maximum degree of our spanner $\tilde{G}$. Specifically, we will show that the degree of $\tilde{G}$ is $O(\Delta(n) \cdot \gamma + \rho)$. Recall that $\gamma = c_0 \cdot (\lceil \log_\rho t \rceil + \lceil \log_\rho c \rceil + 1)$, for some constant c_0; thus $\gamma = O(\log_\rho(t/\epsilon))$. In other words, we will get the desired degree bound of $O(\Delta(n) \cdot \log_\rho(t/\epsilon) + \rho)$.

As shown in Section C.2 (Corollary C.5), the base edges increase the degree bound by at most two units, and so we may disregard them in this analysis. We will also disregard the edges of the path-spanner H and the 0-level auxiliary spanner $\tilde{G}_0$, which together contribute $O(\Delta(n) + \rho)$ units to the degree bound.

Consider an index $j \in [\ell]$. Procedure $Process_j$ builds a spanner G'_j for the set Q_j of representatives of all non-empty j-level bags, including zombies. This spanner is then pruned to obtain the graph $G^*_j = (Q_j, E^*_j)$. If $j > \ell - \gamma$, then $\tilde{G}_j = G^*_j$. Otherwise, Procedure $Process_j$ constructs the subset $\hat{Q}_j$ of Q_j of useful representatives, which are not isolated in G^*_j. It then constructs the spanner $\check{G}_j = (\hat{Q}_j, \check{E}_j)$ for the set $\hat{Q}_j$, and prunes it to obtain the graph $\hat{G}_j = (\hat{Q}_j, \hat{E}_j)$. The union $\tilde{G}_j = (Q_j, \tilde{E}_j = E^*_j \cup \hat{E}_j)$ is the j-level auxiliary spanner. See Section 2.2 for more details.

Observe that the maximum degree $\Delta(\tilde{G}_0)$ of the 0-level auxiliary spanner is bounded above by $\Delta(n)$, and the maximum degree $\Delta(\tilde{G}_j)$ of the j-level auxiliary spanner, $j \in [\ell]$, is bounded above by $\Delta(|Q_j|) + \Delta(|\hat{Q}_j|) \leq 2 \cdot \Delta(n)$. For future reference we summarize this observation below.

Observation D.12 *For each index $j \in [0, \ell]$, $\Delta(\tilde{G}_j) = O(\Delta(n))$.*

We say that a j-level bag v is *active* if its representative $r(v)$ is not isolated in G^*_j. Otherwise it is called *passive*. Note that if v is passive, then its representative $r(v)$ *is not loaded* during the j-level processing, i.e., $load_i(r(v)) = 0$. On the other hand, for an active bag v, its representative *is loaded*, i.e., $load_i(r(v)) = 1$. Recall that v is a growing bag if $|\chi(v)| \geq 2$. (See Section 2.4.)

Recall also (see the beginning of Section C.2) that the relation step-parent - step-child among bags of $\mathcal{F}$ defines another forest $\mathcal{F}'$ over the same set of bags. We now define the forest $\hat{\mathcal{F}}$ over the same bag set in the following way. A bag v is a parent of u in $\hat{\mathcal{F}}$ if and only if $u \in \chi(v)$, i.e., u is either a surviving child or a joining step-child of v. We denote the parent-child relation in $\hat{\mathcal{F}}$ by $\hat{\pi}(\cdot)$. In particular, we write $v = \hat{\pi}(u)$. Note that a bag of level j in $\mathcal{F}$ has level j in $\hat{\mathcal{F}}$ as well.

Note that the forests $\mathcal{F}$ and $\hat{\mathcal{F}}$ are very similar. The only bags v that have step-parents (different from their parents) are disappearing zombies. We summarize this observation below.

Observation D.13 *For a bag $v \in \mathcal{F}_j, j \in [\ell - 1]$, which is not a disappearing zombie, $\pi(v) = \hat{\pi}(v)$.*

We say that a bag w is an $\mathcal{F}$-*descendant* (respectively, $\mathcal{F}$-*ancestor*) of the bag u if it is a descendant (resp., ancestor) of u in $\mathcal{F}$. Similarly, we say that a bag w is an $\hat{\mathcal{F}}$-*descendant* (respectively, $\hat{\mathcal{F}}$-*ancestor*) of the bag u if it is a descendant (resp., ancestor) of u in $\hat{\mathcal{F}}$.

Definition D.14 *For a positive integer parameter β, we say that a bag $v \in \mathcal{F}_j$ is* β-prospective*, if one of its β immediate $\hat{\mathcal{F}}$-ancestors $\hat{v}^{(1)} = \hat{\pi}(v), \hat{v}^{(2)} = \hat{\pi}(\hat{v}^{(1)}), \ldots, \hat{v}^{(\gamma)} = \hat{\pi}(\hat{v}^{(\beta)})$ is a growing bag. We also use the shortcut* prospective *for γ-prospective. For $j > \ell - \beta$, all bags $v \in \mathcal{F}_j$ are called β-prospective.*

Next, we argue that a small safe bag is necessarily prospective. Such a bag is either crowded, or a zombie, or an incubator. We start with the case of a crowded bag.

Lemma D.15 *Let $j \in [\ell]$ be an arbitrary index. Any crowded bag $v \in \mathcal{F}_j$ is prospective.*

Proof: The case $j > \ell - \gamma$ is trivial. We henceforth assume that $j \le \ell - \gamma$. Since v is a crowded j-level bag, its cage $\mathcal{C}(v)$ contains another useful j-level bag u. Note that u is crowded as well, and thus both v and u are safe. Let $w \in \mathcal{F}_k$ be the least common $\mathcal{F}$-ancestor of v and u. The index k satisfies $j + 1 \le k \le j + \gamma$. Write $v = v^{(0)}, u = u^{(0)}$, and consider the $(k - j)$ immediate $\mathcal{F}$-ancestors of v and u, $v^{(1)} = \pi(v^{(0)}), \ldots, v^{(k-j)} = \pi(v^{(k-j-1)}) = w$ and $u^{(1)} = \pi(u^{(0)}), \ldots, u^{(k-j)} = \pi(u^{(k-j-1)}) = w$, respectively. It can be shown by induction on $(k - j)$ that all these bags, except maybe w itself, are crowded and safe. Hence none of them is a zombie, and so for each index $i, 1 \le i \le k - j, v^{(i-1)} \in \mathcal{S}(v^{(i)}) \subseteq \chi(v^{(i)}), u^{(i-1)} \in \mathcal{S}(u^{(i)}) \subseteq \chi(u^{(i)})$. It follows that $v^{(k-j-1)}$ (respectively, $u^{(k-j-1)}$) is an $\hat{\mathcal{F}}$-ancestor of v (resp., u), and $v^{(k-j-1)}, u^{(k-j-1)} \in \mathcal{S}(w) \subseteq \chi(w)$. Hence w is the least common $\hat{\mathcal{F}}$-ancestor of v and u, and $|\chi(w)| \ge |\mathcal{S}(w)| \ge 2$. Thus w is a growing bag, and v is a prospective one. ▮

Next, we consider the case of a zombie bag.

Lemma D.16 *Let $j \in [\ell]$. A zombie bag $v \in \mathcal{F}_j$ is prospective.*

Proof: We only need to prove the assertion for $j \le \ell - \gamma$. By construction, there exists an attached bag $w = w^{(0)}$, which is an $\mathcal{F}$-descendant of v. By Lemma D.5, the bags $w = w^{(0)}, w^{(1)} = \pi(w^{(0)}), \ldots, w^{(i)} = v, \ldots, w^{(\gamma-1)}$ are identical, where $i \in [0, \gamma - 1]$ is some index. The bag $w^{(\gamma-1)}$ is a disappearing zombie.

Observe that $w = w^{(0)} \in \mathcal{F}_{j-i}$. There exists a bag $u \in \mathcal{F}_{j-i}$, so that the attachment $\mathcal{A}(u, v)$ took place during the $(j-i)$-level processing. The bag $u = u^{(0)}$ is the initiator of this attachment. Denote by $u^{(1)} = \pi(u^{(0)}), \ldots, u^{(\gamma)} = \pi(u^{(\gamma-1)})$ the γ immediate $\mathcal{F}$-ancestors of the initiator u. The bags $u^{(1)}, \ldots, u^{(\gamma-1)}$ are incubators, and $u^{(\gamma)}$ is the actual adopter. By Lemma D.1, neither of the incubator bags $u^{(1)}, \ldots, u^{(\gamma-1)}$ is a disappearing zombie. Hence, by Observation D.13, the actual adopter $u^{(\gamma)}$ is an $\hat{\mathcal{F}}$-ancestor of all the bags $u^{(0)}, u^{(1)}, \ldots, u^{(\gamma-1)}$. The initiator bag $u = u^{(0)}$ is non-empty, and thus the incubators $u^{(1)}, \ldots, u^{(\gamma-1)}$ are non-empty as well. Hence for each index $h \in [0, \gamma - 1], u^{(h)} \in \mathcal{S}(u^{(h+1)}) \subseteq \chi(u^{(h+1)})$.

Moreover, the attached bag $w = w^{(0)}$ is non-empty. Hence the zombie bags $w^{(1)}, w^{(2)}, \ldots, w^{(\gamma-1)}$ are non-empty as well, and for each index $h \in [0, \gamma - 2], w^{(h)} \in \mathcal{S}(w^{(h+1)}) \subseteq \chi(w^{(h+1)})$. Since the bags $w^{(0)}, w^{(1)}, \ldots, w^{(\gamma-2)}$ are not disappearing zombies, Observation D.13 implies that the disappearing zombie $w^{(\gamma-1)}$ is an $\hat{\mathcal{F}}$-ancestor of all these bags, and in particular, of $v = w^{(i)}$.

Finally, the disappearing zombie $w^{(\gamma-1)}$ is a joining step-child of the actual adopter $u^{(\gamma)}$, i.e., $w^{(\gamma-1)} \in \mathcal{J}(u^{(\gamma)}) \subseteq \chi(u^{(\gamma)})$. Hence $u^{(\gamma)}$ is an $\hat{\mathcal{F}}$-ancestor of v. By Lemma D.1, $u^{(\gamma-1)} \ne w^{(\gamma-1)}$. Hence $|\chi(u^{(\gamma)})| \ge 2$, i.e., $u^{(\gamma)}$ is a growing bag. Thus the bag v is prospective. ▮

A symmetric argument shows that an incubator bag is prospective as well.

Lemma D.17 *Let $j \in [\ell]$. An incubator bag $v \in \mathcal{F}_j$ is prospective.*

Lemmas D.15, D.16 and D.17 imply the following statement.

Corollary D.18 *Let $j \in [\ell]$. A small safe bag $v \in \mathcal{F}_j$ is prospective.*

For a positive integer parameter β, we say that a bag v is *β-safe-prospective* if either v or one of its β immediate $\hat{\mathcal{F}}$-ancestors is safe. (We remark that for v to be β-prospective, one of its β immediate ancestors must be growing, i.e., it is not enough for v to be a growing bag. This is not the case for a β-safe-prospective bag. That is, if v is safe, then it is β-safe-prospective.)

Denote $\kappa = \lceil \log_\rho t \rceil$ and $\eta = 2\kappa + 3$. Recall that $\gamma = c_0 \cdot (\kappa + \lceil \log_\rho c \rceil + 1)$. Next, we argue that any active small bag is either η-prospective or η-safe-prospective.

Lemma D.19 *Let $j \leq \ell - \eta$, and $u \in \mathcal{F}_j$ be an active small bag that is not η-prospective. Then u is η-safe-prospective.*

Proof: The bag u is active, and thus non-empty. Since u is not η-prospective, it follows that the η immediate $\hat{\mathcal{F}}$-ancestors of $u = \hat{u}^{(0)}$, namely, $\hat{u}^{(1)} = \hat{\pi}(\hat{u}^{(0)}), \ldots, \hat{u}^{(\eta)} = \hat{\pi}(\hat{u}^{(\eta-1)})$ are stagnating bags. Hence all these bags are identical to u, and moreover, they have the same representative as u, i.e., $r(u) = r(\hat{u}^{(0)}) = r(\hat{u}^{(1)}) = \ldots = r(\hat{u}^{(\eta)})$. (See Section 2.4.)

Suppose for contradiction that all these bags $\hat{u}^{(0)}, \hat{u}^{(1)}, \ldots, \hat{u}^{(\eta)}$ are risky. Note that for each index $i \in [0, \eta]$, if $\hat{u}^{(i)}$ is a zombie, then it must be safe. Hence the bags $\hat{u}^{(0)}, \hat{u}^{(1)}, \ldots, \hat{u}^{(\eta)}$ are not zombies, and thus useful. By Observation D.13, $\hat{u}^{(1)} = u^{(1)}, \ldots, \hat{u}^{(\eta)} = u^{(\eta)}$, i.e., the η immediate $\hat{\mathcal{F}}$-ancestors of u are its η immediate $\mathcal{F}$-ancestors.
Consider the j-level processing. Since u is active, the representative $r(u)$ of u is not isolated in $G^*_j = (Q_j, E^*_j)$. Moreover, the bag u is useful, hence $r(u) \in \hat{Q}_j$. If $r(u)$ is not isolated in $E^*_j(\hat{Q}_j)$, then it is not isolated in the j-level attachment graph $G_j = (\hat{Q}_j, \mathcal{E}_j), \mathcal{E}_j = E^*_j(\hat{Q}_j) \cup \hat{E}_j$. However, in this case $r(u)$ belongs to a star $S \in \Gamma_j$ in the star forest Γ_j formed by Procedure *Attach* (within Procedure *Process*$_j$). As a result the bag u becomes either an attachment initiator or an attached bag, and its parent $u^{(1)} = \hat{u}^{(1)}$ becomes an incubator or a zombie, respectively. In either case it becomes safe, a contradiction.

Hence $r(u)$ is isolated in $E^*_j(\hat{Q}_j)$. Since it is not isolated in E^*_j, it follows that there exists a zombie z, such that $r(z) \in Q_j \setminus \hat{Q}_j$ and the edge $(r(u), r(z)) \in E^*_j$. Hence $\delta(r(u), r(z)) \leq \tau_j$.

Also, the same argument applies for every index $h, h \in [j, j + (\eta - 1)]$, and not only for $h = j$. If $r(u^{(h-j)})$ is not isolated in the h-level attachment graph G_h, then the parent $u^{(h-j+1)} = \hat{u}^{(h-j+1)}$ of $u^{(h-j)}$ is safe. Hence in this case u is η-safe-prospective, a contradiction.

Therefore, from now on we assume that for all indices $h, h \in [j, j + (\eta - 1)]$, the representative $r(u) = r(\hat{u}^{(h-j)})$ is isolated in G_h.

Next, we argue that $r(u)$ is quite far from any useful representative on levels $j, j + 1, \ldots, j + (\eta - 1)$.

Claim D.20 *For any index $h, h \in [j, j + (\eta - 1)]$, and any useful bag $w \in \mathcal{F}_h, w \neq u^{(h-j)}$, it holds that*

$$\delta(r(u), r(w)) > \frac{\tau_h}{t}. \qquad (2)$$

Proof: Suppose for contradiction that for some index $h \in [j, j + (\eta - 1)]$ and a bag w as above, Equation (2) does not hold. Since $u^{(h-j)}$ and w are useful bags, their representatives belong to $\hat{Q}_h$, i.e., $r(u) = r(u^{(h-j)}), r(w) \in \hat{Q}_h$. Part II of Procedure *Process*$_j$ constructs a t-spanner $\check{G}_h = (\hat{Q}_h, \check{E}_h)$ for the metric $M[\hat{Q}_h]$ induced by $\hat{Q}_h$. Hence there exists a t-spanner path $\Pi = \Pi(r(u), r(w))$ in $\check{G}_h$ between $r(u)$ and $r(w)$. Since $\delta(r(u), r(w)) \leq \frac{\tau_h}{t}$, it follows that $\omega(\Pi) \leq \tau_h$. Therefore all edges of Π also have weight at most τ_h. Hence the path Π is contained in the pruned graph $\hat{G}_h = (\hat{Q}_h, \hat{E}_h)$. Moreover, $\hat{E}_h \subseteq \mathcal{E}_h$, where $\mathcal{E}_h$ is the edge set of the h-level attachment graph $G_h = (\hat{Q}_h, \mathcal{E}_h)$. Hence $\Pi \subseteq \mathcal{E}_h$ as well. Therefore $r(u) = r(u^{(h-j)})$ is not isolated in G_h, a contradiction. ∎

Now we continue to prove Lemma D.19.

Consider again the zombie $z \in \mathcal{F}_j$, such that $\delta(r(u), r(z)) \le \tau_j$. Let $i, j-(\gamma-1) \le i < j$, be the index such that an identical descendant y of z became an attached bag during the i-level processing. More specifically, during the i-level processing the bag y was attached to another i-level bag v by an attachment $\mathcal{A}(v, y)$. As a result of this attachment, the $(\gamma-1)$ immediate $\mathcal{F}$-ancestors of $y = y^{(0)}$, i.e., $y^{(1)} = \pi(y^{(0)}), y^{(2)} = \pi(y^{(1)}), \ldots, y^{(j-i)} = z, \ldots, y^{(\gamma-1)} = \pi(y^{(\gamma-2)})$, are labeled as zombies. Since none of these bags except $y^{(\gamma-1)}$ are disappearing zombies, Observation D.13 implies that $y^{(1)} = \hat{\pi}(y^{(0)}) = \hat{y}^{(1)}, \ldots, y^{(\gamma-1)} = \hat{\pi}(\hat{y}^{(\gamma-2)}) = \hat{y}^{(\gamma-1)}$. Moreover, all these bags have the same representative $r(y) = r(z) = r(y^{(1)}) = \ldots = r(y^{(\gamma-1)})$. The bag $v \in \mathcal{F}_i$ is the initiator of the attachment $\mathcal{A}(v, y)$. The $(\gamma-1)$ immediate $\mathcal{F}$-ancestors $v^{(1)}, v^{(2)}, \ldots, v^{(\gamma-1)}$ of $v = v^{(0)}$ are labeled as a result of the attachment $\mathcal{A}(v, y)$ as incubators. By Lemma D.1, none of them is a zombie, and, in particular, none of them is a disappearing zombie. Thus, again by Observation D.13, $v^{(1)} = \hat{\pi}(v^{(0)}) = \hat{v}^{(1)}, \ldots, v^{(\gamma-1)} = \hat{\pi}(\hat{v}^{(\gamma-2)}) = \hat{v}^{(\gamma-1)}$.

Denote $x = v^{(j-i)} \in \mathcal{F}_j$. The representing edge of the attachment $\mathcal{A}(v, y)$ is the edge $(r(v), r(y))$. Hence $\delta(r(v), r(y)) \le \tau_i$. The bag $x \in \mathcal{F}_j$ is an incubator. Hence it is safe. On the other hand, the bag $u \in \mathcal{F}_j$ is risky. Hence $u \ne x$. Therefore, $Q(u) \cap Q(x) = \emptyset$. (Recall that point sets of two distinct j-level bags are disjoint.)

Denote $k = j + \kappa + 1$. Let $x' = \hat{x}^{(k-j)}$ denote the k-level $\hat{\mathcal{F}}$-ancestor of the bag x. By construction, $Q(x) \subseteq Q(x')$. Since $\kappa + 1 \le \eta - 1$, the bag $u' = \hat{u}^{(k-j)}$ is identical to u. (Since the $\eta - 1$ immediate $\mathcal{F}$-ancestors, and $\hat{\mathcal{F}}$-ancestors, are all identical to u.) Both bags u' and x' are non-empty k-level bags, and $Q(u') = Q(u), Q(x') \supseteq Q(x)$, and $Q(u) \cap Q(x) = \emptyset$. Hence $Q(u') \ne Q(x')$, and thus $Q(u') \cap Q(x') = \emptyset$, and u' and x' are distinct bags. (See Figure 2 for an illustration.)

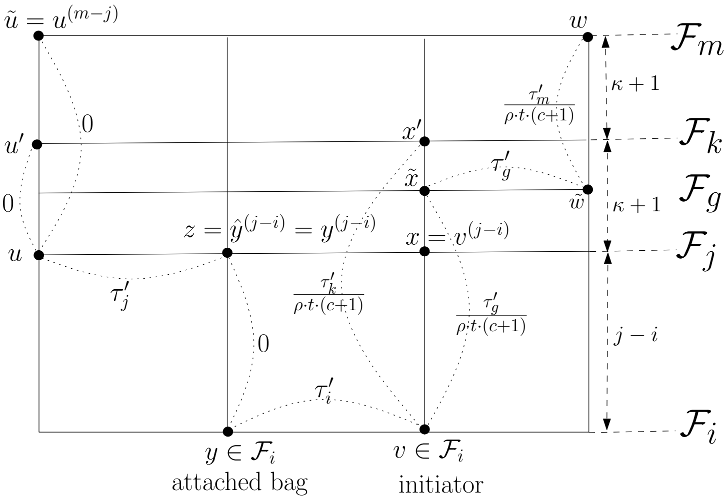

Figure 2: A schematic illustration for the proof of Lemma D.19. Expressions that appear next to dotted lines reflect upper bounds on distances between the representatives of their endpoints. For example, $r(\tilde{u}) = r(u)$, and thus 0 appears next to the dotted line that connects $\tilde{u}$ and u. Similarly, $\delta(r(v), r(x')) \le \frac{\tau_k}{\rho \cdot t \cdot (c+1)}$, and thus $\frac{\tau_k}{\rho \cdot t \cdot (c+1)}$ appears next to the dotted line that connects v and x'.

The analysis splits into two cases now, depending on whether the bag x' is a zombie or not. We start with the case that it is not a zombie. (It may be an attached bag.) By definition, x' is useful. The representative $r(v)$ of v belongs to $Q(v) \subseteq Q(x) \subseteq Q(x')$. Hence $r(v), r(x') \in Q(x')$. By Corollary 3.3, $\delta(r(v), r(x')) \le 2 \cdot \mu_k = \frac{\tau_k}{\rho \cdot t \cdot (c+1)}$. Recall that $r(u') = r(u)$ and $r(z) = r(y)$. Hence $\delta(r(u'), r(y)) = \delta(r(u), r(z)) \le \tau_j$. By the triangle inequality,

$$\begin{aligned} \delta(r(u'), r(x')) &\le \delta(r(u'), r(y)) + \delta(r(y), r(v)) + \delta(r(v), r(x')) \\ &\le \tau_j + \tau_i + \frac{\tau_k}{\rho \cdot t \cdot (c+1)} = \tau_k \cdot \left(\frac{1}{\rho^{k-j}} + \frac{1}{\rho^{k-i}} + \frac{1}{\rho \cdot t \cdot (c+1)} \right). \end{aligned}$$

Recall that $k - j = \kappa + 1 = \lceil \log_\rho t \rceil + 1$. Also, $i \leq j - 1$, and thus $k - i \geq \log_\rho t + 2$. Since $\rho \geq 2$ and $c \geq 1$, it follows that

$$\begin{aligned} \delta(r(u'), r(x')) &\leq \tau_k \cdot \left(\frac{1}{\rho^{\log_\rho t + 1}} + \frac{1}{\rho^{\log_\rho t + 2}} + \frac{1}{\rho \cdot t \cdot (c+1)} \right) \\ &= \tau_k \cdot \left(\frac{1}{\rho \cdot t} + \frac{1}{\rho^2 \cdot t} + \frac{1}{\rho \cdot t \cdot (c+1)} \right) \leq \frac{\tau_k}{t} \cdot \left(\frac{1}{2} + \frac{1}{4} + \frac{1}{4} \right) = \frac{\tau_k}{t}. \end{aligned}$$

The bag x' is useful, and $r(u) = r(u')$. Also, $k = j + (\kappa + 1) \in [j, j + (\eta - 1)]$, contradicting Equation (2).

Next, we turn to the case that x' is a zombie (but not an attached bag). There exists an index $g, j < g < k$, so that an $\hat{\mathcal{F}}$-descendant $\tilde{x}$ of x' (and an $\hat{\mathcal{F}}$-ancestor of x and v) became an attached bag. Hence there exists an initiator $\tilde{w} \in \mathcal{F}_g$, so that the attachment $\mathcal{A}(\tilde{w}, \tilde{x})$ occurred during the g-level processing. The representing edge of this attachment is $(r(\tilde{w}), r(\tilde{x}))$. It follows that $\delta(r(\tilde{w}), r(\tilde{x})) \leq \tau_g$. Denote $m = j + 2\kappa + 2 = j + (\eta - 1)$. Let $w \in \mathcal{F}_m$ denote the m-level $\mathcal{F}$-ancestor of $\tilde{w}$. Observe that $m - g \leq m - j = 2\kappa + 2 < \gamma$. (The constant c_0 should be set as $c_0 \geq 3$ for this to hold.) Hence the bag w is labeled as a result of the attachment $\mathcal{A}(\tilde{w}, \tilde{x})$ as an incubator. All the $(\gamma - 1)$ immediate $\mathcal{F}$-ancestors of $\tilde{w}$ are incubators, and thus, by Lemma D.1, they are not zombies. In particular, none of them is a disappearing zombie. Hence, by Observation D.13, for each index $h, g < h \leq m$, the h-level $\hat{\mathcal{F}}$-ancestor of $\tilde{w}$ is the same bag as the h-level $\mathcal{F}$-ancestor of $\tilde{w}$.

Hence the bag w is safe. On the other hand the m-level ancestor $u^{(m-j)} = u^{(\eta-1)}$ of u is risky, and so $u^{(m-j)} \neq w$. Denote $\tilde{u} = u^{(m-j)}$. Since $\tilde{x}$ is an g-level $\hat{\mathcal{F}}$-ancestor of v, it follows that $r(v), r(\tilde{x}) \in Q(\tilde{x})$. Hence, by Corollary 3.3, $\delta(r(v), r(\tilde{x})) \leq \frac{\tau_g}{\rho \cdot t \cdot (c+1)}$. Similarly, as w is an $\hat{\mathcal{F}}$-ancestor of $\tilde{w}$, and $w \in \mathcal{F}_m$, it follows that $\delta(r(w), r(\tilde{w})) \leq \frac{\tau_m}{\rho \cdot t \cdot (c+1)}$. Also, $r(\tilde{u}) = r(u)$, and $r(z) = r(y)$. Hence $\delta(r(\tilde{u}), r(y)) = \delta(r(u), r(z)) \leq \tau_j$. By the triangle inequality,

$$\begin{aligned} \delta(r(\tilde{u}), r(w)) &\leq \delta(r(\tilde{u}), r(y)) + \delta(r(y), r(v)) + \delta(r(v), r(\tilde{x})) + \delta(r(\tilde{x}), r(\tilde{w})) + \delta(r(\tilde{w}), r(w)) \\ &\leq \tau_j + \tau_i + \frac{\tau_g}{\rho \cdot t \cdot (c+1)} + \tau_g + \frac{\tau_m}{\rho \cdot t \cdot (c+1)} \\ &= \tau_m \cdot \left(\frac{1}{\rho^{m-j}} + \frac{1}{\rho^{m-i}} + \frac{1}{\rho^{m-g} \cdot \rho \cdot t \cdot (c+1)} + \frac{1}{\rho^{m-g}} + \frac{1}{\rho \cdot t \cdot (c+1)} \right) \\ &\leq \tau_m \cdot \left(\frac{1}{\rho^{2 \cdot \log_\rho t + 2}} + \frac{1}{\rho^{2 \cdot \log_\rho t + 3}} + \frac{1}{\rho^{\log_\rho t + 2} \cdot \rho \cdot t \cdot (c+1)} + \frac{1}{\rho^{\log_\rho t + 2}} + \frac{1}{\rho \cdot t \cdot (c+1)} \right) \\ &= \tau_m \cdot \left(\frac{1}{\rho^2 \cdot t^2} + \frac{1}{\rho^3 \cdot t^2} + \frac{1}{\rho^3 \cdot t^2 \cdot (c+1)} + \frac{1}{\rho^2 \cdot t} + \frac{1}{\rho \cdot t \cdot (c+1)} \right) \\ &\leq \frac{\tau_m}{t} \cdot \left(\frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{4} + \frac{1}{4} \right) < \frac{\tau_m}{t}. \end{aligned}$$

Therefore, $\delta(r(\tilde{u}), r(w)) < \frac{\tau_m}{t}$. The bag w is useful and $r(\tilde{u}) = r(u)$. Also, $\tilde{u} \in \mathcal{F}_m$, and $m \in [j, j + (\eta - 1)]$. Hence this is also a contradiction to Equation (2).

It follows that at least one of the bags $\hat{u}^{(0)}, \hat{u}^{(1)}, \ldots, \hat{u}^{(\eta)}$ is safe, and thus u is η-safe-prospective. This completes the proof of Lemma D.19. ▮

Next, we combine Corollary D.18 and Lemma D.19 to conclude that any active small bag is $(\gamma + \eta)$-prospective.

Lemma D.21 *Let $j \in [\ell]$. Any active small bag $v \in \mathcal{F}_j$ is $(\gamma + \eta)$-prospective.*

Proof: If $j > \ell - (\gamma + \eta)$ then the assertion is trivial. So we henceforth assume that $j \leq \ell - (\gamma + \eta)$. If v is η-prospective, then we are done. Otherwise, by Lemma D.19, it is η-safe-prospective. In other words, for some index $i, j \leq i \leq j + \eta$, the i-level $\hat{\mathcal{F}}$-ancestor $\tilde{v}$ of v is safe. Since v is not η-prospective, the bags v and $\tilde{v}$ are identical, and thus $\tilde{v}$ is small. Corollary D.18 implies that $\tilde{v}$ is γ-prospective. It follows that v is $(\gamma + \eta)$-prospective. ▮

The large (respectively, small) counter of a point $p \in Q$ grows during the j-level processing (for some index $j \in [\ell]$) if p is a representative of some large (resp., small) j-level bag v, and if p is not isolated in the j-level auxiliary graph $\tilde{G}_j$. (See the description in Section 2.4.)

Consider a small bag $v \in \mathcal{F}_j$, for some index $j \in [\ell]$. All its $\hat{\mathcal{F}}$-descendants are small as well. Also, for a point $p \in Q(v)$, and an index $i, 1 \leq i \leq j$, the i-level host bag $v_i(p)$ is an $\hat{\mathcal{F}}$-descendant of v. Hence any point $p \in Q(v)$ belongs only to small i-level bags, for $i, 1 \leq i \leq j$. Hence $CTR_j(p) = 0$. We summarize this argument in the following observation.

Observation D.22 *Let $j \in [\ell]$ be an index, and $v \in \mathcal{F}_j$ be a small bag. Then for any point $p \in Q(v)$, it holds that $CTR_j(p) = 0$.*

Next, consider a large bag $v \in \mathcal{F}_j$, for some index $j \in [\ell]$. Only small bags may be labeled as zombies. Hence v is useful. Observation D.13 implies that it will not have a step-parent, i.e., $\hat{\pi}(v) = \pi(v)$ and $v \in \mathcal{S}(\pi(v))$. Also, we have by construction $K(\pi(v)) \supseteq K(v)$. Consider now $\pi(v)$, and notice that $Q(\pi(v)) \supseteq Q(v)$. Hence $\pi(v)$ will be large as well, and we can apply this argument for $\pi(v)$.

Observation D.23 *Let $j \in [\ell]$, and v be a large j-level bag. For every $\hat{\mathcal{F}}$-ancestor v' of v, $K(v') \supseteq K(v)$.*

Next, we argue that the large counter of any point $p \in Q$ is at most 1.
We say that a large bag $v \in \mathcal{F}_j$ is *atomically large*, for some index $j \in [\ell]$, if all its extended children (if any) are small. In particular, all large 1-level bags are atomically large. We will use this definition in the proof of the following lemma.

Lemma D.24 *For each point $p \in Q$, $CTR_\ell(p) \leq 1$.*

Proof: Suppose for contradiction that there is a point $p \in Q$, with $CTR_\ell(p) \geq 2$, and let $j, j \in [\ell - 1]$, be the index for which $CTR_j(p) = 1, CTR_{j+1}(p) = 2$. By construction, p is the representative of its $(j+1)$-level host bag $v_{j+1}(p)$. Moreover, the j-level and $(j+1)$-level host bags $v_j(p)$ and $v_{j+1}(p)$ of p, respectively, are both large. If $v_j(p)$ is atomically large, set $v = v_j(p)$. Otherwise, set v to be an arbitrary $\hat{\mathcal{F}}$-descendant of $v_j(p)$ that is atomically large. (Note that p may not belong to $Q(v)$.) We have $v \in \mathcal{F}_g$, where $1 \leq g \leq j \leq \ell - 1$. By Observation 2.1, $|K(v)| \geq \ell$. Write $K(v) = \{q_1, \ldots, q_k\}$, where $k \geq \ell$.

Next, we argue that

$$CTR_{g-1}(q_1) \;=\; CTR_{g-1}(q_2) \;=\; \ldots \;=\; CTR_{g-1}(q_k) \;=\; 0. \tag{3}$$

Equation (3) clearly holds if $g = 1$. For $g \geq 2$, all the extended children $z \in \chi(v)$ of v are small by definition. Also, by construction, $Q(v) = \bigcup_{z \in \chi(v)} Q(z)$. Therefore, by Observation D.22, we have $CTR_{g-1}(p) = 0$, for each point $p \in Q(v)$. Equation (3) now follows as $K(v) \subseteq Q(v)$.

Consider the $j - g + 1$ immediate $\hat{\mathcal{F}}$-ancestors of $v = v^{(0)}$, i.e., $v^{(1)}, \ldots, v^{(j-g)} = v_j(p), v^{(j-g+1)} = v_{j+1}(p)$. Observation D.23 implies that for each index $i \in [j - g + 1]$, $K(v^{(i)}) \supseteq K(v) = \{q_1, \ldots, q_k\}$. For each index $i \in [g, j]$, at most one point from $K(v) = \{q_1, \ldots, q_k\}$ is appointed as a representative during the i-level processing; that point is the only one from $K(v)$ whose large counter increases during the i-level processing. Since $|K(v)| \geq \ell > j - g + 1$, there must be at least one point $q \in K(v)$, with $CTR_j(q) = 0$. Also, $q \in K(v) \subseteq K(v_{j+1}(p))$, and the point p is the representative of $v_{j+1}(p)$. Recall that for any large $(j+1)$-level bag u, Algorithm *LightSp* sets its representative $r(u)$ to be a point $\tilde{p} \in K(u)$ with the smallest large counter $CTR_j(\tilde{p})$. Hence $CTR_j(p) \leq CTR_j(q) = 0$, a contradiction. ∎

Next, we turn to analyzing single counters of points $p \in Q$. Recall that for a point $p \in Q$ and an index $j \in [\ell]$, $single\_ctr_j(p)$ counts the number of indices $i \in [j]$ such that the point p is not isolated in the i-level auxiliary graph $\tilde{G}_i$ *and* its host bag $v_i(p)$ satisfies $Q(v_i(p)) = \{p\}$.

Lemma D.25 *For any point $p \in Q$, $single\_ctr_\ell(p) \leq \gamma + \eta$.*

Proof: For a point $p \in Q$, let $i \in [\ell]$ be the smallest index such that p is not isolated in the i-level auxiliary graph $\tilde{G}_i$, *and* the host bag $v_i(p)$ of p satisfies $Q(v_i(p)) = \{p\}$. If such an index does not exist, then obviously $single\_ctr_\ell(p) = 0$. We henceforth assume that the index i exists, and write $v = v_i(p)$. Notice that $ctr_1(p) = \ldots = ctr_{i-1}(p) = 0$, and so $single\_ctr_1(p) = \ldots = single\_ctr_{i-1}(p) = 0$. By definition, the bag v is active. Thus, Lemma D.21 implies that v is $(\gamma + \eta)$-prospective. It follows that there is an index $k, 1 \le k \le (\gamma + \eta)$, such that the $(i+k)$-level $\hat{\mathcal{F}}$-ancestor $\hat{v}^{(k)}$ of v is a growing bag. Therefore, the bag $\hat{v}^{(k)}$ contains at least one point, in addition to p. Moreover, each $\hat{\mathcal{F}}$-ancestor of $\hat{v}^{(k)}$ also contains at least one point, in addition to p. Hence $single\_ctr_{i+k}(p) = single\_ctr_{i+k+1}(p) = \ldots = single\_ctr_\ell(p)$. In other words, the single counter of p may be incremented only during the h-level processing, for $h = i, i+1, \ldots, i+(k-1)$, i.e., for at most $k \le \gamma + \eta$ times. Therefore $single\_ctr_\ell(p) \le \gamma + \eta$. ∎

Next, we argue that $plain\_ctr_\ell(p)$ is small as well. Recall that for a point $p \in Q$ and an index $j \in [\ell]$, $plain\_ctr_j(p)$ is the number of indices $i, 1 \le i \le j$, such that p serves as a representative of an i-level small bag v with $|Q(v)| \ge 2$, *and* p is not isolated in $\tilde{G}_i$.

Lemma D.26 *Let $v \in \mathcal{F}_j$ be a growing small bag, for some index $j \in [2, \ell]$. Then the kernel set $K(v)$ of v contains at least two points p, q with $plain\_ctr_{j-1}(p) = plain\_ctr_{j-1}(q) = 0$.*

Proof: Let i be the minimum index such that there exists an i-level growing bag v. By definition, $i \ge 2$. Also, there are no growing h-level bags, for any index $1 \le h \le i-1$.
The proof is by induction on j.
Basis: $j = i$. Consider a growing small bag $v \in \mathcal{F}_i$. Since it is growing, we have $|\chi(v)| \ge 2$. Let $u, w \in \chi(v)$ be two distinct extended children of v. Since v is small, both u and w are small too. As there are no growing bags of level $h, 1 \le h \le i-1$, it follows that there exist 1-level bags u' and w' such that u' is identical to u and w' is identical to w. Moreover, all bags on the path in $\mathcal{F}$ that connects u' to u (respectively, w' to w) are identical to both of them and have the same representative $r(u)$ (resp., $r(w)$).

If the point set $Q(u)$ of u contains just one single point, i.e., $Q(u) = \{r(u)\}$, then $plain\_ctr_{i-1}(r(u)) = 0$. (Its single counter $single\_ctr_{i-1}(r(u))$ might be larger, but it is taken care of separately.) Otherwise, $|Q(u)| \ge 2$. Hence $Q(u) \setminus \{r(u)\}$ contains at least one additional point $p(u), p(u) \ne r(u)$. This point satisfies $plain\_ctr_{i-1}(p(u)) = 0$. In either case the bag u contains a point $q(u) \in Q(u)$, such that $plain\_ctr_{i-1}(q(u)) = 0$. The same is true for w. Moreover, $Q(u), Q(w) \subseteq Q(v)$ and $Q(u) \cap Q(w) = \emptyset$, and so $q(u)$ and $q(w)$ are two distinct points in $Q(v)$. Hence $Q(v)$ contains two distinct points $q(u), q(w)$ such that $plain\_ctr_{i-1}(q(u)) = plain\_ctr_{i-1}(q(w)) = 0$. By Observation 2.1, since v is a small bag, $K(v) = Q(v)$, and we are done.
Induction Step: Assume the correctness of the statement for all smaller values of $j, j \ge i+1$, and prove it for j. For a growing small bag $v \in \mathcal{F}_j$, there exist two distinct small extended children $u, w \in \chi(v) \subseteq \mathcal{F}_{j-1}$. Either u is growing, or there exists an extended child $u^{(-1)}$ of $u = u^{(0)}$, which is identical to u. The same argument applies to $u^{(-1)}$. Hence, there is a sequence of bags $u = u^{(0)}, u^{(-1)}, \ldots, u^{(-h)}$, for some index $h \le j-2$, with $u^{(-k+1)} = \hat{\pi}(u^{(-k)})$, for each $k \in [h]$. The bag $\tilde{u} = u^{(-h)} \in \mathcal{F}_{j-h-1}$ is either growing or belongs to $\mathcal{F}_1$. Moreover, all bags $u = u^{(0)}, u^{(-1)}, \ldots, \tilde{u} = u^{(-h)}$ are identical.

If the point set $Q(u)$ of u contains just one single point $r(u)$, then $plain\_ctr_{j-1}(r(u)) = 0$. (Even though its single counter may be larger.)

Otherwise $Q(u) \setminus \{r(u)\}$ contains at least one additional point $p(u), p(u) \ne r(u)$. If $\tilde{u} \in \mathcal{F}_1$, then $plain\_ctr_{j-1}(p(u)) = 0$. Otherwise $j-h-1 \ge 2$ and $\tilde{u}$ is a growing bag. By the induction hypothesis, $\tilde{u}$ contains at least two points $p_1(u), p_2(u)$ with $plain\_ctr_{j-h-2}(p_1(u)) = plain\_ctr_{j-h-2}(p_2(u)) = 0$. One of these points may become the representative of $\tilde{u}$ (and, consequently, of all the bags $u^{(-h+1)}, u^{(-h+2)}, \ldots, u^{(0)} = u$ that are identical to $\tilde{u}$), and, as a result its $(j-1)$-level plain counter may become positive. However, the other one will have plain counter equal to 0 on all levels $j-h-1, j-h, \ldots, j-1$. Thus either $plain\_ctr_{j-1}(p_1(u)) = 0$ or $plain\_ctr_{j-1}(p_2(u)) = 0$ must hold. Hence in both cases $Q(u) \setminus \{r(u)\}$ contains at least one point $q(u)$ with $plain\_ctr_{j-1}(q(u)) = 0$.

We showed that in all cases $Q(u)$ contains at least one point $q(u)$ with $plain\_ctr_{j-1}(q(u)) = 0$. Similarly, the bag w also contains a point $q(w) \in Q(w)$ with $plain\_ctr_{j-1}(q(w)) = 0$. Since $u, w \in \chi(v)$, it follows that $q(u), q(w) \in Q(v)$. Moreover, $Q(u) \cap Q(w) = \emptyset$, and so $q(u)$ and $q(w)$ are distinct. By Observation 2.1, since v is a small bag, $K(v) = Q(v)$, which completes the proof. ∎

Next, we provide an upper bound for plain counters of points in Q.

Lemma D.27 *For any point $p \in Q$, $plain\_ctr_\ell(q) \leq \gamma + \eta$.*

Proof: Consider a point $q \in Q$. Suppose that $plain\_ctr_\ell(q) > 0$. Let $i \in [\ell]$ be the smallest index such that the plain counter of q is incremented during the i-level processing, i.e., $plain\_ctr_{i-1}(q) = 0, plain\_ctr_i(q) = 1$. It follows that the i-level host bag $v = v_i(p)$ is active and small, and also $q = r(v)$. Moreover $|Q(v)| \geq 2$. Denote $\beta = \gamma + \eta$. If $i > \ell - \beta$ then the plain counter of q is incremented at most β times, on levels $i, i+1, \ldots, \ell$. Hence in this case $plain\_ctr_\ell(q) \leq \beta$, as required. Otherwise, let j denote the smallest level of an $\hat{\mathcal{F}}$-ancestor u of v such that u is a growing bag. By Lemma D.21, j is well-defined, with $i < j \leq i + \beta \leq \ell$. Consider the $j - i$ immediate $\hat{\mathcal{F}}$-ancestors of $v = \hat{v}^{(0)}$, i.e., $\hat{v}^{(1)} = \hat{\pi}(\hat{v}^{(0)}), \ldots, \hat{v}^{(j-i)} = u = \hat{\pi}(\hat{v}^{(j-i-1)})$. The bags $\hat{v}^{(1)}, \ldots, \hat{v}^{(j-i-1)}$ are identical to v, and have the same representative $r(v) = q$. If u is a large bag then all its $\hat{\mathcal{F}}$-ancestors are large as well. Also, $p \in Q(u)$, and for all indices $k \geq j$, p belongs to the point set of the k-level $\hat{\mathcal{F}}$-ancestor of u. Hence the plain counter of p is not incremented during the k-level processing, for all $k \geq j$.

Suppose now that u is small. Since it is growing, by Lemma D.26, its kernel set $K(u)$ contains at least two points p, p' with plain counter zero, i.e., $plain\_ctr_{j-1}(p) = plain\_ctr_{j-1}(p') = 0$. On the other hand, $plain\_ctr_{j-1}(q) \geq plain\_ctr_i(q) = 1$. Hence q is not the representative of u. More generally, we have the following claim.

Claim D.28 *Let $w = v_k(q)$ be the k-level host bag of q, for some index $k \geq j$. If w is small then q is not the representative of w.*

Proof: The proof is by induction on k. The basis $k = j$ was already proved.
Induction Step: Assume the correctness of the statement for all smaller values of $k, k \geq j+1$, and prove it for k. If w is not growing, then it is identical to an $\hat{\mathcal{F}}$-descendant $w' \in \mathcal{F}_{k'}$, for some $k' < k$. Hence $r(w) = r(w')$. By the induction hypothesis, $r(w') \neq q$, and so $r(w) \neq q$ as well.

Otherwise, w is growing. By Lemma D.26, its kernel set $K(w)$ contains at least two points p, p' with plain counter zero, i.e., $plain\_ctr_{k-1}(p) = plain\_ctr_{k-1}(p') = 0$. On the other hand, $plain\_ctr_{k-1}(q) \geq plain\_ctr_i(q) = 1$. Hence q is not the representative of w. Claim D.28 follows. ∎

We now continue proving Lemma D.27.
By Claim D.28, if $v_k(q)$ is small then the plain counter of q is not incremented during the k-level processing, for all $k \geq j$. If $v_k(q)$ is large, then obviously, it is not incremented either. Hence, for any $k \geq j$, the plain counter of q is not incremented during the k-level processing, and so $plain\_ctr_\ell(q) = plain\_ctr_j(q)$. Thus the plain counter of q may grow only during the k-level processing, for $i \leq k < j$. It follows that $plain\_ctr_\ell(q) \leq j - i \leq \beta = \gamma + \eta$. ∎

Recall that for any $q \in Q$, $load\_ctr_\ell(q) = CTR_\ell(q) + ctr_\ell(q) = CTR_\ell(q) + single\_ctr_\ell(q) + plain\_ctr_\ell(q)$. Hence, Lemmas D.24, D.25 and D.27 imply the following corollary.

Corollary D.29 *For any point $q \in Q$, $load\_ctr_\ell(q) \leq 2 \cdot (\gamma + \eta) + 1$.*

Observe that each time that the load counter of a point q is incremented, its degree in the constructed spanner grows by at most $O(\Delta(n))$. (This is because the maximum degree of the j-level auxiliary spanner $\tilde{G}_j$ is $O(\Delta(n))$, for each $j \in [\ell]$; see Observation D.12.) Hence, Corollary D.29 implies that the maximum degree of any point $q \in Q$ in the graph $\tilde{G}_1 \cup \ldots \cup \tilde{G}_\ell$ is $O(\Delta(n) \cdot (\gamma + \eta)) = O(\Delta(n) \cdot \gamma)$. By Observation D.12, the 0-level auxiliary spanner $\tilde{G}_0$ contributes at most $O(\Delta(n))$ to the maximum degree of the final spanner $\tilde{G}$. The path-spanner H has maximum degree $O(\rho)$, and the base edge set $\mathcal{B}$ contributes an additive term of $O(1)$ to $\Delta(\tilde{G})$; see the beginning of this appendix. We summarize the degree analysis with the following statement.

Lemma D.30 $\Delta(\tilde{G}) = O(\Delta(n) \cdot \gamma + \rho) = O(\Delta(n) \cdot \log_\rho(t/\epsilon) + \rho)$.

Deriving Theorem 1.3. Lemmas 3.4, D.9, D.11 and D.30, and Corollary D.10 imply Theorem 1.3.

References

[1] I. Abraham, Y. Bartal, and O. Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):30263126, 2011.

[2] P. K. Agarwal, Y. Wang, and P. Yin. Lower bound for sparse Euclidean spanners. In *Proc. of 16th SODA*, pages 670–671, 2005.

[3] I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[4] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.

[5] S. Arya, D. M. Mount, and M. H. M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. of 35th FOCS*, pages 703–712, 1994.

[6] S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. In *Proc. of 2nd ESA*, pages 48–59, 1994.

[7] P. Assouad. Plongements lipschitziens dans $\setminus^n$. *Bull. Soc. Math. France*, 111(4):429448, 1983.

[8] Y. Bartal, L. Gottlieb, and R. Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proc. of 44th STOC (to appear)*, 2012.

[9] H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006.

[10] H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. On hierarchical routing in doubling metrics. In *Proc. of 16th SODA*, pages 762–771, 2005.

[11] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. In *Proc. of 8th SOCG*, pages 192–201, 1992.

[12] D. Z. Chen, G. Das, and M. H. M. Smid. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Applied Mathematics*, 110(2-3):151–167, 2001.

[13] L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. of 2nd SOCG*, pages 169–177, 1986.

[14] K. L. Clarkson. Approximation algorithms for shortest path motion planning. In *Proc. of 19th STOC*, pages 56–65, 1987.

[15] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.*, 110(1):6393, 1999.

[16] R. Cole and L. Gottlieb. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proc. of 38th STOC*, pages 574–583, 2006.

[17] T. H. Corman, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edition.* McGraw-Hill Book Company, Boston, MA, 2001.

[18] G. Das, P. J. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional Euclidean space. In *Proc. of 9th SOCG*, pages 53–62, 1993.

[19] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *Proc. of 10th SOCG*, pages 132–139, 1994.

[20] G. Das, G. Narasimhan, and J. S. Salowe. A new way to weigh malnourished Euclidean graphs. In *Proc. of 6th SODA*, pages 215–222, 1995.

[21] Y. Dinitz, M. Elkin, and S. Solomon. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Proc. of 49th FOCS*, pages 519–528, 2008.

[22] J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. In *Proc. of 20th SoCG*, pages 190–199, 2004.

[23] L. Gottlieb, A. Kontorovich, and R. Krauthgamer. Efficient regression in metric space via approximate lipschitz extension. Manuscript (submitted to SIAM J. Comput.), 2012.

[24] L. Gottlieb and L. Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *Proc. of 19th SODA*, pages 591–600, 2008.

[25] L. Gottlieb and L. Roditty. An optimal dynamic spanner for doubling metric spaces. In *Proc. of 16th ESA*, pages 478–489, 2008.

[26] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.*, 31(5):1479–1500, 2002.

[27] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric graphs. In *Proc. of 13th SODA*, pages 828–837, 2002.

[28] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric spanners. *ACM Transactions on Algorithms*, 4(1), 2008.

[29] J. Gudmundsson, G. Narasimhan, and M. H. M. Smid. Fast pruning of geometric spanners. In *Proc. of 22nd STACS*, pages 508–520, 2005.

[30] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. of 44th FOCS*, page 534543, 2003.
[31] S. Har-Peled and M. Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.
[32] Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. In *Proc. of 19th PODC*, pages 41–50, 2000.
[33] J. M. Keil. Approximating the complete Euclidean graph. In *Proc. of 1st SWAT*, pages 208–213, 1988.
[34] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.
[35] R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proc. of 15th SODA*, page 791801, 2004.
[36] H. P. Lenhof, J. S. Salowe, and D. E. Wrege. New methods to mix shortest-path and minimum spanning trees. manuscript, 1994.
[37] Y. Mansour and D. Peleg. An approximation algorithm for min-cost network design. *DIMACS Series in Discr. Math and TCS*, 53:97–106, 2000.
[38] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
[39] S. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proc. of 30th STOC*, pages 540–550, 1998.
[40] L. Roditty. Fully dynamic geometric spanners. In *Proc. of 23rd SoCG*, pages 373–380, 2007.
[41] J. S. Salowe. Construction of multidimensional spanner graphs, with applications to minimum spanning trees. In *Proc. of 7th SoCG*, pages 256–261, 1991.
[42] M. H. M. Smid. The weak gap property in metric spaces of bounded doubling dimension. In *Proc. of Efficient Algorithms*, pages 275–289, 2009.
[43] S. Solomon. An optimal time construction of Euclidean sparse spanners with tiny diameter. In *Proc. of 22nd SODA*, pages 820–839, 2011.
[44] S. Solomon and M. Elkin. Balancing degree, diameter and weight in Euclidean spanners. In *Proc. of 18th ESA*, pages 48–59, 2010.
[45] K. Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proc. of 36th STOC*, page 281290, 2004.
[46] P. M. Vaidya. A sparse graph almost as good as the complete graph on points in k dimensions. *Discrete & Computational Geometry*, 6:369–381, 1991.