Optimal Euclidean Spanners: Really Short, Thin, and Lanky

MICHAEL ELKIN, Ben-Gurion University of the Negev
SHAY SOLOMON, Weizmann Institute of Science

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. [1995] devised a construction of Euclidean $(1+\epsilon)$-spanners that achieves constant degree, diameter $O(\log n)$, weight $O(\log^2 n) \cdot \omega(MST)$, and has running time $O(n \cdot \log n)$. 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, and within the same running time.

This conjecture of Arya et al. became one of the most central open problems in the area of Euclidean spanners. Nevertheless, the only progress since 1995 towards its resolution was achieved 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 [Dinitz et al. 2008; Agarwal et al. 2005].

In this article we resolve the long-standing conjecture of Arya et al. in the affirmative. We present a spanner construction with the same stretch, degree, diameter, and running time, 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 three ways. First, we demonstrate that the conjecture holds true not only in constant-dimensional Euclidean spaces, but also in *doubling metrics*. Second, we provide a general trade-off between the three involved parameters, which is *tight in the entire range*. Third, we devise a transformation that decreases the lightness of spanners in *general metrics*, while keeping all their other parameters in check. Our main result is obtained as a corollary of this transformation.

35

Categories and Subject Descriptors: F.2.3 [**Analysis of Algorithms and Problem Complexity**]: Tradeoffs between Complexity Measures; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Graph algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Doubling metrics, Euclidean spaces, Euclidean spanners

ACM Reference Format:
Michael Elkin and Shay Solomon. 2015. Optimal Euclidean spanners: Really short, thin, and lanky. J. ACM 62, 5, Article 35 (October 2015), 45 pages.
DOI: http://dx.doi.org/10.1145/2819008

A preliminary version of this article appeared in *Proceedings of STOC'13*.
The work of M. Elkin is supported by the BSF grant No. 2008430, by the ISF grant No. 87209011, and by the Lynn and William Frankel Center for Computer Sciences.
Part of this work was done while S. Solomon was a graduate student in the Department of Computer Science, Ben-Gurion University of the Negev, under the support of the Clore Fellowship grant No. 81265410, the BSF grant No. 2008430, and the ISF grant No. 87209011.
Authors' addresses: M. Elkin, Department of Computer Science, Ben-Gurion University of the Negev, POB 653, Beer-Sheva 84105, Israel; S. Solomon, Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot 76100, Israel; email: solo.shay@gmail.com.

DOI: http://dx.doi.org/10.1145/2819008

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 article we focus on the regime $t = 1 + \epsilon$, for $\epsilon > 0$ being an arbitrarily small constant. We will also concentrate on spanners with $|E| = O(n)$ edges. Euclidean spanners were introduced by Chew [1986] in 1986. The first constructions of $(1 + \epsilon)$-spanners with $O(n)$ edges were devised soon afterwards [Clarkson 1987; Keil 1988], and the running time of such constructions was improved to $O(n \cdot \log n)$ a few years later [Vaidya 1991; Salowe 1991].

Euclidean spanners turned out to be a fundamental geometric construct, with numerous applications. In particular, they were found useful in geometric approximation algorithms [Rao and Smith 1998; Gudmundsson et al. 2002b, 2002c, 2008], geometric distance oracles [Gudmundsson et al. 2002b, 2002c, 2005, 2008] and network design [Hassin and Peleg 2000; Mansour and Peleg 2000]. Various properties of Euclidean spanners are a subject of intensive ongoing research effort [Keil and Gutwin 1992; Chandra et al. 1992; Althöfer et al. 1993; Das et al. 1993; Das and Narasimhan 1994; Arya et al. 1995; Das et al. 1995; Rao and Smith 1998; Gudmundsson et al. 2002a; Agarwal et al. 2005; Chan and Gupta 2006; Dinitz et al. 2008]. See also the book by Narasimhan and Smid [2007], and the references therein. This book is titled *Geometric Spanner Networks*, and it is devoted almost exclusively to Euclidean spanners and their numerous 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 *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.

In this section we may write "spanner" as a shortcut for a "$(1 + \epsilon)$-spanner with $O(n)$ edges". Feder and Nisan devised a construction of spanners with bounded degree (see [Arya and Smid 1994; Salowe 1992; Vaidya 1991]). In FOCS '94, Arya et al. [1994] 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. [1995] in STOC '95. (Further work on the trade-off between the diameter and number of edges in spanners can be found [Chan and Gupta 2006; Narasimhan and Smid 2007; Solomon 2011].)

Also, in the beginning of the nineties researchers started to systematically investigate spanners that *combine* several parameters (among degree, diameter and lightness). Arya and Smid [1994] devised a construction of spanners with constant degree and lightness. The running time of their construction is $O(n \cdot \log^d n)$, where $d = O(1)$ stands for the Euclidean dimension. Other spanner constructions with constant degree and lightness, but with running time of $O(n \cdot \log n)$, were subsequently devised [Das and Narasimhan 1994; Gudmundsson et al. 2002a]. Arya et al. [1994] and [1995] devised a construction of spanners with logarithmic diameter and logarithmic lightness. (This combination was shown to be optimal by Dinitz et al. [2008] in FOCS '08; see also Lenhof et al. [1994] and Agarwal et al. [2005] for previous lower bounds on this

Table I.
A comparison of previous and new constructions of $(1+\epsilon)$-spanners with $O(n)$ edges for low-dimensional euclidean metrics. All these constructions have the same running time $O(n \cdot \log n)$.

Reference	Degree	Diameter	Lightness
[Gudmundsson et al. 2002a]	$O(1)$	unspecified	$O(1)$
[Arya et al. 1995]	unspecified	$O(\log n)$	$O(\log n)$
[Arya et al. 1995]	$O(1)$	$O(\log n)$	$O(\log^2 n)$
[Arya et al. 1995]	unspecified	$O(\alpha(n))$	unspecified
[Solomon and Elkin 2010]	$O(\rho)$	$O(\log_\rho n + \alpha(\rho))$	$O(\rho \cdot \log_\rho n \cdot \log n)$
New	$\boldsymbol{O(1)}$	$\boldsymbol{O(\log n)}$	$\boldsymbol{O(\log n)}$
New	$\boldsymbol{O(\rho)}$	$\boldsymbol{O(\log_\rho n + \alpha(\rho))}$	$\boldsymbol{O(\rho \cdot \log_\rho n)}$

problem.) This construction of Arya et al. [1995] may have, however, an arbitrarily large degree. On the other hand, Arya et al. [1995] also devised 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. [1995] conjectured that one can obtain a spanner with constant degree, logarithmic diameter and logarithmic lightness within time $O(n \cdot \log n)$. Specifically, they wrote the following.

Conjecture 1 ([Arya et al. 1995]). For any $t > 1$, and any dimension d, there is a t-spanner, constructible in $O(n \cdot \log n)$ time, with bounded degree, $O(\log n)$ diameter, and weight $O(\omega(MST) \cdot \log n)$.

In this article[1] we prove the conjecture of Arya et al. [1995], and devise a construction of $(1+\epsilon)$-spanners with bounded degree, and with logarithmic diameter and lightness. The running time of our construction is $O(n \cdot \log n)$, matching the time bound conjectured in Arya et al. [1995]. Moreover, this running time is optimal in the algebraic computation-tree model [Chen et al. 2001]. (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, even for 2-dimensional point sets.)

In fact, our result is far more general than this. Specifically, we provide a trade-off parametrized by a degree parameter $\rho \geq 2$, summarized in the following.

THEOREM 1.1. *For any set of n points in Euclidean space of any constant dimension d, any constant $\epsilon > 0$ and any 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 our construction is $O(n \cdot \log n)$.*

Due to lower bounds by Chan and Gupta [2006] and Dinitz et al. [2008], this trade-off is *optimal in the entire range* of the parameter ρ. See Table I for a concise summary of previous and our results for low-dimensional Euclidean metrics. See also Theorem A.1 for explicit dependencies on ϵ in Theorem 1.1.

1.2. Doubling Metrics

Our result extends in another direction as well. Specifically, it applies to any *doubling metric*. 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. This generalizes the Euclidean dimension, because the doubling dimension of Euclidean space $\mathbb{R}^d$ is proportional to d. We will denote the doubling dimension of an arbitrary metric M by $dim(M)$. Metric M is called *doubling* if its doubling dimension $dim(M)$ is constant.

[1] An earlier version of this article can be found in Elkin and Solomon [2013].

Table II. A Comparison of Previous and New Constructions of $(1+\epsilon)$-Spanners with $O(n)$ Edges for Doubling Metrics

Reference	Degree	Diameter	Lightness	Running Time
[Chan et al. 2005]	$O(1)$	unspecified	unspecified	unspecified
[Gottlieb and Roditty 2008b]	$O(1)$	unspecified	unspecified	$O(n \cdot \log n)$
[Chan and Gupta 2006]	unspecified	$O(\alpha(n))$	unspecified	$O(n \cdot \log n)$
[Smid 2009]	unspecified	unspecified	$O(\log n)$	$O(n^2 \cdot \log n)$
[Gottlieb et al. 2012]	$O(1)$	$O(\log n)$	unspecified	$O(n \cdot \log n)$
New	$\boldsymbol{O(1)}$	$\boldsymbol{O(\log n)}$	$\boldsymbol{O(\log n)}$	$\boldsymbol{O(n \cdot \log n)}$
New	$\boldsymbol{O(\rho)}$	$\boldsymbol{O(\log_\rho n + \alpha(\rho))}$	$\boldsymbol{O(\rho \cdot \log_\rho n)}$	$\boldsymbol{O(n \cdot \log n)}$

Doubling metrics, implicit in the works of Assouad [1983] and Clarkson [1999], were explicitly defined by Gupta et al. [2003]. They were subject of intensive research since then [Krauthgamer and Lee 2004; Talwar 2004; Har-Peled and Mendel 2006; Chan and Gupta 2006; Abraham et al. 2011; Bartal et al. 2012].

Spanners for doubling metrics were also intensively studied [Gao et al. 2004; Chan et al. 2005; Har-Peled and Mendel 2006; Roditty 2007; Gottlieb and Roditty 2008a; 2008b; Smid 2009]. They were also found useful for approximation algorithms [Bartal et al. 2012], and for machine learning [Gottlieb et al. 2012]. In SODA '05 Chan et al. [2005] showed that for any doubling metric there exists a spanner with constant degree. In SODA '06, Chan and Gupta [2006] devised a construction of spanners with diameter $O(\alpha(n))$. Smid [2009] showed that in doubling metrics a greedy construction produces spanners with logarithmic lightness. (The greedy spanner can be constructed within time $O(n^2 \cdot \log n)$ in doubling metrics [Bose et al. 2010].) Gottlieb et al. [2012] devised a construction of spanners with constant degree and logarithmic diameter, within $O(n \cdot \log n)$ time. To the best of our knowledge, prior to our work, there were no known constructions of spanners for doubling metrics that provide logarithmic diameter and lightness simultaneously (even allowing arbitrarily large degree).[2]

We show that our construction extends to doubling metrics without incurring any overhead (beyond constants) in the degree, diameter, lightness, and running time. In other words, Theorem 1.1 applies to doubling metrics. (In Appendix A we provide a more general statement of Theorem 1.1 with explicit dependencies on ϵ and on the doubling dimension.) See Table II for a summary of previous and our results for doubling metrics.

1.3. Our and Previous Techniques

Our starting point is the paper of Chandra et al. [1992] from SoCG '92 (see also Chandra et al. [1995]). In that paper the authors devised a general transformation: given a 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 article we devise a much more refined transformation. Our transformation enjoys all the useful properties of the transformation of Chandra et al. [1992], but, in addition, it preserves (up to constant factors) the degree and the diameter of the original 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. [1995] in the Euclidean case, and due to Gottlieb et al. [2012] in the case of doubling metrics). As a result we obtain a construction of spanners with constant degree,

[2]On the other hand, as was mentioned in Section 1.1, for Euclidean metrics such a construction was devised by Arya et al. [1995]. However, the degree in the latter construction is unbounded.

logarithmic diameter and logarithmic lightness. The latter proves the conjecture of Arya et al. [1995].

We remark that our transformation can be applied not only for Euclidean or doubling metrics, but rather in much more general scenarios. In fact, in Elkin and Solomon [2013] we have already obtained some improved results for spanners in general graphs that are based on a variant of this transformation. We have also obtained several new results as additional corollaries of our transformation, including optimal constructions of spanners for metrics induced by graphs of bounded tree-width or bounded tree-length, and general constructions (i.e., constructions that provide a general trade-off between the degree, diameter and lightness) of fault-tolerant spanners for doubling metrics. These new results are part of ongoing research, and are outside the scope of the current article.

Next, we provide a schematic overview of the two transformations (the one due to Chandra et al. [1992], and our refined one). The transformation of Chandra et al. [1992] 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 $c \cdot n$ intervals of length $\frac{|\mathcal{L}|}{c \cdot n}$ each, for a constant $c > 1$. 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 $c \cdot n/2$ intervals of length $2 \cdot \frac{|\mathcal{L}|}{c \cdot n}$ each. The hierarchy $\mathcal{F}$ consists of $\ell = \log n$ levels, with c intervals of length $\frac{|\mathcal{L}|}{c}$ each in the last level $\mathcal{F}_\ell$.

In each level $j \in [\ell]$ of the hierarchy each nonempty 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, that is, "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 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 in many different levels.

To fix the problem with the diameter we 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 in 1995 by Arya et al. [1995]. Plugging[3] this 1-dimensional spanner construction into the transformation of Chandra et al. [1992] gives rise to an improved transformation that keeps the diameter in check, but still blows up the degree.

To fix the problem with the degree, it is natural to try distributing the degree load evenly between "nearby" points along $\mathcal{L}$. Alas, if one sticks with the original hierarchy $\mathcal{F}$ of partitions of $\mathcal{L}$ into intervals, this turns out to be impossible. The problem is that the same point may well be the only eligible representative for many levels of the hierarchy. Overcoming this hurdle is the heart of our article. 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. As our algorithm proceeds it carefully moves points between bags so as to guarantee that no point will

[3] In fact, we use our own more recent construction [Solomon and Elkin 2010] of 1-spanners for 1-dimensional spaces with the given properties. Having stretch 1 instead of $(1 + \epsilon)$ simplifies the analysis.

ever be overloaded. At the same time we never put points that are far away from one another in the original metric into the same bag. Indeed, if remote points 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 providing short $(1+\epsilon)$-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 $\mathcal{L}$. Our construction balances carefully between these two contradictory requirements.

1.4. Related Work

Most of the related work was already discussed earlier. One more relevant result is the ESA '10 paper [Solomon and Elkin 2010] by the authors of the current article. There we devised a construction of spanners that trades gracefully between the degree, diameter and lightness. That construction, however, could only match the previous suboptimal bounds of Arya et al. [1995], but not improve them. In particular, the lightness of the construction of Solomon and Elkin [2010] is $\Omega(\log^2 n)$, regardless of the other parameters.

1.5. Consequent Work

A preliminary version of this article started to circulate in April 2012 [Elkin and Solomon 2013]. It sparked a number of follow-up papers. First, in Elkin and Solomon [2013] we used the technique developed in that paper to devise an efficient construction of light spanners for general graphs. Second, Chan et al. [2013] came up with an alternative construction of spanners for doubling metrics with constant degree, and logarithmic diameter and lightness. Their construction and analysis are arguably simpler than ours. In addition, they extended this result to the fault-tolerant setting. A yet alternative construction of fault-tolerant spanners with the same properties and with running time $O(n \cdot \log n)$ was devised recently by Solomon [2014]. However, while our construction provides an optimal trade-off between the diameter and lightness ($O(\log_\rho n + \alpha(\rho))$ versus $O(\rho \cdot \log_\rho n)$ for the entire range of the parameter $\rho \geq 2$), the constructions of Chan et al. [2013] and Solomon [2014] do the job only for $\rho = O(1)$. As far as we know they cannot be extended to provide the general trade-off. Finally, the constructions of Chan et al. [2013] and Solomon [2014] do not provide a transformation for converting spanners into light spanners in general metrics.

Finally, we stress that both constructions [Chan et al. 2013] and [Solomon 2014] are consequent to our work. These constructions build upon ideas and techniques that we present in the current article.

1.6. Structure of the Article

In Section 2 we describe our construction (Algorithm *LightSp*). The description of the algorithm is provided in Sections 2.1–2.6. A detailed outline of Section 2 appears in the paragraph preceding Section 2.1. We analyze the properties of the spanners produced by our algorithm in Section 3. The most elaborate and technically involved parts of the analysis concern the stretch and diameter (Section 3.3) and the degree (Section 3.4) of the produced spanners.

1.7. Preliminaries

The following theorem provides optimal spanners for *1-dimensional* Euclidean metrics with respect to all three parameters (degree, diameter and lightness).

THEOREM 1.2 ([ARYA ET AL. 1995; SOLOMON AND ELKIN 2010]). *For any n-point 1-dimensional space M and any $\rho \geq 2$, there exists a 1-spanner H with $|H| = O(n)$,*

$\Delta(H) = O(\rho)$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$ and $\Psi(H) = O(\rho \cdot \log_\rho n)$. The running time of this construction is $O(n)$.

The following theorem provides spanners for doubling metrics with an optimal trade-off between the degree and diameter. Note, however, that this trade-off does not involve lightness.

THEOREM 1.3 [ARYA ET AL. 1995; GOTTLIEB AND RODITTY 2008B; SOLOMON AND ELKIN 2010]. *For any n-point doubling metric $M = (P, \delta)$, any constant $\epsilon > 0$ and any $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner H with $|H| = O(n)$, $\Delta(H) = O(\rho)$ and $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$. The running time of this construction is $O(n \cdot \log n)$.*

For the sake of completeness we provide a proof of Theorem 1.3 in Appendix B. Moreover, we detail there the dependencies on ϵ and the doubling dimension $dim(M)$ on various parameters of the spanner constructed by this theorem.

Our transformation theorem is formulated as follows.

THEOREM 1.4. *Let $M = (P, \delta)$ be an arbitrary metric. Let $t \geq 1, \rho \geq 2$ be arbitrary parameters. Suppose that for any subset $Q \subseteq P$, $|Q| = n$, there exists an algorithm (henceforth, Algorithm BasicSp) which builds a t-spanner H for the submetric $M[Q]$ of M induced by the point set Q, so that $|H| \leq SpSz(n)$, $\Delta(H) \leq \Delta(n)$, $\Lambda(H) \leq \Lambda(n)$. 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 nondecreasing, while the functions $SpSz(n)$ and $SpTm(n)$ are also convex and vanish at zero.*

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 $|H'| = O(SpSz(n) \cdot \max\{1, \log_\rho(t/\epsilon)\})$, $\Delta(H') = O(\Delta(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + \rho)$, $\Lambda(H') = O(\Lambda(n) + \log_\rho n + \alpha(\rho))$, $\Psi(H') = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot (t^3/\epsilon))$. The running time of Algorithm LightSp is $O(SpTm(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + n \cdot \log n)$.

Given this theorem we derive our main result by instantiating the algorithm from Theorem 1.3 as Algorithm *BasicSp* in Theorem 1.4. As a result we obtain a construction of $(1+\epsilon)$-spanners 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 $O(n \cdot \log n)$. (We substituted $t = 1 + \epsilon$. Observe that in the case that ϵ is constant, the terms $\max\{1, \log_\rho(t/\epsilon)\}$ and (t^3/ϵ) that appear in the statement of Theorem 1.4 are constant too.) In Appendix A we detail the dependencies on ϵ and the doubling dimension $dim(M)$ on various parameters of the spanner constructed by Theorem 1.4.

For a pair of nonnegative integers $i, j, i \leq j$, we denote $[i, j] = \{i, i+1, \ldots, j\}$, $[i] = \{1, 2, \ldots, i\}$.

For paths Π, Π' connecting vertices v and u and u and w, respectively, denote by $\Pi \circ \Pi'$ the concatenation of these paths.

2. ALGORITHM *LIGHTSP*

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

Algorithm *LightSp* starts with computing an MST, or an approximate MST, T, for the metric $M[Q]$. In general, a t-approximate MST can be computed within time $O(SpTm(n) + n \cdot \log n)$ by running Prim's MST Algorithm over the t-spanner produced by Algorithm *BasicSp*. In low-dimensional Euclidean and doubling metrics, a $(1+\epsilon)$-spanner with $O(n)$ edges can be built in $O(n \log n)$ time (cf. [Gottlieb and Roditty 2008b]), implying that a $(1+\epsilon)$-approximate MST in such metrics can be computed within $O(n \cdot \log n)$ time.

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 ([Cormen et al. 2001], ch. 36) that $L \le 2 \cdot \omega(T)$, and so $L = O(t \cdot \omega(MST(M[Q])))$. 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*), that is, $\delta_{\mathcal{L}}(v_k, v_{k'}) = \sum_{i=k}^{k'-1} \delta(v_i, v_{i+1})$, for every pair k, k' of indices, $1 \le k < k' \le n$. We employ Theorem 1.2 to build in $O(n)$ time a 1-spanner $H_{\mathcal{L}}$ for $M_{\mathcal{L}}$ with $|H_{\mathcal{L}}| = O(n)$, $\Delta(H_{\mathcal{L}}) = O(\rho)$, $\Lambda(H_{\mathcal{L}}) = O(\log_\rho n + \alpha(\rho))$ and $\Psi(H_{\mathcal{L}}) = O(\rho \cdot \log_\rho n)$. Let $H = (Q, E_H)$ be the graph obtained from $H_{\mathcal{L}}$ by assigning weight $\delta(p, q)$ to each edge $(p, q) \in H_{\mathcal{L}}$. Since edge weights in H are no greater than the corresponding edge weights in $H_{\mathcal{L}}$, we have (i) $\omega(H) \le \omega(H_{\mathcal{L}}) = O(\rho \cdot \log_\rho n) \cdot L$, 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 $\prec_{\mathcal{L}}$ on the point set Q. Specifically, we write $q_i \prec_{\mathcal{L}} q_j$ (respectively, $q_i \preceq_{\mathcal{L}} q_j$) iff $i < j$ (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 (t and ϵ will be set as constants). For $j \in [\ell]$, we define $\xi_j = \rho^{j-1} \cdot \frac{L}{n}$. Divide the path $\mathcal{L}$ into $n_j = \lceil \frac{c \cdot L}{\xi_j} \rceil = \lceil \frac{c \cdot n}{\rho^{j-1}} \rceil$ intervals of length $\mu_j = \frac{\xi_j}{c}$ each (except for maybe one interval of possibly shorter length). From now on we assume that each n_j is equal to $\frac{c \cdot L}{\xi_j} = \frac{c \cdot n}{\rho^{j-1}}$, because non-integrality of this expression has no effect whatsoever on the analysis. Define also the *j-level threshold* $\tau_j = 2\mu_j \cdot \rho \cdot t \cdot (c + 1) = 2 \cdot \frac{L}{n} \cdot t \cdot (1 + \frac{1}{c}) \cdot \rho^j$. These intervals induce a partition of the point set Q in the obvious way; denote these intervals by $I_j^{(1)}, I_j^{(2)}, \ldots, I_j^{(n_j)}$. (Note that the point set of an interval $I_j^{(i)}$ may be empty.)

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. (Similar to before, we may assume that ρ is an integer.) 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*) correspond to intervals from $\mathcal{I}$; we call $\mathcal{F}$ the *(original) bag forest*. With a slight abuse of notation we denote by $\mathcal{I}$ also the set of bags in $\mathcal{F}$, and by $\mathcal{F}_j = \mathcal{I}_j$ the set of j-level bags in $\mathcal{F}$, for each $j \in [\ell]$. Each of the trees in $\mathcal{F}$ is rooted at an ℓ-level interval. Thus, the number of trees in the bag forest $\mathcal{F}$ is equal to the number $|\mathcal{F}_\ell| = |\mathcal{I}_\ell| = n_\ell$ of ℓ-level intervals. Specifically, $n_\ell = \frac{c \cdot n}{\rho^{\ell-1}}$, and so $c < n_\ell \le c \cdot \rho$. Denote the interval that corresponds to a bag v of $\mathcal{F}$ by $I(v)$, and denote the point set of $I(v)$ by $N(v)$. We call the point set $N(v)$ the *native point set* of v. (Note that $N(v)$ may be empty.) For an inner (i.e., non-leaf) 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 $N(v) = \bigcup_{i=1}^{\rho} N(c_i(v))$. Note that $\bigcup_{v \in \mathcal{F}_j} I(v) = [q_1, q_n]$, and $\bigcup_{v \in \mathcal{F}_j} N(v) = Q$. Also, for any pair of distinct bags $u, v \in \mathcal{F}_j$, $I(u) \cap I(v) = N(u) \cap N(v) = \emptyset$.

In Algorithm *LightSp* we (implicitly) maintain another forest $\hat{\mathcal{F}}$ over the same bag set $\mathcal{I}$; we call $\hat{\mathcal{F}}$ the *adoption bag forest*. Specifically, a j-level bag v, for some index $j \in [\ell - 1]$, may become a child of some $(j + 1)$-level bag u, other than the parent $\pi(v)$ of v in the original bag forest $\mathcal{F}$. If this happens we say that u becomes a *step-parent* of v in the original bag forest $\mathcal{F}$ (and u is a parent of v in the adoption bag forest $\hat{\mathcal{F}}$), and v becomes a *step-child* of u in the original bag forest $\mathcal{F}$ (and v is a child of u in the adoption bag forest $\hat{\mathcal{F}}$). As a result the points associated with the bag v become associated with u. We will soon provide more details on this.

Observe that in the original bag forest $\mathcal{F}$ each bag v corresponds to a specific interval $I(v) \in \mathcal{I}$, and contains only points that lie within this interval (i.e., the points of the native point set $N(v)$ of v). On the other hand, each bag v in the adoption bag forest $\hat{\mathcal{F}}$ may contain points from many different intervals of $\mathcal{I}$. We denote by $\hat{\mathcal{F}}_j$ the set of j-level bags of the adoption bag forest $\hat{\mathcal{F}}$. If a bag $v \in \mathcal{F}_j$ becomes a child in the adoption bag forest $\hat{\mathcal{F}}$ of a bag u, then it will hold that $u \in \mathcal{F}_{j+1}$. This guarantees that $\mathcal{F}_j = \hat{\mathcal{F}}_j$, for every $j \in [\ell]$.

The rest of this section is organized as follows. In Section 2.1 we describe point sets that are associated with bags of the adoption bag forest $\hat{\mathcal{F}}$. In Section 2.2 we describe an important subset of edges of the ultimate spanner that the algorithm constructs. This subset is called the *base edge set*. During the execution of the algorithm some bags are labeled as *zombies* or *incubators*. These notions are discussed in Section 2.3. In Section 2.4 we describe how our algorithm selects representatives of different bags. Section 2.5 is devoted to Procedure *Attach*, which is a subroutine of our algorithm. The algorithm itself is described in Section 2.6.

2.1. Point Sets

In addition to the native point set $N(v)$, the algorithm will also maintain for each bag v 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 $B(v) \subseteq N(v)$. 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 the *j-level processing*, for each index $j \in [\ell]$. (It will be described in Section 2.6.) 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 $B(v) = K(v) = Q(v) = N(v)$.

A nonempty $(j-1)$-level bag z, $j \in [2, \ell]$, may become a step-child of 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 joins* v. Denote by $\mathcal{J}(v)$ the set of bags z that join the bag v. They will be referred to as the *joining step-children* (or, in short, *step-children*) of v. Denote also by $\mathcal{S}(v)$ the set of *surviving children* of v, that is, the nonempty bags z with $v = \pi(z)$ that did not join 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 nonempty.

Each bag z will be a step-child of at most one bag v. Also, for any bag v, each nonempty child u of v which is not surviving will necessarily be a step-child of some other bag $v' \neq v$. (The bags v and v' are of the same level.) Hence, for each level $j \in [\ell]$, the collection $\{Q(v) \mid v \in \mathcal{F}_j\}$ is a partition of Q. In particular, for distinct $u, v \in \mathcal{F}_j$, $Q(u) \cap Q(v) = \emptyset$.

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, that is, $B(v) = \bigcup_{z \in \mathcal{S}(v)} B(z)$, $Q(v) = \bigcup_{z \in \chi(v)} Q(z)$. 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 $Q(v)$ 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. This will guarantee that points of $K(v)$ provide good substitutes for points of $B(v)$. Consequently, the points of $K(v)$ 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 discussed in the introduction, if one selects representatives only from the native point set $N(v)$, 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 kernel set* $K'(v)$ is given by $K'(v) = \bigcup_{z \in \mathcal{S}(v)} K(z)$. If $|K'(v)| \geq \ell$ then the kernel set of v is set to be equal to its surviving kernel, that is, $K(v) = K'(v)$. Otherwise (if $|K'(v)| < \ell$), 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 set $K'(v)$ (i.e., setting $K(v) = K'(v) \cup \bigcup_{z \in \mathcal{J}(v)} K(z)$) in the case that $|K'(v)| < \ell$ is that in this case the surviving kernel set is too small. Hence one needs to add to it more points to alleviate the degree load.

In the complementary case ($|K'(v)| \geq \ell$), one can distribute the load of the $O(\ell)$ auxiliary spanners that Algorithm *LightSp* constructs (see Section 2.6) among the points of $K'(v)(= K(v))$ in such a way that no kernel point is overloaded.

Definition 2.1. A bag v is called *small* if $|Q(v)| < \ell$, and *large* otherwise.

The next lemma follows from these definitions.

LEMMA 2.2. *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 $|K(v)| \geq \ell$.*

PROOF. We prove both assertions of the lemma by induction on j.
Basis: $j = 1$. In this case $B(v) = K(v) = Q(v)$. Also, if v is large, then $|K(v)| = |Q(v)| \geq \ell$.
Induction Step: Assuming the correctness of the statement for all index values smaller than j, for some $j \in [2, \ell]$, we prove its correctness for index value j.

We start with proving the first assertion of the lemma, that is, we assume that v is small and show that $K(v) = Q(v)$. Recall that $Q(v)$ is given by $Q(v) = \bigcup_{z \in \chi(v)} Q(z)$. Moreover, $K'(v) \subseteq Q(v)$, and thus $|K'(v)| \leq |Q(v)| < \ell$. By definition, $K(v) = \bigcup_{z \in \chi(v)} K(z)$. Observe that for every $z \in \chi(v)$, $Q(z) \subseteq Q(v)$. Hence all bags $z \in \chi(v)$ are small as well. The first assertion of the induction hypothesis implies that $K(z) = Q(z)$, for each $z \in \chi(v)$. Hence $K(v) = Q(v)$.

Next, we prove the second assertion of the lemma, that is, we assume that v is large and show that $|K(v)| \geq \ell$.

Suppose first that $|K'(v)| = |\bigcup_{z \in \mathcal{S}(v)} K(z)| \geq \ell$. In this case $K(v) = K'(v)$, and so $|K(v)| \geq \ell$.

We are now left with the case that $|K'(v)| < \ell$. In this case $K(v) = \bigcup_{z \in \chi(v)} K(z)$.

If there exists a large bag $z \in \chi(v)$, then the second assertion of the induction hypothesis yields $|K(z)| \geq \ell$, which implies that $|K(v)| \geq |K(z)| \geq \ell$.

Otherwise, all bags $z \in \chi(v)$ are small. The first assertion of the induction hypothesis yields $K(z) = Q(z)$, for all bags $z \in \chi(v)$, and thus

$$K(v) = \bigcup_{z \in \chi(v)} K(z) = \bigcup_{z \in \chi(v)} Q(z) = Q(v).$$

Hence $|K(v)| = |Q(v)| \geq \ell$. □

As mentioned before, for every index $j \in [\ell]$, $Q = \bigcup_{v \in \mathcal{F}_j} Q(v)$, and for any pair u, v of distinct j-level bags, $Q(u) \cap Q(v) = \emptyset$. It can also be readily verified that $Q(v) = \emptyset$ iff

$B(v) = \emptyset$. Recall that $B(v) \subseteq N(v)$; thus, if $N(v) = \emptyset$, then both $B(v)$ and $Q(v)$ are empty as well. A bag v with empty $Q(v)$ is an empty bag. Our algorithm essentially disregards empty bags. However, in our analysis we manipulate with the base and kernel sets of all bags, including empty ones.

2.2. The Base Edge Set

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

For each nonempty 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 $P(v) = (p_1, \ldots, p_k)$. We will show that $\Delta(\mathcal{B}) \leq 2$ and $\Psi(\mathcal{B}) = O(\ell)$.

Observe that the base point set $B(v)$ of a bag v may well be a *proper* subset of its native point set $N(v)$. This can happen if some of the children of v (or more generally, its descendants) become step-children of other bags. As a result the points $p_1, \ldots, p_k$ of $B(v)$ are not necessarily consecutive along the Hamiltonian path $\mathcal{L}$. Consequently, some of the edges of $P(v)$ may not belong to $\mathcal{L}$.

Fix an index j, $j \in [\ell]$. For each nonempty 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 definition of $B(v)$ ($B(v) = \bigcup_{z \in \mathcal{S}(v)} B(z)$), implies that the order relation $\prec_{\mathcal{L}}$ can be used in the obvious way to define a total order on the nonempty bags of $\mathcal{F}_j$.

OBSERVATION 2.3. *For any pair u, v of distinct nonempty 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 a detailed description of the way that the base edge set $\mathcal{B}$ is constructed.

On the bottom-most level ($j = 1$), for each bag $v \in \mathcal{F}_1$, we order all points of $B(v) = N(v)$ 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, that is, $\mathcal{B}_j = \bigcup_{v \in \mathcal{F}_j} \mathcal{B}(v)$. Finally, the *base edge set* $\mathcal{B}$ is formed as the union $\mathcal{B} = \bigcup_{j=1}^{\ell} \mathcal{B}_j$. (See Figure 1 for an illustration.)

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 $\mathcal{B}(v)$ of v. In other words, $\hat{\mathcal{B}}(v) = \mathcal{B}(v) \cup \bigcup_{i=1}^{h} \hat{\mathcal{B}}(c^{(i)}(v))$. The following lemma follows from the construction by a straightforward induction.

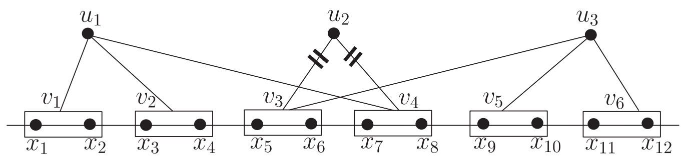

Fig. 1. In this example $\mathcal{L} = (x_1, x_2, \ldots, x_{12})$. In level 1 there are 6 bags, $v_1, \ldots, v_6$, with $v_i = (x_{2i-1}, x_{2i})$, for $i \in [6]$. In the original bag forest $\mathcal{F}$, u_i is the parent of v_{2i-1} and v_{2i}, for $i \in [3]$. In the adoption bag forest $\hat{\mathcal{F}}$, $v_4 \in \mathcal{J}(u_1)$ (i.e., v_4 is a step-child of u_1), and $v_3 \in \mathcal{J}(u_3)$. The bag u_2 becomes empty. The base edge sets are $\mathcal{B}_1 = \{(x_1, x_2), (x_3, x_4), \ldots, (x_{11}, x_{12})\}$ and $\mathcal{B}_2 = \{(x_2, x_3), (x_{10}, x_{11})\}$.

LEMMA 2.4. *Fix an arbitrary index $j \in [\ell]$, and let v be an arbitrary nonempty 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 2.4, the edge set of the path $P(v)$ is equal to the recursive base edge set $\hat{\mathcal{B}}(v)$ of 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 and index $j \in [\ell]$, 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 $1 \le i \le j$, and there are no i-level base bags for p, for all indices $j + 1 \le i \le \ell$. We will say that the base bags of p in levels $1, \ldots, j-1$ are *surviving*, and the base bag of p in level 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 $p \in Q$, $rightdeg_{\mathcal{B}}(p) \le 1$. Symmetrically, it also holds that $leftdeg_{\mathcal{B}}(p) \le 1$. We will conclude that $deg_{\mathcal{B}}(p) = leftdeg_{\mathcal{B}}(p) + rightdeg_{\mathcal{B}}(p) \le 2$, and thus $\Delta(\mathcal{B}) \le 2$.

LEMMA 2.5. *For every point $p \in Q$, $rightdeg_{\mathcal{B}}(p) \le 1$.*

Remark. Intuitively, this lemma holds because the base edges form disjoint paths, and at each level the only edges added join two such paths making them into longer paths. We now we substantiate this intuition with a formal proof.

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

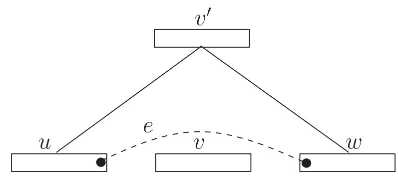

Fig. 2. The bag v is disintegrated from its parent v'. The dashed edge $e \in \mathcal{B}(v')$, viewed as an interval, contains all the intervals of the base edge set $\mathcal{B}(v)$ of v.

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 before.) 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 2.6. $\Delta(\mathcal{B}) \leq 2$ *and* $|\mathcal{B}| \leq n$.

Next we analyze the lightness $\Psi(\mathcal{B})$ of the base edge set $\mathcal{B}$. Observe that for each index $j \in [\ell]$, the j-level base edge set $\mathcal{B}_j$ is a collection of vertex-disjoint paths. By the triangle inequality, the weight $\omega(\mathcal{B}_j)$ of $\mathcal{B}_j$ is bounded above by the weight $L = \omega(\mathcal{L}) = O(t \cdot \omega(MST(M[Q])))$ 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 $\omega(\mathcal{B}) = \omega(\bigcup_{j=1}^{\ell} \mathcal{B}_j) \leq \sum_{j=1}^{\ell} \omega(\mathcal{B}_j) \leq \ell \cdot O(t \cdot \omega(MST(M[Q]))) = O(\ell \cdot t) \cdot \omega(MST(M[Q]))$.

Corollary 2.7. $\Psi(\mathcal{B}) = O(\ell \cdot t)$.

Remarks. (1) If the metric is low-dimensional Euclidean or doubling, then $\omega(\mathcal{L}) \leq 2 \cdot \omega(MST(M[Q]))$, and so $\Psi(\mathcal{B}) = O(\ell)$. (2) Note that for a bag v, its base edge set $\mathcal{B}(v)$ can be viewed as a collection of vertex-disjoint intervals with respect to the Hamiltonian path $\mathcal{L}$. However, for a pair of bags $v \in \mathcal{B}_j$, $v' \in \mathcal{B}_{j'}$, $j < j'$, their respective collections of intervals may well overlap. This can happen, for instance, if v' is a parent of three bags, u, v and w, which satisfy $u \prec_{\mathcal{L}} v \prec_{\mathcal{L}} w$. Suppose also that v is disintegrated from v', while u and w are surviving children of v'. Then the base edge set $B(v')$ of v' will contain an edge connecting the rightmost point in u with the leftmost point in w. This edge, viewed as an interval, will however contain all the intervals of $\mathcal{B}(v)$. See Figure 2 for an illustration.

2.3. Zombies and Incubators

Algorithm *LightSp* starts with computing the path-spanner $H = (Q, E_H)$ 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 submetric $M[Q_0]$ of M induced by $Q = Q_0$. Define $\tilde{E}_0$ to be the edge set obtained by *pruning* E'_0, that is, 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 (details will be provided in Section 2.6), the algorithm builds auxiliary j-level spanners $\tilde{G}_j = (Q_j, \tilde{E}_j)$, for each $j \in [\ell]$. The spanner $\tilde{G}_j$ is a graph over the set of representatives of the nonempty j-level bags. The representatives are determined

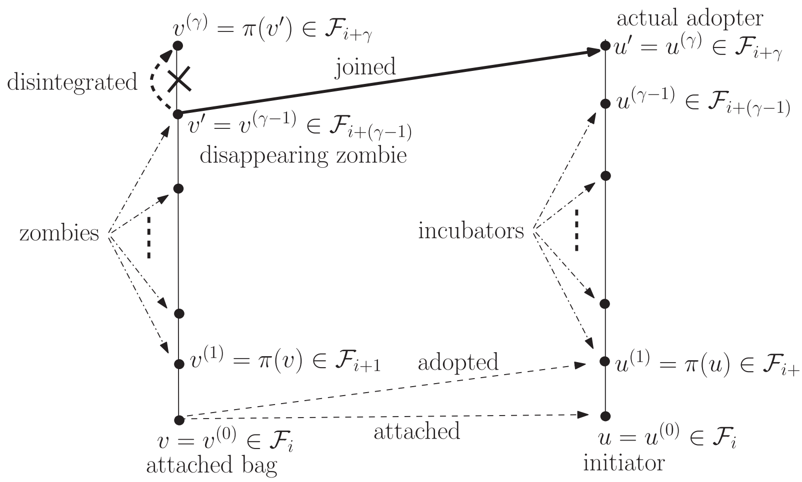

Fig. 3. An illustration of an attachment $\mathcal{A}(u, v)$.

according to rules that will be specified in Section 2.4. The union $\mathcal{B} \cup E_H \cup \bigcup_{j=0}^{\ell} \tilde{E}_j$ is the ultimate spanner $\tilde{G} = (Q, \tilde{E})$ that Algorithm *LightSp* returns.

As discussed earlier, during the algorithm an i'-level bag v' may join as a step-child of some $(i'+1)$-level bag u', $u' \neq \pi(v')$. We now take a closer look at this process.

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 γ, which determines the length of the incubation period (which lasts between levels i and $i + \gamma - 1$). The $(i+\gamma-1)$-level ancestor v' of v will actually be *disintegrated* from its parent $\pi(v')$, and *join* the $(i+\gamma)$-level ancestor u' of u. (The index i' mentioned before is actually equal to $i + \gamma - 1$.) The bag u' is referred to as the *actual adopter*. It will be shown later (see Corollary 3.5 in Section 3.1) that adoption rules (which we have not finished specifying) imply that $\pi(v') \neq u'$. We remark that attachments 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 2.1.

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 joins u' rather than 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. They will be now 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 actual 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 3 for an illustration.)

2.4. Representatives

In this section we specify how Algorithm *LightSp* selects representatives for bags.

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

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 (i.e., it has at least one neighbor) 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). (See Definition 2.1.) 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, that is, $load\_ctr_0(p) = CTR_0(p) = ctr_0(p) = plain\_ctr_0(p) = single\_ctr_0(p) = 0$.)

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 as follows.

The representative $r(v)$ of a nonempty 1-level bag $v \in \mathcal{F}_1$ is selected arbitrarily from $K(v) = Q(v)$. Next, consider a nonempty j-level bag, $j \in [2, \ell]$. The bag v is said to be a *growing bag* if $|\chi(v)| \ge 2$, that is, if v is obtained as a result of a merge of two or more nonempty $(j-1)$-level bags. Otherwise, the bag v is called *stagnating*. We will show in Lemma 3.2 that, 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, that is, $r(v) = r(w)$. 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 $r(v)$.

2.5. Procedure *Attach*

In this section we present a simple graph procedure, called Procedure *Attach*, which will be used as a building block by Algorithm *LightSp*. More specifically, this procedure is used to determine which bags will participate in attachments in the j-level processing. (The j-level processing will be described in Section 2.6.) It accepts as input an n-vertex graph $G = (V, E)$, whose vertices are labeled as either *safe* or *risky*. (The meaning of these labels will become clear in Section 2.6.) The procedure returns a *star forest*, that is, a collection Γ of vertex disjoint stars, that satisfies the following two conditions.

(1) Let R be the set of vertices in V that are not isolated[4] in G, and labeled as risky. Then $\bigcup_{S \in \Gamma} V(S) \supseteq R$.

[4]A vertex v in a (possibly directed) graph G is called *isolated* if no edge in G is incident on v.

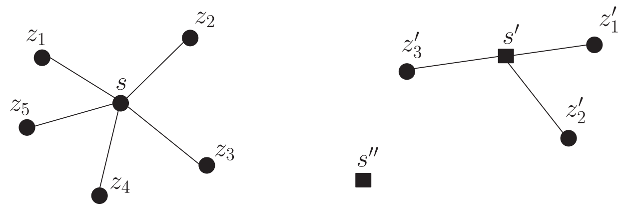

Fig. 4. A star forest. Safe bags are depicted by squares, while risky ones are depicted by circles.

(2) Each star $S \in \Gamma$ contains a *center* $s \in V$ 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 Figure 4 for an illustration.

Intuitively, Procedure *Attach* attaches each risky vertex to some other vertex. Each star of Γ will eventually be merged into a single supervertex in a certain supergraph in our algorithm. This will be, roughly speaking, our way to "get rid" of risky vertices. It is instructive to view each star center s as an attachment initiator, and leaves of the star centered at s as zombies that will eventually join an appropriate ancestor of s as its step-children.

Procedure *Attach* manipulates with three graphs: The input (undirected) graph G, an intermediate attachment digraph $\mathcal{G}$, and a collection Γ of undirected stars. Procedure *Attach* consists of two phases. In the first phase it creates the attachment digraph $\mathcal{G}$ from the input graph G, and in the second phase it extracts the stars of Γ from the digraph $\mathcal{G}$. Next we describe these two phases. The attachment digraph $\mathcal{G}$ is created as follows: for every nonisolated 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 $\mathcal{G}$. (Note that the vertex set of $\mathcal{G}$ is V rather than just R.)

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

The second phase of Procedure *Attach* (i.e., extracting the collection Γ of stars from the attachment digraph $\mathcal{G}$) proceeds in two stages. The first stage is carried out iteratively. At each iteration the procedure picks an arbitrary nonisolated vertex z in the attachment digraph $\mathcal{G}$ with in-degree zero, and handles it as follows. Since z is nonisolated, it must have an outgoing neighbor s; since only vertices of R have outgoing neighbors in $\mathcal{G}$, we conclude that 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 the star 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 nonisolated vertices in $\mathcal{G}$ have in-degree at least one. Let V' be the set of nonisolated 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'. Denote by E' the edge set of $\mathcal{G}'$. If E' 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 $|E'| \geq |V'|$. Also, the out-degree of each vertex in $\mathcal{G}'$ is at most one, and so $|E'| = |V'|$. 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.

Finally, we ignore the orientation of edges, that is, the resulting star forest Γ is viewed as an undirected graph.

This completes the description of the procedure *Attach*.

It is easy to verify that the graph Γ constructed by Procedure *Attach* is a star forest that satisfies the two conditions listed before. Also, it is straightforward to implement this procedure in time $O(|V|)$.

COROLLARY 2.8. *Procedure Attach, given a graph $G = (V, E)$ whose vertices are labeled as either safe or risky, produces a star forest that satisfies conditions 1 and 2 (listed in the beginning of this section). The running time of this procedure is $O(|V|)$.*

2.6. *j*-Level Processing

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

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

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, that is, 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 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 $q \in Q_j$ that 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 nonempty bag v is called *safe* if it satisfies at least one of the following three conditions: (1) v is large, (2) v is crowded, (3) v is an incubator or 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

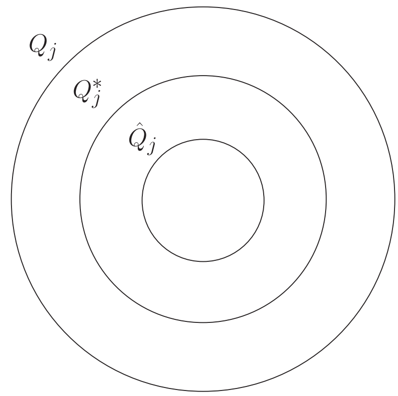

Fig. 5. Subsets of Q_j that are used during the j-level processing. Q_j^* is the subset of Q_j of points with positive degree in G_j^*, and $\hat{Q}_j$ is the subset of Q_j^* of nonzombie representatives.

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.

Intuitively, for a safe bag v there is no danger that one of the points $p \in Q(v)$ will become overloaded, that is, that its degree in the spanner will be too large. Indeed, if v is a large bag, then it contains many points which can share the load. If v is a crowded bag or an incubator or a zombie, then it will soon be merged with at least one other nonempty bag u, and through this merge it will acquire additional points that can participate in sharing the load. If v is a crowded bag, then u is a bag that belongs to v's cage. Otherwise v is an incubator or a zombie. In this case u does not belong to v's cage. However, in either case, we will argue later that any point in $Q(v)$ is quite close to any point in $Q(u)$ in the original metric M. Moreover, these points will also stay close in the spanner.

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. (See Figure 5 for an illustration.) 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 $\check{G}_j$, that is, 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* $G_j = (\hat{Q}_j, \mathcal{E}_j)$, which is the restriction of the j-level auxiliary spanner $\tilde{G}_j$ to the set $\hat{Q}_j$, that is, $\mathcal{E}_j = \tilde{E}_j(\hat{Q}_j) = \hat{E}_j \cup E_j^*(\hat{Q}_j)$. (Note that all points of Q_j, and so 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 2.8 (see Section 2.5), this procedure returns a star forest Γ_j that covers R_j (i.e., $R_j \subseteq \bigcup_{S \in \Gamma_j} V(S)$), where $R_j \subseteq \hat{Q}_j$ is the set of all risky points in $\hat{Q}_j$. (Note that no point of R_j is 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.

Intuitively, risky bags cannot be left on their own, because the degrees of their points will inevitably explode. (See Section 1.3.) Hence the algorithm merges them either with one another, or with some safe bags. The attachment graph G_j is used to determine which bags will merge. A special care is taken to exclude zombie representatives from G_j. Recall that a zombie bag is already on its way to be merged with some other bag. If another bag were merged into a zombie bag, this would ultimately lead to the creation of "zombie paths", that is, paths $(z_1, z_2, \ldots, z_h)$ of zombie bags, where z_1 merges into $z_2, \ldots, z_{h-1}$ merges into z_h. This would result in an uncontrolled growth of the diameter.

Next, Part II of Procedure $Process_j$ performs attachments. 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, the $\gamma - 1$ immediate ancestors $v^{(1)}(s) = \pi(v(s)), \ldots, v^{(\gamma-1)}(s)$ of $v(s)$ (in $\mathcal{F}$) are labeled as incubators, and the $\gamma - 1$ immediate ancestors $v^{(1)}(q_i), \ldots, v^{(\gamma-1)}(q_i)$ of $v(q_i)$ (in $\mathcal{F}$), for each $i \in [k]$, are labeled as zombies. Note that $v^{(\gamma-1)}(q_i)$ is a disappearing zombie, for each $i \in [k]$, 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 auxiliary spanner $\tilde{G}_j$, and consequently, 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 *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 step-children, respectively, for every bag $v \in \mathcal{F}_{j+1}$. This is done according to the set of attachments which were computed in previous levels. In particular, a child w of v which joins 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, joins the set $\mathcal{J}(v)$. Given the sets $\mathcal{S}(v)$ and $\mathcal{J}(v)$, Part III of Procedure $Process_j$ computes the sets $B(v), K(v), Q(v)$ according to the rules specified in Section 2.1, and computes the representative $r(v)$ of v according to the rules specified in Section 2.4. 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*.

Section 3.1 focuses on properties of zombies and incubators. In Section 3.2 we analyze the number of edges and lightness of our spanners, as well as the running time of our algorithm. In Section 3.3 we analyze the stretch and diameter of our spanners, and in Section 3.4 we study their degree.

The incubation period γ is set as $\gamma = c_0 \cdot (\lceil \log_\rho t \rceil + \lceil \log_\rho c \rceil + 1)$, for a sufficiently large constant c_0. (Recall that $c = \Theta(t/\epsilon)$, hence $\gamma = \Theta(\max\{1, \log_\rho(t/\epsilon)\})$.)

3.1. Zombies and Incubators

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

A j-level bag v is an *attached bag* if it is unlabeled, and is adopted during the execution of (part II of) Procedure $Process_j$. Observe that an attached bag v must be lonely, that is, the cage $\mathcal{C}(v)$ does not contain any useful bags. In other words, all the nonempty 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. (Recall that the only possible labels are "incubator" and "zombie". On the other hand, a bag labeled as an incubator or a zombie is safe, and therefore will not be attached.) Hence an attached bag v is *never* labeled as 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 3.4 that there exists exactly one such path.)

LEMMA 3.1. *Fix an arbitrary index $j \in [\ell]$, and let v be a nonempty j-level bag. Then (1) if v is not labeled as a zombie, there is a path Υ_v of nonempty 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. Assuming the correctness of the statement for all index values smaller than j, for some $j \in [2, \ell]$, we prove its correctness for index value j.

We start with proving the first assertion. Let v be a nonempty j-level bag which is not labeled as a zombie.

Suppose for contradiction that all nonempty 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 nonempty 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 nonempty 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'. Since v' is an attached bag, it must be lonely. (See the beginning of Section 3.1.) Hence, all the nonempty 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 nonempty 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 nonempty bag z' which is not labeled as a zombie. However, z' belongs to $\mathcal{C}(v')$, yielding a contradiction. See Figure 6 for an illustration. Therefore, all the nonempty children of v must be labeled as zombies, and

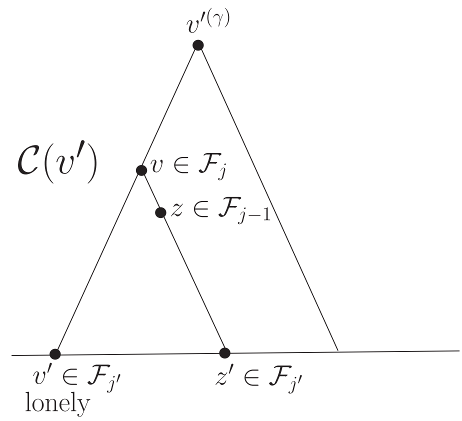

Fig. 6. z and z' are not zombies, and $v', z' \in \mathcal{C}(v')$. Hence v' is not lonely, a contradiction.

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. □

The following lemma is a useful property of stagnating bags.

LEMMA 3.2. *If v is a stagnating bag, then $|\mathcal{S}(v)| = 1$, $\mathcal{J}(v) = \emptyset$.*

PROOF. Since v is a stagnating bag, we have $|\mathcal{S}(v)| \leq |\chi(v)| = 1$. Suppose for contradiction that $\mathcal{S}(v) = \emptyset$. In this case all children of v are disappearing zombies, and thus, by Lemma 3.1, none of them is an incubator. Hence v is not an actual adopter, that is, $\mathcal{J}(v) = \emptyset$. However, this contradicts the assumption that v is a stagnating bag (i.e., $|\chi(v)| = 1$). It follows that $|\mathcal{S}(v)| = 1$, $\mathcal{J}(v) = \emptyset$. □

We use the next claim to prove Lemma 3.4.

CLAIM 3.3. *Fix an arbitrary index $j \in [\gamma, \ell]$, and let v be a nonempty 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 $j - \gamma + 1 \leq j' \leq j$. 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 attached j'-level bag v'. As the hop-distance of this path is at most $\gamma - 1$, we have $j' \geq j - \gamma + 1$. By construction, to become an attached bag, v' must be useful, as required.

Consequently, Lemma 3.1 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 3.4. *Fix an arbitrary index $j \in [\gamma, \ell - 1]$, and let v be a nonempty $(j + 1)$-level bag. If v has a zombie child z, then all its other children are empty and it has no step-children, that is, $\mathcal{S}(v) = \{z\}$, $\mathcal{J}(v) = \emptyset$.*

PROOF. Suppose for contradiction that v has a nonempty 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 3.3. 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 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 become attached bags. Any other useful i-level descendant of v is not lonely, and thus it is safe as well, for each index $i = j', j'+1, \ldots, j$, hence it does not become an attached bag. It follows 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 incubator. 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 3.4 implies the following corollary.

COROLLARY 3.5. *Fix an arbitrary index $j \in [\ell - 1]$, and let v be a (nonempty) j-level bag which is a disappearing zombie. (Notice that $j \geq \gamma$.) 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), that is, $\pi(v) \neq v'$.*

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 $Q(w) = Q(w')$.

LEMMA 3.6. *Let $w \in \mathcal{F}_j$ be a disappearing zombie. (Hence $j \geq \gamma$.) Then there exists a unique useful descendant $\tilde{w} \in \mathcal{F}_{j-(\gamma-1)}$ of w in $\mathcal{F}$, 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 $\gamma - 1$ zombie bags $\tilde{w}^{(i)} \in \mathcal{F}_{j-(\gamma-1)+i}$ along the path between $\tilde{w}^{(1)}$ and $w = \tilde{w}^{(\gamma-1)}$, $i \in [\gamma - 1]$, 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 $w = \tilde{w}^{(\gamma-1)}$, that is, $\tilde{w}$ 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: Assuming the correctness of the statement for all index values smaller than i, for some $i \in [\gamma - 1]$, we prove its correctness for index value i. By the induction hypothesis, the bag $\tilde{w}^{(i-1)}$ is identical to $\tilde{w}$, that is, $Q(\tilde{w}^{(i-1)}) = Q(\tilde{w})$. Also, since the bag $\tilde{w}^{(i-1)}$ is a zombie child of $\tilde{w}^{(i)}$, Lemma 3.4 yields $\mathcal{S}(\tilde{w}^{(i)}) = \{\tilde{w}^{(i-1)}\}$, $\mathcal{J}(\tilde{w}^{(i)}) = \emptyset$. By construction,

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

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 3.6, 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 nonempty $(j-(\gamma-1))$-level descendant of w.

3.2. Number of Edges, Weight, and Running Time

In this section we analyze the number of edges, weight, and running time, of the spanner $\tilde{G} = (Q, \tilde{E})$ computed by Algorithm *LightSp*.

3.2.1. Auxiliary Statements. We start with providing a few auxiliary lemmas. They will be used in the analysis of the number of edges, weight, and running time of our construction. (See the beginning of Section 2 for the definitions of $n_j, \tau_j, c, \ell, \rho$ and L.)

OBSERVATION 3.7. *Let f be a monotone nondecreasing convex function that vanishes at zero, and let $n'_1, n'_2, \ldots, n'_\ell$ be a sequence of positive numbers that satisfy that $n'_j \le \min\{n, n_j\}$, for each index $j \in [\ell]$. Then for each index $j \in [\ell]$, $f(n'_j) \le \frac{c}{\rho^{j-1}} \cdot f(n)$. Moreover, for each index $1 \le j < \log_\rho c + 1$, $f(n'_j) \le f(n) < \frac{c}{\rho^{j-1}} \cdot f(n)$.*

PROOF. Suppose first that $1 \le j < \log_\rho c + 1$; in this case, we have $\frac{c}{\rho^{j-1}} > 1$, and so $n < \frac{c \cdot n}{\rho^{j-1}} = n_j$. It follows that $n'_j \le n$, which yields $f(n'_j) \le f(n) < \frac{c}{\rho^{j-1}} \cdot f(n)$. We henceforth assume that $\log_\rho c + 1 \le j \le \ell$.

In this case, we have $\frac{c}{\rho^{j-1}} \le 1$, and so $n \ge \frac{c \cdot n}{\rho^{j-1}}$. It follows that $n'_j \le n_j = \frac{c \cdot n}{\rho^{j-1}}$. Also, the assumptions about f imply that $f(\frac{c \cdot n}{\rho^{j-1}}) \le \frac{c}{\rho^{j-1}} \cdot f(n)$. We conclude that $f(n'_j) \le f(\frac{c \cdot n}{\rho^{j-1}}) \le \frac{c}{\rho^{j-1}} \cdot f(n)$. $\square$

Recall that $|Q_j| \le \min\{n, n_j\} = \min\{n, \frac{c \cdot n}{\rho^{j-1}}\}$. Observation 3.7 implies the following corollary.

COROLLARY 3.8. *For any monotone nondecreasing convex function f that vanishes at zero:*
(1) $\sum_{j=1}^{\ell} f(|Q_j|) = O(f(n) \cdot \max\{1, \log_\rho(t/\epsilon)\})$, and (2) $\sum_{j=1}^{\ell} f(|Q_j|) \cdot \tau_j = O(\frac{f(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon) \cdot L$.

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

$$\begin{aligned}
\sum_{j=1}^{\ell} f(|Q_j|) &= \sum_{1 \le j < \log_\rho c + 1, j \in \mathbb{N}} f(|Q_j|) + \sum_{\log_\rho c + 1 \le j \le \ell, j \in \mathbb{N}} f(|Q_j|) \\
&\le (\log_\rho c + 1) \cdot f(n) + \sum_{\log_\rho c + 1 \le j \le \ell, j \in \mathbb{N}} \frac{c}{\rho^{j-1}} \cdot f(n) \\
&= (\log_\rho c + 1) \cdot f(n) + \sum_{\lceil \log_\rho c \rceil + 1 \le j \le \ell, j \in \mathbb{N}} \frac{c}{\rho^{j-1}} \cdot f(n) \\
&= (\log_\rho c + 1) \cdot f(n) + \sum_{0 \le j \le \ell - \lceil \log_\rho c \rceil - 1, j \in \mathbb{N}} \frac{c}{\rho^{(j + \lceil \log_\rho c \rceil + 1) - 1}} \cdot f(n) \\
&= (\log_\rho c + 1) \cdot f(n) + \sum_{0 \le j \le \ell - \lceil \log_\rho c \rceil - 1, j \in \mathbb{N}} \frac{c}{\rho^j \cdot \rho^{\lceil \log_\rho c \rceil}} \cdot f(n) \\
&\le (\log_\rho c + 1) \cdot f(n) + \sum_{j=0}^{\infty} \frac{1}{\rho^j} \cdot f(n) \;=\; O(f(n) \cdot \max\{1, \log_\rho(t/\epsilon)\}).
\end{aligned}$$

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,

$$\sum_{j=1}^{\ell} f(|Q_j|) \cdot \tau_j \le \sum_{j=1}^{\ell} \frac{c}{\rho^{j-1}} \cdot f(n) \cdot O\left(\rho^j \cdot \frac{L}{n} \cdot t\right) = O\left(\frac{f(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon\right) \cdot L. \quad \square$$

3.2.2. Number of Edges. In this section we bound the number of edges in $\tilde{G}$.

LEMMA 3.9. $|\tilde{E}| = O(SpSz(n) \cdot \max\{1, \log_\rho(t/\epsilon)\})$.

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

The path-spanner H contains at most $O(n)$ edges, that is, $|E_H| = O(n)$, and the graph $\tilde{G}_0 = (Q_0 = Q, \tilde{E}_0)$ contains at most $SpSz(n)$ edges, that is, $|\tilde{E}_0| \le SpSz(|Q_0|) = SpSz(n)$. Also, as shown in Section 2.2 (see Corollary 2.6), 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 nondecreasing convex function that vanishes at zero. Consequently,

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

(The last inequality follows from the first assertion of Corollary 3.8.) $\square$

3.2.3. Weight. In this section we bound the weight of $\tilde{G}$.

LEMMA 3.10. $\omega(\tilde{G}) = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon) \cdot L$.

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 2.2 (see Corollary 2.7), 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 the 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

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

(The last inequality follows from the second assertion of Corollary 3.8.) □

In Theorem 1.4 we assumed the existence of Algorithm *BasicSp* that constructs a t-spanner for any submetric $M[Q]$ of M (including M itself) with certain properties. This algorithm can be used to construct a t-approximate MST for M. Specifically, running Prim's MST Algorithm over this t-spanner results in a t-approximate MST. The weight $L = \omega(\mathcal{L})$ of the Hamiltonian path $\mathcal{L}$ which is computed in this way is $O(t \cdot \omega(MST(M[Q])))$, and the running time of this computation is $SpTm(n) + O(SpSz(n) + n \cdot \log n) = O(SpTm(n) + n \cdot \log n)$. Alternatively, we may add to the statement of Theorem 1.4 an assumption that it is provided with Algorithm *LightTree*, which computes an $O(1)$-approximate MST for $M[Q]$ within time $TrTm(n)$. In particular, in low-dimensional Euclidean and doubling metrics running Prim's Algorithm over an $O(1)$-spanner results in a routine that computes an $O(1)$-approximate MST within $O(n \cdot \log n)$ time. In these cases (i.e., if we are provided with Algorithm *LightTree* or if the input metric is doubling), $L = \omega(\mathcal{L}) = O(\omega(MST(M[Q])))$.

To summarize, we give the following corollary.

COROLLARY 3.11. *In the variant of Algorithm LightSp which employs Algorithm LightTree (or if M is a low-dimensional Euclidean or doubling metric),* $\omega(\tilde{G}) = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^2/\epsilon) \cdot \omega(MST(M[Q]))$. *In the variant of Algorithm LightSp which does not employ it,* $\omega(\tilde{G}) = O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot t^3/\epsilon) \cdot \omega(MST(M[Q]))$.

3.2.4. Running Time. In this section we analyze the running time of Algorithm *LightSp*.

LEMMA 3.12. *The variant of Algorithm LightSp that invokes (respectively, does not invoke) Algorithm LightTree can be implemented in time* $O(SpTm(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + TrTm(n))$ *(resp.,* $O(SpTm(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + n \cdot \log n)$*).*

PROOF. The tree T can be built within $O(TrTm(n))$ time by Algorithm *LightTree*, and within $O(SpTm(n) + n \cdot \log n)$ without it. In the former case T will have constant lightness. In the latter a t-approximate MST T for $M[Q]$ is built as outlined in Section 3.2.3. In another $O(n) = O(TrTm(n))$ time we can compute the preorder traversal of T, thus obtaining the Hamiltonian path $\mathcal{L}$. The 1-spanner $H_{\mathcal{L}}$ can be built in $O(n)$ time. The path-spanner H can be obtained from $H_{\mathcal{L}}$ in another $O(n)$ time. Also, it is easy to see that the graph $\tilde{G}_0$ can be built within $O(SpTm(n))$ time.

In each level $j = 1, \ldots, \ell$, 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 2.6). By similar considerations, the graph $\hat{G}_j$ can be built in time $O(SpTm(|\hat{Q}_j|)) = O(SpTm(|Q_j|))$; also, the j-level 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 $\mathit{Process}_j$). Updating the load indicators and counters of representatives requires another $O(\mathit{SpSz}(|Q_j|)) = O(\mathit{SpTm}(|Q_j|))$ time. By Corollary 2.8 (see Section 2.5), Procedure *Attach* runs in time that is linear in the number of vertices of the attachment graph G_j, that is, in time $O(|Q_j|) = O(\mathit{SpTm}(|Q_j|))$. The number of nonempty 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 $O(n \cdot \gamma) = O(n \cdot \max\{1, \log_\rho(t/\epsilon)\})$. 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 levels is at most $\sum_{j=1}^{\ell} O(\mathit{SpTm}(|Q_j|)) + O(n \cdot \max\{1, \log_\rho(t/\epsilon)\}) \le O(\mathit{SpTm}(n) \cdot \max\{1, \log_\rho(t/\epsilon)\})$. (Recall that $\mathit{SpTm}(\cdot)$ is a monotone nondecreasing convex function that vanishes at zero, and see the first assertion of Corollary 3.8.)

On level j, determining which bags are crowded requires $O(|Q_j|)$ time. Hence, by Corollary 3.8, in all ℓ levels altogether this step requires $\sum_{i=1}^{\ell} O(|Q_j|) = O(n \cdot \max\{1, \log_\rho(t/\epsilon)\})$ time.

Altogether, the variant of Algorithm *LightSp* that uses Algorithm *LightTree* takes time $O(\mathit{SpTm}(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + \mathit{TrTm}(n))$. The variant of Algorithm *LightSp* that does not use Algorithm *LightTree* takes time $O(\mathit{SpTm}(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + n \cdot \log n)$. $\square$

Remark. In the case of low-dimensional Euclidean or doubling metrics, $\mathit{TrTm}(n) = O(n \cdot \log n)$, and so the running time becomes $O(\mathit{SpTm}(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + n \cdot \log n)$.

3.3. Stretch and Diameter

In this section we analyze the stretch and diameter of the spanner $\tilde{G}$.

The following lemma is central to our analysis.

LEMMA 3.13. *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 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 in $\Pi_j(p)$ belong to the point set $Q(v)$ of v. (The point $b_j(p)$ is called the j-level* base point *of p.)*

PROOF. The proof is by induction on j. The basis $j = 1$ is immediate since $B(v) = Q(v)$.

Induction Step. Assuming the correctness of the statement for all index values smaller than j, for some $j \in [2, \ell]$, we prove its correctness for index value j. Let $p \in Q(v)$, and let $u \in \chi(v) \subseteq \mathcal{F}_{j-1}$ be the $(j-1)$-level host bag of p.

Suppose first that $u \in \mathcal{S}(v)$, that is, u is a surviving child of v in $\mathcal{F}$. In this case, $B(u) \subseteq B(v)$, $K(u) \subseteq K(v)$, $Q(u) \subseteq Q(v)$. (See Section 2.1 to recall the basic properties of these point sets.) Consider the path $\Pi_{j-1}(p)$ between p and its $(j-1)$-level 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 $\Pi_j(p) = \Pi_{j-1}(p)$ and $b_j(p) = b_{j-1}(p)$.

We henceforth assume that u is a disappearing zombie, that is, $u \in \mathcal{J}(v)$ is a step-child of v. In this case, since $v \in \mathcal{F}_j$ is an actual adopter, it must hold that $j \ge \gamma + 1$. 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 $Q(y^{(i)}) = Q(u)$. See Lemma 3.6.) In particular, $u = y^{(\gamma-1)}$ is a disappearing zombie, and $y^{(0)} = y$ is an

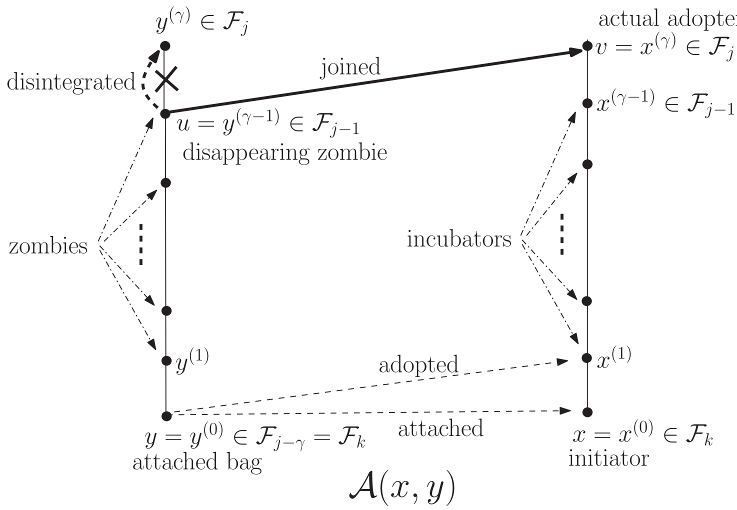

Fig. 7. u is a disappearing zombie, a step-child of (the actual adopter) v.

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, and the corresponding disappearing zombie is $y^{(\gamma-1)} = u$. 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 $\tilde{G}$. (Recall that $r(x)$ and $r(y)$ are the representatives of the bags x and y, respectively.) See Figure 7 for an illustration.

We will use the following claim to prove Lemma 3.13.

Claim 3.14. *Define $k = j - \gamma$. There is a 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 $Q(y) = Q(u) \subseteq Q(v)$.*

Proof. 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, that is, $p, r(y) \in Q(y)$. 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 subpath of $P(y)$ between $b_k(p)$ and $b_k(r(y))$; by the triangle inequality, the weight of this path is at most $\delta_{\mathcal{L}}(b_k(p), b_k(r(y))) \le \mu_k = \frac{\mu_j}{\rho^{\gamma}}$. We set $\Pi(p, r(y)) = \Pi_k(p) \circ \Pi(b_k(p), b_k(r(y))) \circ \Pi_k(r(y))$. (We assume that $\Pi(p, r(y))$ is a simple path. Otherwise we transform it into such by eliminating loops.) 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 $2 \cdot \mu_k = 2 \cdot \frac{\mu_j}{\rho^{\gamma}}$. 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 $|Q(y)| \le \ell - 1$. Since $\Pi(p, r(y))$ is a simple path, it consists of at most $\ell - 2$ edges, which completes the proof of Claim 3.14. See Figure 8 for an illustration. $\square$

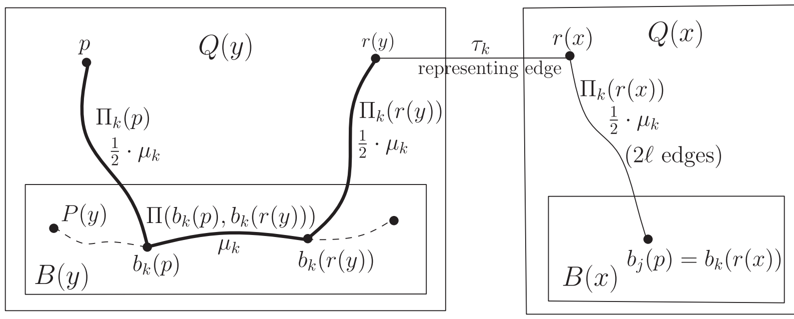

Fig. 8. The path $\Pi(p, r(y))$ is depicted by a bold solid line. It is a subpath of the path $\Pi_j(p)$, which connects p with $b_j(p)$, and is depicted by a solid line.

Observe that at this point we have built a "good path" $\Pi(p, r(y)) \circ (r(y), r(x))$ from p to $r(x)$. We now need to "connect" $r(x)$ to a point $b_j(p) \in B(v)$, which will be designated as the j-level base point of p. Observe that all bags $x = x^{(0)}, x^{(1)} = \pi(x^{(0)}), \ldots, v = x^{(\gamma)} = \pi(x^{(\gamma-1)})$ along the path in $\mathcal{F}$ between the attachment initiator x and the actual adopter $v = x^{(\gamma)}$ (which is an ancestor of x in $\mathcal{F}$) are not zombies. In particular, none of these bags is a disappearing zombie. It follows that $B(x) \subseteq B(v)$, $K(x) \subseteq K(v)$, $Q(x) \subseteq Q(v)$. Also, since 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 k-level 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 $r(x) \in K(x)$, it consists of at most 2ℓ edges. We set $b_j(p) = b_k(r(x)) \in B(v)$, and $\Pi_j(p) = \Pi(p, r(y)) \circ (r(y), r(x)) \circ \Pi_k(r(x))$. (See Figure 8.) It is easy to see that $\Pi_j(p)$ is a path between p and its j-level base point $b_j(p) = b_k(r(x))$, and that all points of $\Pi_j(p)$ belong to $Q(v)$. Notice that $\omega(r(y), r(x)) \leq \tau_k$. Therefore, the total weight $\omega(\Pi_j(p))$ of the path $\Pi_j(p) = \Pi(p, r(y)) \circ (r(y), r(x)) \circ \Pi_k(r(x))$ satisfies (for sufficiently large c_0)

$$\omega(\Pi_j(p)) \;\leq\; 2\mu_k + \tau_k + \frac{1}{2} \cdot \mu_k \;=\; \mu_j \cdot \frac{\left(\frac{5}{2} + 2 \cdot (c+1) \cdot \rho \cdot t\right)}{(c \cdot \rho \cdot t)^{c_0}} \;<\; \frac{1}{2} \cdot \mu_j.$$

(Recall that c_0 is a sufficiently large constant of our choice. Setting $c_0 \geq 8$ is enough here.) 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 is a small bag. (This case is characterized by $u \in \mathcal{J}(v)$, $p \in Q(u) \cap K(v)$.) 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

$$K'(v) \;=\; \bigcup_{z \in \mathcal{S}(v)} K(z) \;\supseteq\; K(x^{(\gamma-1)}) \;\supseteq\; K(x).$$

By Lemma 2.2, $|K'(v)| \;\geq\; |K(x)| \;\geq\; \ell$. 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 step-children. However, $p \in Q(u)$, and u is a step-child of v. Hence $p \notin K(v)$, a contradiction.

Therefore x is a small bag, and so $|Q(x)| < \ell$. We may assume that $\Pi_k(r(x))$ is a simple path. Since all points of $\Pi_k(r(x))$ belong to $Q(x)$, this path consists of at most $\ell - 2$ edges (rather than at most 2ℓ edges as in the general case). Hence, $|\Pi_j(p)| \;=\; |\Pi(p, r(y))| + 1 + |\Pi_k(r(x))| \;\leq\; (\ell - 2) + 1 + (\ell - 2) \;\leq\; 2\ell$.

Lemma 3.13 implies the following corollary.

COROLLARY 3.15. *Fix an arbitrary index $j \in [\ell]$, and let v be an arbitrary nonempty 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 $2 \cdot \mu_j$.*

PROOF. 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 3.13, having weight at most $\frac{1}{2} \cdot \mu_j$ and at most $3\ell = 3\lceil \log_\rho n \rceil$ 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 $\Pi(p, q) = \Pi_j(p) \circ \Pi_H(b_j(p), b_j(q)) \circ \Pi_j(q)$. 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 $3\lceil \log_\rho n \rceil + O(\log_\rho n + \alpha(\rho)) + 3\lceil \log_\rho n \rceil = O(\log_\rho n + \alpha(\rho))$ edges. □

The next lemma implies that $\tilde{G}$ is a $(t + \epsilon)$-spanner for $M[Q]$ with diameter $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$. Recall that $\Lambda(n)$ is an upper bound on the diameter of the auxiliary spanners, produced by Algorithm *BasicSp*. (See the statement of Theorem 1.4.)

LEMMA 3.16. *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.

OBSERVATION 3.17. *Fix any index $j \in [0, \ell]$. For any pair $u, v \in \mathcal{F}_j$ of nonempty 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.*

PROOF. Since u and v are nonempty 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 2.6), 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^*. Observation 3.17 follows. □

Next, we continue with the proof of Lemma 3.16.

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 3.17 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 $\rho^{j-1} \cdot \frac{L}{n} < \delta(p, q) \leq \rho^j \cdot \frac{L}{n}$, that is, $\xi_j < \delta(p, q) \leq \rho \cdot \xi_j$. Let $u = v_j(p)$ (respectively, $w = v_j(q)$) be the j-level host bag of p (resp., q). By Corollary 3.15, the metric distance between every pair of points in the same j-level bag is at most $2 \cdot \mu_j < \xi_j$. (See the beginning of Section 2 for the definitions of μ_j and ξ_j, and for other relevant notation.) 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 3.15, $\delta(p, r(u)), \delta(q, r(w)) \le 2 \cdot \mu_j = 2 \cdot \frac{\xi_j}{c}$. It follows that

$$\begin{aligned}\delta(r(u), r(w)) &\le \delta(p, r(u)) + \delta(p, q) + \delta(q, r(w)) \le \delta(p, q) + 4 \cdot \frac{\xi_j}{c} \qquad (1)\\ &\le \rho \cdot \xi_j + 4 \cdot \frac{\xi_j}{c} = 2\rho^j \cdot \frac{L}{n} \cdot \left(\frac{1}{2} + \frac{2}{\rho \cdot c}\right) \le 2\rho^j \cdot \frac{L}{n} \cdot \left(1 + \frac{1}{c}\right) = \frac{\tau_j}{t}.\end{aligned}$$

By Observation 3.17, 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 $\omega(\Pi^*(r(u), r(w))) \le t \cdot \delta(r(u), r(w))$. Also, by Corollary 3.15, 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), $t \cdot \delta(r(u), r(w)) \le t \cdot (\delta(p, q) + 4 \cdot \frac{\xi_j}{c})$. Also, recall that $c = \lceil \frac{4 \cdot (t+1)}{\epsilon} \rceil$. It follows that

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

Hence, $\Pi(p, q)$ is a $(t+\epsilon)$-spanner path in $\tilde{G}$ between p and q with $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$ edges.

3.4. Degree

In this section 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 $\gamma = O(\max\{1, \log_\rho (t/\epsilon)\})$. In other words, we will get the desired degree bound of $O(\Delta(n) \cdot \max\{1, \log_\rho (t/\epsilon)\} + \rho)$.

As shown in Section 2.2 (Corollary 2.6), the base edge set $\mathcal{B}$ increases the degree bound by at most two units, and so we may disregard it in this analysis. We will also disregard 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.

The degree analysis is probably the most technically involved part of our proof. We start with an intuitive sketch and then proceed to the rigorous proof.

Our algorithm makes a persistent effort to merge small bags together to form large bags. Intuitively, a large bag is easy to handle because its kernel set contains enough (at least ℓ) points to share the load.

If a j-level bag v is large, that is, $|Q(v)| \ge \ell$, then all its $\ell - j$ ancestors are large as well. Moreover, by Lemma 2.2, the kernel set $K(v)$ of v contains at least ℓ points. A point $p \in Q(v)$ may get 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. (In particular, the only points of $Q(v)$ that may get loaded belong to the kernel set $K(v)$.) However, we have at least ℓ points in $K(v) \subseteq Q(v)$ that can be used 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 $p \in K(v)$ 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 intuitive 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.)

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 2.6). It means that there exists another j-level representative $r(u)$, such that $\delta(r(v), r(u)) \le \tau_j$; 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, that is, they grow geometrically with the level j. Hence when v is merged into u', it contributes only an $\exp(-\Omega(\gamma))$-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.

We next proceed to the rigorous analysis of the degree of the spanner $\tilde{G}$.

Consider an index $j \in [\ell]$. In the next paragraph we provide a brief recap of how the j-level auxiliary spanner $\tilde{G}_j$ is constructed. Procedure $Process_j$ builds a spanner G'_j for the set Q_j of representatives of all nonempty j-level bags, including zombies. This spanner is then pruned to obtain the graph $G_j^* = (Q_j, E_j^*)$. (By "pruning" we mean removing edges of weight greater than τ_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 (i.e., nonempty and non-zombie) 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 2.6 for more details.

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

OBSERVATION 3.18. *For each index $j \in [\ell]$, $\Delta(\tilde{G}_j) = O(\Delta(n))$.*

Observation 3.18 implies directly that $\Delta(\tilde{G}) = O(\log n \cdot \Delta(n) + \rho)$. Such a bound on the maximum degree can, in fact, be achieved by a much simpler construction. (See Section 1.3 for its outline.) In this section we show that our much more intricate construction guarantees $\Delta(\tilde{G}) = O(\Delta(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + \rho)$. As t is typically a constant, and ρ and ϵ can be set as constants, this would essentially imply that $\Delta(\tilde{G}) = O(\Delta(n))$.

Definition 3.19. A j-level bag v is called *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, that is, $load_j(r(v)) = 0$. On the other hand, for an active bag v, its representative *is loaded*, that is, $load_j(r(v)) = 1$. Recall that v is a growing bag if $|\chi(v)| \ge 2$. (See Section 2.4.)

Recall also (see the beginning of Section 2) that the relation step-parent - step-child among bags of the original bag forest $\mathcal{F}$ defines another forest $\hat{\mathcal{F}}$ over the same set of bags (called the adoption bag forest). Specifically, a bag v is a parent of u in the adoption bag forest $\hat{\mathcal{F}}$ iff $u \in \chi(v)$, that is, u is an extended child of v (either a surviving child or a step-child of v). We denote the parent-child relation in the adoption bag forest $\hat{\mathcal{F}}$ by

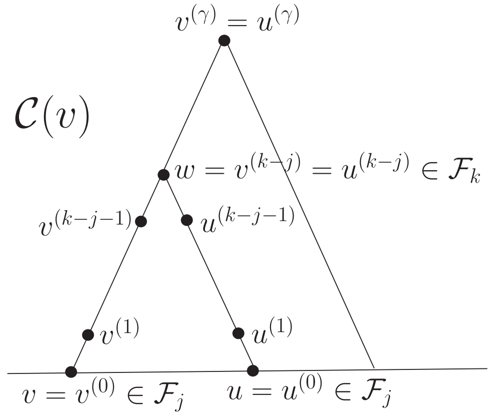

Fig. 9. The cage $\mathcal{C}(v)$ of v and u. All vertices $v^{(0)}, v^{(1)}, \ldots, v^{(k-j-1)}, u^{(0)}, u^{(1)}, \ldots, u^{(k-j-1)}$ are crowded, and thus they are not zombies. Thus w is growing.

$\hat{\pi}(\cdot)$, that is, we write $v = \hat{\pi}(u)$. Note that a bag of level j in the original bag forest $\mathcal{F}$ has level j in the adoption bag forest $\hat{\mathcal{F}}$ as well.

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

Observation 3.20. *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 the original bag forest $\mathcal{F}$. 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 the adoption bag forest $\hat{\mathcal{F}}$.

Definition 3.21. 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 $\hat{v}^{(1)} = \hat{\pi}(v), \hat{v}^{(2)} = \hat{\pi}(\hat{v}^{(1)}), \ldots, \hat{v}^{(\beta)} = \hat{\pi}(\hat{v}^{(\beta-1)})$ is a growing bag. For $j > \ell - \beta$, all bags $v \in \mathcal{F}_j$ are called β-prospective. We also use the shortcut *prospective* for γ-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 3.22. *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 $j \leq \ell - \gamma$. 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 $j + 1 \leq k \leq j + \gamma$. 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. By induction on $(k - j)$, it is easy to see 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, $1 \leq i \leq k - j$, $v^{(i-1)} \in \mathcal{S}(v^{(i)}) \subseteq \chi(v^{(i)}), u^{(i-1)} \in \mathcal{S}(u^{(i)}) \subseteq \chi(u^{(i)})$. 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 $|\chi(w)| \geq |\mathcal{S}(w)| \geq 2$. Thus w is a growing bag, and v is a prospective one. See Figure 9 for an illustration. $\square$

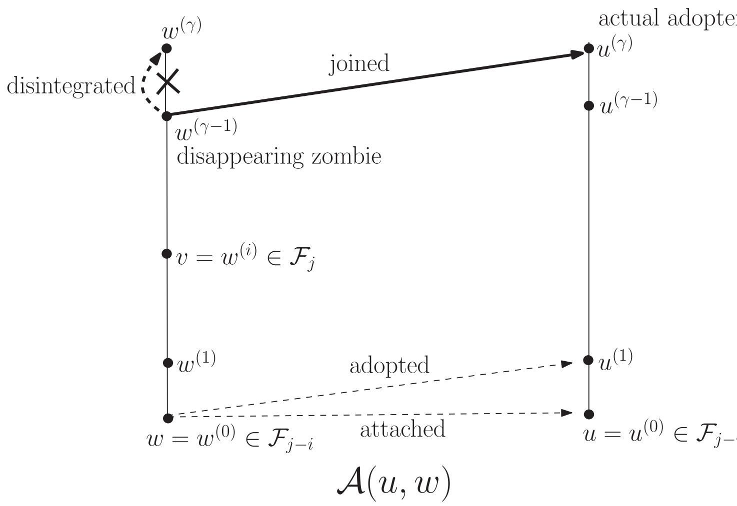

Fig. 10. The case that $v = w^{(i)}$ is a zombie. The bags $w^{(\gamma-1)}$ and $u^{(\gamma-1)}$ are two distinct nonempty extended children of $u^{(\gamma)}$. Thus $u^{(\gamma)}$ is a growing $\hat{\mathcal{F}}$-ancestor of $v = w^{(i)}$.

Next, we consider the case of a zombie bag.

LEMMA 3.23. *Let $j \in [\ell]$. A zombie bag $v \in \mathcal{F}_j$ is prospective.*

PROOF. We only need to prove the assertion for $j \leq \ell - \gamma$. By construction, there exists an attached bag $w = w^{(0)}$, which is an $\mathcal{F}$-descendant of v. By Lemma 3.6, the bags $w = w^{(0)}, w^{(1)} = \pi(w^{(0)}), \ldots, w^{(i)} = v, \ldots, w^{(\gamma-1)}$ are identical, where $i \in [\gamma - 1]$ 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 $\mathcal{A}(u, w)$ 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 3.1, none of the incubator bags $u^{(1)}, \ldots, u^{(\gamma-1)}$ is a disappearing zombie. Hence, by Observation 3.20, 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 nonempty, and thus the incubators $u^{(1)}, \ldots, u^{(\gamma-1)}$ are nonempty 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 nonempty. Hence the zombie bags $w^{(1)}, w^{(2)}, \ldots, w^{(\gamma-1)}$ are nonempty 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 3.20 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 step-child of the actual adopter $u^{(\gamma)}$, that is, $w^{(\gamma-1)} \in \mathcal{J}(u^{(\gamma)}) \subseteq \chi(u^{(\gamma)})$. Hence $u^{(\gamma)}$ is an $\hat{\mathcal{F}}$-ancestor of v. By Lemma 3.1, $u^{(\gamma-1)} \neq w^{(\gamma-1)}$. Hence $|\chi(u^{(\gamma)})| \geq 2$, that is, $u^{(\gamma)}$ is a growing bag. Thus the bag v is prospective. See Figure 10. □

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

LEMMA 3.24. *Let $j \in [\ell]$. An incubator bag $v \in \mathcal{F}_j$ is prospective.*

Lemmas 3.22, 3.23, and 3.24 imply the following statement.

COROLLARY 3.25. *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, that is, 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. Before proving it we shortly outline the main idea of our degree analysis. Intuitively, large bags are easy to handle because they contain enough points to share the load. As a result of this load-sharing, no point in a large bag ever becomes overloaded. To handle small bags we show that once a small bag becomes active (i.e., its points start being loaded), it will soon get merged into a larger bag. These merges will allow for a more uniform load-sharing, resulting in a small (constant) load for all points in $\boldsymbol{Q}$.

LEMMA 3.26. *Let $j \le \ell - \eta$, 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 nonempty. 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, that is, $r(u) = r(\hat{u}^{(0)}) = r(\hat{u}^{(1)}) = \cdots = r(\hat{u}^{(\eta)})$. (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 any 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 3.20, $\hat{u}^{(1)} = u^{(1)}, \ldots, \hat{u}^{(\eta)} = u^{(\eta)}$, that is, the η immediate $\hat{\mathcal{F}}$-ancestors of u are its η immediate $\mathcal{F}$-ancestors.

Consider the j-level processing. (It is described in Section 2.6.) Since u is active, the representative $r(u)$ of u is not isolated in $G^*_j = (\boldsymbol{Q}_j, E^*_j)$. Hence $r(u) \in \boldsymbol{Q}^*_j$. Moreover, the bag u is useful, hence $r(u) \in \hat{\boldsymbol{Q}}_j$. If $r(u)$ is not isolated in $E^*_j(\hat{\boldsymbol{Q}}_j)$, then it is not isolated in the j-level attachment graph $G_j = (\hat{\boldsymbol{Q}}_j, \mathcal{E}_j)$, $\mathcal{E}_j = E^*_j(\hat{\boldsymbol{Q}}_j) \cup \hat{E}_j$. 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 $E^*_j(\hat{\boldsymbol{Q}}_j)$. Recall that $\hat{\boldsymbol{Q}}_j$ is the subset of $\boldsymbol{Q}^*_j$ which contains only nonzombie representatives. Since $r(u)$ is not isolated in $G^*_j = (\boldsymbol{Q}_j, E^*_j)$, there must exist a zombie z, such that $r(z) \in \boldsymbol{Q}_j \setminus \hat{\boldsymbol{Q}}_j$ and the edge $(r(u), r(z)) \in E^*_j$. It follows that

$$\delta(r(u), r(z)) \ \le \ \tau_j. \tag{2}$$

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 $r(u) = r(u^{(h-j)}) = r(\hat{u}^{(h-j)})$ 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 3.27. *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* $\delta(r(u), r(w)) > \frac{\tau_h}{t}$.

PROOF. Suppose for contradiction that for some index $h \in [j, j+(\eta-1)]$ and a bag w as before, it holds that $\delta(r(u), r(w)) \leq \frac{\tau_h}{t}$. It follows that $r(u) = r(u^{(h-j)})$ and $r(w)$ are not isolated in G^*_h. Moreover, since $u^{(h-j)}$ and w are useful bags, their representatives belong to $\hat{Q}_h$, that is, $r(u) = r(u^{(h-j)}), r(w) \in \hat{Q}_h$. 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 3.26. Intuitively, we will show that $r(u)$ cannot be close to a zombie representative $r(z)$ (in the sense of Equation (2)), but far from any useful representative $r(w)$ in all levels $h \in [j, j+(\gamma-1)]$ (see Claim 3.27). This would lead to a contradiction.

Consider again the zombie $z \in \mathcal{F}_j$, such that $\delta(r(u), r(z)) \leq \tau_j$. Let i be the index such that $j-(\gamma-1) \leq i < j$ and 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 $y = y^{(0)}$, that is, $y^{(1)} = \pi(y^{(0)}), y^{(2)} = \pi(y^{(1)}), \ldots, y^{(j-i)} = z, \ldots, y^{(\gamma-1)} = \pi(y^{(\gamma-2)})$, are labeled as zombies. Since none of these bags except $y^{(\gamma-1)}$ are disappearing zombies, Observation 3.20 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 $r(y) = r(z) = r(y^{(1)}) = \cdots = r(y^{(\gamma-1)})$. 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 3.1, none of them is a zombie, and, in particular, none of them is a disappearing zombie. Thus, again by Observation 3.20, $v^{(1)} = \hat{\pi}(v^{(0)}) = \hat{v}^{(1)}, \ldots, v^{(\gamma-1)} = \hat{\pi}(\hat{v}^{(\gamma-2)}) = \hat{v}^{(\gamma-1)}$.

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 $\delta(r(v), r(y)) \leq \tau_i$. 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 $u \neq x$. 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 $\kappa + 1 \leq \eta - 1$, 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 nonempty k-level bags, and $Q(u') = Q(u)$, $Q(x') \supseteq Q(x)$, and $Q(u) \cap Q(x) = \emptyset$. Hence $Q(u') \neq Q(x')$, and thus $Q(u') \cap Q(x') = \emptyset$, and u' and x' are distinct bags. (See Figure 11 for an illustration.)

The analysis splits into two cases now, depending on whether the bag $x' = \hat{x}^{(k-j)}$ is a zombie or not. (Note that if $k - i \leq \gamma - 1$ then x' is an incubator, and not a zombie. But $k - i$ may be larger than $\gamma - 1$.) We start with the case that it is not a zombie. (It may be an attached bag.) By definition, x' is useful. We will show that $r(x')$ is prohibitively close to $r(u)$, and this would yield a contradiction.

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 3.15, $\delta(r(v), r(x')) \leq 2 \cdot \mu_k = \frac{\tau_k}{\rho \cdot t \cdot (c+1)}$. Recall that $r(u') = r(u)$ and $r(z) = r(y)$. Hence, by Equation (2), $\delta(r(u'), r(y)) = \delta(r(u), r(z)) \leq \tau_j$. By the triangle

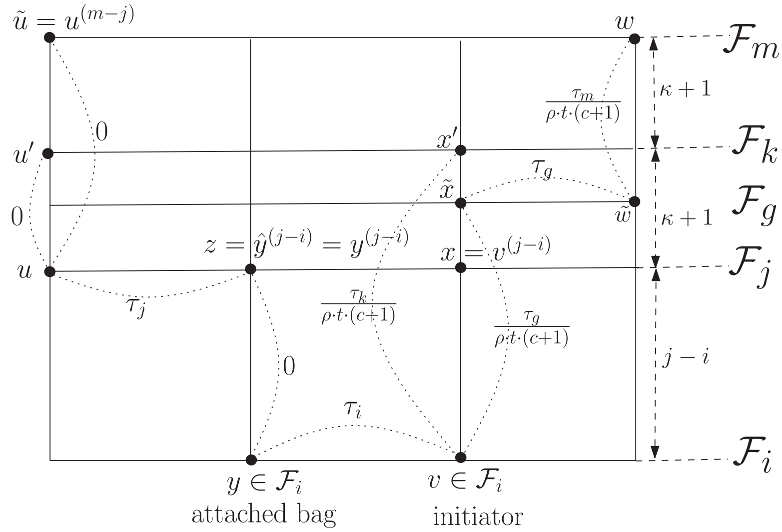

Fig. 11. A schematic illustration for the proof of Lemma 3.26. Expressions that appear next to dotted lines reflect upper bounds on distances between the representatives of their endpoints. For example, $r(\tilde{u}) = r(u)$, and thus 0 appears next to the dotted line that connects $\tilde{u}$ and u. Similarly, $\delta(r(v), r(x')) \leq \frac{\tau_k}{\rho \cdot t \cdot (c+1)}$, and thus $\frac{\tau_k}{\rho \cdot t \cdot (c+1)}$ appears next to the dotted line that connects v and x'.

inequality,

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

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 Claim 3.27.

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. We will show that $r(w)$ is prohibitively close to $r(u)$, yielding a contradiction. All the $(\gamma - 1)$ immediate $\mathcal{F}$-ancestors of $\tilde{w}$ are incubators, and thus, by Lemma 3.1, they are not zombies. In particular, none of them is a disappearing zombie. Hence, by Observation 3.20, 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 3.15, $\delta(r(v), r(\tilde{x})) \leq \frac{\tau_g}{\rho \cdot t \cdot (c+1)}$. 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,

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

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 $m = j + (\eta - 1) \in [j, j + (\eta - 1)]$. Hence this is also a contradiction to Claim 3.27.

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 3.26.

Next, we combine Corollary 3.25 and Lemma 3.26 to conclude that any active small bag is $(\gamma + \eta)$-prospective.

LEMMA 3.28. *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 3.26, 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 3.25 implies that $\tilde{v}$ is γ-prospective. It follows that v is $(\gamma + \eta)$-prospective. □

Recall that 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 spanner $\tilde{G}_j$. (See Section 2.4 for details.)

OBSERVATION 3.29. *Let $j \in [\ell]$, and $v \in \mathcal{F}_j$ be a small bag. Then for any point $p \in Q(v)$, it holds that $CTR_j(p) = 0$.*

PROOF. All $\hat{\mathcal{F}}$-descendants of v are small. 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 all $1 \leq i \leq j$. Hence $CTR_j(p) = 0$. □

OBSERVATION 3.30. *Let $j \in [\ell]$, and $v \in \mathcal{F}_j$ be a large bag. For every $\hat{\mathcal{F}}$-ancestor v' of v, $K(v') \supseteq K(v)$.*

PROOF. Only small bags may be labeled as zombies. Hence v is useful. Observation 3.20 implies that it will not have a step-parent, that is, $\hat{\pi}(v) = \pi(v)$ 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 to $\pi(v)$. $\square$

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 3.31. *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, by Observation 3.29 and by the construction, 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 Lemma 2.2, $|K(v)| \geq \ell$. Write $K(v) = \{q_1, \ldots, q_k\}$, where $k \geq \ell$.

Next, we argue that

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

Since all counters with index 0 are 0, 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 3.29, 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 $v = v^{(0)}$, that is, $v^{(1)}, \ldots, v^{(j-g)} = v_j(p)$, $v^{(j-g+1)} = v_{j+1}(p)$. Observation 3.30 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. $\square$

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 spanner $\tilde{G}_i$ *and* its host bag $v_i(p)$ is a singleton, that is, $Q(v_i(p)) = \{p\}$.

LEMMA 3.32. *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 spanner $\tilde{G}_i$, *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 $ctr_1(p) = \cdots = ctr_{i-1}(p) = 0$, and so $single\_ctr_1(p) = \cdots = single\_ctr_{i-1}(p) = 0$. By definition, the bag v is active. Thus, Lemma 3.28 implies that v is $(\gamma + \eta)$-prospective. It follows that there is an index k, $1 \leq k \leq (\gamma + \eta)$, 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 $single\_ctr_{i+k}(p) = single\_ctr_{i+k+1}(p) = \cdots = single\_ctr_{\ell}(p)$. In other words, the single counter of p may be incremented only during the h-level processing, for $h = i, i+1, \ldots, i+(k-1)$, that is, for at most $k \leq \gamma + \eta$ times. Therefore $single\_ctr_{\ell}(p) \leq \gamma + \eta$. □

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 $i \in [j]$, such that p serves as a representative of an i-level small bag v with $|Q(v)| \geq 2$, *and* p is not isolated in $\tilde{G}_i$.

LEMMA 3.33. *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 small bag v. By definition, $i \geq 2$. Also, there are no growing small h-level bags, for any index $1 \leq h \leq i-1$.
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 $|\chi(v)| \geq 2$. 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 small bags of level h, $1 \leq h \leq i-1$, it follows that there exist 1-level bags u' and w' such that u' is identical to u (and thus an $\mathcal{F}$-descendant of it) and w' is identical to w (and an $\mathcal{F}$-descendant of it). 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, that is, $Q(u) = \{r(u)\}$, then $plain\_ctr_{i-1}(r(u)) = 0$. (Its single counter $single\_ctr_{i-1}(r(u))$ might be larger, but it was taken care of separately. See Lemma 3.32.) Otherwise, $|Q(u)| \geq 2$. Hence $Q(u) \setminus \{r(u)\}$ contains at least one additional point $p(u)$, $p(u) \neq r(u)$. 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 Lemma 2.2, since v is a small bag, $K(v) = Q(v)$, and we are done.

Induction Step: Assuming the correctness of the statement for all index values smaller than j, for some $j \geq i+1$, we prove its correctness for index value 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 $h \in [0, j-2]$, 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 $p(u)$, $p(u) \neq r(u)$. If $\tilde{u} \in \mathcal{F}_1$, then $plain\_ctr_{j-1}(p(u)) = 0$. Otherwise $j-h-1 \geq 2$ 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 h 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 Lemma 2.2, 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 3.34. *For any point $p \in Q$, $plain\_ctr_\ell(q) \leq \gamma + \eta$.*

PROOF. Consider a point $q \in Q$, and 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, that is, $plain\_ctr_{i-1}(q) = 0$, $plain\_ctr_i(q) = 1$. 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 3.28, j is well-defined, with $i < j \leq i + \beta \leq \ell$. Consider the $j - i$ immediate $\hat{\mathcal{F}}$-ancestors of $v = \hat{v}^{(0)}$, that is, $\hat{v}^{(1)} = \hat{\pi}(\hat{v}^{(0)}), \ldots, \hat{v}^{(j-i)} = u = \hat{\pi}(\hat{v}^{(j-i-1)})$. 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 3.33, its kernel set $K(u)$ contains at least two points p, p' with plain counter zero, that is, $plain\_ctr_{j-1}(p) = plain\_ctr_{j-1}(p') = 0$. 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 3.35. *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. Assuming the correctness of the statement for all index values smaller than k, for some $k \geq j + 1$, we prove its correctness for index value 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 3.33, its kernel set $K(w)$ contains at least two points p, p' with plain counter zero, that is, $plain\_ctr_{k-1}(p) = plain\_ctr_{k-1}(p') = 0$. On the other hand, $plain\_ctr_{k-1}(q) \geq plain\_ctr_i(q) = 1$. Hence q is not the representative of w. Claim 3.35 follows. □

We now continue proving Lemma 3.34.
By Claim 3.35, 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 3.31, 3.32, and 3.34 imply the following corollary.

COROLLARY 3.36. *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 3.18.) Hence, Corollary 3.36 implies that the maximum degree of any $q \in Q$ in the graph $\tilde{G}_1 \cup \ldots \cup \tilde{G}_\ell$ is $O(\Delta(n) \cdot (\gamma + \eta)) = O(\Delta(n) \cdot \gamma)$. The 0-level auxiliary spanner $\tilde{G}_0$ contributes at most $O(\Delta(n))$ to the maximum degree of the final spanner $\tilde{G}$; also, 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 section.) We summarize the degree analysis with the next statement.

LEMMA 3.37. $\Delta(\tilde{G}) = O(\Delta(n) \cdot \gamma + \rho) = O(\Delta(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + \rho)$.

Deriving Theorem 1.4. Lemmas 3.9, 3.12, 3.16, and 3.37, and Corollary 3.11, imply Theorem 1.4.

In other words, we devised a transformation that, given a construction of t-spanners with $SpSz(n)$ edges, degree $\Delta(n)$ and diameter $\Lambda(n)$ which requires $SpTm(n)$ time, and given parameters $\rho \geq 2$ and $\epsilon > 0$, provides a construction of $(t + \epsilon)$-spanners with $O(SpSz(n) \cdot \max\{1, \log_\rho(t/\epsilon)\})$ edges, degree $O(\Delta(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + \rho)$, diameter $O(\Lambda(n) + \log_\rho n + \alpha(\rho))$, and lightness $O(\frac{SpSz(n)}{n} \cdot \rho \cdot \log_\rho n \cdot (t^3/\epsilon))$. The latter construction requires $O(SpTm(n) \cdot \max\{1, \log_\rho(t/\epsilon)\} + n \cdot \log n)$ time.

Substitute into this transformation a construction of $(1 + \epsilon)$-spanners with $O(n)$ edges, degree $O(\rho)$ and diameter $O(\log_\rho n + \alpha(\rho))$, which runs within $O(n \cdot \log n)$ time [Arya et al. 1995; Gottlieb and Roditty 2008b; Solomon and Elkin 2010]. We obtain a construction of $(1+2\epsilon)$-spanners with $O(n)$ edges, degree $O(\rho)$, diameter $O(\log_\rho n+\alpha(\rho))$ and lightness $O(\rho \cdot \log_\rho n)$, which requires $O(n \cdot \log n)$ time as well. (Observe that $t = 1 + \epsilon$, and so $\log_\rho(t/\epsilon) = \frac{\log(1+\frac{1}{\epsilon})}{\log \rho} = O(1)$. Also, we can rescale $2\epsilon = \epsilon'$.) For $\rho = O(1)$ this proves Conjecture 1. Moreover, due to lower bounds of [Chan and Gupta 2006; Dinitz et al. 2008], this result is tight up to constant factors in the entire range of the parameter ρ.

APPENDIXES

A. THE GENERAL RESULT

In this appendix we detail the dependencies on ϵ and the doubling dimension in our main result.

THEOREM A.1. *For any n-point metric M with an arbitrary (not necessarily constant) doubling dimension dim(M), any* $\epsilon > 0$ *and any 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* $(n \cdot \log n) \cdot \epsilon^{-O(dim(M))}$.

Theorem A.1 is obtained by instantiating the algorithm from Theorem B.1 (see Appendix B) as Algorithm *BasicSp* in Theorem 1.4. (Theorem B.1 is a version of Theorem 1.3, which details the dependencies on ϵ and the doubling dimension.) Note that we substitute $t = 1+\epsilon$ in Theorem 1.4. Thus the terms $\max\{1, \log_\rho(t/\epsilon)\}$ and (t^3/ϵ) that appear in the statement of Theorem 1.4 are upper bounded by $\epsilon^{-O(dim(M))}$.

The $(1 + \epsilon)$-spanner H from Theorem B.1 satisfies $|H| = n \cdot \epsilon^{-O(dim(M))}$, $\Delta(H) = \rho \cdot \epsilon^{-O(dim(M))}$, $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$, and its construction time is $(n \cdot \log n) \cdot \epsilon^{-O(dim(M))}$.

As a result we obtain a spanner construction H' for doubling metrics M with

$$|H'| \;=\; O(|H|)\cdot \max\{1, \log_\rho(t/\epsilon)\} \;=\; (n\cdot \epsilon^{-O(dim(M))})\cdot \max\{1, \log_\rho(t/\epsilon)\} \;=\; n\cdot \epsilon^{-O(dim(M))},$$

$$\Delta(H') \;=\; O(\Delta(H)\cdot \max\{1, \log_\rho(t/\epsilon)\} + \rho) \;=\; \rho\cdot \epsilon^{-O(dim(M))},$$

$$\Lambda(H') \;=\; O(\Lambda(H) + \log_\rho n + \alpha(\rho)) \;=\; O(\log_\rho n + \alpha(\rho)),$$

$$\Psi(H') \;=\; O\left(\frac{|H|}{n}\cdot \rho\cdot \log_\rho n\cdot (t^3/\epsilon)\right) \;=\; (\rho\cdot \log_\rho n)\cdot \epsilon^{-O(dim(M))}.$$

Moreover, the construction time of the resulting spanner H' is bounded by

$$O((n\cdot \log n)\cdot \epsilon^{-O(dim(M))}\cdot \max\{1, \log_\rho(t/\epsilon)\} + n\cdot \log n) \;=\; (n\cdot \log n)\cdot \epsilon^{-O(dim(M))}.$$

Finally, while the stretch of the constructed spanner H' is $t+\epsilon = 1+2\epsilon$, we can reduce it to $1+\epsilon$ by scaling the parameter ϵ (i.e., setting $\epsilon' = 2\epsilon$).

B. PROOF OF THEOREM 1.3

This appendix is devoted to the proof of Theorem 1.3. The following theorem details the dependencies on ϵ and the doubling dimension in Theorem 1.3.

THEOREM B.1 ([ARYA ET AL. 1995; GOTTLIEB AND RODITTY 2008B; SOLOMON AND ELKIN 2010]). *For any n-point metric $M = (P, \delta)$ with an arbitrary (not necessarily constant) doubling dimension $dim(M)$, any $\epsilon > 0$ and any $\rho \geq 2$, there exists a $(1+\epsilon)$-spanner H with $|H| = n\cdot \epsilon^{-O(dim(M))}$, $\Delta(H) = \rho\cdot \epsilon^{-O(dim(M))}$ and $\Lambda(H) = O(\log_\rho n + \alpha(\rho))$. The running time of this construction is $(n\cdot \log n)\cdot \epsilon^{-O(dim(M))}$.*

For Euclidean metrics Arya et al. [1995] proved this theorem for the case $\rho = 2$, and the authors of the current article generalized it in Solomon and Elkin [2010] to the entire range of the degree parameter ρ. For doubling metrics the proof of this theorem is based on the works of Gottlieb and Roditty [2008b] and Solomon and Elkin [2010]. 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 [Gottlieb and Roditty 2008b] proved the following theorem. (See also [Gao et al. 2004; Chan et al. 2005; Cole and Gottlieb 2006; Roditty 2007; Gottlieb and Roditty 2008a] for a number of earlier related works.)

THEOREM B.2 ([GOTTLIEB AND RODITTY 2008B]). *For any n-point doubling metric $M = (P, \delta)$ and any $\epsilon > 0$, one can build in $O(n\cdot \log n)$ 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 [Gottlieb and Roditty 2008b] may have a large diameter. To reduce the diameter, we employ the following tree-shortcutting theorem from Solomon and Elkin [2010].

THEOREM B.3 (THEOREM 3 IN [SOLOMON AND ELKIN 2010]). *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 $\rho \geq 2$, 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 1.3.

We start by building the spanner H and its tree-skeleton T that are guaranteed by Theorem B.2. 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.3. 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 1.3.

ACKNOWLEDGMENTS

S. Solomon is indebted to Michiel Smid for many helpful and timely comments, and for his constant support and willingness to help. Both authors wish to thank Adi Gottlieb for referring us to Gottlieb and Roditty [2008b], and for many helpful discussions.

REFERENCES

I. Abraham, Y. Bartal, and O. Neiman. 2011. Advances in metric embedding theory. *Advances in Mathematics* 228, 6, 3026–3126.

P. K. Agarwal, Y. Wang, and P. Yin. 2005. Lower bound for sparse Euclidean spanners. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 670–671.

I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. 1993. On sparse spanners of weighted graphs. *Discrete Comput. Geom.* 9, 81–100.

S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. 1995. Euclidean spanners: Short, thin, and lanky. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 489–498.

S. Arya, D. M. Mount, and M. H. M. Smid. 1994. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proceedings of the Annual Symposium on Foundations of Computer Science*. 703–712.

S. Arya and M. H. Smid. 1994. Efficient construction of a bounded degree spanner with low weight. In *Proceedings of the Annual European Symposium on Algorithms.* 48–59.

P. Assouad. 1983. Plongements lipschitziens dans R*n*. *Bull. Soc. Math. France* 111, 4, 429–448.

Y. Bartal, L. Gottlieb, and R. Krauthgamer. 2012. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 663–672.

P. Bose, P. Carmi, M. Farshi, A. Maheshwari, and M. H. M. Smid. 2010. Computing the greedy spanner in near-quadratic time. *Algorithmica* 58, 3, 711–729.

H. T.-H. Chan and A. Gupta. 2006. Small hop-diameter sparse spanners for doubling metrics. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms.* 70–78.

H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. 2005. On hierarchical routing in doubling metrics. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 762–771.

T.-H. H. Chan, M. Li, L. Ning, and S. Solomon. 2013. New doubling spanners: Better and simpler. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming*. Part 1, 315–327. (Tech. Rep., CoRR abs/1207.0892 (2012).)

B. Chandra, G. Das, G. Narasimhan, and J. Soares. 1992. New sparseness results on graph spanners. In *Proceedings of the Symposium on Computational Geometry*. 192–201.

B. Chandra, G. Das, G. Narasimhan, and J. Soares. 1995. New sparseness results on graph spanners. *Int. J. Comput. Geometry Appl.* 5, 125–144.

D. Z. Chen, G. Das, and M. H. M. Smid. 2001. Lower bounds for computing geometric spanners and approximate shortest paths. *Discrete Appl. Math.* 110, 2–3, 151–167.

L. P. Chew. 1986. There is a planar graph almost as good as the complete graph. In *Proceedings of the Symposium on Computational Geometry*. 169–177.

K. L. Clarkson. 1987. Approximation algorithms for shortest path motion planning. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 56–65.

K. L. Clarkson. 1999. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.* 110, 1, 63–93.

R. Cole and L. Gottlieb. 2006. Searching dynamic point sets in spaces with bounded doubling dimension. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 574–583.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. *Introduction to Algorithms,* 2nd ed. McGraw-Hill Book Company, Boston, MA.

G. Das, P. J. Heffernan, and G. Narasimhan. 1993. Optimally sparse spanners in 3-dimensional Euclidean space. In *Proceedings of the Symposium on Computational Geometry*. 53–62.

G. Das and G. Narasimhan. 1994. A fast algorithm for constructing sparse Euclidean spanners. In *Proceedings of the Symposium on Computational Geometry*. 132–139.

G. Das, G. Narasimhan, and J. S. Salowe. 1995. A new way to weigh malnourished Euclidean graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 215–222.

Y. Dinitz, M. Elkin, and S. Solomon. 2008. Shallow-low-light trees, and tight lower bounds for Euclidean spanners. In *Proceedings of the Annual Symposium on Foundations of Computer Science*. 519–528.

M. Elkin and S. Solomon. 2013. Fast constructions of light-weight spanners for general graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*.

M. Elkin and S. Solomon. 2013. Optimal Euclidean spanners: Really short, thin and lanky. In *Proceedings of the Symposium on the Theory of Computing*. 645–654. (Tech. Rep. CS12-04, Ben-Gurion University (2012).)

J. Gao, L. J. Guibas, and A. Nguyen. 2004. Deformable spanners and applications. In *Proceedings of the Symposium on Computational Geometry*. 190–199.

L. Gottlieb, A. Kontorovich, and R. Krauthgamer. 2012. Efficient regression in metric space via approximate lipschitz extension. Manuscript.

L. Gottlieb and L. Roditty. 2008a. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 591–600.

L. Gottlieb and L. Roditty. 2008b. An optimal dynamic spanner for doubling metric spaces. In *Proceedings of the Annual European Symposium on Algorithms*. 478–489.

J. Gudmundsson, C. Levcopoulos, G. Arasimhan. 2002a. Fast greedy algorithms for constructing sparse geometric spanners. *SIAM J. Comput.* 31, 5, 1479–1500.

J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. 2002b. Approximate distance oracles for geometric graphs. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 828–837.

J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. 2002c. Approximate distance oracles revisited. In *Proceedings of the 13th International Symposium on Algorithms and Computation*. 357–368.

J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. 2008. Approximate distance oracles for geometric spanners. *ACM Trans. Algorithms* 4, 1.

J. Gudmundsson, C. Levcopoulos, and M. H. M. Smid. 2005. Fast pruning of geometric spanners. In *Proceedings of the 22nd Annual Symposium on Theoretical Aspects of Computer Science*. 508–520.

A. Gupta, R. Krauthgamer, and J. R. Lee. 2003. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the Annual Symposium on Foundations of Computer Science*. 534–543.

S. Har-Peled, and M. Mendel. 2006. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.* 35, 5, 1148–1184.

Y. Hassin and D. Peleg. 2000. Sparse communication networks and efficient routing in the plane. In *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing*. 41–50.

J. M. Keil. 1988. Approximating the complete Euclidean graph. In *Proceedings of the 1st Scandinavian Workshop on Algorithm Theory*. 208–213.

J. M. Keil and C. A. Gutwin. 1992. Classes of graphs which approximate the complete Euclidean graph. *DiscreteComput. Geom.* 7, 13–28.

R. Krauthgamer and J. R. Lee. 2004. Navigating nets: Simple algorithms for proximity search. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 791–801.

H. P. Lenhof, J. S. Salowe, and D. E. Wrege. 1994. New methods to mix shortest-path and minimum spanning trees. manuscript.

Y. Mansour and D. Peleg. 2000. An approximation algorithm for min-cost network design. *DIMACS Series in Discr. Math and TCS* 53, 97–106.

G. Narasimhan and M. Smid. 2007. *Geometric Spanner Networks*. Cambridge University Press.

S. Rao and W. D. Smith. 1998. Approximating geometrical graphs via "spanners" and "banyans". In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 540–550.

L. Roditty. 2007. Fully dynamic geometric spanners. In *Proceedings of the Symposium on Computational Geometry*. 373–380.

J. S. Salowe. 1991. Construction of multidimensional spanner graphs, with applications to minimum spanning trees. In *Proceedings of the Symposium on Computational Geometry*. 256–261.

J. S. Salowe. 1991. On Euclidean spanner graphs with small degree. In *Proceedings of the Symposium on Computational Geometry*. 186–191.

M. H. M. Smid. 2009. The weak gap property in metric spaces of bounded doubling dimension. In *Proceedings of the Conference on Efficient Algorithms*. 275–289.

S. Solomon. 2011. An optimal time construction of Euclidean sparse spanners with tiny diameter. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*. 820–839.

S. Solomon. 2014. From hierarchical partitions to hierarchical covers: Optimal fault-tolerant spanners for doubling metrics. In *Proceedings of the Symposium on the Theory of Computing*. 363–372. (Tech. Rep., CoRR abs/1207.7040 (2012).)

S. Solomon and M. Elkin. 2010. Balancing degree, diameter and weight in Euclidean spanners. In *Proceedings of the Annual European Symposium on Algorithms*. 48–59.

K. Talwar. 2004. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 281–290.

P. M. Vaidya. 1991. A sparse graph almost as good as the complete graph on points in k dimensions. *Discrete Comput. Geom.* 6, 369–381.

Received November 2012; revised August 2014; accepted August 2015