

## THE TRAVELING SALESMAN PROBLEM: LOW-DIMENSIONALITY IMPLIES A POLYNOMIAL TIME APPROXIMATION SCHEME\*

YAIR BARTAL<sup>†</sup>, LEE-AD GOTTLIEB<sup>‡</sup>, AND ROBERT KRAUTHGAMER<sup>§</sup>

**Abstract.** The traveling salesman problem (TSP) is among the most famous NP-hard optimization problems. We design for this problem an algorithm that for any fixed  $\varepsilon > 0$  computes in randomized polynomial time a  $(1 + \varepsilon)$ -approximation to the optimal tour in TSP instances that form an *arbitrary* metric space with bounded intrinsic dimension. The celebrated results of Arora [*J. ACM*, 45 (1998), pp. 753–782] and Mitchell [*SIAM J. Comput.*, 28 (1999), pp. 1298–1309] prove that the above result holds in the special case of TSP in a fixed-dimensional Euclidean space. Thus, our algorithm demonstrates that the algorithmic tractability of metric TSP depends on the dimensionality of the space and not on its specific geometry. This result resolves a problem that has been open since the quasi-polynomial time algorithm of Talwar [*Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, 2004, pp. 281–290].

**Key words.** traveling salesman, approximation algorithm, hierarchies

**AMS subject classification.** 68W25

**DOI.** 10.1137/130913328

**1. Introduction.** The *traveling salesman problem (TSP)* is a fundamental and extensively studied NP-complete problem. Indeed, numerous articles and even whole books [42, 34, 26, 4] are devoted to TSP, studying various algorithms for different families of instances. In fact, some of the most basic techniques in combinatorial optimization were devised to tackle TSP, including, for instance, cutting planes [19]. The input for (the optimization version of) TSP is a complete graph, whose vertex set we denote by  $S = [n]$ , together with edge-weights  $w(\cdot, \cdot)$  that are nonnegative and symmetric,<sup>1</sup> and the goal is to find a closed tour of  $S$  of minimum (total) weight, where a tour is simply a permutation of  $S$ , i.e., it visits every vertex exactly once.

A prominent special case of TSP, called *metric TSP*, is where the edge-weights satisfy the triangle inequality,<sup>2</sup> and hence the input is simply a (finite) metric space on the point set  $S = [n]$ . The importance of this variant lies in the fact that edge-weights arising in many of the typical applications naturally represent lengths and distances. Metric TSP offers some basic structure that may be leveraged by algorithms. In particular, Christofides [14] designed a 1.5-approximation, meaning a polynomial time algorithm that computes a tour whose weight exceeds the optimum by a factor of at most 1.5. It is a long-standing open problem to improve this approximation for metric

---

\*Received by the editors March 18, 2013; accepted for publication (in revised form) August 17, 2015; published electronically August 30, 2016. An extended abstract of this paper appeared in *Proceedings of STOC 2012*.

<http://www.siam.org/journals/sicomp/45-4/91332.html>

<sup>†</sup>Hebrew University, Jerusalem, Israel (yair@cs.huji.ac.il). This author's work was supported in part by Israel Science Foundation grant 1609/11.

<sup>‡</sup>Ariel University, Ariel, Israel (leead@ariel.ac.il).

<sup>§</sup>Weizmann Institute of Science, Rehovot 76100, Israel (robert.krauthgamer@weizmann.ac.il). This author's work was supported in part by US–Israel BSF grant 2010418 and by the Citi Foundation.

<sup>1</sup>Formally,  $w(x, y) = w(y, x) \geq 0$  for all  $x, y \in S$ .

<sup>2</sup>The triangle inequality says that  $w(x, y) \leq w(x, z) + w(z, y)$  for all  $x, y, z \in S$ .

TSP, but it is known that there exists a constant  $c > 1$ , for which  $c$ -approximation is NP-hard [40, 38, 32].

Celebrated results of Arora [5] and Mitchell [36] prove that the important special case of metric TSP where the input metric forms a Euclidean metric admits a PTAS.<sup>3</sup> To be more precise, these PTAS results apply to input metrics that are finite subsets of a *fixed-dimensional* Euclidean metric (in the case of [36], the Euclidean plane). Observing that these PTAS results require two separate conditions—Euclidean space and fixed dimensionality—it is only natural to ask the following.

QUESTION 1.1. *Do TSP instances that satisfy only one of the two properties, bounded dimension and Euclidean metric, admit a PTAS?*

The bounded-dimensionality requirement turns out to be necessary, as Trevisan [45] shows that TSP in Euclidean metrics (of dimension  $\log n$ ) is NP-hard to approximate within some constant  $c > 1$ . It is therefore not surprising that the running time of the aforementioned PTAS is doubly exponential in the dimension.

Eliminating the Euclidean requirement was first addressed by Talwar [44]. Observe that a basic premise of this question is that the notion of dimension applies to an arbitrary (non-Euclidean) metric space. This is indeed possible, and Talwar relied on a definition put forth by Gupta, Krauthgamer, and Lee [25] (following [7, 15]): The *doubling dimension* of a (finite) metric space  $S$ , denoted  $\text{ddim}(S)$ , is the smallest  $k > 0$  such that every ball in the metric can be covered by  $2^k$  balls of half the radius. This definition is essentially based on volume growth, and indeed simple volume estimates imply that a  $k$ -dimensional Euclidean metric has doubling dimension  $\Theta(k)$ . The opposite direction, however, is not true and in fact the family of metrics with bounded doubling dimension is significantly larger than that of bounded-dimensional Euclidean metrics (see [30, 33, 31, 25] for details). Talwar [44] generalized much of Arora's machinery [5] and showed that TSP in metrics with fixed doubling dimension admits a QPTAS.<sup>4</sup> But despite repeated attempts, the original goal remained open, as follows.

QUESTION 1.2. *Does TSP in metrics of bounded doubling dimension admit a PTAS?*

This question has fascinated researchers (see, e.g., [35]) for several reasons. First, the existence of a QPTAS may be interpreted as evidence that a PTAS is possible. Second, the above question accords well with a research program that was initiated in [25, 29, 44] and studies the analogy between Euclidean metrics of fixed dimension and general metrics of fixed doubling dimension, from the perspective of algorithmic tractability. It has been observed that many algorithms dealing with the former family of metrics can be adapted to deal with the latter; see, e.g., [16, 24, 3, 22] and references therein for recent instantiations. Likewise, the doubling dimension has been established as a good measure of intrinsic dimension in the theory of metric embeddings [13, 2, 23, 10].

A natural approach to resolving Question 1.2 in the positive would be to embed the original metric space in a bounded-dimensional Euclidean space (such embeddings were studied in [1, 2]), and then apply the PTAS of Arora [5]. While this general approach has been quite successful in resolving many other algorithmic problems

<sup>3</sup>PTAS, which stands for polynomial time approximation scheme, means that for every *fixed*  $\varepsilon > 0$  there is a  $(1 + \varepsilon)$ -approximation. Note that for every constant  $\varepsilon > 0$ , the runtime is polynomial in  $n$ .

<sup>4</sup>QPTAS, which stands for quasi-polynomial time approximation scheme, means that for every *fixed*  $\varepsilon > 0$  there is a  $(1 + \varepsilon)$ -approximation running in quasi-polynomial time  $2^{\text{poly} \log(n)}$ .

(see, for example, [8]), it fails here since any such embedding must have nonconstant distortion [30, 33, 31], in fact  $\Omega(\sqrt{\log n})$  [25]. It appears that achieving a PTAS for arbitrary bounded-dimensional metric spaces requires a new approach to bypass the limitations of the embedding.

**1.1. Results.** Our central contribution is a PTAS for TSP in metrics of fixed doubling dimension.

**THEOREM 1.3.** *A  $(1 + \varepsilon)$ -approximation to the optimal tour of a metric TSP instance  $S$  on  $n = |S|$  points can be computed by a randomized algorithm in time  $n^{2^{O(\text{ddim}(S))}} \cdot 2^{(2^{\text{ddim}(S)}/\varepsilon)^{O(\text{ddim}(S))}} \sqrt{\log n}$ .*

The previously known running time is quasi-polynomial in  $n$ , namely,  $2^{(\text{ddim}(S) \cdot \varepsilon^{-1} \log n)^{O(\text{ddim}(S))}}$ , due to Talwar [44, Theorem 8].

**1.2. Techniques.** We build upon the framework of [5, 44] and introduce two main new ideas (and several more minor ones). Our baseline is a carefully chosen variant of Talwar’s algorithm, and as described in section 2, it includes (1) a randomized hierarchical clustering (partitioning) of  $S$ ; (2) the introduction of portals around every cluster; (3) slightly modifying the optimal tour (for the sake of analysis only) so that the tour is portal-respecting (crosses every cluster only at its portals) and has few crossings into the cluster; (4) a dynamic program that computes a tour for each cluster based on the tours already computed for its subclusters.

Our first new idea (in section 1.5) is to estimate the cost incurred by an optimal tour inside a ball. Intuitively, the estimate is merely an instantiation of the well-known 2-approximation of TSP using the minimum spanning tree (stated as Lemma 1.7). But in reality, edges entering and exiting the ball interfere with this calculation, and thus the estimate includes both multiplicative and additive error terms.

Our second new idea is to treat separately dense regions in the metric space, meaning balls in which an optimal tour incurs a relatively large cost. If all regions are sparse (not dense), then we are almost done—in this case we use limited randomization and enumeration, to determine the hierarchical clustering. Specifically, we draw at random  $O(\log n)$  radii-values for every cluster center and argue that with high probability at least one of them is useful for the construction of a good partition. We then augment the aforementioned dynamic program to search also for the correct radii-values for the different cluster centers. (This appears in section 3.1.) If there is a dense region, then we can use the first idea above to find the (nearly) smallest one. We then “split” the TSP instance into two portions, roughly the inside and the outside of that dense region. The outside is solved recursively. The inside portion is nearly sparse because it can be covered by a limited number of smaller (and thus sparse) balls, and so it can be solved directly by our algorithm for sparse regions. Stitching the solutions for the two portions may be costly, but since the region is dense, we can effectively charge our algorithm’s cost to that of the optimum. (This appears in section 3.2.)

**1.3. Related work.** A few hardness of approximation results are known. That the general (not necessarily metric) TSP is NP-hard follows immediately from Karp’s original NP-hardness proof for the Hamiltonian cycle [27]. Moreover, this proof shows that TSP does not admit any finite factor approximation in polynomial time, unless  $P=NP$ . Papadimitriou and Yannakakis [40] showed that metric TSP is hard to approximate within some constant factor  $c > 1$ , even if all the metric distances are either 1 or 2. Papadimitriou and Vempala [38] proved that approximating metric TSP within

factor 220/219 is NP-hard, and Lampis [32] recently improved this bound to 185/184. Papadimitriou [39] showed that the two-dimensional Euclidean TSP is NP-hard.

The runtime of Arora's algorithm [5] was later improved in [41], and his geometric approach was subsequently employed for other Euclidean problems in [17, 6, 18, 28]. Further extension of the algorithms of [5, 44] to the problem of TSP with neighborhoods (under mild conditions) include [37] and [11]. Chan and Gupta [12] gave an algorithm for TSP that runs in subexponential time in a larger family of instances, in which an alternative notion of dimension is assumed to be bounded.

**1.4. Preliminaries.** Recall our notation for the metric TSP instance:  $S$  denotes the set of points,  $d(\cdot, \cdot)$  their pairwise distances,  $\text{ddim}(S)$  its doubling dimension, and  $n = |S|$  its size. We fix  $0 < \varepsilon < 1/20$ , which determines the approximation we eventually achieve to be  $1 + O(\varepsilon)$ . We may assume that  $\varepsilon > 1/n$ , as otherwise all our results hold trivially—TSP can be solved exactly in time  $O(n!)$  by straightforward enumeration, providing better approximation and faster runtime than our claimed runtime (which is exponential in  $\text{poly}(1/\varepsilon)$ ). By arguments found in [5, section 2.1.1], namely, a suitable scaling and moving points at most distance  $\varepsilon n$ , we may assume that the minimum interpoint distance in  $S$  is 1 and the diameter is  $O(n/\varepsilon) = O(n^2)$ .

As usual, the metric may be viewed as the complete graph on  $S$ , with edge-weights corresponding to pairwise distances, denoted  $w(x, y) \stackrel{\text{def}}{=} d(x, y)$ . A subset of points  $S' \subseteq S$  is sometimes called a *cluster*. We let  $\text{MST}(S')$  denote a minimum-weight spanning tree (breaking ties arbitrarily) of the complete graph on  $S'$ . The *ball* centered at  $x \in S$  with radius  $R > 0$  is defined as  $B(x, R) \stackrel{\text{def}}{=} \{y \in S : d(x, y) \leq R\}$ . We define  $B^*(x, R)$  to be the edges of the complete graph on  $B(x, R)$ .

*Tours.* Throughout, a *tour*  $T$  is a finite sequence of points; by convention, it is undirected and may visit a point more than once. A *transition* in  $T$  is a pair of successive points in the sequence, which may be viewed as an edge in the complete graph on  $S$  (or a self-loop of zero weight, which can be eliminated if needed). A *closed* tour is defined in the natural way by adding a transition between the last and first points in the sequence.

The *weight* (or *length*) of a multiset  $M$  of transitions is defined as  $w(M) \stackrel{\text{def}}{=} \sum_{(x,y) \in M} w(x, y)$ . This notation naturally extends to a tour  $T$ , by viewing  $T$  as sequence of transitions; hence  $w(T)$  represents the total length of the tour  $T$ .

Let  $\text{OPT}(S')$  denote a minimum-length closed tour that visits all points of  $S' \subseteq S$ .

**LEMMA 1.4.** *Let  $T$  be a tour that traverses some edge  $e$  more than once in the same direction. Then there exists a lighter (smaller weight) tour  $T'$  that visits all the points visited by  $T$  and begins and ends at the same points as  $T$ . Moreover, the edges of  $T'$  are a subset of the edges of  $T$  (though  $T'$  does not necessarily traverse them in the same direction as  $T$ ).*

*Proof.* We will prove the case where  $T$  traverses some edge  $e$  exactly twice in a single direction; a similar proof holds for additional traversals. An ordering of the edges of  $T$  must take the form  $E_1 e E_2 e E_3$ , where each  $E_i$  is a (possibly empty) sequence of edges, and  $e = (u, v)$  is traversed twice in the same direction, say, from  $u$  to  $v$ . Let  $\bar{E}_2$  be a backward ordering of  $E_2$ , which begins at  $u$  and ends at  $v$ . Then  $E_1 \bar{E}_2 E_3$  visits all points visited by  $T$  and has the same endpoints as  $T$ , but it skips two traversals of  $e$  and is thus lighter than  $T$ .  $\square$

*Doubling dimension.* Let  $\lambda_S > 0$  be the *doubling constant* of the point set  $S$ , the smallest value such that every ball in  $S$  can be covered by  $\lambda_S$  balls of half the radius. Recall that the *doubling dimension* of  $S$  is  $\text{ddim}(S) \stackrel{\text{def}}{=} \log_2 \lambda_S \geq 1$  (assuming  $|S| \geq 2$ ).

(For convenience, one may round up the doubling dimension to the nearest integer.) The following packing property can be demonstrated via repeated applications of the doubling property (see, e.g., [25]).

LEMMA 1.5 (packing property). *Let  $S' \subseteq S$  have minimum interpoint distance  $\alpha > 0$ . Then*

$$|S'| \leq \left( \frac{2 \operatorname{diam}(S')}{\alpha} \right)^{\operatorname{ddim}(S)},$$

and whenever  $\frac{\operatorname{diam}(S')}{\alpha} \geq 2$ , we can further bound  $|S'| \leq \left( \frac{\operatorname{diam}(S')}{\alpha} \right)^{2 \operatorname{ddim}(S)}$ .

*Nets.* Similar to what was described in [21, 29], a subset  $S' \subseteq S$  is called a *b-net* of  $S$  if it satisfies the following two properties.

- (i) Packing: For every  $u, v \in S'$  we have  $d(u, v) > b$ .
- (ii) Covering: Every  $v \in S$  is within distance  $b$  of some point  $u \in S'$ , i.e.,  $S \subseteq \cup_{u \in S'} B(u, b)$ .

We say that  $u \in S'$  covers  $v \in S$  if  $d(u, v) \leq b$ . The two conditions above require that the points of  $S'$  be spaced out yet cover all points of  $S$ .

*Hierarchy of nets (or point hierarchies).* Recall that  $\operatorname{diam}(S) \leq O(n^2)$  and set  $L \stackrel{\text{def}}{=} \lceil \log_s \operatorname{diam}(S) \rceil = O(\log_s n)$  for a parameter  $s \geq 4$ . (Section 3 will require that  $s$  is roughly  $(\log n)^{1/\operatorname{ddim}(S)}$ .) For each  $i = 0, \dots, L$ , fix  $H_i \subseteq S$  to be an  $s^i$ -net of  $S$ , called the net of level  $i$ , or of scale  $s^i$ . We may assume that the nets are nested, i.e.,  $H_i \subseteq H_{i-1}$ . Having constructed  $H_i$ , we may initialize  $H_{i-1} = H_i$  and then greedily add to  $H_{i-1}$  uncovered points of  $S$  as needed [29]. Notice that the bottom level  $i = 0$  contains all points, and the top level  $i = L$  contains only a single point.

*Net-respecting tours.* A tour  $T$  is said to be *net respecting (NR)* relative to a given hierarchy  $\{H_i\}_{i=0}^L$  and value  $\varepsilon > 0$  if for every transition in  $T$ , say, of length  $\ell$ , both of its endpoints belong to  $H_i$  for  $i$  such that  $s^i \leq \varepsilon \ell < s^{i+1}$ . When the hierarchy is nested, this implies that both points belong to every net  $H_j$  with  $j \leq i$ , although we will find it convenient to view the edge as connecting the *occurrences* of the endpoints in the single level  $H_i$ . (When  $\ell < \frac{1}{\varepsilon}$ , it suffices to connect  $H_0$  level points; in this case the hierarchy implicitly contains levels  $H_i$  for all  $i < 0$ , and like  $H_0$  these nets contain all points of  $S$ .) We denote by  $\operatorname{OPT}^{NR}(S')$  an optimal (minimum length) net-respecting tour that visits all points of  $S' \subseteq S$ .

LEMMA 1.6. *Let  $0 < \varepsilon \leq \frac{1}{8}$ . Then every tour  $T$  can be converted to a net-respecting tour  $T'$  which visits all points visited by  $T$ , such that*

$$w(T') \leq (1 + 16\varepsilon)w(T).$$

*Proof.* For every transition  $(x, y)$  in  $T$  do the following. Let  $x', y'$  be the  $i$ -level net points covering  $x, y$ , respectively, where  $i$  is the highest level such that  $s^i \leq 2\varepsilon d(x, y)$ . Replace transition  $(x, y)$  with  $(x', y')$ , and also add transitions  $(x, x')$  and  $(y, y')$ . The total cost of the new path is

$$d(x, x') + d(x', y') + d(y', y) \leq [d(x, y) + 2 \cdot 2\varepsilon d(x, y)] + 2 \cdot 2\varepsilon d(x, y) = (1 + 8\varepsilon)d(x, y).$$

While transition  $(x', y')$  is net-respecting, transitions  $(x, x')$  and  $(y, y')$  may not be. These transitions are themselves replaced by the procedure above. This leads to a series of transition replacements. As the replacement procedure is activated on  $2^j$  transitions of length at most  $(2\varepsilon)^j d(x, y)$ , and recalling that  $\varepsilon \leq \frac{1}{8}$ , the total additive cost is bounded by

$$\sum_{j=0}^{\infty} 8\varepsilon \cdot 2^j \cdot (2\varepsilon)^j d(x, y) \leq \sum_{j=0}^{\infty} 8\varepsilon 2^{-j} d(x, y) = 16\varepsilon d(x, y). \quad \square$$

*Spanning trees, tours, and patching.* It is well-known that the optimal tour on a set  $S'$  is approximated within factor 2 by the minimum spanning tree on  $S'$ .

LEMMA 1.7. *Let  $S' \subseteq S$ . Then  $w(\text{MST}(S')) \leq w(\text{OPT}(S')) \leq 2w(\text{MST}(S'))$ .*

The following lemma, due to Talwar [44] (see also [43]), uses the doubling dimension to bound  $w(\text{MST}(S'))$ .

LEMMA 1.8. *Let  $S' \subseteq S$ . Then  $w(\text{MST}(S')) \leq 4|S'|^{1-1/\text{ddim}(S)} \cdot \text{diam}(S')$ .*

The next lemma, due to [5, 44], is known as the patching lemma for doubling spaces. We say that a transition  $(x, y)$  in a tour  $T$  crosses a cluster  $C \subseteq S$  if exactly one of  $x, y$  belongs to  $C$ . The point (among  $x, y$ ) that belongs to  $C$  is called a cross-point. A tour  $T$  may cross  $C$  multiple times at multiple cross-points.

LEMMA 1.9 (patching lemma). *Let  $T$  be a tour that crosses a cluster  $C$  at most  $r$  times, at cross-points  $\hat{C} \subset C$ . Then there is a tour  $T'$  with the same endpoints as  $T$  which visits all points visited by  $T$ , crosses  $C$  at most twice, and*

$$w(T') \leq w(T) + 4w(\text{MST}(\hat{C})) \leq w(T) + 16r^{1-\frac{1}{\text{ddim}(S)}} \text{diam}(\hat{C}).$$

*Remarks.* The last inequality is due to Lemma 1.8. We sketch below the proof of this lemma for completeness, as it is omitted from [44]. We also note for later reference that the bound  $w(T') \leq w(T) + 4w(\text{MST}(C))$  follows by the same proof, except for replacing the minimum spanning tree for  $\hat{C}$  with one for  $C$ .

*Proof (sketch).* For simplicity, we shall consider only a closed tour  $T$  and omit the adaptations needed for an open tour. Break the tour  $T$  at each crossing of  $C$ , and fix arbitrarily two crossings to keep (the two is because  $T$  is closed).

Consider first the portions of the tour that are inside  $C$ . As these portions are “disconnected” only at points of  $\hat{C}$  (in fact, excluding at most two of the points), adding a minimum spanning tree on  $\hat{C}$  of total weight  $w(\text{MST}(\hat{C}))$  results in a set of edges that is connected. Let  $C_{\text{odd}}$  be the points in  $\hat{C}$  that have an odd degree under the current set of edges, and add a minimum-weight matching on  $C_{\text{odd}}$ . We claim (and will prove shortly) that the matching’s total weight is at most  $w(\text{MST}(\hat{C}))$ . Using the claim, the current set of edges (consisting of portions of the tour, a spanning tree, and a matching) is both connected and has even degree at all but two vertices (the two crossings we keep). By Euler’s theorem, these edges can be arranged as an open tour connecting the two retained crossings. These manipulations increase the tour length by at most  $2w(\text{MST}(\hat{C}))$ .

The same arguments apply separately to the portions outside  $C$ , Patching them as before into an open tour between the two retained crossings increases the tour length again by at most  $2w(\text{MST}(\hat{C}))$ . However, the patching described above introduces edges *inside*  $C$ , and so we add a final step to “shortcut” around these edges. This shortcut maintains all visits to vertices outside  $C$  (and their order) and does not increase the tour length (by the triangle inequality). The lemma follows by combining the two open tours.

To prove the claim concerning the minimum-weight matching on  $C_{\text{odd}}$ , define new weights  $w'$  between points in  $\hat{C}$  as follows. Let  $w'(x, y)$  be the shortest-path distance on the tree  $\text{MST}(\hat{C})$ , and observe that  $w'(x, y) \geq w(x, y)$ . Thus, it suffices to upper bound a minimum-weight matching (on  $C_{\text{odd}}$ ) under the tree weights  $w'$ . Consider such a matching, and view every edge in the matching as a path in the tree  $\text{MST}(\hat{C})$ . These tree-paths must be disjoint, because if two tree-paths were to use the same tree-edge, then a simple swap would decrease the weight of the matching. Thus, this matching’s weight (under  $w'$ ) is at most the total weight of the tree, that is, at most  $w(\text{MST}(\hat{C}))$ .  $\square$

We prove here another version of the patching lemma, tailored to our specific needs.

LEMMA 1.10. *Let  $T$  be a tour that visits all of  $S$ , and suppose it crosses a cluster  $C$  at most  $r$  times, at cross-points  $\hat{C} \subset C$ . Let  $\{T_i\}_{i=1}^k$  be the maximal subtours of  $T$  that are entirely inside  $C$ . Then there exists a closed tour  $T'$  that visits all points of  $C$ , contains only edges in  $\cup_i T_i$  and in  $\text{MST}(\hat{C})$ , and*

$$w(T') \leq \sum_{i=1}^k w(T_i) + 2w(\text{MST}(\hat{C})) \leq \sum_{i=1}^k w(T_i) + 8r^{1-\frac{1}{\text{ddim}(S)}} \text{diam}(C).$$

*Remark.* The last inequality is due to Lemma 1.8 and  $\text{diam}(\hat{C}) \leq \text{diam}(C)$ .

*Proof.* The idea is to “stitch” together the different subtours  $T_i$ , by constructing a suitable multigraph on the vertex set  $C$ . We initialize the multigraph to be the tours  $T_i$ . The endpoints of these tours are all included in the cross-points  $\hat{C}$ , so by adding to our multigraph the edges of  $\text{MST}(\hat{C})$ , we make sure the multigraph is connected. Let  $C_{\text{odd}} \subseteq C$  be the vertices of odd degree in our multigraph, and observe that  $C_{\text{odd}} \subseteq \hat{C}$  because all vertices in  $C \setminus \hat{C}$  have an even degree in  $\cup_i T_i$  and degree zero in  $\text{MST}(\hat{C})$ . Now add to our multigraph a minimum-weight perfect matching  $M_{\text{odd}}$  on  $C_{\text{odd}}$ . By well-known arguments which date back to Christofides [14],

$$w(M_{\text{odd}}) \leq \frac{1}{2}w(\text{OPT}(C_{\text{odd}})) \leq \frac{1}{2}w(\text{OPT}(\hat{C})) \leq w(\text{MST}(\hat{C})),$$

and thus the total edge-weight in our multigraph is at most  $\sum_{i=1}^k w(T_i) + 2w(\text{MST}(\hat{C}))$ . Furthermore, the multigraph is Eulerian—connected with even degrees—and thus admits a closed tour  $T'$  that visits all points visited by  $\cup_i T_i$ , and hence all points of  $C$ , and whose weight  $w(T')$  is bounded as desired.  $\square$

Note that the previous lemma does not address connecting the tour segments outside the cluster. This can be done via the minimum spanning tree, which adds an additional weight of  $2w(\text{MST}(\hat{C})) \leq 8r^{1-\frac{1}{\text{ddim}(S)}} \text{diam}(C)$  to the final tour.

*Exponential distribution.* In the construction of our hierarchy (sections 2 and 3), we will need to create a ball centered at a point  $u$ , with a random radius chosen according to the exponential distribution. Having fixed some value  $a$ , the density function of this distribution can take the form

$$f(r) = \frac{2^{8 \text{ddim}(S)}}{1 - 2^{-8 \text{ddim}(S)}} \cdot \frac{8 \text{ddim}(S) \ln 2}{a} \cdot 2^{-\frac{8 \text{ddim}(S)}{a} r}$$

for  $r \in [a, 2a]$  and 0 for all other values of  $r$ ; see [2].

**1.5. Local behavior of optimal tour.** We next show that the weight of the optimal net-respecting tour inside some neighborhood can be approximated using a minimum spanning tree of points in that neighborhood.

LEMMA 1.11. *Let  $\text{OPT}^{NR}(S)$  be an optimal net-respecting tour visiting all points in  $S$  (for  $0 < \varepsilon \leq \frac{1}{8}$  and  $s \geq 6$ ). Then for all  $u \in S$  and any radius  $R > 0$ ,*

- (i)  $w(\text{OPT}^{NR}(S) \cap B^*(u, R)) \leq 6(1 + 16\varepsilon) \cdot w(\text{MST}(B(u, R)))$ ,
- (ii)  $w(\text{OPT}^{NR}(S) \cap B^*(u, 4R)) \geq w(\text{MST}(B(u, R))) - (s/\varepsilon)^2 \text{ddim}(S) R$ .

*Proof.* We show that if (i) does not hold, we can modify the tour to reduce its weight, which then contradicts the assumption that the tour is optimal. Applying the patching lemma (Lemma 1.9, with the subsequent remark) to the tour  $\text{OPT}^{NR}(S)$  with respect to the cluster  $B(u, R)$ , we get a modified tour which visits all of  $S$

and crosses that cluster at most twice, while increasing the tour's length by at most  $4w(\text{MST}(B(u, R)))$ . Now replace the portion of this tour inside the cluster with a tour that is derived from an MST of  $B(u, R)$ , and thus adds total length of at most  $2w(\text{MST}(B(u, R)))$  (Lemma 1.8). Finally, convert the newly added edges to be net-respecting (Lemma 1.6); this entire process first removes from the tour a total length of  $w(\text{OPT}^{NR}(S) \cap B^*(u, R))$ , while adding a total length of at most  $6(1 + 16\varepsilon) \cdot w(\text{MST}(B(u, R)))$ . Part (i) follows from the optimality of  $\text{OPT}^{NR}(S)$ .

To prove part (ii), consider a tour  $\text{OPT}^{NR}(S)$ . Ball  $B(u, R)$  partitions the tour into subtours  $T_1, T_2, T_3, \dots$ , where  $T_k$  for odd  $k$  contains only edges in  $B^*(u, R)$  and  $T_{k+1}$  contains only edges not in  $B^*(u, R)$ . Note that the first and last points in  $T_{k+1}$  must be in  $B(u, R)$ . By definition,  $w(\text{OPT}^{NR}(S)) = \sum_k w(T_k)$ .

We will now construct a connected graph  $G$  whose edges are all in  $B^*(u, 4R)$  and which spans all points of  $B(u, R)$ , and we use a charging argument to bound its weight against  $w(\text{OPT}^{NR}(S) \cap B^*(u, 4R))$ . First, add to  $G$  all subtours inside  $B^*(u, R)$ —that is,  $T_k$  for odd  $k$ . The cost of these edges of  $G$  is charged to the contribution of  $T_k$  to  $\text{OPT}^{NR}(S) \cap B^*(u, 4R)$ . Now consider subtours  $T_{k+1}$ . If  $T_{k+1}$  visits only points inside  $B(u, 4R)$ , then add  $T_{k+1}$  to  $G$ , and the cost of these edges of  $G$  is charged to the contribution of  $T_{k+1}$  to  $\text{OPT}^{NR}(S) \cap B^*(u, 4R)$ .

If  $T_{k+1}$  exits  $B(u, 4R)$ , then consider two more cases: (i) If  $T_{k+1}$  touches a point of the annulus  $v \in B(u, 4R) \setminus B(u, 3R)$  before its initial exit from  $B(u, 4R)$  or after its final entrance into  $B(u, 4R)$ , then add to  $G$  an edge connecting the first and last points of  $T_{k+1}$ . Since the endpoints of  $T_{k+1}$  are in  $B(u, R)$ , the added edge has weight at most  $2R$ . Since  $T_{k+1}$  connects one of its endpoints to a point in the annulus, we have that  $w(T_{k+1} \cap B^*(u, 4R)) \geq 3R - R = 2R$ , so the added edge can be charged to the contribution of  $T_{k+1}$  to  $\text{OPT}^{NR}(S) \cap B^*(u, 4R)$ . (ii) If  $T_{k+1}$ 's first exit and final entrance into  $B(u, 4R)$  are from points not in the annulus, then we add to  $G$  an edge connecting the exit and entry points in  $B(u, 3R)$ . Now, since these cross-points are in  $B(u, 3R)$ , the edges crossing  $B(u, 4R)$  have length at least  $R$ . Let  $i$  be the value for which  $s^i \leq \varepsilon R < s^{i+1}$ ; by the net-respecting property, these cross-points must belong to an  $s^i$ -net. Since by Lemma 1.5 the number of  $s^i$ -net points in  $B(u, 3R)$  is at most  $(\frac{2 \cdot 6R}{\varepsilon R / s})^{\text{ddim}(S)} < \frac{1}{3}(s/\varepsilon)^{2 \text{ddim}(S)}$ , the cost of adding edges connecting all  $s^i$ -net points in  $B(u, 3R)$  is at most  $3R \cdot \frac{1}{3}(s/\varepsilon)^{2 \text{ddim}(S)} = (s/\varepsilon)^{2 \text{ddim}(S)} R$ , from which the lemma follows.  $\square$

**2. TSP via hierarchical clustering (Arora and Talwar).** As an exposition to our PTAS, we review a variant of the algorithm of Talwar [44] (and in turn Arora [5]), which uses hierarchical clustering to compute a  $(1 + \varepsilon)$ -approximate tour in quasi-polynomial time. Recall that we may assume that the instance of TSP is a set  $S$  with minimum interpoint distance 1 and diameter  $O(n/\varepsilon) = O(n^2)$ . The construction uses a hierarchy of nets as described above. We first introduce the single-scale partition invoked by the algorithm—i.e., a partition which functions separately on each hierarchical level. This partition follows the same framework used in [8, 9, 20, 25, 2] and is slightly different from the one that appeared in [44] in that it uses the exponential distribution.

*Single-scale probabilistic partition.* Fix a set  $S' \subseteq S$  to be partitioned. Fix a level  $i$ , and impose an arbitrary ordering  $\pi$  on the points of the  $s^i$ -net  $H_i \subseteq S$ . The clusters are formed one by one following the ordering  $\pi$ . Each point of  $H_i$  constitutes a cluster center. With each net point  $u \in H_i$  we associate a random radius  $h_u \in [s^i, 2s^i]$  from an exponential distribution. The ball  $B(u, h_u)$  constitutes a new fixed cluster of  $S'$ , and then the process continues to form the rest of the clusters. The boundary of  $u$ 's



cluster is determined only by the ordering imposed by  $\pi$  and by the balls associated with cluster centers at distance at most  $4s^i$  from  $u$ . By the packing property (Lemma 1.5), there are at most  $2^{3\text{ddim}(S)}$  such cluster centers.

The next claim follows from [2].

CLAIM 2.1. *For every  $u, v \in S' \subseteq S$ , the probability that the single-scale probabilistic partition assigns  $u$  and  $v$  to different clusters (they are cut) is at most  $\frac{c' \text{ddim}(S)d(u,v)}{s^i}$  for some absolute constant  $c' > 0$ .*

*Hierarchical clustering.* To create the hierarchical clustering, we first choose a single-scale partition for the top level  $L$ . As described above, each net point chooses a radius in the range  $[s^L, 2s^L]$ , and then every point in  $S$  is assigned to the first ball in  $\pi$  that covers it. For the next hierarchical level  $L - 1$ , we take each  $L$ -level cluster separately and build for its points a new partition with random radius in the range  $[s^{L-1}, 2s^{L-1}]$ . The construction continues recursively until level 0, the bottom level. Note that each cluster has  $s^{O(\text{ddim}(S))}$  child clusters.

It follows that an  $i$ -level cluster is ultimately formed by independent single-scale partitions acting on all levels  $i$  and higher. It further follows that for any point pair  $u$  and  $v$ , the probability that the pair are *cut* at level  $i$  (found in different  $i$ -level hierarchical clusters) is bounded by the sum of the probabilities that they are cut by each single-scale partition acting on a level  $i$  or higher, that is,  $\sum_{j=i}^L \frac{c' \text{ddim}(S)d(u,v)}{s^j} = O(\frac{\text{ddim}(S)d(u,v)}{s^i})$ .

*TSP algorithm and analysis.* The dynamic programming TSP algorithm functions on the hierarchical clustering above. A tour is  $(m, r)$ -light with respect to a fixed hierarchical partition if it crosses each  $i$ -level cluster at most  $r$  times, and only at a set of  $m$  predetermined points, called the *portals*. Following [44], we define the  $m$  portals to be the  $\frac{s^i}{M}$ -net points in the cluster for some value  $M$  to be fixed below.<sup>5</sup> Recall that the diameter of an  $i$ -level cluster is at most  $4s^i$ , and so it follows from Lemma 1.5 that  $m \leq (8M)^{\text{ddim}(S)}$ . Throughout this section, we will take  $s = 6$  (the minimum admissible value of  $s$  in Lemma 1.11).

An optimal  $(m, r)$ -light tour for the hierarchical clustering can be computed by dynamic programming as follows. Consider a cluster  $C$ . Any valid  $(m, r)$ -light tour crosses  $C$  at most  $r$  times and only at portals, so the restriction of the path to  $C$  consists of at most  $r$  paths starting and ending at portals. A *configuration* is a multiset of  $r$  or fewer portals partitioned into pairs (each representing an entry/exit pair). A single portal may appear more than once in the configuration if the tour crosses it multiple times, but each instance counts toward  $r$ . A cluster has  $m$  portals, so there are no more than  $m^r$  possible configurations. Now, assuming inductively that optimal  $(m, r)$ -light tours have already been computed for all configurations for all  $s^{O(\text{ddim}(S))}$  children of  $C$  under the hierarchical clustering, the optimal  $(m, r)$ -light tour for each possible configuration of  $C$  can be computed by a brute-force algorithm: Since the  $(m, r)$ -light tour of each child cluster enters and exits via a portal, we can “stitch” together the child tours through the child portals. For each fixed configuration of  $C$  (at most  $m^r$  possible configurations), we consider all possible child configurations  $(m^{s^{O(\text{ddim}(S))}})_r$ . Having fixed a configuration for every child cluster, we have  $s^{O(\text{ddim}(S))}_r$  candidate child portals where the tour may cross. Since each

---

<sup>5</sup>In the event that the hierarchical cluster does not include the  $\frac{s^i}{M}$ -net points that cover the cluster points, we can always add to the cluster copies of the net points. In this case, the added points function as portals for the cluster, but do not necessarily need to be covered by a tour for the cluster.

child portal may be connected to one of  $s^{O(\text{ddim}(S))r}$  other candidate child portals, all possible graphs connecting these portals can be enumerated in time bounded by  $(s^{O(\text{ddim}(S))r})^{s^{O(\text{ddim}(S))r}}$ . Below we will choose a value for  $r$  satisfying  $r = s^{\omega(\text{ddim}(S))}$ , so we can bound the previous term by  $r^{s^{O(\text{ddim}(S))r}}$ . For each parent configuration, we choose the valid graph with the least cost tour. The total runtime is  $(mr)^{s^{O(\text{ddim}(S))r}} = m^{s^{O(\text{ddim}(S))r}}$ .

Crucially, it follows from [44] that with constant probability, the hierarchical clustering for  $S$  admits an  $(m, r)$ -light tour with weight at most  $(1 + \varepsilon)\text{OPT}(S)$ . Define  $M$  to be the smallest power of  $s$  that is greater than or equal to  $\frac{\text{ddim}(S)L}{\varepsilon}$ —that is,  $\frac{M}{s} < \frac{\text{ddim}(S)L}{\varepsilon} \leq M$ , and so  $m \leq (8M)^{\text{ddim}(S)} \leq (8s \text{ddim}(S)L/\varepsilon)^{\text{ddim}(S)}$ . (Recall that  $L = \log_s n$ .) Set  $r$  equal to  $m$ . The proof proceeds as in [5] by showing that an optimal tour can be slightly modified to observe the  $(m, r)$ -light property. The cost of modifying the tour is charged to the tour’s edges, and the analysis shows that the cost charged to each edge is small. Briefly, the probability that an edge  $e = (u, v)$  is cut by the  $i$ -level partition is bounded by  $\frac{c'd(u,v)\text{ddim}(S)}{s^i}$ . If the edge is cut, it is rerouted through  $\frac{s^i}{M}$ -net points, at an additive cost (increase in tour length) of  $\frac{4s^i}{M}$ . Hence, the expected cost of rerouting  $e$  due to a cut at level  $i$  is  $\frac{c'd(u,v)\text{ddim}(S)}{s^i} \cdot \frac{4s^i}{M} = O(\frac{\varepsilon d(u,v)}{L})$ , and the expected cost of rerouting  $e$  due to a cut in any of  $O(L)$  levels is  $O(\varepsilon \cdot d(u, v))$ . The previous step ensures that all edges crossing the cluster are incident on valid portals.

Now, if the optimal tour crosses an  $i$ -level cluster some  $\tilde{r} \geq r$  times, the number of crossing must be decreased. In this event, the tour is patched via the minimum spanning tree on the cross-points (as in Lemma 1.9). The cost is charged to the edges participating in the patching, at a per edge cost of  $O(\frac{s^i \tilde{r}^{1-1/\text{ddim}(S)}}{\tilde{r}}) = O(\frac{s^i \varepsilon}{\text{ddim}(S)L})$ . But an edge participates in a patching only if it is cut (which happens with the probability stated above), and hence the expected charged cost to  $e$  due to patchings at one level is  $O(\frac{d(u,v)\text{ddim}(S)}{s^i} \cdot \frac{s^i \varepsilon}{\text{ddim}(S)L}) = O(\frac{\varepsilon \cdot d(u,v)}{L})$  and due to patchings for all  $L$  levels is  $O(\varepsilon \cdot d(u, v))$ . The values for  $m$  and  $r$  imply that the algorithm above runs in quasi-polynomial time  $m^{s^{O(\text{ddim}(S))r}} = 2^{(\frac{L}{\varepsilon})\tilde{O}(\text{ddim}(S))} = 2^{(\frac{\log n}{\varepsilon})\tilde{O}(\text{ddim}(S))}$ .

*Runtime bottleneck.* In closing this section, we will elaborate on why the above algorithm does not achieve a PTAS. The runtime is directly affected by the dependence of  $r$  on  $L$ , which causes the term  $L^{O(\text{ddim}(S))} = (\log_s n)^{O(\text{ddim}(S))}$  to appear in the exponent. The dependence of  $r$  on  $L$  itself stems from the fact that the probability of a pair to be cut by each single-scale partition is calculated separately, and then these probabilities are summed over  $L$  levels, resulting in a term  $L$  appearing in the summation. Indeed, in the hierarchical clustering employed by the algorithm, the event that edge  $e$  is cut by an  $i$ -level single-scale partition, and by no other single-scale partition, is  $\Theta(\frac{\text{ddim}(S)}{s^i})$ . Hence, the expected cost of participating in an  $i$ -level patching is mostly independent of the expected cost of participating in a  $j$ -level patching for all  $i \neq j$ , and so a term of  $L$  must appear in  $r$ . This is precisely the reason why the analysis presented by Talwar [44] does not achieve a PTAS for metric TSP.

**3. Obtaining a PTAS.** In this section, we prove Theorem 1.3, the central contribution of this paper. Henceforth, we assert the conditions of Lemma 1.11, namely,  $s \geq 6$  and  $0 < \varepsilon \leq \frac{1}{6}$ .

Our algorithm for TSP is an adaptation of the one employed by Talwar [44].

His algorithm requires a hierarchical partition, yet we cannot directly employ the partition of section 2. As mentioned above, that clustering essentially decides the cluster assignment for each level separately, and hence we cannot successfully invoke the analysis of [5] to bound the expected cost of patchings per level.

Instead, we will employ a modified version of the above partition and analyze its performance on net-respecting tours. We will show that if a tour obeys some edge-sparsity property, then it admits an  $(m, r)$ -light tour on a hierarchy very similar to the one above. Crucially, the edge-sparsity property allows us to achieve  $r = O((\log n)^c)$  for a small constant  $c < 1$ , which implies a polynomial runtime. (Although we fix the value of  $c$  in the analysis, it can in fact be taken as an arbitrarily small constant.) This partition can be found by a “brute-force” version of the above dynamic programming algorithm. We then show that if the tour has edge-dense areas, then we can segment  $S$  into sparse pieces and solve TSP separately on each.

In what follows, we will mostly consider net-respecting tours. Note that for an optimal tour which is not net-respecting, we may impose the assumption that no point of  $S$  is visited more than once: If the optimal tour visits a point  $v$  more than once, we may shortcut around  $v$  by directly connecting its antecedent and successor points in the tour. However, this shortcutting is not always possible for optimal net-respecting tours, since connecting the antecedent and successor points may violate the net-respecting property of the tour. Indeed, an  $i$ -level net point  $v$  may possess links to  $\varepsilon^{-O(\text{ddim}(S))}$   $j$ -level net points at each level  $j \leq i$ . To address this issue, we will consider each occurrence of  $v$  in the hierarchy to be a separate copy of  $v$ : The copy of  $v$  in  $H_j$  is connected to at most  $\varepsilon^{-O(\text{ddim}(S))}$  other net points of  $H_j$ . We will also connect the copy of  $v$  in  $H_j$  to copies of  $v$  in  $H_{j-1}$  and  $H_{j+1}$  (if applicable), via edges of infinitesimally small length. (We shall assume that these edges are never cut and so do not figure into the cut analysis.) By Lemma 1.4, an optimal net-respecting tour traverses each of these edges at most twice. We consider an  $i$ -level partition to cut only edges incident on copies in levels  $j \leq i$ —the longer edges are only cut by higher level partitions.

**3.1. An algorithm for sparse tours.** In this section we show the following: For any fixed hierarchy, if there exists some net-respecting tour  $T$  whose edges obey a specific edge-sparsity condition, then there exists some clustering on the hierarchy which supports an  $(m, r)$ -light tour  $T'$  with low weight  $w(T') \leq (1 + \varepsilon)w(T)$  (for favorable values of  $m, r$ , see Lemma 3.1). Further, we can find this hierarchical clustering and the tour  $T$  in polynomial time (Lemma 3.2). Later in section 3.2, we will show that  $S$  can always be broken down into subsets which admit edge-sparse tours.

A tour  $T$  is said to be  $q$ -sparse with respect to a hierarchy  $H_1, \dots, H_L$  if for all  $i \in [L]$  and  $u \in H_i$ , the edges of  $T$  fully contained inside the ball  $B(u, 3s^i)$  have weight  $w(T \cap B^*(u, 3s^i)) \leq qs^i$ . The ball  $B^*(u, 3s^i)$  is said to be  $q$ -sparse with respect to the tour.

Suppose that an oracle had informed us that  $S$  admits a net-respecting tour that is a  $(1 + \varepsilon)$ -approximation to  $\text{OPT}(S)$  and is  $q$ -sparse. (An oracle with a similar capability is presented in section 3.2, for some values of  $q$ .) Then we can prove the following lemma. (Recall that  $c'$  is the constant appearing in Claim 2.1.)

LEMMA 3.1. *Suppose  $S$  admits a net-respecting  $q$ -sparse tour  $T$ . Then there exists a hierarchical clustering for  $S$  which admits an  $(m, r)$ -light tour  $T'$  with  $w(T') \leq (1 + \varepsilon)w(T)$  for*

$$m := (8 \log_s n \cdot s \text{ddim}(S) / \varepsilon)^{\text{ddim}(S)}$$

and

$$r = r(q) := 18q \cdot 2^{6 \operatorname{ddim}(S)} \operatorname{ddim}(S) \log_s \log n + (2c' \operatorname{ddim}(S)/\varepsilon)^{\operatorname{ddim}(S)} + (4s/\varepsilon)^{2 \operatorname{ddim}(S)}.$$

We remark that the tour  $T'$  need not be net-respecting.

*Proof.* The proof proceeds in three steps. In Step 1, we show how to construct the hierarchical clustering. We then prove the existence of  $(m, r)$ -light tour  $T'$  by showing that  $T$  can be modified to cross the clusters only at portal points (Step 2) and to cross the portal points only  $r$  times (Step 3).

*Step 1.* Fix  $T$ . The hierarchical clustering closely follows the description from section 2, with the only difference being that the cluster radii are chosen a little more carefully. Consider a net point  $u \in H_j$ . Let  $E_{j\text{-short}}$  be the edges of  $T$  of length at most  $s^j$ , and let  $\tilde{E}_{j\text{-short}} \subseteq E_{j\text{-short}}$  include only edges of  $T$  with at least one endpoint inside  $B(u, 2s^j)$ . As a consequence of the  $q$ -sparsity of the ball  $B(u, 3s^j)$ , we have that  $\sum_{e \in \tilde{E}_{j\text{-short}}} w(e) \leq qs^j$ .

Recall that we wish to assign  $u$  a random radius  $h_u \in [s^j, 2s^j]$ . Let  $V$  include all values in the (continuous) range  $[s^j, 2s^j]$  which cut fewer than  $9q2^{3 \operatorname{ddim}(S)} \operatorname{ddim}(S)$  edges of  $\tilde{E}_{j\text{-short}}$ . Since  $B(u, 2s^j)$  is  $q$ -sparse, the sum of edge lengths in  $\tilde{E}_{j\text{-short}}$  is at most  $qs^j$ , and so a simple averaging argument gives that less than a fraction  $\frac{1}{9 \cdot 2^{3 \operatorname{ddim}(S)} \operatorname{ddim}(S)}$  of radii in  $[s^j, 2s^j]$  intersect more than  $9q \cdot 2^{3 \operatorname{ddim}(S)} \operatorname{ddim}(S)$  edges of  $\tilde{E}_{j\text{-short}}$ . We choose  $h_u$  randomly from an exponential distribution on  $[s^j, 2s^j]$  and resample until finding a  $h_u \in V$ . Recalling that the density function  $f(\cdot)$  of the exponential distribution is decreasing, and setting parameter  $a = s^j$ , the probability that a sampled radius is invalid is less than  $f(a) \cdot \frac{a}{9 \cdot 2^{3 \operatorname{ddim}(S)} \operatorname{ddim}(S)} < 2^{-3 \operatorname{ddim}(S)} \ln 2$ .

The rest of the clustering is done exactly as before by iterating over centers. Note that knowledge of  $T$  was necessary to determine which radii are valid choices for  $h_u$ .

*Step 2.* We now analyze the expected cost of converting the tour  $T$  to cross every cluster only through its  $m$  cluster portals. Consider some  $j$ -level cluster  $C$ , and recall that  $C$  is formed by combining a sequence of single-scale partitions in levels  $i \geq j$ . Let  $M$  be the smallest power of  $s$  at least  $\frac{\operatorname{ddim}(S) \log_s n}{\varepsilon}$ , and as above we define the  $m$  portals to be  $\frac{s^j}{M}$ -net points in the cluster. So the number of portals is at most  $(8M)^{\operatorname{ddim}(S)} \leq (\frac{8s \log_s n \operatorname{ddim}(S)}{\varepsilon})^{\operatorname{ddim}(S)}$  as required.

Recall from above that  $E_{j\text{-short}}$  is the set of edges of  $T$  of length at most  $s^j$ , and similarly define  $E_{j\text{-long}}$  to be the edges of  $T$  of length greater than  $s^j$ . Since  $T$  is net-respecting, the edges of  $E_{j\text{-long}}$  must all be incident on  $s^k$ -net points (or higher level points) for  $s^k \leq \varepsilon s^j < s^{k+1}$ . Since  $M$  is a power of  $s$  and  $\varepsilon s^j > \frac{s^j}{M}$ , we conclude that  $s^k \geq \frac{s^j}{M}$ . Hence these long edges cross  $C$  at a subset of the  $\frac{s^j}{M}$ -net points, and no further action is required.

We turn to edges  $E_{j\text{-short}}$ . Claim 2.1 asserted that the probability that a given edge  $e = (u, v) \in T$  is cut by an  $i$ -level single-scale partition is bounded by  $\frac{c' d(u, v) \operatorname{ddim}(S)}{s^i}$ . We show that the probability that  $e$  is cut *conditioned* on choosing only valid radii (those belonging to  $V$ ) is at most  $\frac{16c' d(u, v) \operatorname{ddim}(S)}{s^i}$ : An  $i$ -level ball cutting  $e$  is within distance  $2s^i$  of an endpoint of  $e$ , and so by Lemma 1.5 at most  $b = 2(2 \cdot 4)^{\operatorname{ddim}(S)} = 2 \cdot 2^{3 \operatorname{ddim}(S)}$  balls may cut  $e$ . The partition imposed an ordering on these balls. Let  $E_\ell$  be the event that the radius of the  $\ell$ th ball covers exactly one endpoint of  $e$ , and let  $F_\ell$  be the event that it covers neither. Then the probability that  $e$  is cut is exactly  $\Pr[E_1] + \Pr[E_2] \Pr[F_1] + \Pr[E_3] \Pr[F_1] \Pr[F_2] + \dots + \Pr[E_b] \prod_{i=1}^{b-1} \Pr[F_i]$ .

Conditioning on choosing a valid radius increases this sum by at most a factor  $(1 - 2^{-3 \text{ddim}(S)} \ln 2)^{-b} < e^{2b2^{-3 \text{ddim}(S)} \ln 2} = 16$ , as claimed.

If  $e \in E_{j\text{-short}}$  is cut, we reroute it through  $\frac{s^i}{M}$ -net points, increasing the tour by at most  $\frac{4s^i}{M}$ . So the expected cost of moving  $e$  due to a cut in level  $i$  is at most  $\frac{16c'd(u,v) \text{ddim}(S)}{s^i} \cdot \frac{4s^i}{M} = O\left(\frac{\varepsilon d(u,v)}{\log_s n}\right)$ , and the expected cost of moving  $e$  due to cuts in all  $L = O(\log_s n)$  levels is  $O(\varepsilon d(u,v))$ .

*Step 3.* Finally, we turn to the analysis of reducing the number of utilized cross-points to  $r$  via patching. First consider edges of  $E_{j\text{-long}}$ . As explained above, these are incident on  $s^k$ -net points (or higher level net points) for  $s^k \leq \varepsilon s^j < s^{k+1}$ . By the packing property, these cross-points account for at most  $(4s^j/s^k)^{\text{ddim}(S)} < (4s/\varepsilon)^{\text{ddim}(S)}$  active portals. Since a cluster may have at most  $(4s)^{\text{ddim}(S)}$  sibling clusters, and a tour may traverse each edge at most twice (Lemma 1.4), these account for at most  $2(4s)^{\text{ddim}(S)}(4s/\varepsilon)^{\text{ddim}(S)} < (4s/\varepsilon)^{2 \text{ddim}(S)}$  crossings. We can afford to retain all these crossings.

We turn to the short edges of  $E_{j\text{-short}}$ . Consider some  $j$ -level cluster  $C$  centered at  $u$ , and recall that  $C$  is formed by choosing a valid radius  $h_u \in [s^j, 2s^j]$ , and further combining a sequence of single-scale partitions in levels  $i \geq j$ . Since the radius of each  $(i \geq j)$ -level single-scale partition is chosen from  $V$ , it cuts at most  $9q \text{ddim}(S)$  edges of length at most  $s^j$ . Further, edges crossing  $C$  could have actually been cut by any  $i$ -level ball whose center is within distance  $h_u + 2s^i \leq 4s^i$  from the center of  $C$ ; there are at most  $2^{3 \text{ddim}(S)}$  such balls at each level  $i$ . It follows that the number of edges in  $\tilde{E}_{j\text{-short}}$  cut by each  $i$ -level single-scale partition is at most  $2^{3 \text{ddim}(S)} \cdot 9q \cdot 2^{3 \text{ddim}(S)} \text{ddim}(S)$ . Set

$$r' = r'(q) := \max \left\{ 2^{6 \text{ddim}(S)} \cdot 18q \text{ddim}(S) \log_s \log n, \left( \frac{2c' \text{ddim}(S)}{\varepsilon} \right)^{\text{ddim}(S)} \right\}$$

and consider the case where at least  $r'$  edges of  $\tilde{E}_{j\text{-short}}$  are cut. Then at least  $r'/2$  edges of  $\tilde{E}_{j\text{-short}}$  must have been cut by balls in levels  $i \geq j + \log_s \log n$ , and we can charge a patching for  $j$ -level cluster  $C$  only to these short edges.

Now, if more than  $r'$  short edges cross  $C$ , the tour is patched via the minimum spanning tree (à la Lemma 1.9), at a per-edge cost of  $O\left(\frac{s^j r'^{1-1/\text{ddim}(S)}}{r'/2}\right) = O\left(\frac{s^j \varepsilon}{\text{ddim}(S)}\right)$ . Recall though that the edges charged for this patching are edges cut by balls at levels  $j + \log_s \log n$  or higher. It follows that the expected cost to edge  $(u, v)$  due to a patching for  $j$ -level cluster  $C$  is  $O\left(\frac{d(u,v) \text{ddim}(S)}{s^{j+\log_s \log n}} \cdot \frac{s^j \varepsilon}{\text{ddim}(S)}\right) = O\left(\frac{\varepsilon d(u,v)}{\log n}\right)$  and due to patchings at all levels is  $O(\varepsilon d(u,v))$ . This concludes the analysis for the short edges, and together with the long edges the total number of cross-points is at most  $r := r' + (4s/\varepsilon)^{\text{ddim}(S)}$ .  $\square$

Let  $s = (\log n)^{1/(c'' \text{ddim}(S))}$  for some constant  $c'' \geq 32$ . We can now provide an efficient algorithm to find a tour with the guarantees of the last lemma.

**LEMMA 3.2.** *If  $S$  admits a net-respecting  $q$ -sparse tour  $T$ , then there exists a randomized algorithm that, with constant probability, finds a tour  $T'$  with  $w(T') \leq (1 + \varepsilon)w(T)$  in time  $n^{O(2^4 \text{ddim}(S))} \cdot 2^{O(q(\text{ddim}(S)/\varepsilon)^3 \text{ddim}(S) \sqrt[4]{\log n})}$ .*

*Proof.* If we could compute a hierarchical clustering that realizes Lemma 3.1, then the standard dynamic program from section 2 would give a tour for  $S$  satisfying Lemma 3.2. However, we cannot compute this hierarchical clustering, since we do not have access to  $T$  and cannot know which radii are valid choices for  $h_u$ . Instead, we present a dynamic program that guesses the proper value of  $h_u$ . Recall that the

exponential distribution of [2] implies that a random guess for the value of  $h_u$  is a valid value with probability at least  $1/2$ . Hence,  $O(\log n)$  independent random choices ensure that at least one choice for  $h_u$  is valid with probability  $1 - \frac{1}{n^2}$ , which by a union bound implies that with constant probability, for each net point at least one of its  $O(\log n)$  choices is valid.

We begin by fixing  $O(\log n)$  random radius choices for each net point. Now consider some  $j$ -level cluster  $C$  centered at  $u \in H_j$ .  $C$  is formed by cuts from neighboring balls in levels  $j$  and above, and we wish to enumerate all possible *formations* of  $C$ : Recall that we make  $O(\log n)$  random choices for  $h_u \in [s^j, 2s^j]$ . Further, since for all  $i \geq j$ ,  $u$  is within distance  $h_u + 2s^i < 4s^i$  of  $2^{3 \text{ddim}(S)}$  other  $s^i$ -net points whose radii may cut  $C$ , and we guess  $O(\log n)$  radii for each of these net points,  $C$  may be cut in  $(O(\log n))^{2^{3 \text{ddim}(S)}}$  different ways by the  $i$ -level partition. Since  $C$  may be cut from above in all  $L - j$  levels, it follows that the number of possible formations for  $C$  is bounded by  $(O(\log n))^{2^{3 \text{ddim}(S)}L}$ . Recall that  $s = (\log n)^{1/c'' \text{ddim}(S)}$ , and it follows that the number of levels in the hierarchy is  $L = O(\log_s n) = O(\frac{\text{ddim}(S) \log n}{\log \log n})$ . So the number of possible formations for  $C$  is bounded by  $(O(\log n))^{2^{3 \text{ddim}(S)}L} = n^{O(2^4 \text{ddim}(S))}$ .

Having fixed all random radii, we compute tour  $T$  via a dynamic programming algorithm which executes an exhaustive search. The dynamic programming table possesses a single entry for each possible portal configuration of each possible formation of each cluster center. So the table possesses  $m^r \cdot n^{O(2^4 \text{ddim}(S))}$  entries. The algorithm must compute for each entry an optimal cluster tour for the particular cluster formation and portal configuration. The table is filled in a bottom-up fashion, from level 0 to level  $L$ , and the algorithm computes the entry for a  $j$ -level cluster by consulting the entries of its child clusters in level  $j - 1$ : An  $j$ -level cluster  $C$  has at most  $s^{2 \text{ddim}(S)}$  child clusters, and since each cluster has  $O(\log n)$  possible radii, there are  $(O(\log n))^{s^{2 \text{ddim}(S)}}$  possible child formations. For each fixed child formation, we consider each portal configuration for the set of children ( $(m^r)^{s^{2 \text{ddim}(S)}}$  possibilities) and consult the appropriate table entries for the cost of the optimal child cluster tours. We then compute the cost of connecting the child portals to form a valid tour through the portals of  $C$ : This can be done by enumerating all graphs with one edge on each vertex, in at most  $(rs^{2 \text{ddim}(S)})^{rs^{2 \text{ddim}(S)}}$  different ways.

Note that (assuming sufficiently large  $\text{ddim}(S) = \Omega(1)$ )

$$\begin{aligned} r \log m &= O(q2^{6 \text{ddim}(S)} \log \log n \cdot (s \text{ddim}(S)/\varepsilon)^{2 \text{ddim}(S)} \cdot \log \log n \cdot \log(1/\varepsilon)) \\ &= O(q(\text{ddim}(S)/\varepsilon)^{3 \text{ddim}(S)} s^{2 \text{ddim}(S)} (\log \log n)^2). \end{aligned}$$

So we can bound each of the expressions above by

$$\begin{aligned} (mr \log n)^{O(rs^{2 \text{ddim}(S)})} &= 2^{O(r \log m s^{2 \text{ddim}(S)})} = 2^{O(q(\text{ddim}(S)/\varepsilon)^{3 \text{ddim}(S)} s^{4 \text{ddim}(S)} (\log \log n)^2)} \\ &= 2^{O(q(\text{ddim}(S)/\varepsilon)^{3 \text{ddim}(S)})} \sqrt[4]{\log n}. \end{aligned}$$

Lemma 3.2 follows.  $\square$

**3.2. Eliminating dense areas.** Lemmas 3.1 and 3.2 show that sparse tours admit efficient hierarchical decompositions and algorithms. Here, we consider tours that have dense neighborhoods and show that the point set can be divided into areas with all light tours. We then solve TSP on each subset and join the resulting subtours into a single tour.

Set  $q := (s/\varepsilon)^{O(\text{ddim}(S))} \cdot 2^{O(\text{ddim}^2(S))}$ .

LEMMA 3.3. *There is a (randomized) polynomial time algorithm that given a set  $S$  (with  $|S| > 1$ ) computes two subsets  $S_1 \subset S$  and  $S_2 \subsetneq S$  with  $S_1 \cup S_2 = S$  and  $S_1 \cap S_2 \neq \emptyset$ , such that*

- (a)  $\text{OPT}^{NR}(S_1)$  is  $q'$ -sparse, for  $q' = O(q \sqrt[8]{\log n})$ ; and
- (b)  $w(\text{OPT}^{NR}(S_1)) + w(\text{OPT}^{NR}(S_2)) \leq w(\text{OPT}^{NR}(S)) + \varepsilon w(\text{OPT}^{NR}(S_1))$ .

Before proving Lemma 3.3, we will demonstrate that it can be used to complete the proof of Theorem 1.3.

*Proof of Theorem 1.3.* Given a point set  $S$ , if  $S$  contains a single point then we are done. Otherwise, we use the procedure of Lemma 3.3 to create two instances of TSP,  $S_1 \subseteq S$  and  $S_2 \subset S$ .  $S_1$  admits a  $q'$ -sparse and net-respecting tour  $\text{OPT}^{NR}(S_1)$  as promised by Lemma 3.3. A tour of almost the same cost (at most  $1 + \varepsilon$  factor larger) can be computed by the algorithm of Lemma 3.2, obtaining a tour  $T_1$ , where  $w(T_1) \leq (1 + \varepsilon)w(\text{OPT}^{NR}(S_1))$ . The tour for  $S_2$  is solved recursively (that is,  $S$  is replaced by  $S_2$ ), obtaining a tour  $T_2$ . The inequality  $S_1 \cap S_2 \neq \emptyset$  implies that separate tours  $T_1$  and  $T_2$  can be joined together to obtain a complete tour  $T$  at no additional cost.

We now prove inductively that  $w(T) \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right) \cdot w(\text{OPT}^{NR}(S))$ . By the induction hypothesis we have that  $w(T_2) \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right) \cdot w(\text{OPT}^{NR}(S_2))$ . Lemma 3.3 implies that

$$w(T_1) \leq (1 + \varepsilon)w(\text{OPT}^{NR}(S_1)) \leq \left(\frac{1 + \varepsilon}{1 - \varepsilon}\right) \cdot (w(\text{OPT}^{NR}(S)) - w(\text{OPT}^{NR}(S_2))).$$

Therefore

$$w(T) = w(T_1) + w(T_2) \leq w(T_1) + \left(\frac{1 + \varepsilon}{1 - \varepsilon}\right) \cdot w(\text{OPT}^{NR}(S_2)) \leq \left(\frac{1 + \varepsilon}{1 - \varepsilon}\right) \cdot w(\text{OPT}^{NR}(S)),$$

proving the inductive claim. Finally, by Lemma 1.6 we have that  $w(T) = (1 + O(\varepsilon)) \cdot w(\text{OPT}(S))$ . The runtime follows from executing the algorithm of Lemma 3.2 on the  $q'$ -sparse sets of Lemma 3.3.  $\square$

We now return to proving Lemma 3.3. We require a preliminary lemma. Define the annulus  $A(v, r_1, r_2) = B(v, r_2) \setminus B(v, r_1)$ , and let  $A^*(v, r_1, r_2)$  be the set of edges with both endpoints inside the annulus  $A(v, r_1, r_2)$ .

LEMMA 3.4. *For any level  $i$ , let  $v \in S$  be a point for which  $w(\text{MST}(B(v, s^i)))$  is maximized, and let this weight be  $q^* s^i$ . Let  $T = \text{OPT}^{NR}(S)$ . If  $q^* \geq 6.5$  the following hold:*

- (i)  $w(\text{MST}(B(v, 13s^i))) < 2^{5 \text{ddim}(S)} \cdot q^* s^i$ .
- (ii) Set  $\delta \leq \frac{1}{12}$ . There exists a radius  $h \in [12s^i, 13s^i]$  for which

$$w(T \cap A^*(v, h - 6\delta s^i, h + 6\delta s^i)) < 144\delta(1 + 16\varepsilon)w(\text{MST}(B(v, 13s^i))).$$

- (iii) Let  $h$  be as above, let  $k$  satisfy  $s^k \leq \delta s^i < s^{k+1}$ , and let  $N(h)$  denote the set of all  $k$ -level points which cover points of  $A(v, h - \delta s^i, h + \delta s^i)$ . Then

$$\begin{aligned} \sum_{u \in N(h)} w(\text{MST}(u, s^k)) &< 2^{5 \text{ddim}(S)} w(T \cap A^*(v, h - 6\delta s^i, h + 6\delta s^i)) \\ &+ (2s^2/\varepsilon\delta)^{2 \text{ddim}(S)} s^k. \end{aligned}$$

*Proof.* To prove the first item, Lemma 1.5 implies that  $B(v, 13s^i)$  can be covered by  $(2 \cdot 13/3)^{\text{ddim}} < 2^{4 \text{ddim}(S)}$  balls of radius  $3s^i$  centered at points of  $S$ . Note that

$w(\text{MST}(B(v, 13s^i)))$  is bounded by the cost of constructing a minimum spanning tree inside each of these small balls and then connecting the balls together. By choice of  $v$ , each small ball has minimum spanning tree weight at most  $q^*s^i$ , so the sum of the weights of these minimum spanning trees is less than  $2^{4\text{ddim}(S)}q^*s^i$ . By Lemma 1.8, the centers of the small balls can be joined by a spanning tree of weight  $4(2^{4\text{ddim}(S)})^{1-1/\text{ddim}(S)} \cdot 26s^i = 2^{4\text{ddim}(S)} \cdot 6.5s^i$ . It follows that  $w(\text{MST}(B(v, 13s^i))) < 2^{4\text{ddim}(S)}(q^* + 6.5)s^i \leq 2^{5\text{ddim}(S)} \cdot q^*s^i$ .

To prove the second item, by an averaging argument, there is a value for  $h$  for which  $T \cap A^*(v, h - 6\delta s^i, h + 6\delta s^i)$  contains edges of total weight at most  $12 \cdot 2\delta \cdot w(T \cap B^*(v, 13s^i))$ . By Lemma 1.11(i),  $w(T \cap B^*(v, 13s^i)) \leq 6(1 + 16\varepsilon)w(\text{MST}(B(v, 13s^i)))$ . The item follows.

To prove the third item, by Lemma 1.11(ii) for each  $k$ -level net point  $u$  we have  $w(\text{MST}(B(u, s^k))) \leq w(T \cap B^*(u, 4s^k)) + (s/\varepsilon)^{2\text{ddim}(S)}s^k$ . By Lemma 1.5, the number of  $k$ -level net points covering the annulus is upper bounded by  $(2 \cdot 2((13 + \delta)s^i + s^k)/s^k)^{\text{ddim}(S)} < (2 \cdot 2(1 + 13.1s/\delta))^{\text{ddim}(S)} < (2s/\delta)^{2\text{ddim}(S)}$ . Hence,

$$\sum_{u \in N(h)} w(\text{MST}(B(u, s^k))) \leq \sum_{u \in N(h)} [w(T \cap B^*(u, 4s^k))] + (2s^2/\varepsilon\delta)^{2\text{ddim}(S)}s^k.$$

Now, each ball  $B(u, 4s^k)$  is fully contained in the larger annulus  $A^*(v, h - 6\delta s^i, h + 6\delta s^i)$  and intersects at most  $(2 \cdot 16)^{\text{ddim}(S)} = 2^{5\text{ddim}(S)}$  other balls. So  $\sum_{u \in N(h)} w(T \cap B^*(u, 4s^k)) \leq 2^{5\text{ddim}(S)}w(T \cap A^*(v, h - 6\delta s^i, h + 6\delta s^i))$ , and the item follows.  $\square$

Finally, we complete the proof of Lemma 3.3.

*Proof of Lemma 3.3.* Suppose first that for all level  $i$  and  $u \in S$ , the edge-sparsity condition  $w(\text{MST}(B(u, 3s^i))) \leq 2qs^i$  holds. Then we conclude by Lemma 1.11(i) that  $\text{OPT}^{NR}(S)$  is  $13q$ -sparse, and our lemma is trivial: Set  $S_1 = S$  and  $S_2$  includes an arbitrary single point. Assume then that the edge-sparsity condition does not hold. The algorithm begins by locating the lowest level  $i$  for which there exists  $u \in S$  such that  $w(\text{MST}(B(u, 3s^i))) > 2qs^i$  and setting  $v$  to be such that  $w(\text{MST}(B(v, 3s^i)))$  is maximized. Let  $q^* := w(\text{MST}(B(v, 3s^i)))/s^i$ , and it follows that  $q^* > 2q$ .

Fix a tour  $T = \text{OPT}^{NR}(S)$ . Ideally, we would now like to choose some radius  $h$ , partition  $S$  into two point sets  $\tilde{S}_1 = B(v, h)$  and  $\tilde{S}_2 = S \setminus \tilde{S}_1$ , and then provide tours for the two sets whose combined weight is only slightly greater than that of  $T$ . Let  $\tilde{S}_i^*$  denote the edges of the complete graph on the points of  $\tilde{S}_i$  (for  $i = 1, 2$ ); then we could bound the weight of the subtours by showing that each  $T_i = \tilde{S}_i^* \cap T$  (for  $i = 1, 2$ ) can be made into a closed tour by adding only a lightweight collection of edges to “patch” the edges of  $T$  cut by the partition (as in Lemma 1.10). However, this plan may be costly because the patchings might be expensive; for example, a radius  $h$  ball can cut many edges of  $T$  which then need to be patched. Moreover, in order to ensure that the subtours are net-respecting, we need to augment the sets  $\tilde{S}_i$  with appropriate (nearby) net points. To solve this problem, we will show how to create sets  $S_1 \supset \tilde{S}_1$  (which also contains some points of  $\tilde{S}_2$ ) and  $S_2 \supset \tilde{S}_2$  (which also contains some points of  $\tilde{S}_1$ ) for which the lemma holds.

We choose a radius  $h \in [12s^i, 13s^i]$  given by Lemma 3.4. In what follows, we will show separately how to patch long and short edges crossing  $\tilde{S}_1$ . (Similar arguments allow for patching  $\tilde{S}_2$ .)

Let  $E_{i\text{-long}}$  be the set of long edges of  $T$  crossing  $\tilde{S}_1$ , those of length more than  $\delta s^i$  for  $\delta = O(\varepsilon/2^{10\text{ddim}(S)})$ . Since  $T$  is net-respecting, these edges must cross  $\tilde{S}_1$  at  $j$ -level net points, where  $j$  satisfies  $s^j \leq \varepsilon\delta s^i < s^{j+1}$ . We will patch the edges of  $E_{i\text{-long}}$  crossing  $\tilde{S}_1$  using the minimum spanning tree of *all*  $j$ -level net points covering  $\tilde{S}_1$ .



There are  $(s/\varepsilon\delta)^{4\text{ddim}(S)}$  such net points, and their net-respecting minimum spanning tree has weight less than  $(1 + 16\varepsilon)(s/\varepsilon\delta)^{4\text{ddim}(S)}s^i < \varepsilon qs^i$  for an appropriate choice of  $q$  (see Lemmas 1.6, 1.8), which bounds, up to a constant factor, the cost of patching the long edges crossing  $\tilde{S}_1$ .

We now turn to patching the shorter edges of  $T$  that cross  $\tilde{S}_1$ . Let  $E_{i\text{-short}}$  include edges of length at most  $\delta s^i$ . Now, these edges cross into  $\tilde{S}_1$  from a set of points  $V \subset \tilde{S}_2$  inside the annulus  $A(v, h - \delta s^i, h + \delta s^i)$ . Since we have shown how to patch the long edges of  $T_1$  crossing  $j$ -level net points, the short edges of  $T_1$  crossing  $V$  will be patched by connecting them to  $k$ -level net points (though not necessarily directly) where  $k$  satisfies  $s^k \leq \delta s^i < s^{k+1}$ . To this end, add to  $\tilde{S}_1$  copies of all  $k$ -level net points which cover points in  $V$ —call this set  $N(h)$ —as well as all lower level points within distance  $s^k$  of  $N(h)$ , and let the resulting point set be  $S_1$ . We patch the short edges via the minimum spanning tree of the  $s^k$ -radius balls of points in  $N(h)$ , that is,  $\cup_{u \in N(h)} \text{MST}(B(u, s^k))$ . By the three items of Lemma 3.4, when  $q \geq \varepsilon^{-1}(2s^2/\varepsilon\delta)^{2\text{ddim}(S)}$  we have that the total cost of this patching is

$$\cup_{u \in N(h)} w(\text{MST}(B(u, s^k))) = O(\delta) \cdot 2^{10\text{ddim}(S)} q^* s^i.$$

Since  $\delta = O(\varepsilon/2^{10\text{ddim}(S)})$ , this last bound is  $O(\varepsilon q^* s^i)$ . We then reroute all new edges to be net-respecting (adding to  $\tilde{S}_1$  net points of  $\tilde{S}_2$  as necessary) at a trivial cost (Lemma 1.6).

We conclude that the total weight of all patchings (including short and long edges) is  $O(\varepsilon q^* s^i)$ . By Lemma 1.11(ii) (and since  $S_1 \supset B(v, 12s^i)$ ) we have  $w(\text{OPT}^{NR}(S_1)) \geq w(\text{MST}(B(v, 3s^i))) - (s/\varepsilon)^{2\text{ddim}(S)} 3s^i = q^* s^i - (s/\varepsilon)^{2\text{ddim}(S)} 3s^i \geq \frac{1}{2} q^* s^i$ , so the patching cost is indeed  $O(\varepsilon) \cdot w(\text{OPT}^{NR}(S_1))$ . A similar argument applies to the patchings needed for  $S_2$ , by adding to  $\tilde{S}_2$  all  $j$ -level net points of  $\tilde{S}_1$ , as well as all  $k$ -level net points of  $\tilde{S}_1$  covering points of  $\tilde{S}_2$  and their balls; the resulting set is  $S_2$ . This completes the proof of part (b) of the lemma.

We note that by construction  $S_1 \cap S_2 \neq \emptyset$ , and that  $S_2 \neq S$  (as implied by part (b) of the lemma).

We now complete the proof of part (a) translating balls with light MST to light tours. By the choice of  $i$ , for every  $\ell < i$  and every net point  $u \in H_\ell$ ,  $w(\text{MST}(B(u, 3s^\ell))) \leq 2qs^\ell$ , and so by Lemma 1.11(i),  $w(\text{OPT}^{NR}(S_1) \cap B^*(u, 3s^\ell)) \leq 6(1 + 16\varepsilon)qs^\ell \leq 22qs^\ell$ . While by assumption the  $\ell$ -level balls have light minimum spanning trees, this is not the tree of the  $i$ -level ball  $B(v, 3s^i)$ . Fix  $\ell$  to be the value satisfying  $s^\ell \leq s^i < s^{\ell+1}$ , and by the packing property the  $i$ -level ball covers  $(4s)^{\text{ddim}(S)}$   $\ell$ -level balls. It follows that  $w(\text{OPT}^{NR}(S_1) \cap B^*(u, 3s^i)) \leq (4s)^{\text{ddim}(S)} 22qs^\ell = O(\sqrt[16]{\log n}) \cdot qs^i$ , which completes the proof.  $\square$

**Acknowledgments.** We thank Ittai Abraham, Alex Andoni, Anupam Gupta, Liam Roditty, and Kunal Talwar for helpful discussions.

REFERENCES

[1] I. ABRAHAM, Y. BARTAL, AND O. NEIMAN, *Embedding metric spaces in their intrinsic dimension*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 2008, pp. 363–372.  
 [2] I. ABRAHAM, Y. BARTAL, AND O. NEIMAN, *Advances in metric embedding theory*, Adv. Math., 228 (2011), pp. 3026–3126.  
 [3] I. ABRAHAM, S. CHECHIK, C. GAVOILLE, AND D. PELEG, *Forbidden-set distance labels for graphs of bounded doubling dimension*, in Proceedings of 29th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, ACM, New York, 2010, pp. 192–200.

- [4] D. L. APPLIGATE, R. E. BIXBY, V. CHVATAL, AND W. J. COOK, *The Traveling Salesman Problem: A Computational Study*, Princeton Ser. Appl. Math., Princeton University Press, Princeton, NJ, 2007.
- [5] S. ARORA, *Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems*, J. ACM, 45 (1998), pp. 753–782.
- [6] S. ARORA, P. RAGHAVAN, AND S. RAO, *Approximation schemes for Euclidean  $k$ -medians and related problems*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 106–113.
- [7] P. ASSOUD, *Plongements Lipschitziens dans  $\mathbf{R}^n$* , Bull. Soc. Math. France, 111 (1983), pp. 429–448.
- [8] Y. BARTAL, *Probabilistic approximation of metric spaces and its algorithmic applications*, in Proceedings of the 37th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, New York, 1996, pp. 184–193.
- [9] Y. BARTAL, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 161–168.
- [10] Y. BARTAL, B. RECHT, AND L. J. SCHULMAN, *Dimensionality reduction: Beyond the Johnson-Lindenstrauss bound*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11, SIAM, Philadelphia, 2011, pp. 868–887.
- [11] T.-H. H. CHAN AND K. M. ELBASSIONI, *A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics*, Discrete Comput. Geom., 46 (2011), pp. 704–723.
- [12] T.-H. H. CHAN AND A. GUPTA, *Approximating TSP on metrics with bounded global growth*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2008, pp. 690–699.
- [13] T.-H. H. CHAN, A. GUPTA, AND K. TALWAR, *Ultra-low-dimensional embeddings for doubling metrics*, in Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '08, SIAM, Philadelphia, 2008, pp. 333–342.
- [14] N. CHRISTOFIDES, *Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem*, Technical report, Carnegie-Mellon University Management Sciences Research Group, Pittsburgh, PA, 1976.
- [15] K. L. CLARKSON, *Nearest neighbor queries in metric spaces*, Discrete Comput. Geom., 22 (1999), pp. 63–93.
- [16] R. COLE AND L.-A. GOTTLIEB, *Searching dynamic point sets in spaces with bounded doubling dimension*, in Proceedings of the 38th Annual ACM Symposium on Theory of Computing, 2006, pp. 574–583.
- [17] A. CZUMAJ AND A. LINGAS, *A polynomial time approximation scheme for Euclidean minimum cost  $k$ -connectivity*, in Proceedings of the 25th International Colloquium on Automata, Languages and Programming, ICALP '98, Springer-Verlag, New York, 1998, pp. 682–694.
- [18] A. CZUMAJ, A. LINGAS, AND H. ZHAO, *Polynomial-time approximation schemes for the Euclidean survivable network design problem*, in Proceedings of the 29th International Colloquium on Automata, Languages and Programming, ICALP '02, Springer-Verlag, New York, 2002, pp. 973–984.
- [19] G. DANTZIG, R. FULKERSON, AND S. JOHNSON, *Solution of a large-scale traveling-salesman problem*, Oper. Res., 2 (1954), pp. 393–410.
- [20] J. FAKCHAROENPHOL, S. RAO, AND K. TALWAR, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 448–455.
- [21] J. GAO, L. J. GUIBAS, AND A. NGUYEN, *Deformable spanners and applications*, Comput. Geom. Theory Appl., 35 (2006).
- [22] L.-A. GOTTLIEB, L. KONTOROVICH, AND R. KRAUTHGAMER, *Efficient classification for metric data*, in Proceedings of the 23rd Conference on Learning Theory, Omnipress, Madison, WI, 2010, pp. 433–440.
- [23] L.-A. GOTTLIEB AND R. KRAUTHGAMER, *A nonlinear approach to dimension reduction*, in Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '11, SIAM, Philadelphia, 2011, pp. 888–899.
- [24] L.-A. GOTTLIEB AND L. RODITTY, *An optimal dynamic spanner for doubling metric spaces*, in Proceedings of the 16th Annual European Symposium on Algorithms, ESA '08, Springer-Verlag, New York, 2008, pp. 478–489.
- [25] A. GUPTA, R. KRAUTHGAMER, AND J. R. LEE, *Bounded geometries, fractals, and low-distortion embeddings*, in Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS '03, IEEE Computer Society, New York, 2003, pp. 534–543.
- [26] G. GUTIN AND A. P. PUNNEN, EDs., *The Traveling Salesman Problem and Its Variations*, Comb. Optim. 12, Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

- [27] R. KARP, *Reducibility Among Combinatorial Problems*, in Complexity of Computer Computations, R. Miller and J. Thatcher, eds., Plenum Press, London, 1972.
- [28] S. G. KOLLIPOULOS AND S. RAO, *A nearly linear-time approximation scheme for the Euclidean  $k$ -median problem*, SIAM J. Comput., 37 (2007), pp. 757–782.
- [29] R. KRAUTHGAMER AND J. R. LEE, *Navigating nets: Simple algorithms for proximity search*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, 2004, pp. 791–801.
- [30] T. J. LAAKSO, *Ahlfors  $Q$ -regular spaces with arbitrary  $Q > 1$  admitting weak Poincaré inequality*, Geom. Funct. Anal., 10 (2000), pp. 111–123.
- [31] T. J. LAAKSO, *Plane with  $A_\infty$ -weighted metric not bi-Lipschitz embeddable to  $\mathbb{R}^N$* , Bull. London Math. Soc., 34 (2002), pp. 667–676.
- [32] M. LAMPIS, *Improved inapproximability for TSP*, in Approximation, Randomization, and Combinatorial Optimization, Lecture Notes in Comput. Sci. 7408, Springer, New York, 2012, pp. 243–253.
- [33] U. LANG AND C. PLAUT, *Bilipschitz embeddings of metric spaces into space forms*, Geom. Dedicata, 87 (2001), pp. 285–307.
- [34] E. L. LAWLER, J. K. LENSTRA, A. H. G. RINNOOY KAN, AND D. B. SHMOYS, EDS., *The Traveling Salesman Problem*, Wiley-Intersci. Ser. Discrete Math., Wiley, Hoboken, NJ, 1985.
- [35] J. R. LEE, *The Gödel Prize, TSP, and Volume Growth*, <http://tcsmath.wordpress.com/2010/06/24/the-godel-prize-tsp-and-volume-growth> (2010).
- [36] J. S. B. MITCHELL, *Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP,  $k$ -MST, and related problems*, SIAM J. Comput., 28 (1999), pp. 1298–1309.
- [37] J. S. B. MITCHELL, *A PTAS for TSP with neighborhoods among fat regions in the plane*, in Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2007, pp. 11–18.
- [38] C. PAPADIMITRIOU AND S. VEMPALA, *On the approximability of the traveling salesman problem*, Combinatorica, 26 (2006), pp. 101–120.
- [39] C. H. PAPADIMITRIOU, *The Euclidean travelling salesman problem is NP-complete*, Theoret. Comput. Sci., 4 (1977), pp. 237–244.
- [40] C. H. PAPADIMITRIOU AND M. YANNAKAKIS, *The traveling salesman problem with distances one and two*, Math. Oper. Res., 18 (1993), pp. 1–11.
- [41] S. B. RAO AND W. D. SMITH, *Approximating geometrical graphs via “spanners” and “banyans,”* in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 540–550.
- [42] G. REINELT, *The Traveling Salesman*, Lecture Notes in Comput. Sci. 840, Springer-Verlag, Berlin, 1994.
- [43] M. SMID, *On Some Combinatorial Problems in Metric Spaces of Bounded Doubling Dimension*, manuscript, <http://people.scs.carleton.ca/~michieli/research.html> (2010).
- [44] K. TALWAR, *Bypassing the embedding: Algorithms for low dimensional metrics*, in Proceedings of the 36th Annual ACM Symposium on Theory of Computing, ACM, New York, 2004, pp. 281–290.
- [45] L. TREVISAN, *When Hamming meets Euclid: The approximability of geometric TSP and Steiner tree*, SIAM J. Comput., 30 (2000), pp. 475–485.