Reducing Curse of Dimensionality: Improved PTAS for TSP (with Neighborhoods) in Doubling Metrics

T-H. Hubert Chan* Shaofeng H.-C. Jiang*

Abstract

We consider the Traveling Salesman Problem with Neighborhoods (TSPN) in doubling metrics. The goal is to find a shortest tour that visits each of a given collection of subsets (regions or neighborhoods) in the underlying metric space.

We give a randomized polynomial time approximation scheme (PTAS) when the regions are fat weakly disjoint. This notion of regions was first defined when a QPTAS was given for the problem in [SODA 2010: Chan and Elbassioni]. We combine the techniques in the previous work, together with the recent PTAS for TSP [STOC 2012: Bartal, Gottlieb and Krauthgamer] to achieve a PTAS for TSPN.

Moreover, more refined procedures are used to improve the dependence of the running time on the doubling dimension k from the previous $\exp[O(1)^{k^2}]$ (even for just TSP) to $\exp[O(1)^{O(k \log k)}]$.

1 Introduction

We consider the Traveling Salesman Problem with Neighborhoods (TSPN) in a metric space (V, d). An instance of the problem is given by a collection W of n subsets $\{P_1, P_2, \ldots, P_n\}$ in V. Each subset $P_j \subset V$ is known as a *neighborhood* or *region*. The objective is to find a minimum length tour that visits at least one point from each region.

This problem generalizes the well-known Traveling Salesman Problem (TSP), for which there are polynomial time approximation schemes (PTAS) for low-dimensional Euclidean metrics [22, 3, 25]. For some time, only a quasi-polynomial[1] time approximation scheme (QPTAS) is known for doubling metrics [27], where a metric space has doubling dimension [4, 9, 18] at most k, if any ball in the space can be covered by at most 2^k balls with half its radius. It was only recent that Bartal et al. [6] gave a PTAS for TSP on doubling metrics.

The neighborhood version of the problem was first introduced by Arkin and Hassin [2], who gave constant approximations for the case when the regions are in the plane and "well-behaved" (e.g., disks, parallel and similar length segments, bounded ratio between the largest and smallest diameters). The general version of the problem was shown to have an inapproximability threshold of $\Omega(\log^{2-c} n)$ for any $c > 0$ by Halperin and Krauthgamer [19]. There is an upper bound of $O(\log N \log k \log n)$-approximation, using the results of Garg et al. [15] and Fakcharoenphol et al. [14], where N is the total number of points in V and k is the maximum number of points in each region.

TSPN on The Euclidean Plane. As in the case for TSP, the special case when (V, d) is a subset of the Euclidean plane is considered to achieve better approximation ratios for TSPN. However, even for regions that are intersecting connected subsets, the problem remains APX-hard [10, 26].

In order for the problem to admit $(1+\epsilon)$ approximation, restrictions are placed on the regions; examples include diameter similarity, *fatness* and disjointness. Intuitively, the fatness of a region measures the ratio between the smallest circumscribing radius and the largest inscribing radius. For instance, a disk is fat, while a line segment is not.

Different assumptions on the regions in the Euclidean plane are considered, and the following approximation ratios are achieved: (i) $O(\log n)$ [20, 17], (ii) constant ratio [24, 10, 13], (iii) $(1+\epsilon)$-ratio PTAS [12, 23].

TSPN on Doubling Metrics. Chan and Elbassioni [7] considered $(1+\epsilon)$-approximation for TSPN on doubling metrics. They combined the notions of diameter variation, fatness and disjointness for geometric spaces, and defined for regions in general metrics the notion of α-*fat weak disjointness* (Definition 2.1). Intuitively, the regions are partitioned into Δ groups, where regions in each group should have similar diameters and each region designates a point within, such that these points are far away from one another. The regions can otherwise intersect arbitrarily, and need not even be convex or connected, where such notions might be inapplicable

*Department of Computer Science, the University of Hong Kong. {hubert,sfjiang}@cs.hku.hk

[1] A non-negative function $f(n)$ is quasi-polynomial in n if there exists a constant c such that $f(n) \leq \exp(O(\log^c n))$.

in the first place. More motivation and examples for fat weakly disjoint regions are given in [7]. The assumption that there is only a bounded number Δ of types of region diameters is necessary though, as they also showed that otherwise TSPN remains APX-hard for doubling metrics.

Using the hierarchical decomposition and dynamic programming techniques by Arora [3] and Talwar [27], they gave a QPTAS for fat weakly disjoint neighborhoods in doubling metrics. It should be noted that a PTAS was not yet known even for TSP on doubling metrics then.

In our previous unpublished manuscript [8], we designed a PTAS for TSPN based on the techniques in [7, 6]. For any parameter $0 < c < 1$, our construction gives a PTAS with running time $n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[(\frac{\Delta}{\epsilon})^{O(k)} \cdot O(\alpha)^{2k^2} \cdot o(\log^c n)]$. For the special case $\Delta = \alpha = 1, c = \frac{1}{2}$, our bound is comparable to the running time of $n^{O(1)^k} \cdot \exp[(\frac{1}{\epsilon})^{O(k)} \cdot O(1)^{k^2} \cdot \sqrt{\log n}]$ for the PTAS for TSP in [6]. Observe that the dependence on the doubling dimension k is $\exp[O(1)^{k^2}]$. In this paper, we reduce the dependence on k to $\exp[O(1)^{O(k \log k)}]$. Since the dependence on k is doubly exponential, our new scheme offers a significant improvement on the running time in terms of k, when $n \ll \exp[O(1)^{O(k^2)}]$.

Main Result. We combine the techniques of the dynamic program for TSPN [7] and the PTAS for TSP [6] to give a PTAS for TSPN with improved running time.

THEOREM 1.1. *Fix any $0 < c, \epsilon < 1$. There is a PTAS that solves the following variation of* TSPN. *Suppose in a metric space with doubling dimension at most k, there are n regions that are partitioned into Δ groups, each of which is α-fat weakly disjoint. Then, for large enough n (depending on c and ϵ), with constant probability, the algorithm returns a* TSPN *tour of length at most $(1+\epsilon) \cdot$* OPT *in time $n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[\Delta^{2+c} \cdot O(\frac{k\alpha}{\epsilon})^{O(k)} \cdot o(\log^c n)]$.*

Technical Challenges. Our PTAS for TSPN uses the high level idea of the PTAS framework for TSP [6], and in the core utilizes the dynamic program for TSPN in [7]. However, there are a number of technical hurdles, and we briefly explain why novel ideas are needed to overcome them.

(1) Heuristic to detect critical instances. In [6], a minimum spanning tree heuristic MST is computed on the points in some ball B to estimate the weight of the portion within B of some nearly optimal tour T. In our previous attempt [8], we define a similar MST heuristic on representatives of all regions W' intersecting B. However, an optimal tour might choose to visit regions W' (that partially intersect with B) using points that are far away from B, and this eventually leads to a factor $\exp[O(1)^{O(k^2)}]$ in the running time. Instead, we use the TSPN heuristic T (defined in Section 3), which itself is an approximation for the shortest tour visiting regions with small diameters intersecting B. In Lemma 4.2, we show that the new TSPN heuristic T relates to an optimal tour better than the previous MST heuristic.

Moreover, the heuristic T is applied in a bottom-up fashion starting with lower height clusters. Hence, it can be used to detect a lowest height *critical* instance, in which there must be a large number of regions with small diameters intersecting the ball in question.

(2) Resolving partially cut regions in sparse instances in the recursion. In [6], loosely speaking, after a critical instance is identified on a subset S_1 of points (by the MST heuristic), the sub-instance on S_1 is still sparse enough and can be solved with a dynamic program DP similar to [3, 27] in polynomial time to give a partial tour, which is combined with the tour solved recursively in the remaining instance. However, when regions are involved, it is an important issue to decide whether regions partially intersecting S_1 should be solved in the sparse instance, or considered in the remaining instance. Since throughout the recursion, the dynamic program DP might be called $n^{\Omega(1)}$ times, we cannot split cases to assign partially intersected regions in each level of recursion, as even two cases per recursion will lead to a running time of $2^{n^{\Omega(1)}}$. Surprisingly, we can conservatively let the sparse instance handle all regions that have non-empty intersections with S_1. Indeed, a very technical patching argument is made in Lemma 5.2 to ensure that the recursion can be applied as in [6].

In order to improve the final running time, we also use a more careful procedure to remove a critical instance. When a critical instance at distance scale s^i is discovered, instead of just picking a random radius $O(s^i)$ to cut out a ball (as in [6, 8]), we first use the TSPN heuristic T to determine a radius ρ with bounded local growth before applying randomness to cut around ρ. This idea of exploiting local growth is often applied to problems on doubling metrics. For instance, in the context of light spanners for doubling metrics [16, Lemma 4.1], a certain radius is chosen similarly using local growth rate.

(3) Bounding the number of ambiguous regions in sparse DP. In the dynamic program for TSPN in [7], the number H of *ambiguous* regions each cluster needs to keep track of is poly-logarithmic in n. However, there is a factor $2^{O(H)}$ in the running time, which gives a quasi-polynomial overhead in [7]. We improve the analysis (Corollary 6.1 and Lemma 6.2) by using the sparsity of the instance to obtain a better bound on

H with respect to both n and k. Hence, the dynamic programs in [7] and [6] can be combined together to give a PTAS with improved running time.

2 Preliminaries

We consider a finite metric space $M = (V, d)$. (For basic properties of metric spaces, we refer the reader to standard texts [11, 21].) A *ball* $B(x, \rho)$ is the set $\{y \in V \mid d(x, y) \leq \rho\}$. The *diameter* $\mathsf{Diam}(Z)$ of a set Z is the maximum distance between points in Z. A set Z of points is a ρ-packing, if any two distinct points in Z are at a distance more than ρ away from each other. Given a positive integer m, we denote $[m] := \{0, 1, 2, \ldots, m-1\}$. In this paper, we work with metric spaces with *doubling dimension* [4, 18] at most k; this means that for all $x \in V$, for all $\rho > 0$, every ball $B(x, 2\rho)$ can be covered by the union of at most 2^k balls of the form $B(z, \rho)$, where $z \in V$. For convenience, we sometimes round the doubling dimension up to the nearest integer k.

Fact 2.1. (Packing in Doubling Metrics [18]) *Suppose for some non-negative integer t, Z is a ρ-packing contained in some ball of radius $2^t \rho$ in a metric space with doubling dimension at most k. Then, $|Z| \leq 2^{(t+1)k}$.*

Problem Definition. An instance of the metric TSP with neighborhoods (TSPN) is given by a metric space $M = (V, d)$ with doubling dimension at most k and a collection of n *neighborhoods* or *regions* $W := \{P_j \mid j \in [n]\}$, where each P_j is a subset of V, and $V = \cup_j P_j$. The objective is to find a minimum TSP tour that visits at least one point from each region. As in [7], the regions are partitioned into Δ groups $\{W_l\}_{l \in [\Delta]}$, such that for some $\alpha \geq 1$, each group W_l satisfies some α-fat weak disjointness condition as follows.

Definition 2.1. (α-Fat Weakly Disjoint Regions) [7] *For $\alpha \geq 1$, a group W_l of regions are* α-fat weakly disjoint *if for some $\rho > 0$ the following conditions hold.*

1. *For each region $P \in W_l$, there exists some point $z(P) \in P$ such that the set $\{z(P)\}_{P \in W_l}$ is a ρ-packing. We say that P has* center *$z(P)$ and the regions in W_l have* core radius *ρ.*
2. *Every region P in W_l is contained in the ball $B(z(P), \alpha\rho)$.*

Lemma 2.1. (Lower Bound on Tour Length: Corollary 3.2 in [7]) *Suppose W_l is a group of regions as in Definition 2.1 such that $|W_l| > (8\alpha)^k$. Then, any tour that visits all regions in W_l must have length at least $\frac{1}{2(8\alpha)^k} \sum_{P \in W_l} \mathsf{Diam}(P)$.*

Corollary 2.1. (Sum of Truncated Diameters) *Suppose T is a tour visiting all regions in W, which consists of Δ groups of α-fat weakly disjoint regions. Then, for any real $D > 0$, $\sum_{P \in W} \min\{\mathsf{Diam}(P), D\} \leq \Delta \cdot (8\alpha)^k \cdot \max\{2w(T), D\}$.*

Claim 2.1. (Number of Intersecting Regions with Large Diameters) *Suppose $0 < \delta < 1 \leq t$ and let B be a ball with radius ts^i. Then, the number of regions with diameter at least δs^i that intersect B is at most $\Delta \cdot O(\frac{t\alpha}{\delta})^k$.*

Proof. Recall that the regions are partitioned into Δ groups of α-fat weakly disjoint regions. We give the upper bound for each group. Let D be the maximum diameter of all the regions in some group that intersect the ball B. We consider the case $D \geq \delta s^i$; otherwise, the upper bound is trivially 0.

By the definition of α-fat weak disjointness, these regions have centers that form a ρ-packing, where $\rho \geq \frac{D}{\alpha}$. Moreover, these centers lie in a ball with radius at most $ts^i + D$. Hence, by the packing property (Fact 2.1), the number of regions in this group that intersect B is at most $O(\frac{\alpha(ts^i+D)}{D})^k \leq O(\frac{\alpha t}{\delta})^k$. ■

Remark 2.1. (Assumptions on the Partition $\{W_l\}_{l \in [\Delta]}$) *We assume the existence of the partition $\{W_l\}_{l \in [\Delta]}$ of regions, such that each group is guaranteed to be α-fat weakly disjoint. We assume that only the parameter α and Δ are given to us, and our algorithm does not need to know how the regions are partitoned, and within each group W_l, our algorithm does not need to know the core radius or how the centers of the regions are assigned in Definition 2.1.*

We let $\mathsf{OPT}(S, W)$ denote an optimal tour using points in S that visits every region in W; when the context is clear, we also use $\mathsf{OPT}(S, W)$ (or just OPT) to denote the length of the tour.

Restricting the Tour inside B_0. We assume that there is a region P_0 that contains only one point p_0. For finite metrics, we can have this assumption because we can try each p_0 in P_0, and consider those TSPN tours that pass through p_0. We let R be the minimum radius of a ball centered at p_0 that intersects all regions. Suppose OPT is the length of an optimal tour. Then, it follows that $2R \leq \mathsf{OPT} \leq 2nR$. Hence, an optimal tour must be contained in the ball $B_0 := B(p_0, nR)$. Therefore, without loss of generality, we only need to consider the points in B_0.

Remark 2.2. *Since we consider a PTAS, we fix $\epsilon > 0$, and consider sufficiently large n such that $\frac{1}{\epsilon} < n$. Suppose an optimal tour visits p_j in each P_j. If we* *replace each p_j by (possibly another) point $p'_j \in P_j$ such that $d(p_j, p'_j) \leq \frac{\epsilon R}{2n}$, then we change the length of the tour by at most $\epsilon\mathsf{OPT}$. Hence, we can assume that each region has radius of either 0 or at least $\frac{\epsilon R}{2n}$. We can rescale distances such that the minimum inter-point distance among all points is 1, and the maximum distance is at most $\frac{n^2}{\epsilon} < n^3$. By Fact 2.1, we can assume that $|V| \leq n^{O(k)}$. For simplicity, we often argue that the tour returned by the algorithm has expected length at most $1+\epsilon$ times the optimal length; by using standard repetition argument, one can show this implies that with constant probability, approximation ratio $1+\epsilon$ can be obtained.*

Given $\rho > 0$, recall that a *ρ-net* for a set U of points is a subset S such that S is a ρ-packing, and every point in U is within a distance of ρ from some point in S.

Hierarchical Nets. Fix $c > 0$. As in [6], we consider some parameter $s = (\log n)^{\frac{c}{4k}} \geq 4$ (i.e., $n \geq 2^{2^{\Omega(k)}}$). Set $L := O(\log_s n) = O(\frac{k \log n}{c \log\log n})$. A greedy algorithm can construct $N_{L-1} \subseteq \cdots \subseteq N_1 \subseteq N_0 = V$ such that for each $i \in [L]$, N_i is an s^i-net for V, where we say *distance scale* s^i is of *height* i. As in [6], we use the randomized decomposition scheme defined in [5, 1].

Definition 2.2. (Single-Scale Decomposition [1]) *At height i, an arbitrary ordering π_i is imposed on the net N_i. Each net-point $u \in N_i$ corresponds to a* cluster center *and samples random h_u from a truncated exponential distribution Exp_i having density function $t \mapsto \frac{\chi}{\chi-1} \cdot \frac{\ln \chi}{s^i} \cdot e^{-\frac{t \ln \chi}{s^i}}$ for $t \in [0, s^i]$, where $\chi = O(1)^k$. Then, the cluster at u has random radius $r_u := s^i + h_u$.*

The clusters induced by N_i and the random radii form a decomposition Π_i, where a point $p \in V$ belongs to the cluster with center $u \in N_i$ such that u is the first point in π_i to satisfy $p \in B(u, r_u)$. We say that the partition Π_i cuts a set P if P is not totally contained within a single cluster.

The results in [1] imply that the probability that a set P is cut by Π_i is at most $\frac{\beta \cdot \mathsf{Diam}(P)}{s^i}$, where $\beta = O(k)$.

Definition 2.3. (Hierarchical Decomposition) *Given a configuration of random radii for $\{N_i\}_{i \in [L]}$, decompositions $\{\Pi_i\}_{i \in [L]}$ are induced as in Definiton 2.2. At the top height $L-1$, the whole space is partitioned by Π_{L-1} to form height-$(L-1)$ clusters. Inductively, each cluster at height $i+1$ is partitioned by Π_i to form height-i clusters, until height 0 is reached. Observe that a cluster has $K := O(s)^k$ child clusters.*

Hence, a set P is cut at height i iff *the set P is cut by some partition Π_j such that $j \geq i$; this happens with probability at most $\sum_{j \geq i} \frac{\beta \cdot \mathsf{Diam}(P)}{s^i} = \frac{O(k) \cdot \mathsf{Diam}(P)}{s^i}$.*

Net-Respecting Tour. As defined in [6], a tour T is net-respecting with respect to $\{N_i\}_{i \in [L]}$ and $\epsilon > 0$ if for every transition (x, y) in the tour, both x and y belong to N_i, where $s^i \leq \epsilon \cdot d(x, y) < s^{i+1}$. Given a subset $S \subseteq V$ and a set W of regions, let $\mathsf{OPT}^{nr}(S, W)$ be an optimal net-respecting tour using points in S that visits every region in W; when the context is clear, we also use $\mathsf{OPT}^{nr}(S, W)$ to denote the length of the tour.

It is shown in [6, Lemma 1.11] that net-points can be inserted between every transition of a tour T to make the tour net-respecting, while increasing the length by only a factor of $1 + O(\epsilon)$. Hence, we can assume that the optimal TSPN tour is net-respecting, but observe that the approximation algorithm needs not return a net-respecting tour.

Portals. As in [3, 27, 6], each height-i cluster is equipped with portals such that a tour is *portal-respecting* if it enters and exits a cluster only through its portals. As mentioned in [6], the portals of a cluster need not be points of the cluster itself, but are just used as entry or exit points. For a height-i cluster C, its portals is the subset of net-points in $N_{i'}$ that cover C, where i' is the maximum index such that $s^{i'} \leq \max\{1, \frac{\epsilon}{4\beta L} \cdot s^i\}$.

A transition (x, y) in a tour can be made portal-respecting in the following way. Suppose height i is the highest scale that separates the pair (x, y), and p_x and p_y are the closest height-i portals in the clusters containing x and y, respectively. Then, the transition (x, y) is replaced by (i) a portal-respecting tour from x to p_x found recursively, (ii) p_x to p_y, (iii) a portal-respecting tour from p_y to y found recursively.

Fact 2.2. (Portal-Respecting Tour) *Any tour T can be converted to a portal-respecting tour (that visits all the points in T) whose expected length is at most $1+\epsilon$ times that of the original tour, where the randomness is over the hierarchical decomposition*

Since a height-i cluster has diameter $O(s^i)$, by Fact 2.1, the cluster has at most $m := O(\frac{\beta L s}{\epsilon})^k$ portals.

(m, r)-Light Tour. An *(m, r)-light tour* is a portal-respecting tour that visits each cluster only through its m portals, and crosses each cluster at most r times; a tour crosses a cluster when it either enters or exits a cluster.

A dynamic program can be used [27, 7] to find the best (m, r)-light tour whose length is at most $(1+\epsilon)$ times the optimal with $r = O(m)$, which leads to only a QPTAS. The idea in [6] is to exploit some sparsity conditions to reduce r in order to obtain a PTAS.

3 Overview of Method

We adopt the PTAS framework for TSP in [6], and apply it to TSPN.

Dynamic Programming. We use a subroutine $\mathsf{DP}(S, W)$, which can be applied when the instance is sparse according to some heuristic described below. The subroutine is described in Section 6, and is a dynamic program that returns a tour in S visiting all regions W. Recall that k is an upper bound on the doubling dimension.

Sparsity Heuristic. In order to estimate the local sparsity, given a net-point $u \in N_i$ at height i and $t > 0$, we consider the heuristic $\mathsf{T}^{(i)}(u, t)$, which is some constant approximation (say, with ratio 1.000001) of the length of the shortest net-respecting tour contained within $B(u, (t + \delta) \cdot s^i)$ that visits all regions with diameter at most $\delta \cdot s^i$ that intersect $B(u, ts^i)$, where $\delta := \frac{\epsilon}{1000k}$. Observe that we only try to compute $\mathsf{T}^{(i)}$ after checking that the heuristic $\mathsf{T}^{(j)}$ is small for all $j < i$. Hence, as we shall see later, we can use DP to estimate $\mathsf{T}^{(i)}$.

Given a set S of points and a set W of regions, we give a high level description of our main algorithm $\mathsf{ALG}(V, W)$ that returns a tour in V visiting all regions in W.

1. **Base Case.** If $|W| = n$ is smaller than some constant threshold, solve the problem by brute force, recalling that $|V| \leq O(\frac{n^2}{\epsilon})^k$ (See Remark 2.2).
2. **Sparse Instance.** If for all $i \in [L]$, for all $u \in N_i$, $\mathsf{T}^{(i)}(u, 4)$ is at most $q_0 \cdot s^i$ (where the reason for choosing $q_0 := \Delta \cdot O(\frac{k^2 \alpha s}{\epsilon})^k$ is given in Lemma 5.2), call the subroutine $\mathsf{DP}(V, W)$ to return a tour, and terminate.
3. **Identify Critical Instance.** Otherwise, let i be the smallest height such that there exists $u \in N_i$ with *critical* $\mathsf{T}^{(i)}(u, 4) > q_0 \cdot s^i$; in this case, choose $u \in N_i$ such that $\mathsf{T}^{(i)}(u, 4)$ is maximized.
4. **Remove Critical Instance.** Decompose (possibly using randomness) $W := W_1 \cup W_2$ such that loosely speaking W_1 are the regions around u at distance scale s^i, and pick $S_1 \subseteq V$ to be some ball around u with radius $O(ks^i)$ such that (S_1, W_1) is "sparse" enough.
5. Call the subroutine $T_1 := \mathsf{DP}(S_1, W_1 + \{u\})$, and solve $T_2 := \mathsf{ALG}(V, W_2 + \{u\})$ recursively; combine the tours T_1 and T_2 at the point u to return a tour.

In order to complete the description of the algorithm and prove that it has the desired properties (approximation ratio and running time), we need to supply the following details.

Define DP to handle "sparse" instance (V, W). We define a dynamic program in Section 4 that handles sparse instances, and in particular, has the following meta-property.

(MP1) If (S, W) is "sparse" enough, then $\mathsf{DP}(S, W)$ runs in polynomial time, and with high probability (say at least $1 - \frac{1}{2n}$), returns a tour in S visiting all regions in W whose length is at most $(1 + \epsilon)$ times $\mathsf{OPT}(S, W)$. The formal version is obtained by Lemma 4.1 and Corollary 6.2.

Define decomposition procedure to remove critical instance. Suppose i is the smallest height such that there exists $\mathsf{T}^{(i)}(u, 4) > q_0 \cdot s^i$, where $u \in N_i$ is chosen to maximize the heuristic. In Section 5, we shall define a (deterministic) ball S_1 centered at u with radius $O(ks^i)$ and a (random) ball $B \subset S_1$. Then, we set $W_1 := \{P \cap S_1 : P \cap B \neq \emptyset, P \in W\}$, and $W_2 := \{P \in W : P \cap B = \emptyset\}$; observe that if $q_0 > 10$, then $|W_1| \geq 2$. We shall prove that the decomposition has the following meta-property.

(MP2) The above randomized procedure produces a "sparse" enough instance $(S_1, W_1 + \{u\})$ such that $\mathbb{E}[\mathsf{OPT}(S_1, W_1 + \{u\})] \leq \frac{1}{1-\epsilon} \cdot (\mathsf{OPT}^{nr}(V, W) - \mathbb{E}[\mathsf{OPT}^{nr}(V, W_2 + \{u\})])$, where expectation is over the random radius of B. The formal version is obtained by Corollary 4.1 and Lemma 5.2.

Proof of Theorem 1.1: We show how (MP1) and (MP2) imply our main result.

Analysis of approximation ratio. We follow the inductive proof as in [6] to show that with constant probability (where the randomness comes from DP), $\mathsf{ALG}(V, W)$ returns a tour with expected length at most $\frac{1+\epsilon}{1-\epsilon} \cdot \mathsf{OPT}^{nr}(V, W)$, where expectation is over the randomness of decomposing critical instances in (MP2).

Observe that in $\mathsf{ALG}(V, W)$, the subroutine DP is called at most $\mathrm{poly}(n)$ times (either explicitly in the recursion or estimating the heuristic $\mathsf{T}^{(i)}$). Hence, with constant probability, all the tours returned by all instances of DP have appropriate lengths in (MP1).

Suppose T_1 and T_2 are the tours returned by $\mathsf{DP}(S_1, W_1 + \{u\})$ and $\mathsf{ALG}(V, W_2 + \{u\})$, respectively. By (MP1), T_1 has length at most $(1 + \epsilon) \cdot \mathsf{OPT}(S_1, W_1 + \{u\})$, while the induction hypothesis states that $\mathbb{E}[w(T_2)] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \mathsf{OPT}^{nr}(V, W_2 + \{u\})$.

By (MP2), $\mathbb{E}[\mathsf{OPT}(S_1, W_1 + \{u\})] \leq \frac{1}{1-\epsilon} \cdot (\mathsf{OPT}^{nr}(V, W) - \mathbb{E}[\mathsf{OPT}^{nr}(V, W_2 + \{u\})])$. Hence, it follows that $\mathbb{E}[w(T_1) + w(T_2)] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \mathsf{OPT}^{nr}(V, W) = (1 + O(\epsilon)) \cdot \mathsf{OPT}(V, W)$, achieving the desired ratio.

Analysis of running time. We see that the decomposition procedure in (MP2) is carried out at most $O(n)$ times. Hence, the running time is dominated by the calls to the dynamic program DP in (MP1). We shall show the argument in [6] can be augmented with regions to still achieve polynomial time. In the rest of the paper, we shall prove formal versions of (MP1) and (MP2). ■

4 Sparse Heuristic T Gives Sparse Optimal Tour

In this section, we give formal treatments for (MP1) in Section 3. A tour T can be interpreted as a set of edges with end-points in V; given $B \subseteq V$, $T|_B$ is the set of edges in T such that both end-points are in B.

Sparse Tour [6]. A tour T is q-sparse with respect to $\{N_i\}_{i\in[L]}$, if for all $i \in [L]$, for all $u \in N_i$, the weight $w(T|_{B(u,3s^i)})$ of the portion of tour T within the ball $B(u, 3s^i)$ is at most $q \cdot s^i$. We use the previous result in [6, Lemma 3.1] paraphrased as follows.

Lemma 4.1. (q-Sparsity Allows (m,r)-Lightness [6]) *Suppose a net-respecting T is q-sparse with respect to $\{N_i\}_{i\in[L]}$. Moreover, for each $i \in [L]$, for each $u \in N_i$, point u samples $O(\log|V|) = O(k \log n)$ independent random radii as in Definition 2.2. Then, with constant probability, there exists a configuration from the sampled radii that defines a hierarchical decomposition, under which there exists an (m,r)-light tour T' that visits all the points in T and has weight $w(T') \leq (1+\epsilon)\cdot w(T)$, where $m := O(\frac{sk\log_s n}{\epsilon})^k$ and $r := O(1)^k \cdot q \log_s \log n + O(\frac{k}{\epsilon})^k + O(\frac{s}{\epsilon})^k$.*

We next show that the heuristic $\mathsf{T}^{(i)}(u,t)$ can be used to detect sparse tours visiting regions, which is analogous to [6, Lemma 1.12(i)].

Lemma 4.2. (Heuristic T Gives Sparse Optimal Tours) *Suppose T is an optimal net respecting tour for instance (S, W). Then, for any height-i net point u and integer $t \geq 1$,*

$$w(T|_{B(u,ts^i)}) \leq \mathsf{T}^{(i)}(u, t+1) + \Delta \cdot O(\tfrac{k\alpha ts}{\epsilon})^k \cdot s^i.$$

Proof. We denote $B := B(u, ts^i)$ and $B' := B(u, (t+1)\cdot s^i)$. The idea of the proof is to delete edges in $T|_B$ and some other edges from T, and add edges corresponding to the heuristic $\mathsf{T}^{(i)}(u, t+1)$ and additional edges to construct another net-respecting tour T' that visit all regions in W. Then, the optimality of T implies that $w(T) \leq w(T')$, and this gives an upper bound on the weight of edges in $T|_B$. Define l to be such that $s^l \leq \frac{\epsilon}{2}\cdot s^i < s^{l+1}$

After we delete edges in $T|_B$, the remaining edges in T form path segments. Suppose x and y are end-points of such a path segment remaining in T, where $x, y \in B$. We need to add edges to make sure all regions are still visited, and to patch end-points (such as x and y) of path segments to form a tour again.

For forming a connected graph later, we first add a minimum spanning tree F on $N_l \cap B(u, (t+2)s^i)$. This has cost $O(\frac{st}{\epsilon})^k \cdot ts^i$.

To ensure all regions are still visited, we add edges in the tour corresponding to the heuristic $\mathsf{T}^{(i)}(u, t+1)$, which is a tour within $B(u, (t+1+\delta)s^i)$ that visits all regions with diameter at most $\delta \cdot s^i$ that intersect $B' = B(u, (t+1)\cdot s^i)$, where $\delta = \Theta(\frac{\epsilon}{k})$. By Claim 2.1, at most $O(\frac{k\alpha t}{\epsilon})^k$ regions from each group with diameter at least $\delta \cdot s^i$ can intersect B'. Hence, edges can be added to makes sure these regions with large diameters are connected to the nearest net-point in N_l with cost $\Delta \cdot O(\frac{k\alpha t}{\epsilon})^k \cdot s^i$.

We next describe how to patch end-points x and y of a remaining path segment P in T. Observe that x and y are in B, and the next point after x on P escapes B.

- Suppose the whole segment P lies in B'. Then, all regions that P might visit have already been taken care of. Hence, the segment P can be removed, and no patching is needed.
- Suppose traversing along P starting from x, the first node encountered that is outside B' is x'', and the previous point x' is still inside B'. We shall remove edges from x to x', and connect x' to the nearest net-point in N_l. We shall do a charging argument such that this step does not incur any net extra cost. For the case $d(x, x') \leq \frac{\epsilon}{2}\cdot s^i$, we have $d(x', x'') \geq \frac{s^i}{2}$, and hence, x' is already in N_l, because T is net-respecting. On the other hand, for the case $d(x, x') > \frac{\epsilon}{2}\cdot s^i$, the edges removed from x to x' can be used to pay for the cost of the edge connecting x' to the net-point in N_l.

  The same procedure is applied to the other end-point y.

Observe that edges for the tour $\mathsf{T}^{(i)}(u, t+1)$ are already net-respecting. Other edges can be made net-respecting by incurring a small multiplicative factor $1 + O(\epsilon)$.

Hence, we have $w(T|_B) \leq \mathsf{T}^{(i)}(u, t+1) + \Delta \cdot O(\frac{k\alpha ts}{\epsilon})^k \cdot s^i$. ∎

As we shall see later, we consider the heuristic with $t = 3$, and set $q_0 := \Delta \cdot O(\frac{k^2\alpha s}{\epsilon})^k$. We have the following corollary.

Corollary 4.1. *If ALG in Section 3 is run with threshold $q_0 := \Delta \cdot O(\frac{k^2\alpha s}{\epsilon})^k$ to determine critical instances, then instances $(S_1, W_1 + \{u\})$ passed to DP will have q-sparse net-respecting optimal tours, where $q := 2q_0$.*

Lemma 4.1 and Corollary 4.1 ensure the existence of an (m,r)-light TSPN tour that is $(1+\epsilon)$-optimal. We describe a dynamic program to compute such a tour in Section 6, which also includes the analysis of running time.

5 Identifying and Removing Critical Instances

As mentioned in the description of $\mathsf{ALG}(V,W)$ defined in Section 3, we need to describe exactly how a critical instance is removed.

Removing Critical Instances. Recall that $q_0 := \Delta \cdot O(\frac{k^2 \alpha s}{\epsilon})^k$, and i is the smallest height such that there exists $u \in N_i$ with heuristic $\mathsf{T}^{(i)}(u,4) > q_0 \cdot s^i$, where u is chosen to maximize the heuristic. Recall that the heuristic $\mathsf{T}^{(i)}$ itself is an estimate for a sparse instance of TSPN with approximation ratio slightly larger than 1 (say, 1.000001). For simplicity, we use the notation $a \lesssim b$ to mean $a \leq 1.00001b$.

Choose an integer $\lambda \in \{0,1,\ldots,k-1\}$ such that $\mathsf{T}^{(i)}(u,6+2\lambda) \leq 30k \cdot \mathsf{T}^{(i)}(u,4+2\lambda)$. We claim that such a λ must exist. Otherwise, we have $\mathsf{T}^{(i)}(u,4+2k) > (30k)^k \cdot \mathsf{T}^{(i)}(u,4)$. Suppose N is the set of net-points in $N_i \cap B(u,(5+2k)s^i)$. Then, a tour visiting all the regions with diameter at most $\delta \cdot s^i$ intersecting $B(u,4+2k)$ can be formed from the tours corresponding to $T^{(i)}(v,4)$ over $v \in N$ and a minimum spanning tree on N. This has cost at most $(4(5+2k))^k \cdot (\mathsf{T}^{(i)}(u,4)+(5+2k)s^i) < \frac{1}{10} \cdot (30k)^k \cdot \mathsf{T}^{(i)}(u,4)$, from the choice of u and q_0.

Define $S_1 := B(u,(5+2\lambda)s^i)$. Sample $h \in [0,\frac{1}{2}]$ uniformly at random (as opposed to using distribution Exp_i as in Definition 2.2); let $B := B(u,(4+2\lambda+h)s^i)$.

We set $W_1 := \{P \cap S_1 : P \cap B \neq \emptyset, P \in W\}$, and $W_2 := \{P \in W : P \cap B = \emptyset\}$. Recall that $(S_1, W_1 + \{u\})$ is passed to DP, while $(V, W_2 + \{u\})$ is solved recursively by ALG.

Lemma 5.1. (Critical Instance Gives a Lower Bound on Tour Length) *Suppose T_1 is a tour that visits all regions in W_1 and u. Then, $q_0 \cdot s^i < \mathsf{T}^{(i)}(u,4) \lesssim \mathsf{T}^{(i)}(u,4+2\lambda) \lesssim w(T_1)$.*

Proof. The first strict inequality follows from the construction that the instance is critical. The next two approximate inequalities can be proved in the same manner. Both are in the form $T^{(i)}(u,r) \lesssim w(\widehat{T})$.

In both cases, the tour $\widehat{T}$ visits all regions with diameter at most $\delta \cdot s^i$ that intersect $B(u, rs^i)$. Hence, by shortcutting and using the triangle inequality and converting the shortcutting edges net-respecting, we can produce a tour $\overline{T}$ (with weight $w(\overline{T}) \leq (1+O(\epsilon)) \cdot w(\widehat{T})$) that is in $B(u,(r+\delta)s^i)$ and visits all the regions that are also supposed to be visited by the tour corresponding to $\mathsf{T}^{(i)}(u,r)$. However, since $\mathsf{T}^{(i)}(u,r)$ is only an approximation to some corresponding optimal tour, we have $\mathsf{T}^{(i)}(u,r) \leq 1.000001 w(\overline{T}) \lesssim w(\widehat{T})$, as required. ■

The following result is the formal version of (MP2) in Section 3; it is an analogue of [6, Lemma 3.3], and turns out to be the most technical part to adapt the argument for TSPN.

Lemma 5.2. (Removing Critical Instances) *Suppose S_1, W_1 and W_2 are as defined above, and T is an optimal net-respecting tour in V visiting regions in W. We set $q_0 := \Delta \cdot O(\frac{k^2 \alpha s}{\epsilon})^k$. Then, for each random $h \in [0,\frac{1}{2}]$, there exist tours T_1 and T_2 such that the following holds.*

1. *Tour T_1 is in S_1 and visits all regions in W_1 and u.*
2. *Tour T_2 is net-respecting and visits all regions in W_2 and u.*
3. *$\mathbb{E}[w(T_1)] \leq \frac{1}{1-\epsilon} \cdot (w(T) - \mathbb{E}[w(T_2)])$, where the expectation is over random $h \in [0,\frac{1}{2}]$.*

Proof. Recall $\delta = \frac{\epsilon}{1000k}$, and let $\eta := \epsilon\delta$. Let l be the largest height such that $s^l \leq \max\{1, \eta s^i\}$, and $\rho := 4+2\lambda$.

Then, $S_1 := B(u,(\rho+1)s^i)$ and $B := B(u,(\rho+h)s^i)$, where h is sampled from $[0,\frac{1}{2}]$ uniformly at random. We also write $S_0 := B(u,\rho s^i) \subset B \subset S_1$.

Given some $h \in [0,\frac{1}{2}]$, we shall construct T_1 and T_2 using edges in T and adding extra edges. In the construction, we ensure that (1) each tour visits all the regions that it is supposed to visit, (2) the edges added form a connected component such that the corresponding tour can be formed (by possibly shortcutting edges).

In the cost analysis, some edges are charged to T once. Typically, these edges are path segments in T, and the intermediate nodes have degree 2. When a tour is being formed finally, it is important that these edges will only be used once. As mentioned above, during the construction, some edges are added to make a connected component. When a tour is formed, these edges could be used a multiple number of times (say, at most 10 times), whose cost is charged to $\epsilon \cdot w(T_1)$ via Lemma 5.1.

We next describe the construction in stages, and state the cost involved in each stage.

Ensuring Connectivity. Define $N := N_l \cap S_1$ to be the net-points inside S_1. For each of T_1 and T_2, we add a minimum spanning tree F on N, which has cost $O(\frac{\rho s}{\eta})^k \cdot \rho s^i$. Since $u \in N$, both T_1 and T_2 visit u. We remark that as we add edges in the construction, we make sure that only nodes in N or nodes connected to N (with edges of length at most ηs^i) can have odd degrees. Hence, to form each of the tours T_1 and T_2, the standard technique of considering Euler tour on the tree F will incur a cost that is a constant factor (say, at most 10 times) of $w(F)$. By the choice of q_0 and Lemma 5.1, this will account for at most $\frac{\epsilon}{4} \cdot w(T_1)$.

We partition the edges in T into three sets: (i) E^{in}: edges totally within B, (ii) E^{cr}: edges crossing B, (iii) E^{out}: edges totally outside B.

Ensuring Regions Are Visited: Part I. We add edges in E^{in} to T_1 and edges in E^{out} to T_2. Hence, we have ensured that T_2 will visit all the regions that do not intersect B. However, E^{in} might not be enough to visit all regions that intersect B. We will take care of this in Part II. For the time being, we describe how the end-points of edges in E^{cr} are connected to the spanning tree F in each of T_1 and T_2.

Suppose $e = \{x, x'\} \in E^{cr}$ such that $x \in B$ and $x' \notin B$. If $w(e) \geq \delta s^i$, then both x and x' are in N because T is net-respecting; in this case, we add e to T_2 (by extending the path segment in T_2), and the end-point x is N, which is already spanned by the tree F. Observe that edges in T (whose cost is charged to $w(T)$) added to T_2 are already net-respecting, and other edges added to T_2 can be made net-respecting by increasing a small factor that can still be charged to $\epsilon \cdot w(T_1)$.

If $w(e) < \delta s^i$, then we add the edge e to T_1. We remark that at this point, in our charging scheme, the part of the cost charged exactly once to $w(T)$ has been used, and any further cost will be charged to $\epsilon \cdot w(T_1)$. Next, we connect x' to the closest net-point in N, which incurs a cost of at most ηs^i. Observe that we need to charge this to $\epsilon \cdot w(T_1)$ somehow. We use the randomness due to h, and observe that this cost is charged to e only if the edge e is cut by B, which happens with probability at most $\frac{2w(e)}{s^i}$. Moreover, e can be charged with non-zero probability only if $e \in T|_{S_1}$. Hence, the expected cost due to this part is $C_1 := 2\eta \cdot w(T|_{S_1})$.

Ensuring Regions Are Visited: Part II. Observe that T_1 is supposed to visit all regions that intersect B. However, T might visit such a region outside B. We next add edges to make sure such regions that might be missed by E^{in} are visited by T_1.

By Claim 2.1, at most $\Delta \cdot O(\frac{\rho\alpha}{\delta})^k$ regions with diameter at least δs^i can intersect B. Hence, the cost to connect each one of them to the closest net-point in N is at most ηs^i. Therefore, by the choice of q_0 and Lemma 5.1, the cost of connecting them to N is at most $\frac{\epsilon}{4} \cdot w(T_1)$.

We next consider the regions with diameter at most δs^i that intersect B. Define $B_1 := B(u, (\rho + h + \delta)s^i)$ and $B_2 := B(u, (\rho + h + 3\delta)s^i)$. Hence, we have $S_0 \subset B \subset B_1 \subset B_2 \subset S_1$. We define the annulus $A := B_1 \setminus B$, and include all path segments in $T|_A$ to T_1. Observe that we need to include even the trivial path segments consisting of single nodes. At this point, we have ensured that T_1 visits all regions that intersect B.

We next describe how the end-points of the path segments in $T|_A$ are connected. Suppose x' is an end-point of such a path segment. We try to extend the path segment along the tour T, and consider the following cases.

(i) The next point x is in B. If $d(x, x') < \delta s^i$, then the edge $\{x, x'\}$ has already been added to T_1 in Part I; otherwise, by the net-respecting property of T, x is a point in N.

(ii) The next point x is not in B, i.e., $x \notin B_1$. Here, we start from $p = x'$, and try to extend the path segment along the tour T that goes outside B_1. Whenever we have $p \in B_2$ and the next hop $\{p, p'\}$ has distance at least δs^i, then we have encountered a net-point p in N, because T is net-respecting. Otherwise, the tour either returns to the annulus A or exits the ball B_2. If the former happens first, then this tour segment is merged with the next one in $T|_A$; if the latter happens first, suppose x'' is the last node in B_2 before the tour leaves B_2.

(iii) If $d(x', x'') \leq \delta s^i$, then the jump from x'' to outside B_2 has distance at least δs^i, which means that x'' is in N. Otherwise, if $d(x', x'') > \delta s^i$, we shall use the edges from x' to x'' along T to pay for the cost of connecting x' to the nearest point in N.

Hence, to summarize, the cost of taking care of the regions with diameter at most δs^i intersecting B can be charged to the edges of T that are totally in the annulus $A_2 := B_2 \setminus B$, whose width is 3δ. Because of the randomness of h, for any edge e regardless of its length, the probability that e lies totally in A_2 is at most 10δ. On the other hand, an edge can be in A_2 with non-zero probability only if $e \in T|_{S_1}$. Hence, the expected cost is at most $C_2 := 10\delta \cdot w(T|_{S_1})$.

Analyzing the cost $\mathbb{E}[w(T_1) + w(T_2)]$. In the description above, we state that the expected cost can be charged to $w(T)$ or $\epsilon \cdot w(T_1)$. The charging is straightforward in most cases, but we elaborate how the expected cost $C_1 + C_2$ is charged to $\epsilon \cdot w(T_1)$.

Observe that C_2 dominates C_1, and $w(T|_{S_1}) \leq \mathsf{T}^{(i)}(u, \rho + 2) + \Delta \cdot O(\frac{k\alpha\rho s}{\epsilon})^k \cdot s^i$, where the inequality follows from Lemma 4.2. It is sufficient to analyze $\mathsf{T}^{(i)}(u, \rho + 2) + \Delta \cdot O(\frac{k\alpha\rho s}{\epsilon})^k \cdot s^i$.

By the choice of λ, we have $\mathsf{T}^{(i)}(u, \rho + 2) = \mathsf{T}^{(i)}(u, 6+2\lambda) \leq 30k \cdot \mathsf{T}^{(i)}(u, 4+2\lambda) \lesssim 30k \cdot w(T_1)$, where the last approximate inequality follows from Lemma 5.1. Therefore, by the choice of q_0, we have $w(T|_{S_1}) \leq 20k \cdot w(T_1)$. Hence, we have $C_1 + C_2 \leq 20\delta \cdot w(T|_{S_1}) \leq \frac{\epsilon}{4} \cdot w(T_1)$.

Hence, in conclusion, the following inequality holds with probability 1: $\mathbb{E}[w(T_1) + w(T_2)] \leq w(T) + \epsilon \cdot w(T_1)$. Taking expectation gives the required result. ■

6 Dynamic Program for TSPN

In Section 4, we see that the heuristic $\mathsf{T}^{(i)}(u, t)$ can ensure that the instance (S, W) received by DP defined in Section 3 has a sparse optimal tour, which by Lemma 4.1 implies the existence of an (m, r)-light $(1+\epsilon)$-optimal tour for appropriate values of m and r. In this section, we describe details of the dynamic program $\mathsf{DP}(S, W)$ that finds such a tour in S visiting all regions in W. The dynamic program is a combination of the ones in [7] and [6], which are themselves extensions of the ones in [3, 27]. We first review some properties for TSPN as in [7].

Common and Rare Groups. Recall that the set W of regions are grouped into sets $\{W_l\}_{l\in[\Delta]}$. We say a group W_l is *common* if $|W_l| > (8\alpha)^k$, and otherwise is *rare*. Let $W_c := \cup_{l:|W_l|>(8\alpha)^k} W_l$ be the regions in common groups, and let $W_r := W \setminus W_c$ be those in rare groups. By Lemma 2.1, $\sum_{P\in W_c} \mathsf{Diam}(P) \leq 2\Delta \cdot (8\alpha)^k \mathsf{OPT}$, and observe that $|W_r| \leq \Delta \cdot (8\alpha)^k$.

Configuration of Random Radii. In Lemma 4.1, we see a procedure that samples $O(k \log n)$ random radii for each net-point at each height. By a *configuration of random radii*, we mean picking some radius for each net-point at each height. Recall that a configuration of random radii induces a hierarchical decomposition in Definition 2.3.

Given a hierarchical decomposition, the idea of *anchor points* and *potential sites* are used in [7] to give an efficient way to keep track of which clusters are responsible for which regions. Since later we shall consider different configurations of random radii, we give an alternative description here. Let $0 < \gamma < 1$ be some parameter associated with the detour made when a region is visited via an anchor point.

Anchor Points for Making Detours for Common Regions W_c. Consider some tour T that visits all regions in W. Given a hierarchical decomposition induced by a configuration of random radii, we show how anchor points are assigned to a region P in a common group. Moreover, we describe the detour made to T in each case.

Suppose that region P is first divided at height-i, i.e., it is totally contained in some height-$(i+1)$ cluster.

1. Suppose $\mathsf{Diam}(P) \leq \gamma s^i$. Then, we pick an arbitrary point $p \in P$ and replace the region P with the singleton $\{p\}$; we emphasize that in this case p is NOT an *anchor point* for the region P. Observe that visiting the region via p will cost a detour of length at most $2\mathsf{Diam}(P)$.
2. Suppose j is the largest height such that $s^j \leq \max\{1, \gamma s^i\}$. For each height-$j$ cluster C_u (centering at some $u \in N_i$) that intersects region P, assign u as an *anchor point* from cluster C_u for region P. We say that u is the *potential site* for the cluster C_u. Observe that u might not be a point in C_u; when the potential site u is *activated*, point u acts like a special portal for the cluster C_u to visit regions as follows. If the tour T visits a point p in C_u, then a detour can be made to visit the activated potential site u, and then to a point in P closest to u, after which we backtrack to p to finish the detour; since the cluster C_u has radius at most $2s^j$, this detour has length at most $8s^j \leq 8\gamma s^i$.

Note that we do not know which point in the region the optimal tour would visit, but we can ensure that the correct point would have an anchor point within a distance of $2\gamma s^i$.

The following lemma gives a slightly better analysis than [7, Lemma 3.3]. This simple improvement later removes the dependence of γ on $L = O(\log_s n)$, which ensures that the number of regions each cluster needs to keep track of is independent of n.

Lemma 6.1. (Approximate Point Location for Divided Regions) *Suppose a hierarchical partition is sampled as in Definition 2.3. Suppose a detour is made to visit a common region P as above. Then, the expected increase in the length of the tour is at most $O(\beta\gamma \log_s \frac{1}{\gamma}) \cdot \mathsf{Diam}(P)$.*

Proof. First, observe that the probability that a region P with $D := \mathsf{Diam}(P)$ is first divided at the height-i is at most $\min\{1, O(\beta) \cdot \frac{D}{s^i}\}$, as stated in Definition 2.2. We consider different cases for i.

1. Case $s^i \geq \frac{D}{\gamma}$. We have $D \leq \gamma s^i$, and so P is replaced by a singleton, and the detour has length at most $2D$. Suppose l is the smallest height such that $s^l \geq \frac{D}{\gamma}$. Then, summation over $i \geq l$ gives contribution $\sum_{i\geq l} O(\beta) \cdot \frac{D}{s^i} \cdot 2D \leq O(\beta) \cdot \frac{D}{s^l} \cdot O(D) \leq O(\beta\gamma) \cdot D$.
2. Case $D \leq s^i < \frac{D}{\gamma}$. There are $\log_s \frac{1}{\gamma}$ such i's, each of which gives contribution at most $O(\beta) \cdot \frac{D}{s^i} \cdot \gamma s^i = O(\beta\gamma) \cdot D$. Summation over i in this range gives contribution $O(\beta\gamma \log_s \frac{1}{\gamma}) \cdot D$.
3. Case $s^i < D$. In this case, the probability of P cut at height-i is at most 1; and the sum of contribution over such i's is at most $O(\gamma) \cdot D$.

Hence, the expected increase in length after the detour is at most $O(\beta\gamma \log_s \frac{1}{\gamma}) \cdot \mathsf{Diam}(P)$, as required. ■

Combining Lemma 2.1 and Lemma 6.1, we show that γ can be chosen such that the detour will cause the tour to increase by only ϵ fraction of the optimal tour.

Corollary 6.1. (Low Cost Detours) *Suppose a hierarchical decomposition is sampled as in Definition 2.3, and the portal assignment procedure is carried out to make detour for each common region as described above. Then, the expected increase in the tour length is at most $\Delta \cdot (8\alpha)^k \cdot O(\beta\gamma \log_s \frac{1}{\gamma}) \cdot \mathsf{OPT}$, where $\beta = O(k)$. In particular, we can choose $\frac{1}{\gamma} = \frac{\Delta\beta\cdot O(\alpha)^k}{\epsilon} \log \frac{\Delta\beta\cdot O(\alpha)^k}{\epsilon}$*

(independent of n) such that this expected increase is at most $\epsilon \cdot \mathsf{OPT}$.

Ambiguous Regions for a Cluster. Recall that, ultimately, we want to limit the number of regions that intersect a cluster for which the dynamic program has to explicitly consider. Given a cluster C at height-i, its *ambiguous regions* are those regions P partially intersecting C that satisfy one of the following properties.

1. The region P is in W_r, i.e., it is in a rare group; observe that no anchor point is assigned for regions in a rare group.
2. The cluster C or any of its descendant clusters contain potential sites that can be anchor points for the region P.

According to [7, Lemma 3.5], the number of ambiguous regions a cluster needs to consider is $\Delta \cdot O(\frac{\alpha}{\gamma})^k$. However, since we have $\frac{1}{\gamma} = \Omega(1)^k$, this could lead to a factor of $2^{2^{\Theta(k^2)}}$ in the final running time. Here, we improve the upper bound on the number of ambiguous regions using the sparsity of the heuristic.

Lemma 6.2. (Number of Ambiguous Regions) *Suppose in an instance for all but the top height i, for all $u \in N_i$, $\mathsf{T}^{(i)}(u, 4) \le q_0 \cdot s^i$. Then, the number of ambiguous regions for any cluster is at most $H := \Delta \cdot O(\frac{\alpha}{\epsilon})^k \cdot \frac{q_0}{\gamma}$.*

Proof. Consider a height-i cluster C, whose center is $u \in N_i$ and diameter is $O(s^i)$. The number of ambiguous regions in rare groups is at most $\Delta \cdot O(\alpha)^k$. We next focus on the common groups.

If some region P has diameter at most γs^i, then P cannot be ambiguous, because in the approximate point location procedure, an arbitrary point $p \in P$ is picked to replace P.

Therefore, it suffices to bound the number of ambiguous regions of diameter at least γs^i. Observe that, by α-fat weakly disjointness (Claim 2.1), the number of regions of diameter at least ϵs^i is at most $\Delta \cdot O(\frac{\alpha}{\epsilon})^k$.

Hence, it remains to bound the number of ambiguous regions of diameter in $[\gamma s^i, \epsilon s^i]$ that intersect cluster C. Define W to be the set of these regions. Observe that the regions in W are totally contained in $B(u, 3s^i)$.

By Corollary 2.1, $\sum_{P \in W} \mathrm{diam}(P) \le \Delta \cdot O(\alpha)^k \cdot \mathsf{T}^{(i)}(u, 4) \le \Delta \cdot O(\alpha)^k \cdot q_0 s^i$. Hence, it follows that $|W| \le \frac{\sum_{P \in W} \mathrm{diam}(P)}{\gamma s^i} \le \Delta \cdot O(\frac{\alpha}{\epsilon})^k \cdot \frac{q_0}{\gamma}$.

Combining the above cases gives the required result. ∎

6.1 Description of Dynamic Program for TSPN

Our dynamic program DP is a combination of the dynamic programs in [7] and [6]. In [6], the number of random radii considered by each net-point at each height is $O(k \log n)$. To avoid considering an exponential number of configurations, doubling dimension is used to exploit the locality of the hierarchical decomposition. We first describe the information needed to identify each cluster at each height.

Information to Identify a Cluster. Each cluster is identified by the following information.

1. Height i and cluster center $u \in N_i$. This has $L \cdot O(n^k)$ combinations, recalling that $|N_i| \le O(n^k)$.
2. For each $j \ge i$, and $v \in N_j$ such that $d(u, v) \le O(s^j)$, the random radius chosen by (v, j). Observe that the space around $B(u, O(s^i))$ can be cut by net-points in the same or higher heights that are nearby with respect to their distance scales. As argued in [6], the number of configurations that are relevant to (u, i) is at most $O(k \log n)^{L \cdot O(1)^k} = n^{\frac{1}{c} \cdot O(1)^k}$, where $L = O(\log_s n)$ and $s = (\log n)^{\frac{c}{4k}}$, where $c > 0$ is fixed in advance.
3. For each $j > i$, which cluster at height j (specified by the cluster center $v_j \in N_j$) contains the current cluster at height i. This has $O(1)^L = n^{O(\frac{k}{c \log \log n})}$ combinations.

Therefore, the whole dynamic program considers at most $n^{\frac{1}{c} \cdot O(1)^k}$ clusters. As in [3, 27], the dynamic program looks for the best (m, r)-light tour, where the values of m and r are determined by Lemmas 4.1, Corollary 4.1 and Lemma 5.2 as follows:

$m := O(\frac{sk \log_s n}{\epsilon})^k$ and $r := \Delta \cdot O(\frac{k^2 \alpha}{\epsilon})^k \cdot s^k \cdot \log_s \log n$.

As in the case [7], we look for a tour that visits every region. We describe the *attributes* used to index each entry of a cluster.

Attributes of a Cluster Entry. As in [7], each cluster C has a number of entries, each of which is indexed by the following attributes. Suppose C is at height i and has center $u \in N_i$.

1. A collection I of portal entry/exit points. Recall that (m, r)-lightness implies that $|I| \le r$, and there are at most m^{2r} combinations.
2. A bit vector of length equal to the number of ambiguous regions that cluster C has. Each such bit indicates whether the cluster is responsible for the corresponding ambiguous region.
   Observe that the information used to identify the cluster C specifies how the space in $B(u, O(s^i))$ is cut at height j, for $j \ge i$. Hence, it is sufficient to determine which are the ambiguous regions for C. By Lemma 6.2, the number of ambiguous regions for a cluster is at most
   $H := \Delta \cdot O(\frac{\alpha}{\epsilon})^k \cdot \frac{q_0}{\gamma} = \frac{\Delta}{\gamma} \cdot O(\frac{k\alpha}{\epsilon})^{2k} \cdot s^k$, and so there are at most 2^H combinations.
3. A bit indicating whether the potential site of cluster

C is activated.

Filling Out Dynamic Program Entries. The dynamic program entries are computed bottom up in the fashion described in [7, Section 4]. Observe that the information identifying a cluster contains the relevant configuration of random radii that can determine the cluster's parent and siblings. The following result can be derived from [7, Theorem 4.1] and compares the running time of the dynamic program for TSPN with that for TSP.

THEOREM 6.1. (COMPARING RUNNING TIMES) *With constant probability, the dynamic program gives an (m, r)-light tour for TSPN of length at most $(1+\epsilon)\mathsf{OPT}$ in time $\mathsf{TIME}(\mathsf{TSP}) \cdot 2^{O(HK)}$, where $\mathsf{TIME}(\mathsf{TSP})$ is the time for approximating TSP with dynamic program in Bartal et al. [6], H is an upper bound on the number of ambiguous regions for each cluster, and $K = O(s)^k$ is an upper bound on the number of children for each cluster.*

COROLLARY 6.2. (RUNNING TIME OF $\mathsf{DP}(V, W)$) *Fix any $c > 0$, and suppose an instance (S_1, W_1) with large enough $n = |W_1|$ is passed to DP in Section 3. Then, with high probability, the dynamic program DP can return a TSPN tour visiting all regions in W_1 with length at most $(1+\epsilon) \cdot \mathsf{OPT}(S_1, W_1)$ in time*

$$n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[\Delta^{2+c} \cdot O(\tfrac{k\alpha}{\epsilon})^{O(k)} \cdot o(\log^c n)].$$

Proof. Repeating the algorithm in Theorem 6.1 for $O(\log n)$ times, we can convert constant success probability to high probability $1 - \frac{1}{\mathrm{poly}(n)}$. We show our dynamic program runs in polynomial time in n, and give the dependence of the running time on the parameters.

Recall that the dynamic program for TSP [6] finds the optimal (m, r)-light tour in hierarchical decompositions where each cluster has at most K children. The number of clusters (induced by all relevant configurations of radii) from all heights is at most $n^{\frac{1}{c} \cdot O(1)^k}$, and the time to process all entries of a cluster is $(mKr)^{2Kr}$.

Recall that $K = O(s)^k$, $m := O(\frac{sk \log_s n}{\epsilon})^k = O(\frac{k \log^2 n}{\epsilon})^k$, and $r := \Delta \cdot O(\frac{k^2 \alpha}{\epsilon})^k \cdot s^k \cdot \log_s \log n$.

For any $c > 0$, for sufficiently large n, we can set $s := (\log n)^{\frac{c}{4k}} \geq 4$, and the term $s^{2k} = (\log n)^{\frac{c}{2}}$ can be used to absorb sub-logarithmic terms $O(\log \log n)$. Hence, $\ln(mKr)^{2Kr} = \Delta \cdot O(\frac{k\alpha}{\epsilon})^{O(k)} \cdot o(\log^c n)$.

Finally, $H := \frac{\Delta}{\gamma} \cdot O(\frac{k\alpha}{\epsilon})^{2k} \cdot s^k$ and $\frac{1}{\gamma} := \frac{\Delta\beta \cdot O(\alpha)^k}{\epsilon} \log \frac{\Delta\beta \cdot O(\alpha)^k}{\epsilon}$. Hence, $HK = \Delta^{2+c} \cdot O(\frac{k\alpha}{\epsilon})^{O(k)} \cdot o(\log^c n)$.

Therefore, the total running time is

$$n^{\frac{1}{c} \cdot O(1)^k} \cdot (mKr)^{2Kr} \cdot 2^{O(HK)} = n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[\Delta^{2+c} \cdot O(\tfrac{k\alpha}{\epsilon})^{O(k)} \cdot o(\log^c n)].$$

■

References

[1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011.

[2] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[3] S. Arora. Approximation algorithms for geometric TSP. In *The traveling salesman problem and its variations*, volume 12 of *Comb. Optim.*, pages 207–221. Kluwer Acad. Publ., Dordrecht, 2002.

[4] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[5] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th FOCS*, pages 184–193, 1996.

[6] Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: low-dimensionality implies a polynomial time approximation scheme. In *STOC*, pages 663–672, 2012.

[7] T.-H. Hubert Chan and Khaled M. Elbassioni. A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *Discrete & Computational Geometry*, 46(4):704–723, 2011.

[8] T-H. Hubert Chan and Shaofeng H.-C. Jiang. A PTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. *http://www.optimization-online.org/DB_HTML/2014/11/4649.html*.

[9] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.*, 22(1):63–93, 1999.

[10] M. de Berg, J. Gudmundsson, M.J. Katz, C. Levcopoulos, M.H. Overmars, and A. F. van der Stappen. TSP with Neighborhoods of varying size. *J. of Algorithms*, 57:22–36, 2005.

[11] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.

[12] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *J. Algorithms*, 48(1):135–159, 2003.

[13] K. Elbassioni, A.V. Fishkin, N. Mustafa, and R. Sitters. Approximation algorithms for Euclidean group TSP. In *ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1115–1126. Springer, 2005.

[14] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004.

[15] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000. (Preliminary version in *9th SODA*, pages 253–259, 1998).

[16] Lee-Ad Gottlieb. A light metric spanner. *CoRR*, abs/1505.03681, 2015.

[17] J. Gudmundsson and C. Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nordic J. Computing*, 6(4):469–488, 1999.

[18] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *44th FOCS*, pages 534–543, 2003.

[19] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 585–594. ACM Press, 2003.

[20] C. S. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems (extended abstract). In *Proc. 11th Annual ACM Symposium on Computational Geometry*, pages 360–369, 1995.

[21] J. Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.

[22] 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(4):1298–1309, 1999.

[23] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *SODA*, pages 11–18, 2007.

[24] Joseph S. B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Symposium on Computational Geometry*, pages 183–191, 2010.

[25] S. B. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *STOC '98 (Dallas, TX)*, pages 540–550. ACM, New York, 1999.

[26] Shmuel Safra and Oded Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. *Computational Complexity*, 14(4):281–307, 2006.

[27] K. Talwar. Bypassing the embedding: Algorithms for low-dimensional metrics. In *36th STOC*, pages 281–290, 2004.