A PTAS for TSP with Fat Weakly Disjoint 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. However, several non-trivial technical hurdles need to be overcome for applying the PTAS framework to TSPN.

(1) Heuristic to detect sparse instances. In the STOC 2012 paper, a minimum spanning tree heuristic is used to estimate the portion of an optimal tour within some ball. However, for TSPN, it is not known if an optimal tour would use points inside the ball to visit regions that intersect the ball.

(2) Partially cut regions in the recursion. After a sparse ball is identified by the heuristic, the PTAS framework for TSP uses dynamic program to solve the instance restricted to the sparse ball, and recursively solve the remaining instance. However, for TSPN, it is an important issue to decide whether each region partially intersecting the sparse ball should be solved in the sparse instance or considered in the remaining instance.

Surprisingly we show that both issues can be resolved by conservatively making the ball in question responsible for all intersecting regions. In particular, a sophisticated charging argument is needed to bound the cost of combining tours in the recursion.

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

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 [Mit99, Aro02, RS99]. For some time, only a quasi-polynomial[1] time approximation scheme (QPTAS) is known for doubling metrics [Tal04], where a metric space has doubling dimension [Ass83, Cla99, GKL03] 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. [BGK12] gave a PTAS for TSP on doubling metrics.

The neighborhood version of the problem was first introduced by Arkin and Hassin [AH94], who gave constant approximation 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 $\Omega(\log^{2-\epsilon} n)$ for any $\epsilon > 0$ by Halperin and Krauthgamer [HK03]. There is an almost matching upper bound of $O(\log N \log k \log n)$-approximation, using the results of Garg et al. [GKR00] and Fakcharoenphol et al. [FRT04], where N is the total number of points in V and k is the maximum number of points in each region.

TSPN on 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 if the regions are allowed to be intersecting connected subsets, the problem remains APX-hard [dBGK$^+$05, SS06].

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) $O(\log n)$ [MM95, GL99], (ii) constant ratio [Mit10, dBGK$^+$05, EFMS05], (iii) $(1 + \epsilon)$-ratio PTAS [DM03, Mit07].

TSPN on Doubling Metrics. Chan and Elbassioni [CE11] 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 in the first place. More motivation and examples for fat weakly disjoint regions are given in [CE11]. 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 [Aro02] and Talwar [Tal04], 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.

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

Main Result. We combine the techniques of the dynamic program for TSPN [CE11] and the PTAS for TSP [BGK12] to give a randomized algorithm that approximates TSPN.

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

Our running time for the special case $\Delta = \alpha = 1, c = \frac{1}{2}$ 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 [BGK12].

Technical Challenges. Our PTAS for TSPN uses the high level idea of the PTAS framework for TSP [BGK12], and in the core utilizes the dynamic program for TSPN in [CE11]. However, there are a number of technical hurdles, and we briefly outline how we overcome them.

(1) Heuristic to detect sparse instances. In [BGK12], a minimum spanning tree heuristic MST is computed on some subset B of points to estimate the weight of the portion within B of some nearly optimal tour T. We generalize this heuristic to consider a minimum spanning tree F of representatives picked arbitrarily for all regions W' intersecting B. However, the points in W' spanned by F might be different from the points visited by the portion of tour T in B, and tour T might choose to visit regions W' (that partially intersect with B) using points outside B. Surprisingly, we can relate tree F with tour T in Lemma 4.2, and show that our MST heuristic is not too pessimistic in Lemma 5.1.

(2) Resolving partially cut regions in sparse instances in the recursion. In [BGK12], loosely speaking, after a sparse instance is identified on a subset S_1 of points (by the MST heuristic), the sparse sub-instance on S_1 is solved with a dynamic program DP similar to [Aro02, Tal04] 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 5.3 to ensure that the recursion can be applied as in [BGK12].

(3) Bounding the number of ambiguous regions in sparse DP. In the dynamic program for TSPN in [CE11], 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 [CE11]. We improve the analysis (Corollary 6.1 and Lemma 6.2) such that H is independent of n. Hence, the dynamic programs in [CE11] and [BGK12] can be combined together to run in time polynomial in n.

2 Preliminaries

We consider a finite metric space $M = (V, d)$. (For basic properties of metric spaces, we refer the reader to standard texts [DL97, Mat02].) 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* [Ass83, GKL03] 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 $z \in V$.

Fact 2.1 (PACKING IN DOUBLING METRICS [GKL03]) *Suppose Z is a ρ-packing contained in some ball of radius $2^s\rho$ in a metric space with doubling dimension at most k. Then, $|Z| \leq 2^{(s+1)k}$.*

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 [CE11], 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) [CE11] *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 [CE11]) *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 $D > 0$,*

$\sum_{P \in W} \min\{\mathsf{Diam}(P), D\} \leq \Delta \cdot (8\alpha)^k \cdot \max\{2w(T), D\}$.

Remark 2.1 (Assumptions on the Partition $\{W_l\}_{l \in [\Delta]}$) *As in [CE11], we assume that the partition $\{W_l\}_{l \in [\Delta]}$ of regions and the parameter α are given to us such that each group is guaranteed to be α-fat weakly disjoint. 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 denote by $\mathsf{OPT}(S, W)$ be 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 which 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 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 $p'_j \in P_j$ 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 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 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 [BGK12], we consider some parameter $s = (\log n)^{\frac{c}{2k^2}} \geq 4$ (i.e., $n \geq 2^{2^{\Omega(k^2)}}$). Set $L := O(\log_s n) = O(\frac{k^2 \log n}{c \log\log n})$. 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 [BGK12], we use the randomized decomposition scheme defined in [Bar96, ABN11].

Definition 2.2 (Single-Scale Decomposition [ABN11]) *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 [ABN11] 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 level $L-1$, 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 [BGK12], 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 [BGK12, 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.

However, this notion alone cannot be directly applied to TSPN (see remark after Lemma 4.2), and in addition, we also make the distinction of different scales of portals as follows.

Portals. As in [Aro02, Tal04, BGK12], 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 [BGK12], the portals of a cluster need not be points of the cluster itself, but are just used as entry or exit points.

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. The difference here is that there are two scales of portals for each cluster.

- **Coarse Portals.** At height i, the *coarse portals* for a cluster C is the subset of net-points in $N_{i'}$ that cover C, where i' is the maximum index such that $s^{i'} \leq \max\{1, \frac{\epsilon}{8} \cdot s^i\}$. In the above transition (x, y), if $d(x, y) \geq s^i$, then the closest coarse portals p_x and p_y are used.
- **Fine Portals.** At height i, the *fine portals* for a cluster are defined similarly, except that i' is the maximum index such that $s^{i'} \leq \max\{1, \frac{\epsilon}{4\beta L} \cdot s^i\}$; observe that the fine portals include the coarse portals. In the above transition (x, y), if $d(x, y) < s^i$, then the closest fine portals p_x and p_y are used.

Lemma 2.2 (Portal-Resepcting 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*

Proof: It suffices to show that each transition (x, y) in a tour can be made portal-respecting. Suppose $s^j \leq d(x, y) < s^{j+1}$. Consider the case when (x, y) is first cut at height-i, which happens with probability $\min\{1, \frac{\beta}{s^i} \cdot d(x, y)\}$.

For $i \leq j$, this probability is at most 1, and the extra routing cost is at most $\frac{\epsilon}{4} \cdot s^i$; summing over $i \leq j$ gives $\frac{\epsilon}{2} \cdot d(x, y)$.

For $i > j$, the probability is at most $\frac{\beta}{s^i} \cdot d(x, y)$, in which case the extra routing cost is at most $O(\frac{\epsilon}{4\beta L} \cdot s^i)$; summing over $i > j$, the expected contribution is at most $\frac{\epsilon}{2} \cdot d(x, y)$.

Hence, the expected length of the portal-respecting segment from x to y is at most $(1 + \epsilon) \cdot d(x, y)$. ■

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

(m, r)**-Light Tour.** An (m, r)-*light tour* is a portal-respecting tour that visits each cluster only through its m portals (following rules for coarse/fine 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 [Tal04, CE11] 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 [BGK12] 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 [BGK12], and apply it to TSPN. Given a net-point $u \in N_i$ at height i and a set W of regions, we shall define a heuristic $\mathsf{MST}_W(u, i)$ that measures the sparsity around u at scale s^i with respect to the regions W.

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.

It uses a subroutine $\mathsf{DP}(S, W)$, which can be applied when the instance is "sparse"; it 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.

1. **Base Case.** If $|W| = n$ is smaller than some constant threshold, solve the problem by brute force, recalling that $|V| \leq O(\frac{n^2}{\epsilon})^k$.
2. **Sparse Instance.** If for all $i \in [L]$, for all $u \in N_i$, $\mathsf{MST}_W(u,i)$ is at most $q_0 \cdot s^i$ (where the reason for choosing $q_0 := (\frac{\Delta^2}{\epsilon})^k \cdot \Theta(\alpha^2 s)^{k^2}$ is given in Lemma 5.3), 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* $\mathsf{MST}_W(u,i) > q_0 \cdot s^i$.
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 set of points around u with diameter $O(s^i)$ 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 $\mathsf{MST}_W(u,i)$. We define the heuristic as follows. Let W' be the subset of regions in W that have non-empty intersection with $B := B(u, 3s^i)$. For each $P \in W'$, pick a representative by choosing a point in $P \cap B$ (for instance, one that is nearest to u); suppose R is the set of representatives of W'. Then, return the weight of the minimum spanning tree $\mathsf{MST}_u(R)$ for $R \cup \{u\}$.

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 Theorem 6.1.

Define decomposition procedure to remove critical instance. Suppose i is the smallest height such that there exists $\mathsf{MST}_W(u,i) > q_0 \cdot s^i$. Let B be a ball centered at u with radius sampled uniformly from $[3s^i, 4s^i]$. Define $S_1 := B(u, 5s^i)$, $W_1 := \{P \cap S_1 : P \cap B \neq \emptyset, P \in W\}$, 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 Corollary 4.1 and Lemma 5.3.

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 [BGK12] 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 $n = |W|$ times. 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 $\mathbb{E}[T_2] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \mathsf{OPT}^{nr}(V, W_2 + \{u\})$.

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 $\mathbb{E}[T_1 + T_2] \leq \frac{1+\epsilon}{1-\epsilon} \cdot \mathsf{OPT}^{nr}(V, W) = (1 + O(\epsilon)) \cdot \mathsf{OPT}(V, W)$, 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 [BGK12] 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 MST 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 [BGK12]. 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 modify the previous result in [BGK12, Lemma 3.1] as follows.

Lemma 4.1 (q-Sparsity Allows (m, r)-Lightness [BGK12]) *Suppose a tour T (not necessarily net-respecting) 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 $r := O(1)^k \cdot q \log_s \log n + O(\frac{k}{\epsilon})^k + O(\frac{s}{\epsilon})^k$.*

Proof: The only difference from [BGK12, Lemma 3.1] is that the given tour T might not be net-respecting, and we use both coarse and fine portals. The cost incurred to make the tour portal-respecting (following rules for using coarse/fine portals) is analyzed in Lemma 2.2.

To bound the number r times that the tour crosses a cluster at height i, we carefully observe where the net-respecting property is used in [BGK12, Lemma 3.1]. Precisely, it is used to argue that when an edge of length at least s^i crosses a cluster, then it must do so via net-points of scale around $\epsilon \cdot s^i$. However, this is guaranteed exactly by our rule of using coarse portals when the cross edge has length at least s^i. ∎

We next show that the heuristic $\mathsf{MST}_W(u, i)$ can be used to detect sparse tours visiting regions, which is analogous to [BGK12, Lemma 1.12(i)]. Given a tour T that visits all regions in W, a *designated* mapping $f : W \to V(T)$ maps each region to a point visited by T such that for each $P \in W$, $f(P)$ is a point in region P. The set $f(W)$ is known as the *designated* points of W with respect to f. Given a subset $B \subseteq V$, recall that $T|_B$ consists of path segments of T that are totally inside B. For each such path segment, we start from each end, and discard nodes until the first designated point is reached; we denote by $T|_B^f$ the union of these truncated path segments.

Lemma 4.2 (MST Heuristic Gives Sparse Optimal Tours) *Suppose T is an optimal (not necessarily net-respecting) or optimal net-respecting tour visiting all regions in W, and let $f : W \to V(T)$ be a designated mapping. Then, for each height i, and each $u \in N_i$, $w(T|^f_{B(u,3s^i)}) \leq \Delta \cdot O(\alpha)^k \cdot \max\{\mathsf{MST}_W(u, i), s^i\}$.*

In particular, if T is an optimal tour visiting all regions in W, then f is a surjection (without loss of generality), and hence $T|^f_{B(u,3s^i)} = T|_{B(u,3s^i)}$ in this case.

Remark. For optimal net-respecting T, Lemma 4.2 could only give a weaker notion of sparsity (in terms of truncated path segments $T|_B^f$ with respect to designating function f), which is not enough for Lemma 4.1. Hence, we apply Lemma 4.1 to an optimal tour, and use the idea of coarse/fine portals to replace the net-respecting property in the proof of Lemma 4.1. On the other hand, the weaker sparsity notion for net-respecting tours is enough for the proof of Lemma 5.3 in Section 5.

Proof: Observe that $T|^{f}_{B(u,3s^i)}$ consists of path segments. Let Z be the set of end-points of these path segments. Our goal is to construct a subgraph G that spans Z and covers all designated points covered by $T|^{f}_{B(u,3s^i)}$.

Observe that it is possible to replace $T|^{f}_{B(u,3s^i)}$ by routing on G with weight $2w(G)$ to form a tour to visit all regions in W; moreover, it is possible to convert each edge in G to be net-respecting with a multiplicative factor $1+O(\epsilon)$. The optimality of T implies that $w(T|^{f}_{B(u,3s^i)}) \leq 2(1+O(\epsilon)) \cdot w(G)$. We next construct G and analyze its weight.

Let X be the set of designated points in $T|^{f}_{B(u,3s^i)}$. For each $x \in X$, pick P_x to be any region in $f^{-1}(x)$ using x as its designated point. Let W_X be the collection of the regions P_x's over $x \in X$.

Recall the definition of $\mathsf{MST}_W(u,i)$. Suppose W' is the subset of regions in W that intersect $B := B(u, 3s^i)$. Observe that if a region is in W_X, then it will be included in W'. For each $P \in W'$, a representative is picked to form the set of representatives R. Then, $\mathsf{MST}_W(u,i)$ is the weight of the minimum spanning tree $\mathsf{MST}_u(R)$ spanning R and u.

The subgraph G is formed as follows.

1. Include $\mathsf{MST}_u(R)$, which has weight $\mathsf{MST}_W(u,i)$.
2. Each point $x \in X$ is connected to G via the representative of P_x in R with an edge of weight at most $\min\{\mathsf{Diam}(P_x), 6s^i\}$.

We next analyze the weight of G. Recall that there are Δ groups, each of which is α-fat weakly disjoint. Observing that the Euler tour on $\mathsf{MST}_u(R)$ visits all regions in W_X and has weight at most $2\mathsf{MST}_W(u,i)$, Corollary 2.1 gives $\sum_{x\in X} \min\{\mathsf{Diam}(P_x), 6s^i\} \leq \Delta \cdot (8\alpha)^k \cdot \max\{4\mathsf{MST}_W(u,i), 6s^i\}$.

Hence, it follows that the weight of G is at most $\Delta \cdot O(\alpha)^k \cdot \max\{\mathsf{MST}_W(u,i), s^i\}$. ■

The next corollary is analogous to [BGK12, Lemma 3.3(a)]; however, we simplify the analysis by disregarding decomposition at the critical height i.

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

Proof: Observe that for S_1, it is sufficient to consider net-points up to height $i-1$, where there are still at most $O(s)^k$ height-$(i-1)$ clusters covering S_1. Since we do not consider height-i clusters, the value of the MST_W heuristic at height i is irrelevant.

Recall that in Section 3, each region P in W_1 is restricted to S_1. Hence, for each height-j (where $j < i$) net-point v_j covering S_1, the heuristic $\mathsf{MST}_{W_1}(v_j, j)$ and $\mathsf{MST}_W(v_j, j)$ might be different because a region P might intersect $B(v_j, 3s^j)$ and $B(v_j, 3s^j) \cap S_1$ at different points.

We see how this affect the proof in Lemma 4.2. The change is that the subgraph G might involve points that are not in S_1. However, since we are using G for routing the tour, when a point not in S_1 is encountered, we can simply skip it and move on to the next point, without increasing the cost of the tour; observe that we need not ensure that the tour is net-respecting. Hence, the same proof gives the sparsity of an optimal tour in $(S_1, W_1 + \{u\})$. ■

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 in Section 6 to compute such a tour.

5 Identifying and Removing Critical Instances

We recall how a critical instance is removed in the description of $\mathsf{ALG}(V, W)$ defined in Section 3.

Removing Critical Instance. Recall that i is the smallest height such that there exists $u \in N_i$ with $\mathsf{MST}_W(u, i) > q_0 \cdot s^i$. Sample $h \in [0, s^i]$ uniformly at random (as opposed to using distribution Exp_i as in Definition 2.2); let $B := B(u, 3s^i + h)$.

We define $S_1 := B(u, 5s^i)$, $W_1 := \{P \cap S_1 : P \cap B \neq \emptyset, P \in W\}$, 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 (MST Gives a Lower Bound on Tour Length) *Suppose T_1 is a tour that visits all regions in W_1 and u, and $q_0 \geq 12 \cdot (8\alpha)^k$. Then, $q_0 \cdot s^i < \mathsf{MST}_W(u, i) \leq w(T_1) \cdot (1 + 2\Delta \cdot (8\alpha)^k)$, where Δ is the number of groups of regions, each of which is α-fat weakly disjoint.*

Proof: Recall that $\mathsf{MST}_W(u, i)$ is the weight of the minimum spanning tree $\mathsf{MST}_u(R)$ spanning u and R, where $R \subset B(u, 3s^i)$ is some set of representatives of regions W' in W that intersect $B(u, 3s^i)$.

We construct a subgraph spanning R and u by (i) including T_1 (which visits every region in W' and u), and (ii) connecting each representative in R for each region $P \in W'$ with the corresponding point in T_1 using an edge of weight at most $\min\{\mathsf{Diam}(P), 6s^i\}$.

The cost of (i) is $w(T_1)$. To bound the cost of (ii), we observe that T_1 visits all regions in W', use Corollary 2.1 to obtain $\sum_{P \in W'} \min\{\mathsf{Diam}(P), 6s^i\} \leq \Delta \cdot (8\alpha)^k \cdot \max\{2w(T_1), 6s^i\}$.

Hence, it follows that $q_0 \cdot s^i < \mathsf{MST}_W(u, i) = w(\mathsf{MST}_u(R)) \leq w(T_1) + \Delta \cdot (8\alpha)^k \cdot \max\{2w(T_1), 6s^i\}$.

We first show that $w(T_1) > 3s^i$. Otherwise, $q_0 \cdot s^i < w(T_1) + \Delta \cdot (8\alpha)^k \cdot 6s^i \leq w(T_1) + \frac{q_0}{2} \cdot s^i$, which implies that $w(T_1) > \frac{q_0}{2} \cdot s^i > 3s^i$, a contradiction.

Hence, we have $q_0 \cdot s^i < \mathsf{MST}_W(u, i) \leq w(T_1) \cdot (1 + 2\Delta \cdot (8\alpha)^k)$, as required. ■

Lemma 5.2 *There exists a tour that visits all regions that intersect $B(u, 4s^i)$ with length at most $O(s)^k \cdot q_0 \cdot s^i$.*

Proof: Let W' be the set of regions that intersect $B(u, 4s^i)$, and let $\widehat{N}_{i-1} \subset N_{i-1}$ be the subset of height-$(i-1)$ net-points that cover $B(u, 4s^i)$. Observe that $|\widehat{N}_{i-1}| \leq O(s)^k$, and each region in W' must be covered by the minimum spanning tree T_v defining $\mathsf{MST}_W(v, i-1)$ for some $v \in \widehat{N}_{i-1}$. Observe that since $\mathsf{MST}_W(u, i)$ is critical, $\mathsf{MST}_W(v, i-1) \leq q_0 \cdot s^{i-1}$.

We next construct a subgraph that intersects all regions in W'. First, we include all the minimum spanning trees T_v over $v \in \widehat{N}_{i-1}$; this has cost $O(s)^k \cdot q_0 \cdot s^{i-1}$. Second, for each $v \in N_{i-1}$, pick any point in T_v and connect it to u; this has cost $O(s)^k \cdot s^i$. Hence, there exists a tour with length at most $O(s)^k \cdot q_0 \cdot s^i$ visiting all regions in W', as required. ■

The following result is the formal version of (MP2) in Section 3; it is an analogue of [BGK12, Lemma 3.3], and turns out to be the most technical part to adapt the argument for TSPN.

Lemma 5.3 (Removing Critical Instance) *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 $q_0 := (\frac{\Delta^2}{\epsilon})^k \cdot \Theta(\alpha^2 s)^{k^2}$. Then, for each random $h \in [0, s^i]$, 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 $h \in [0, s^i]$.*

Proof: Let $B := B(u, 3s^i + h)$, where $h \in [0, s^i]$ is drawn uniformly at random.

Since T visits all regions in W, we choose a designated mapping $f : W \to V(T)$ such that for all regions P that are visited by $T|_B$, $f(P)$ is covered by $T|_B$.

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. Let Z^{in} be the end-points of edges E^{cr} that are inside B, and Z^{out} be those that are outside B.

Let $\delta := \frac{\epsilon}{\Delta^2 \beta \cdot O(\alpha^2 s)^k}$, and l be the largest height such that $s^l \leq \max\{1, \delta s^i\}$.

Let $\widehat{N}_l$ be the subset of net-points N_l that cover S_1. Observe that $|\widehat{N}_l| \leq O(\frac{s}{\delta})^k$, and let F_l be a minimum spanning tree of $\widehat{N}_l$, which has weight at most $O(\frac{s}{\delta})^k \cdot s^i$.

Patching Construction. We construct T_1 that visits all designated points in $T|_B$, as well as regions intersecting B and the point $u \in \widehat{N}_l$; we also construct net-respecting T_2 that visits all points covered by E^{out} and the point u. Hence, T_1 visits all regions in $W_1 + \{u\}$, and T_2 visits all regions in $W_2 + \{u\}$. Ideally, T_1 should use edges in E^{in}, and T_2 should use edges in E^{out}. Then, the cross points are patched up to complete the tours, where the cost is charged to E^{cr} and the sparsity of the space in $B(u, O(s^i))$. However, the situation for the cross edges are actually more complicated, and we need a sophisticated procedure to enable the charging argument.

In the below procedure, special attention must be paid such that the weight of each edge $e \in T$ is charged in exactly one of the cases: (i) e appears in T_1, (ii) e appears in T_2, (iii) if there is a path segment in T from x to y of length at least $5\delta s^i$ such that none of the edges in the segment appears in T_1 or T_2, then the **exact** weight (with no increase in $(1+\epsilon)$ factor) of these edges can be used to pay for connecting x and y to the corresponding nearest net-points in $\widehat{N}_l$ (and subsequent routing). We use G to denote other edges not in T that are used for patching, and they can be used a constant number of times for patching cross points as follows; some of them are charged directly to edges in T that do not appear in T_1 or T_2, while others are charged to the tree F_l and the sparsity of $B(u, O(s^i))$ via Lemma 4.2.

1. The edges in F_l are used to patch both T_1 and T_2.
2. We consider cross edges E^{cr}. Suppose $(x, y) \in E^{cr}$, where $x \in Z^{in}$ and $y \in Z^{out}$. We start at x, and travel along the path segment T_s in $T|_B$ inside B until the first designated point $v_P = f(P)$ (for some region P) is reached; if no such designated point v_P is found, the whole path segment T_s (together with the cross edges) is not necessary to visit regions in W_1, and will be used in T_2. A similar designated point $v_{P'}$ is found from the cross edge at the other end of the path segment T_s; it is possible that $v_P = v_{P'}$. The path segment between v_P and $v_{P'}$ belongs to T_1; we next describe how patching is performed at v_P (and similarly at $v_{P'}$).
   We start at v_P and travel along T towards cross point x. We perform case analysis on when the next designated point is encountered.
   (i) Suppose no designated point is encountered after traveling a distance within $5\delta s^i$. (We can assume that when the tour T leaves B, it must travel a distance at least $10\delta s^i$ outside B before re-entering; otherwise, the short segment outside B must be contained in $B(u, 5s^i)$, and we concatenate it with the preceding and succeeding segments in $T|_B$ together as a single path segment.)
   Let a and b be adjacent points on T such that $d_T(v_P, a) < 5\delta s^i \leq d_T(v_P, b)$. If $d(a, b) > \frac{\delta}{\epsilon} \cdot s^i$, then a must be a height-l net-point in $\widehat{N}_l$ (already connected to F_l), because T is net-respecting; in this case, point a is used in both T_1 and T_2, where the portion from v_P to a belongs to T_1 and the portion after a towards b belongs to T_2. If $d(a, b) \leq \frac{\delta}{\epsilon} \cdot s^i$, then

$b \in B(u, 5s^i)$ and the exact weight of the portion of T between v_P and b is enough to connect v_P and b to the corresponding closest net-points in $\widehat{N}_l$, and also allows these connections to be used once more during tour patching.

(ii) Suppose another designated point v_Q is found on T such that $d_T(v_P, v_Q) \leq 5\delta s^i$. In this case, observe that v_Q is outside B; v_P belongs to T_1 and v_Q belongs to T_2. The designated points v_P and v_Q are connected to their corresponding closest net-points in $\widehat{N}_l$. However, since $d_T(v_P, v_Q)$ is small, we cannot charge the patching cost $O(\delta) \cdot s^i$ directly to $d_T(v_P, v_Q)$, but instead will charge this cost to the sparsity of $B(u, O(s^i))$ via Lemma 4.2. When such charging occurs for the pair $\{v_P, v_Q\}$ of designated points, we say that the path segment $T(v_P, v_Q)$ is *activated*; note that the probability that v_P and v_Q are separated by B is at most $\frac{d_T(v_P, v_Q)}{s^i}$, which also gives an upper bound on the probability that $T(v_P, v_Q)$ is activated.

3. For each region P in W_1 that intersects with B (that is not already covered by the designated points in $T|_B$), pick the point in P closest to u and connect it with the closest point in $\widehat{N}_l$.

Observe that each edge $e \in T$ is already net-respecting, and we emphasize that its weight $w(e)$ is used exactly once with no extra $(1+\epsilon)$ factor. On the other hand, each edge added in G can be used a constant number of times, and may be replaced by a net-respecting path (in the case for patching T_2) whose length is a factor at most $1 + O(\epsilon)$ of the original edge length.

Patching Cost. We next analyze the expected cost of adding edges in G (that are not charged directly to T), where the randomness is over $h \in [0, s^i]$.

We start from bounding the expected cost due to a region being cut by B. Let W' be the set of regions in W that intersect with $B(u, 4s^i)$; observe that $W_1 \subseteq W'$. Each region $P \in W'$ is cut by the random ball B with probability at most $\min\{1, \frac{\mathsf{Diam}(P)}{s^i}\}$, in which case the cost to connect a point in P to the nearest net-point in $\widehat{N}_l$ is at most $\delta \cdot s^i$. Hence, each region $P \in W'$ has expected contribution of at most $O(\delta) \cdot \min\{\mathsf{Diam}(P), s^i\}$.

By Lemma 5.2, there exists a tour $T_{W'}$ that visits all regions in W' and has weight at most $O(s)^k \cdot q_0 \cdot s^i$. Hence, Corollary 2.1 gives $\sum_{P \in W'} \min\{\mathsf{Diam}(P), s^i\} \leq \Delta \cdot (8\alpha)^k \cdot 2 \cdot O(s)^k \cdot q_0 \cdot s^i$.

(1) Hence, we conclude that the expected contribution from the regions in W' is at most

$\delta \cdot \Delta \cdot O(\alpha s)^k \cdot q_0 \cdot s^i \leq \frac{\epsilon}{3} \cdot w(T_1)$, by the choice of δ and Lemma 5.1.

(2) The edges in F_l are deterministic, and the cost is $O(\frac{s}{\delta})^k \cdot s^i \leq \delta \cdot \Delta \cdot O(\alpha s)^k \cdot q_0 \cdot s^i$, since $q_0 \geq (\frac{\Delta^2}{\epsilon})^k \cdot \Omega(\alpha^2 s)^{k^2}$. Hence, the contribution is also at most $\frac{\epsilon}{3} \cdot w(T_1)$, as in (1).

(3) We next bound the expected cost associated with activated path segments. Define $\mathcal{T}$ to be the collection of path segments $T(v_P, v_Q)$ between successive designated points in T such that $d_T(v_P, v_Q) \leq 5\delta s^i$ and at least one of the designated points is in $B(u, 4s^i)$. Observe that the probability that the path segment $T(v_P, v_Q)$ is activated is at most $\frac{d_T(v_P, v_Q)}{s^i}$, in which case the patching cost is $O(\delta) \cdot s^i$. Hence, the total expected cost due to activated path segments is $O(\delta) \cdot \sum_{T(v_P, v_Q) \in \mathcal{T}} d_T(v_P, v_Q)$.

To analyze the sum of the weights of the potential path segments, it suffices to observe that the union of those path segments is in $T|^f_{B(u, 5s^i)}$. However, since $\mathsf{MST}_W(u, i)$ is critical, we will cover those path segments instead by the union of $T|^f_{B(v, 3s^{i-1})}$ over $v \in \widehat{N}_{i-1}$.

Since T is an optimal net-respecting tour visiting all regions in W, it follows from Lemma 4.2 that $w(T|^f_{B(v, 3s^{i-1})}) \leq \Delta \cdot O(\alpha)^k \cdot q_0 \cdot s^{i-1}$.

Observing that $|\widehat{N}_{i-1}| \leq O(s)^k$, we conclude that the expected cost associated with activated path segments is at most $\delta \cdot \Delta \cdot O(\alpha s)^k \cdot q_0 \cdot s^i \leq \frac{\epsilon}{3} \cdot w(T_1)$, as in (1).

Therefore, combining cases (1) to (3), we have $\mathbb{E}[w(T_1) + w(T_2)] \leq w(T) + \epsilon \cdot w(T_1)$. Taking expectation and rearranging, we have $\mathbb{E}[w(T_1)] \leq \frac{1}{1-\epsilon} \cdot (w(T) - \mathbb{E}[w(T_2)])$, as required. ■

6 Dynamic Program for TSPN

In Section 4, we see that the MST_W heuristic 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 [CE11] and [BGK12], which are themselves extensions of the ones in [Aro02, Tal04]. We first review some properties for TSPN as in [CE11].

6.1 Structural Properties of TSPN

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 [CE11] 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 [CE11, 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.

We rephrase the following result from [CE11, Lemma 3.5] to give an upper bound on the number of ambiguous regions a cluster needs to consider; observe that the improvement due to Lemma 6.1 implies that the number of ambiguous regions for each cluster is independent of n.

Lemma 6.2 (Number of Ambiguous Regions [CE11]) *The number of ambiguous regions for a cluster is at most $H := \Delta \cdot O(\frac{\alpha}{\gamma})^k$.*

6.2 Description of Dynamic Program for TSPN

Our dynamic program DP is a combination of the dynamic programs in [CE11] and [BGK12]. In [BGK12], 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 $|N_i| \leq O(n^k)$.
2. For each $j \geq i$, and $v \in N_j$ such that $d(u, v) \leq O(s^j)$, 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 [BGK12], 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 $s = (\log n)^{\frac{c}{2k^2}}$, 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 $O(1)^L = n^{O(\frac{k^2}{c \log \log n})}$ combinations.

Therefore, the whole dynamic program considers at most $n^{\frac{1}{c} \cdot O(1)^k}$ clusters. As in [Aro02, Tal04], 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 5.3 as follows:

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

As in the case [CE11], 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 [CE11], 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 $|I| \leq r$, 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 $j \geq i$. 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 $H := \Delta \cdot O(\frac{\alpha}{\gamma})^k$, 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 [CE11, 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 [CE11, 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* $(1+\epsilon)$OPT *in time* TIME(TSP)$\cdot 2^{O(HK)}$*, where* TIME(TSP) *is the time for approximating* TSP *with dynamic program in Bartal et al. [BGK12], 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 $n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[(\frac{\Delta}{\epsilon})^{O(k)} \cdot O(\alpha)^{2k^2} \cdot o(\log^c n)]$.*

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 [BGK12] 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

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

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

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

Therefore, the total running time is

$n^{\frac{1}{c} \cdot O(1)^k} \cdot (mKR)^{2KR} \cdot 2^{O(HK)} = n^{\frac{1}{c} \cdot O(1)^k} \cdot \exp[(\frac{\Delta}{\epsilon})^{O(k)} \cdot O(\alpha)^{2k^2} \cdot o(\log^c n)]$. ∎

References

[ABN11] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011.

[AH94] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[Aro02] 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.

[Ass83] P. Assouad. Plongements lipschitziens dans $\mathbf{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[Bar96] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th FOCS*, pages 184–193, 1996.

[BGK12] 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.

[CE11] 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.

[Cla99] K. L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete Comput. Geom.*, 22(1):63–93, 1999.

[dBGK+05] 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.

[DL97] M. M. Deza and M. Laurent. *Geometry of cuts and metrics*, volume 15 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1997.

[DM03] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *J. Algorithms*, 48(1):135–159, 2003.

[EFMS05] 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.

[FRT04] 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.

[GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low–distortion embeddings. In *44th FOCS*, pages 534–543, 2003.

[GKR00] 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).

[GL99] J. Gudmundsson and C. Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nordic J. Computing*, 6(4):469–488, 1999.

[HK03] 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.

[Mat02] J. Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.

[Mit99] 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.

[Mit07] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *SODA*, pages 11–18, 2007.

[Mit10] 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.

[MM95] 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.

[RS99] 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.

[SS06] Shmuel Safra and Oded Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. *Computational Complexity*, 14(4):281–307, 2006.

[Tal04] K. Talwar. Bypassing the embedding: Algorithms for low-dimensional metrics. In *36th STOC*, pages 281–290, 2004.