Faster Approximation Scheme for Euclidean k-TSP

Ernest van Wijland
École Normale Supérieure Paris, France

Hang Zhou
École Polytechnique, Institut Polytechnique de Paris, France

Abstract

In the Euclidean k-traveling salesman problem (k-TSP), we are given n points in the d-dimensional Euclidean space, for some fixed constant $d \geq 2$, and a positive integer k. The goal is to find a shortest tour visiting at least k points.

We give an approximation scheme for the Euclidean k-TSP in time $n \cdot 2^{O(1/\varepsilon^{d-1})} \cdot (\log n)^{2d^2 \cdot 2^d}$. This improves Arora's approximation scheme of running time $n \cdot k \cdot (\log n)^{\left(O\left(\sqrt{d}/\varepsilon\right)\right)^{d-1}}$ [J. ACM 1998]. Our algorithm is Gap-ETH tight and can be derandomized by increasing the running time by a factor $O(n^d)$.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization

Keywords and phrases approximation algorithms, optimization, traveling salesman problem

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.81

Funding This research work is supported by the Hi! PARIS Center.

Acknowledgements We thank Tobias Mömke, Noé Weeks, and Antoine Stark for discussions.

1 Introduction

In the Euclidean k-traveling salesman problem (k-TSP), we are given n points in $\mathbb{R}^d$, for some fixed constant $d \geq 2$, and a positive integer $k \leq n$. The goal is to find a shortest tour visiting at least k points out of the n points.

The Euclidean k-TSP is NP-hard [4], so researchers turned to approximation algorithms, e.g., [1, 2, 8, 9].[1] The best-to-date approximation for the Euclidean k-TSP is due to the approximation scheme[2] of Arora [1], which is among the most prominent results in combinatorial optimization. The randomized version of Arora's approximation scheme has a running time of

$$n \cdot k \cdot (\log n)^{\left(O\left(\sqrt{d}/\varepsilon\right)\right)^{d-1}}.$$

In this work, we give a faster approximation scheme for the Euclidean k-TSP; see Theorem 1. Compared with Arora [1], our running time sheds the factor k and, in addition, achieves an asymptotically optimal dependence on ε.

▶ **Theorem 1.** *Let $d \geq 2$ be a fixed constant. For any $\varepsilon > 0$, there is a randomized $(1+\varepsilon)$-approximation algorithm for the Euclidean k-TSP that runs in time*

$$n \cdot 2^{O(1/\varepsilon^{d-1})} \cdot (\log n)^{2d^2 \cdot 2^d}.$$

The dependence on ε in the running time is asymptotically optimal under the Gap-Exponential Time Hypothesis (Gap-ETH). *The algorithm can be derandomized by increasing the running time by a factor $O(n^d)$.*

[1] k-TSP has also been referred as "quota TSP" in the literature.
[2] An *approximation scheme* is a $(1+\varepsilon)$-approximation algorithm for any $\varepsilon > 0$.

In the rest of the section, we outline the proof of Theorem 1.

First, in a preprocessing step, the instance is partitioned into *well-rounded* subinstances. The partition algorithm in [1] takes $O(n \cdot k \cdot \log n)$ time. We improve the running time of the partition algorithm to $O(n)$; see the partition theorem (Theorem 3). To that end, we use a result on the *enclosing circles* due to Har-Peled and Raiche [5]. See Section 3.

Next, each subinstance is solved independently using a dynamic program based on the *quadtree* [1]. The dynamic program in [1] takes $n \cdot k \cdot (\log n)^{(O(\sqrt{d}/\varepsilon))^{d-1}}$ time. We improve the running time of the dynamic program to $n \cdot 2^{O(1/\varepsilon^{d-1})} \cdot (\log n)^{2d^2 \cdot 2^d}$; see the dynamic programming theorem (Theorem 7). In order to improve the dependence in ε in the running time in [1], we exploit a structure theorem (Theorem 11) that is a corollary of the approaches of Arora [1] and Kisfaludi-Bak, Nederlof, and Węgrzycki [6]; see Section 4.1. In order to remove the factor k in the running time in [1], we discretize the possible lengths of a tour into values called *budgets*; see Section 4.2. This is inspired by Kolliopoulos and Rao in the context of k-*median* [7]. The combination of the structure theorem and the budgets is non-trivial and is the key to the improved running time of the dynamic program, see Sections 4.3 and 4.4.

The overall running time and the derandomization in Theorem 1 follow from the partition theorem (Theorem 3) and the dynamic programming theorem (Theorem 7). The Gap-ETH tightness in Theorem 1 is a corollary of the Gap-ETH lower bound for the Euclidean TSP [6, Theorem I.1].

2 Notations

Let P denote a set of n points in $\mathbb{N}^d$ for some fixed constant $d \geq 2$. Let k be an integer in $[1, n]$. A path π in $\mathbb{N}^d$ is a k-*salesman tour* if π is a closed path visiting at least k points from P. Let $w(\pi)$ denote the *length* of π. In the *Euclidean k-traveling salesman problem (k-TSP)*, we look for a k-salesman tour π that minimizes $w(\pi)$. Let opt denote the minimum length of a k-salesman tour. For notational convenience, let ω denote 2^d.

▶ **Definition 2** (well-rounded instance, [1, Section 3.2]). *Consider an instance for the Euclidean k-TSP. Let $L \in \mathbb{N}$ denote the side length of the bounding box for the instance. We say that the instance is* well-rounded *if $L = O(k^2)$, all points in the instance have integral coordinates in $\{0, \ldots, L\}^d$, and the minimum nonzero internode distance is at least 8.*

3 Partitioning Into Subinstances

▶ **Theorem 3** (partition theorem). *Let $\mathcal{I}$ be an instance for the Euclidean k-TSP. There is a randomized algorithm that computes in expected $O(n)$ time a partition of $\mathcal{I}$ into a family of well-rounded subinstances $\mathcal{I}_1, \ldots, \mathcal{I}_\ell$ for some $\ell \geq 1$, such that with probability at least $1 - 2/\log k$, an optimal solution to $\mathcal{I}$ is completely within $\mathcal{I}_j$ for some $j \in [1, \ell]$. The algorithm can be derandomized by increasing the running time by a factor n.*

In the rest of the section, we prove Theorem 3.

Recall that, in the approach of Arora [1], an important step is to compute a good approximation for the optimal cost. Fact 4 relates the optimal cost with the side length of the smallest d-dimensional hypercube containing k points.

▶ **Fact 4** ([1, Section 3.2]). *The cost of the optimal solution to the Euclidean k-TSP is at most $dk^{1-(1/d)}$ times larger than the side length of the smallest d-dimensional hypercube containing k points.*

Estimating the smallest d-dimensional hypercube containing k points takes $O(nk \log n)$ time in [1]. We improve this running time to $O(n)$ in Lemma 6. This is achieved using a result of Har-Peled and Raiche [5] on the estimation of the smallest d-dimensional *ball* (Lemma 5).

▶ **Lemma 5** ([5, Corollary 4.18]). *Let $\lambda > 0$. For a set of n points in $\mathbb{R}^d$ and a positive integer k, one can $(1+\lambda)$-approximate, in expected $O(n/\lambda^d)$ time, the radius of the smallest d-dimensional ball containing k points.*

▶ **Lemma 6.** *For a set of n points in $\mathbb{R}^d$ and a positive integer k, one can $2\sqrt{d}$-approximate, in expected $O(n)$ time, the side length of the smallest d-dimensional hypercube containing k points.*

Proof. Let R^* denote the radius of the smallest d-dimensional ball containing k points. Applying Lemma 5 with $\lambda = 1$, one can compute in expected $O(n)$ time an estimate R such that $R^* \le R \le 2R^*$. On the one hand, there exists a d-dimensional hypercube of side length $2R^* \le 2R$ that contains at least k points. On the other hand, since the longest diagonal of a d-dimensional hypercube is equal to $\sqrt{d}$ times the side length of that hypercube, any d-dimensional hypercube of side length $2R^*/\sqrt{d} \ge R/\sqrt{d}$ contains at most k points. So the side length of the smallest d-dimensional hypercube containing k points is in $[R/\sqrt{d}, 2R]$. The claim follows. ◀

Proof of the partition theorem (Theorem 3). From Fact 4 and Lemma 6, there is a randomized algorithm that computes in expected $O(n)$ time an estimate A for the cost of the optimal solution such that

$$\text{opt} \le A \le 2d^{3/2}k^{1-(1/d)} \cdot \text{opt}.$$

The first part of the claim follows from arguments that are almost identical to [1], except by modifying the definition of the parameter ρ to

$$\rho := \frac{A\varepsilon}{16d^{3/2}k^{2-(1/d)}}.$$

Now we prove the second part of the claim. We only need to derandomize the algorithm in Lemma 6. This requires derandomizing the algorithm in Lemma 5, which is called `ndpAlg` in [5, Figure 3.1]. To that end, we remove Line 1 of the algorithm `ndpAlg` in [5], which randomly picks a point p from the set W_{i-1}. Instead, we enumerate *all* points p from W_{i-1}. For each such point p, we compute a set W_i^p using Lines 2–7 of the algorithm `ndpAlg` in [5]. Finally, we let W_i be the set W_i^p with minimum cardinality for all points $p \in W_{i-1}$. This completes the description of the derandomized algorithm, see Algorithm 1.

Algorithm 1 Derandomization for `ndpAlg`. $W \subseteq \mathbb{R}^d$ denotes a set of n input points.

```
W_0 ← W
i ← 1
while W_{i-1} ≠ ∅ do
    forall p ∈ W_{i-1} do
        Compute W_i^p                    ▷ Lines 2-7 of algorithm ndpAlg in [5]
    Let W_i be the set W_i^p with minimum cardinality for all p ∈ W_{i-1}
    i ← i + 1
```

It remains to show that the running time of the derandomized algorithm (Algorithm 1) is $O(n^2)$. Let z denote the number of iterations in the **while** loop in Algorithm 1. Consider any integer $i \in [1, z]$. From the analysis in [5, Lemma 3.12], for each $p \in W_{i-1}$, the set W_i^p in Algorithm 1 can be computed in $O(|W_{i-1}|)$ time. So the overall time to compute W_i^p over all $p \in W_{i-1}$ is $O(|W_{i-1}|^2)$. Using an analysis similar to [5, Lemma 3.12], there exists some $p \in W_{i-1}$ such that $|W_i^p| \leq (15/16)|W_{i-1}|$, thus $|W_i| \leq (15/16)|W_{i-1}|$ by the definition of W_i. Therefore, the overall running time of Algorithm 1 is

$$\sum_{i=1}^{z} O(|W_{i-1}|^2) \leq \sum_{i=1}^{z} (15/16)^{i-1} \cdot O(|W_0|^2) = O(|W_0|^2) = O(n^2).$$

This completes the proof of the second part of the claim. ◀

4 Dynamic Programming

▶ **Theorem 7** (dynamic programming theorem). *Consider a well-rounded instance for the Euclidean k-TSP. There is a randomized algorithm with running time $n \cdot 2^{O(1/\varepsilon^{d-1})} \cdot (\log n)^{2d^2 \cdot 2^d}$ such that, with probability at least $1/2$, the algorithm outputs a $(1+\varepsilon)$-approximate solution. The algorithm can be derandomized by increasing the running time by a factor $O(n^d)$.*

In the rest of the section, we prove Theorem 7.

4.1 Preliminaries: Notations, Quadtree, and Structure Properties

Let L denote the side length of the bounding box of the instance. Since the instance is well-rounded (Definition 2), we may assume that $P \subseteq \{0, \ldots, L\}^d$ and $L = O(k^2)$.

We review the *quadtree* [1] as well as its structural properties established by Arora [1] and Kisfaludi-Bak, Nederlof, and Węgrzycki [6].

We follow the notations in [6]. We pick $a_1, \ldots, a_d \in \{1, \ldots, L\}$ independently and uniformly at random and define $\boldsymbol{a} := (a_1, \ldots, a_d) \in \{0, \ldots, L\}^d$. Consider the hypercube

$$C(\boldsymbol{a}) := \bigtimes_{i=1}^{d} [-a_i + 1/2, 2L - a_i + 1/2].$$

Note that $C(\boldsymbol{a})$ has side length $2L$ and each point from P is contained in $C(\boldsymbol{a})$.

We define the *dissection* of $C(\boldsymbol{a})$ to be a tree constructed recursively, where each vertex is associated with a hypercube in $\mathbb{R}^d$. The root of the tree is associated with $C(\boldsymbol{a})$. Each non-leaf vertex of the tree that is associated with a hypercube $\bigtimes_{i=1}^{d} [l_i, u_i]$ has ω children with which we associate hypercubes $\bigtimes_{i=1}^{d} I_i$, where I_i is either $[l_i, (l_i + u_i)/2]$ or $[(l_i + u_i)/2, u_i]$. Each leaf vertex of the tree is associated with a hypercube of unit length.

A *quadtree* is defined similarly as the dissection of $C(\boldsymbol{a})$, except we stop the recursive partitioning as soon as the associated hypercube of a vertex contains at most one point from P. Each hypercube associated with a vertex in the quadtree is called a *cell* in the quadtree.

For each cell C in the quadtree, let ∂C denote the union of all facets of C.

▶ **Definition 8** (grid, [6, Definition II.4]). *Let F be a $(d-1)$-dimensional hypercube. Let t be a positive integer. We define* grid$(F, t) \subseteq \mathbb{R}^{d-1}$ *to be an orthogonal lattice of t points in F. Thus, if the hypercube has side length l, the minimum distance between any pair of points of* grid(F, t) *is $l/t^{1/(d-1)}$.*

▶ **Definition 9** (fine multiset, adaptation from [6, Section 5.1 in the full version]). *Let m and r be positive integers. Let C be a cell in the quadtree. Let B be a multiset of points in ∂C. For each facet F of C, let b_F denote the number of points in B that are in F. We say that B is* (m, r)-fine *if, for all facets F of C, either one of the two following cases holds:*

1. *$b_F \leq 1$ and $B \cap F \subseteq \mathrm{grid}(F, m)$;*
2. *$b_F \geq 2$ and $B \cap F \subseteq \mathrm{grid}(F, g(b_F))$, where $g(\cdot)$ is an integer-valued function such that $g(b_F) \leq r^{2d-2}/b_F$. Moreover, each point in $\mathrm{grid}(F, g(b_F))$ occurs at most twice in $B \cap F$.*

The parameters (m, r) are omitted when clear from the context.

▶ **Definition 10** ((m, r)-simple paths, adaptation from [1, Definition 1] and [6, Definition III.2]). *Let m and r be positive integers. A collection $\mathcal{Q}$ of paths in $\mathbb{R}^d$ is* (m, r)-simple *if, for every cell C, the intersection between $\mathcal{Q}$ and ∂C is (m, r)-fine.*

▶ **Theorem 11** (structure theorem, corollary of [1] and [6]). *Let $\boldsymbol{a}$ be a random vector in $\{1, \dots, L\}^d$. Let $m = (O((\sqrt{d}/\varepsilon) \log L))^{d-1}$ and $r = O(d^2/\varepsilon)$. With probability at least $1/2$, there is a k-salesman tour π such that both of the following properties hold:*

- *the path collection $\{\pi\}$ is (m, r)-simple;*
- *$w(\pi) \leq (1 + \varepsilon) \cdot \mathrm{opt}$.*

Proof. From Theorem 5 and Theorem 10 in [1], for some $m = (O((\sqrt{d}/\varepsilon) \log L))^{d-1}$, the quadtree defined by $\boldsymbol{a}$ has an associated k-salesman tour π_0 such that[3]

$$\mathbb{E}[w(\pi_0)] \leq (1 + \varepsilon/6) \cdot \mathrm{opt}, \tag{1}$$

and π_0 crosses each facet F of each cell of the quadtree only at points from $\mathrm{grid}(F, m)$.

We then apply Theorem III.3 from [6] on π_0 for some parameter $r \in \mathbb{R}$. This results in an (m, r)-simple tour π visiting the same set of points as π_0, such that

$$\mathbb{E}[w(\pi)] \leq (1 + O(d^2/r)) \cdot w(\pi_0) \leq (1 + \varepsilon/6) \cdot w(\pi_0), \tag{2}$$

where the second inequality holds for some $r = O(d^2/\varepsilon)$ that is well-chosen.

If π crosses a facet F of a cell of the quadtree only once, letting q denote that crossing, then q belongs to π_0. Since π_0 crosses each facet F only at points from $\mathrm{grid}(F, m)$, the above crossing q belongs to $\mathrm{grid}(F, m)$.

From (1) and (2), we have

$$\mathbb{E}[w(\pi)] \leq (1 + \varepsilon/6)^2 \cdot \mathrm{opt} < (1 + \varepsilon/2) \cdot \mathrm{opt}.$$

Markov's inequality implies that, with probability at least $1/2$, we have $w(\pi_0) \leq (1 + \varepsilon) \cdot \mathrm{opt}$. This completes the proof of the claim. ◀

4.2 Budget Multipath Problem

▶ **Definition 12** (budgets). *Let $\Phi = dk^{1-1/d}L$. We say that $s \in \mathbb{R}$ is a* budget *if either*

- *$s = 0$; or*
- *$s \in [1/(r^2 + m^{1/(d-1)}), (1 + \varepsilon)^2 \cdot \Phi]$ and there exists $i \in \mathbb{N}$ such that*

$$(1 + \varepsilon/(2d \cdot r^{d-1} + 3 \log_2 n))^i = s.$$

Let $\mathcal{S}$ be the set of all budgets.

---

[3] The bound in expectation is obtained in the proofs in [1].

▶ **Definition 13** (budget multipath problem). *We are given*
- *a cell C in the quadtree,*
- *a fine multiset $B \subseteq \partial C$,*
- *a perfect matching M on B,*
- *a budget $s \in \mathcal{S}$.*

We look for a collection $\mathcal{Q}$ of paths in C satisfying all of the following properties:
- *$\mathcal{Q}$ is (m, r)-simple;*
- *$\mathcal{Q}$ has total length at most s;*
- *the intersection between $\mathcal{Q}$ and ∂C is B;*
- *there is a one-to-one correspondence between the paths in $\mathcal{Q}$ and the edges in M, where we say that a path $q \in \mathcal{Q}$* corresponds *to an edge $(u, v) \in M$ if and only if u and v are the two endpoints of q.*

The goal is to maximize the number of points in P visited by $\mathcal{Q}$.

The Euclidean k-TSP can be reduced to the budget multipath problem. To see the reduction, consider the budget multipath problem for the root cell C_0 in the quadtree, the multiset $B := \emptyset$, the set $M := \emptyset$, and every budget $s \in \mathcal{S}$. Let $s^* \in \mathcal{S}$ be the minimum s such that the solution $\mathcal{Q}$ to the above budget multipath problem on $(C_0, \emptyset, \emptyset, s)$ visits at least k points. We will show in Section 4.4 that s^* is a near-optimal solution to the Euclidean k-TSP.

4.3 First Algorithm: Dynamic Program with Budgets

To simplify the presentation, we start by presenting in this section a first algorithm for the budget multipath problem that conveys the main ideas in the algorithmic design, although its running time is not as good as claimed in Theorem 7.

The algorithm is a dynamic program parameterized by the budget; see Section 4.3.1. The analysis of the algorithm contains the main technical novelty of the paper; see Section 4.3.2.

Later in Section 4.4, we improve the running time of the algorithm so as to achieve the claimed running time in Theorem 7.

4.3.1 Construction

Consider a fixed cell C, a fixed budget $s \in \mathcal{S}$, and a fixed fine multiset $B \subseteq \partial C$. We construct a set $\mathcal{M}_s^C[B]$ of pairs (M, κ), where M is a perfect matching on B and κ is an integer. Intuitively, κ indicates the number of points that can be visited by a collection of paths $\mathcal{Q}$ such that there is a one-to-one correspondence between the paths in $\mathcal{Q}$ and the edges in M and $\mathcal{Q}$ has total cost at most s.

For notional convenience, we denote

$$\mathcal{M}_s^C := \bigcup_{\text{fine multiset } B \subseteq \partial C} \mathcal{M}_s^C[B].$$

We construct $\mathcal{M}_s^C[B]$ in the bottom up order of the cell C in the quadtree; for a fixed cell C, in increasing order of the budget $s \in \mathcal{S}$; and for a fixed budget s, in non-decreasing order of cardinality of the fine multiset $B \subseteq \partial C$.

Leaf Cells

Consider a leaf cell C. We construct $\mathcal{M}_s^C[B]$ in non-decreasing order of $|B|$.
Case 1: $|B| = 0$. $\mathcal{M}_s^C[B] := \{(\emptyset, 0)\}$.

Case 2: $|B| = 2$. Let u and v be the two elements in B. Since C is a leaf cell, there are two subcases.

Subcase 2.1: $C \cap P = \emptyset$. Let

$$\mathcal{M}_s^C[\{u,v\}] := \begin{cases} \{(\{(u,v)\}, 0)\} & \text{if } \mathrm{dist}(u,v) \le s \\ \emptyset & \text{if } \mathrm{dist}(u,v) > s \end{cases}$$

Subcase 2.2: $C \cap P = \{p\}$ for some point p, letting $n_p \in \mathbb{N}$ denote the multiplicity of p in P. Let

$$\mathcal{M}_s^C[\{u,v\}] := \begin{cases} \{(\{(u,v)\}, n_p)\} & \text{if } \mathrm{dist}(u,p) + \mathrm{dist}(p,v) \le s \\ \{(\{(u,v)\}, 0)\} & \text{if } \mathrm{dist}(u,v) \le s < \mathrm{dist}(u,p) + \mathrm{dist}(p,v) \\ \emptyset & \text{if } \mathrm{dist}(u,v) > s \end{cases}$$

Case 3: $|B| > 2$. We construct $\mathcal{M}_s^C[B]$ using the following formula:

$$\mathcal{M}_s^C[B] := \bigcup_{\substack{s_1+s_2 \le s \\ s_1,s_2 \in \mathcal{S}}} \left(\bigcup_{\substack{u,v \in B \\ u \neq v}} \{(M \cup \{(u,v)\}, \kappa) | (M,\kappa) \in \mathcal{M}_{s_1}^C[B \setminus \{u,v\}], \mathrm{dist}(u,v) \le s_2\} \right). \tag{3}$$

Non-Leaf Cells

Consider a non-leaf cell C. Let $C_1, \ldots, C_\omega$ be the ω children of C in the quadtree. First, we enumerate all possible budgets $s_1, \ldots, s_\omega$ for the ω children, such that $\sum_i s_i \le s$. Next, we enumerate all possible pairs $(M_1, \kappa_1) \in \mathcal{M}_{s_1}^{C_1}, \ldots, (M_\omega, \kappa_\omega) \in \mathcal{M}_{s_\omega}^{C_\omega}$. As in [6], we say that the matchings $M_1, \ldots, M_\omega$ are *compatible* if (1) for any pair of neighboring cells C' and C'', the endpoints of the matchings on a shared facet are the same; and (2) combining $M_1, \ldots, M_\omega$ results in a set of paths with endpoints in ∂C. If the matchings $M_1, \ldots, M_\omega$ are compatible, we let M denote the matching that is the result of `Join`$(M_1, \ldots, M_\omega)$, where the `Join` operation is defined in [6]. If B equals the multiset consisting of the endpoints of the edges in M, we insert the pair $(M, \sum_i \kappa_i)$ into $\mathcal{M}_s^C[B]$.

4.3.2 Analysis

The following lemma shows that discretizing the possible lengths of a path into budgets in the construction of Section 4.3.1 preserves the near-optimality of the cost of the solution. Its proof is delicate.

▶ **Lemma 14.** *Let C be any cell. Let $\mathcal{Q}$ be an (m,r)-simple path collection in C of length at most $(1+\varepsilon)\cdot\Phi$. Let κ denote the number of points visited by $\mathcal{Q}$. Let multiset B consist of the intersection points between $\mathcal{Q}$ and ∂C. Let M denote a perfect matching on the points in B such that there is a one-to-one correspondence between the paths in $\mathcal{Q}$ and the edges in M. Then there exist $\kappa' \ge \kappa$ and $s \in \mathcal{S}$ such that $(M, \kappa') \in \mathcal{M}_s^C[B]$ and s is at most $(1+\varepsilon)$ times the total length of the paths in $\mathcal{Q}$.*

In the rest of Section 4.3.2, we prove Lemma 14.

Let τ denote the total length of the paths in $\mathcal{Q}$. We prove the claim in two cases, depending on whether τ is smaller or greater than $1/(r^2 + m^{1/(d-1)})$.

Case 1: $\tau < 1/(r^2 + m^{1/(d-1)})$

We show that for any $(u, v) \in M$, $u = v$. If $B = \emptyset$, this is trivial. Assume that $B \neq \emptyset$, and let p_1 and p_2 be two points in B.

▶ **Fact 15.** *Either* $\mathrm{dist}(p_1, p_2) \geq 1/(r^2 + m^{1/(d-1)})$ *or* $p_1 = p_2$.

Proof. Let l be the side length of C. By Section 4.1, $l \geq 1$.

Case (a): There exists a facet F of C that contains both p_1 and p_2. Since $\mathcal{Q}$ is an (m, r)-simple collection of paths, by the definition of B in the claim and by the definition of (m, r)-simple paths (Definition 10), B is fine. Let b_F denote the number of points of B that are in F. By the definition of a fine multiset (Definition 9), either $p_1 = p_2$ or $p_1, p_2 \in \mathrm{grid}(F, g(b_F))$, where $g(\cdot)$ is an integer-valued function such that $g(b_F) \leq r^{2d-2}/b_F$. Hence either $p_1 = p_2$ or, by Definition 8, $\mathrm{dist}(p_1, p_2) \geq l/(r^{2d-2})^{1/(d-1)} \geq 1/(r^2 + m^{1/(d-1)})$.

Case (b): There there exists no facet of C that contains both p_1 and p_2. Let F_1 and F_2 be two facets of C, such that F_1 contains p_1 and F_2 contains p_2. The non-trivial case is when F_1 and F_2 are neighboring faces. In this case, neither p_1 nor p_2 can lie on the intersection, because we are in the case where no facet of C contains both p_1 and p_2. Since B is a fine multiset, there are two cases according to Definition 9. In the first case, by Definition 8, for every facet F of C, the minimum distance between two points in $\mathrm{grid}(F, m)$ is $l/m^{1/(d-1)}$. Therefore, for both $i \in \{1, 2\}$, we have $\mathrm{dist}(p_i, F_1 \cap F_2) \geq l/m^{1/(d-1)}$. Similarly, in the second case, we have $\mathrm{dist}(p_i, F_1 \cap F_2) \geq l/(r^{2d-2})^{1/(d-1)}$. Furthermore, F_1 and F_2 are orthogonal. Therefore,

$$\mathrm{dist}(p_1, p_2) \geq \sqrt{d} \min\{l/m^{1/(d-1)}, l/(r^{2d-2})^{1/(d-1)}\} \geq 1/(r^2 + m^{1/(d-1)}).$$

This completes the proof of the claim. ◀

Hence for any $(u, v) \in M$, either $u = v$ or $\mathrm{dist}(u, v) \geq 1/(r^2 + m^{1/(d-1)})$. Since $1/(r^2 + m^{1/(d-1)}) > \tau \geq \sum_{(u,v) \in M} \mathrm{dist}(u, v)$, it is impossible that there exists $(u, v) \in M$ such that $\mathrm{dist}(u, v) \geq 1/(r^2 + m^{1/(d-1)})$. Thus, for any $(u, v) \in M$, $u = v$.

By the definition of the quadtree in Section 4.1, the distance from any point in P to ∂C is at least $1/2$, so $\mathcal{Q}$ does not visit any point from P. Hence $\kappa = 0$. By induction on the size of M', for any perfect matching M' on B such that $\forall (u, v) \in M', u = v$, we have $(M', 0) \in \mathcal{M}_0^C[B]$. Thus, $(M, \kappa') \in \mathcal{M}_s^C[B]$, where $s = 0 \leq (1 + \varepsilon) \cdot \tau$ and $\kappa' = 0 \geq \kappa$. The lemma holds when $\tau < 1/(r^2 + m^{1/(d-1)})$.

Case 2: $\tau \geq 1/(r^2 + m^{1/(d-1)})$

▶ **Fact 16.** *Let C be a cell in the quadtree. Let B be a fine multiset of points in ∂C. We have $|B| \leq 2d \cdot 2r^{d-1}$.*

Proof. Let F be a facet of C and b_F denote the number of points of B that are on F. By Definition 9, either one of the following two cases holds: (i) $b_F \leq 1$ and $B \cap F \subseteq \mathrm{grid}(F, m)$; (ii) $b_F \geq 2$ and $B \cap F \subseteq \mathrm{grid}(F, g(b_F))$ for some $g(b_F) \leq r^{2d-2}/b_F$. Moreover, each point in $\mathrm{grid}(F, g(b_F))$ occurs at most twice in $B \cap F$.

We show that $b_F \leq 2r^{d-1}$. In case (i), this is trivial. In case (ii), since each point from $\mathrm{grid}(F, g(b_F))$ is contained at most twice in $B \cap F$, we have $b_F \leq 2g(b_F)$. Together with $g(b_F) \leq r^{2d-2}/b_F$, we have $b_F \leq 2r^{d-1}$. Furthermore, since C is a d-dimensional hypercube, C has $2d$ facets. The claim follows. ◀

▶ **Lemma 17.** *Let C be any cell. Let h denote the height of the subtree rooted at C. Let $\mathcal{Q}$ be an (m, r)-simple path collection in C of length at most $(1 + \varepsilon) \cdot \Phi$. Let τ denote the length of $\mathcal{Q}$. Let κ denote the number of points visited by $\mathcal{Q}$. Let multiset B consist of the intersection points between $\mathcal{Q}$ and ∂C. Let M denote a perfect matching on the points in B such that there is a one-to-one correspondence between the paths in $\mathcal{Q}$ and the edges in M. Then there exist $\kappa' \geq \kappa$ and $s \in \{\left(1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n)\right)^i, i \in \mathbb{Z}\}$ such that $(M, \kappa') \in \mathcal{M}_s^C[B]$ and $s \leq \tau \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h}$.*

Proof. We proceed by induction in the bottom-up order of the cell C in the quadtree.

Case (a): C is a leaf cell. Observe that B is obtained by $|B|/2$ inclusion operations of pairs of points in the construction in Section 4.3.1. By Definition 10, B is a fine multiset. Therefore, by Fact 16, we have $|B| \leq 2d \cdot 2r^{d-1}$. Since the cost inside C is obtained after at most $|B|/2 \leq 2d \cdot r^{d-1}$ rounding operations, there exists $s \in \{\left(1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n)\right)^i, i \in \mathbb{Z}\}$ such that $s \leq \tau \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1}}$ such that $(M, \kappa') \in \mathcal{M}_s^C[B]$ for some $\kappa' \geq \kappa$.

Case (b): C is a non-leaf cell. Let $C_1, \ldots, C_\omega$ be the children of C in the decomposition. Let κ_i denote the number of points visited by $\mathcal{Q}$ inside of C_i. Let τ_i denote the cost of $\mathcal{Q}$ inside of C_i. Let h_i denote the height of the subtree rooted at C_i. By induction, for each $i \in [1, \omega]$, there exist $\kappa_i' \geq \kappa_i$ and $s_i \in \{\left(1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n)\right)^i, i \in \mathbb{Z}\}$ such that $(M_i, \kappa_i') \in \mathcal{M}_{s_i}^{C_i}[B_i]$ and $s_i \leq \tau_i \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h_i}$.

Let $\kappa' := \sum_i \kappa_i'$ and let s be the smallest budget in $\mathcal{S}$ that is at least $\sum_i s_i$.

Combining the solutions in all C_i and noting that $h \geq h_i + 1$ for all i, we have

$$(M, \kappa') \in \mathcal{M}_s^C[B]$$

and

$$\kappa' = \sum_i \kappa_i' \geq \sum_i \kappa_i = \kappa$$

and

$$\begin{aligned} s &\leq (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n)) \cdot \sum_i s_i \\ &\leq (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n)) \cdot \sum_i \tau_i \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h_i} \\ &\leq (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h} \sum_i \tau_i \\ &= (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h} \cdot \tau. \end{aligned}$$

This completes the proof of the claim. ◀

Finally, let us bound $2d \cdot r^{d-1} + h$. Since the instance is well-rounded, there exists an absolute constant D such that the size of the bounding box is at most Dk^2 (Definition 2). Therefore, the height of the quadtree is at most $\lceil \log_2(Dk^2) \rceil \leq \log_2(Dn^2) + 1 \leq \log_2 D + 2\log_2 n + 1 \leq 3\log_2 n$. Thus, $2d \cdot r^{d-1} + h \leq 2d \cdot r^{d-1} + 3\log_2 n$ for n large enough. Hence

$$\begin{aligned} \tau \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + h} &\leq \tau \cdot (1 + \varepsilon/(2d \cdot r^{d-1} + 3\log_2 n))^{2d \cdot r^{d-1} + 3\log_2 n} \\ &\leq \tau \cdot (1 + \varepsilon). \end{aligned}$$

By Lemma 17, there exists $s \in \{\left(1+\varepsilon/(2d \cdot r^{d-1}+3\log_2 n)\right)^i, i \in \mathbb{Z}\}$ and $\kappa' \geq \kappa$ such that $(M, \kappa') \in \mathcal{M}_s^C[B]$ and $s \leq (1+\varepsilon/(2d \cdot r^{d-1}+3\log_2 n)^{2d \cdot r^{d-1}+h} \cdot \tau$. We have $s \leq (1+\varepsilon) \cdot \tau \leq (1+\varepsilon)^2 \cdot \Phi$. Therefore, $s \in \mathcal{S}$. This completes the proof of Lemma 14.

4.4 Improved Algorithm and Proof of Theorem 7

4.4.1 Construction

In order to achieve the claimed running time in Theorem 7, we combine the algorithm in Section 4.3 with the *rank-based approach* from [6].

Let Γ_B denote the set of all perfect matchings on B. We say that M_1 and M_2 in Γ_B *fit* if their union is a Hamiltonian Cycle on B.

▶ **Definition 18** (representation, [6]). *Let B be a set. Let $\mathcal{A}$ and $\mathcal{A}'$ be two subsets of $\Gamma_B \times \mathbb{N}$. We say that $\mathcal{A}'$* represents *$\mathcal{A}$ if for all $M \in \Gamma_B$ we have*

$$\max\{\kappa | (M', \kappa) \in \mathcal{A}' \text{ and } M \text{ fits } M'\} = \max\{\kappa | (M', \kappa) \in \mathcal{A} \text{ and } M \text{ fits } M'\}.$$

▶ **Lemma 19** (reduce, [3, Theorem 3.7], see also [6, Lemma 5.2 in the full version]). *There exists an algorithm, called* `reduce`*, that given a set B and $\mathcal{A} \subseteq \Gamma_B \times \mathbb{N}$, computes in time $|\mathcal{A}| \cdot 2^{O(|B|)}$ a set $\mathcal{A}' \subseteq \mathcal{A}$ such that $\mathcal{A}'$ represents $\mathcal{A}$ and $|\mathcal{A}'| \leq 2^{|B|-1}$.*

Let C be a cell in the quadtree and let $s \in \mathcal{S}$. We define the family $\{\mathcal{R}_s^C[B]\}_B$ and the set $\mathcal{R}_s^C$ in the same way as we define the family $\{\mathcal{M}_s^C[B]\}_B$ and the set $\mathcal{M}_s^C$ in Section 4.3, except that we use `reduce` so as to keep the number of elements in $\mathcal{R}_s^C[B]$ bounded.

▶ Remark 20. It is standard to enrich the dynamic program so that we obtain a collection of paths instead of the total length of that collection. Indeed, once the dynamic programming table is computed, one can recursively reconstruct the corresponding path.

4.4.2 Analysis

▶ **Lemma 21** (adaptation from [6, Lemma 5.3 in the full version]). *For any cell C in the quadtree, any budget $s \in \mathcal{S}$, and any fine multiset $B \subseteq \partial C$, the set $\mathcal{R}_s^C[B]$ represents $\mathcal{M}_s^C[B]$.*

Lemma 22 is an adaptation from [6].

▶ **Lemma 22** (adaptation from [6, Lemma 5.4 and Claim 5.5 in the full version]). *The running time of the algorithm for all cells C in the quadtree, for all budgets $s \in \mathcal{S}$, and for all fine multisets $B \subseteq \partial C$ is $n \cdot 2^{O(r^{d-1})} \cdot \log^{2d^2 \cdot 2^d} n$.*

Proof of Theorem 7. From the structure theorem (Theorem 11), with probability at least 1/2, there exists a k-salesman tour π that is (m, r)-simple and such that

$$w(\pi) \leq (1+\varepsilon) \cdot \text{opt}. \tag{4}$$

We condition on the above event in the rest of the analysis.
According to [1],

$$\text{opt} \leq dk^{1-1/d} L = \Phi,$$

where the equality follows by the definition of Φ (Definition 12). Therefore,

$$w(\pi) \leq (1+\varepsilon) \cdot \Phi.$$

Let C_0 be the root cell of the quadtree. Since π is a closed path strictly contained in C_0, the set of points of ∂C_0 is $B := \emptyset$, and the only matching on B is $M := \emptyset$. Let κ be the number of points visited by π. Since π is a k-salesman tour, $\kappa \geq k$. Since $w(\pi) \leq (1+\varepsilon) \cdot \Phi$, we may apply Lemma 14 on C_0 and $\{\pi\}$ and obtain an integer $\kappa' \geq \kappa$ and a budget $s \in \mathcal{S}$ such that $(\emptyset, \kappa') \in \mathcal{M}_s^{C_0}$ and

$$s \leq (1+\varepsilon) \cdot w(\pi). \tag{5}$$

Lemma 21 ensures that $\mathcal{R}_s^{C_0}[\emptyset]$ represents $\mathcal{M}_s^{C_0}[\emptyset]$, hence

$$\max\{\kappa'' | (M', \kappa'') \in \mathcal{M}_s^{C_0}[\emptyset] \text{ and } \emptyset \text{ fits } M'\} = \max\{\kappa'' | (M', \kappa'') \in \mathcal{R}_s^{C_0}[\emptyset] \text{ and } \emptyset \text{ fits } M'\}.$$

Since $\emptyset$ fits $\emptyset$ and $(\emptyset, \kappa') \in \mathcal{M}_s^{C_0}[\emptyset]$, we have $\kappa' \leq \max\{\kappa'' | (M', \kappa'') \in \mathcal{M}_s^{C_0}[\emptyset]$ and $\emptyset$ fits $M'\}$. Therefore, there exists $(M', \kappa'') \in \mathcal{R}_s^{C_0}[\emptyset]$ such that M' fits $\emptyset$ and $\kappa'' \geq \kappa'$. The only matching M' on $\emptyset$ that fits $\emptyset$ is $\emptyset$, hence $M' = \emptyset$. Thus, $(\emptyset, \kappa'') \in \mathcal{R}_s^{C_0}$, for some $\kappa'' \geq \kappa' \geq k$.

Let s^* be the output of the algorithm, which is the minimum budget such that $(\emptyset, \kappa'') \in \mathcal{R}_{s^*}^{C_0}$ for some $\kappa'' \geq k$. From (4) and (5), we have

$$s^* \leq (1 + 3\varepsilon) \cdot \text{opt}.$$

Replacing ε by $\varepsilon' := \varepsilon/3$ leads to the approximation ratio in the claim.

The running time in the claim follows from Lemma 22.

For the derandomization, observe that the only step using randomness is the random shift to construct the quadtree. Since there are $O(n^d)$ possible shifts, the algorithm can be derandomized by increasing the running time by a factor $O(n^d)$.

This completes the proof of Theorem 7. ◀

▶ Remark 23. The spanner techniques introduced by Rao and Smith [10] lead to a better running time for TSP, but those techniques do not seem to apply to k-TSP. Indeed, a key property for TSP is the existence of a near-optimal solution using the spanner only (see Lemma 5.1 of [6]). However, this property does not hold for k-TSP, since the solution to k-TSP might be much less expensive compared with the spanner of the entire graph.

References

1 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

2 Baruch Awerbuch, Yossi Azar, Avrim Blum, and Santosh Vempala. Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of Computing (STOC)*, pages 277–283, 1995.

3 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015.

4 Michael R. Garey, Ronald L. Graham, and David S. Johnson. Some NP-complete geometric problems. In *Proceedings of the eighth annual ACM Symposium on Theory of Computing (STOC)*, pages 10–22, 1976.

5 Sariel Har-Peled and Benjamin Raichel. Net and prune: A linear time algorithm for Euclidean distance problems. *Journal of the ACM (JACM)*, 62(6):1–35, 2015.

6 Sándor Kisfaludi-Bak, Jesper Nederlof, and Karol Węgrzycki. A Gap-ETH-tight approximation scheme for Euclidean TSP. In *IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 351–362, 2021. Full version at `arXiv:2011.03778`.

7 Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the Euclidean k-median problem. *SIAM Journal on Computing*, 37(3):757–782, 2007.

8 Cristian S. Mata and Joseph S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proceedings of the eleventh annual Symposium on Computational Geometry (SoCG)*, pages 360–369, 1995.

9 Joseph SB Mitchell, Avrim Blum, Prasad Chalasani, and Santosh Vempala. A constant-factor approximation algorithm for the geometric k-mst problem in the plane. *SIAM Journal on Computing*, 28(3):771–781, 1998.

10 Satish B Rao and Warren D Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 540–550, 1998.