A (5/3+ε)-Approximation for Tricolored Non-Crossing Euclidean TSP

Júlia Baligács
Technische Universität Darmstadt, Germany

Yann Disser
Technische Universität Darmstadt, Germany

Andreas Emil Feldmann
University of Sheffield, UK

Anna Zych-Pawlewicz
University of Warsaw, Poland

Abstract

In the Tricolored Euclidean Traveling Salesperson problem, we are given $k = 3$ sets of points in the plane and are looking for disjoint tours, each covering one of the sets. Arora (1998) famously gave a PTAS based on "patching" for the case $k = 1$ and, recently, Dross et al. (2023) generalized this result to $k = 2$. Our contribution is a $(5/3 + \varepsilon)$-approximation algorithm for $k = 3$ that further generalizes Arora's approach. It is believed that patching is generally no longer possible for more than two tours. We circumvent this issue by either applying a conditional patching scheme for three tours or using an alternative approach based on a weighted solution for $k = 2$.

2012 ACM Subject Classification Mathematics of computing → Combinatorial optimization; Mathematics of computing → Approximation algorithms; Theory of computation → Computational geometry

Keywords and phrases Approximation Algorithms, geometric Network Optimization, Euclidean TSP, non-crossing Structures

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.15

Related Version *Full Version*: `https://arxiv.org/abs/2402.13938`

Funding *Anna Zych-Pawlewicz*: This work is a part of project BOBR that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 948057).

1 Introduction

We consider the *k-Colored Euclidean Traveling Salesperson problem (k-ETSP)* where k sets of points have to be covered by k disjoint curves in the plane (cf. Figure 1). This is a fundamental problem in geometric network optimization [24] and generalizes the well-known *Euclidean Traveling Salesperson problem (ETSP)*. It captures applications ranging from VLSI design [12, 21, 29, 30] to set visualisation of spatial data [1, 7, 11, 16, 27].

Formally, an instance of k-ETSP is a partition $(T_c)_{c\in C}$ of a set of *terminals* $T \subseteq \mathbb{R}^2$ in the Euclidean plane, where $|C| = k$. We consider every $c \in C$ to be a *color* and every point in T_c to be of color c. A solution to the instance is a k-tuple $\Pi = (\pi_c)_{c\in C}$ of closed curves in $\mathbb{R}^2$, also referred to as *tours*, such that every curve π_c visits all terminals of color c, i.e., $T_c \subseteq \pi_c$,[1] and the curves are pairwise disjoint, i.e., $\pi_c \cap \pi_{c'} = \emptyset$ for $c \neq c'$.

[1] For convenience, we identify curves with their images.

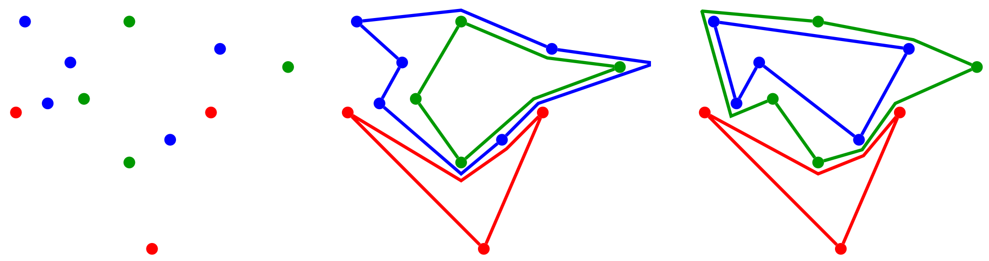

Figure 1 An instance of 3-ETSP together with two possible solutions. An optimum solution does not exist: The curves can get arbitrarily close but must not touch. Observe that, in the middle subfigure, the red and green tour are not δ-close for any $\delta > 0$, as the blue tour lies in between.

The objective of k-ETSP is to minimize the total length of the tours, i.e., to minimize $l(\Pi) := \sum_{c \in C} l(\pi_c)$, where $l(\pi)$ denotes the Euclidean length of π. It is important to note that, for $k > 1$, an optimum does not always exist (cf. Figure 1).[2] In order to still define an approximation, we follow the approach of [10] by defining the value $\textsc{Opt}^* := \inf\{l(\Pi) : \Pi \text{ is a solution}\}$ and saying that a solution Π is an α-approximation if $l(\Pi) \leq \alpha \textsc{Opt}^*$.

The k-ETSP problem inherits NP-hardness from its special case ETSP [25] for $k = 1$. It is well-known that 1-ETSP (i.e., ETSP) admits a Polynomial-Time Approximation Scheme (PTAS) [2], and the result was recently extended to a PTAS for 2-ETSP [10]. The best known approximation factor for 3-ETSP was $(10/3 + \varepsilon)$ via doubling of the solution to 3-Colored Non-crossing Euclidean Steiner Forest from [5].

Our results. Our main result is the following.

▶ **Theorem 1.** *For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3} + \varepsilon\right)$-approximation for* 3-ETSP *in time $\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}$.*

To prove Theorem 1, we adapt Arora's algorithm for Euclidean TSP [2]. One of the key ingredients of that algorithm is the so-called *Patching Lemma* which allows to locally modify any tour such that the number of crossings with a line segment is bounded, without increasing the length of the tour too much. It was shown in [10] that this is still possible for two tours, but it does not seem to be possible for more than two non-crossing tours (see Figure 2, [5, 10]). We show how to circumvent this issue by imposing an additional condition on the tours to be patched. For this, we say that two tours are *δ-close* if they can be connected by a straight line segment of length at most δ that is disjoint from the third tour (cf. Figure 1).

▶ **Lemma 2.** *Let s be a straight line segment and $\delta = l(s)$ be its length. Let a solution to* 3-ETSP *be given in which two of the three tours are not δ-close. For every $\delta' > 0$, the solution can be modified inside a δ'-neighbourhood*[3] *of s such that it intersects s in at most* 18 *points and its cost is increased by at most $O(\delta)$.*

For the case where patching is not possible, we take a different approach. For this, we define a *two-tour presolution* to be a pair of disjoint tours $(\pi_{cc'}, \pi_{c''})$ such that $\pi_{cc'}$ visits all terminals colored c and c', and $\pi_{c''}$ visits all terminals colored c''. Such tours can easily be transformed into a feasible solution to 3-ETSP by "doubling" $\pi_{cc'}$ (cf. Figure 3, Observation 6). We call the resulting solution an *induced two-tour solution.*

[2] For $k = 1$, an optimum always exists because tours are allowed to self-intersect.
[3] A formal definition is given in Section 3.

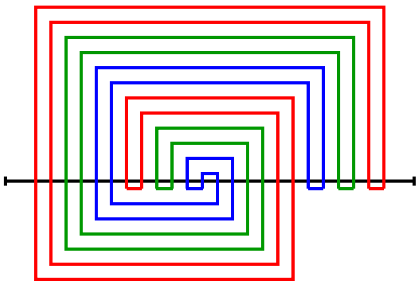

Figure 2 A modified example from [5] that is presumably non-patchable.

▶ **Lemma 3.** *For every $\varepsilon > 0$, there exists $\delta > 0$ such that, for every $(1+\varepsilon)$-approximate solution to* 3-ETSP *in which the two shorter tours are δ-close, we can find a two-tour presolution (π_1, π_2) with $2l(\pi_1) + l(\pi_2) \leq \left(\frac{5}{3} + 2\varepsilon\right) \cdot \textsc{Opt}^*$.*

Similarly as in [2], we place a suitable grid on the plane and place so-called *portals* on the grid lines (cf. Section 3.1). Roughly speaking, a solution to k-ETSP is *portal-respecting* if it only intersects grid lines at portals and intersects every portal at most a constant number of times (see Section 3.1 for a formal definition). Combining the ideas in [2] with Lemmas 2 and 3, we obtain the following result.

▶ **Theorem 4.** *For every instance of* 3-ETSP *and $\varepsilon > 0$, either, there is a solution that is a $(1+\varepsilon)$-approximation and portal-respecting with respect to a suitable grid, or there is a portal-respecting two-tour presolution that induces a $\left(\frac{5}{3} + \varepsilon\right)$-approximation.*

Theorem 4 allows us to restrict ourselves to finding portal-respecting solutions. The last step is to show that such solutions can be computed in polynomial time using dynamic programming. For this, we generalize the approach in [10] to any number of colors k while simultaneously allowing for weighted tours. We denote this generalized problem by k-ETSP$'$ (see Section 4 for a formal definition).

▶ **Theorem 5.** *For every $k \in \mathbb{N}$, there is a polynomial-time algorithm that computes a parametric solution $\Pi(\lambda)$ to k-ETSP$'$ such that $\lim_{\lambda \to 0} l(\Pi(\lambda)) = \textsc{Opt}^*$.*

Here, a parametric solution is a function $\Pi\colon (0,\infty) \to \{\Pi' : \Pi' \text{ is a solution}\}$ that continuously[4] interpolates between solutions. Intuitively, the algorithm of Theorem 5 computes the optimal combinatorial structure of a solution, i.e., the optimal order in which portals and terminals are visited or bypassed, and the parameter λ sets the spacing between the tours. Importantly, our solution allows to efficiently recover the (non-parametric) solution $\Pi(\lambda)$ for given $\lambda > 0$ and to compute $\textsc{Opt}^*$.

Comparison to previous work. Our work is closely related to the papers by Dross et al. [10] and Bereg et al. [5]. While each work adapts Arora's algorithm [2], we have to overcome significant additional difficulties. The main contribution of [5] is that Arora's patching lemma can be adapted to two Steiner trees, and three Steiner trees if one of them may use parts of another. The main contribution of [10] is that Arora's patching lemma can be adapted to two TSP tours. In contrast to these two results, our patching procedure needs to be

[4] For example, with respect to the Fréchet-distance on the space of curves, which is defined as follows: For $\pi_1, \pi_2\colon [0,1] \to \mathbb{R}^2$, the Fréchet distance between π_1 and π_2 is $d_{\text{Fr}}(\pi_1,\pi_2) := \sup_{t\in[0,1]} \|\pi_1(t) - \pi_2(t)\|$.

more involved: in our setting we have to deal with more complex crossing patterns whose mono-colored groups cannot be reduced to a single size (they were reduced to size 1 in [5] and 2 in [10]), and we have to ensure that the modified tours remain connected (which is more immediate in [5] and [10]) and that they remain Jordan curves (which is more immediate in [10] and not needed in [5]). Additionally, in contrast to [10], we develop an alternative approach when patching is not possible. We also generalize Arora's dynamic program to any number of weighted, portal-respecting tours, and its portal-snapping technique to arbitrary arrangements of line segments.

Related work. Our results build upon the celebrated PTAS by Arora [2] for ETSP, which was gradually improved [26, 3] to an EPTAS [20] with the running time proven tight under the gap-ETH. The 2-ETSP problem also admits a gap-ETH tight EPTAS [10], which is based on the techniques introduced in [2, 20], but relies on a more involved patching lemma compared to the one in [2]. The authors of [10] claim that patching is unlikely to work for the 3-ETSP problem (cf. Figure 2) and they leave it as a central open problem whether there is a PTAS for 3-ETSP. We present a $(\frac{5}{3} + \varepsilon)$-approximation algorithm that combines a new patching method for three tours with a complementary approach for the case where patching is not possible. We believe that our approach is applicable for a wider range of non-crossing problems, for instance for the Red-Blue-Green-Yellow Separation problem (cf. [10]).

Interestingly, similar progress was earlier obtained for the problem of computing k pairwise non-crossing Euclidean Steiner trees, one for each color of a k-colored set of terminals in the plane. The k-Colored Non-crossing Euclidean Steiner Forest problem (k-CESF for short) was introduced and studied in [11]. Later, Bereg et al. [5] showed a PTAS for 2-CESF and a $(\frac{5}{3} + \varepsilon)$-approximation algorithm for 3-CESF, leaving the existence of PTAS for 3-CESF a main open question. This may suggest that $\frac{5}{3}$ could be some natural barrier for computing three non-crossing curves interconnecting three point sets in the plane.

Other problems in geometric network optimization include the following: In the k-Minimum Spanning Tree problem, we have to find a spanning tree connecting a subset of size k of the terminals [6, 26]. In the k-Traveling Repairperson problem, we can use k tours (that are allowed to intersect) to cover the terminals, objective to minimizing the latency, i.e., the sum of the times at which a terminal is visited [8, 9, 13]. In the Traveling Salesperson problem with neighbourhoods, the task is to find a shortest tour that visits at least one point in each of a set of neighbourhoods [15, 22, 23, 28].

Moreover, the TSP problem has been extensively studied for other metric spaces. For example, it is known that there is a PTAS in the case of a metric space of bounded doubling dimension [4, 14]. On the other hand, it is known that a PTAS for general metric spaces does not exist [19]. Currently, the best approximation algorithm known in general metric spaces was suggested by Karlin et al. [17, 18], achieving an approximation factor of $1.5 - 10^{-36}$.

2 Two-tour presolutions

Recall that a *two-tour presolution* for 3-ETSP is a pair of disjoint closed curves $(\pi_{cc'}, \pi_{c''})$ such that $\pi_{cc'}$ visits all terminals in $T_c \cup T_{c'}$ and $\pi_{c''}$ visits all terminals in $T_{c''}$ for some $\{c, c', c''\} = \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$, where $\{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ denotes throughout the paper the color set C in the case of 3-ETSP, standing for red, green, and blue. In this section, let $c'' = \mathrm{B}$ without loss of generality. We first investigate how two-tour presolutions can be transformed into solutions to 3-ETSP.

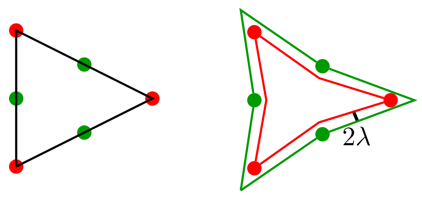

Figure 3 On the left, we have a single curve π_{RG} visiting all red and green points. On the right, we have replaced π_{RG} by two parametrized disjoint curves $\pi_{\text{R}}(\lambda)$ and $\pi_{\text{G}}(\lambda)$ with Fréchet-distance at most λ to π_{RG}, visiting the terminals of the corresponding color. In particular, the Fréchet-distance between π_{R} and π_{G} is at most 2λ.

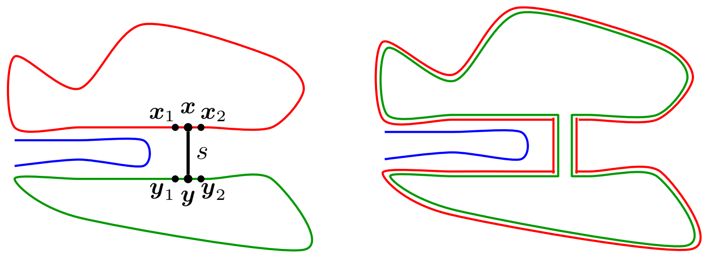

Figure 4 On the left, we are given a solution to 3-ETSP where the red and green tour are δ-close. On the right, we see how the solution can be transformed into an induced two-tour solution.

For this, note that, if we are given a single tour π_{RG} that visits all red and green terminals, it is possible to replace it by two parametrized disjoint tours $\pi_{\text{R}}(\lambda)$ and $\pi_{\text{G}}(\lambda)$ that have Fréchet-distance at most λ from π_{RG} such that $\pi_{\text{R}}(\lambda), \pi_{\text{G}}(\lambda)$ visit all red, respectively green, terminals and $\lim_{\lambda\to 0} l(\pi_{\text{R}}(\lambda)) = \lim_{\lambda\to 0} l(\pi_{\text{G}}(\lambda)) = l(\pi_{\text{RG}})$ (cf. Figure 3). Choosing $\lambda > 0$ small enough and considering the tours $\pi_{\text{R}}(\lambda), \pi_{\text{G}}(\lambda)$, we obtain the following.

▶ **Observation 6.** *Fix an instance of* 3-ETSP *and let* $\pi_{\text{RG}}, \pi_{\text{B}}$ *be a two-tour presolution. For every* $\delta > 0$*, there is a solution to* 3-ETSP *of cost at most* $2 \cdot l(\pi_{\text{RG}}) + l(\pi_{\text{B}}) + \delta$*, called an* induced two-tour solution.

Next, we show that, if the two shorter tours of a $(1+\varepsilon)$-approximation for 3-ETSP are in some sense close to each other, then there is a good two-tour presolution. However, note that this is not a reduction to 2-ETSP: In 2-ETSP, the objective is to minimize $l(\pi_{\text{B}}) + l(\pi_{\text{RG}})$. In our case, we need to minimize $l(\pi_{\text{B}}) + 2 \cdot l(\pi_{\text{RG}})$, i.e., we need to solve a weighted variant of 2-ETSP. We will see later that we can compute a $(1+\varepsilon)$-approximation for this weighted variant of 2-ETSP in polynomial time (cf. Theorem 15).

Recall that two tours are δ-close if they can be connected by a straight line segment of length at most δ that does not intersect the third tour. The construction for proving the following lemma is illustrated in Figure 4.[5]

▶ **Lemma 7.** *Let* $\Pi = (\pi_{\text{R}}, \pi_{\text{G}}, \pi_{\text{B}})$ *be a solution to a given instance of* 3-ETSP *and let* $\delta > 0$*. Wlog., let* π_{B} *be the longest tour, i.e.,* $l(\pi_{\text{B}}) \geq l(\pi_{\text{R}}), l(\pi_{\text{G}})$*. Assume that* π_{R} *and* π_{G} *are* δ*-close. Then, there is a two-tour presolution* $(\pi_{\text{RG}}, \pi_{\text{B}})$ *with* $2 \cdot l(\pi_{\text{RG}}) + l(\pi_{\text{B}}) \leq \frac{5}{3} \cdot l(\Pi) + 8\delta$*.*

Note that applying Lemma 7 to a $(1+\varepsilon)$-approximation with $\delta \leq \varepsilon\text{OPT}^*/24$ gives a two-tour presolution $(\pi_{\text{RG}}, \pi_{\text{B}})$ with

$$2 \cdot l(\pi_{\text{RG}}) + l(\pi_{\text{B}}) \leq \frac{5}{3} \cdot (1+\varepsilon) \cdot \text{OPT}^* + 8\delta = \left(\frac{5}{3} + \frac{5}{3}\varepsilon\right) \cdot \text{OPT}^* + 8\delta \leq \left(\frac{5}{3} + 2\varepsilon\right) \cdot \text{OPT}^*,$$

which completes the proof of Lemma 3.

[5] This and all other missing proofs are deferred to the full version.

3 Our structure theorem

In this section, we prove our structure theorem for 3-ETSP (cf. Theorem 4) which, roughly speaking, states the following: For every $\varepsilon > 0$, either, there is a two-tour presolution that induces a $\left(\frac{5}{3} + \varepsilon\right)$-approximation, or there is a $(1+\varepsilon)$-approximate solution that fulfills some additional constraints (or both). Later, we will see that it is possible to find a good solution that fulfills these additional constraints and a good two-tour presolution in polynomial time.

Our final algorithm for 3-ETSP will preprocess the input such that the terminals remain distinct points and have integer coordinates. This allows us to assume throughout this section that terminals lie in $\{0, \ldots, L\}^2$ for some integer L that is a power of 2. As the problem is not interesting for small L, we also assume $L \geq 4$ whenever necessary. In Section 5, we explain in more detail how we preprocess the input and show that a near-optimal solution to the preprocessed input can be transformed in polynomial time to a near-optimal solution to the original input.

Following [10], we assume without loss of generality that, for every $\varepsilon > 0$ and $\delta > 0$, there is a $(1+\varepsilon)$-approximate solution to 3-ETSP whose tours consist of straight line segments, where each segment connects two points that each are at distance at most δ from a terminal. To see this intuitively, interpret each tour of a solution as a sequence of terminals to visit or bypass. The cheapest way to realize such a sequence is by straight line segments with endpoints arbitrarily close to terminals (cf. Figure 1). For this reason, we will assume from now on that all tours that we work with consist of such straight line segments. Since we assume in this section that terminals lie in $\{0, \ldots, L\}^2$, we have in particular, that the straight line segments have endpoints in $N_\delta\left(\{0, \ldots, L\}^2\right)$, where $N_\delta(A) := \{\boldsymbol{x} : \|\boldsymbol{x} - \boldsymbol{a}\| < \delta \text{ for some } \boldsymbol{a} \in A\}$ denotes the δ-neighbourhood of a set A.

3.1 Dissection and portals

In this subsection, we place a suitable grid on the Euclidean plane and place some portals on it through which the tours will later be allowed to cross the grid lines. For this, we follow the definitions as in [2]. The aforementioned additional constraints for the $(1+\varepsilon)$-approximate solution in Theorem 4 strongly relate to this construction and are crucial for efficient computation.

Fix an instance of k-ETSP with $T \subseteq \{0, \ldots, L\}^2$ where L is a power of two. We pick a *shift vector* $\boldsymbol{a} = (a_1, a_2) \in \{0, \ldots, L-1\}^2$ and consider the square

$$C(\boldsymbol{a}) := \left[-a_1 - \frac{1}{2}, 2L - a_1 - \frac{1}{2}\right] \times \left[-a_2 - \frac{1}{2}, 2L - a_2 - \frac{1}{2}\right],$$

i.e., $C(\boldsymbol{a})$ is the square $[0, \ldots, 2L]^2$ shifted by $-\boldsymbol{a} - (0.5, 0.5)$. Note that $C(\boldsymbol{a})$ contains every terminal.

The *dissection* $D(\boldsymbol{a})$ is a full 4-ary tree defined as follows (illustrated in Figure 5): Each node is a square in $\mathbb{R}^2$. The root of $D(\boldsymbol{a})$ is $C(\boldsymbol{a})$. Given a node S of the tree of side length more than one, we partition S into four smaller equal sized squares and these define the four children of S. If S has side length one, it is a leaf. Note that this is well-defined because L is a power of two.[6]

Given a square S, we define its *border edges* to be the unique four straight line segments bounding it and we define its *border* ∂S to be the union of the border edges.

[6] In previous work, the well-known *quad-trees* are defined as a subtree of the dissection, on which the dynamic program of [2] is based, which however we do not rely on in this work.

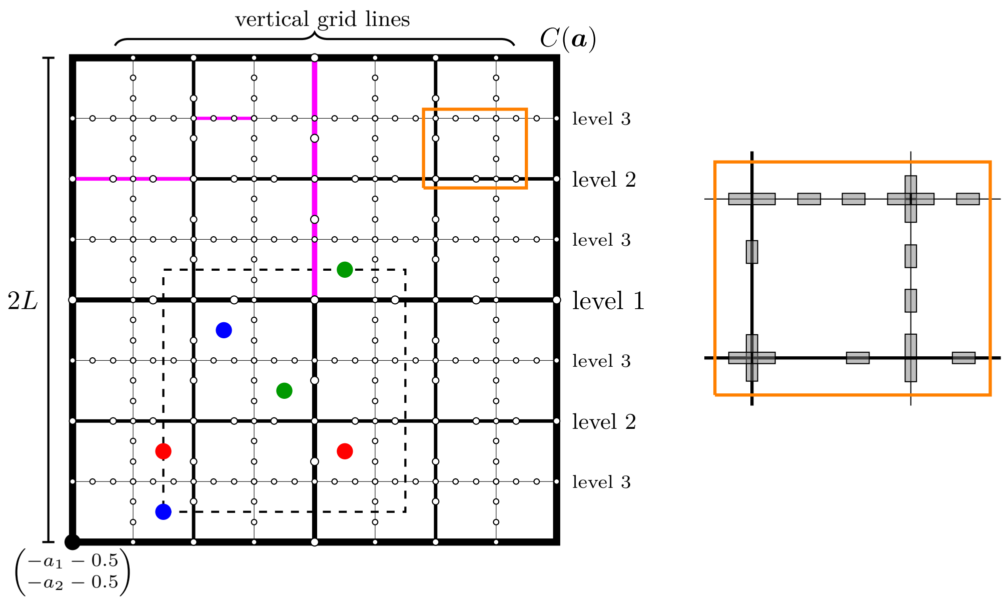

Figure 5 The figure on the left illustrates the dissection $D(\boldsymbol{a})$ with $L = 4, \boldsymbol{a} = (1, 0)$. The dashed lines denote $\partial[0, L]^2$. The three pink lines are examples of boundaries of levels one, two, and three. The levels of all horizontal grid lines are indicated. We have placed 4 portals on every boundary, represented as circles. For better overview, we have drawn only one portal at endpoints of boundaries. Note that the endpoint of a boundary is actually contained in up to four portals. This is illustrated on the right hand side, where the exact portal placement of the marked orange area is given.

A *grid line* is either a horizontal line containing $(0, -a_2 - 0.5 + k)$ or a vertical line containing $(-a_1 - 0.5 + k, 0)$ for some $k \in \{1, \dots, 2L - 1\}$. Note that every border edge of a square in $D(\boldsymbol{a})$ is either contained in a grid line or contained in a border edge of $C(\boldsymbol{a})$ (which is not on a grid line). Since terminals have coordinates in $\mathbb{Z}$ and grid lines have coordinates in $0.5 + \mathbb{Z}$, no terminal lies on a grid line. More precisely, every terminal lies exactly in the center of a leaf of $D(\boldsymbol{a})$.

A *boundary* is a border edge of a non-root node in $D(\boldsymbol{a})$ not contained in another border edge (see Figure 5 for an example). Observe that its length is $\frac{2L}{2^i}$ for some $i \in \{1, \dots, \log(2L)\}$. Then, we define its *level* as i. Note that a grid line only contains boundaries of the same level so we can define the level of a grid line as the level of the boundaries that it contains (cf. Figure 5). If two boundaries (or grid lines) of level i and j are given with $i < j$, we say that level j is *deeper* than level i (resembling the property that the corresponding boundary belongs to a node that is deeper in the dissection), and level i is *shallower* than level j.

Observe that there is precisely one vertical (respectively horizontal) grid line of level one and, for every $i \in \{1, \dots, \log(2L) - 1\}$, there are twice as many grid lines of level $i + 1$ as grid lines of level i. In total, there are $2L - 1$ horizontal and $2L - 1$ vertical grid lines. With this, we immediately obtain the following property.

▶ **Observation 8.** *Let g be either a vertical line containing point $(k - 0.5, 0)$ or a horizontal line containing point $(0, k - 0.5)$ for some $k \in \{1, \dots, L\}$. Consider the dissection $D(\boldsymbol{a})$ for a vector $\boldsymbol{a} \in \{0, \dots, L - 1\}^2$ chosen uniformly at random. Then, g is a grid line with respect to $D(\boldsymbol{a})$, and, for every $i \in \{1, \dots, \log(2L)\}$, we have*

$$\Pr_{\boldsymbol{a}}(\textit{the level of } g \textit{ is } i) = \frac{2^{i-1}}{2L - 1}.$$

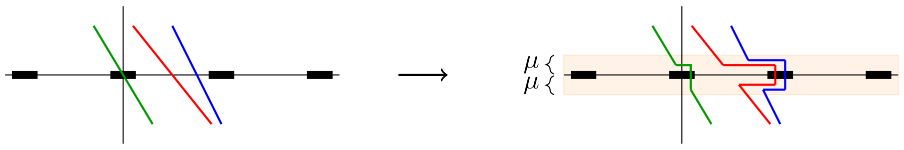

Figure 6 On the left, the red and blue tour cross a boundary outside of a portal and the green tour crosses in an intersection of grid lines. On the right, the tours are modified in a μ-Neighbourhood of the grid line such that they are still non-crossing, only cross boundaries at portals and do not cross in intersections of grid lines.

A *δ-portal* (or, in short, *portal*) on a straight line segment is a subsegment of length $\delta \in (0,1)$. Given a segment s, we define $\mathrm{grid}(s,k,\delta)$ as the set of k equispaced δ-portals on s such that the endpoints of s are contained in the first and last δ-portal respectively.

We place portals on $D(\boldsymbol{a})$ as follows: We will choose a large enough integer $r \in \mathbb{N}$ (called the *portal density factor*) and $\delta > 0$ (the *portal length*) small enough. Then, for every boundary b, we place the portals $\mathrm{grid}(b, r\log L, \delta)$ on b (cf. Figure 5). Note hereby that $\log L \in \mathbb{N}$ because L is a power of two. Observe that, on a deeper level boundary, portals are placed more densely, which will turn out to be a key property.

3.2 Snapping non-crossing curves to portals

In this subsection, we show that disjoint tours can be modified so that they only intersect grid lines in portals, without increasing their lengths too much. To prove this, we follow the same ideas as in [2, Section 2.2]. Nevertheless, the snapping technique in [2, Section 2.2] needs some adaptation to work in the setting of non-crossing curves. This technique was used in previous work for pairs of non-crossing tours [10] and for Steiner trees [5]. Here, we provide a unified framework for this technique, which may be of wider interest and can be applied to a variety of non-crossing Euclidean problems.

In the following, if $s = \overline{\boldsymbol{x}_1\boldsymbol{x}_2}$ is a straight line segment, we let $s^\circ := s \setminus \{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. This allows us to specify more precisely where the segments are allowed to intersect. In particular, if we require that s_1° and s_2° are disjoint for two segments s_1 and s_2, they are allowed to share an endpoint.

▶ **Lemma 9.** *Let $\mathcal{S} = \{s_1, \dots, s_m\}$ be a finite set of straight line segments in the Euclidean plane such that each s_i connects two points in $N_{\frac{1}{4}}\left(\{0,\dots,L\}^2\right)$ and assume $L \geq 4$. Choose a vector $\boldsymbol{a} \in \{0,\dots,L-1\}^2$ uniformly at random and consider the dissection $D(\boldsymbol{a})$. For every portal density factor $r \in \mathbb{N} \setminus \{0\}$, portal length $\delta \in (0,1)$, and $\delta' > 0$, there is a set of curves $\mathcal{S}' = \{s_1', \dots, s_m'\}$ (not necessarily straight line segments) such that*

a) *s_i' differs from s_i only in $N_{\delta'}(G)$ where G denotes the union of all grid lines in $D(\boldsymbol{a})$,*

b) *if the segments s_i° are pairwise disjoint, then the curves $(s_i')^\circ$ are pairwise disjoint as well,*

c) *every s_i' intersects every boundary b of $D(\boldsymbol{a})$ only in the portals $\mathrm{grid}(b, r\log L, \delta)$, i.e., $s_i' \cap b \subseteq \mathrm{grid}(b, r\log L, \delta)$,*

d) *no s_i' contains an intersection point of two grid lines, i.e., $g_1 \cap g_2 \cap s_i' = \emptyset$ for every $i \in \{1,\dots,m\}$ and grid lines $g_1 \neq g_2$,*

e) *the curves of $\mathcal{S}'$ intersect the grid lines in finitely many points,*

f) *$\mathbb{E}_{\boldsymbol{a}}\left[l(\mathcal{S}') - l(\mathcal{S})\right] \leq 7\sqrt{2} \cdot \frac{l(\mathcal{S})}{r}$, where $l(\mathcal{S}) := \sum_{s \in \mathcal{S}} l(s)$.*

Proof sketch. We apply the modifications illustrated in Figure 6 one by one on every boundary. When estimating the cost of these modifications, we make use of the fact that, on a boundary of a deeper level, the portals are placed more densely and, by Observation 8, given

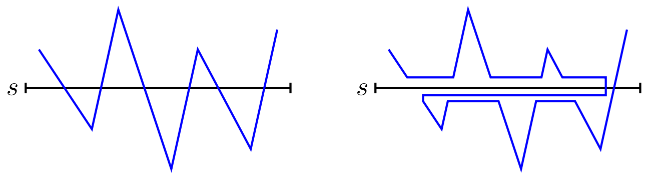

Figure 7 Illustration of the patching scheme in [2] for a single tour: On the left, we are given a tour π intersecting the segment s in six points. On the right, we see how the tour can be modified such that the number of crossings is at most two and the length of is increased at most by $3 \cdot l(s)$.

a fixed line intersected by $\mathcal{S}$, the probability that it is of a shallower level in $D(\boldsymbol{a})$ is small. The total cost of the modifications in Figure 6 depends on the total number of intersection points with grid lines. Therefore, the last step is to relate the number of intersection points to $l(\mathcal{S})$, making use of the fact that $\mathcal{S}$ consists of straight lines segments connecting points close to $\{0,\ldots,L\}^2$. ◀

3.3 The patching technique for three disjoint tours

In the previous section, we have seen that a (reasonable) solution to k-ETSP can be modified such that it only intersects grid lines in portals. In this section, we investigate how the tours can further be modified to reduce the number of intersection points per portal. This will be important for our algorithm because it considers all possible ways that a solution can cross the squares of $D(\boldsymbol{a})$ through the portals. To obtain a reasonable running time, we need a constant bound on the number of crossings. As briefly explained in the introduction, we cannot hope to show this for every solution, even for $k = 3$ (cf. Figure 2). Therefore, we restrict ourselves to 3-ETSP and show the desired properties for this problem under some additional assumptions.

Before delving into the proof, let us introduce some useful notation and establish the prerequisites. Let $\pi_{\mathrm{R}}, \pi_{\mathrm{G}}, \pi_{\mathrm{B}}$ be a solution to an instance of 3-ETSP and s be a straight line segment such that $\pi_c \cap s$ consists of finitely many distinct points for all $c \in \{\mathrm{R},\mathrm{G},\mathrm{B}\}$. Then we say that s is *non-aligned* to the solution and we call the points in $\bigcup_{c\in C} \pi_c \cap s$ *crossings*. We define an order on the crossings by rotating the plane such that s is parallel to the x-axis and then ordering them by their x-coordinates. This allows us to speak of a crossing to be "next to" or "in between" other ones. The *color* of a crossing $\boldsymbol{x}$, denoted $c(\boldsymbol{x})$, is d if $\boldsymbol{x} \subseteq s \cap \pi_d$. With this, we can classify the occurring patterns by sequences of the three colors and we call this a *crossing pattern*. For example, if $x_1,\ldots,x_6$ denote the ordered crossings, by the crossing pattern RRGGBB, we mean that $c(x_1) = c(x_2) = \mathrm{R}$, $c(x_3) = c(x_4) = \mathrm{G}$ and $c(x_5) = c(x_6) = \mathrm{B}$.

Our work builds on existing results for one and two tours. Arora [2] showed that the number of crossings of a single tour with a line segment can be reduced as follows (cf. Figure 7).

▶ **Lemma 10** (Arora [2]). *Let π be a closed curve and s be a non-aligned straight line segment. For every $\delta > 0$, there is a curve π' differing from π only inside $N_\delta(s)$ such that $|s \cap \pi'| \leq 2$ and $l(\pi') \leq l(\pi) + 3 \cdot l(s)$.*

Dross et al. [10] proved that the number of crossings between two disjoint tours and a straight line segment s can be reduced to a constant number, at additional cost $O(l(s))$. Here, we only need the two-color patching schemes for the two special crossing patterns given in the following result. To see why the following lemma holds, one can carefully investigate the proof in [10] or observe that the scheme illustrated in Figure 8 works as desired.

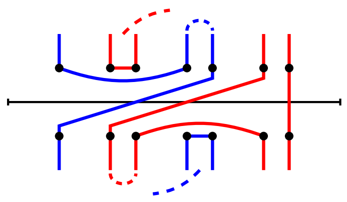

Figure 8 Illustration of the patching scheme for the crossing pattern BRRBBRR, see also [10]. Observe that the connections illustrated by dashed lines must exist (up to symmetry).

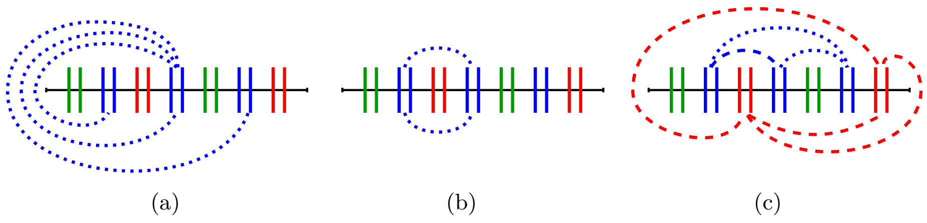

Figure 9 Impossible connections of the blue groups.

▶ **Lemma 11** (Dross et al. [10]). *Let π_c, π_d be disjoint closed curves and s be a non-aligned straight line segment. Assume the crossing pattern is given by cddccdd. For every $\delta > 0$, there are disjoint closed curves π'_c, π'_d differing from π_c and π_d only inside $N_\delta(s)$ such that the new crossing pattern is cdd, and $l(\pi'_c) + l(\pi'_d) \leq l(\pi_c) + l(\pi_d) + 4 \cdot l(s)$.*

Now, we have all the prerequisites in place to give a patching procedure in the case that no red crossing is next to a green crossing on the considered segment (or for any other other choice of two colors), i.e., to prove Lemma 2. For this, we first give a more precise formulation of Lemma 2.

▶ **Lemma 12** (Tricolored Patching). *Let $\pi_\mathrm{R}, \pi_\mathrm{G}, \pi_\mathrm{B}$ be disjoint closed curves and s be a non-aligned straight line segment. Assume that, in the crossing pattern on s, there is no red crossing next to a green crossing. Then, for every $\delta > 0$, there are disjoint closed curves $\pi'_\mathrm{R}, \pi'_\mathrm{B}, \pi'_\mathrm{G}$ such that*

a) *for all $c \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$, π_c differs from π'_c only inside $N_\delta(s)$,*
b) $|(s \cap \pi'_\mathrm{R}) \cup (s \cap \pi'_\mathrm{B}) \cup (s \cap \pi'_\mathrm{G})| \leq 18$,
c) $l(\pi'_\mathrm{R}) + l(\pi'_\mathrm{G}) + l(\pi'_\mathrm{B}) \leq l(\pi_\mathrm{R}) + l(\pi_\mathrm{G}) + l(\pi_\mathrm{B}) + 75 \cdot l(s)$.

Proof sketch. We modify the three curves in several steps to reduce the number of crossings with s. We arrange the crossings into *groups*, where a group is a maximal set of consecutive monochromatic crossings. The road map for our proof is roughly as follows:

First, we reduce the number of crossings inside a group by applying Lemma 10 so that each group consists of either one or two crossings. Next, we show that all green and red groups consist of two crossings, except for possibly the first and last group along the segment s. Then, we further simplify the occurring patterns by applying Lemma 11, eliminating longer sequences where only two of the three colors appear. As a next step, we argue that, after these reductions, it is only left to eliminate crossing patterns of the form $GGB^*RRB^*GGB^*RR$, where $B^* \in \{B, BB\}$.

To eliminate occurrences of this pattern, we will cut the tours open close enough to s and reconnect them. However, we must ensure that this does not disconnect the tours. For this, we investigate how the crossings are connected via the tours outside of $N_\delta(s)$ and we show

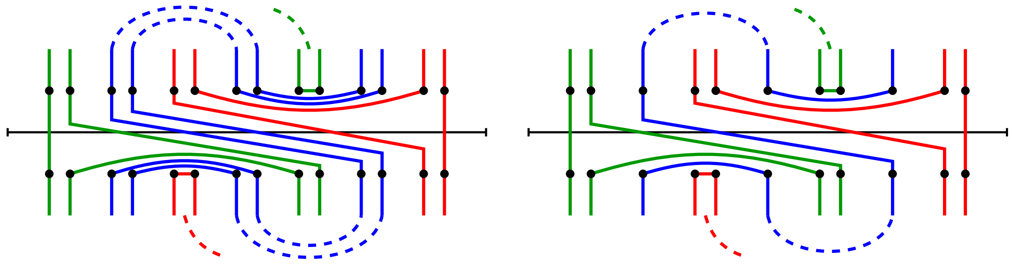

Figure 10 Patching scheme for three non-crossing tours in the two considered crossing patterns.

that the connections illustrated by the dashed lines in Figure 10 must exist (up to symmetry), which is the technically most involved part of our proof. To prove the existence of these connection, we carefully investigate all possible ways that the crossings can be connected. We argue that (some combinations of) the connections illustrated in Figure 9 cannot exist and that this implies the existence of the desired connections.

Then, one can observe that modifying the tours as illustrated in Figure 10 gives three closed curves with a reduced number of crossings with s. We apply this modification repeatedly whenever the crossing pattern $GGB^*RRB^*GGB^*RR$ ($B^* \in \{B, BB\}$) appears while simultaneously redirecting the new crossings towards the right endpoint of s. This will ensure that we do not occur an additional cost of $O(l(s))$ for every application of the scheme, but rather a cost of $O(l(s_i))$ for some almost non-overlapping subsegements s_i of s.

The last step is to argue that every crossing pattern that does not contain the described subpatterns consists of at most 18 crossings. ◀

3.4 Structure theorem for three non-crossing tours

Now, we have all the prerequisites in place to state and prove our structure theorem (cf. Theorem 4) for three non-crossing tours.

For this, given an instance of k-ETSP with terminals in $\{0, \ldots, L\}^2$ and a shift vector $\boldsymbol{a} \in \{0, \ldots, L-1\}^2$, we say that a solution $\Pi = (\pi_c)_{c \in C}$ is (r, m, δ)*-portal respecting* if, for every boundary b in $D(\boldsymbol{a})$, the intersection points $b \cap \bigcup_{c \in C} \pi_c$ are contained in the the portals $\text{grid}(b, r \log L, \delta)$ and every portal is intersected in at most m points in total.

▶ **Theorem 13** (Structure Theorem for 3-ETSP). *Let an instance of* 3-ETSP *with* $T \subseteq \{0, \ldots, L\}^2$ *and* $\varepsilon > 0$ *be given. Then there exists a shift vector* $\boldsymbol{a} \in \{0, \ldots, L-1\}^2$ *and* $\delta > 0$ *such that there is either a* $(\lceil (15\sqrt{2} + 4)/\varepsilon \rceil, 18, \delta)$*-portal respecting solution of cost at most* $(1 + \varepsilon) \cdot \text{OPT}^*$*, or there is a* $(\lceil (15\sqrt{2} + 4)/\varepsilon \rceil, 18, \delta)$*-portal respecting two-tour presolution* (π_1, π_2) *with* $2l(\pi_1) + l(\pi_2) \le \left(\frac{5}{3} + \frac{\varepsilon}{2}\right) \cdot \text{OPT}^*$*.*

Proof sketch. First, we move all crossings to portals by applying Lemma 9. Then, we consider boundaries b one by one in non-decreasing order of their levels and apply patching (Lemma 12). Note that patching can create new crossings on another boundary perpendicular to b. We argue that such a boundary is of a deeper level so that it has not been modified yet. Since we can choose the portal-length δ arbitrarily small, the total cost of patching is negligible. ◀

Recall that, given a two-tour presolution, one can "double" one of the tours to obtain an induced two-tour solution for 3-ETSP (cf. Observation 6). Applying this to a $(\lceil(15\sqrt{2}+4)/\varepsilon\rceil, 18, \delta)$-portal respecting two-tour presolution obtained from Theorem 13, this gives a $(\lceil(15\sqrt{2}+4)/\varepsilon\rceil, 36, \delta)$-portal respecting solution for 3-ETSP. Combining this with Theorem 13, we obtain the following.

▶ **Corollary 14.** *For every instance of* 3-ETSP *with terminals in* $\{0,\ldots,L\}^2$ *and* $\varepsilon > 0$*, there is a shift vector* $\boldsymbol{a} \in \{0,\ldots,L-1\}^2$ *and* $\delta > 0$ *such that there exists a* $(\lceil(15\sqrt{2}+4)/\varepsilon\rceil, 36, \delta)$*-portal respecting solution of cost at most* $\left(\frac{5}{3}+\varepsilon\right) \cdot \textsc{Opt}^*$*.*

4 A dynamic programming algorithm

In the previous section, we have seen that there is a $\left(\frac{5}{3}+\varepsilon\right)$-approximate portal-respecting solution. In this section, we give a polynomial-time algorithm that computes an "optimal" (in the sense of Theorem 5) portal-respecting solution.

Our algorithm is based on the same ideas as Arora's dynamic programming algorithm for Euclidean TSP [2] and the algorithm by Dross et al. for 2-ETSP [10]. The difference to our work is that we need to solve a more general problem: First, we allow for any fixed number of colors of terminals and search for non-crossing tours. Second, we have weighted colors, i.e., the tours of different colors contribute differently to the total cost.

More precisely, by k-ETSP$'$ we denote the following problem: a set C of k colors is given together with an integer L that is a power of two. The input consists of a set of terminals $T_c \subseteq \{0,\ldots,L\}^2$ for each color $c \in C$, a color weight $w_c \geq 0$ for each $c \in C$, a shift vector $\boldsymbol{a} \in \{0,\ldots,L-1\}^2$, $\delta > 0$ (sufficiently small) and two integers $r, m \in \mathbb{N}$. We consider the dissection $D(\boldsymbol{a})$ and, as before, we place $r \log L$ portals on every boundary. A *solution* to k-ETSP$'$ is a k-tuple of tours $\Pi = (\pi_c)_{c\in C}$ such that every terminal is visited by the tour of the same color (i.e., $T_c \subseteq \pi_c$ for every $c \in C$), the tours are pairwise disjoint, and (r, m, δ)-portal respecting. The *cost of a solution* for k-ETSP$'$ is then $l(\Pi) := \sum_{c\in C} w_c \cdot l(\pi_c)$. Similarly as for k-ETSP (cf. Figure 1), a solution minimizing the cost does not necessarily exist. By $\textsc{Opt}^* := \inf\{l(\Pi)) : \Pi \text{ is a solution}\}$, we denote the value that we want to approximate.

The aim of this section is to prove the following theorem.

▶ **Theorem 15.** *There is an algorithm that computes a parametric solution* $\Pi(\lambda)$ *to* k-ETSP$'$ *in time* $L^{O(mr\log k)}$ *such that* $\lim_{\lambda\to 0} l(\Pi(\lambda)) = \textsc{Opt}^*$*.*

Proof sketch. The main idea is to use dynamic programming. More precisely, we consider the nodes of $D(\boldsymbol{a})$ (i.e., squares in $\mathbb{R}^2$) one by one from leaves to the root and compute all possible ways that a solution can cross the border of the square (the so-called *multipath problem*). For a non-leaf node, we will find these possibilities by combining the solutions for the four subsquares. ◀

5 Perturbation

In the previous section, we have focused on solving 3-ETSP when the terminals have integer coordinates. In this section, we show how an input for general 3-ETSP can be preprocessed such that we only have to solve an instance with terminals in $\mathbb{Z}^2$ and how a solution to the preprocessed instance can be transformed back into a solution to the original instance. For this, we use similar ideas as in [2]. Note that, if the terminal sets of different colors are in some sense far away from each other, we can find a tour for each color separately such that they are are disjoint. More precisely, we call an instance to 3-ETSP *ε-reducible*, if there is a choice of the three colors $\{c, c', c''\} = \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ such that, when applying the algorithm in [10] on $T_c, T_{c'}$ (with the given ε), and, independently applying Arora's algorithm [2] on $T_{c''}$, the resulting tours are disjoint and, therefore, provide a $(1+\varepsilon)$-approximation for 3-ETSP. For this reason, we restrict ourselves to instances that are non-reducible.

▶ **Theorem 16.** *Let $\varepsilon > 0$ and $\mathcal{I}$ be a non-ε-reducible instance of* 3-ETSP*.*

a) *There is an algorithm* Perturbation *that has running time $O(n^2)$ and returns an instance $\mathcal{I}'$ of* 3-ETSP *with terminals in $\{0, \dots, L\}^2$ where $L = O(n/\varepsilon)$ is a power of two.*

b) *There is an algorithm* Back-Perturbation *that has running time $O(n^2)$ and, given a $(1+\varepsilon')$-approximate solution to the instance $\mathcal{I}'$, returns a $(1+\varepsilon'+O(\varepsilon))$-approximate solution to the instance $\mathcal{I}$.*

Proof sketch. The main idea for the algorithm Perturbation is the following: Choose a square S of minimum size containing all terminals and place a grid in S with $O(n/\varepsilon)$ many grid lines. Then, snap each terminal to a close enough intersection point of these grid lines without moving two terminals of different colors to the same point.

For Back-Perturbation, include a doubled straight line segment between terminals of the original instance and the solution to the perturbed instance. If such a segment crosses the other two tours, apply patching and then replace each crossing by a detour around the segment. ◀

6 A $(\frac{5}{3}+\varepsilon)$-approximation algorithm for 3-ETSP

We have all the prerequisites in place to prove our main result. We begin by recalling the theorem.

▶ **Theorem 1** (restated)**.** *For every $\varepsilon > 0$, there is an algorithm that computes a $\left(\frac{5}{3}+\varepsilon\right)$-approximation for* 3-ETSP *in time $(\frac{n}{\varepsilon})^{O(1/\varepsilon)}$.*

Proof sketch. It is straightforward how to check whether an instance is ε-reducible and how we can find a solution in that case. Therefore, assume we are given a non-reducible instance. First, we apply algorithm Perturbation to the instance. By Corollary 14, there is a portal-respecting solution to the perturbed instance $\mathcal{I}'$ that gives a $\left(\frac{5}{3}+\varepsilon'\right)$-approximation (where we need to choose ε' to be ε divided by a large enough constant). Therefore, we can apply Theorem 15 to compute a $\left(\frac{5}{3}+\varepsilon'\right)$-approximate portal-respecting solution to $\mathcal{I}'$. Last, we apply Back-Perturbation to the solution. ◀

References

1 Basak Alper, Nathalie Henry Riche, Gonzalo Ramos, and Mary Czerwinski. Design study of LineSets, a novel set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, 2011.

2 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. `doi:10.1145/290179.290180`.

3 Yair Bartal and Lee-Ad Gottlieb. A linear time approximation scheme for Euclidean TSP. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 698–706, 2013. `doi:10.1109/FOCS.2013.80`.

4 Yair Bartal, Lee-Ad Gottlieb, and Robert Krauthgamer. The traveling salesman problem: Low-dimensionality implies a polynomial time approximation scheme. *SIAM Journal on Computing*, 45(4):1563–1581, 2016.

5 Sergey Bereg, Krzysztof Fleszar, Philipp Kindermann, Sergey Pupyrev, Joachim Spoerhase, and Alexander Wolff. Colored non-crossing euclidean steiner forest. In *Proceedings of the 26th International Symposium on Algorithms and Computation (ISAAC)*, pages 429–441, 2015.

6 Paul B. Callahan and S. Rao Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 42(1):67–90, 1995.

7 Thom Castermans, Mereke van Garderen, Wouter Meulemans, Martin Nöllenburg, and Xiaoru Yuan. Short plane supports for spatial hypergraphs. *Journal of Graph Algorithms and Applications*, 23(3):463–498, 2019.

8 Kamalika Chaudhuri, Brighten Godfrey, Satish Rao, and Kunal Talwar. Paths, trees, and minimum latency tours. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 36–45, 2003.

9 Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, volume 3122, pages 72–83, 2004.

10 François Dross, Krzysztof Fleszar, Karol Wegrzycki, and Anna Zych-Pawlewicz. Gap-ETH-tight approximation schemes for red-green-blue separation and bicolored noncrossing Euclidean travelling salesman tours. In *Proceedings of the 34nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1433–1463, 2023.

11 Alon Efrat, Yifan Hu, Stephen Kobourov, and Sergey Pupyrev. Mapsets: Visualizing embedded and clustered graphs. *Journal on Graph Algorithms and Applications*, 19(2):571–593, 2015.

12 Jeff Erickson and Amir Nayyeri. Shortest non-crossing walks in the plane. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 297–308, 2011.

13 Jittat Fakcharoenphol, Chris Harrelson, and Satish Rao. The k-traveling repairmen problem. *ACM Transactions on Algorithms*, 3(4):40, 2007.

14 Lee-Ad Gottlieb. A light metric spanner. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 759–772, 2015.

15 Joachim Gudmundsson and Christos Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nordic Journal of Computing*, 6(4):469, 1999.

16 Ferran Hurtado, Matias Korman, Marc J. van Kreveld, Maarten Löffler, Vera Sacristán, Akiyoshi Shioura, Rodrigo I. Silveira, Bettina Speckmann, and Takeshi Tokuyama. Colored spanning graphs for set visualization. *Computational Geometry*, 68:262–276, 2018.

17 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A (slightly) improved approximation algorithm for metric TSP. In *Proceedings of the 53rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 32–45, 2021.

18 Anna R. Karlin, Nathan Klein, and Shayan Oveis Gharan. A deterministic better-than-3/2 approximation algorithm for metric TSP. In *Proceedings of the 24th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 13904, pages 261–274, 2023.

19 Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8):1665–1677, 2015.

20 Sándor Kisfaludi-Bak, Jesper Nederlof, and Karol Wegrzycki. A Gap-ETH-Tight Approximation Scheme for Euclidean TSP. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 351–362. IEEE, 2022.

21 Yoshiyuki Kusakari, Daisuke Masubuchi, and Takao Nishizeki. Finding a noncrossing Steiner forest in plane graphs under a 2-face condition. *Journal of Combinatorial Optimization*, 5(2):249–266, 2001.

22 Cristian S. Mata and Joseph S. B. Mitchell. Approximation algorithms for geometric tour and network design problems (extended abstract). In *Proceedings of the 11th Annual Symposium on Computational Geometry (SCG)*, pages 360–369, 1995.

23 Joseph S. B. Mitchell. A constant-factor approximation algorithm for TSP with pairwise-disjoint connected neighborhoods in the plane. In *Proceedings of the 26th Annual Symposium on Computational Geometry (SCG)*, pages 183–191, 2010.

24 Joseph S. B. Mitchell. Shortest paths and networks. In *Handbook of Discrete and Computational Geometry, Third Edition*, chapter 31. CRC Press LLC, 3 edition, 2017.

25 Christos H. Papadimitriou. The Euclidean Traveling Salesman Problem is NP-Complete. *Theoretical Computer Science*, 4(3):237–244, 1977.

26 Satish Rao and Warren D. Smith. Approximating Geometrical Graphs via "Spanners" and "Banyans". In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 540–550, 1998.

27 Iris Reinbacher, Marc Benkert, Marc van Kreveld, Joseph S. B. Mitchell, Jack Snoeyink, and Alexander Wolff. Delineating boundaries for imprecise regions. *Algorithmica*, 50(3):386–414, 2008.

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

29 J. Takahashi, H. Suzuki, and T. Nishizeki. Finding shortest non-crossing rectilinear paths in plane regions. In *International Symposium on Algorithms and Computation*, pages 98–107. Springer, 1993.

30 Jun-Ya Takahashi, Hitoshi Suzuki, and Takao Nishizeki. Algorithms for finding non-crossing paths with minimum total length in plane graphs. In *Proceedings of the 3rd International Symposium on Algorithms and Computation (ISAAC)*, pages 400–409. Springer, 1992.