# On Euclidean Vehicle Routing with Allocation

Jan Remy  Reto Spöhel*  Andreas Weißl

Institute of Theoretical Computer Science
ETH Zürich, 8092 Zürich, Switzerland
{jremy|rspoehel|aweissl}@inf.ethz.ch

Abstract. The (Euclidean) Vehicle Routing Allocation Problem (VRAP) is a generalization of Euclidean TSP. We do not require that all points lie on the salesman tour. However, points that do not lie on the tour are allocated, i.e., they are directly connected to the nearest tour point, paying a higher (per-unit) cost. More formally, the input is a set of $n$ points $P \subset \mathbb{R}^d$ and functions $\alpha : P \to [0, \infty)$ and $\beta : P \to [1, \infty)$. We wish to compute a subset $T \subseteq P$ and a salesman tour $\pi$ through $T$ such that the total length of the tour plus the total allocation cost is minimum. The allocation cost for a single point $p \in P \setminus T$ is $\alpha(p) + \beta(p) \cdot d(p, q)$, where $q \in T$ is the nearest point on the tour. We give a PTAS with complexity $\mathcal{O}\left(n \log^{d+3} n\right)$ for this problem. Moreover, we propose an $\mathcal{O}(n \operatorname{polylog}(n))$-time PTAS for the Steiner variant of this problem. This dramatically improves a recent result of Armon *et al.* [3].

## 1. Introduction

Let $P \subset \mathbb{R}^2$ denote a set of $n$ points in the plane, and let penalty functions $\alpha : P \to [0, \infty)$ and $\beta : P \to [1, \infty)$ be given. (In fact, the 1 can be replaced by any fixed $\beta_{\min} > 0$, where this constant is understood as part of the problem definition and not of the input instance. For simplicity we set $\beta_{\min} = 1$ throughout. Moreover, we shall see that if $\beta(p) > 2$ for some point $p \in P$, the precise value of $\beta(p)$ is irrelevant and we might as well set $\beta(p) = \infty$.) A solution to the (Euclidean) Vehicle Routing Allocation Problem (VRAP) is a subset of *tour points* $T \subseteq P$ and a tour $\pi$ through $T$. Each *allocation point* $p \in A := P \setminus T$ is allocated to the nearest tour point $q \in T$ at a cost of $\alpha(p) + \beta(p) \cdot d(p, q)$. We wish to minimize the length of the tour plus the total allocation cost, i.e., we minimize

$$\operatorname{val}(T, \pi) = \sum_{\{p,q\} \in \pi} d(p, q) + \sum_{p \in A} \left( \alpha(p) + \beta(p) \min_{q \in T} d(p, q) \right).$$

Throughout, let $T^* \subseteq P$ and $\pi^*$ denote an optimal choice for $T$ and $\pi$, i.e., $\operatorname{val}(T^*, \pi^*)$ is minimum.

VRAP is motivated by vehicle routing. For instance, each point represents a bank and we wish to transport cash to the banks using an armored vehicle. The vehicle can visit each bank (which would be a shortest salesman tour), but it might be cheaper to visit only some of the banks while the staff of the other banks have to pick up the cash at the visited banks. Although the total distance is smaller, this way of cash transportation is more risky and needs additional insurance (which can be modeled using the functions $\alpha$ and $\beta$).

Observe that VRAP becomes the well-known Euclidean traveling salesman problem (TSP) if we have $\beta(p) > 2$ for all $p \in P$, since by the triangle inequality it is always cheaper to include a given point on the tour than to allocate it. As VRAP includes TSP as a special case, we know that VRAP is $\mathcal{NP}$-hard, even in the strong sense (cf. [9]). Given the $\mathcal{NP}$-hardness of the problem,

---

*The author was supported by Swiss National Science Foundation, grant 200021-108158.
A preliminary version of this paper appeared in the proceedings of the 10th International Workshop on Algorithms and Data Structures (WADS'07) [14].

approximation algorithms are of interest. A *polynomial-time approximation scheme* (PTAS) is an algorithm that for any fixed $\varepsilon > 0$ approximates the optimum within a factor of $(1+\varepsilon)$ in time polynomial in $n$. Note that the complexity of a PTAS might be exponential in $1/\varepsilon$.

In this paper we show that VRAP admits a PTAS. As our main result, we propose a randomized nearly-linear time approximation scheme.

**Theorem 1.** *There is a randomized PTAS for* VRAP *with time complexity* $\mathcal{O}\left(n \log^5 n\right)$.

Moreover, we consider the problem Steiner VRAP where we are allowed to include additional points on the tour and allocate to these in order to further reduce the cost. A solution $(T, S, \pi)$ to Steiner VRAP consists of point sets $T \subseteq P$ and $S \subset \mathbb{R}^2$, and a salesman tour $\pi$ through $T \dot\cup S$. With $A := P \setminus T$ as before, we wish to find $T^*$, $S^*$ and $\pi^*$ minimizing

$$\mathrm{val}^\bullet(T, S, \pi) = \sum_{\{p,q\}\in\pi} d(p,q) + \sum_{p\in A}\left(\alpha(p) + \beta(p) \min_{q\in T\dot\cup S} d(p,q)\right). \tag{1}$$

**Theorem 2.** *There is a randomized PTAS for* Steiner VRAP *with time complexity* $\mathcal{O}\left(n \log^{\mathcal{O}(1/\varepsilon)} n\right)$.

Theorem 2 improves a recent result of Armon *et al.* [3], where the authors give a (deterministic) PTAS with complexity $\mathcal{O}\left(n^{\mathcal{O}(1/\varepsilon)}\right)$ for a problem called Purchase Cooperative TSP, which is the special case of Steiner VRAP where $\alpha(p) = 0$ and $\beta(p) = 1$ for all points $p \in P$. Their algorithm seems to extend to Steiner VRAP as long as $\beta(p) = \beta(q)$ for all $p, q \in P$.

Both our algorithms extend to the case when $P \subset \mathbb{R}^d$ for some fixed dimension $d$.

**Theorem 3.** *In any fixed dimension $d$, there is a randomized PTAS for* VRAP *with time complexity* $\mathcal{O}\left(n \log^{d+3} n\right)$.

**Theorem 4.** *In any fixed dimension $d$, there is a randomized PTAS for* Steiner VRAP *with time complexity* $\mathcal{O}\left(n \log^{\xi(d,\varepsilon)} n\right)$*, where* $\xi(d,\varepsilon) = \mathcal{O}(\sqrt{d}/\varepsilon)^{d-1}$.

Moreover, both algorithms can be derandomized, increasing their complexity to $\mathcal{O}\left(n^{d+1} \log^{d+3} n\right)$ and $\mathcal{O}\left(n^{d+1} \log^{\xi(d,\varepsilon)} n\right)$, respectively. Lastly, if $\beta(p) = \beta(q)$ for all $p, q \in P$, the running time of our PTAS for VRAP can be reduced by a factor of $\mathcal{O}(\log n)$, yielding a complexity of $\mathcal{O}\left(n \log^4 n\right)$ for the two-dimensional case.

**Our methods.** Essentially, we prove Theorem 1 by combining the *adaptive dissection* technique due to Kolliopoulos and Rao [11] with dynamic programming on *r-vapid graphs*, as proposed by Rao and Smith [13].

The adaptive dissection technique is used for estimating allocation costs. Its main advantage over the well-known quad tree based methods introduced by Arora [4] is that it allows us to work with only a *constant* (instead of logarithmic) number of portals per rectangle. This improvement is achieved by two key ideas: On the one hand, the location of the tour points is guessed by dynamic programming, and if their bounding box is small, the *zoom tree* – which replaces the quad tree – zooms directly into the 'region of interest', potentially skipping many levels in between. On the other hand, in the resulting near-optimal portal-respecting solution, a point is not necessarily allocated to its nearest point, but possibly to a different nearby tour point. This added flexibility turns out to be of advantage. It is worth pointing out that – in contrast to Arora's technique – in the adaptive dissection framework it is necessary to allocate many points simultaneously to the same tour point, since allocating them individually would be too time-consuming. This can be done using range searching techniques (see e.g. [1]).

To estimate the tour length, we transfer ideas presented in [13] for Euclidean TSP from the quad tree setting to the zoom tree setting. To compute a Euclidean spanner quickly, we use the algorithm by Gudmundsson *et al.* [10].

In order to prove Theorem 2, we make use of a relatively simple geometric observation and employ standard quad tree techniques developed in [4] and [5].

**Related Work.** It is well-known that Euclidean TSP admits a PTAS [4, 12], even one with complexity $\mathcal{O}(n \log n)$ [13]. A slightly more general version of VRAP was introduced in 1996 by Beasley and Nascimento [7] as a unifying framework for various network problems (where the input is a weighted graph instead of points in the plane). In more recent literature, the network version of the problem is usually called Median Cycle or Ring Star. References can be found in [16]. Applications of VRAP in bookmobile routing [8] and grass-mower scheduling [15] have been reported.

As mentioned above, a related problem called Purchase Cooperative TSP was recently studied by Armon *et al.* [3] in both the network and Euclidean setting. Using methods by Arora [4], they proposed a PTAS with complexity $\mathcal{O}\left(n^{\mathcal{O}(1/\varepsilon)}\right)$ for this problem. In addition, they studied several variants of the problem. As those variants are quite different from VRAP and Steiner VRAP, we refer to [3] for details.

**Organization of this Paper.** In Section 2 we show that it is sufficient to consider instances with input points that lie on an $\mathcal{O}(n/\varepsilon) \times \mathcal{O}(n/\varepsilon)$ integer grid. In Section 3 we introduce the concepts of *zoom trees* and *portal-respecting distances*, following Kolliopoulos and Rao [11]. In Section 4, we adapt the notion of *$r$-vapid graphs* due to Rao and Smith [13] to our purposes. In Section 5 we describe and analyze our PTAS for VRAP, and in Section 6 we outline how to improve the PTAS for Steiner VRAP proposed by Armon *et al.* [3]. In closing, we discuss the generalization to higher dimensions and explain how our algorithms can be derandomized in Section 7.

## 2. Perturbation

In this section we argue that we may restrict our attention to instances in which the input points have odd integral coordinates and the side length of the bounding box is a power of 2 and of order $n/\varepsilon$. We start with a simple but (throughout this paper) important observation: every input point $p$ with $\beta(p) > 2$ is a tour point in every optimal solution due to the triangle inequality. With this fact in hand, we can prove the following statement.

**Lemma 5.** *To prove Theorem* 1*, it suffices to consider only instances with* $P \subseteq \{1, 3, 5, \ldots, L-1\}^2$*, where* $L = 2^\tau$ *for the smallest* $\tau$ *such that* $2^\tau \geq 30n/\varepsilon$*.*

*Proof.* First, observe that we may assume w.l.o.g. that $P \subseteq [0, C]^2$ for an appropriate $C$, and that there exist points $(0, y_1) \in P$ and $(C, y_2) \in P$. Moreover, rescaling the coordinates by any factor does not change the structure of an optimal solution to $P$. Hence, for any $L > 0$, we may assume that $P \subseteq [0, L]^2$ and, since $\beta(p) \geq 1$ for all $p \in P$, also that any solution for $P$ has cost at least $L$.

Now fix $L$ as in the lemma, and construct a point set $P'$ from $P$ by moving every point in $P$ to the closest point with odd integer coordinates. Let $(T^*, \pi^*)$ denote an optimal solution for $P$ and $(T, \pi)$ an optimal solution for the modified point set $P'$. As the distance between a point in $P$ and its copy in $P'$ is at most $\sqrt{2}$, all distances increase at most by $2\sqrt{2}$. Moreover, since in the optimal solution, only points $p$ with $\beta(p) \leq 2$ are allocated, the cost increases by at most $2\sqrt{2}$ per tour edge and $4\sqrt{2}$ per allocation edge. In total, $\mathrm{val}(T^*, \pi^*)$ increases by at most $4\sqrt{2}n \leq 6n$, since the number of edges in $(T^*, \pi^*)$ is at most $n$. Thus, we have

$$\mathrm{val}(T, \pi) \leq \mathrm{val}(T^*, \pi^*) + 6n.$$

Similarly, we can easily transform an optimal solution $(T', \pi')$ for $P'$ into a solution $(\widetilde{T}, \widetilde{\pi})$ for $P$ such that

$$\mathrm{val}\left(\widetilde{T}, \widetilde{\pi}\right) \leq \mathrm{val}\left(T', \pi'\right) + 6n.$$

Applying a PTAS with $\varepsilon' = \varepsilon/2$ to the shifted instance $P'$, we obtain a solution $(T', \pi')$ satisfying

$$\mathrm{val}\left(T', \pi'\right) \leq (1 + \varepsilon/2)\, \mathrm{val}\left(T, \pi\right).$$

Assuming that $\varepsilon \leq 1$, it follows for the corresponding solution $(\widetilde{T}, \widetilde{\pi})$ of the original problem $P$ that

$$\begin{aligned} \mathrm{val}\left(\widetilde{T}, \widetilde{\pi}\right) &\leq (1 + \varepsilon/2)\left(\mathrm{val}\left(T^*, \pi^*\right) + 6n\right) + 6n \\ &\leq (1 + \varepsilon/2)\, \mathrm{val}\left(T^*, \pi^*\right) + 15n \\ &\leq (1 + \varepsilon)\, \mathrm{val}\left(T^*, \pi^*\right), \end{aligned}$$

where the last inequality holds due to $\mathrm{val}\left(T^*, \pi^*\right) \geq L \geq 30n/\varepsilon$. As the perturbation described above can be accomplished in linear time, the existence of a randomized $\mathcal{O}\left(n \log^5 n\right)$ time PTAS for shifted instances implies Theorem 1. □

We henceforth assume that $P \subseteq \{1, 3, \ldots, L-1\}^2$, where $L = \mathcal{O}\left(n/\varepsilon\right)$ is a power of two. Those assumptions will be crucial to the proofs.

## 3. Zoom Trees and Portal-Respecting Allocations

In this section we mainly simplify concepts appearing in [11], defining a certain distance measure between an allocated point $p$ and the set of tour points $T$. This distance measure is defined with respect to a dissection tree, called a *zoom tree*, which adapts to a given solution to VRAP. The main result of this section is that, in expectation, this distance measure approximates the real allocation costs quite closely.

3.1. **Concepts and Results.** For fixed $a, b \in \{0, 2, ..., L-2\}$, let $G_{a,b}(i)$ denote a grid of granularity $2^i$ with origin $(a, b)$, i.e., the vertical and horizontal grid lines have coordinates $\left\{a + j2^i : j \in \mathbb{Z}\right\}$ and $\left\{b + j2^i : j \in \mathbb{Z}\right\}$, respectively. Let $i_0$ denote the smallest integer such that $L = 2^{i_0} \geq 30n/\varepsilon$, and let $\mathfrak{Q}_0 := (a, b) + [-L, L]^2$ denote the square of side length $2L$ with center $(a, b)$. Note that $P \subset \mathfrak{Q}_0$. For any rectangle $\mathfrak{R} \subset \mathbb{R}^2$, we use $|\mathfrak{R}|$ to denote the length of one of the longer sides of $\mathfrak{R}$. We also refer to $|\mathfrak{R}|$ as the *side length* of $\mathfrak{R}$. By $\partial\mathfrak{R}$ we denote the boundary of $\mathfrak{R}$.

For $1 \leq i \leq i_0$, a rectangle $\mathfrak{R}$ is said to be *$i$-allowable* if and only if it satisfies the following properties.

- $\mathfrak{R}$ lies in $\mathfrak{Q}_0$ and is bounded by lines of $G_{a,b}(i)$.
- If $i \geq 2$ then $7 \cdot 2^i \leq |\mathfrak{R}| < 7 \cdot 2^{i+1}$.
  If $i = 1$ then $|\mathfrak{R}| < 7 \cdot 2^{i+1} = 28$.

We say that $i$ is the *level* of $\mathfrak{R}$. Note that $|\mathfrak{R}| = \Theta(2^i)$. $\mathfrak{R}$ is said to be *allowable* if there exists an $i$, $1 \leq i \leq i_0$, such that $\mathfrak{R}$ is $i$-allowable.

**Observation 6.** *The aspect ratio of an allowable rectangle is bounded by* 14*, and the (non-empty) intersection of two allowable rectangles is an allowable rectangle.*

Moreover, we have the following lemma, which will be useful when arguing about running times.

**Lemma 7.**

*(i) There are $\mathcal{O}\left(n \log n\right)$ allowable rectangles that contain at least one point of $P$.*

*(ii) There are $\mathcal{O}\left(n \log^2 n\right)$ pairs of allowable rectangles $(\mathfrak{R}', \mathfrak{R})$ such that $\mathfrak{R}'$ contains at least one point of $P$ and $\mathfrak{R}' \subset \mathfrak{R}$.*

*Proof.* For a fixed $i$, any of the $n$ input points is contained in a constant number of $i$-allowable rectangles, as such rectangles are bounded by lines of $G_{a,b}(i)$ and have bounded aspect ratio. Thus (i) follows from the fact that $i \leq \log L = \mathcal{O}(\log n)$.

By the same argument, if $\mathfrak{R}'$ contains at least one point there are $\mathcal{O}(\log n)$ allowable rectangles $\mathfrak{R}$ containing $\mathfrak{R}'$. As there are $\mathcal{O}(n \log n)$ choices for $\mathfrak{R}'$, (ii) follows. $\square$

Next, we introduce a dissection tree that adapts to a given solution to VRAP. The idea is to subdivide $\mathfrak{Q}_0$ recursively by alternately splitting the current rectangle and zooming into the 'area of interest'. We call such a subdivision a *zoom tree*. In principle, a zoom tree $ZT_{a,b}$ is defined with respect to $a$, $b$ and any fixed subset $T \subseteq P$. However, in this section, as well as in the other analytical parts of this paper, we only consider the zoom tree corresponding to the set $T^* \subseteq P$ of tour points in the optimal solution (of course, the actual algorithm does not know this set in advance and will have to guess $T^*$ and, therefore, also the structure of the zoom tree considered here). The root of $ZT_{a,b}$ is $\mathfrak{Q}_0$, and the nodes of $ZT_{a,b}$ are the allowable rectangles recursively obtained from the following parent-child relations (see Fig. 1 for an illustration). For every rectangle in $ZT_{a,b}$ we either say that it is *split* or that it is *zoomed*, depending on how it is obtained from its parent. A zoomed rectangle will be split in the next step, and a split rectangle will be zoomed in the next step.

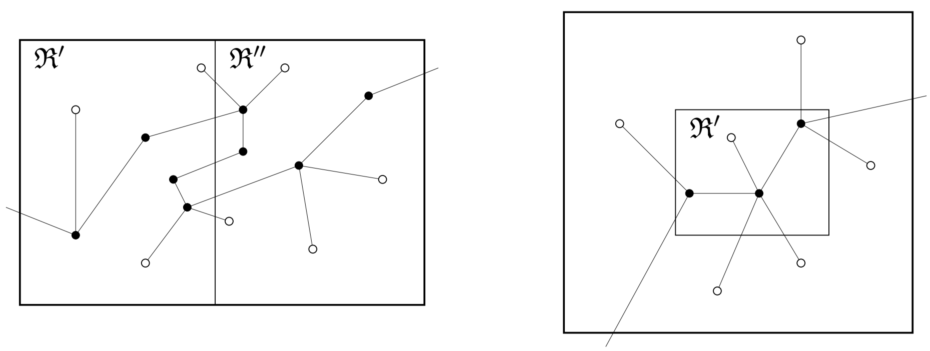


FIGURE 1. If the rectangle $\mathfrak{R}$ is zoomed, we subdivide it into two children $\mathfrak{R}'$ and $\mathfrak{R}''$ (left). Otherwise, if $\mathfrak{R}$ is split, its child $\mathfrak{R}'$ is a small allowable rectangle that contains all points on the salesman tour (right).

If an $i$-allowable rectangle $\mathfrak{R} \in ZT_{a,b}$ is *zoomed*, we obtain its two children $\mathfrak{R}'$ and $\mathfrak{R}''$ by cutting $\mathfrak{R}$ parallel to its shorter side along the line of $G_{a,b}(i)$ that minimizes $|\text{area}(\mathfrak{R}') - \text{area}(\mathfrak{R}'')|$ (that is, we aim to nearly bisect $\mathfrak{R}$). If this cut is not unique we prefer the leftmost (bottommost) one. We call the line $C$ along which we split $\mathfrak{R}$ the *cutting line*. It is easily seen that the two rectangles we obtain are $j$-allowable for some $j \in \{i-2, i-1, i\}$. $\mathfrak{R}'$ and $\mathfrak{R}''$ are split rectangles (as they are obtained by splitting $\mathfrak{R}$).

A *split* rectangle $\mathfrak{R}$ has only one child $\mathfrak{R}'$, which is constructed as follows: consider the minimal rectangle $B$ containing all points in $\mathfrak{R} \cap T^*$. (We will see that a split rectangle always contains a tour point, cf. Lemma 15.) For any rectangle $\widehat{\mathfrak{R}}$, let $d(\partial\widehat{\mathfrak{R}}, \partial B)$ denote the distance between the boundaries of $\widehat{\mathfrak{R}}$ and $B$, and choose $\widetilde{\mathfrak{R}}$ as the allowable rectangle with smallest circumference such that $d(\partial\widetilde{\mathfrak{R}}, \partial B) \geq |B|/4$. If this does not uniquely define $\widetilde{\mathfrak{R}}$, choose the left- and bottommost candidate. Let $\mathfrak{R}' := \mathfrak{R} \cap \widetilde{\mathfrak{R}}$. As the intersection of two allowable rectangles, the resulting rectangle $\mathfrak{R}'$ is allowable. $\mathfrak{R}'$ is a zoomed rectangle.

We stop the subdivision process at $\mathfrak{R}$ if $\mathfrak{R}$ is either 1-allowable or contains at most one point of $P$. Such rectangles become *leaves* of $ZT_{a,b}$. We define $\mathfrak{Q}_0$ to be split, so that the first dissection step is a zoom step. For an allocation point $p \in P \setminus T^*$ and a tourpoint $q \in T^*$, we say that the rectangle $\mathfrak{R} \in ZT_{a,b}$ *separates* $p$ and $q$ if and only if it is the rectangle in $ZT_{a,b}$ closest to the root such that

- either $\mathfrak{R}$ is split and $p \in \mathfrak{R}$ and $q \notin \mathfrak{R}$,
- or $\mathfrak{R}$ is zoomed and $p \notin \mathfrak{R}$ and $q \in \mathfrak{R}$.

One easily checks that this uniquely defines $\mathfrak{R}$.

In the following, we will introduce the concept of *portal-respecting allocations.* For the time being, the parameter $m \in \mathbb{N}$ is an arbitrary number; we will specify it according to our purposes later (cf. Lemma 9 below). For a given (allowable) rectangle $\mathfrak{R}$, we place a point on each corner and $m-1$ equidistant points subdividing each side. We call these points *portals* and denote by $G_{\text{alloc}} = G_{\text{alloc}}(\mathfrak{R})$ the set of portals on $\partial\mathfrak{R}$. The *portal-respecting distance* $d_{\mathfrak{R}}(p,q)$ between $p \in \mathfrak{R}$ and $q \notin \mathfrak{R}$ is defined as

$$d_{\mathfrak{R}}(p,q) := \min_{g \in G_{\text{alloc}}(\mathfrak{R})} d(p,g) + d(g,q).$$

In other words, we detour the line segment $pq$ over the nearest portal on $\partial\mathfrak{R}$.

The next lemma gives an easy bound on the difference between the Euclidean distance and the portal-respecting distance with respect to some rectangle $\mathfrak{R}$.

**Lemma 8.** *For any rectangle $\mathfrak{R}$ and points $p \in \mathfrak{R}$ and $q \notin \mathfrak{R}$, we have*

$$d_{\mathfrak{R}}(p,q) - d(p,q) \leq \frac{|\mathfrak{R}|}{m}.$$

*Proof.* Let $x \in \partial\mathfrak{R}$ denote the point where the straight line from $p$ to $q$ crosses the border of $\mathfrak{R}$, and let $g \in G_{\text{alloc}}(\mathfrak{R})$ denote the portal which is closest to $x$. As the portals are equidistant, we have $d(x,g) \leq |\mathfrak{R}|/(2m)$. By the triangle inequality, we can bound the portal-respecting distance by

$$d(p,x) + d(x,q) + 2d(x,g) \leq d(p,q) + |\mathfrak{R}|/m.$$

□

The *portal-respecting distance* $d_{ZT_{a,b}}(p,q)$ between $p \in P \setminus T^*$ and $q \in T^*$ is the portal-respecting distance w.r.t. the rectangle $\mathfrak{R} \in ZT_{a,b}$ that separates $p$ and $q$. The main result of this section is the next lemma, which appears in similar form already in [11]. It asserts that a *constant* number $m$ of portals per rectangle suffices to guarantee that, in expectation, the portal-respecting distances are good estimates for the real distances. Here we denote by $d_{ZT_{a,b}}(p,T^*)$ the infimum over all portal-respecting distances $d_{ZT_{a,b}}(p,q)$, $q \in T^*$.

**Lemma 9.** *For given $\varepsilon > 0$, there exists $m = m(\varepsilon)$ such that for every allocation point $p \in P \setminus T^*$ and for $a$ and $b$ uniformly at random from $\{0,2,\ldots,L-2\}$, we have*

$$\mathbb{E}\left[d_{ZT_{a,b}}(p,T^*)\right] \leq (1+\varepsilon)d(p,T^*).$$

In fact, our proof yields the following more general statement, which may seem strange at the moment but will be needed in Section 5. Essentially it states that Lemma 9 still holds if an arbitrary constant factor is inserted in Lemma 8. We will need this because our algorithm introduces several other errors of order $\mathcal{O}(|\mathfrak{R}|/m)$ when estimating allocation costs.

**Lemma 10.** *Let $d_*(p,q)$ be any distance measure which overestimates the Euclidean distance from a given allocation point $p$ to a given tour point $q$ by an absolute error of order $\mathcal{O}(|\mathfrak{R}|/m)$ for the* *rectangle $\mathfrak{R}$ separating $p$ and $q$. Then for given $\varepsilon > 0$, there exists $m = m(\varepsilon)$ such that for every allocation point $p \in P \setminus T^*$ and for $a$ and $b$ uniformly at random from $\{0, 2, \ldots, L-2\}$, we have*

$$\mathbb{E}\left[d_*(p, T^*)\right] \leq (1+\varepsilon) d(p, T^*).$$

The remainder of this section is devoted to the proof of Lemma 9.

3.2. **Proof of Lemma 9.** It suffices to prove that for a fixed allocation point $p$, the random variable

$$\Delta_p := d_{ZT_{a,b}}(p, T^*) - d(p, T^*)$$

satisfies

$$\mathbb{E}\left[\Delta_p\right] = \mathcal{O}\left(\log m/m\right) d(p, T^*).$$

Lemma 9 then follows by choosing $m$ appropriately.

For ease of readability, the proof is divided into three parts. We start by collecting a couple of simple but useful geometrical facts about dissection trees and portal-respecting allocations. We proceed by defining events used in a rather lengthy case distinction, and proving some technical statements about the probabilities involved. Finally, we put everything together in the main part of the proof, the actual case distinction.

3.2.1. *Geometrical Facts.* As mentioned above, we begin with some geometric observations. We start with a very simple fact about how good the dissection along $i$-allowable lines in split steps approximates an exact bisection.

**Lemma 11.** *Let $\mathfrak{R}$ denote a zoomed rectangle of level $i \geq 2$, and assume that $\mathfrak{R}$ is split along a line $C$ into its two children. The distance between $C$ and any of the two shorter sides of $\mathfrak{R}$ is at least* $(3/7)|\mathfrak{R}|$.

*Proof.* The line $C'$ cutting $\mathfrak{R}$ exactly in half has distance $|\mathfrak{R}|/2$ from any shortest side of $\mathfrak{R}$ but does not belong to $G_{a,b}(i)$ in general. However, there is always a line $C$ on $G_{a,b}(i)$ within distance $2^i/2$ of $C'$. Thus, $C$ is at least at distance

$$\frac{|\mathfrak{R}|}{2} - \frac{2^i}{2} \geq \frac{|\mathfrak{R}|}{2} - \frac{|\mathfrak{R}|}{14} = \frac{3}{7}|\mathfrak{R}|$$

from any of the two shorter sides of $\mathfrak{R}$. □

From Lemma 11, we immediately obtain the next technical observation.

**Lemma 12.** *Let $\mathfrak{R}_1$ and $\mathfrak{R}_2$ be two zoomed rectangles such that $\mathfrak{R}_2$ is a descendant of $\mathfrak{R}_1$. Let $C_1$ and $C_2$ denote the cutting lines splitting $\mathfrak{R}_1$ and $\mathfrak{R}_2$, respectively, and assume that they are parallel to each other. Then we have $|\mathfrak{R}_2| \leq (7/3) d(C_1, C_2)$.*

*Proof.* Clearly, the distance from $C_2$ to $C_1$ is bounded from below by the distance from $C_2$ to one of the shorter sides of $\mathfrak{R}_2$. Hence, by Lemma 11 we have

$$d(C_1, C_2) \geq \frac{3}{7}|\mathfrak{R}_2|,$$

which implies the claim. □

The next two statements are concerned with the geometric properties of zoom steps.

**Lemma 13.** *Let $\mathfrak{R}$ denote a zoomed rectangle of level $i \geq 2$. Then each side of $\mathfrak{R}$ has a tour point within distance at most* $(11/28)|\mathfrak{R}|$.

*Proof.* Let $\mathfrak{R}_0$ denote the parent of $\mathfrak{R}$. Recall the construction of $\mathfrak{R}$: we consider the minimal rectangle $B$ that contains $\mathfrak{R}_0 \cap T^*$ and choose $\widetilde{\mathfrak{R}}$ as the smallest allowable rectangle containing $B$ such that $d(\partial\widetilde{\mathfrak{R}}, \partial B) \geq |B|/4$. Then $\mathfrak{R} = \mathfrak{R}_0 \cap \widetilde{\mathfrak{R}}$. By construction, every side of $B$ contains at least one tour point. If $\mathfrak{R}$ is $i$-allowable we thus have a tour point within distance at most $|B|/4 + 2^i \leq |\mathfrak{R}|/4 + |\mathfrak{R}|/7 = (11/28)|\mathfrak{R}|$ of each side of $\mathfrak{R}$. □

**Lemma 14.** *Let $p$ be an allocation point and $q$ a tour point, and assume that $\mathfrak{R} \in ZT^*_{a,b}$ is a zoomed rectangle of level $i \geq 2$ separating $p$ and $q$. Then we have $|\mathfrak{R}| \leq 9d(p,q)$.*

*Proof.* We use the same notation as in the proof of Lemma 13. Let $j \geq i \geq 2$ denote the level of $\widetilde{\mathfrak{R}}$. As we assumed that $\mathfrak{R}$ is the (first!) rectangle separating $p$ and $q$, the point $p$ is in $\mathfrak{R}_0 \setminus \mathfrak{R}$ and thus not in $\widetilde{\mathfrak{R}}$. As on the other hand $q \in B$, we have $d(p,q) \geq |B|/4$. By construction, the corresponding sides of $\widetilde{\mathfrak{R}}$ and $B$ are within distance at most $|B|/4 + 2^j$. Hence, we obtain

$$|\widetilde{\mathfrak{R}}| \leq (6/4)|B| + 2 \cdot 2^j \leq 6d(p,q) + (2/7)|\widetilde{\mathfrak{R}}|,$$

and thus $|\mathfrak{R}| \leq |\widetilde{\mathfrak{R}}| \leq 7/5 \cdot 6d(p,q) \leq 9d(p,q)$. □

In combination, Lemma 11 and Lemma 13 yield the following observation about the structure of $ZT_{a,b}$.

**Lemma 15.** *Every rectangle of $ZT_{a,b}$ contains at least one point of $T^*$. In particular, the zoom step is well-defined.*

*Proof.* Since we stop the subdivision at 1-allowable rectangles, the parent $\mathfrak{R}_0$ of any split rectangle has at least level 2. By Lemma 13, there is a tour point within distance $(11/28)|\mathfrak{R}_0|$ of all sides of $\mathfrak{R}_0$. Since by Lemma 11 the line $C$ cutting $\mathfrak{R}_0$ has distance at least $(3/7)|\mathfrak{R}_0| > (11/28)|\mathfrak{R}_0|$ from the shorter sides of $\mathfrak{R}_0$, it follows that every split rectangle contains a tour point. Therefore, the zoom step is well-defined, and it follows that also every zoomed rectangle contains at least one tour point. □

3.2.2. *Events and Probabilities.* Let $q \in T^*$ denote the tour point closest to $p$, i.e., $p$ is allocated to $q$ in the optimal solution. In the following, we use the notation $x := d(p,q)$ and let the random variable $I$ denote the level of the rectangle $\mathfrak{R}$ separating $p$ and $q$. Thus we have $|\mathfrak{R}| = \Theta(2^I)$.

As the structure of the zoom tree $ZT_{a,b}$ changes with the choice of $a$ and $b$, we have to distinguish several cases. Clearly, as long as $I \leq \log x + c$ for some constant $c$, Lemma 8 immediately guarantees that

$$\Delta_p \leq d_{\mathfrak{R}}(p,q) - d(p,q) \leq \frac{|\mathfrak{R}|}{m} = \mathcal{O}(x/m).$$

This already settles the case for the event

$$\mathcal{E}_0 := \{I \leq \log x + 2\}.$$

Formally, we have

$$\mathbb{E}[\Delta_p \mid \mathcal{E}_0] \Pr\{\mathcal{E}_0\} = \mathcal{O}(x/m). \tag{2}$$

For the rest of this proof, we focus on the event $\overline{\mathcal{E}_0}$. Note that we have $I > \log x + 2 \geq 2$ in this case, and thus the geometrical lemmas of the previous section are applicable to $\mathfrak{R}$ and its ancestors.

The next statement is crucial to our analysis and is illustrated in Figure 2.

**Lemma 16.** *If $\overline{\mathcal{E}_0}$ occurs, then $\mathfrak{R}$ is split and the following is true. Let $C$ denote the cutting line separating $p$ from $q$, let $S$ denote the line through $p$ parallel to $C$, and let $S^-$ denote the halfspace not containing $q$ and $C$. Then $S^-$ contains at least one tour point and we have $I \geq \log y - 5$, where $y := d(p,r)$ denotes the distance to the* closest *such tour point $r$.*

*Proof.* If $\mathfrak{R}$ is zoomed, we have by Lemma 14 that $7 \cdot 2^I \leq |\mathfrak{R}| < 14x$, i.e., $I < \log x + 1$, contradicting the assumption that $\overline{\mathcal{E}_0}$ occurred. Hence $\mathfrak{R}$ is split. Without loss of generality, we assume that $C$ is vertical, and that $q$ is to the right of $p$. We argue by contradiction. If the point $r$ exists but $I < \log y - 5$, we have that $|\mathfrak{R}| < 7y \cdot 2^{-4} \leq y/2$, which implies that there is no tour point inside $\mathfrak{R}$ and to the left of $p$. If $r$ does not exist, this is clear anyway. Either way, we arrive at the following contradiction. Since, by Lemma 13, there is a tour point within distance at most $(11/28)|\mathfrak{R}|$ from the left border of $\mathfrak{R}$, it follows that $p$ is within distance $(11/28)|\mathfrak{R}|$ of the left border of $\mathfrak{R}$. Since, by Lemma 11, the cutting line $C$ has distance at least $(3/7)|\mathfrak{R}|$ from the left border, we obtain that $x \geq (3/7 - 11/28)|\mathfrak{R}| = |\mathfrak{R}|/28 \geq 2^I/4$, i.e., $I \leq \log x + 2$, contradicting the assumption that $\overline{\mathcal{E}_0}$ occurred. ☐

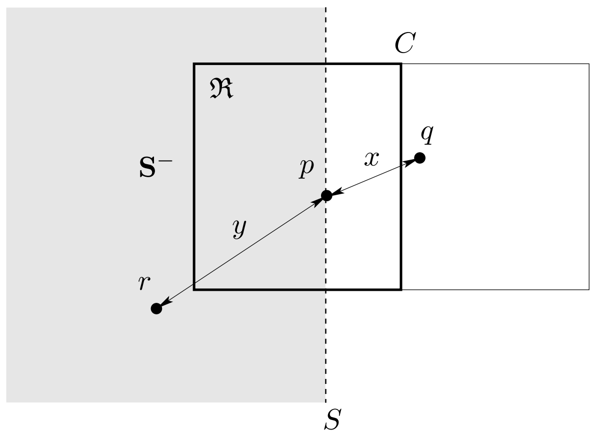


FIGURE 2. Sketch to Lemma 16.

In the remainder of this proof, we stick to the notations $r$ for the closest tour point to $p$ on the 'other side' than $q$, and $y = d(p, r)$ for its distance from $p$. Recall that $\mathfrak{R}$ and $I$ denote the rectangle separating $p$ and $q$ and its level, respectively. Analogously, let $\mathfrak{R}'$ denote the rectangle separating $p$ and $r$, and $J$ its level. By $C$ we denote the cutting line separating $p$ and $q$ (which coincides with one of the sides of $\mathfrak{R}$), and by $C'$ analogously the cutting line separating $p$ and $r$ (if $\mathfrak{R}'$ is split).

Note that in this setup, both $p$ and $q$ (and thus $x$) are fixed, but $r$ (and thus $y$) as well as $\mathfrak{R}$ and $\mathfrak{R}'$ (and thus $I$ and $J$) are indeed random variables with respect to the random choice of $(a, b) \in \{0, 2, \ldots, L-2\}^2$. Observe that $r$ is fixed once we have the orientation of $C$. Now, let $\mathcal{E}_v \subseteq \overline{\mathcal{E}_0}$ denote the event that $C$ is vertical, and let $r_v$ denote the corresponding value of the point $r$. Analogously, let $\mathcal{E}_h \subseteq \overline{\mathcal{E}_0}$ denote the event that $C$ is horizontal, and let $r_h$ denote the corresponding value of $r$. In the following, we assume that $\mathcal{E}_v$ occurs and assume in addition that $q$ is to the right and $r = r_v$ to the left of $p$. The other cases are symmetric. Note that $y = d(p, r_v) =: y_v$ is fixed on $\mathcal{E}_v$. We consider the two main events

$$\mathcal{E}_A := \mathcal{E}_v \cap \{\text{The zoom tree } ZT_{a,b} \text{ separates first } p \text{ and } r_v\text{, then } p \text{ and } q\},$$
$$\mathcal{E}_B := \mathcal{E}_v \cap \{\text{The zoom tree } ZT_{a,b} \text{ separates first } p \text{ and } q\text{, then } p \text{ and } r_v\}$$

and also the events

$$\mathcal{E}_1 := \mathcal{E}_v \cap \{\mathfrak{R}' \text{ is zoomed}\}$$
$$\mathcal{E}_2 := \mathcal{E}_v \cap \{\mathfrak{R}' \text{ is split, } C \text{ and } C' \text{ are parallel}\}$$
$$\mathcal{E}_3 := \mathcal{E}_v \cap \{\mathfrak{R}' \text{ is split, } C \text{ and } C' \text{ are orthogonal}\}.$$

Note that $ZT_{a,b}$ cannot separate $p$ from both $q$ and $r$ simultaneously, as $q$ is separated from $p$ by a vertical cutting line to the right of $p$, and $r$ is to the left of $p$. Throughout, we use the notations $\mathcal{E}_{A1} := \mathcal{E}_A \cap \mathcal{E}_1$, $\Pr_{A1}\{\dots\} = \Pr\{\dots \mid \mathcal{E}_{A1}\}$, $\mathbb{E}_{A1}[\dots] = \mathbb{E}[\dots \mid \mathcal{E}_{A1}]$, and so on.

For any two points $p_1, p_2 \in P$ and $1 \le i \le i_0$, let $\mathcal{V}(p_1p_2, i)$ and $\mathcal{H}(p_1p_2, i)$ denote the events that the line segment $p_1p_2$ is cut by a vertical, respectively horizontal, $i$-allowable line.

To calculate $\Pr\{\mathcal{V}(pq, i)\}$, we observe that the line segment $pq$ is cut by at most $x/2 + 1$ vertical lines of the grid $G_{0,0}(1)$. For each of these lines, the probability that it is in $G_{a,b}(i)$ is $2^{-(i-1)}$. As $p$ and $q$ have odd integer coordinates, we have $x \ge 2$ and thus

$$\Pr\{\mathcal{V}(pq, i)\} \le (x/2 + 1) \cdot 2^{-(i-1)} = \mathcal{O}\left(x/2^i\right),$$

and similarly for $\Pr\{\mathcal{H}(pq, i)\}$. It follows that

$$\Pr\{I = i\} \le \Pr\{\mathcal{V}(pq, i) \cup \mathcal{H}(pq, i)\} = \mathcal{O}\left(x/2^i\right)$$

and by summing up also

$$\Pr\{I \ge i\} = \mathcal{O}\left(x/2^i\right). \tag{3}$$

We will mostly use these bounds in the form

$$\Pr\{\, I = i \mid \mathcal{E}\} \Pr\{\mathcal{E}\} = \Pr\{\{I = i\} \cap \mathcal{E}\} = \mathcal{O}\left(x/2^i\right), \tag{4}$$

and

$$\Pr\{\, I \ge i \mid \mathcal{E}\} \Pr\{\mathcal{E}\} = \mathcal{O}\left(x/2^i\right), \tag{5}$$

for some event $\mathcal{E}$. By analogous arguments one obtains for any $\mathcal{E} \subseteq \mathcal{E}_v$ that

$$\Pr\{\, J = j \mid \mathcal{E}\} \Pr\{\mathcal{E}\} = \mathcal{O}\left(y_v/2^j\right).$$

The events $\{I = i\}$ and $\{J = j\}$ are not independent. However, conditioning on $\mathcal{E}_3$, i.e., $C$ and $C'$ being orthogonal, they can be shown to behave as if they were independent: we can bound them by $\mathcal{V}(pq, i)$ and $\mathcal{H}(pr_v, j)$, respectively, which are independent in the unconditioned probability space. Formally, we have for any event $\mathcal{E} \subseteq \mathcal{E}_3$ that

$$\begin{aligned} \Pr\{\, I = i, J = j \mid \mathcal{E}\} \Pr\{\mathcal{E}\} &\le \Pr\{\, \mathcal{V}(pq, i) \cap \mathcal{H}(pr_v, j) \mid \mathcal{E}\} \Pr\{\mathcal{E}\} \\ &\le \Pr\{\mathcal{V}(pq, i) \cap \mathcal{H}(pr_v, j)\} \\ &= \Pr\{\mathcal{V}(pq, i)\} \Pr\{\mathcal{H}(pr_v, j)\} \\ &= \mathcal{O}\left(xy_v/2^{i+j}\right). \end{aligned} \tag{6}$$

3.2.3. *Case Distinction.* We now consider the subcases of the case that $\overline{\mathcal{E}_0}$ occurs one by one. Recall that by Lemma 16, we have $I \ge \log y - 5$ in all these cases. By (3), this implies in particular that

$$\Pr\{\mathcal{E}_v\} = \mathcal{O}\left(x/y_v\right). \tag{7}$$

**Case A. $ZT_{a,b}$ separates first $p$ and $r_v$, then $p$ and $q$.** This is the easier case, because here we can guarantee that the portal-respecting distance $d_{ZT_{a,b}}(p, q) = d_{\mathfrak{R}}(p, q)$ does not overestimate the actual distance $d(p, q)$ by too much. (In Case B we will have to use in addition that $d_{ZT_{a,b}}(p, T^*) \le d_{\mathfrak{R}'}(p, r_v)$.) This allows us to work with the bound

$$\Delta_p \le d_{\mathfrak{R}}(p, q) - d(p, q) \le |\mathfrak{R}|/m \tag{8}$$

guaranteed by Lemma 8. We have to consider several subcases, which are illustrated in Figure 3.

*Case A1. $\mathfrak{R}'$ is zoomed.* This case cannot occur, as $\mathfrak{R}'$ then contains by construction both $r_v$ and $q$, and thus also separates $p$ from $q$. However, we assumed that $\overline{\mathcal{E}_0}$ occurred, which by Lemma 16 in particular implies that $p$ and $q$ are separated by a split rectangle.

*Case A2. $\mathfrak{R}'$ is split, $C$ and $C'$ are parallel.* As the two cutting lines are parallel and within distance at most $x + y_v \leq 2y_v$, we obtain by Lemma 12 that $|\mathfrak{R}| = \mathcal{O}(y_v)$. Therefore, by (8), $\Delta_p$ is at most $\mathcal{O}(y_v/m)$, and we have with (7) that

$$\mathbb{E}_{A2}[\Delta_p] \Pr\{\mathcal{E}_{A2}\} = \mathcal{O}(y_v/m)\, \mathcal{O}(x/y_v) = \mathcal{O}(x/m)\,. \tag{9}$$

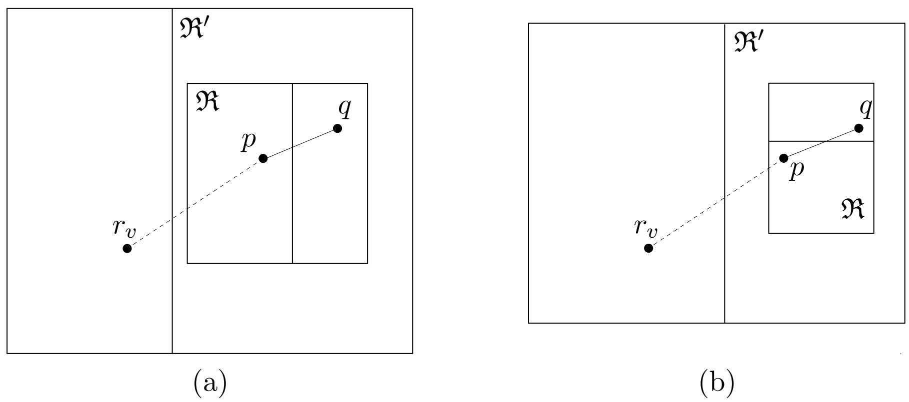


FIGURE 3. Sketches of the situation we have in case A2 (a) and case A3 (b). Note that $p$, $q$ and $r_v$ are fixed, but $\mathfrak{R}$ and $\mathfrak{R}'$ depend on the choice of $a$ and $b$.

*Case A3. $\mathfrak{R}'$ is split, $C$ and $C'$ are orthogonal.* Here we have no guarantee that $|\mathfrak{R}| = \mathcal{O}(y_v)$. Due to $\mathfrak{R}' \supset \mathfrak{R}$ and Lemma 16, we have $J \geq I \geq \log y_v - 5$. Hence, we obtain with (6) that

$$\begin{aligned}
\mathbb{E}_{A3}[\Delta_p] \Pr\{\mathcal{E}_{A3}\} &= \sum_{j \geq \log y_v - 5} \sum_{i=\lceil \log y_v - 5\rceil}^{j} \mathbb{E}_{A3}[\Delta_p \mid I = i, J = j] \cdot \Pr_{A3}\{I = i, J = j\} \Pr\{\mathcal{E}_{A3}\} \\
&= \sum_{j \geq \log y_v - 5} \sum_{i=\lceil \log y_v - 5\rceil}^{j} \mathcal{O}\left(2^i/m\right) \cdot \mathcal{O}\left(x y_v / 2^{i+j}\right) \\
&= \mathcal{O}(x/m) \sum_{j \geq \log y_v - 5} (j - \log y_v + 6) \cdot \mathcal{O}\left(y_v/2^j\right) \\
&= \mathcal{O}(x/m)\,.
\end{aligned} \tag{10}$$

**Case B. $ZT_{a,b}$ separates first $p$ and $q$, then $p$ and $r_v$.** As already mentioned, our strategy here is to let $p$ be allocated to $r_v$ instead of $q$ if the portal-respecting distance to $q$ becomes too large. By Lemma 8, we have

$$\begin{aligned}
\Delta_p &\leq \min\{d_{\mathfrak{R}}(p,q) - d(p,q), d_{\mathfrak{R}'}(p, r_v)\} \\
&\leq \min\left\{|\mathfrak{R}|/m,\, y_v + |\mathfrak{R}'|/m\right\}.
\end{aligned} \tag{11}$$

Recall that $|\mathfrak{R}| = \Theta(2^I)$ and $|\mathfrak{R}'| = \Theta(2^J)$. Again we have three subcases, which are illustrated in Figure 4.

*Case B1. $\mathfrak{R}'$ is zoomed.* By Lemma 14 we have that $|\mathfrak{R}'| = \mathcal{O}(y_v)$, as $\mathfrak{R}'$ is zoomed and separates $p$ and $r_v$ (if $\mathfrak{R}'$ has level 1, this is clear anyway). We deduce from (11) that

$$\Delta_p \leq \min\{\mathcal{O}\left(2^I/m\right), \mathcal{O}(y_v)\}$$

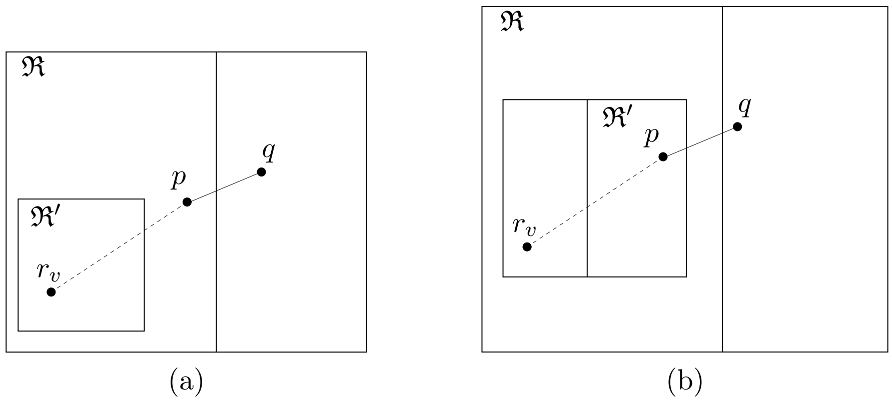


FIGURE 4. Sketches of the situation we have in case B1 (a) and case B2 (b). Case B3 is similar to B2; only the orientation of the line separating $p$ and $r_v$ is horizontal.

and thus with (4) and (5) that

$$\begin{aligned}\mathbb{E}_{B1}[\Delta_p]\Pr\{\mathcal{E}_{B1}\} &= \sum_{i=\lceil\log y_v-5\rceil}^{\lfloor\log(my_v)\rfloor}\mathbb{E}_{B1}[\Delta_p \mid I=i]\Pr_{B1}\{I=i\}\Pr\{\mathcal{E}_{B1}\}\\ &\quad+\mathbb{E}_{B1}[\Delta_p \mid I>\log(my_v)]\Pr_{B1}\{I>\log(my_v)\}\Pr\{\mathcal{E}_{B1}\}\\ &= \sum_{i=\lceil\log y_v-5\rceil}^{\lfloor\log(my_v)\rfloor}\mathcal{O}\left(2^i/m\right)\cdot\mathcal{O}\left(x/2^i\right)+\mathcal{O}\left(y_v\right)\cdot\mathcal{O}\left(x/(my_v)\right)\\ &= \mathcal{O}\left(\log m\right)\cdot\mathcal{O}\left(x/m\right)+\mathcal{O}\left(x/m\right)\\ &= \mathcal{O}\left(x\log m/m\right).\end{aligned} \tag{12}$$

*Case B2. $\mathfrak{R}'$ is split, $C$ and $C'$ are parallel.* Similarly to Case A2, we obtain from Lemma 12 that $|\mathfrak{R}'| = \mathcal{O}(y_v)$, and it follows with the same calculation as in Case B1 that

$$\mathbb{E}_{B2}[\Delta_p]\Pr\{\mathcal{E}_{B2}\} = \mathcal{O}\left(x\log m/m\right). \tag{13}$$

*Case B3. $\mathfrak{R}'$ is split, $C$ and $C'$ are orthogonal.* In this case, we have no guarantee that $|\mathfrak{R}'| = \mathcal{O}(y_v)$. On the contrary: as the line $pr_v$ is inside the parent $\mathfrak{R}_0$ of $\mathfrak{R}'$, we have that $y_v \le 2|\mathfrak{R}_0| \le 6|\mathfrak{R}'| < 6\cdot 7\cdot 2^{J+1}$ (cf. Lemma 11) and, therefore, $J \ge \log y_v - 7$.

Thus, we have $I \ge J \ge \log y_v - 7$ and obtain from (11) that

$$\Delta_p \le \min\{\mathcal{O}\left(2^I/m\right), \mathcal{O}\left(2^J\right)\}.$$

We obtain with (4) and (6), that

$$\begin{aligned}
\mathbb{E}_{B3}\left[\Delta_p\right]\Pr\left\{\mathcal{E}_{B3}\right\} &= \sum_{i=\lceil \log y_v - 7\rceil}^{\lfloor \log(my_v)\rfloor} \mathbb{E}_{B3}\left[\Delta_p \mid I=i\right]\Pr_{B3}\left\{I=i\right\}\Pr\left\{\mathcal{E}_{B3}\right\} \\
&\quad + \sum_{i>\log(my_v)}\sum_{j=\lceil \log y_v - 7\rceil}^{i} \mathbb{E}_{B3}\left[\Delta_p \mid I=i, J=j\right]\Pr_{B3}\left\{I=i, J=j\right\}\Pr\left\{\mathcal{E}_{B3}\right\} \\
&= \mathcal{O}\left(x\log m/m\right) + \sum_{i>\log(my_v)}\sum_{j=\lceil \log y_v - 7\rceil}^{i} \mathcal{O}\left(2^j\right)\cdot\mathcal{O}\left(xy_v/2^{i+j}\right) \\
&= \mathcal{O}\left(x\log m/m\right) + \mathcal{O}\left(x\right)\sum_{i>\log(my_v)}\left(i-\log y_v+8\right)\cdot\mathcal{O}\left(y_v/2^i\right) \\
&= \mathcal{O}\left(x\log m/m\right) + \mathcal{O}\left(x\right)\cdot\mathcal{O}\left(\log m/m\right) \\
&= \mathcal{O}\left(x\log m/m\right),
\end{aligned} \tag{14}$$

where the first sum is dealt with exactly as in the previous cases, using only the bound $\Delta_p = \mathcal{O}\left(2^I/m\right)$.

With the law of total probability we obtain from (2), (9), (10), (12), (13), and (14) that

$$\mathbb{E}\left[\Delta_p\right] = \mathcal{O}\left(x\log m/m\right),$$

and Lemma 9 follows after choosing $m$ appropriately.

## 4. VRAP on Straight-Line Graphs

The results in this section extend work by Rao and Smith [13] to the adaptive dissection setting. In the following, a *straight-line graph* on some point set $\widehat{P} \subseteq \mathbb{R}^2$ is a graph on vertex set $\widehat{P}$ whose edges are identified with straight line segments connecting two points of $\widehat{P}$ in the obvious way. Using an algorithm due to Gudmundsson *et al.* [10], we can quickly compute a straight-line graph $\mathcal{S}'$ on a superset of $P$ which has few 'relevant' crossings with the allowable rectangles introduced in the previous section, and such that there exists an expected nearly-optimal tour through $\mathcal{S}'$. This will allow us to quickly find such a tour by dynamic programming.

4.1. **Concepts and Results.** In the following it is of advantage to look at a solution $(T,\pi)$ from a slightly different viewpoint. Recall that $\mathfrak{Q}_0$ has side length $2L = \mathcal{O}\left(n\right)$ and its center at $(a,b)$, where $a,b \in \{0,2,\ldots,L-2\}$ uniformly at random. For any connected straight-line graph (SLG) $\mathcal{G}$ on a vertex set $P' \supseteq P$, we denote the induced shortest path metric by $d_{\mathcal{G}}(\cdot,\cdot)$. For a given solution $(T,\pi)$, let

$$\operatorname{val}_{\mathcal{G}}\left(T,\pi\right) = \sum_{\{p,q\}\in\pi} d_{\mathcal{G}}(p,q) + \sum_{p\in A}\left(\alpha(p) + \beta(p)\min_{q\in T} d(p,q)\right). \tag{15}$$

Note that only the length of the tour is measured in the shortest path metric. Every solution to VRAP gives rise to a closed walk $\mathcal{W} = \mathcal{W}(T,\pi)$ formed by the shortest paths between subsequent tour points. In principle, this walk may include non-tour points, but in order to minimize (15), it is always better to 'pick up' such points and include them in $T$. Thus, a solution minimizing (15) can be described as a walk $\mathcal{W}$ through $\mathcal{G}$, where by definition the tourpoints are exactly the points

$T_{\mathcal{W}} := P \cap \mathcal{W}$ on the walk, and the remaining points $A_{\mathcal{W}} := P \setminus T_{\mathcal{W}}$ are allocated. Denoting the entire length of the walk $\mathcal{W}$ by $\ell(\mathcal{W})$, we can rewrite (15) as

$$\mathrm{val}_{\mathcal{G}}(\mathcal{W}) = \ell(\mathcal{W}) + \sum_{p \in A_{\mathcal{W}}} \left( \alpha(p) + \beta(p) \min_{q \in T_{\mathcal{W}}} d(p,q) \right). \tag{16}$$

Note that an edge contributes $s$ times to $\ell(\mathcal{W})$ if we traverse it $s$ times on the walk. In the following, we denote the problem of finding a walk that minimizes (16) by VRAP($\mathcal{G}$).

We say that a crossing of an edge $e$ of a SLG $\mathcal{G}$ and a rectangle $\mathfrak{R}$ is *relevant* if $e$ intersects $\partial\mathfrak{R}$ and exactly one endpoint of $e$ is within $\mathfrak{R}$. The graph $\mathcal{G}$ is said to be *$r$-sparse* if any allowable rectangle $\mathfrak{R}$ has at most $r$ relevant crossings with edges of $\mathcal{G}$. Note that it depends on the choice of $a$ and $b$ whether a fixed SLG $\mathcal{G}$ is $r$-sparse or not. The main result of this section is the next lemma.

**Lemma 17.** *Let $T^*$ denote the set of tour points of the optimal solution to VRAP. For any given $\varepsilon > 0$, there exists $r = r(\varepsilon)$ such that for all choices of $a, b \in \{0, 2, \ldots, L-2\}$, one can compute in $\mathcal{O}(n \log^2 n)$ time a point set $S$ and an $r$-sparse SLG $\mathcal{S}'$ on the point set $P' = P \dot\cup S$ satisfying the following: If $a$ and $b$ are chosen uniformly at random, the shortest walk $\mathcal{W}^*$ on $\mathcal{S}'$ visiting all points of $T^*$ has expected length*

$$\mathbb{E}\left[\ell(\mathcal{W}^*)\right] \le \sum_{\{p,q\} \in \pi^*} d(p,q) + \varepsilon \, \mathrm{val}(T^*, \pi^*).$$

*Moreover, $\mathcal{W}^*$ uses no edge of $\mathcal{S}'$ more than twice.*

Since $T_{\mathcal{W}^*} \supseteq T^*$, the allocation costs in $\mathrm{val}_{\mathcal{S}'}(\mathcal{W}^*)$ do not exceed those in $\mathrm{val}(T^*, \pi^*)$, and thus Lemma 17 immediately implies that

$$\mathbb{E}\left[\mathrm{val}_{\mathcal{S}'}(\mathcal{W}^*)\right] \le (1+\varepsilon)\, \mathrm{val}(T^*, \pi^*).$$

Together with $\mathrm{val}(T^*, \pi^*) \le \mathrm{val}_{\mathcal{S}'}(\mathcal{W}^*)$, it follows that $\mathcal{W}^*$ induces an expected nearly-optimal solution to VRAP. The rest of this section is devoted to the proof of Lemma 17.

4.2. **Proof of Lemma 17.** In order to prove this statement, we need to review two important concepts. Firstly, a SLG $\mathcal{S}$ on $P$ is a *$(1+\varepsilon)$-spanner* if for all $p, q \in P$, we have $d_{\mathcal{S}}(p,q) \le (1+\varepsilon) \cdot d(p,q)$. Gudmundsson et al. [10] prove that for every point set $P$, there exists a $(1+\varepsilon)$-spanner $\mathcal{S}$ with $\ell(\mathcal{S}) \le C(\epsilon) \cdot \ell(\mathcal{MST})$, where $\mathcal{MST}$ denotes the minimum spanning tree on $P$. They also show that such a spanner has $\mathcal{O}(n)$ edges and can be constructed in $\mathcal{O}(n \log n)$ time[1], where the $\mathcal{O}$-notation is hiding constants depending on $\varepsilon$.

Secondly, consider the quadtree $QT_{a,b}$ obtained by dividing $\mathfrak{Q}_0$ into four equal-sized squares and recursively repeating this division process until the squares have side length 2 (and therefore contain at most one input point). It is easy to see that this tree has $\Theta(n^2)$ squares. We call $QT_{a,b}$ a *shifted dissection*, as $\mathfrak{Q}_0$ depends on $(a,b)$ and is thus shifted relatively to the point set $P$. We have the following relation between the grid $G_{a,b}(i)$ of Section 3.1 and $QT_{a,b}$: if the levels of $QT_{a,b}$ are numbered bottom-up such that level 1 contains the leaves of $QT_{a,b}$, the grid $G_{a,b}(i)$ dissects $\mathfrak{Q}_0$ into squares of level $i$.

Usually, one stops the division process at squares which contain at most one point of $P$. It then follows similarly to the proof of Lemma 7(i) that the resulting truncated quadtree has $\mathcal{O}(n \log n)$ squares. For our purposes however, we truncate the quadtree slightly differently: Consider the $\mathcal{O}(n \log n)$ allowable rectangles containing at least one point from $P$, and divide each $i$-allowable rectangle into at most $13^2 = 169$ squares of level $i$. Now consider the truncated quadtree $QT^\circ_{a,b}$

[1] A paper of Arya *et al.* [6] claiming this result prior to [10] is incorrect.

consisting of these $\mathcal{O}(n \log n)$ squares and all their ancestors. As each square has $\mathcal{O}(\log n)$ ancestors, $QT^{\circ}_{a,b}$ has $\mathcal{O}\left(n \log^2 n\right)$ squares.

A SLG $\mathcal{G}$ is said to be *$r$-vapid* with respect to a (truncated) quadtree $QT_{a,b}$ if every square $\mathfrak{Q}$ in $QT_{a,b}$ has at most $r$ relevant crossings with $\mathcal{G}$.

**Lemma 18.** *For all choices of $a, b \in \{0, 2, \ldots, L-2\}$, any straight-line graph $\mathcal{G}$ that is $r$-vapid w.r.t. $QT^{\circ}_{a,b}$ is $169r$-sparse.*

*Proof.* Assume that $\mathfrak{R}$ is $i$-allowable. $\mathfrak{R}$ is the union of at most $13^2 = 169$ level-$i$ squares of $QT^{\circ}_{a,b}$, and any relevant crossing of $\mathcal{G}$ with $\mathfrak{R}$ is also relevant with respect to one of these squares. Thus $\mathcal{G}$ is $169r$-sparse if it is $r$-vapid w.r.t. $QT^{\circ}_{a,b}$. □

The concept of shifted dissections and that of $r$-vapidness were introduced in [4] and [13], respectively.

*Proof of Lemma* 17. First, we compute in $\mathcal{O}(n \log n)$ time a $(1+\varepsilon)$-spanner on $P$ of total length

$$\ell(\mathcal{S}) \leq C(\varepsilon) \cdot \ell(\mathcal{MST}), \tag{17}$$

using the algorithm of Gudmundsson *et al.* [10].

Let $\mathcal{W}_1 = \mathcal{W}(T^*, \pi^*)$ denote the walk on $\mathcal{S}$ induced by the optimal solution $(T^*, \pi^*)$. Since $\mathcal{S}$ is a $(1+\varepsilon)$-spanner, we have

$$\ell(\mathcal{W}_1) \leq (1+\varepsilon) \cdot \sum_{\{p,q\} \in \pi^*} d(p,q) \leq \sum_{\{p,q\} \in \pi^*} d(p,q) + \varepsilon \cdot \mathrm{val}(T^*, \pi^*). \tag{18}$$

Let $r := c \cdot C(\varepsilon)/\varepsilon$, for the constant $C(\varepsilon)$ from (17) and a suitably large constant $c > 0$. We transform $\mathcal{S}$ into a graph $\mathcal{S}'$ which is $r_0$-vapid w.r.t. $QT^{\circ}_{a,b}$ for $r_0 := r/169$. Lemma 18 then states that $\mathcal{S}'$ is $r$-sparse. To obtain $\mathcal{S}'$, we proceed along the lines of Rao and Smith [13], adding a set $S$ of artificial points to $\mathcal{S}$ such that the resulting graph $\mathcal{S}'$ is indeed a SLG on a point set $P' = P \mathbin{\dot\cup} S \supseteq P$.

We proceed bottom-up through $QT^{\circ}_{a,b}$ and transform $\mathcal{S}$ into $\mathcal{S}'$ by a sequence of local modifications. Let $\mathfrak{Q}$ denote a square in $QT^{\circ}_{a,b}$ where we encounter more than $r_0$ relevant crossings. Then at least one side of $\mathfrak{Q}$ has more then $r_0/4 - 1$ relevant crossings. At each such side, we modify the SLG according to Figure 5 (see [13] for details). At the same time, we detour the walk $\mathcal{W}_1$ over the single new edge that crosses the side of $\mathfrak{Q}$. The artificial points we add are within small constant distance of $\partial\mathfrak{Q}$, such that the edges we 'patch in' at different levels overlap. Thus, at most one new crossing per side of an already patched square is created by patching operations at higher levels of $QT^{\circ}_{a,b}$. In [13] it was shown that

$$\mathbb{E}\left[\ell(\mathcal{S}') - \ell(\mathcal{S})\right] = \mathcal{O}(1/r)\, \ell(\mathcal{S}), \tag{19}$$

where the expectation is with respect to the random choice of $a$ and $b$. Note that an edge $e$ of $\mathcal{S}$ has relevant crossings with at most $\mathcal{O}(\log n)$ squares of $QT^{\circ}_{a,b}$. Since $\mathcal{S}$ contains $\mathcal{O}(n)$ edges, the total time we need for the modifications according to Figure 5 is $\mathcal{O}(n \log n)$. As we have to consider all $\mathcal{O}\left(n \log^2 n\right)$ squares in $QT^{\circ}_{a,b}$, the total time required by the bottom-up procedure described above is $\mathcal{O}\left(n \log^2 n\right)$.

We obtain a graph $\mathcal{S}'$ which is $r_0$-vapid and thus $r$-sparse, and a walk $\mathcal{W}_2$ on $\mathcal{S}'$.

Observe that $\ell(\mathcal{W}_2) - \ell(\mathcal{W}_1)$ is bounded by the sum of the detours, which is exactly the length of the new edges weighted with their multiplicity in $\mathcal{W}_2$. In order to bound these multiplicities, we further modify $\mathcal{W}_2$ as follows. Consider the multigraph $\mathscr{W}_2$ on vertex set $T \mathbin{\dot\cup} S$ whose edge set is given by the edges of $\mathcal{W}_2$ with their corresponding multiplicity. Since $\mathcal{W}_2$ is a closed walk, $\mathscr{W}_2$ is Eulerian. We replace all edges of odd multiplicity by a single edge, and all edges of even non-zero

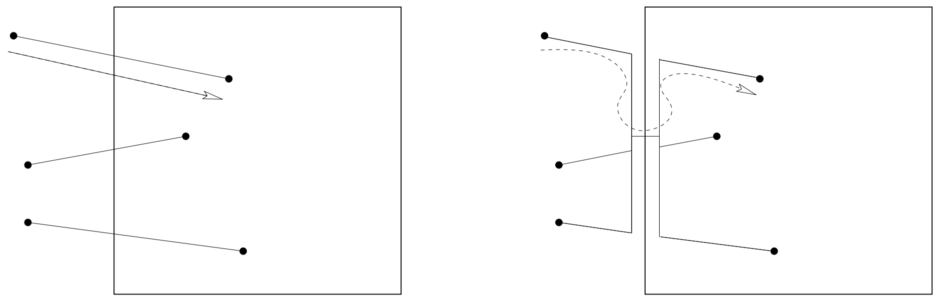

Figure 5. Illustration to the proof of Lemma 17.

multiplicity by two edges. It is easy to see that the multigraph $\mathscr{W}_3 \subseteq \mathscr{W}_2$ we obtain is still Eulerian, and that any Eulerian tour through $\mathscr{W}_3$ is a closed walk $\mathcal{W}_3$ in $\mathcal{S}'$. This argument shows that any shortest salesman tour in any graph uses no edge more than twice.

As $\mathcal{W}_3$ traverses every edge at most twice, we have

$$\ell(\mathcal{W}_3) - \ell(\mathcal{W}_1) \leq 2 \cdot \left(\ell\left(\mathcal{S}'\right) - \ell(\mathcal{S})\right),$$

and consequently

$$\begin{aligned}
\mathbb{E}\left[\ell(\mathcal{W}_3) - \ell(\mathcal{W}_1)\right] \;&\leq\; 2 \cdot \mathbb{E}\left[\ell\left(\mathcal{S}'\right) - \ell(\mathcal{S})\right] \\
&\overset{(19)}{=}\; \mathcal{O}(1/r)\,\ell(\mathcal{S}) \\
&\overset{(17)}{\leq}\; \mathcal{O}(1/r) \cdot C(\varepsilon) \cdot \ell(\mathcal{MST}) \\
&\leq \varepsilon \operatorname{val}(T^*, \pi^*),
\end{aligned}$$

where the last inequality follows from $r = c \cdot C(\varepsilon)/\varepsilon$ with $c$ chosen large enough and the fact that $\ell(\mathcal{MST})$ is a lower bound for $\operatorname{val}(T^*, \pi^*)$. Hence, it follows from (18) that

$$\mathbb{E}\left[\ell(\mathcal{W}_3)\right] \leq \ell(\mathcal{W}_1) + \varepsilon \cdot \operatorname{val}(T^*, \pi^*) \leq \sum_{\{p,q\} \in \pi^*} d(p,q) + 2\varepsilon \cdot \operatorname{val}(T^*, \pi^*),$$

and adapting $\varepsilon$ appropriately completes the proof. □

## 5. A PTAS for VRAP

We now introduce a PTAS for VRAP. Lemma 17 plays a crucial role in our approach. In principle, our PTAS chooses $a$ and $b$ at random, computes $\mathcal{S}'$ and then tries to find an optimal solution $\mathcal{W}_0^*$ to VRAP($\mathcal{S}'$) by dynamic programming, guessing $T_{\mathcal{W}_0^*}$ and the corresponding zoom tree $ZT_{a,b}$ in the process. By Lemma 17, we know that this approach should yield an expected nearly-optimal solution to the original problem VRAP. (Note that $\mathcal{W}_0^*$ does not necessarily equal $\mathcal{W}^*$, as in Lemma 17 we only minimized the tour length and ignored allocation costs.)

This approach needs several extra twists to achieve the desired running time. Most notably, we estimate the allocation costs by the portal-respecting distances. This means that we will not necessarily find $\mathcal{W}_0^*$, but an optimal solution to a slightly modified problem. However, since, by Lemma 9, the portal-respecting distances are good estimates for the real distances in expectation, we can keep the expected total error caused by this small.

Moreover, it would be too time-consuming to allocate all points individually, as the same point is considered in many different steps of the dynamic program. We overcome this difficulty by partitioning the points that need to be allocated in a given step into classes and assigning all points of a class to the same tour point. The errors introduced by this are of the same order of magnitude as the errors inherent in the idea of portal-respecting allocations.

Lastly, to avoid costly shortest path computations, we shortcut between vertices of $\mathcal{S}'$ whenever this does not spoil the sparseness properties of $\mathcal{S}'$ that are crucial to our algorithm. One can think of these shortcuts as additional edges that are added to $\mathcal{S}'$. Indeed, this causes no problems, as it only decreases the length of the tour the algorithm will output, and does not change the allocation costs.

5.1. **Dynamic Programming.** Throughout this section, consider $m = m(\varepsilon)$ and $r = r(\varepsilon)$ fixed according to Lemma 10 (for an as yet unspecified distance measure $d_*$ that will emerge in the course of this section) and Lemma 17. For any allowable rectangle $\mathfrak{R}$ containing at least one point from $P$, let $G_{\text{alloc}} = G_{\text{alloc}}(\mathfrak{R})$ denote the set of the $4m$ portals on $\partial\mathfrak{R}$ as in Section 3, and let $E_{\text{cross}}$ denote the set of edges of $\mathcal{S}'$ crossing the boundary of $\mathfrak{R}$ such that one endpoint is contained in $\mathfrak{R}$. As $\mathcal{S}'$ is $r$-sparse, we have $|E_{\text{cross}}| \leq r$. A *configuration* $\mathcal{C}$ of $\mathfrak{R}$ is given by

(1) a collection $S_{\text{con}}$ of pairs from $E_{\text{cross}}$, where each element appears at most twice,
(2) functions $\zeta_{\text{in}} : G_{\text{alloc}} \to \{1, ..., 2m, \infty\}$ and $\zeta_{\text{out}} : G_{\text{alloc}} \to \{1, ..., 7m, \infty\}$, and
(3) a bit $\sigma \in \{\text{split}, \text{zoomed}\}$.

Since $m$ and $r$ are constant, the total number of configurations of $\mathfrak{R}$ is bounded by a constant depending only on the desired approximation ratio $\varepsilon$.

A configuration $\mathcal{C}$ describes a subproblem, i.e., a local problem for $\mathfrak{R}$, which is interpreted as follows. Firstly, the pairs in $S_{\text{con}}$ determine which of the edges in $E_{\text{cross}}$ must be connected by walks inside $\mathfrak{R}$. As the walk $\mathcal{W}^*$ we are after might visit edges of $\mathcal{S}'$ twice (cf. Lemma 17), we allow $S_{\text{con}}$ to contain duplicate edges. Secondly, the functions $\zeta_{\text{in}}$ and $\zeta_{\text{out}}$ describe the distance from a given portal to the next point on the tour inside resp. outside of $\mathfrak{R}$. More precisely, for every $g \in G_{\text{alloc}}$, the distance from $g$ to the next point on the salesman tour inside $\mathfrak{R}$ is within distance $(|\mathfrak{R}|/m)\zeta_{\text{in}}(g)$. Analogously, the distance to the next point on the salesman tour outside of $\mathfrak{R}$ is encoded by $\zeta_{\text{out}}(g)$. We shall see below that it is sufficient to encode these distances up to $7|\mathfrak{R}|$ only.

We ask for a best possible local solution for a given configuration $\mathcal{C}$. More precisely, we try to minimize the length of all tour edges which lie *completely* inside $\mathfrak{R}$ plus the full allocation costs for *all* non-tour points in $\mathfrak{R}$ (regardless of whether they are allocated to a tour point inside or outside $\mathfrak{R}$), subject to the constraints and guarantees given by $\mathcal{C}$. As we do not know the zoom tree corresponding to the optimal solution in advance, we cannot proceed by a top-down divide and conquer approach along the zoom tree. Instead, we proceed bottom-up by dynamic programming in a much larger structure, which can be seen as the union of the zoom trees for all possible choices of $T \subseteq P$. By dynamic programming, we calculate close upper bounds $T[\mathfrak{R}, \mathcal{C}]$ for the optimal solutions to these local optimization problems. The bit $\sigma$ indicates whether we look at $\mathfrak{R}$ as a split or as a zoomed rectangle in a zoom tree, and thus how our dynamic program calculates $T[\mathfrak{R}, \mathcal{C}]$ from the values previously found for smaller rectangles.

It is crucial that we consider only the $\mathcal{O}(n \log n)$ allowable rectangles $\mathfrak{R}$ containing at least one point of $P$. We topologically sort these rectangles w.r.t. the partial order given by normal set inclusion, and process them in this order, going through all possible configurations of each rectangle.

Let $\mathfrak{R}$ and $\mathcal{C}$ denote the rectangle and the configuration we currently consider. We now distinguish two cases: if $\mathfrak{R}$ is 1-allowable or $|P \cap \mathfrak{R}| = 1$ (case A), we find $T[\mathfrak{R}, \mathcal{C}]$ by exhaustive search. Otherwise, we calculate $T[\mathfrak{R}, \mathcal{C}]$ by dynamic programming from previously found values in one of two possible ways (case B), as specified by $\sigma$.

**Case A. $\mathfrak{R}$ is 1-allowable or $|P \cap \mathfrak{R}| = 1$.** Note that this means that $\mathfrak{R}$ is a leaf in any zoom tree it is contained in. Also note that a 1-allowable rectangle contains at most $13^2 = 169$ input points from $P$. This allows us to proceed in brute force fashion.

Let $\mathfrak{R}$ and $\mathcal{C}$ be given. First we choose from $P \cap \mathfrak{R}$ the points that lie on the salesman paths inside $\mathfrak{R}$. Let $T_0 \subseteq P \cap \mathfrak{R}$ denote this point set. Since we know that points $p$ with $\beta(p) > 2$ are visited by the optimal solution $T^*$ (and therefore also by the walk $\mathcal{W}^*$ from Lemma 17), we always include such points in $T_0$. Also, by definition of the zoom step, it is clear that $T_0$ should not be empty if $\sigma = $ zoomed. The points in $A_0 := (P \cap \mathfrak{R}) \setminus T_0$ need to be allocated to some tour point either inside or outside of $\mathfrak{R}$. We have $\mathcal{O}(1)$ choices of $T_0$. For every such choice, we check whether it satisfies the restrictions given by $\zeta_{\text{in}}$, i.e., whether the distance from every portal $g \in G_{\text{alloc}}$ to $T_0$ is bounded by $(|\mathfrak{R}|/m)\zeta_{\text{in}}(g)$. If this is not the case, we reject this choice of $T_0$. If all possible choices of $T_0$ are rejected, we reject $\mathcal{C}$ and set $T[\mathfrak{R}, \mathcal{C}] = \infty$.

Next, we compute optimal walks visiting exactly the points in $T_0$ subject to the constraints given by $S_{\text{con}}$. We do not require these walks to use edges of $\mathcal{S}'$, but calculate them on the complete graph induced by the (at most 169) points in $T_0$ and the (at most $r$) endpoints of the edges from $S_{\text{con}}$ (note that this can be viewed as adding shortcut edges to $\mathcal{S}'$). Thus, the optimal walks can be found in constant time.

Moreover, we estimate for each $p \in A_0$ its allocation cost. It is trivial to compute the distances to points in $T_0$. On the other side, $\mathcal{C}$ guarantees that the nearest tour point outside $\mathfrak{R}$ has portal-respecting distance at most

$$\min_{g \in G_{\text{alloc}}} \left\{ d(p, g) + \frac{|\mathfrak{R}|}{m} \zeta_{\text{out}}(g) \right\}$$

by definition of $\zeta_{\text{out}}(g)$. So we decide for every point $p \in A_0$ whether it is cheaper to allocate it to a point inside or a point outside $\mathfrak{R}$ (actually, a portal). As we have $4m$ portals and at most 169 points, this can be done in time $\mathcal{O}(1)$.

The total cost for this choice of $T_0$ is the total length of all edges on the salesman paths that are entirely in $\mathfrak{R}$, plus the total allocation cost for all points in $A_0$ calculated as explained above. Note that this overestimates the allocation cost for points which are allocated to tour points outside $\mathfrak{R}$. We identify the choice of $T_0$ minimizing this cost and store the corresponding value in $T[\mathfrak{R}, \mathcal{C}]$. Thus, we can compute $T[\mathfrak{R}, \mathcal{C}]$ in $\mathcal{O}(1)$ time.

**Case B. Otherwise.** In this case we in particular have no upper bound on the number of input points inside $\mathfrak{R}$. Thus, it is not longer possible to compute $T[\mathfrak{R}, \mathcal{C}]$ using brute force search. As we process the rectangles in ascending order, we may assume that we already have the values $T[\mathfrak{R}', \mathcal{C}']$ for all allowable rectangles $\mathfrak{R}' \subset \mathfrak{R}$ and all configurations $\mathcal{C}'$ of $\mathfrak{R}'$.

***Case B1.*** $\sigma = $ **zoomed.** We split $\mathfrak{R}$ into two allowable rectangles $\mathfrak{R}'$ and $\mathfrak{R}''$ according to the properties of zoom trees (cf. Section 3). By Lemma 15, we know that both rectangles should contain at least one point of $P$. If this is not the case, we set $T[\mathfrak{R}, \mathcal{C}] = \infty$ for all configurations of $\mathfrak{R}$ with $\sigma = $ zoomed.

We enumerate all choices $(\mathcal{C}', \mathcal{C}'')$ of pairs of configurations of $\mathfrak{R}'$ and $\mathfrak{R}''$ with $\sigma' = \sigma'' = $ split, checking for each choice whether it is consistent in itself and compatible with $\mathcal{C}$, i.e., whether a set of salesman paths satisfying $S_{\text{con}}$ can be obtained by joining salesman paths satisfying $S'_{\text{con}}$ and $S''_{\text{con}}$, and whether the functions $\zeta'_{\text{in}}, \zeta''_{\text{in}}, \zeta_{\text{out}}$ guarantee that the requirements on the position of the tour points given by the functions $\zeta_{\text{in}}, \zeta'_{\text{out}}, \zeta''_{\text{out}}$ are satisfied. That is, we have to check the equations

$$\frac{|\mathfrak{R}|}{m}\zeta_{\text{in}}(g) \geq \min\left\{\min_{g'\in G'_{\text{alloc}}}\left\{d(g,g') + \frac{|\mathfrak{R}'|}{m}\zeta'_{\text{in}}(g')\right\}, \min_{g''\in G''_{\text{alloc}}}\left\{d(g,g'') + \frac{|\mathfrak{R}''|}{m}\zeta''_{\text{in}}(g'')\right\}\right\} \tag{20}$$

for the portals $g \in G_{\text{alloc}}$ on $\partial\mathfrak{R}$, and

$$\frac{|\mathfrak{R}'|}{m}\zeta'_{\text{out}}(g') \geq \min\left\{\min_{g\in G_{\text{alloc}}}\left\{d(g',g) + \frac{|\mathfrak{R}|}{m}\zeta_{\text{out}}(g)\right\}, \min_{g''\in G''_{\text{alloc}}}\left\{d(g',g'') + \frac{|\mathfrak{R}''|}{m}\zeta''_{\text{in}}(g'')\right\}\right\} \tag{21}$$

for the portals $g' \in G'_{\text{alloc}}$ on $\partial\mathfrak{R}'$ and analogously for $g'' \in G''_{\text{alloc}}$ on $\partial\mathfrak{R}''$.

For all pairs of configurations which remain, we compute $T[\mathfrak{R}',\mathcal{C}'] + T[\mathfrak{R}'',\mathcal{C}'']$ and add the total length of the edges in $S'_{\text{con}} \cap S''_{\text{con}}$, which are exactly the tour edges with one endpoint in $\mathfrak{R}'$ and the other in $\mathfrak{R}''$. A given edge may be counted twice in this calculation, which is in line with our definition of $S_{\text{con}}$. We choose the pair $(\mathcal{C}',\mathcal{C}'')$ which minimizes this sum and write its value to $T[\mathfrak{R},\mathcal{C}]$. Note that all computations can be accomplished in $\mathcal{O}(1)$ time.

***Case B2.*** $\sigma$ = split**.** We enumerate all allowable rectangles $\mathfrak{R}' \subset \mathfrak{R}$ containing at least one point from $P$ and all points $p \in P \cap \mathfrak{R}$ with $\beta(p) > 2$. For any such rectangle $\mathfrak{R}'$, we consider all configurations $\mathcal{C}'$ with $\sigma'$ = zoomed that are compatible with $\mathcal{C}$ in the same sense as in case B1. For the allocations, we have to check whether the equations

$$\frac{|\mathfrak{R}|}{m}\zeta_{\text{in}}(g) \geq \min_{g'\in G'_{\text{alloc}}}\left\{d(g,g') + \frac{|\mathfrak{R}'|}{m}\zeta'_{\text{in}}(g')\right\}, \quad \forall g \in G_{\text{alloc}} \tag{22}$$

and

$$\frac{|\mathfrak{R}'|}{m}\zeta'_{\text{out}}(g') \geq \min_{g\in G_{\text{alloc}}}\left\{d(g',g) + \frac{|\mathfrak{R}|}{m}\zeta_{\text{out}}(g)\right\}, \quad \forall g' \in G'_{\text{alloc}} \tag{23}$$

are satisfied. For a given choice of $\mathfrak{R}'$ and $\mathcal{C}'$, the total cost for $\mathfrak{R}$ is $T[\mathfrak{R}',\mathcal{C}']$ plus the additional tour costs plus the additional allocation cost. Recall that by definition of the zoom step, there are no tour points in $\mathfrak{R} \setminus \mathfrak{R}'$.

To compute the additional tour cost, we enumerate all possible ways to connect the edges in $S_{\text{con}} \cup S'_{\text{con}}$ such that one obtains exactly the salesman paths required by $S_{\text{con}}$. Note that a salesman path through $\mathfrak{R}$ may enter and leave $\mathfrak{R}'$ several times. We add the total length of the edges in $S'_{\text{con}} \setminus S_{\text{con}}$ (as before twice if necessary), plus the Euclidean distances between corresponding end points of edge pairs in $S_{\text{con}} \cup S'_{\text{con}}$ (as before, this corresponds to introducing shortcut edges into $\mathcal{S}'$). As there are $\mathcal{O}(r)$ edges in $S_{\text{con}}$ and $S'_{\text{con}}$, we can find the 'connection pattern' which minimizes this cost in constant time.

Computing the additional allocation cost is somewhat tricky. We need to allocate all input points in $\mathfrak{R} \setminus \mathfrak{R}'$. As we do not have a bound on the number of such points, doing this for each point separately, i.e., calculating

$$C_{\text{alloc}} := \sum_{p\in P\cap(\mathfrak{R}\setminus\mathfrak{R}')}\Big(\alpha(p) + \beta(p)\cdot\min\Big\{\min_{g\in G_{\text{alloc}}}\big\{d(p,g) + \frac{|\mathfrak{R}|}{m}\zeta_{\text{out}}(g)\big\}, \min_{g'\in G'_{\text{alloc}}}\big\{d(p,g') + \frac{|\mathfrak{R}'|}{m}\zeta'_{\text{in}}(g')\big\}\Big\}\Big) \tag{24}$$

exactly, would be prohibitively expensive. However, as we shall show in a moment, we can quickly approximate these allocation costs close enough for our purposes. This is basically achieved by subdividing $\mathfrak{R} \setminus \mathfrak{R}'$ into cells and assigning all points in a cell to the same portal.

**Lemma 19.** *One can preprocess $P$ in $\mathcal{O}(n \log n)$ time such that one can calculate in $\mathcal{O}\left(\log^3 n\right)$ time for any pair of allowable rectangles $\mathfrak{R}' \subset \mathfrak{R}$ an upper bound $\widehat{C}_{alloc}$ on $C_{alloc}$ which is tight up to a relative error of $\mathcal{O}(1/m)$ and an absolute error of $\mathcal{O}(k(|\mathfrak{R}|/m) + k'(|\mathfrak{R}'|/m))$, where $k$ and $k'$ denote the number of points for which the minimum in (24) is attained by a portal of $G_{\text{alloc}}$, respectively $G'_{\text{alloc}}$.*

We choose the configuration $\mathcal{C}'$ which minimizes the accumulated cost and store the total cost in $T[\mathfrak{R}, \mathcal{C}]$.

*Proof of Lemma* 19. We subdivide $\mathfrak{R} \setminus \mathfrak{R}'$ into rectangular rings $A_1, \ldots, A_t$. The outer boundary of ring $A_i$ is at distance

$$d_i = |\mathfrak{R}'| (1 + 1/m)^i$$

from $\mathfrak{R}'$, whereas the inner boundary is at distance $d_{i-1}$ (or 0 if $i = 1$), i.e., the inner boundary of $A_i$ is the outer boundary of $A_{i-1}$. This is illustrated in Figure 6. The cells are constructed by subdividing each ring $A_i$ into $\Theta(m^2)$ equally-sized cells. Note that the side length $|\mathfrak{C}|$ of a cell $\mathfrak{C}$ in ring $A_i$ is $\mathcal{O}(d_i/m)$.

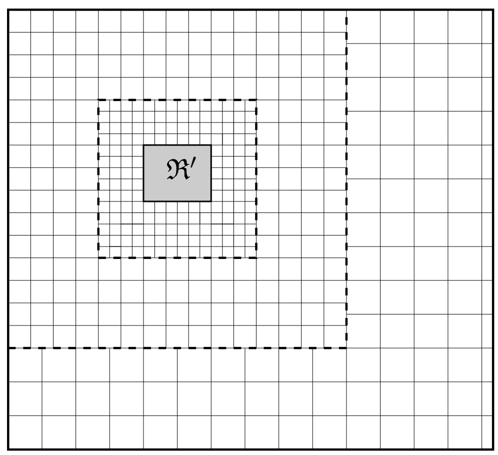


FIGURE 6. Subdivision of $\mathfrak{R} \setminus \mathfrak{R}'$ into cells. First we subdivide $\mathfrak{R} \setminus \mathfrak{R}'$ into rings of geometrically increasing radius. Then, we subdivide each ring into $\Theta(m^2)$ cells.

We have $t = \mathcal{O}(\log(|\mathfrak{R}|/|\mathfrak{R}'|)) = \mathcal{O}(\log n)$ many rings, and therefore also $\mathcal{O}(\log n)$ many cells. By orthogonal semigroup range searching (see [1, 2] for references), we compute for each cell $\mathfrak{C}$ the values $\alpha(\mathfrak{C}) := \sum_{p \in P \cap \mathfrak{C}} \alpha(p)$ and $\beta(\mathfrak{C}) := \sum_{p \in P \cap \mathfrak{C}} \beta(p)$. This can be done in $\mathcal{O}\left(\log^2 n\right)$ time per cell and requires preprocessing the points in $\mathcal{O}(n \log n)$ time before the start of the dynamic program.

Recall that the idea is to assign all input points within a cell to the same portal. With the values $\alpha(\mathfrak{C})$ and $\beta(\mathfrak{C})$ at hand, we can calculate

$$\widehat{C}_{\text{alloc}} := \sum_{\mathfrak{C} \subseteq \mathfrak{R} \setminus \mathfrak{R}'} \Big( \alpha(\mathfrak{C}) + \beta(\mathfrak{C}) \cdot \min \Big\{ \min_{g \in G_{\text{alloc}}} \big\{ d(c(\mathfrak{C}), g) + |\mathfrak{C}| + \frac{|\mathfrak{R}|}{m} \zeta_{\text{out}}(g) \big\}, \min_{g' \in G'_{\text{alloc}}} \big\{ d(c(\mathfrak{C}), g') + |\mathfrak{C}| + \frac{|\mathfrak{R}'|}{m} \zeta_{\text{in}}(g') \big\} \Big\} \Big)$$

in time proportional to the number of cells, i.e., $\mathcal{O}(\log n)$. Here $c(\mathfrak{C})$ denotes the center of $\mathfrak{C}$.

It remains to show that this is a good upper bound on $C_{\text{alloc}}$. Clearly, we have $d(c, g) - |\mathfrak{C}| \leq d(p, g) \leq d(c, g) + |\mathfrak{C}|$ for a cell $\mathfrak{C}$ with center $c$, $p \in \mathfrak{C}$ and any portal $g$. It follows that $C_{\text{alloc}} \leq \widehat{C}_{\text{alloc}}$, and that for every point $p \in \mathfrak{C} \cap P$ we overestimate its allocation cost by at most $2\beta(p)|\mathfrak{C}| \leq 4|\mathfrak{C}|$ (recall that points with $\beta(p) > 2$ are in $\mathfrak{R}'$ and therefore not considered here.)

The largest cells in our subdivision have side length $|\mathfrak{C}| = \Theta(|\mathfrak{R}|/m)$. Therefore, the absolute error per point is bounded by $\mathcal{O}(|\mathfrak{R}|/m)$, which already suffices for points that are allocated to portals of $G_{\text{alloc}}$. For points which are allocated to portals of $G'_{\text{alloc}}$, we distinguish two cases. If $\mathfrak{C}$ belongs to ring $A_1$, we have $|\mathfrak{C}| = \Theta(|\mathfrak{R}'|/m)$. Otherwise, we obtain with $d(p, g) \geq d_{i-1}$ that $|\mathfrak{C}| = \mathcal{O}(d_{i-1}/m) = \mathcal{O}(d(p, g)/m)$.

Summing up these pointwise error guarantees yields that $\widehat{C}_{\text{alloc}}$ indeed approximates $C_{\text{alloc}}$ as claimed. □

At the end of the dynamic program, we obtain a value $T[\mathfrak{Q}_0, \bigcirc] =: T[\mathfrak{Q}_0]$ for $\mathfrak{Q}_0 = (a, b) + [-L, L]^2$ and the configuration $\bigcirc$ with $S_{\text{con}} = \emptyset$, $\zeta_{\text{in}}(g) = \zeta_{\text{out}}(g) = \infty$ for all portals $g \in G_{\text{alloc}}(\mathfrak{Q}_0)$, and $\sigma = \text{split}$.

5.2. **Analysis.** We prove time complexity and correctness separately, starting with the former.

*Proof of Theorem* 1 *(Complexity).* Preprocessing the points as required by Lemma 19 and computing the $r$-sparse PSL guaranteed by Lemma 17 takes time $\mathcal{O}\left(n \log^2 n\right)$.

In the dynamic program, by Lemma 7(i), cases A and B1 apply $\mathcal{O}(n \log n)$ times and can be computed in time $\mathcal{O}(1)$. By Lemma 7(ii), case B2 applies to $\mathcal{O}\left(n \log^2 n\right)$ pairs of rectangles, and can be computed in time $\mathcal{O}\left(\log^3 n\right)$, by Lemma 19. This yields an overall complexity of $\mathcal{O}\left(n \log^5 n\right)$. □

It remains to argue that our algorithm produces a nearly-optimal solution. In order to check this we need the following statement.

**Lemma 20.** *The value $T[\mathfrak{Q}_0]$ calculated by the dynamic program satisfies*

$$\mathbb{E}\left[T[\mathfrak{Q}_0]\right] \leq (1 + \varepsilon)\text{val}\left(T^*, \pi^*\right).$$

*Proof of Lemma* 20. Let $ZT_{a,b}$ denote the zoom tree of $T^*$ as in Section 3, and let $\mathcal{W}^*$ denote the shortest walk on $\mathcal{S}'$ visiting all points of $T^*$ as in Lemma 17. Recall that $ZT_{a,b}$ stops the subdivision at a rectangle $\mathfrak{R}'$ if either $\mathfrak{R}'$ is 1-allowable or contains at most one point of $P$, and that every rectangle $ZT_{a,b}$ contains a point from $T^*$ by Lemma 15. For all rectangles of $ZT_{a,b}$, we now specify configurations $\mathcal{C}^*$ that are compatible with each other (and therefore considered by the dynamic program), and result in the claimed bound for $\mathbb{E}\left[T[\mathfrak{Q}_0]\right]$.

Firstly, set for all rectangles $\sigma^*$ appropriately, i.e., $\sigma^* = \text{zoomed}$ if $\mathfrak{R}$ is zoomed in $ZT_{a,b}$ and $\sigma^* = \text{split}$, otherwise. Furthermore, choose $S^*_{\text{con}}$ according to $\mathcal{W}^*$.

Secondly, specify the functions $\zeta^*_{\text{in}}$ as follows. For leaves $\mathfrak{R}'$ of $ZT_{a,b}$ set them as small as possible such that still $d(T^* \cap \mathfrak{R}', g) \leq (|\mathfrak{R}'|/m)\zeta_{\text{in}}(g)$, and propagate these restrictions up the zoom tree such that (20) and (22) hold. Due to the integrality of $\zeta^*_{\text{in}}$, this introduces absolute errors of order $\mathcal{O}\left(2^i/m\right)$ for rectangles at level $i$. These errors sum up as a geometric series along the zoom tree, resulting in a total absolute error of $\mathcal{O}(|\mathfrak{R}|/m)$ for the portals on $\partial\mathfrak{R}$.

Finally, set $\zeta^*_{\text{out}}$ to $\infty$ on portals of $\mathfrak{Q}_0$, and use (21) and (23) to calculate $\zeta^*_{\text{out}}$ for all rectangles top-down along the zoom tree. Note that finite values are introduced due to (21) by the values $\zeta^*_{\text{in}}$ we just calculated. As these errors are propagated *down* the zoom tree, the total absolute error for portals on a rectangle $\mathfrak{R}$ is not necessarily $\mathcal{O}(|\mathfrak{R}|/m)$. However, observe that for every portal $g \in G_{\text{alloc}}$ on $\partial\mathfrak{R}$ and every tour point $q \in T^*$ we have

$$\frac{|\mathfrak{R}|}{m}\zeta^*_{\text{out}}(g) \leq d(g, q) + \mathcal{O}\left(|\widetilde{\mathfrak{R}}|/m\right), \tag{25}$$

where $\widetilde{\mathfrak{R}}$ is the rectangle separating the points in $\mathfrak{R}$ from $q$ (provided the right hand side is at most $7|\mathfrak{R}|$).

As the $\zeta^*_{\text{out}}$-functions only encode distances up to $7|\mathfrak{R}|$, some of these might be set to $\infty$ even if there are tour points outside of $\mathfrak{R}$. However, our PTAS does not require that we encode longer distances. This is easily checked as follows. Whenever we allocate a point $p \in P \cap \mathfrak{R}$ we are either in case A or in case B2 of the dynamic program. If $\mathfrak{R}$ is split, its sibling contains at least one tour point, say $q$ (cf. Lemma 15). Let $\mathfrak{R}_0$ denote the parent of $\mathfrak{R}$, and note that $d(g,q) \leq 2|\mathfrak{R}_0| \leq 6|\mathfrak{R}|$ for every portal $g \in G_{\text{alloc}}$ on $\partial\mathfrak{R}$ (cf. Lemma 11). As $q$ is in the sibling of $\mathfrak{R}$, we know that $\mathfrak{R}$ separates $q$ from the points in $\mathfrak{R}$. Thus, the right hand side of (25) and consequently also the left hand side of (25) is at most $7|\mathfrak{R}|$. Therefore, our encoding suffices in this case. If $\mathfrak{R}$ is zoomed, we are in case A and can allocate any point $p \in \mathfrak{R}$ (exactly!) to a tourpoint inside $\mathfrak{R}$, which has distance at most $2|\mathfrak{R}|$ from $p$. This shows that it suffices to encode $\zeta_{\text{out}}$ only up to $7m$.

The inaccuracy in the encoding of distances is one of three sources of error in the calculation of allocation costs. The second one is the error inherent in the concept of portal-respecting allocations (cf. Lemma 8), and the third one is introduced by Lemma 19. In the calculation of the distance of a given point $p$ to a (nearby) tourpoint $q \in T^*$ , these errors sum to an absolute error of $\mathcal{O}\left(|\widetilde{\mathfrak{R}}|/m\right)$ for the rectangle $\widetilde{\mathfrak{R}}$ separating $p$ and $q$, and a relative error of $\mathcal{O}\,(1/m)$ which is easily bounded by $\varepsilon$ choosing $m$ large enough. By Lemma 10, it follows that the absolute errors result in an expected relative error of $\varepsilon$. In total, the expected allocation cost our algorithm calculates is at most $(1+\varepsilon)^2$ times the allocation cost in $(T^*, \pi^*)$.

Moreover, the tour's length (implicitly) calculated in $T[\mathfrak{Q}_0]$ is deterministically bounded by $\ell\,(\mathcal{W}^*)$, which in expectation overestimates the tour cost of the optimum by at most $\varepsilon\text{val}\,(T^*, \pi^*)$ due to Lemma 17. Adapting $\varepsilon$, the claim follows by linearity of expectation. □

*Proof of Theorem* 1 *(Correctness).* Let $\mathcal{S}^* \supseteq \mathcal{S}'$ denote the graph obtained by inserting the shortcut edges the dynamic program used in cases A and B2 into $\mathcal{S}'$. Our PTAS outputs a walk $\mathcal{W}$ on $\mathcal{S}^*$ and a number $T[\mathfrak{Q}_0]$. Since we overestimated the allocation costs in the dynamic program, $T[\mathfrak{Q}_0]$ is an upper bound on $\text{val}_{\mathcal{S}^*}(\mathcal{W})$. Moreover, the solution $(T, \pi)$ to VRAP induced by $\mathcal{W}$ has the same allocation costs as $\mathcal{W}$ but possibly shorter tour length, since it is not restricted to $\mathcal{S}^*$. Therefore, we have

$$\text{val}\,(T,\pi) \leq \text{val}_{\mathcal{S}^*}(\mathcal{W}) \leq T[\mathfrak{Q}_0],$$

and the claim follows from Lemma 20. □

## 6. Steiner VRAP

In this section we show Theorem 2, sketching a PTAS with nearly-linear time complexity for Steiner VRAP. Recall that a solution $(T, S, \pi)$ to Steiner VRAP is determined by point sets $T \subseteq P$ and $S \subset \mathbb{R}^2$, and a salesman tour $\pi$ through $T \,\dot\cup\, S$. We wish to find a solution $(T^*, S^*, \pi^*)$ such that $\text{val}^\bullet\,(T^*, S^*, \pi^*)$ is minimum (cf. (1) on p. 2).

In the following, it is crucial that we may restrict our attention to instances in which the input points have odd integral coordinates and the side length of the bounding box is $\mathcal{O}\,(n/\varepsilon)$ and a power of 2. This is easily checked using the arguments given in Section 2 for VRAP.

The PTAS for Purchase Cooperative TSP proposed in [3] proceeds by dynamic programming in a shifted quadtree $QT_{a,b}$ (cf. p. 14), quite similar to Arora's $\mathcal{O}(n \log^{\mathcal{O}(1/\varepsilon)} n)$ PTAS for TSP [4]. Here, a configuration of a given square $\mathfrak{Q} \in QT_{a,b}$ is defined by specifying for each of $\mathcal{O}\,(\log n/\varepsilon)$ portals whether a tour- and/or an allocation edge runs through it. This results in $\mathcal{O}\left(n^{\mathcal{O}(1/\varepsilon)}\right)$ possible configurations for $\mathfrak{Q}$ and the same overall complexity for the algorithm. The key observation that allows us to improve on this is the following:

**Lemma 21.** *Let* $(T, S, \pi)$ *denote a solution to* STEINER VRAP *crossing a fixed line segment* $L$ *of length* $x$ *five or more times. Then there exists a solution* $(T, S', \pi')$ *with* $S \subseteq S'$ *crossing* $L$ *no more than four times and satisfying* $\mathrm{val}^{\bullet}(T, S', \pi') \leq \mathrm{val}^{\bullet}(T, S, \pi) + c \cdot x$ *for some constant* $c$.

*Proof of Lemma* 21. If two or more allocation edges cross $L$ and allocate a point to the left to a tour point to the right, we proceed as depicted in Figure 7. Observe that the value of the new solution exceeds that of the old one by at most $2x$, as we add tour segments of total length at most $2x$ along $L$ and replace the allocation segments to the right of $L$ by two tour segments which have the length of the shortest allocation segment being replaced. The second operation only decreases the cost because $\beta(p) \geq 1$ for all $p \in P$.

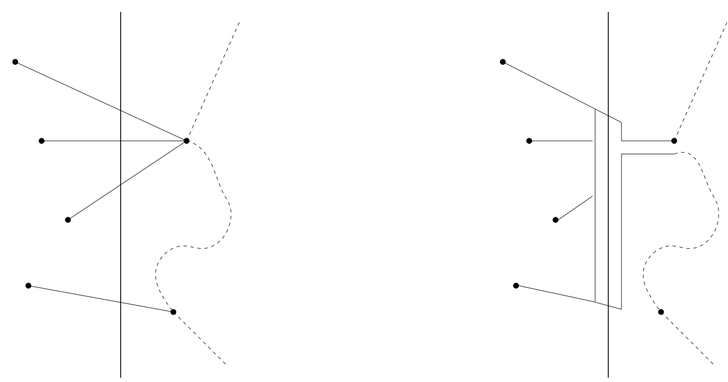

FIGURE 7. Illustration to the proof of Lemma 21.

We proceed analogously if two or more allocation edges cross $L$ from right to left. After these operations, there remains at most one allocation edge per direction, and the value of the modified solution is increased by at most $4x$. Note that the modified salesman tour $\widetilde{\pi}$ visits a point set $T \dot\cup S'$ which consists of all points of the original tour $\pi$ plus some newly added Steiner points.

Now, we apply Lemma 3 of [4] to the salesman tour $\widetilde{\pi}$. This lemma states that, if a salesman tour $\pi$ crosses $L$ three or more times, there exists a salesman tour $\pi'$ visiting the same points as $\pi$ but crossing $L$ at most twice and satisfying $\ell(\pi') - \ell(\pi) \leq g \cdot x$ for some constant $g$.

Thus, by replacing $\widetilde{\pi}$ by a new tour $\pi'$ through $T \dot\cup S'$, we obtain a modified solution $(T, S', \pi')$ such that the total number of crossings is at most four and

$$\mathrm{val}^{\bullet}\left(T, S', \pi'\right) \leq \mathrm{val}^{\bullet}(T, S, \pi) + g \cdot x + 4 \cdot x.$$

This completes the proof. □

Lemma 21 is an extension of Lemma 3 in [4]. With Lemma 21 in hand, one can show as in [4] that there exists an expected $(1+\varepsilon)$-approximation crossing each square of the shifted quadtree $QT_{a,b}$ only $r = \mathcal{O}(1/\varepsilon)$ times. This makes it possible to bound the number of configurations per square by $\mathcal{O}(\log^{\mathcal{O}(1/\varepsilon)} n)$. Combining techniques presented in [4] and [5], one obtains a randomized PTAS for STEINER VRAP with complexity $\mathcal{O}(n \log^{\mathcal{O}(1/\varepsilon)} n)$. In the following, we outline the key ideas.

We proceed by dynamic programming in a shifted quadtree $QT_{a,b}$ that is truncated at squares that contain only one input point. For convenience, we also include all siblings of these squares, so that every non-leaf of $QT_{a,b}$ has exactly four children. Around every square $\mathfrak{Q} \in QT_{a,b}$, we place

$m = \mathcal{O}(\log n/\varepsilon)$ equally spaced portals. Similar to [4] and [5], we only consider portal-respecting solutions, i.e., both tour and allocation edges may cross square boundaries only at portals. In some sense, the portals play the role of 'auxiliary' Steiner points that, however, are not used for allocations. Clearly, removing them in the end only improves the found solution. A *configuration* $\mathcal{C}$ of a square $\mathfrak{Q} \in QT_{a,b}$ is given by

(1) up to $r$ portals where the tour crosses the boundary of $\mathfrak{Q}$, and information how those portals are connected within $\mathfrak{Q}$,
(2) up to $r$ portals that are used for allocations, and for each of these portals a value $\zeta_{\text{in}} \in \{1, ..., 2m, \infty\}$ or $\zeta_{\text{out}} \in \{1, ..., 4m, \infty\}$,
(3) a bit $\sigma \in \{\text{allocate}, \text{recurse}\}$.

Similarly to Section 5, the values $\zeta_{\text{in}}$ and $\zeta_{\text{out}}$ specify the distances to the next tour or Steiner point inside resp. outside $\mathfrak{Q}$ with a precision of $|\mathfrak{Q}|/m$. We shall see that it suffices to encode these distances only up to $4|\mathfrak{Q}|$.

There are $m^{2r}$ many ways to specify the location of the $2r$ portals, and $\mathcal{O}(m)^r$ ways to choose the $r$ values $\zeta_{\text{in}}$ and $\zeta_{\text{out}}$. Since $m = \mathcal{O}(\log n/\varepsilon)$ and $r = \mathcal{O}(1/\varepsilon)$, the total number of configurations of a given square $\mathfrak{Q} \in QT_{a,b}$ is $\mathcal{O}(\log^{\mathcal{O}(1/\varepsilon)} n)$.

The dynamic program works similar to that proposed in Section 5. We try to find good solutions to the local optimization problems posed by the configurations $\mathcal{C}$ of the squares $\mathfrak{Q} \in QT_{a,b}$, calculating, for each configuraction $\mathcal{C}$, a value $T[\mathfrak{Q}, \mathcal{C}]$ that is a close approximation of the sum of the total length of *all* tour segments inside $\mathfrak{Q}$ and the *full* allocation costs of all allocation points in $\mathfrak{Q}$. The bit $\sigma$ indicates how these values are calculated. The interpretation of $\sigma = \text{allocate}$ is that $\mathfrak{Q}$ contains no tour segment. In particular, the tour does not cross the boundary of $\mathfrak{Q}$, and all input points in $\mathfrak{Q}$ are allocation points. In this case we calculate $T[\mathfrak{Q}, \mathcal{C}]$ directly, using the portal-respecting distances as estimates for the real distances as before. If $\sigma = \text{recurse}$, we calculate $T[\mathfrak{Q}, \mathcal{C}]$ recursively, using the values found for its four children (unless $\mathfrak{Q}$ is a leaf of $QT_{a,b}$).

In total, we distinguish three cases, depending on $\sigma$ and the number of points $\mathfrak{Q}$ contains. We start with the two base cases.

**Case A1.** $\sigma =$ **allocate.** As in Section 5, we estimate the allocation cost for each point using the function $\zeta_{\text{out}}$. (Recall that $\sigma = \text{allocate}$ means that $\mathfrak{Q}$ contains no tour segment, so we can only allocate to the outside of $\mathfrak{Q}$.) Since we only recurse as long as there are tour segments in the square at hand, we know that one of the siblings of $\mathfrak{Q}$ contains a tour segment, which is at distance at most $4|\mathfrak{Q}|$ from any portal on $\mathfrak{Q}$. Therefore, our encoding of the distances suffices. As we only consider solutions that cross each rectangle at most $r$ times, we know that $\mathfrak{Q}$ contains at most $r$ points. Hence, one application of case A1 requires constant time.

**Case A2.** $\sigma =$ **recurse and** $|P \cap \mathfrak{Q}| \leq 1$**.** This means that $\mathfrak{Q}$ is a leaf of $QT_{a,b}$. If $\mathfrak{Q}$ contains a point $p \in P$, we guess whether it is a tour point or not. Also, we guess the coordinates of the Steiner points on an $m \times m$ grid subdividing $\mathfrak{Q}$. Note that it makes no sense to introduce more than one Steiner point per allocation. Therefore, it suffices to guess the locations of at most $r+1$ Steiner points, at most $r$ for the $\zeta_{\text{in}}$'s and possibly one for $p$. For each such choice, we check whether the restrictions given by the functions $\zeta_{\text{in}}$ are satisfied, and calculate optimal salesman paths on the tour portals, Steiner points, and possibly $p$ (if we guessed it to be a tour point). If we guessed $p$ to be an allocation point, we allocate it as usual either to a Steiner point inside $\mathfrak{Q}$ or to an allocation portal on $\mathfrak{Q}$.

We keep the choice of Steiner points minimizing the total cost and write its value to $T[\mathfrak{Q}, \mathcal{C}]$. There are at most $m^{2(r+1)}$ choices for the location of the Steiner points, and for every such choice, all calculations can be done in constant time. Therefore, case A2 takes time $\mathcal{O}(\log^{\mathcal{O}(1/\varepsilon)} n)$ to compute whenever it occurs.

**Case B.** $\sigma =$ **recurse and** $|P \cap \mathfrak{Q}| \geq 2$**.** This is a divide-and-conquer step quite similar to case B1 in Section 5.1. We split the square into its four children, go through all quadruples of configurations which are consistent with each other and $\mathcal{C}$, and minimize the sum of the four values $T[\mathfrak{Q}_i, \mathcal{C}_i]$. For $\mathcal{C}$ fixed, there are $(\mathcal{O}(\log^{\mathcal{O}(1/\varepsilon)} n))^4$ choices for the configurations of its children, and for every such choice all calculations can be done in constant time. Hence case B1 takes time $\mathcal{O}(\log^{\mathcal{O}(1/\varepsilon)} n)$ to compute whenever it occurs.

It immediately follows that the complexity of our dynamic program is $\mathcal{O}(n \log^{\mathcal{O}(1/\varepsilon)} n)$. The approximation ratio can be bounded as in [4] and [5], since all errors are of order $\mathcal{O}(|\mathfrak{Q}|/m)$ and can be charged to the line segment crossed by the corresponding allocation or tour edge as in Arora's original argument. There is one exception to this statement: If the single input point $p$ in case A2 is allocated to a Steiner point within the same square, its allocation edge does not cross any square boundary to which its error can be charged. However, if this occurs, there is at least one tour edge crossing the square boundary to which we may charge the error. This does not spoil the analysis given in [4] and [5].

## 7. Concluding Remarks

It is easily checked that our PTAS for VRAP extends to higher dimensions with minimal modifications. The running time increases to $\mathcal{O}\left(n \log^{d+3} n\right)$, as the range searching in Lemma 19 takes time $\mathcal{O}\left(\log^d n\right)$ per cell in dimension $d$. This can be reduced by a factor of $\mathcal{O}(\log n)$ if $\beta(p) = \beta(q)$ for all $p, q \in P$, since then it suffices to count the number of points in a given cell (see [2]).

Our PTAS for Steiner VRAP extends to higher dimensions analogously to Arora's PTAS for TSP [4]. In particular, Lemma 21 extends similarly to Lemma 3 in [4] to any dimension $d$, yielding a complexity of $\mathcal{O}(n \log^{\xi(d,\varepsilon)} n)$ with $\xi(d,\varepsilon) = \mathcal{O}(\sqrt{d}/\varepsilon)^{d-1}$. As these extensions are completely along the lines of [4], we omit the details here.

Lastly, our algorithms can be trivially derandomized by enumerating all $\mathcal{O}\left(n^d\right)$ choices for the initial random shift of the zoom, respectively quad tree.

**Acknowledgment.** The authors would like to thank the anonymous referees for their thorough review and helpful suggestions.

## References


[1] P. K. Agarwal. Range searching. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*. CRC Press, Boca Raton, FL, 2nd edition, 2004.

[2] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pages 1–56. American Mathematical Society, Providence, RI, 1999.

[3] A. Armon, A. Avidor, and O. Schwartz. Cooperative TSP. In *Proceedings of the 14th Annual European Symposium on Algorithms*, pages 40–51, 2006.

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

[5] S. Arora, P. Raghavan, and S. Rao. Approximation schemes for Euclidean $k$-medians and related problems. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 106–113, 1998.

[6] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proceedings of the 27th annual ACM Symposium on Theory of Computing*, pages 489–498, 1995.

[7] J. E. Beasley and E. Nascimento. The vehicle routing allocation problem: a unifying framework. *TOP*, 4:65–86, 1996.

[8] L. R. Foulds, S. W. Wallace, J. Wilson, and L. Sagvolden. Bookmobile routing and scheduling in Buskerud County, Norway. In *Proceeding of the 36th Annual Conference of the Operational Research Society of New Zealand*, pages 67–77, 2001.

[9] M. Garey, R. Graham, and D. Johnson. Some NP-complete geometric problems. In *Proceedings of the 8th Annual ACM Symposium on Theory of Computing*, pages 10–22, 1976.

[10] J. Gudmundsson, C. Levcopoulos, and G. Narasimhan. Improved greedy algorithms for constructing sparse geometric spanners. In *Proceedings of the 7th Scandinavian Workshop on Algorithm Theory*, pages 163–174, 2000.

[11] S. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the euclidean $k$-median problem. In *Proceedings of the 7th Annual European Symposium on Algorithms*, pages 378–389, 1999.

[12] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, $k$-MST and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[13] S. B. Rao and W. D. Smith. Approximating geometrical graphs via 'spanners' and 'banyans'. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 106–113, 1998.

[14] J. Remy, R. Spöhel, and A. Weißl. On Euclidean vehicle routing with allocation. In *Algorithms and Data Structures*, volume 4619 of *Lecture Notes in Comput. Sci.*, pages 601–612. Springer, Berlin, 2007.

[15] J. Tipping. Scheduling and routing grass mowers around Christchurch. In *Proceeding of the 37th Annual Conference of the Operational Research Society of New Zealand*, 2002.

[16] L. Vogt, C. A. Poojari, and J. E. Beasley. A tabu search algorithm for the single vehicle routing allocation problem. *Journal of the Operational Research Society*, 58:647–480, 2007.