A Constant-Factor Approximation Algorithm for TSP with Neighborhoods in the Plane*

Joseph S. B. Mitchell†

Abstract

In the Euclidean TSP with neighborhoods (TSPN) problem we seek a shortest tour that visits a given set of n *neighborhoods*. The Euclidean TSPN generalizes the standard TSP on points.

We present the first constant-factor approximation algorithm for TSPN on an arbitrary set of disjoint, connected neighborhoods in the plane. Prior approximation bounds were $O(\log n)$, except in special cases. Our approximation algorithm applies to arbitrary connected neighborhoods of any size or shape.

1 Introduction

The traveling salesperson problem with neighborhoods (TSPN) is a generalization of the classic TSP in which the traveler is to visit a set of neighborhoods rather than a set of specific points. We focus primarily on the case in which the neighborhoods are *connected* regions in the plane.

The TSPN was introduced by Arkin and Hassin [1], who gave the first algorithmic study of the problem and provided the first approximation algorithms for special cases of the problem. The problem arises naturally in in various optimal covering tour problems, network design problems, relay placement in sensor networks, robotics mission planning, and robot localization.

Our main result is a polynomial-time algorithm that gives the first constant-factor approximation for the TSPN for the general case of disjoint connected neighborhoods in the plane. The regions can be of arbitrary size and shape. In contrast, the best approximation ratio previously known for the problem is $O(\log n)$. Only special cases in which the regions are fat and/or of nearly the same size (diameter) were known previously to have constant-factor approximation algorithms or a polynomial-time approximation scheme (PTAS).

1.1 Related Work The TSP with neighborhoods has been actively studied since its first formalization and study by Arkin and Hassin [1].

The TSPN is a generalization of the classic geometric TSP on points and is therefore NP-hard. While the TSP on point sets in geometric domains admits a PTAS by the results of Arora [2], Mitchell [15, 14], and Rao and Smith [19], the TSP with neighborhoods seems to be a considerably harder optimization problem to approximate.

Dumitrescu and Mitchell [7] give a PTAS, based on the m-guillotine method, if the regions are all about the same size (the ratio of largest to smallest diameter is bounded), have bounded depth (no point lies in more than a constant number of regions), and are *fat*. Feremans and Grigoriev [10] give a PTAS, based on Arora [2], if the regions are disjoint fat polygons of comparable size in the plane; their algorithm applies also in higher dimensions.

*Partially supported by grants from the National Science Foundation (CCF-0729019), Metron Aviation, and NASA Ames.
†Stony Brook University, Stony Brook, NY 11794-3600

By mapping to the "one-of-a-set" TSP, de Berg et al. [5] gave an $O(1)$-approximation for disjoint fat convex regions. Elbassioni et al. [8] improved the running time and generalized to the case in which the neighborhoods to be visited may be arbitrary sets of points, with each set lying within a (disjoint) fat region that is not necessarily convex. Elbassioni, Fishkin, and Sitters [9] consider the case in which the corresponding regions are intersecting, convex, and fat, of comparable size; they give an $O(1)$-approximation.

Mitchell [18] obtained a PTAS for bounded depth fat regions in the plane, allowing them to be of arbitrary sizes. His definition of "fat" is very weak; he requires only that the area is at least a constant times the square of the diameter. His results apply also to the case in which the regions to be visited may be disconnected sets of points that each lie within one of the (nearly) disjoint fat polygons.

Very recently, Chan and Elbassioni [4] have given a quasipolynomial-time approximation scheme (QPTAS) for geometric instances of TSPN in any fixed dimension for the case of *fat, weakly disjoint* regions; their results apply also to metric spaces of bounded doubling dimension.

If the regions are allowed to overlap, the problem is APX-hard [5, 20], even if the regions are segments all of very nearly the same length [9]. The problem is also APX-hard if the regions are disconnected, with each being a pair of points in the plane [6]. Thus, we do not expect to find a PTAS for the general case. There might be a PTAS in the case of disjoint regions (the case for which we give the first constant-factor approximation); this is a fascinating open problem.

For connected sets in the plane, arbitrarily overlapping, the best known approximation ratio is $O(\log n)$, given in an algorithm discovered more than 15 years ago [13]; the running time has been improved more recently [9, 11].

For further background about the geometric TSP, the TSP with neighborhoods, and related problems, see the surveys [3, 16, 17].

1.2 Our Contribution, Outline of Our Approach All prior constant-factor approximation algorithms (and PTAS's) relied on exploiting special structure that comes from assuming that regions are either all about the same size or all are fat in some sense. Fatness was crucial in order to apply a packing argument to give effective lower bounds on the length of an optimal tour. Even the special case of horizontal line segments has had no constant-factor approximation prior to this work.

Several new ideas are needed in order to address the general case in which regions can be of arbitrary sizes and are not fat. We outline our approach:

1. We address the "skinniness" of general regions by surrounding each region by its four "directional hulls", which we prove are fat, for an appropriate choice of one of *two* coordinate systems. This allows us to partition the regions into two sets ("blue" regions and "red" regions); we separately find approximating tours of each set, and then appeal to the Combination Lemma of [1] to combine into a tour of the entire set. Thus, it suffices to consider only the set of blue regions, which have all four of their directional hulls in our (x, y)-coordinate system fat.

2. While the original regions are assumed to be disjoint, the fat directional hulls of the regions can overlap. Thus, we start by selecting a disjoint subset, $\mathcal{E}_0$, of the directional hulls; we do this greedily, in order of increasing size. We use the algorithm of [18] to compute an approximately optimal tour, T, of the disjoint subset $\mathcal{E}_0$ of (fat) directional hulls.

3. We convert T into a polygonal subdivision, G, each of whose faces are *histograms*; this

increases the total length of T by only a constant factor.

4. We argue that if all four of the (fat) directional hulls of the original (possibly skinny) region P_i is visited by G, then G also visits P_i. This follows from the fact that the faces of G are histograms. (Thus, if G happens to visit all of the directional hulls of all of the (blue) regions, then G already visits all of the regions themselves, and we are done.)

5. The fact that the regions $\mathcal{E}_0$ were selected greedily by size allows us to argue that any region P_i that is *not* visited already by T must be *close* to the tour T (and therefore close to the boundaries of faces of G), where "close" means within distance that is $O(diam(P_i))$. (I.e., there can be no "missed" regions P_i that are substantially interior to some face of G.) This motivates the definition of a *stratified grid* within a histogram face, H; this grid has small squares near the boundary of H, then progressively larger squares as the distance to the boundary increases. Since regions P_i that are "far" from the boundary of H must be "large", we know that we are able to visit all regions with a slight refinement of the squares of the stratified grid. Also, each square of the stratified grid can be attached to the boundary of H by a connection whose length is proportional to the size of the square.

6. We then focus on a single histogram face, H, and its stratified grid. We first argue a lower bound on the total length of any forest F that visits all regions P_i interior to H and for which the union $F \cup \partial H$ is connected: Such a forest must have length at least a constant fraction of the weight of a minimum-weight cover of the regions P_i by squares of the stratified grid, where "weight" refers to the sum of the sizes of the squares. Thus, our goal is to solve (or approximate) a weighted set cover problem that has special structure. We are able to exploit the disjointness of the regions P_i to obtain a dynamic programming algorithm to compute a minimum-weight cover.

2 Preliminaries

Let $\mathcal{R} = \{P_1, P_2, \ldots, P_n\}$ be the input set of n disjoint, connected regions in the plane.

We say that a tree or a tour T *visits* region P_i if $T \cap P_i \neq \emptyset$. We say that T visits $\mathcal{R}$ if T visits each of the regions P_i, for $i = 1, \ldots, n$. Let let L^* denote the length of an optimal tree/tour.

We utilize some basic notation and observations from [18]. Let R_0 be a minimum-diameter (blue) axis-aligned rectangle that intersects or contains all regions P_i, and let D be the diameter of R_0. (Note that R_0 is easily computed in polynomial time by critical placement arguments; for our purposes, a constant-factor approximation suffices, and this is even more readily computed.) Then, D provides a lower bound on the length, L^*, of an optimal tour; as shown in [18], Lemma 2.2, $2D \leq L^* \leq nD$. Let $\mathcal{G}$ denote the regular grid (lattice) of points $(i\delta, j\delta)$, for integers i and j, where $\delta = D/n$. We let Γ_i denote the subset of grid points $\mathcal{G}$ at distance at most $\delta/\sqrt{2}$ from region P_i. Then, $\Gamma_i \neq \emptyset$. Note that it may be that $\Gamma_i = \Gamma_j$ for distinct regions P_i and P_j, $i \neq j$. (In particular, many "tiny" regions may all map to the same singleton grid point.) By Lemma 2.3 of [18] (with $\epsilon = 1$), we know that any tour T (of length L) that visits $\mathcal{R} = \{P_1, P_2, \ldots, P_n\}$ can be modified (by adding n detours, each of length at most $2\delta/\sqrt{2} = \sqrt{2}D/n \leq L^*/(\sqrt{2}n)$) to be a tour $T_{\mathcal{G}}$, of length $O(L)$, that visits $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$. Similarly, any tour $T_{\mathcal{G}}$, of length $L_{\mathcal{G}}$, that visits $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$ can be modified to be a tour T, of length $O(L_{\mathcal{G}})$, that visits $\mathcal{R} = \{P_1, P_2, \ldots, P_n\}$. Further, by Lemma 2.4 of [18], it suffices to search for optimal tours within the ball, $B(c_0, 2D)$, of radius $2D$ centered at the center point, c_0, of R_0; this *localizes* our problem.

Thus, for our approximation purposes, it suffices to consider polygonal tours whose vertices lie on the grid $\mathcal{G}$ (and within distance $O(D)$ of the points c_0); in fact, giving up at most a factor $\sqrt{2}$, we can assume, for simplicity, that we are searching over the set of rectilinear tours on $\mathcal{G}$, with all edges axis-parallel, on the grid. By rescaling, we can also assume that $\delta = 1$, without loss of generality.

Finally, it suffices to give an approximation algorithm for the *minimum spanning tree with neighborhoods* (MSTN) problem, since associated with a spanning tree T of length $|T|$ is a tour of length $2|T|$ obtained by walking around the tree T. Thus, we desire a tree, T, that visits all of the regions $\mathcal{R} = \{P_1, P_2, \ldots, P_n\}$ and has (Euclidean) length close to $L^* = |T^*|$, the length of an optimal spanning tree, T^*. We can assume that the tree is rectilinear, lies on the integer grid, and the goal is to find a short tree T that visits, or comes close (within distance $1/\sqrt{2}$) to, each region P_i.

3 Reduction to Case of Overlapping Fat Regions

We consider two cartesian coordinate systems: (1) The "base" (or "blue") system has coordinates (x, y), and (2) the "rotated" (or "red") system is rotated by 45 degrees with respect to the blue system and has coordinates (x', y'). Each region P_i has an axis-aligned bounding box in each of the two coordinate systems; we let $BB(P_i)$ denote the blue bounding box and $BB'(P_i)$ denote the red bounding box.

We let $E^{+x}(P_i) \subseteq BB(P_i)$ denote the set of all points $p \in BB(P_i)$ for which the rightwards ray (in the $+x$ direction) from p intersects P_i. We similarly define $E^{-x}(P_i)$, $E^{+y}(P_i)$, and $E^{-y}(P_i)$, and refer to these as the *blue directional hulls* of P_i. Exactly analogously, we define the four red directional hulls of P_i, with respect to the red coordinates x', y' and the red bounding box, $BB'(P_i)$. Refer to Figure 1.

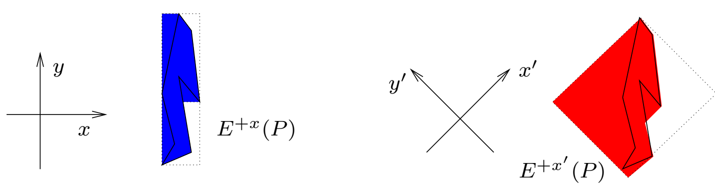

Figure 1: Directional hulls: blue $E^{+x}(P)$ and red $E^{+x'}(P)$ for polygon P.

We say that a connected planar region X is *α-fat* (or simply *fat*) if X contains a disk of radius at least $diam(X)/\alpha$, where $diam(X)$ denotes the diameter of X. Note that this definition of fat applies to convex as well as nonconvex regions.

LEMMA 3.1. *Let X be a connected subset of the plane. Either the four blue directional hulls of X are fat or the four red directional hulls of X are fat.*

Proof. If the four blue directional hulls are fat, we are done. Thus, assume that at least one of the blue directional hulls, say $E^{-y}(X)$, is not fat.

We claim that all four red directional hulls are fat.

There are two cases: (1) The aspect ratio of $BB(X)$ is at least 10; and (2) The aspect ratio of $BB(X)$ is less that 10. (The number 10 is not specially significant or optimally chosen, but it suffices for our purposes.)

In case (1) we assume, without loss of generality, that $BB(X)$ has its longer sides horizontal (in the (x, y)-coordinate system), and we partition $BB(X)$ into a chain of 10 subrectangles by partitioning the longer dimension of $BB(X)$ into 10 equal-length pieces. By the fact that $BB(X)$

is a minimal bounding box, we know that there must be a point $a \in X$ within the first subrectangle and a point $b \in X$ within the last (10th) subrectangle of the chain. (In fact, by connectivity, each subrectangle has a connected portion of X within it.) We readily see that the red bounding box $BB'(\{a,b\})$ extends enough outside $BB(X)$ to contain the (x',y')-aligned square, A, of size $diam(X)/10$ in the upper corner of $BB'(\{a,b\})$; similarly, there is a square B of size $diam(X)/10$ in the lower corner of $BB'(\{a,b\})$. Refer to Figure 2. Since X is connected, there is a path, γ, within X (and therefore within $BB(X)$ and $BB'(X)$) connecting a and b. We see that $A \subset BB'(\{a,b\}) \subseteq BB'(X)$, and, by definition of directional hulls, $A \subset E^{-y'}(X)$ and $A \subset E^{-x'}(X)$; similarly, $B \subset BB'(\{a,b\}) \subseteq BB'(X)$, and $B \subset E^{+y'}(X)$ and $B \subset E^{+x'}(X)$. Thus, $BB'(X)$ has all four of its red directional hulls fat (containing A or B, each of which has a constant fraction fraction of the area of $BB'(X)$).

In case (2), we again assume, without loss of generality, that $BB(X)$ has its longer sides horizontal (in the (x,y)-coordinate system). We partition the longer dimension of $BB(X)$ into 10 equal-length pieces, and consider the 10 corresponding squares within $BB(X)$ along the bottom edge of $BB(X)$. Refer to Figure 2. Since $E^{-y}(X)$ is not fat, we know that none of these squares is disjoint from X (otherwise, such a square would lie inside $E^{-y}(X)$, contradicting the assumption that $E^{-y}(X)$ is not fat). Thus, we know that there must be a point $a \in X$ within the leftmost square and a point $b \in X$ within the rightmost square. While the red bounding box $BB'(\{a,b\})$ extends enough below $BB(X)$ to contain the (x',y')-aligned square, B, of size $diam(X)/10$ in the lower corner of $BB'(\{a,b\})$, the similar square, A, in the upper corner of $BB'(\{a,b\})$ may lie partly or entirely within $BB(X)$. We have $B \subset BB'(\{a,b\}) \subseteq BB'(X)$, $B \subset E^{+y'}(X)$ and $B \subset E^{+x'}(X)$; thus, $E^{+y'}(X)$ and $E^{+x'}(X)$ are fat. We need a different argument to show that $E^{-y'}(X)$ and $E^{-x'}(X)$ are fat, since it may be that X enters square A, so it is not immediate that A is contained in $E^{-y'}(X)$ and $E^{-x'}(X)$. Consider a path γ within X from b to a. Such a path must at some point cross the polygonal path (p,q,r) shown in Figure 2. (Here, pq lies on the vertical (y-parallel) line supporting A on its right; qr lies on the y'-parallel line supporting A from below.) If the path γ hits segment pq before segment qr, then γ is a witness to the containment of square C within $E^{-y}(X)$, a contradiction to the fact that $E^{-y}(X)$ is assumed not to be fat. Otherwise, if γ hits segment qr before (or at the same point as) it hits segment pq, then γ is a witness to the containment of A in the red directional hull $E^{-y'}(X)$, showing that $E^{-y'}(X)$ is fat. A similar argument shows that $E^{-x'}(X)$ is fat. □

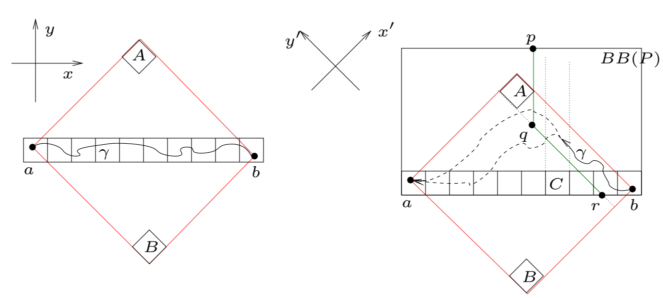

Figure 2: Proof of Lemma 3.1. Left: Case (1), aspect ratio is high; Right: Case (2), aspect ratio is low.

The next lemma shows that the problem of visiting a set of arbitrary connected regions can be reduced to the problem of visiting a set of fat regions (that may overlap, even of the original regions are disjoint).

LEMMA 3.2. *Let T be a tree that visits all four of the blue directional hulls of all n of the regions, $\mathcal{R} = \{P_1, P_2, \ldots, P_n\}$. Then there exists a connected planar network, $G \supseteq T$, of length $|G| = O(|T|)$ that visits all of the regions $\mathcal{R}$.*

Proof. We can assume that T is rectilinear (with axis-parallel edges). First, we add to T the boundary of the (axis-aligned) bounding box, $BB(T)$, of tree T. The set $BB(T) \setminus T$ consists of a set of simple rectilinear polygonal *pockets*, each with at least one edge (the pocket "lid") on the boundary of $BB(T)$.

For each pocket, Q, we build its *histogram decomposition*, as follows. Let e be one edge of Q; for specificity, we can take e to be a lid of pocket Q, and assume that e is horizontal. We illuminate edge e within Q, using *rectilinear visibility*: the set $VP(e,Q)$ of all points in Q that are rectilinearly visible from e is the set of all points $q \in Q$ for which the axis-parallel segment, $qq_\perp$, joining q to the point $q_\perp$ that is the orthogonal projection of q onto the line through e, satisfies $q_\perp \in e$ and $qq_\perp \subset Q$. The set $VP(e,Q)$ is a rectilinear histogram with base e. The boundary of $VP(e,Q)$ consists of segments contained in the tree T and chord segments, known as *windows*, that pass through the interior of Q. Each window segment w is a chord separating $VP(e,Q)$ from a a subpolygon ("subpocket") of Q. We then consider each window w, and the corresponding subpocket Q', and determine the histogram, $VP(w,Q')$, consisting of the points within Q' (and therefore within Q) that are rectilinearly visible from w. We continue this process until each subpocket has no windows (only edges contained in T). This process is a rectilinear version of the "window partition tree" used to compute link distances in polygons [21]. It was used also in [13], where it was argued that the total length of the resulting histogram decomposition is $O(|T|)$, where $|T|$ denotes the total length of tree T: The key observation is that any axis-parallel chord of Q crosses at most two windows; thus, the length of each window (the edges that were added to T to make the decomposition) can be charged to the length of T.

Next we argue that the connected planar network, G, given by the union of the histogram decompositions of the pockets of T, visits all of the original regions, $\mathcal{R}$. Suppose to the contrary that P_i lies fully interior to a (histogram) face, H, of G. (We assume here that H is a bounded face, not the face at infinity; however, the argument below applies even more readily to the case that H is the face at infinity, i.e., the complement of the rectangle $BB(T)$.) Suppose that H is a vertical histogram. Since T visits all four of the blue directional hulls of P_i, we know that G (whose edges are a superset of T) visits the hull $E^{+y}(P_i)$ in particular.

If some point $p \in E^{+y}(P_i)$ lies left of the leftmost (vertical) side of H, then, by definition of $E^{+y}(P_i)$, the upwards ray from p intersects P_i at some point $p' \in P_i$; thus, p' lies to the left of the bounding box of H, so p' is a point of P_i outside H, a contradiction. Similarly, we know that no part of $E^{+y}(P_i)$ lies to the right of the rightmost (vertical side of H.

If the bottom edge (base) of $E^{+y}(P_i)$ lies below the bottom edge (base) of H, then some point of P_i that supports the base of $E^{+y}(P_i)$ lies outside H, a contradiction.

Finally, if some point $p \in E^{+y}(P_i)$ lies above the top (x-monotone) boundary of H, then, by definition of $E^{+y}(P_i)$, the upwards ray from p intersects P_i at some point $p' \in P_i$; thus, p' lies above the top boundary of H, so p' is a point of P_i outside H, a contradiction. Thus, $E^{+y}(P_i)$ lies completely below the top boundary of H. Since $E^{+y}(P_i)$ also lies completely above the base of H and to the right of the left side of H and to the left of the right side of H, we conclude that $E^{+y}(P_i)$ lies interior to H, a contradiction to the fact that G visits $E^{+y}(P_i)$. □

Based on the above lemmas, we can suggest an approach to computing a constant-factor approximation for an arbitrary set of connected regions $\mathcal{R}$ in the plane. First, we partition the regions $\mathcal{R}$ into two sets – the "blue" regions that all four blue directional hulls fat, and the remaining "red" regions, which, by Lemma 3.1, must have all four red directional hulls fat. Then we compute a tree, T_{blue}, visiting all of the (fat) blue directional hulls of the blue regions and separately a tree, T_{red}, visiting all of the (fat) red directional hulls of the red regions. By Lemma 3.2, we know that T_{blue} visits all of the blue regions and T_{red} visits all of the red regions. We can then add a segment of length $O(D)$ (and recall that $L^* \geq 2D$) connecting T_{blue} and T_{red}, forming a single tree that visits all of the regions. (Alternatively, we can appeal to the Combination Lemma of [1] to combine the two solutions.) If we have a constant-factor approximation algorithm for computing a short tree spanning a set of (overlapping) fat regions, then this approach yields an overall constant-factor approximation for MSTN (and therefore for TSPN). In fact, this approach applies to arbitrarily overlapping regions $\mathcal{R}$; we made no assumption about disjointness. We summarize:

Proposition 3.1. *A c-approximation for TSPN/MSTN for arbitrary connected fat regions in the plane yields an $O(c)$-approximation for TSPN/MSTN for arbitrary connected regions in the plane.*

The difficulty with the above approach is that we do not yet know how to give a constant-factor approximation for TSPN on a set of arbitrarily overlapping fat regions. Thus, we take a different approach. We *do* know how to compute a nearly optimal tree/tour of a set of *disjoint* fat regions (even if the regions are not disjoint, but have bounded depth); in fact, a PTAS is given in [18]. Thus, we will compute such an approximating tree/tour T of an appropriately chosen subset of *disjoint* fat regions, and then augment T so that it visits all of those original regions that T does not already visit. Since we can afford to add to T its histogram decomposition, and still have total length $O(|T|)$, it suffices to solve the augmentation problem for regions that lie within a single face (histogram), H, of the decomposition.

It is important for our approach that we have additional structure associated with the "floating" regions that lie interior to faces of the histogram decomposition. This structure comes from the method we use to select the set of disjoint fat regions. We now describe that method.

Let $\mathcal{E} = \{E_1, E_2, \ldots\}$ be the set of fat blue directional hulls of the input regions $\mathcal{R}$. (Recall that we are going to assume that the regions $\mathcal{R}$ are disjoint; however, the fat blue directional hulls of $\mathcal{R}$ may overlap.) We pick a subset $\mathcal{E}_0 \subseteq \mathcal{E}$ of *disjoint* fat regions by an iterative algorithm, examining the regions E_i in increasing order of diameter, $diam(E_i)$ (which matches the diameter of the input region of which E_i is a directional hull). As we examine region E_i, if it is disjoint from the regions already placed in the set $\mathcal{E}_0$, then we add it to $\mathcal{E}_0$; otherwise, we skip it and go to the next larger region. (In case of ties, we examine the regions of equal diameter in arbitrary order.)

Next, we compute a tree (or tour), $T_{\mathcal{E}_0}$, visiting the regions $\mathcal{E}_0$. For this, we can use the PTAS of [18]. We know that the length, L^*, of an optimal tree/tour of $\mathcal{R}$ is at least the length of an optimal tree/tour of $\mathcal{E}_0$. We then extend $T_{\mathcal{E}_0}$ to a histogram decomposition, G, of the bounding box and pockets of $T_{\mathcal{E}_0}$.

Now, the remaining problem is that of augmenting G so that the augmented graph, $\overline{G}$, is connected and visits *all* of the regions $\mathcal{R}$. It suffices to consider one face, H, in the histogram decomposition G, and the set $\mathcal{R}_H$ of (input) regions that lie interior to H. We will exploit special structure of the set of regions $\mathcal{R}_H$ with respect to the boundary of H.

Lemma 3.3. *Each $P \in \mathcal{R}_H$ lies within distance $diam(P)$ of the boundary, ∂H, of H.*

Proof. Let $P \in \mathcal{R}_H$. Since at least one of the four directional hulls, E, of P was not put in the set $\mathcal{E}_0$ during the selection process, there must be another region $E' \in \mathcal{E}_0$ with diameter at most $diam(P) = diam(E)$ and with $E \cap E' \neq \emptyset$. Since $T_{\mathcal{E}_0}$ visits E', $T_{\mathcal{E}_0}$ comes within distance $diam(E') \leq diam(P)$ of E, and therefore of P. Thus, P lies within distance $diam(P)$ of ∂H. □

In the next section, we give an algorithm to solve (approximately) the *histogram problem*, which is the following: For a histogram H and a set $\mathcal{R}_H$ of connected, disjoint regions interior to H, with each $P \in \mathcal{R}_H$ within distance $diam(P)$ of ∂H, construct a minimum-length network, F, such that $F \cup \partial H$ is connected and visits all of the regions $\mathcal{R}_H$.

4 Solving the Histogram Problem

Let $\mathcal{R}_H$ be a set of disjoint regions interior to a histogram H. As discussed earlier, we can assume that the vertices of H are integral (on the grid $\mathcal{G}$). Also, without loss of generality, H is a vertical histogram. We make the *closeness assumption*: each $P \in \mathcal{R}_H$ lies within distance $diam(P)$ of ∂H. Our goal is to construct a minimum-length network of edges, F, such that $F \cup \partial H$ is connected and visits all of the regions $\mathcal{R}_H$. If F is truly of minimum-length, it is a forest (since any cycle can be cut); the network we construct for our approximation will not, in general, be a forest, but we still often refer to the minimization problem in terms of forests and we let F^* denote an optimal forest, with length $|F^*|$.

4.1 The Stratified Grid We now define the *stratified grid* decomposition of H. Informally, the decomposition is formed as follows: We place squares of size 1 (forming the first strata, S_1) inside H, around its boundary; this results in an "eroded" (shrunk by L_∞ distance 1) region (union of histograms). We then place squares of size 2 (forming the second strata, S_2) inside each component of the eroded region, around its boundary; this results in the eroded region getting smaller by L_∞ distance 2. This process continues until all of H has been packed (and covered) by squares of sizes 1, 2, 4, 8, etc., forming the strata $S_1, S_2, S_4, \ldots$. This description is informal since it does not take into account the situation in which the eroded region may contain narrow "necks" of insufficient width to accommodate squares of the appropriate size for the next strata of erosion. Thus, we make the process more formal with the following definitions. (There are multiple ways to establish stratified grids that work equally well to the method we describe here; we will point out below the basic properties that we need the stratification to have.)

The *standard squares* of size 2^i (or *standard 2^i-squares*) are those squares of size 2^i whose defining coordinates are integer multiples of 2^i ($i = 0, 1, 2, \ldots$).

The stratified grid decomposition of H is a polygonal subdivision of H into standard squares of sizes 1, 2, 4, 8, etc; we let S_i denote the set of squares of size i. We abuse notation slightly by using S_i also to denote the region within H that is the union of the squares of S_i.

Let H_{-1} denote the 1-offset of H, consisting of the set of all points within H that are at L_∞ distance at least 1 from the boundary of H. (Note that H_{-1} may consist of more than one connected component, each of which is a histogram with integral coordinates.) Let $H_{-1}^{(2)}$ denote the union of all standard 2-squares within H_{-1}. (Note that $H_{-1}^{(2)}$ is a union of histograms whose vertex coordinates are multiples of 2; thus, $H_{-1}^{(2)}$ can be exactly tiled by standard 2-squares.) We define the *1-strata*, S_1, to be the set of all (standard) 1-squares that lie within $H \setminus H_{-1}^{(2)}$. Note that each 1-square of S_1 is either in contact with the boundary of H or at L_∞ distance 1 from the boundary of H.

We define H_{-2} to be the 2-offset of $H_{-1}^{(2)}$, and $H_{-2}^{(4)}$ to be the union of all standard 4-squares within H_{-2}. The *2-strata*, S_2, is the set of all standard 2-squares that lie within $H_{-1}^{(2)} \setminus H_{-2}^{(4)}$. We similarly define H_{-2^i} and the regions $H_{-2^i}^{(2^{i+1})}$ for $i = 0, 1, 2, \ldots$; the 2^i-strata, S_{2^i}, is the set of all standard 2^i-squares that lie within $H_{-2^{i-1}}^{(2^i)} \setminus H_{-2^i}^{(2^{i+1})}$. (Note that both $H_{-2^{i-1}}^{(2^i)}$ and $H_{-2^i}^{(2^{i+1})}$ can be exactly tiled by standard 2^i-squares.) An example of a stratified grid of H is shown in Figure 3.

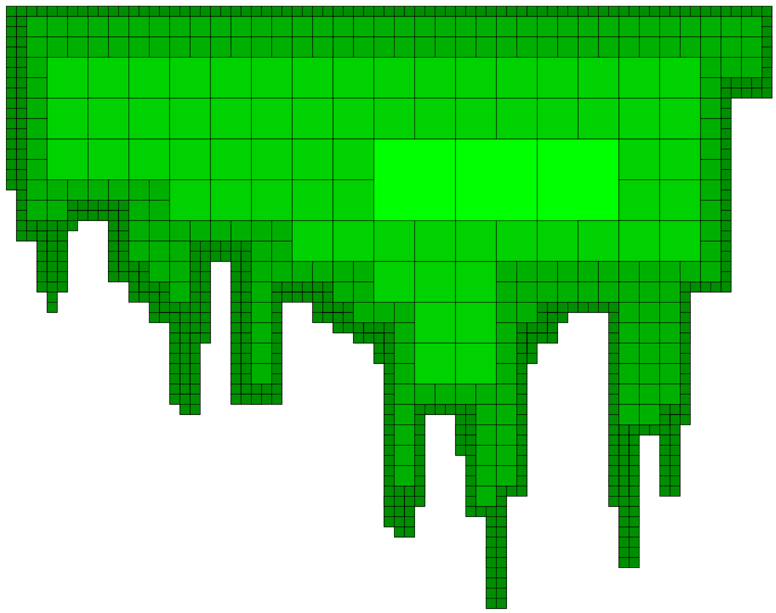

Figure 3: The stratified grid of a histogram H.

By the specification of the stratified grid, each grid square σ of size $|\sigma| = 2^i$ lies at a distance $\Theta(|sigma|)$ from ∂H, (More precisely, the distance is between $2^i - 1$ and $3 \cdot 2^i - 2$.)

We subdivide each grid square of the stratified grid into a constant number of subsquares. This is done so that no region $P \in \mathcal{R}_H$ can lie substantially inside any square (after subdivision), where "substantially" means at distance more than $1/\sqrt{2}$ (recall that we only need to get within distance $\delta/\sqrt{2}$ of each region, and we rescaled so that $\delta = 1$). There is no need to subdivide the finest squares (the squares of size 1 or 2). For all other squares σ, it suffices to partition σ with a "+" *mullion* into 4 equal-size "panes" (subsquares), each of half the size. We consider these mullions to be associated with each square σ having $|\sigma| > 1$. Then, since σ is at distance at least $|\sigma| - 1$ to ∂H, and we know that any $P \in \mathcal{R}_H$ is at distance at most $diam(P)$ from ∂H, we know that, while P may lie inside a square σ, it cannot avoid being intersected by the mullions of σ, since each subsquare has diameter $|\sigma|/\sqrt{2}$, and we know that $|\sigma|/\sqrt{2} < |\sigma| - 1$ for sizes $|\sigma| \geq 4$.

4.2 A Lower Bound on OPT We establish a lower bound on the length, $|F^*|$, of a minimum-length forest, F^* whose addition to ∂H spans all of the regions, with $F^* \cup \partial H$ being connected. We argue that we can replace the forest F^* with the network consisting of the boundaries of the set S_{F^*} of all grid squares that F^* intersects, and that this network has length $O(F^*)$. This is a nontrivial statement and utilizes the special structure of the stratified grid, since, in a general tiling by squares of different sizes, a curve of length ℓ can intersect a set of n squares whose total perimeter is $\Omega(|\ell| \log n)$; see the example in [18].

Lemma 4.1. *The total perimeter of the stratified grid squares in the set S_γ of squares intersected by a curve $\gamma \subset H$ of length $|\gamma|$ is $O(|gamma|)$ plus the sizes of the squares containing γ's endpoints.*

Proof. Suppose γ's starting endpoint, g_0, lies in a grid square of size 2^i (within strata $S^{(0)} = S_{2^i}$). As we traverse γ, starting at g_0, we either get to the ending point of γ without ever leaving the initial strata, $S^{(0)}$, or we enter a second, neighboring strata, $S^{(1)}$ (which is either $S_{2^{i-1}}$ or $S_{2^{i+1}}$). If we stay within a single strata, then our result follows immediately by standard packing arguments (a curve of length ℓ can intersect only $O(|\ell|)$ unit disks). Thus, assume we change strata. Then, γ stays within these two strata for some time, until we either reach the endpoint (in which case we are again done, since we have encountered only two different sizes of squares), or we enter a new strata, $S^{(3)}$. We mark this first entry point, g_3 along γ. Then, γ continues within the two strata $S^{(2)}$ and $S^{(3)}$, possibly zig-zagging back and forth, until it finally reaches a new strata $S^{(4)}$ (adjacent to $S^{(2)}$ or to $S^{(3)}$) for the first time at point g_4, etc. The standard packing argument applies to the portion of γ from g_3 to g_4, since the curve lies within squares of only two different sizes (one being twice the size of the other). Further, we know that γ had to traverse a distance of at least the size of one of the squares in the smaller strata (which is within a factor two the same as the size of squares in the other strata). Thus, the sum of the sizes of the squares traversed by γ between point g_j and g_{j+1} is at most $O(|\gamma_{j,j+1}|)$ (the length of the subcurve), *plus* the sizes of the squares at g_j and at g_{j+1}, which we know is also $O(|\gamma_{j,j+1}|)$, since $\gamma_{j,j+1}$ had to cross one of the two strata. Thus, summing along γ, we get the claimed bound. □

4.3 Subproblems We say that a square σ of the stratified grid is (locally) *maximal* if it and any of its neighboring grid squares of size $|\sigma|$ are not adjacent to any grid squares of size $2|\sigma|$.

The following lemma justifies that we can afford to add to our network, F, the boundaries (and mullions) of the maximal squares of the stratified grid, since their length is $O(|\partial H|)$, and we know (Lemma 3.2) that the total length of the histogram decomposition, G, is $\sum_H |\partial H| = O(L^*)$.

Lemma 4.2. *The sum of the sizes of all maximal squares is $O(|\partial H|)$.*

Proof. This follows from a fairly standard charging argument (deferred to the full paper), in which the boundary of each maximal square is charged off to the boundary of H (such that no portion of ∂H is charged more than $O(1)$ times). □

In addition to the boundaries of maximal squares (and their mullions), we add to our network F a set of *spokes*, one for each vertex of each maximal square, each of which is a segment joining the square to ∂H. Since each square σ is within distance $O(|\sigma|)$ of ∂H, the total length of the spokes is also $O(|\partial H|)$, so we can afford it. By using shortest connections to ∂H, we can assure that the spokes are noncrossing (since crossings can be shortcut).

Each grid square, $\sigma \in S_i$, of the grid can be joined by a shortest (Euclidean) segment, L_σ (the "stem"), to ∂H. We know that $|L_\sigma| = O(|\sigma|)$. (Stems do not cross.) Each grid square σ can also be associated with a maximal square, M_σ, that is closest to it; let U_σ be a shortest segment (an "arm") joining σ to a maximal square. Then, for each maximal square, Σ, there is a set, $\mathcal{S}_\Sigma = \{\sigma : M_\sigma = \Sigma\}$, of grid squares associated with it. This allows us to partition the grid squares into *zones*, corresponding to sides of maximal squares Σ. A zone is bounded by a side of a maximal square, the boundary of H, and two spokes. The zones partition H; we have now to augment our network F (which by now includes maximal square boundaries and their mullions and spokes) within each zone, so that all regions $P \in \mathcal{R}_H$ that lie within a zone Z are visited by the network.

For a given zone Z, our goal is to find a minimum-weight *covering* set of grid squares within Z that is covering in the sense that all regions P interior to Z are intersected by the square (and therefore by its boundary or mullions) or its stem that joins it to ∂H; the weight is the sum of the square sizes, which is proportional to the sum of the perimeters, lengths of mullions, and lengths of stems.

For this optimization problem, we use a dynamic programming algorithm. Our approach is similar to that of Katz et al. [12], who study the following covering problem: Given a set of downward-pointing rays and a set of disjoint line segments in the plane, each of which is stabbed by at least one ray, find a minimum-cardinality set of rays that stab all line segments. (In fact, our results here generalize and extend the results of [12] to the problem of covering a set of disjoint *polygons* with a set of downwards-point rays.) The stemmed squares (squares with their associated stems), serve as the set of "rays" that are "downward" in the sense that the stems point to points that get closer and closer to ∂H.

We need a bit more notation. Let $\mathcal{R}_\sigma$ denote the set of regions intersected by $\sigma \cup L_\sigma$. Let r_σ be the "rightmost" point of $\partial\sigma \cup \mathcal{R}_\sigma$, and let P_σ denote the region (or σ) containing r_σ, where we say that $p \in Z$ is to the right of $q \in Z$ if the point $p_H \in \partial H$ closest to p is counterclockwise around ∂H from the point $q_H \in \partial H$ closest to q. We let A_σ denote the shortest segment joining r_σ to Σ.

We define a partial order on stemmed squares within Z. Consider two distinct squares, σ and σ'. We say that σ *dominates* σ' if $\mathcal{R}_\sigma$ intersects $A_{\sigma'}$ and σ' lies to the left of the simple path, ν_σ, from ∂H to P_σ (along L_σ), then along the boundary of the region that contains r_σ, to r_σ, then to the side of the maximal square Σ associated with Z (the "top" of Z) along the shortest segment, A_σ, from r_σ to Σ. (We think of ν_σ as the "vertical line" through σ.) The partial order is well defined. It is easy to verify transitivity. Also, if σ dominates σ', then σ' does not dominate σ. (This follows from the fact that if $\mathcal{R}_\sigma$ intersects $A_{\sigma'}$, then $\mathcal{R}_{\sigma'}$ does not intersect A_σ: If $P \in \mathcal{R}_\sigma$ intersects $A_{\sigma'}$, then the rightmost point of P must be to the right of the point $r_{\sigma'}$ that defines $A_{\sigma'}$, and we know that $r_{\sigma'}$ is rightmost among regions $\mathcal{R}_{\sigma'}$.)

If σ is undominated among squares in Z, then no region $P \in \mathcal{R}_H$ crosses A_σ, since any such region P has a rightmost point to the right of r_σ, and there is some other grid square σ' for which $P \in \mathcal{R}_{\sigma'}$ (e.g., and square σ' that intersects P works).

The above observation allows us to write a dynamic program as follows. The set of stemmed squares within Z are ordered, $(\sigma_1, \ldots, \sigma_{K-1}, \sigma_K)$, according to where the stems touch ∂H (and according to angle among those stems that touch ∂H at the same point); additionally, we let σ_0 (resp., σ_{K+1}) denote the spoke that defines the "left" (resp., "right") side of Z.

A *subproblem* within zone Z is specified by giving two stemmed squares, (σ_i, σ_j), with the property that no regions intersect A_{σ_i} or A_{σ_j}. The subproblem asks us to find a minimum-weight set of stemmed squares in the subzone between ν_{σ_i} and ν_{σ_j} that stab all of the regions P that lie *fully* between ν_{σ_i} and ν_{σ_j}.

The important property about the subproblem is that we are able to specify *exactly which regions are the responsibility of the subproblem* – there are no regions that "sneak across" the boundary and would need to be specified, potentially causing the state space to explode. (This is the typical difficulty with recursive methods of trying to solve TSPN; see [18].) This is where we critically use the disjointness of the regions: No regions can cross the left boundary of the subproblem, ν_{σ_i}, along a portion of the curve ν_{σ_i} that lies inside a region. (And we know separately, from undominance, that no region crosses A_{σ_i}, and any region crossing the stem of σ_i is already visited by the stem, so we need not worry further about them in the subproblem.)

Let $f(i,j)$ be the optimal value associated with the subproblem. Then, we get the dynamic programming recursion: $f(i,j) = \min_{k \in K(i,j)}(weight(\sigma_k) + f(i,k) + f(k,j))$, where $K(i,j)$ is the set of all $k \in (i,j)$ such that σ_k is undominated among stemmed squares that lie between ν_{σ_i} and ν_{σ_j}.

Correctness of the dynamic program follows from the observation that there *exists* an undominated stemmed square in any optimal solution, and the principle of optimality. The output is a set of stemmed squares that have a minimum total weight among sets of stemmed squares that cover all regions interior to the zone.

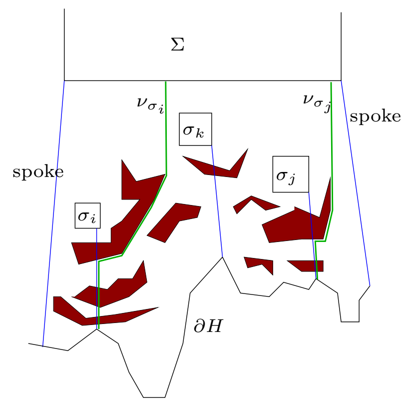

Figure 4: Illustration of the dynamic program. (The diagram is not an accurate depiction of the histogram, grid squares, etc.)

5 The Main Theorem

THEOREM 5.1. *There is a polynomial-time $O(1)$-approximation algorithm for the TSP with neighborhoods in the plane for a set $\mathcal{R}$ of disjoint, connected regions.*

Proof. The regions $\mathcal{R}$ are partitioned into blue and red regions; it suffices to compute an approximate solution for each of the two sets.

Our algorithm computes a tree $T_{\mathcal{E}_0}$ visiting a disjoint subset, $\mathcal{E}_0$, of the directional hulls of the blue (or red) regions. It does so with a known algorithm ([18]) that yields an approximately optimal tree/tour that visits $\mathcal{E}_0$. We know that an optimal tree/tour T^* must visit all of $\mathcal{R}$ and, therefore, must visit all of the regions $\mathcal{E}_0$ (each of which contains a region of $\mathcal{R}$). Thus, the tree $T_{\mathcal{E}_0}$ that we produce has length $O(L^*)$. Then, by Lemma 3.2, the network G that we compute is also of length $O(L^*)$.

The output of our algorithm is the network G, augmented by a set of networks F, one per face H of G. We know that T^* restricted to any one face H of G is a forest of trees, each of which is connected to ∂H, spanning the regions that lie interior to H. Since we compute a set of networks, one per face H (spanning the regions within H, and connecting to ∂H), each of which is within a constant factor of optimal for the subproblem H, we know that the total length of all forests we compute is $O(L^*)$.

Since our solution that is output is the union of the edge set of G and the set of computed networks F for each face H of G, we conclude that the network we compute, which spans $\mathcal{R}$, is of length $O(L^*)$. □

6 Conclusion

We have given the first constant-factor approximation algorithm for TSPN in the plane for a set of arbitrary disjoint connected regions. We have not yet attempted to optimize (or even to compute exactly) the constant factor in our approximation bounds for our algorithm; however, the factor is not particularly large, and we conjecture that with a careful analysis it can be reduced to $(2+\epsilon)$.

There are two main directions we are pursuing for future work on TSPN in the plane: (1) In the case of disjoint regions, it is not known if a PTAS may exist; (2) In the case of overlapping regions of arbitrary depth, it is not known if a constant-factor approximation may exist. Our results point to a possible approach: It suffices, using our results, to obtain a constant-factor approximation to visit a set of *fat* overlapping regions that have the property that each region lies within a constant factor of its diameter to the boundary of a containing histogram (Proposition 3.1). (We are able to show that the problem of visiting a set of overlapping fat convex regions is APX-hard, so we do not expect to obtain a PTAS.) Our dynamic programming algorithm exploits disjointness (though did not require fatness).

Additionally, we are pursuing approximation algorithms for TSPN in higher dimensions for general regions (not necessarily fat, as required in [4]).

References

[1] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Appl. Math.*, 55:197–218, 1994.

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

[3] S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. *Mathematical Programming*, 97(1-2):43–69, 2003.

[4] T.-H. H. Chan and K. Elbassioni. A QPTAS for TSP with fat weakly disjoint neighborhoods in doubling metrics. In *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, page to appear, 2010.

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

[6] M. Dror and J. B. Orlin. Combinatorial optimization with explicit delineation of the ground set by a collection of subsets. Technical report, MIT, 2004.

[7] A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms*, 48:135–159, 2003. Special issue devoted to 12th ACM-SIAM Symposium on Discrete Algorithms, Washington, DC, January, 2001.

[8] K. Elbassioni, A. V. Fishkin, N. H. Mustafa, and R. Sitters. Approximation algorithms for Euclidean group TSP. In *Proc. 32nd Internat. Colloq. Automata Lang. Prog.*, volume 3580 of *Lecture Notes Comput. Sci.*, pages 1115–1126. Springer-Verlag, 2005.

[9] K. Elbassioni, A. V. Fishkin, and R. Sitters. On approximating the TSP with intersecting neighborhoods. In *Proc. 17th Annu. Internat. Sympos. Algorithms Comput.*, volume ?? of *Lecture Notes Comput. Sci.*, page ?? Springer-Verlag, 2006.

[10] C. Feremans and A. Grigoriev. Approximation schemes for the generalized geometric problems with geographic clustering. In *Abstracts 21st European Workshop Comput. Geom.*, page xxx, 2005.

[11] J. Gudmundsson and C. Levcopoulos. A fast approximation algorithm for TSP with neighborhoods. *Nordic J. Comput.*, 6:469–488, 1999.

[12] M. Katz, J. S. B. Mitchell, and Y. Nir. Orthogonal segment stabbing. *Comput. Geom. Theory Appl.*, 30(2):197–205, 2005. Special issue devoted to selected papers of the 19th European Workshop on Computational Geometry (EuroCG2003), March 24 -26, 2003, Bonn, Germany.

[13] C. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages 360–369, 1995.

[14] J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: Part III – Faster polynomial-time approximation schemes for geometric network optimization. Manuscript, University at Stony Brook, 1997.

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

[16] J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[17] J. S. B. Mitchell. Shortest paths and networks. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry (2nd Edition)*, chapter 27, pages 607–641. Chapman & Hall/CRC, Boca Raton, FL, 2004.

[18] J. S. B. Mitchell. A PTAS for TSP with neighborhoods among fat regions in the plane. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'07)*, pages 11–18, 2007.

[19] S. B. Rao and W. D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *Proc. 30th Annu. ACM Sympos. Theory Comput.*, pages 540–550, 1998.

[20] S. Safra and O. Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. In *Proc. 11th Annu. European Sympos. Algorithms*, volume 2832 of *Lecture Notes Comput. Sci.*, pages 446–458. Springer-Verlag, 2003.

[21] S. Suri. A linear time algorithm for minimum link paths inside a simple polygon. *Comput. Vision Graph. Image Process.*, 35:99–110, 1986.