Approximation Algorithms for Geometric Tour and Network Design Problems

(Extended Abstract)

Cristian S. Mata [*] Joseph S. B. Mitchell [†]

1 Introduction

In this paper, we provide a simple method to obtain provably good approximation algorithms for a variety of NP-hard geometric optimization problems having to do with computing shortest constrained tours and networks:

Red-Blue Separation Problem (RBSP): Consider the problem of finding a minimum-perimeter Jordan curve (necessarily, a simple polygon) that separates a set of "red" points, R, from a set of "blue" points, B. This problem is seen to be NP-hard, using a reduction from the Euclidean traveling salesman problem [3, 12]. (Replace each city in the TSP instance by a pair of points, one red and one blue, very close together.) While Euclidean TSP can be approximated to within a factor of 1.5 times optimal (using Christofides' heuristic [9]), the seemingly related RBSP has defied previous attempts to devise a provably good approximation algorithm. We provide an $O(\log m)$ approximation bound algorithm for RBSP, where $m < n$ is the minimum number of sides of a minimum-perimeter rectilinear polygonal separator for the $n = |R| + |B|$ input points.

A similar approximation bound also applies to the case in which R and B are sets of nonoverlapping polygons, and to the "multi-color" case in which there are $K > 2$ sets of points or polygons, and we must find a planar subdivision of minimum total length that has objects of only one color in each face.

TSP with Neighborhoods: Let S be a collection of k possibly overlapping simple polygons ("neighborhoods") in the plane. The "Geometric Covering Salesman Problem", or *TSP with Neighborhoods* (TSPN) problem asks for a shortest tour that visits (intersects) each of the neighborhoods. The special case in which the neighborhoods are singleton points is just the usual Euclidean TSP problem; thus, TSPN is also NP-hard. One can think of the TSP with Neighborhoods problem as an instance of the "One-of-a-Set TSP" (also known as the "Multiple Choice TSP", the "Covering Salesman Problem", and the "Group TSP through sets of nodes"), where the "sets" are *connected* regions in the plane. This problem is widely studied for its importance in several applications, particularly in communication network design ([18]) and VLSI routing ([26]).

The only known approximation results for TSPN are those of [2], who, using judiciously placed "representative" points in each neighborhood, give constant-factor approximation algorithms for the special case in which the neighborhoods all have diameter segments that are parallel to a common direction, and the ratio of the longest to the shortest diameter is bounded by a constant.

We provide an $O(\log k)$ approximation algorithm for the case in which the neighborhoods are arbitrary polygonal regions.

---

*Department of Computer Science, SUNY Stony Brook, NY 11794-4400, **email: cristian@cs.sunysb.edu**; Supported by NSF grants ECSE-8857642 and CCR-9204585, and by a grant from Hughes Aircraft.

†Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY 11794-3600, **email: jsbm@ams.sunysb.edu**; Partially supported by NSF grants ECSE-8857642 and CCR-9204585, and by grants from Boeing Computer Services and Hughes Aircraft.

Watchman Route Problem: Let $\mathcal{P}$ be a polygonal room, possibly with "holes" (obstacles), having n vertices. The problem of computing a shortest tour in order that a mobile guard can "see" all of $\mathcal{P}$ is known as the *Watchman Route Problem* (WRP). If $\mathcal{P}$ is a simple polygon (having no holes), then WRP can be solved exactly, in time $O(n^4)$ [7, 23]. However, WRP is known to be NP-hard if $\mathcal{P}$ has holes (a simple reduction from Euclidean TSP; see [8]), even if $\mathcal{P}$ is rectilinear. But, as with RBSP, no approximation algorithms have previously been found for this problem. We give an $O(\log m)$ approximation algorithm for the WRP when the polygon $\mathcal{P}$ is rectilinear, where $m < n$ is the minimum number of edges in a shortest rectilinear watchman route.

k-MST Problem; Prize-Collecting Traveling Salesman Given a set S of n points in the plane, and an integer $k \leq n$, the k-MST problem asks for a spanning tree, possibly with Steiner points, of minimum length that joins some subset of k of the n points.

A related problem is the *Prize-Collecting Traveling Salesman Problem (PCTSP)*: Consider a salesperson that must sell a given quota, k, of widgets before returning home. Each city (point in the plane) has an integer demand associated with it, indicating how many widgets can be sold there. The goal in the "Prize Collecting Salesman" problem is to find a minimum-length tour (polygon) that visits a set of cities whose demands sum to at least k.

Until recently, no approximation algorithms were known for the k-MST and PCTSP problems. Awerbuch et al. [4] have given an $O(\log^2 k)$ approximation algorithm for both problems, on graphs with non-negative edge weights.

As a simple consequence of our method, we give an $O(\log k)$ approximation algorithm for the geometric instances of both problems, even when the sites are given as polygonal regions. Our bounds match in approximation factor those obtained very recently for point sites by Garg and Hochbaum [15]. In the time since the original submission of this paper, [6] have obtained a *constant* factor approximation bound for point sites, employing an algorithm very similar to our own, together with a new geometric lemma relating the lengths of minimum-spanning trees and "division trees" (trees induced by a binary space partition).

Overview of the Method

Our method is based on a standard approach to obtaining approximation algorithms — (1) prove that any solution to the original problem can be transformed into a solution to a simpler problem (at little change in objective function value); (2) solve the simpler problem to optimality; and (3) show how to transform a solution to the simpler problem into a solution to the original problem (again, with little change in objective function value). Specifically, our steps involve: (1) transforming an optimal tour problem into an optimal subdivision problem (which, by a geometric lemma, is shown to be close in length to the optimal tour length); (2) solving a canonical instance of the optimal subdivision problem by means of a dynamic programming (DP) algorithm; and (3) showing that the DP solution can be transformed into a solution of the original problem, with a small factor increase in total length. This method is not new; e.g., it has been used by Gonzalez et al. [16] in optimal partitioning problems, and by Agarwal and Suri [1] and Mitchell [22] in other geometric separation and approximation problems.

2 Geometric Preliminaries

We restrict our attention to problems in two dimensions. We will work with planar polygonal subdivisions and triangulations, always within some bounded polygonal region, $\mathcal{R}$. The *length* of a planar subdivision, triangulation, or polygon is defined to be the sum of the Euclidean lengths of its edges. We say that a subdivision $\mathcal{S}'$ is a *refinement* of subdivision $\mathcal{S}$ if each edge of $\mathcal{S}$ is a subset of some edge of $\mathcal{S}'$. The *bounding box*, $bb(X)$, for a set X is defined to be the smallest axis-aligned rectangle that contains the set. The *grid* induced by a finite set of points is the graph consisting of all the vertices and bounded edges of the arrangement of all vertical and horizontal lines through the points.

A polygonal subdivision $\mathcal{S}$ of polygon $\mathcal{R}$ is said to be "guillotine" if it is a binary space partitioning (BSP) of $\mathcal{R}$; i.e., either there exist no edges of $\mathcal{S}$ interior to $\mathcal{R}$, or there exists a (straight) edge of $\mathcal{S}$ such that this edge is a chord of $\mathcal{R}$, dividing $\mathcal{R}$ into two regions, $\mathcal{R}_1$ and $\mathcal{R}_2$, such that $\mathcal{S}$ restricted to $\mathcal{R}_i$ is a guillotine subdivision of $\mathcal{R}_i$. (See [11] and standard references on BSP trees [24, 25].) If all faces of the subdivision are rectangles, we obtain a guillotine *rectangular* subdivision (GRS). Here, we will concern ourselves only with guillotine rectangular subdivisions.

The following geometric lemmas will be needed in our proofs and are of independent interest. The first lemma is due to Clarkson [10]; we provide a simple proof based on a recent result of [17]. The second lemma is due to Levcopoulos and Lingas [21]; again, for completeness, we provide a simple proof.

Lemma 1 *(Clarkson [10]) Given a simple polygon P, having n vertices, oone can compute, in $O(n)$ time, a triangulation of P, using $O(n)$ Steiner points (non-vertices of P), whose length is $O(\log n)$ times the length of P.*

Proof. In [17], it is shown that, for any simple polygon P, one can compute in $O(n)$ time a triangulation $\mathcal{T}$ such that (1) the vertices of $\mathcal{T}$ consist of the n vertices of P, plus a set of $O(n)$ additional Steiner points inside P; and (2) any line segment within P intersects $O(\log n)$ triangles of $\mathcal{T}$.

We claim that the low "stabbing number" triangulation $\mathcal{T}$ has the desired property for our lemma. To see this, take any edge (diagonal), e, of $\mathcal{T}$. If e is an edge of P, then we "charge" e's length to the perimeter of P. Otherwise, we charge e's length off to the perimeter of P, as follows.

If the slope of the line through e is greater than 1, then we project e to the right, onto the boundary of P. We say that a point $p \in \partial P$ on the boundary of P is in the *rightward shadow* of e if there exists a point $q \in e$, with q left of p, such that the segment qp is horizontal and lies within P.

The length of the rightward shadow of e is *at least* $|e|/\sqrt{2}$; thus, we charge the length of e to the rightward shadow of e. Since $\mathcal{T}$ has $O(\log n)$ stabbing number, any horizontal segment within P can cross at most $O(\log n)$ diagonals of $\mathcal{T}$. Thus, any point on the boundary of P can be in the rightward shadow of at most $O(\log n)$ diagonals, and can therefore be charged at most $O(\log n)$ times. It follows that the sum of the lengths of all diagonals e having slope greater than 1 is $O(\log n)$ times the length of P.

A similar argument applies to those diagonals with slope less than 1, whose length we charge to their *upward shadows*. $\square$

Remark. Note that it is necessary to allow Steiner points in the above lemma. In fact, a "boomerang" polygon that consists of a triangle, minus a convex pocket, has a unique triangulation, whose length is $\Omega(n)$ times its perimeter. Also, note that the result is essentially tight (see Eppstein [13]).

Lemma 2 *(Levcopoulos-Lingas [21]) Given a simple rectilinear polygon P, having n vertices, one can compute in time $O(n)$ a decomposition of P into $O(n)$ axis-aligned rectangles, whose length is $O(\log n)$ times the length of P.*

Proof. (Sketch) In [5], it is shown that, using the decomposition into histograms given by [19, 20], for any simple rectilinear polygon P, one can compute in $O(n)$ time a decomposition into $O(n)$ rectangles, $\mathcal{R}$, such that any horizontal or vertical line segment within P intersects $O(\log n)$ rectangles of $\mathcal{R}$. We claim that the low "stabbing number" decomposition $\mathcal{R}$ has the desired property for our lemma.

The proof of this claim is similar to the proof of Lemma 1: Simply "charge" (project) each internal edge of the decomposition off to the boundary. No boundary point is charged more than $O(\log n)$ times. $\square$

Remark. In fact, the bound in the above lemma is tight in the worst case. A staircase polygon can be used to show that, in general, a minimum-length rectangular subdivision of P may have length $\Omega(\log n)$ times the length of P.

Lemma 3 *Let $\mathcal{S}$ be a rectangular subdivision of a (rectilinear) polygon $\mathcal{R}$, with $\mathcal{S}$ having a total of n vertices. Then, there exists a guillotine rectangular subdivision, $\mathcal{S}_G$, that is a refinement of $\mathcal{S}$ such that the length of $\mathcal{S}_G$ is at most $O(1)$ times the length of $\mathcal{S}$.*

Proof. (Sketch) We adapt the proof of [11], who show that an optimal GRS is at most twice the length of an optimal rectangular subdivision of a given set of points. (Here, in the optimization problem, the goal is to include each point of the given set on (at least) one of the edges of the subdivision.) A careful examination of their proof reveals that it is not necessary to make any assumptions about the "optimality" of the subdivisions: One can show that an arbitrary rectangular subdivision can be transformed into a GRS, which is a refinement of the original and whose total length is at most twice that of the original. $\square$

3 Red-Blue Separation

We begin with the problem of finding a minimum-perimeter Jordan curve (necessarily, a simple polygon) that separates a set of "red" points, R, from a set of "blue" points, B; we call such a separator a *separating simple polygon*. We let $S = R \cup B$, and $n = |S| = |R| + |B|$. We say that a subdivision is *color-conforming* if each of its open 2-faces contains points of only one color (from R or from B, but not from both).

If, instead of minimizing the Euclidean length, we want to minimize the *combinatorial* size of a separating simple polygon (i.e., the number of vertices or "links"), then an $O(\log k^*)$ approximation bound has recently been established ([1, 22]), where k^* is the size of an optimal separator; the problem is known to be NP-hard in this case [14]. Here, we give an approximation result for the problem of minimizing the Euclidean length of the separator, a problem shown to be NP-hard in [3, 12].

Consider a minimum-length separating simple polygon, P^*, with length ℓ^*. Note that P^* either contains all red points inside (or on the boundary), or all the blue points. Thus, we will solve two problems and pick the shorter length: find a minimum-length polygon that encloses red, while excluding blue; and find a minimum-length polygon that encloses blue, while excluding red.

Without loss of generality, assume that P^* surrounds the blue points and excludes the red points. Let $\mathcal{B}$ be the bounding box for B; note that P^* lies within $\mathcal{B}$ and that $\mathcal{B}$ is the bounding box of P^*.

Lemma 4 *There exists a rectilinear red-blue separating polygon, contained in $\mathcal{B}$, having $O(n)$ vertices, lying on the grid induced by S, and with length at most $\sqrt{2} \cdot \ell^*$. In particular, a minimum-length rectilinear separating polygon must have length at most $\sqrt{2} \cdot \ell^*$.*

Proof. We transform each edge uv of P^* into a "staircase" path π lying on the grid induced by S. The endpoints of the edge, u and v, must lie on the grid, since the vertices of P^* must be vertices in S. We define the rectilinear path π as the *right rectilinear envelope* of uv, a path from u to v in the grid such that it is to the right of uv and such that no vertex of the grid is inside the polygon defined by uv and π. See Figure 1.

It is not hard to show the following claim: The polygon P^*_π obtained from P^* by replacing each edge uv by the corresponding right envelope π is a (possibly degenerate) simple polygon. (proof omitted here) □

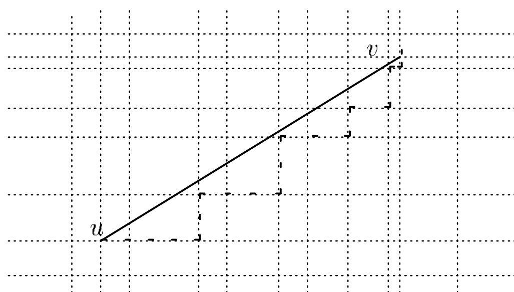

Figure 1: Right rectilinear envelope for edge uv

Among all minimum-length rectilinear separating polygons, let P^*_R be one with a minimum number, m, of vertices; clearly, $m \leq n$.

Consider the planar subdivision consisting of P^*_R and its bounding box, $\mathcal{B}$. Each face of this subdivision is a simple rectilinear polygon, with at most m vertices, containing either red points or blue points on its interior, but not both. Now, by Lemma 2, we know that this subdivision can be refined into a color-conforming rectangular subdivision, $\mathcal{S}$, of $\mathcal{B}$, whose length is $O(\log m)$ times the length of P^*_R (and thus of P^*). Then, by Lemma 3, we know that there exists a color-conforming GRS, $\mathcal{S}_G$, of $\mathcal{B}$, a refinement of $\mathcal{S}$, whose length is $O(1)$ times the length of $\mathcal{S}$, and hence is $O(\ell^* \log m)$. Below, we give a dynamic programming algorithm to compute a minimum-length color-conforming GRS of a given bounding box. Applying this to $\mathcal{B}$, we obtain a subdivision, $\mathcal{S}^*_G$, whose length is at most that of $\mathcal{S}_G$, and hence at most $O(\ell^* \log m)$.

Lemma 5 *For any color-conforming GRS of $\mathcal{B}$, there exists a separating simple polygon whose boundary is a subset of the edges of the subdivision and whose length is $O(1)$ times the length of the subdivision.*

Proof. (Sketch) Consider the binary tree associated with the BSP that defines the color-conforming GRS. Each non-leaf node in this tree has an associated guillotine cut (line segment), ξ. We consider ξ to be "attached", at one of its endpoints, to one edge of the associated bounding box. (Such an edge is a guillotine cut associated with an ancestor of the node, assuming it is not the root). The other endpoint of ξ is considered to be "touching", but not attached to, the opposite side of the bounding box; we consider there to be a tiny gap at this endpoint. In this way, the guillotine cuts of the GRS form a (degenerate, rectilinear) tree within $\mathcal{B}$. We begin with a clockwise tour that is a doubling of this tree. If an (oriented) segment uv along this tour is associated with a leaf cut $\xi = uv$ that has a (leaf) rectangular facet on its left, and this facet contains only blue points, then we replace uv with the 3-link path that detours (clockwise) around the boundary of the facet. The resulting tour encloses all blue points, excludes all red points, and has length at most twice that of the GRS. □

A Dynamic Programming Algorithm We now give a dynamic programming algorithm to compute $\mathcal{S}^*_G$. Let $x_1 < x_2 < \cdots < x_n$ (resp., $y_1 < y_2 < \cdots < y_n$) denote the sorted x (resp., y) coordinates of the $n = |S| = |R| + |B|$ given points.

Let $V(i, j, k, l)$ denote the minimum length of a color-conforming GRS of the rectangle, $R(i, j, k, l)$, defined by x_i, x_j ($x_j > x_i$), y_k, and y_l ($y_l > y_k$). Here, in $V(i, j, k, l)$, we do not count the perimeter of $R(i, j, k, l)$ in the length of the subdivision; we only count the lengths of interior edges. Thus, we get

$V(i, j, k, l) = 0$, if all points of S interior to $R(i, j, k, l)$ are of the same color; otherwise, $V(i, j, k, l) =$

$$\begin{aligned} \min\{(y_l - y_k) &+ \min_{i<i'<j} [V(i, i', k, l) + V(i', j, k, l)], \\ (x_j - x_i) &+ \min_{k<k'<l} [V(i, j, k, k') + V(i, j, k', l)]\}. \end{aligned}$$

In other words, the length of the segments making up a minimum-length partition, excluding the bounding box of the set of points, is the length of the horizontal or vertical partitioning cut plus the sum of the lengths of optimal solutions to the two subproblems defined by the cut, optimized over all possible cuts.

The running time of this algorithm is easily seen to be $O(n^5)$, since there are $O(n^4)$ subproblems (rectangles), and we optimize over $O(n)$ choices of cut (and easily check whether a rectangle has all points of the same color).

Theorem 6 *Given a set of n points in the plane, each either "red" or "blue", a red-blue separating simple polygon whose length is $O(\log m)$ times the minimum possible length can be computed in polynomial ($O(n^5)$) time, where $m < n$ is the minimum number of vertices in a minimum-length rectilinear separating polygon.*

The high running time (and space complexity) of the algorithm can be brought down to $O(n^2)$, at a cost of increasing the approximation bound to $O(\log^3 n)$. This follows from a simple observation of [22]: After transforming an optimal tour to a rectilinear polygon and then obtaining a rectangular subdivision of the bounding box, we further refine the subdivision, by decomposing each rectangle into $O(\log^2 n)$ *canonical* rectangles. A *canonical rectangle* is the direct product of two *canonical intervals*; a canonical x-interval is one whose x-coordinates are x_{i2^j} and $x_{(i+1)2^j}$, for some integers i, j ($0 \leq i < \frac{n}{2^j}$, $j \geq 0$). (The standard segment tree data structure decomposes an interval into its $O(\log n)$ canonical subintervals.) There are only $O(n)$ canonical x-intervals, and only $O(n^2)$ canonical rectangles. We can rewrite the dynamic programming algorithm to optimize only over GRS's into canonical rectangles. For a canonical rectangle, we need consider only *two* possible cuts — a bisecting horizontal or vertical cut. Further, it is easy to tabulate in advance, in time $O(n^2)$, for all $O(n^2)$ canonical rectangles, which of them contain points of a single color. Thus, the dynamic programming algorithm will require only $O(n^2)$ time and space.

Theorem 7 *Given a set of n points in the plane, each either "red" or "blue", a red-blue separating simple polygon whose length is $O(\log^3 n)$ times the minimum possible length can be computed in $O(n^2)$ time.*

Now we observe that the approach given so far for red-blue point separation can be applied more generally to the problem of building a minimum-length subdivision that separates into color classes a given set of points, each having an assigned color. Specifically, we want to compute a simple polygon P, and a decomposition of P into simple polygonal faces, such that each face in the subdivision (including the face at infinity) has points of at most one color in it, and the total length of the decomposition is as small as possible. (There may be more than one face having "red" points in it, but in any one face there can be points of only one color.) Exactly the same asymptotic approximation bounds and time bounds apply to this problem, as given in the previous two theorems.

Finally, we observe that the same method can be applied to the generalization in which, instead of a set of colored points, we have a set of colored rectilinear polygons, and we desire a minimum-length subdivision that separates them into color classes.

4 TSP with Neighborhoods

Let $S = \{R_1, \ldots, R_k\}$ be a collection of possibly overlapping simple polygons (*neighborhoods*), having a total of n vertices. We say that a tour or a subdivision *visits* S if its edges intersect every neighborhood R_i. The *TSP with Neighborhoods* (TSPN) problem asks for a minimum-length tour that visits S. We consider lengths to be measured in the Euclidean metric, but the results apply to other standard metrics as well. Here, we provide an $O(\log k)$ approximation algorithm that runs in polynomial time.

Consider a minimum-length tour, P^*, with length ℓ^*, and let $\mathcal{B}^*$ denote its bounding box. Our first lemma is a simple consequence of the local optimality of P^*:

Lemma 8 *P^* is a* simple *(not self-intersecting) polygon, having at most k vertices.*

Proof. Let p be the leftmost point of P^*; if there is more than one leftmost point, let p be the one of maximum y-coordinate. It is not hard to see from the local optimality of P^* that p must lie in some region, R_j. Consider a traversal of the tour, starting from p. Let p_i be the first point of R_i visited along this traversal. (Then, $p = p_j$.) It is known (and easy to show) that a Euclidean TSP tour on the k points, $p_1, \ldots, p_k$, must be a simple polygon, with vertices among the points p_i. But P^* must itself be a TSP tour on the k points p_i — if it were longer than such a TSP tour, then it could be shortened by replacing it with the TSP tour (which visits all regions R_i). Thus, P^* is a simple polygon with at most k vertices. □

Now, our goal is to convert P^* into a GRS whose edges intersect all regions R_i and whose length is comparable to P^*. We will then perturb this subdivision so that its edges lie on the grid induced by the vertices of S. A dynamic programming algorithm will yield a shortest such GRS, which will be converted into an approximately optimal TSP.

Let uv be one edge of P^*; without loss of generality, assume that uv has negative slope. We would like to replace each edge uv with a rectilinear path. If we replace uv with the 2-link ("L-shaped") path, π_L, that goes downwards out of u, then rightwards to v, then this new path may fail to stab all the neighborhoods visited by the straight segment uv. Instead, we will replace uv with a rectilinear path, π, defined as follows.

Starting at some vertex, p, of P^*, traverse the tour P^*, and let $S_{uv} \subseteq S$ be the set of regions that edge uv intersects, but have not been intersected by an edge of P^* previously during the traversal. Let T be the (right) triangle bounded by uv and π_L. Consider a region $R_i \in S_{uv}$. If R_i has a vertex in T, pick one of them, and call it r_i;

otherwise, R_i must cross π_L. We now define the rectilinear path π to be that portion between u and v of the boundary of the *upper rectilinear hull* of the points $\{u, v\}$, together with the points r_i determined by the regions $R_i \in S_{uv}$. Thus, π is a "staircase" in T, separating segment uv from the points r_i (if any) and from π_L. See Figure 2. (Note that, if there are no points r_i associated with uv, then π will equal π_L.) Since the regions R_i are connected, and they have some point on uv and some point below π, we know that π must intersect each of the regions $R_i \in S_{uv}$.

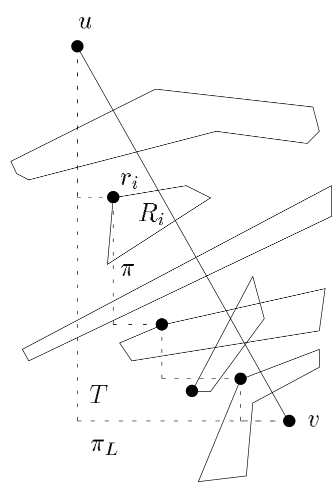

Figure 2: Construction of path π.

So, for each edge uv of P^*, we replace uv with its staircase path π. The result is a rectilinear closed walk, W, having $O(k)$ vertices, whose length is at most $\sqrt{2} \cdot \ell^*$. Note, however, that W is not necessarily simple and not necessarily lying on the grid induced by S (since u and v need not be on the grid). Let P_R^* denote a shortest *rectilinear* tour that visits all neighborhoods R_i; i.e., P_R^* is a shortest tour among all those rectilinear tours that visit all neighborhoods. Clearly, P_R^* is not longer than W. Further, it is easy to argue that P_R^* is a *simple* rectilinear polygon (since uncrossing a tour only improves its length), having $O(k)$ vertices (by a proof similar to that of Lemma 8. We select P_R^* to have a minimum number, m $(= O(k))$, of edges among all shortest rectilinear tours visiting all neighborhoods.

Using Lemmas 2 and 3, we can decompose each bounded face of the arrangement of P_R^* and $\mathcal{B} = bb(P_R^*)$ into rectangles, and then refine the resulting rectangular subdivision into a GRS, $\mathcal{S}$, of $\mathcal{B}$ whose length is $O(\ell^* \log m)$. The edges of the subdivision $\mathcal{S}$ intersect all regions $R_i \in S$, but they do not necessarily lie on the grid induced by S. The following lemma shows, however, that $\mathcal{S}$ can be transformed into a subdivision that lies on the grid, without lengthening it.

Lemma 9 *$\mathcal{S}$ can be transformed to lie on the grid induced by S, while still having its edges intersect all regions $R_i \in S$, and without increasing its length.*

Proof. (Sketch) First, we argue that we can transform $\mathcal{S}$ so that its bounding box lies on the grid. The details are tedious and omitted here, but the main idea is to slide the top edge of the bounding box downwards, hoping to align it with a grid line. Before this happens, though, the corner of the bounding box may "pull away" from a region, and we cannot allow this. So, if a corner c becomes incident on an edge e of some region, we then start sliding c along e, adjusting the bounding box and edges of $\mathcal{S}$ accordingly. Again, a corner may "pull away", and we may end up with all four corners constrained to slide along given edges. The crucial property is, though, that the total length of the modified subdivision is a linear function of the sliding parameter, so we can always do the sliding without lengthening $\mathcal{S}$.

Next, we must transform $\mathcal{S}$ so that all internal edges also lie on the grid. We do this perturbation one (guillotine) cut at a time. Consider a guillotine cut, ξ; assume it is vertical. Consider what happens as we slide it left/right, allowing the (horizontal) edges incident to it to lengthen/shorten. Again, one direction of sliding will only improve matters, by shortening the length of $\mathcal{S}$ (i.e., slide in the direction that has more horizontal edges incident on it). (Again, we may have to reassess which direction we are sliding when a rectangle collapses to zero area, but this presents no real problem.) Further, all regions R_i will continue to be intersected, until a vertex event, when ξ hits a vertex of S. We then recurse for the guillotine cuts on either side of ξ, eventually obtaining a GRS with all edges on the grid. □

The final result of the above discussion is that we can transform an optimal tour, P^*, into a GRS, $\mathcal{S}_G$, that lies on the grid induced by S, with total length $O(\ell^* \log m)$.

Now, by a straightforward variation of the dynamic programming algorithm given earlier (for the RBSP), we can, for each of the $O(n^4)$ possible bounding boxes $\mathcal{B}$ defined by vertices of S, compute a minimum-length GRS of $\mathcal{B}$ whose edges intersect all regions of S (if one exists — this will depend on the choice of $\mathcal{B}$). By optimizing over the choice of $\mathcal{B}$, we obtain a minimum-length GRS, $\mathcal{S}_G^*$, whose length is at most that of $\mathcal{S}_G$, and hence at most $O(\ell^* \log m)$. Finally, we need:

Lemma 10 *For any GRS of $\mathcal{B}$ whose edges intersect all regions of S, there exists a simple polygonal tour P (1) whose boundary is a subset of the edges of the subdivision, (2) that visits S, and (3) whose length is $O(1)$ times the length of the subdivision.*

Theorem 11 *Let $S = \{R_1, \ldots, R_k\}$ be a collection of possibly overlapping simple polygons, having a total of n*

vertices. One can compute, in polynomial time, a tour P that visits S, whose length is $O(\log m)$ times the minimum length of a tour that visits S, where $m = O(k)$ is the minimum number of links in a shortest rectilinear tour visiting S.

5 Watchman Routes

Let $\mathcal{P}$ be a (closed, connected) polygonal domain, possibly with "holes" (obstacles), having n vertices (S) and (consequently) n edges. We say that a tour (or a subdivision) is *watchman* if it "sees" all of $\mathcal{P}$; i.e., every point of $\mathcal{P}$ is seen by some point on the tour (or by some point on an edge of the subdivision). (Point $p \in \mathcal{P}$ *sees* point $q \in \mathcal{P}$ if $\mathcal{P}$ contains the segment pq.) Here, we assume that $\mathcal{P}$ is rectilinear and that no two edges are collinear. (A slight perturbation can guarantee this.) We give an $O(\log n)$ approximation algorithm for computing an optimal (shortest) watchman route. (Note that, in general, there is not a unique optimal route.)

Let P^* be an optimal watchman route, let its (Euclidean) length be ℓ^*, and let $\mathcal{B}^*$ be its bounding box. Our first lemma is a simple consequence of the local optimality of P^*: (proof omitted here)

Lemma 12 *Let $\mathcal{P}$ be an arbitrary polygonal domain (with holes) having n vertices, r of which are reflex. Then, an optimal watchman route for $\mathcal{P}$ is a* simple *(not self-intersecting) polygon, having at most r vertices.*

Now, our goal is, as in the TSP with neighborhoods problem, to convert P^* into a GRS, which is watchman and of comparable length to P^*, and then to perturb this subdivision so that its edges lie on the grid induced by the vertices S.

We replace each edge uv of P^* with the boundary of the maximal rectilinearly convex polygon, P_{uv}, that contains uv but is contained within uv's bounding box (i.e., $uv \subset P_{uv} \subset bb(uv)$). Such a polygon P_{uv} is unique, and is bounded by two staircase paths from u to v. Since P_{uv} surrounds uv and contains no obstacles, any point of $\mathcal{P}$ that sees uv also must see the boundary of P_{uv}. Further, the length of P_{uv} is at most $2\sqrt{2}|uv|$.

Now consider the arrangement, $\mathcal{A}$, formed by the union of all polygons P_{uv}, and the bounding box, $\mathcal{B}^*$. The total length of the edges in this arrangement is at most $2\sqrt{2}\ell^*$. Also, there exists a (rectilinear) tour, W, whose length is $O(1)$ times the length of $\mathcal{A}$, such that W includes all edges of $\mathcal{A}$, so that W is also watchman (but not simple). (Simply go around the boundary of each P_{uv} one-and-one-half times — go clockwise from u to v to u to v.) Among all minimum-length rectilinear watchman tours, let P_R^* be one with a minimum number, m, of vertices. It is not hard to see that $m \leq n$. Clearly, P_R^* is shorter than W. Further, it is easy to argue that P_R^* is a simple rectilinear polygon, and that, because of the nondegeneracy assumption on $\mathcal{P}$, P_R^* will have no two edges collinear.

Using Lemmas 2 and 3, we can decompose each bounded face in the arrangement of P_R^* and $bb(P_R^*)$ into rectangles, and then refine the resulting rectangular subdivision of $bb(P_R^*)$ into a GRS, $\mathcal{S}$, whose length is $O(\ell^* \log m)$. The subdivision $\mathcal{S}$ is watchman, since adding edges can only increase what is seen. Further, there exists a *connected* subset of *feasible* edges of $\mathcal{S}$ that suffice to see all of $\mathcal{P}$; for example, the edges of P_R^* form such a set. (A line segment is *feasible* if it lies within $\mathcal{P}$ — i.e., it does not cross any obstacles.) But $\mathcal{S}$ does not necessarily lie on the grid induced by S (since the segment endpoints, u, v, may not be grid vertices). We will show, however, that $\mathcal{S}$ can be transformed into a watchman GRS, $\mathcal{S}_G$, that lies on the grid, without lengthening it, and while maintaining the property that a connected subset of feasible edges illuminates all of $\mathcal{P}$.

A horizontal edge of $\mathcal{P}$ (outer boundary or obstacle edge) that has the interior of $\mathcal{P}$ locally above (resp., below) it will be called a *bottom* (resp., *top*) edge. Let y_{max} denote the maximum y-coordinate of any bottom edge of $\mathcal{P}$. Similarly, let y_{min} denote the minimum y-coordinate of any *top* edge of $\mathcal{P}$. Define x_{max} and x_{min} analogously.

Lemma 13 *The bounding box, $bb(P_R^*)$, of a shortest rectilinear watchman route is determined by the four coordinates y_{max}, y_{min}, x_{max}, and x_{min}.*

Proof. Consider a bottom edge e of $\mathcal{P}$. A watchman route must see all points on e; in order to see a point in the middle of e, the route must have some point at the same or greater y-coordinate as e. Thus, the top side of $bb(P_R^*)$ must be at or above any such e. Continuing this argument for the bottom, left, and right sides of $bb(P_R^*)$, we get that $bb(P_R^*)$ must contain the rectangle, R_0, determined by y_{max}, y_{min}, x_{max}, and x_{min}. (The rationale is the same as for "essential cuts", as introduced by [8].)

Now we must argue that there is no reason for P_R^* to go outside of the box R_0. Assume to the contrary that the top edge, e', of P_R^* is a horizontal edge with y-coordinate greater than y_{max}. Consider what happens as we slide e' downwards. Since e' lies above all bottom edges of $\mathcal{P}$, e' will not initially be in contact with an obstacle, so the sliding process is initially feasible. If moving e' downwards a small amount ϵ causes P_R^* to stop seeing some point, $p \in \mathcal{P}$, then it must be that p lies on a bottom edge of $\mathcal{P}$ that is collinear with e' and has y-coordinate greater than y_{max}, contradicting the definition of y_{max}. Thus, we must be able to slide e' downwards some amount $\epsilon > 0$, while maintaining feasibility of P_R^* and while maintaining the fact that P_R^* sees all of $\mathcal{P}$. But, as we slide e', P_R^* will decrease in length, contradicting its optimality. $\square$

Corollary 14 *The bounding box, $bb(P_R^*)$, of a shortest rectilinear watchman route contains all holes (obstacles).*

Lemma 15 *$\mathcal{S}$ can be transformed into a watchman GRS, $\mathcal{S}_G$, that lies on the grid induced by S, has length at most that of $\mathcal{S}$, and has a connected subset of feasible edges illuminating $\mathcal{P}$.*

Proof. From Lemma 13, we know that $bb(P_R^*)$ must itself lie on the grid induced by S. Now, consider a guillotine cut, ξ, of $bb(P_R^*)$. Consider the process of sliding ξ parallel to itself, and adjusting the incident edges of $\mathcal{S}$, making them shorter or longer, as appropriate. We can choose to slide ξ in a direction, say leftwards, that yields no increase in overall length of $\mathcal{S}$. We can continue sliding ξ until (1) it hits an obstacle edge (in which case ξ now lies on the grid); or until (2) some edge of $\mathcal{S}$ incident on ξ shrinks to length zero (in which case we collapse and remove the zero-area rectangle and reassess which direction we should slide ξ). In the end, this process must terminate with ξ becoming collinear with an edge of $\mathcal{P}$, and we then recurse on each side of the cut. Also, during the process, the property that a connected subset of feasible edges sees all of $\mathcal{P}$ is preserved. □

Below, we give a dynamic programming algorithm to compute a minimum-length GRS of a given bounding box, with the property that all of $\mathcal{P}$ that is *inside* of the given box is illuminated by a *connected* subset of the *feasible* edges of the subdivision that are connected to a specified set of four feasible edges ("illuminators") on the boundary of the box, each of which has its endpoints at vertices of the grid induced by S. We can use this algorithm to get our approximation result, as follows.

We start by computing $bb(P_R^*)$, which, by Lemma 13, is given by simply computing y_{max}, y_{min}, x_{max}, and x_{min}. Next, we apply the dynamic programming algorithm for each of the $O(n^8)$ choices of four (feasible) illuminator segments on the boundary of $bb(P_R^*)$ — call them σ_ℓ, σ_r, σ_b, and σ_t, on the left, right, bottom, and top, respectively — and we optimize over these choices. (The four edges of P_R^* that lie on the boundary of $bb(P_R^*)$ are a witness to the fact that four illuminators suffice on the boundary of $bb(P_R^*)$; by our nondegeneracy assumption, there is *exactly* one edge of P_R^* per edge of $bb(P_R^*)$.) The result is that we obtain an optimal GRS, $\mathcal{S}_G^*$, whose length is at most that of $\mathcal{S}_G$, and hence at most $O(\ell^* \log m)$.

We now must argue that we can convert $\mathcal{S}_G^*$ into a (feasible) watchman tour that has about the same total length. We know, by the constraint imposed in the dynamic programming algorithm, that $\Sigma = \Sigma_\ell \cup \Sigma_r \cup \Sigma_b \cup \Sigma_t$ illuminates all of $\mathcal{P} \cap bb(P_R^*)$, where Σ_α is the connected component containing σ_α in the set of feasible edges of $\mathcal{S}_G^*$ ($\alpha \in \{\ell, r, b, t\}$).

Let H denote a shortest rectilinear tour that surrounds all obstacles and goes through σ_ℓ, σ_r, σ_b, and σ_t. (H is a rectilinear "relative convex hull", and can easily be computed.) Necessarily, H will lie within $bb(P_R^*)$, since we know that $bb(P_R^*)$ contains all holes of $\mathcal{P}$ (Corollary 14). We omit here the simple proofs of the following:

Lemma 16 *H sees all of $\mathcal{P} \setminus bb(P_R^*)$.*

Lemma 17 *The length of H is at most ℓ^*.*

Now, H must intersect Σ, since both sets touch all four sides of $bb(P_R^*)$. Thus, $F = H \cup \Sigma$ is connected, sees all of $\mathcal{P}$, and has length $O(\ell^* \log m)$. We can easily use F to construct a feasible tour that sees all of $\mathcal{P}$. One way to do this is to convert F into a tree (by making small cuts as necessary), and then to define a tour that traverses the boundary of the tree, at most doubling its length.

Theorem 18 *Given a rectilinear polygonal room with n vertices, a watchman tour whose length is $O(\log m)$ times the minimum possible length can be computed in polynomial (in n) time, where $m \leq n$ is the minimum number of edges in a rectilinear optimal watchman route.*

A Dynamic Programming Algorithm Let $x_1 < x_2 < \cdots < x_n$ (resp., $y_1 < y_2 < \cdots < y_n$) denote the sorted x (resp., y) coordinates of the n vertices, S, of $\mathcal{P}$. We now give a dynamic programming algorithm to solve the following problem:

Input: Rectangle $R(i,j,k,l)$, defined by x_i, x_j ($x_j > x_i$), y_k, and y_l ($y_l > y_k$); segments σ_ℓ ("left"), and σ_r ("right"), σ_b ("bottom"), σ_t ("top"), which are subsegments of the top (rep., bottom, left, right) edge of $R(i,j,k,l)$, each of which has endpoints on the grid induced by S.

Objective: Compute a minimum-length GRS, $\mathcal{S}_G^*$, of $R(i,j,k,l)$ such that $\Sigma_\ell \cup \Sigma_r \cup \Sigma_b \cup \Sigma_t$ illuminates all of $\mathcal{P} \cap R(i,j,k,l)$, where Σ_α is the connected component containing σ_α in the set of feasible edges of $\mathcal{S}^*$ ($\alpha \in \{\ell, r, b, t\}$).

We think of the segments σ_α ($\alpha \in \{\ell, r, b, t\}$) as the "illuminators"; any feasible segment that can be connected to an illuminator by a path of feasible segments also serves as an illuminator. Our goal is to illuminate all of $\mathcal{P}$ that lies inside of $R(i,j,k,l)$, by making certain that any point of $\mathcal{P} \cap R(i,j,k,l)$ can see at least one illuminator.

Let $V(i,j,k,l,\sigma_\ell,\sigma_r,\sigma_b,\sigma_t)$ be the length of an optimal GRS. Here, in $V(i,j,k,l,\sigma_\ell,\sigma_r,\sigma_b,\sigma_t)$, we do not count the perimeter of $R(i,j,k,l)$ in the length of the subdivision; we only count the lengths of interior edges. Thus, we can write the following recursion:

$V(i,j,k,l,\sigma_\ell,\sigma_r,\sigma_b,\sigma_t) = 0$, if $\mathcal{P} \cap R(i,j,k,l)$ is illuminated by $\{\sigma_\ell,\sigma_r,\sigma_b,\sigma_t\}$; otherwise,

$$
\begin{aligned}
&V(i,j,k,l,\sigma_\ell,\sigma_r,\sigma_b,\sigma_t) = \\
&\min\{ \quad (y_l - y_k) + \min_{i<i'<j}[\quad V(i,i',k,l,\sigma_\ell,\sigma_t^\perp,\sigma_t',\sigma_b') + \\
&\qquad\qquad V(i',j,k,l,\sigma_t^\perp,\sigma_r,\sigma_t'',\sigma_b'')], \\
&\quad (y_l - y_k) + \min_{i<i'<j}[\quad V(i,i',k,l,\sigma_\ell,\sigma_b^\perp,\sigma_t',\sigma_b') + \\
&\qquad\qquad V(i',j,k,l,\sigma_b^\perp,\sigma_r,\sigma_t'',\sigma_b'')], \\
&\quad (x_j - x_i) + \min_{k<k'<l}[\quad V(i,j,k,k',\sigma_\ell',\sigma_r',\sigma_b,\sigma_\ell^\perp) + \\
&\qquad\qquad V(i,j,k',l,\sigma_\ell'',\sigma_r'',\sigma_\ell^\perp,\sigma_t)], \\
&\quad (x_j - x_i) + \min_{k<k'<l}[\quad V(i,j,k,k',\sigma_\ell',\sigma_r',\sigma_b,\sigma_r^\perp) + \\
&\qquad\qquad V(i,j,k',l,\sigma_\ell'',\sigma_r'',\sigma_r^\perp,\sigma_t)]\}.
\end{aligned}
$$

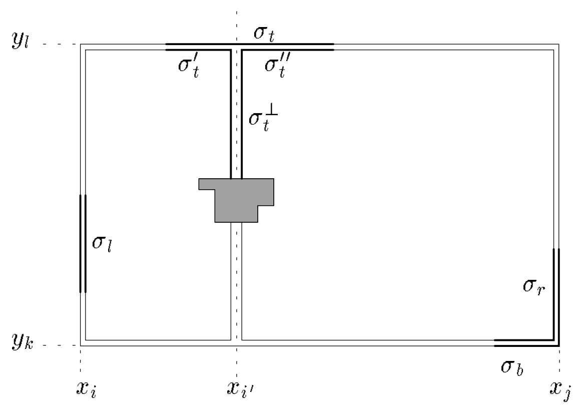

Figure 3: Notation in the recursion.

Here, $\sigma_t' \subset \sigma_t$ (resp., $\sigma_t'' \subset \sigma_t$) is that portion of σ_t that lies to the left (resp., right) of the vertical cut (line) through $x_{i'}$. Also, $\sigma_t^\perp$ is the maximal feasible (grid) segment on the vertical cut through $x_{i'}$, with its top endpoint on σ_t. See Figure 3. (We select the longest feasible segment to serve as illuminator, since having a longer illuminator can only help us, and it does not cost anything additional, since the objective function minimizes the total length of the subdivision.) Similar definitions apply to each of the other terms of the forms σ_α', σ_α'', and $\sigma_\alpha^\perp$, for $\alpha \in \{\ell, r, b, t\}$.

The running time of this algorithm is easily seen to be polynomial ($O(n^{13})$), since there are $O(n^{12})$ subproblems (rectangles, plus choice of illuminators σ_α), and we optimize over $O(n)$ choices of cut. (Determining if all of $\mathcal{P} \cap R(i,j,k,l)$ is illuminated by a given set of illuminator segments is easily tabulated, for all $O(n^{12})$ choices, within the $O(n^{13})$ time bound.)

In fact, we obtain a slightly more reasonable time bound of $O(n^9)$ by noting that the illuminators $\sigma_\alpha^\perp$ that are constructed along cuts during the algorithm will always have one endpoint at the corner of the bounding box of the subproblem and will extend to a maximal length (i.e., until an obstacle or the opposite side of the bounding box is reached). This implies that a subproblem is fully determined by its bounding box, plus a constant number of bits of extra information to tell us on which sides and attached to which corners we have illuminator segments. The exception is that portions of the original bounding box and its four illuminators may also appear on the boundary of a subproblem. In our application, the initial call to the recursion will involve a fixed rectangle, $bb(P_R^*)$, and $O(n^8)$ possible choices for illuminators σ_α. This leads to "only" $O(n^8)$ subproblems to evaluate, since a subproblem is fully determined by its bounding box, some constant number of bits, plus the appropriate portion of the (original) illuminator segments that appear on sides of the bounding box that are common to the initial box ($bb(P_R^*)$). The net result is that $O(n^9)$ time suffices for our problem.

6 k-MST

Given a set S of n points in the plane, and an integer $k \leq n$, the k-MST problem asks for a spanning tree, possibly with Steiner points, of minimum (Euclidean) length that connects some subset of k of the n points. Here, we give an $O(\log k)$ approximation algorithm for this problem, which implies a similar result for the *Prize-Collecting Traveling Salesman Problem (PCTSP).*

Consider a minimum-length k-MST, T^*, with length ℓ^*. It is easy to check that T^* must be *simple* (non-self-crossing). Furthermore, T^* can easily be perturbed into a rectilinear tree, T_R, on the grid induced by S, without affecting its (Euclidean) length by much (factor $\leq \sqrt{2}$).

Consider the planar subdivision consisting of T_R and its bounding box, $\mathcal{B}$. Each face of this subdivision is a simple rectilinear polygon, with at most k vertices. Now, by Lemma 2, we know that this subdivision can be refined into rectangular subdivision, $\mathcal{S}$, of $\mathcal{B}$, whose length is $O(\log k)$ times the length of T_R (and thus of T^*). Then, by Lemma 3, we know that there exists a GRS, $\mathcal{S}_G$, of $\mathcal{B}$, a refinement of $\mathcal{S}$, whose length is $O(1)$ times the length of $\mathcal{S}$, and hence is $O(\ell^* \log k)$.

As in the problems already discussed, we can use dynamic programming, optimized over choice of bounding box, to obtain a minimum-length GRS (of some bounding box), $\mathcal{S}_G^*$, that visits k points of S. (In particular, we define our "value" function $V(i,j,k,l;m)$ to be the minimum length of a GRS that visits m of the points within $R(i,j,k,l)$, and then optimize over all cuts and over all partitions of the integer m.) The length of $\mathcal{S}_G^*$ is at most that of $\mathcal{S}_G$, and hence at most $O(\ell^* \log k)$. Further, any spanning tree for the k points of S on the edges of $\mathcal{S}_G^*$ is a feasible solution to the original problem, and trivially has length at most that of $\mathcal{S}_G^*$.

In fact, by a straightforward application of the ideas used to approximate the TSP with Neighborhoods problem, we can allow the set S to consist of a set of n *polygons*, k of which must be visited by a tree, and obtain the following result for the k-MST *with neighborhoods* problem:

Theorem 19 *Given a set S of n polygons in the plane, having a total of N vertices, and an integer $k \leq n$, one can compute, in polynomial (in N) time, a tree that visits k of the n polygons whose length is within factor $O(\log k)$ of optimal.*

Corollary 20 *The Euclidean* Prize-Collecting Traveling Salesman Problem *with polygonal neighborhoods can be approximated, in polynomial time, with an approximation factor of $O(\log k)$.*

7 Conclusion

There are many other problems to which our method may be applied, and we are currently pursuing some of these. The watchman route problem with arbitrary (nonrectilinear) holes can probably be solved to within a polylogarithmic factor of optimal using our methods. Further, it seems that we can obtain constant factor approximation algorithms for many of the problems we have addressed, e.g., based on the new results of [6]. The full paper will report the results of these pursuits. We are also currently investigating higher dimensional versions of these problems.

Acknowledgements

We thank Estie Arkin and Rafi Hassin for helpful discussions and introducing us to many of the problems discussed here. We thank Andrzej Lingas for discussions and pointing out some useful references.

References

[1] P. K. Agarwal and S. Suri. Surface approximation and geometric partitions. In *Proc. Fifth Annual ACM-SIAM Sympos. on Discrete Algorithms*, 1994, pages 34–43.

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

[3] E. M. Arkin, S. Khuller, and J. S. B. Mitchell. Geometric knapsack problems. *Algorithmica*, 10:399–427, 1993.

[4] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. To appear, STOC 1995.

[5] M. de Berg and M. van Kreveld. Rectilinear decompositions with low stabbing number. Tech. Report RUU-CS-93-25, Dept. Comput. Sci., Utrecht Univ., 1993.

[6] A. Blum, P. Chalasani, and S. Vempala. A constant-factor approximation for the k-MST problem in the plane. To appear, STOC 1995.

[7] S. Carlsson, H. Jonsson, and B. J. Nilsson. Finding the shortest watchman route in a simple polygon. In *Proc. 4th Annu. Internat. Sympos. Algorithms Comput. (ISAAC 93)*, LNCS Vol. 762, pp. 58–67, Springer-Verlag, 1993.

[8] W. Chin and S. Ntafos. Optimum watchman routes. *Inform. Process. Lett.*, 28:39–44, 1988.

[9] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In J. F. Traub, editor, *Sympos. on New Directions and Recent Results in Algorithms and Complexity*, New York, NY, 1976. Academic Press.

[10] K. L. Clarkson. Approximation algorithms for planar traveling salesman tours and minimum-length triangulations. In *Proc. Second Annual ACM-SIAM Sympos. on Discrete Algorithms*, 1991, pages 17–23.

[11] D.-Z. Du, L.-Q. Pan, and M.-T. Shing. Minimum edge length guillotine rectangular partition. Report 02418-86, Math. Sci. Res. Inst., Univ. California, Berkeley, CA, 1986.

[12] P. Eades and D. Rappaport. The complexity of computing minimum separating polygons. *Pattern Recognition Letters*, 14:715–718, 1993.

[13] D. Eppstein. Approximating the minimum weight triangulation. In *Proc. Third Annual ACM-SIAM Sympos. Discrete Algorithms*, 1992, pages 48–57.

[14] S. Fekete. On the complexity of min-link red-blue separation. Manuscript, 1992.

[15] N. Garg and D. S. Hochbaum. An $O(\log k)$ approximation algorithm for the k minimum spanning tree problem in the plane. In *Proc. 26th Annu. ACM Sympos. Theory Comput. (STOC 94)*, pages 432–438, 1994.

[16] T. Gonzalez, M. Razzazi, and S.-Q. Zheng. An efficient divide-and-conquer approximation algorithm for hyperrectangular partitions. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 214–217, 1990.

[17] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. In *Proc. 4th ACM-SIAM Sympos. Discrete Algorithms*, pages 54–63, 1993.

[18] E. Ihler, G. Reich, and P. Widmayer. On shortest networks for classes of points in the plane. In *Computational Geometry — Methods, Algorithms and Applications: Proc. Internat. Workshop Comput. Geom. CG '91*, LNCS Vol. 553, pp. 103–111. Springer-Verlag, 1991.

[19] C. Levcopoulos. Fast heuristics for minimum length rectangular partitions of polygons. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 100–108, 1986.

[20] C. Levcopoulos. Heuristics for minimum decompositions of polygons. PhD diss. No. 155, Linköping University 1987.

[21] C. Levcopoulos and A. Lingas. Bounds on the length of convex partitions of polygons. In *Proc. 4th Conf. Found. Softw. Tech. Theoret. Comput. Sci.*, LNCS Vol. 181, pp. 279–295. Springer-Verlag, 1984.

[22] J.S.B. Mitchell. Approximation algorithms for geometric separation problems. Technical report, AMS Dept., SUNY Stony Brook, NY, July 1993.

[23] B.J. Nilsson. Guarding art galleries — Methods for mobile guards. Ph.D. thesis, Lund University, 1995.

[24] M. S. Paterson and F. F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete Comput. Geom.*, 5:485–503, 1990.

[25] M. S. Paterson and F. F. Yao. Optimal binary space partitions for orthogonal objects. *J. Algorithms*, 13:99–113, 1992.

[26] G. Reich and P. Widmayer. Beyond Steiner's problem: A VLSI oriented generalization. In *Proc. 15th Internat. Workshop Graph-Theoret. Concepts Comput. Sci.*, LNCS Vol. 411, pp. 196–210. Springer-Verlag, 1989.