On the minimum corridor connection problem and other generalized geometric problems

Hans L. Bodlaender

Corinne Feremans

Alexander Grigoriev

Eelko Penninkx

René Sitters

Thomas Wolle

Department of Information and Computing Sciences,
Utrecht University

Technical Report UU-CS-2007-031

www.cs.uu.nl

ISSN: 0924-3275

On the minimum corridor connection problem and other generalized geometric problems *†

Hans Bodlaender‡ Corinne Feremans§ Alexander Grigoriev¶

Eelko Penninkx‖ René Sitters** Thomas Wolle††

Abstract

In this paper we discuss the complexity and approximability of the minimum corridor connection problem where, given a rectilinear decomposition of a rectilinear polygon into "rooms", one has to find the minimum length tree along the edges of the decomposition such that every room is incident to a vertex of the tree. We show that the problem is strongly NP-hard and give a subexponential time exact algorithm. For the special case when the room connectivity graph is k-outerplanar the algorithm running time becomes cubic. We develop a polynomial time approximation scheme for the case when all rooms are fat and have nearly the same size. When rooms are fat but are of varying size we give a polynomial time constant factor approximation algorithm.

Keywords: minimum corridor connection; generalized geometric problems; complexity; exact algorithms; approximations

*This work was partially supported by the Netherlands Organization for Scientific Research NWO (project *Treewidth and Combinatorial Optimization*). Thomas Wolle was also supported by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

†This paper is an expanded and combined version of a talk [14] presented at the European Workshop on Computational Geometry 2005, Eindhoven, The Netherlands, and a talk [6] presented at the Workshop on Approximation and Online Algorithms 2006, Zurich, Switzerland.

‡Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands, hansb@cs.uu.nl

§Department of Quantitative Economics, Maastricht University. c.feremans@ke.unimaas.nl

¶Corresponding author. Department of Quantitative Economics, Maastricht University, P.O.Box 616, 6200 MD Maastricht, The Netherlands. grigoriev@ke.unimaas.nl

‖Institute of Information and Computing Sciences, Utrecht University. penninkx@cs.uu.nl

**Eindhoven University of Technology, Department of Mathematics and Computer Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. r.sitters@tue.nl

††National ICT Australia Ltd, Locked Bag 9013, Alexandria NSW 1435, Australia. thomas.wolle@nicta.com.au

1 Introduction

MCC and other generalized geometric problems. We consider the following geometric problem. Given a rectilinear decomposition of a rectilinear polygon (a subdivision into n "*rooms*"), find the minimum length tree ("*corridor*") along the edges of the decomposition ("*walls*") such that every room is incident to a vertex of the tree (has access to the corridor); for an illustration see Fig. 1 borrowed from [8]. Let us refer to this problem

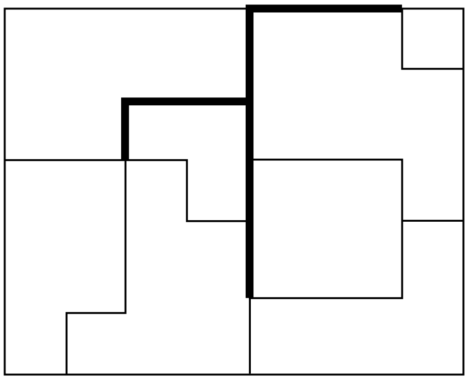

Figure 1: Minimum length tree along the corridors to connect the rooms

as to the *minimum corridor connection* (MCC) problem.

This problem belongs to the class of so-called *generalized geometric problems* where given a collection of objects in the plane, one has to find a minimum length network satisfying certain properties that hits each object at least once. In particular, MCC can be viewed as a special case of the generalized geometric Steiner tree problem where given a set of disjoint groups of points in the plane, the problem is to find a shortest (in some metric space) interconnection tree which includes at least one point from each group.

The most studied generalized geometric problem is the following generalization of the classic Euclidean Traveling Salesman Problem (ETSP). Assume that a salesman has to visit n customers. Each customer has a set of specified locations in the plane (referred to as a *region* or *neighborhood*) where the customer is willing to meet the salesman. The objective is to find a shortest (closed) salesman tour visiting each customer. If each region is a single point, the problem becomes the classic ETSP. This generalization of ETSP is known as the Geometric Covering Salesman Problem [1], or ETSP with neighborhoods [7, 11], or the Generalized ETSP [16]. For short, we shall refer to this problem as to GTSP. In a similar way one can define generalizations for Minimum Steiner Tree (GSTP), Minimum Spanning Tree (GMST) and many other geometric problems.

Applications. Applications for the minimum corridor connection problem and other generalized geometric problems are naturally encountered in telecommunications, in particular in VLSI design. For instance, a metropolitan area is divided by streets and avenues into rectilinear blocks and the blocks must be interconnected by an optical fiber network containing a gateway from each block. For easy maintenance the optical cables must be placed in the collector system which goes strictly under the streets and avenues. The problem is to find the minimum length network connecting all blocks. In Section 4.5 we discuss how our techniques can be applied to even more generalized variants of this problem. For the related problems and for the extended list of applications see Feremans [13], Feremans, Labbé and Laporte [15], Mitchell [28], Reich and Widmayer [30].

Three-dimensional applications of the generalized geometric problems, particularly MCC, appear also in constructions where, e.g., wiring has to be installed along the walls, floors and ceilings of the multistore building such that each room has electricity, phone lines, etc.

Related Work. To the best of our knowledge, the minimum corridor connection problem was first posed by N. Katoh at the 12th Canadian Conference on Computational Geometry CCCG 2000; see [8]. Till recently nothing was known on complexity and approximability of the problem. Then, Bodlaender et al. [6] and Gonzalez-Gutierrez and Gonzalez [20] simultaneously and independently reported that the problem is strongly NP-hard.

For GTSP it is known that the problem cannot be efficiently approximated within $(2-\varepsilon)$ unless $P = NP$, see [31]. Constant factor approximations for GTSP were developed for the special cases where neighborhoods are disjoint convex fat objects [7, 12], for disjoint unit discs [1], and for intersecting unit discs [11]. For the general GTSP, Mata and Mitchell [26] gave an $O(\log n)$-approximation algorithm. Dumitrescu and Mitchell [11] have investigated the case of GTSP with regions given by pairwise disjoint unit disks, and developed a polynomial time approximation scheme (PTAS) for this problem. Recently, Mitchell [27] drastically improved this result showing that GTSP restricted to just fat objects already admits a PTAS.

For the general GSTP, Helvig, Robins and Zelikovsky [22] developed a polynomial time n^{ε}-approximation algorithm where $\varepsilon > 0$ is any fixed constant. For GSTP, GMST and several other generalized geometric problems exact search methods and heuristics have been developed, see e.g. Zachariasen and Rohe [33] and Feremans, Labbé and Laporte [15].

Our results and paper organization. The present paper is a revised and expanded version of talks [14, 6] presented at EWCG 2005 and WAOA 2006. Herewith, we give a broad algorithmic picture on the minimum corridor connection problem. In particular, in Section 2 we show that the problem is strongly NP-hard, answering an open question from CCCG 2000 on the complexity of the minimum corridor connection; see [8]. It is noticeable that our reduction is much shorter and easier than the reduction by Gonzalez-Gutierrez and Gonzalez [20]. Moreover, it relies on completely different class of instances of the minimum corridor connection problem than the one in [20].

In Section 3 we present a subexponential time exact algorithm for MCC and a cubic time algorithm for the special case when the room connectivity graph is k-outerplanar.

Then, in Section 4 we construct a PTAS for MCC with fat rooms having nearly the same size, that partially solves another open question from CCCG 2000 on the approximability of MCC, see [8]. More precisely, we consider the problem where a square of side length q can be inscribed in each room and perimeter of each room is bounded from above by

cq where c is a constant. In fact, we present a framework for construction the PTASs for a variety of generalized geometric problems restricted to (almost) disjoint fat object of nearly the same size. We refer to this restriction as to *geographic clustering* since one can associate disjoint fat objects with countries on a map where all countries have comparable (up to a constant factor) border lengths.

The presented framework for PTASs is based on Arora's algorithm for ETSP [3]. In particular, this framework allows to construct PTASs for GTSP, GSTP, and GMST restricted to geographic clustering. The main advantage of our techniques compared to the recent approximation scheme Mitchell [27] for GTSP on fat objects is that it leads to a more efficient approximation scheme running in time $n(\log n)^{O(1/\varepsilon)}$ compared to $n^{O(1/\varepsilon)}$ in [27]. Moreover, our techniques are applicable to many other norms (e.g., the one which is used in MCC) and to any fixed dimensional spaces, which resolves one of the open questions in [11].

Finally, in Section 5 we show how the algorithm for GTSP from Elbassioni et al [12] can be used to derive a polynomial time constant approximation algorithm for MCC with fat rooms of varying sizes, that complements our partial answer on the open question from CCCG 2000 on the approximability of MCC, see [8].

2 Complexity of MCC

In this section, we show that the decision version of MCC is strongly NP-complete. To show this result, we use a transformation from the CONNECTED VERTEX COVER problem for planar graphs with maximum degree four. In this later problem, given a planar graph $G = (V, E)$ such that each vertex in V has degree at most 4, and a positive integer $R \leq |V|$, the question is whether exists a connected vertex cover of size at most R for G, i.e., does there exist a subset $W \subseteq V$ with $|W| \leq R$ such that the subgraph induced by W is connected and $u \in W$ or $v \in W$ for each edge $\{u, v\} \in E$? CONNECTED VERTEX COVER for planar graphs with maximum degree four is NP-complete, see [19, 18]. Now we state the main result of this section.

Theorem 1 *The minimum corridor connection problem is NP-complete, even when coordinates of corner points are given in unary.*

Proof. Clearly, the problem belongs to NP. As we announced before, to prove NP-completeness of MCC, we reduce CONNECTED VERTEX COVER for planar graphs with maximum degree four to the minimum corridor connection problem. Let the instance of CONNECTED VERTEX COVER be given by a planar graph $G = (V, E)$ on n vertices with maximum vertex degree 4, and let $R \leq n$ be a positive integer. We transform G into an instance of the minimum corridor connection problem in a number of steps.

1. *Make a rectilinear embedding of G.*

   Use the algorithm of Biedl and Kant [4] to find a rectilinear embedding of G in an $n \times n$ grid, and such that each edge has at most two bends. This rectilinear embedding assigns to each vertex of V a point in the plane with integer coordinates in $\{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$, and each edge is represented by a sequence of horizontal and vertices lines; bend points are also on integer coordinates in $\{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$.

2. *Enlarge the drawing.*

   Let L be the maximum over all edges $e \in E$ of the total length of all the segments that represent e. Let $K = \max(7n+1, 16L+1)$. We stretch the drawing horizontally and vertically by a factor of $2K$, i.e., all coordinates of vertices and turning points are multiplied by $2K$.

3. *Create edge-rooms.*

   After Steps 1 and 2, each edge is represented by a sequence of horizontal and vertical lines. In this step, we transform this to a very narrow room.

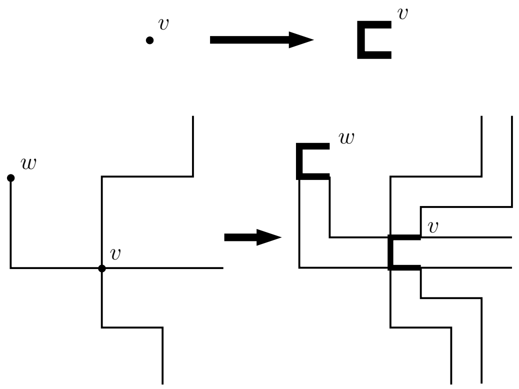

Figure 2: Transforming edges to rooms

   As shown in Fig. 2, each vertex is replaced by four vertices at the corners of a one by one square, with between three of these vertices an edge. Alongside the original path representing an edge, we take a parallel path of distance one. In this way, this path, together with the original path, and some of the edges at the vertices, forms a face in the new drawing.

   We now distinguish two types of faces: original faces (faces that are present in the original embedding of the graph) and edge faces (faces of 'width one' that represent an edge in G.)

4. *Make long edges of equal length.*

   Each edge face consists of two paths between two vertices, plus two segments of length one. Call the former *long edges*. The total length of a long edge is at most $2KL + 4$: originally, edges were represented by a path of length L, in Step 2, these were made $2K$ times as long, and Step 3 can increase the length by at most 4 (2 at each of the at most two bends). It is also at least $2K$.

In this step, we make sure that each of the long edges has a length that is either $2KL + 3$ or $2KL + 4$. This can be done as illustrated in Fig. 3.

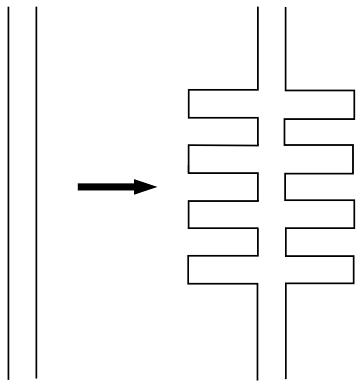

Figure 3: Making edges longer

It is not hard to see that we made K large enough such that this step can be done while there are no overlaps between the extra segments, and such that all long edges have the required length $2KL + 3$ or $2KL + 4$. Also, note that the step does not change the faces, except for the lengths of their boundary edges.

Consider the resulting diagram. It is a rectilinear decomposition of a rectilinear polygon. We claim there is a tree along the edges of the subdivided polygon, such that each room is incident with an edge of the tree of length at most $2KL(R-1) + 7R$, if and only if there is a connected vertex cover of G with at most R vertices.

Suppose there is a connected vertex cover W of size at most R of G. Take a tree T in G that spans exactly the vertices in W. As W is connected, T exists. T contains at most $R-1$ edges. We now build the tree T' that connects the faces. For each edge in T, take one of the two long sides of the room representing this edge, and add that to T'. To this, we add for each vertex in V, the three length one segments we added in Step 2. The total length hence is at most $(R-1)(2KL+4)+3|W| \leq 2KL(R-1)+7R$. This tree covers all the rooms. An edge face is covered because one of its endpoints belongs to W. An original face is covered: look at one of its incident edges; at least one endpoint of the edge belongs to W and hence contains a vertex on T' that is incident to the face.

Suppose we have a tree T' along the edges of the subdivided polygon such that each room is incident with an edge of the tree, and has length at most $2KL(R-1) + 7R$. Let F be the set of edges of G such that T' contains at least one of the long paths of the face room of that edge. F must form a connected subgraph of G. F can contain at most $(2KL(R-1) + 7R)/(2KL + 3) \leq R - 1$ edges, as each edge in R contributes to at least $2K + 3$ length in T'. (We use that $R \leq n$ here.) Suppose that $F \neq \emptyset$. Let W be the set of vertices that are an endpoint of an edge in F. As F is a connected set of at most $R-1$ edges, $|W| \leq R$. We claim that W is a connected vertex cover. Clearly, $G[W]$ is connected (as it has F as spanning tree). Consider an edge $\{v, w\} \in E$. The room of this edge must be seen by T'. It follows that either some of the vertices representing v, or some of the vertices representing w must belong to T'. As T' must connect these to the remainder ($F \neq \emptyset$), an edge with v or w as endpoint must belong to F and hence $v \in W$ or $w \in W$. The case that $F = \emptyset$ is degenerate: simple analysis shows that this can happen only if G is a star, i.e., of the form $K_{1,n-1}$.

Note that $L = O(n)$ and hence $K = O(n^2)$. The steps can be carried out in polynomial time, and thus we have the requested transformation. □

3 Exact algorithms with branchwidth

In this section, we discuss how the problem can be solved exactly exploiting the notion of branchwidth and k-outerplanarity.

A *branch decomposition* of a graph $G = (V, E)$ is a pair (T, σ), with T an unrooted ternary tree and σ a bijection between the leaves of T and the edge set E. For each edge e in T, consider the two subtrees T_1 and T_2 obtained by removing e from T. Let $G_{e,1}$ ($G_{e,2}$) be the subgraph of G, formed by the edges associated with leaves in T_1 (T_2). The *middle set* of an edge e in T is the set of vertices in both G_{e_1} and $G_{e,2}$. The *width* of a branch decomposition is the maximum size over all middle sets, and the *branchwidth* of a graph is the minimum width over all branch decompositions.

A *noose* is a closed simple curve on the plane that intersects a planar graph G only at vertices. To a noose, we can associate two regions of the plane (the "inside" and the "outside"), and likewise two subgraphs: the part of G drawn inside the noose, and the part of G drawn outside the noose. These subgraphs intersect precisely in the vertices on the noose.

A branch decomposition (T, σ) is a *sphere cut decomposition* or *sc-decomposition*, if for every edge e in T, there is a noose such that the two subgraphs associated with it are exactly $G_{e,1}$ and $G_{e,2}$, and the noose touches each face of G at most once. Necessarily, the set of the vertices on the noose is the middle set of e.

A sphere cut decomposition of minimum width can be found in $O(n^3)$ time with the ratcatcher algorithm of Seymour and Thomas [32], see [10]. See also [21, 23, 24] for a necessary improvement to the original algorithm and implementation issues. In particular, Gu and Tamaki [21] show how to obtain a constructive version of the ratcatcher algorithm that uses cubic time.

3.1 Dynamic programming with a branch decomposition

Instead of the MCC problem, we consider a small generalization, which we call FACE COVER TREE.

> FACE COVER TREE:
> **Given:** a plane graph $G = (V, E)$, with edge weights $w : E \to \mathbf{N}$
> **Question:** find a subtree T of G of minimum total weight such that each face has at least one vertex on T.

A solution for this problem does not depend on the embedding of the graph since any tree that hits all faces in fact hits all cycles in the graph and vice versa. Hence, the FACE COVER TREE problem is the problem of finding a feedback vertex set that minimizes the connection cost.

We now give an algorithm that solves the FACE COVER TREE problem using a sphere cut decomposition of G.

Theorem 2 *Suppose a plane graph is given together with a sphere cut decomposition of width at most k. Then the* FACE COVER TREE *problem can be solved in $O((3+\sqrt{5})^{2k}k \cdot n)$ time.*

Proof. To obtain this result, we use techniques from Dorn et al. [10]. The basic idea is that we build a table for each edge in the sphere cut decomposition.

Let some arbitrary node r be a root of T. Now, to each edge $e \in \sigma$, let E_e be the set of edges, associated with leaves that are below e in the tree T, and let G_σ be the subgraph, induced by E_e. Note that G_e is either $G_{e,1}$ or $G_{e,2}$, as in the definition of sphere cut decompositions.

Consider a forest T' that is a subgraph of G_e. Suppose we want to extend T' by adding edges from $E - E_e$ to it, such that we obtain a solution of the FACE COVER PROBLEM (possibly with non-optimal total edge weight), i.e., to a tree such that each face of G has at least one vertex on the tree. Such an extension exists, if and only if T' is a tree and touches each face of G, or each subtree of T' contains at least one vertex in the middle set of e, and each face of G_e that does not intersect the noose is touched.

Characterize forests T' in G_e by the set of vertices in the middle set of e that belong to T', the equivalence relation on these vertices on which of these vertices are connected by T', and the information on which faces that intersect the noose of e are touched by T'.

Note that if T' and T'' have the same characterization, and if we add edges $E' \subseteq E - E_e$ to T' to obtain a solution of the FACE COVER PROBLEM, then we also obtain a solution if we add E' to T''.

This observation gives the basis of the dynamic programming algorithm for the FACE COVER PROBLEM. For each edge e in the tree of the sphere cut decomposition, we compute a table of all characterizations of subforests T' of G_e such that each subtree of T' contains at least one vertex in the middle set of e, and each face of G_e that does not intersect the noose is touched. For each such characterization in this table, we tabulate the minimum weight of such a subforest T'.

In other words, in our dynamic programming algorithm, we tabulate for each edge e in the branch decomposition tree, for each triples (S, R, X), where S is a subset of the middle set of e, R is an equivalence relation on S, and X is a subset of the faces intersecting the noose of e, if there is at least one forest T' in G_e such that S is the set of vertices in the middle set that belong to T', R is the relation on S that there is a path in T', and X is the set of faces intersecting the noose of e that are touched by e, the minimum total weight of such a forest.

Lemma 3 *If $e \in \sigma$ has a middle set of size ℓ, then the table of e contains at most $(3+\sqrt{5})^\ell$ entries.*

Proof. To show this, we can use a counting technique from Dorn et al. [10].

Consider the vertices on the noose e, and look at these, starting at some vertex, in clockwise order. Note that each face that intersects the noose contains two successive vertices on the noose.

For a table entry (S, R, X), note that R is a *non-crossing* partition on S (see [10] for details).

For a given table entry (S, R, X), we map each vertex on the noose to the language $\{1_[, 1_], 1_\Box, 1_m, 0_t, 0_n\}$ as follows: vertices in S are mapped to $1_[$, $1_]$, or 1_m: if $v \in S$ is the first vertex in an equivalence class in R, then v is mapped to $1_[$, if it is the last vertex in an equivalence class in R, then map it to $1_]$, if it is both the first and last vertex in an equivalence class, then it is mapped to $1_\Box$ if it is neither the first nor last vertex in an equivalence class, map it to 1_m. First and last are with respect to the order in which we consider the vertices.

By the fact that R is non-crossing (or, because we obtained R by connectedness by a forest in the place), we have that R can be constructed by the mapping of the vertices in S. Vertices on the noose, but not on R are mapped to 0_t and 0_n. The vertex $v \notin R$ is mapped to 0_t if the face that intersects the noose directly clockwise from v belongs to X (i.e., is touched by the forest T' represented by the table entry.) Otherwise, v is mapped to 0_n.

Thus, for each table entry, we have a unique mapping to $\{1_[, 1_], 1_\Box, 1_m, 0_t, 0_n\}^*$. Thus, it directly follows that a table for a noose with ℓ vertices on it has at most 6^ℓ entries.

A technique from [10] can be used to improve slightly on the constant 6. Note that a vertex that is coded 0_n must be followed by a vertex that is coded 0_n or 0_t: if it is followed by a vertex in S, then the face following the vertex is touched by the forest. Thus, the number of table entries is bounded by the number of strings in $\{1_[, 1_], 1_\Box, 1_m, 0_t, 0_n\}^*$ of length ℓ with the property that a 0_n symbol is followed by a 0_n or 0_t symbol. Write $C_1 = \{0_n\}$, $C_2 = \{0_t\}$, and $C_3 = \{\{1_[, 1_], 1_\Box, 1_m\}$. Let a_{ij} be the number of possible symbols in C_j, after a symbol in C_i. Now, we have

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 4 \\ 1 & 1 & 4 \end{pmatrix}$$

As in [10], we can bound the number of strings by $O(c^\ell)$, with c the largest real eigenvalue of A; in this case, this largest eigenvalue equals $3 + \sqrt{5}$. $\Box$

To compute the table for an edge in T incident to a leaf is trivial. For other edges e, we combine the two tables for the two edges incident to the lower endpoint of e. Basically, we try to combine each table entry of the left table with each table entry of the right table; in $O(k)$ time, we can verify whether these give a new table entry, and of what signature. Thus, the table for an edge can be computed in $O((3 + \sqrt{5})^{2k} \cdot k)$ time.

From the table of the edge to the root, we can then determine the answer to the problem. We computed $O(n)$ tables, and hence used $O((3+\sqrt{5})^{2k} \cdot k \cdot n)$ time. This concludes the proof of Theorem 2. $\square$

Note that $(3+\sqrt{5})^2 = 14 + 6\sqrt{5} = 2^{4.7770}$, i.e., we have an algorithm for FACE COVER TREE that uses $O(2^{4.7770k}kn)$ time.

With a more detailed analysis, it probably is possible to bring the running time down. In particular, when combining two tables, we do not need to look at all possible combinations of strings in $\{1_{[}, 1_{]}, 1_{\square}, 1_m, 0_t, 0_n\}^*$. For instance, we can skip cases we have two successive vertices v, w that belong to both nooses of the left and right table, and v and w are coded 1_m in both tables, as such a case would represent a cycle in T (there is a path from v to w in T in the 'left noose' and in the 'right noose').

3.2 Algorithmic consequences

Given a plane graph $G = (V, E)$, we can divide the vertices of G into layers. All vertices incident to the exterior face are in layer L_1. For $i \geq 1$, all vertices incident to the exterior face after we removed all vertices in layers $L_1, \ldots, L_i$ are in layer L_{i+1}. A planar graph G is k-outerplanar, if it has a plane embedding with at most k non-empty layers. It is well known that a k-outerplanar graph has branchwidth at most $2k$; this can be proved in the same way as the proof in [5] that k-outerplanar graphs have treewidth at most $3k-1$.

It is interesting to note that in some applications, graphs with small outerplanarity will arise in a natural way. For instance, for many buildings, the wall structure of one floor will have bounded outerplanarity, as usually, each room is adjacent to a corridor, and each corridor is adjacent to a room with a window, and thus, unless there is an open air part not at the exterior, this gives small outerplanarity.

It is also long known that planar graphs have branchwidth (and treewidth) $O(\sqrt{n})$. (This statement can be seen to be equivalent to the Lipton-Tarjan planar separator theorem [5, 25].) The best known bound to our knowledge is the following.

Theorem 4 (Fomin and Thilikos [17]) *A planar graph with n vertices has branchwidth at most $\sqrt{4.5 \cdot n}$.*

Thus we have the following consequences.

Corollary 5 *The* FACE COVER TREE*, and hence also the MCC problem can be solved in $O(n^3 + 2^{9.5539k}n)$ time on k-outerplanar graphs, and in $O(2^{10.1335\sqrt{n}})$ time on planar graphs.*

We expect that the actual running times of these algorithms will be better in practice. Again, a detailed analysis of the dynamic programming algorithm probably leads to a better constant factor. It would also be interesting to investigate if the technique of Dorn, based on fast matrix multiplication [9] can help for a further speedup.

If we accept a much higher running time as a function of k, then standard treewidth techniques allow us to solve the FACE COVER TREE and MCC problems in $O(n)$ time for k-outerplanar graphs, k fixed.

4 A PTAS for MCC with geographic clustering

To construct a polynomial time approximation scheme for MCC, we modify Arora's algorithm for ETSP [3, 2]. We assume that the corner points of each of the n rooms have integer coordinates, that each room encloses a $q \times q$ square and has perimeter at most cq, for some constant $c \geq 4$.

4.1 Perturbation and curved dissection

Arora's algorithm for ETSP starts with a perturbation of the instance that, without great increase of the optimum, ensures that in the resulting new instance all nodes lie on the unit grid, and the maximum internode distance is at most $poly(n)$. In MCC, a perturbation is not necessary. All corner points are already on the integer grid. Further, since all rooms are connected and the perimeter of a room is at most cq the smallest *bounding box* (the smallest axis parallel square containing all rooms) has a side length at most cqn. We let the size of the bounding box be $L \in [cqn, 2cqn]$ such that L/cq is a power of 2. On the other hand, a simple packing argument shows that the value of the optimal solution is $OPT = \Omega(qn)$.

First we define the *straight dissection* of the bounding box. We stop the partitioning when the side length of the square is cq. Since $L \leq 2cqn$, the depth of the dissection tree is $O(\log n)$. Let the *level* of a square be its depth from the root in the straight dissection tree and the *level i dissection lines* are the straight lines participating in the division of the level $i-1$ square into level i sub-squares.

A dissection line can cut a room into two or more parts. This causes troubles for the dynamic programming since we have to determine for each room in which square of the dissection it gets connected. To solve this problem we introduce a *curved dissection*.

Consider a horizontal level dissection line. We replace the line by a dissection curve by walking from left to right and whenever we hit the boundary of a room we follow the boundary (in arbitrary direction) until the dissection line is hit again. The obtained curve may go through some boundary segments twice. We shortcut the curve and obtain a simple path partitioning the set of rooms in an upper and lower set. Vertical dissection curves are defined in a similar way. Moreover, we can easily do this such that each horizontal curve crosses each vertical curve exactly once, i.e., the intersection is one point or a simple path. (See Fig. 4 and Fig.5.) Notice that no two horizontal (vertical) dissection curves intersect since, at any point on the curve, the deviation from the dissection line is strictly less than $cq/2$.

The transformation of lines to curves maps each node of the straight dissection tree onto a polygon which we denote by *node polygons* of the *curved dissection* tree of the bounding box.

In Fig. 4 dissection lines are depicted by dotted lines and dissection curves are depicted by fat piece-wise linear curves. Notice that the middle room is crossed by vertical and horizontal dissection lines.

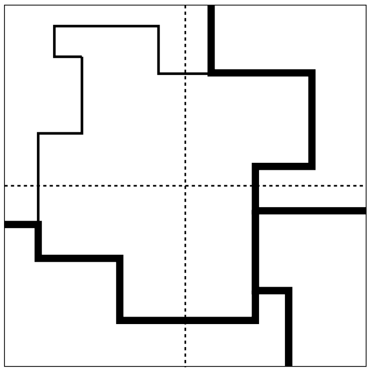

Figure 4: Curved dissection

4.2 Portals and portal respecting trees

Let a level i dissection curve have $2^i m$ special points equally spaced on that curve. Here, by equally spaced we mean that the piece-wise linear fragments of the curve between two consecutive points have the same length. We refer to these points as to *portals* and to number m as to *portal parameter* (to be defined later).

Remember that the intersection of a horizontal and vertical curve is in general a path. The definition above leads to two sets of portals on such paths. We keep only the portals of the highest level curve and pick one set arbitrarily if levels are equal. Further, we define one portal on both endpoints of each path of intersection which we call *corner portals*.

To make the dynamic programming work we have to assume that if some segment of the tree coincides with a dissection curve, it can only connect rooms on one side of the curve. To serve rooms at the other side it has to cross the curve. (See Fig. 5.) We call a feasible tree *portal respecting* if these crossings only appear at portals. We refer to the boundary segment of the node polygon belonging to the dissection curve as to the *side* of node polygon. Notice that sides may overlap. A portal respecting tree is *k-light* if it crosses each side of each node polygon at most k times.

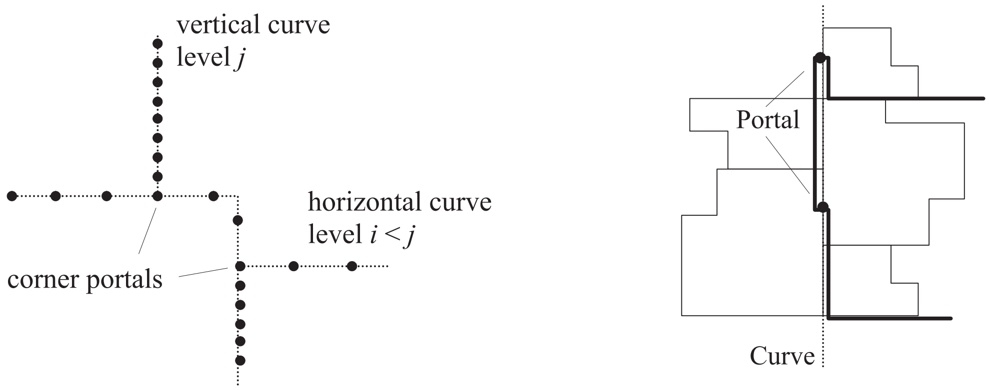

Figure 5: Portals and a feasible portal respecting tree.

4.3 The algorithm

First we construct the bounding box with the dissection curves. Since each room is adjacent to at most two curves the construction can be done in $O(n)$ time. Next we choose $a, b \in \{1, 2, \ldots, L/(cq)\}$ at random and make the a-th horizontal and b-th vertical dissection curve the level zero curves. The curved dissection tree is now build in a wrap-around manner as in Arora [2]. Removing from the 4-ary tree all branches consisting of empty node polygons we derive a tree having at most $O(n)$ leaves and $O(n \log n)$ node polygons. Then we define the portals as in Section 4.2. Starting from the leaves of the dissection tree, in the bottom-up way we update the dynamic programming table. For each node polygon, each k-elementary subset of the portals on the boundary of the polygon, and for each partition $B_1, \ldots, B_p$ of these k portals, we store the length of the optimal forest consisting of p trees which together touch all rooms and the i-th tree connects all portals in B_i.

For the node polygons in the leaves of the dissection tree we simply enumerate all such forests since these polygons contain at most c^2 rooms. For the root polygon we guess the information for the portals on the two level one dissection curves separating the root polygon. We make sure that the four forests together form one tree. The number of different problems in one node polygon is $O(m^{O(k)} f(k))$ for some function f. Taking $m = O(\frac{\log n}{\varepsilon})$ and $k = O(\frac{1}{\varepsilon})$ the size of the look up table is $O(n \log^\gamma n)$, for some constant γ.

4.4 Performance guarantee

The performance guarantee follows from the following theorem.

Theorem 6 (Structure Theorem) *Let $OPT_{a,b,k,m}$ be the length of the minimum k-light portal respecting tree when the portal parameter is m.*

$$E[OPT_{a,b,k,m} - OPT] \leq \left(O\left(\frac{\log n}{m}\right) + O\left(\frac{1}{k-4}\right)\right) OPT,$$

where $E[\cdot]$ is over the random choice of (a, b)-shift.

Proof. Consider an optimal tree in MCC. Let us transform it to a feasible k-light portal respecting tree. First, let us estimate the cost of making the tree *portal respecting*. Whenever the tree crosses a dissection curve we move the crossing to the nearest portal. The transformation is basically the same as in Arora [2] but remember that the intersection of a horizontal and vertical curve is in general a path on which the interportal distance is defined by the highest level curve. In the analysis here we count a crossing of such path with the tree twice and add the cost for making it portal respecting to both curves. Doing this we make sure that we account for the largest interportal distance.

The length of a dissection curve is $O(L)$, where c is hidden in the constant. Given this, the maximum interportal distance for a level i curve, and therefore a detour length for a level i curve is $O(\frac{L}{2^i m})$. Under a random choice of (a, b)-shift the probability that a dissection curve is a level i curve is at $2^{i-1}cq/L$. Therefore, the expected length of a detour is

$$\sum_{i=0}^{\log(L/(cq))} O\left(\frac{L}{2^i m}\right) \cdot \frac{2^{i-1}cq}{L} = \log(L/(cq)) \cdot O\left(cq/m\right) = O\left(\frac{L \log n}{nm}\right)$$

By linearity of expectations the total length of all detours is $O\left(\frac{L \log n}{nm}\right) Q$, where $Q = O(n)$ is twice the number of times the optimal tree crosses the dissection curves. The expected cost of the transformation of the optimal tree to the portal respecting tree is $O(\frac{\log n}{m})OPT$.

Now we make a portal respecting tree *k-light*. Let us call a side of the node polygon *overloaded* if the optimal tree crosses it more than $k - 2$ times.

In a bottom-up fashion starting from leaves of the dissection tree, for overloaded sides of the node polygons we apply the following *patching procedure*. Whenever a side of a node polygon is overloaded, we add this side on each side of the dissection curve to the tree and make one crossing of the curve at an arbitrary portal. It is important that we do these replacements in the right order. Since horizontal and vertical curves overlap, the patching of horizontal curves may add intersections on vertical curves and vice versa. In Arora [2] this problem appears only level down, i.e., patching on a vertical curve of level i may add intersections on a horizontal curve of level $j \geq i$. In our case this problem appears also level up. When we do the patching on a horizontal level i curve this may add crossings also on a vertical level $j \leq i$ curve. Fortunately, this problem is solved by doing the patching bottom up. However, unlike in the TSP problem we have to do this simultaneously for horizontal and vertical curves. We start the patching with all curves (horizontal and vertical) of the lowest level and work level by level until we reached the level zero curves. Hence, each curve will be considered only once.

Now consider a horizontal level i curve s_h. If the side of an adjacent level i polygon contains more than $k - 2$ crossings we add this side on each side of the dissection curve to the tree and make one crossing of the curve at an arbitrary portal. Besides this single crossing this replacement may add crossings to vertical curves. Notice that for such a curve s_v of level $j \geq i$ these additional crossings only appear at the corner portals of the intersection of s_h and s_v. Hence, once we have completed the patching up to level i, then each side of a level $j \geq i$ node polygon may receive additional crossings only at the corner portals at the end of the side. Now we will count the number of times we apply the patching. Each time we apply it we remove more than k crossings and add one new crossing. Further we may add one crossing to a higher level curve, one crossing to a curve of the same level and many crossings to higher level curves. Therefore, we add at most three crossings that will be considered later in the procedure. Since the total number of crossings of the optimal tree is $O(n)$, the total number of times we apply the patching is $O(n/(k-4))$. Each time we apply the patching to curve s_h of level i we add twice the side of a level i square to the tree, whence we add $O(L/2^i)$. On the other hand, the probability that the curve s_h is of level i is $2^i cq/L$ making the expected additional length of the tree for any patching $O(L/2^i) \cdot 2^i cq/L = O(q)$. By linearity of expectation the total expected additional cost is $O(nq/(k-4)) = O(OPT/(k-4))$. □

Taking $m = O(\frac{\log n}{\varepsilon})$ and $k = O(\frac{1}{\varepsilon})$ in Theorem 6 we derive the following result.

Theorem 7 *The randomized algorithm described above returns a feasible tree of length at most* $(1+\varepsilon)OPT$ *in time* $n(\log n)^{O(1/\varepsilon)}$.

To derandomize the algorithm we can simply go through all possible choices for a and b. More sophisticated derandomization techniques are described in Rao and Smith [29]. In fact, straightforward adaption of more careful analysis presented in [29] can also significantly speed up the algorithm presented in this paper. For two dimensional space this would imply even $O(n \log n)$ time and $O(n)$ space PTAS for MCC and other geometric problems with geographic clustering.

4.5 Extensions of the PTAS

As in Arora [3, 2] we did not use much of the specific structure of MCC. The basic idea to tackle the generalized geometric problems with geographic clustering is to introduce the curved dissection, new stoppage criteria and then to use the fact that under geographic clustering the lengths of the dissection curves differ only a constant factor from the lengths of the dissection lines, yielding the same (up to a constant factor) charges to the objective function as in non-generalized versions of the geometric problems. In this way, with slight modifications in the analysis of the algorithm, we can derive PTASs for GTSP, GSTP, GMST and many other generalized geometric problems. Moreover, the approach is naturally applicable to many other norms, e.g., we can straightforwardly adopt the approximation scheme to any L_p norm. Also notice, that the requirement that partition of the polygon must be rectilinear is not crucial. It is sufficient to assume that the walls of each room are given by a sequence of line segments forming a simple closed walk in the plane (here, the only critical assumption is that all rooms must be fat and have comparable sizes, i.e., for each room its perimeter must be bounded by cq where q is the size of the minimal over all rooms maximum inscribed square or ball and c is a fixed constant).

Dumitrescu and Mitchell in [11] pointed out that in their approximation scheme for GTSP only some of the arguments for disjoint discs can be lifted to higher dimensions and, naturally, one of the open questions they listed was: "What approximation bounds can be obtained in higher dimensions?" It is well known, see e.g. [3, 2, 29], that the Arora's algorithm for ETSP is applicable also in higher fixed dimensional spaces. Using literally the same argumentation as in [3] and our construction for MCC with geographic clustering, one can derive the following theorem.

Theorem 8 *If the corner points of the rooms are in* $\mathcal{R}^d$*, the MCC with geographic clustering admits randomized PTAS running in* $n(\log n)^{(O(\sqrt{d}/\varepsilon))^{d-1}}$ *time. Derandomization of the algorithm in this case will cost an additional factor of* $O(n^d)$ *leading to overall running time* $n^{d+1}\ (\log n)^{(O(\sqrt{d}/\varepsilon))^{d-1}}$.

The same holds for GTSP, GSTP and GMST. This resolves the open question from Dumitrescu and Mitchell [11].

5 An approximation algorithm for MCC with rooms of varying sizes

Elbassioni et al. [12] give a simple constant factor approximation algorithm for GTSP, where the factor depends on the fatness of the regions. Here we modify their algorithm and proof to obtain a constant factor approximation algorithm for MCC.

For any room R_i, $i \in \{1, \ldots, n\}$, we define its *size* ρ_i as the side length of the smallest enclosing square of the room. We restrict to rooms for which the perimeter is bounded by the size of the room, lets say at most $4\rho_i$. A room R is said to be *α-fat* if for any square Q which boundary intersects R and whose center lies in R, the area of the intersection of R and Q is at least $\alpha/4$ times the area of Q. Notice that the fatness of a square is 1 and in general $\alpha \in]0, 1]$.

Algorithm Greedy:

(1) Pick the corner points $p_i \in R_i$, $i \in \{1, \ldots, n\}$, that minimize $\sum_{i=2}^{n} d(p_1, p_i)$, where $d(x, y)$ is the shortest distance between x and y along the walls.

(2) Let G be a graph with a vertex v_i for every room R_i and $d(v_i, v_j) = d(p_i, p_j)$. Find a minimum spanning tree T in G.

(3) Construct a solution to MCC as follows. For every edge (v_i, v_j) in T, let the minimum length (p_i, p_j)-path belong to the corridor. If the resulting corridor is not a tree, break the cycles (removing edges) arbitrarily.

Lemma 9 *Algorithm* Greedy *gives an $(n-1)$-approximate solution for MCC.*

Proof. Consider an optimal solution and let OPT be its length. Identify for each room R_i a point p'_i in the room that is connected to the optimal tree. The optimal tree contains a path from p'_1 to p'_i for all $i \in \{2, \ldots, n\}$. Therefore, $(n-1)OPT \geq \sum_{i=2}^{n} d(p'_1, p'_i) \geq \sum_{i=2}^{n} d(p_1, p_i)$, which is at most the length of the tree constructed by the algorithm. □

Lemma 10 *The length of the shortest corridor that connects k rooms is at least $\rho_{\min}(k\alpha/2 - 2)$, where $\rho_{\min}$ is the size of the smallest of these rooms.*

Proof. Let P be a connecting corridor and let $d(P)$ denote its length (along the walls). Let the center of a square with side length $2\rho_{\min}$ follow corridor P. The total area A covered by the moving square is at most $(2\rho_{\min})^2 + 2\rho_{\min} \cdot d(P)$. On the other hand, suppose a room is connected with P at point p. Putting the center of the square in point p we see that its boundary intersect the room. By definition of α at least a fraction $\alpha/4$ of the room is contained in the square. Therefore, $k(2\rho_{\min})^2\alpha/4$ is a lower bound on the area A. We have $k(2\rho_{\min})^2\alpha/4 \leq A \leq (2\rho_{\min})^2 + 2\rho_{\min} \cdot d(P)$, yielding $d(P) \geq \rho_{\min}(k\alpha/2 - 2)$, that completes the proof. □

Algorithm Connect:

(1) Order the rooms by their sizes $\rho_1 \leq \rho_2 \leq \ldots \leq \rho_n$. Pick any p_1 on the boundary of R_1. For $i = 2$ up to n pick the point p_i in R_i that minimizes $\min\{d(p_i, p_1), d(p_i, p_2), \ldots, d(p_i, p_{i-1})\}$, i.e., pick the point that is closest to the already chosen points.

(2) Let G be a graph with a vertex v_i for every room R_i and $d(v_i, v_j) = d(p_i, p_j)$. Find a minimum spanning tree T in G.

(3) Construct a solution to MCC as follows. For every edge (v_i, v_j) in T, let the minimum length (p_i, p_j)-path belongs to the corridor. If the resulting corridor is not a tree, break the cycles (removing edges) arbitrarily. Output the minimum of the obtained tree and the tree constructed by algorithm GREEDY.

Theorem 11 *Algorithm* CONNECT *gives a* $(16/\alpha - 1)$*-approximate solution for the minimum corridor connection in which the fatness of every room is at least* α*.*

Proof. If $n - 1 \leq 16/\alpha - 1$ then GREEDY guarantees a $16/\alpha - 1$approximation. So assume $n \geq 16/\alpha$. Denote the set of points chosen by CONNECT as $P' = \{p_1, \ldots, p_n\}$. Let p_i^* be the point from $\{p_1, \ldots, p_{i-1}\}$ that is at minimum distance from p_i. Denote the distance $d(p_i, p_i^*)$ by x_i.

Consider some closed closed walk Ω connecting all rooms and assume its length is minimum. The length of this walk is clearly an upper bound on OPT. For each room R_i, $i \in \{1, \ldots, n\}$, we define one connection point r_i on Ω in which it hits the room. Consider one of the two possible directions of Ω and assume that the tour connects the rooms in the order $1, 2, \ldots, n$. Let $k \in \{1, \ldots, n\}$. We define T_i as the part of this directed walk that connects exactly k rooms at their connection points and starts from point r_i. Let t_i be the length of the (not necessarily simple) path T_i. We have $OPT \leq d(\Omega) = \sum_{i=1}^{n} t_i/(k-1)$.

Consider some $i \in \{1, \ldots, n\}$ and let $R_{h(i)}$ be the smallest room among those from the k rooms on the path T_i. Since R_i is on this path T_i and we ordered the rooms by their size we may assume $1 \leq h(i) \leq i$. We partition the rooms in two sets. Let F be the set of rooms for which $h(i) = i$ and let H contain the remaining rooms. Let T' be an MST on the point set P' restricted to the rooms in F. Then $d(T') \leq OPT + 2\sum_{i \in F} \rho_i$. The connected graph that we construct consists of the edges of T' and for all rooms i in H we add the path (p_i, p_i^*) which has length x_i. Notice that the resulting graph is indeed connected and has total length at most

$$OPT + \sum_{i \in F} 2\rho_i + \sum_{i \in H} x_i.$$

We define $\gamma = k\alpha/2 - 2$. From Lemma 10 we know

$$t_i \geq \gamma \rho_i, \text{ for all } i \in F. \tag{1}$$

If $i \in H$, then we argue as follows. Since the algorithm picked point p_i we know that the distance from any point in R_i to the point $p_{h(i)}$ (which is chosen before p_i) is at least x_i.

Hence, the distance from any point in R_i to any point in $R_{h(i)}$ is at least $x_i - 2\rho_{h(i)}$, implying $t_i \geq x_i - 2\rho_{h(i)}$. Additionally, we know from Lemma 10 that $t_i \geq \gamma\rho_{h(i)}$. Combining the two bounds we get

$$t_i \geq \max\{\gamma\rho_{h(i)}, x_i - 2\rho_{h(i)}\} \geq \frac{\gamma}{\gamma+2} x_i, \text{ for all } i \in H. \tag{2}$$

Combining (1) and (2) we see that the MST given by the algorithm has length at most

$$\begin{aligned} OPT + \sum_{i \in F} 2/\gamma t_i + \sum_{i \in H} (1+2/\gamma) t_i &\leq OPT + \sum_{i=1}^{n} (1+2/\gamma) t_i \\ &\leq OPT + (1+2/\gamma)(k-1)OPT \\ &= OPT + (1+2/(k\alpha/2-2))(k-1)OPT \\ &= OPT + \tfrac{k(k-1)}{k-4/\alpha} OPT \end{aligned}$$

It is easy to show that $k(k-1)/(k-4/\alpha)$ equals $16/\alpha - 2$ for $k = 8/\alpha - 1$ and also for $k = 8/\alpha$. Further, it is strictly smaller for any value in between. Hence, there is an integer $k \in [8/\alpha - 1, 8/\alpha]$ such that $k(k-1)/(k-4/\alpha) \leq 16/\alpha - 2$. Notice that by the assumption in the first line of the proof we satisfy $k \in \{1, \ldots, n\}$. We conclude that the length of the tree given by the algorithm is at most $(16/\alpha - 1)OPT$. □

References

[1] E.M. Arkin and R. Hassin, Approximation algorithms for the geometric covering salesman problem, Discr. Appl. Math. 55 (1994) 197-218.

[2] S. Arora, Approximation schemes for NP-hard geometric optimization problems: A survey, Math. Prog. 97 (2003) 43-69.

[3] S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, J. ACM 45 (1998) 1-30.

[4] T.C. Biedl and G. Kant, A better heuristic for orthogonal graph drawings, Comput. Geom. 9 (1998) 159-180.

[5] H.L. Bodlaender, A partial k-arboretum of graphs with bounded treewidth, Theoret. Comput. Sci. 209 (1998) 1-45.

[6] H.L. Bodlaender, C. Feremans, A. Grigoriev, E. Penninkx, R. Sitters and T. Wolle, On the minimum corridor connection and other generalized geometric problems, in: Proc. WAOA'06, Lecture Notes in Computer Science, Vol. 4368 (Springer, Berlin, 2007) 69-82.

[7] M. de Berg, J. Gudmundsson, M. Katz, C. Levcopoulos, M. Overmars, and A. van der Stappen, TSP with neighborhoods of varying size, J. Algor., 57 (2005) 22-36.

[8] E.D. Demaine and J. O'Rourke, Open problems and planning, in: Proc. CCCG'00 (Fredericton, New Brunswick, Canada, 2000).

[9] F. Dorn, Dynamic programming and fast matrix multiplication, in: Proc. ESA'06, Lecture Notes in Computer Science, Vol. 4168 (Springer, Berlin, 2006) 280-291.

[10] F. Dorn, E. Penninkx, H.L. Bodlaender and F.V. Fomin, Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions, in: Proc. ESA'05, Lecture Notes in Computer Science, Vol. 3669 (Springer, Berlin, 2005) 95106.

[11] A. Dumitrescu and J.S.B. Mitchell, Approximation algorithms for TSP with neighborhoods in the plane, J. Algor. 48 (2003) 135-159.

[12] K. Elbassioni, A.V. Fishkin, N.H. Mustafa and R. Sitters, Approximation algorithms for Euclidean group TSP, in: Proc. ICALP'05, Lecture Notes in Computer Science, Vol. 3580 (Springer, Berlin, 2005) 1115-1126.

[13] C. Feremans, Generalized Spanning Trees and Extensions, Ph.D. Thesis, Université Libre de Bruxelles, Brussels, 2001.

[14] C. Feremans and A. Grigoriev, Approximation schemes for the generalized geometric problems with geographic clustering, in: Proc. EWCG'05 (Eindhoven University of Technology, Eindhoven, The Netherlands, 2005) 101-102.

[15] C. Feremans, M. Labbé and G. Laporte, Generalized network design problems, Europ. J. Oper. Res. 148 (2003) 1-13.

[16] M. Fischetti, J.J. Salazar, and P. Toth, A branch-and-cut algorithm for the symmetric generalized traveling salesman problem, Oper. Res. 45 (1997) 378-394.

[17] F.V. Fomin and D.M. Thilikos, New upper bounds on the decomposability of planar graphs, J. Graph Theory 51 (2006) 53-81.

[18] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness (W.H. Freeman, San Francisco, 1979).

[19] M.R. Garey and D.S. Johnson, The rectilinear Steiner tree problem is NP-complete, SIAM J. Appl. Math. 32 (1977) 826-834.

[20] A. Gonzalez-Gutierrez and T.F. Gonzalez, Complexity of the minimum-length corridor problem, Comput. Geom. 37 (2007) 72103.

[21] Q. Gu and H. Tamaki, Optimal branch-decomposition of planar graphs in $O(n^3)$ time, in: Proc. ICALP'05, Lecture Notes in Computer Science, Vol. 3580 (Springer, Berlin, 2005) 373-384.

[22] C.S. Helvig, G. Robins and A. Zelikovsky, An improved approximation scheme for the Group Steiner Problem, Networks 37 (2001) 8-20.

[23] I.V. Hicks, Planar branch decompositions I: The ratcatcher, INFORMS J. Comput. 17 (2005) 402-412.

[24] I.V. Hicks, Planar branch decompositions II: The cycle method, INFORMS J. Comput. 17 (2005) 413-421.

[25] R.J. Lipton and R.E. Tarjan, A separator theorem for planar graphs, SIAM J. Appl. Math. 36 (1979) 177-189.

[26] C.S. Mata and J.S.B. Mitchell, Approximation algorithms for geometric tour and network design problems, in: Proc. SoCG'95 (ACM, 1995) 360-369.

[27] J.S.B. Mitchell, A PTAS for TSP with neighborhoods among fat regions in the plane, in: Proc. SODA'07 (ACM-SIAM, 2007) 11-18.

[28] J.S.B. Mitchell, Geometric shortest paths and network optimization, in: Handbook of Computational Geometry (Elsevier, North-Holland, Amsterdam, 2000) 633-701.

[29] S. Rao and W.D. Smith, Approximating geometric graphs via "spanners" and "banyans", in: Proc. STOC'98 (ACM, 1998) 540-550.

[30] G. Reich and P. Widmayer, Beyond Steiner's problem: a VLSI oriented generalization, in: Proc. WG'89, Lecture Notes in Computer Science, Vol. 411 (Springer, Berlin, 1990) 196-210.

[31] S. Safra and O. Schwartz, On the complexity of approximating TSP with neighborhoods and related problems, in: Proc. ESA'03, Lecture Notes in Computer Science, Vol. 2832 (Springer, Berlin, 2003) 446-458.

[32] P.D. Seymour and R. Thomas, Call routing and the ratcatcher, Combinatorica 14 (1994) 217-241.

[33] M. Zachariasen and A. Rohe, Rectilinear group Steiner trees and applications in VLSI design, Math. Prog. 94 (2003) 407-433.