On the minimum corridor connection problem and other generalized geometric problems ⋆,⋆⋆

Hans L. Bodlaender [a] Corinne Feremans [b]
Alexander Grigoriev [b,*] Eelko Penninkx [a] René Sitters [c]
Thomas Wolle [d,1]

[a] *Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands*

[b] *Department of Quantitative Economics, Maastricht University, P.O.Box 616, 6200 MD Maastricht, The Netherlands*

[c] *Eindhoven University of Technology, Department of Mathematics and Computer Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

[d] *NICTA Sydney, Locked Bag 9013, Alexandria NSW 1435, Australia*

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.

Key words: minimum corridor connection, generalized geometric problems, complexity, exact algorithms, approximations
PACS:

1 Introduction

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 as the *minimum corridor connection* problem.

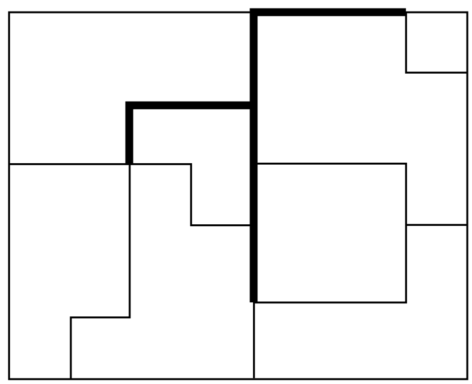

Fig. 1. Minimum length tree along the corridors to connect the rooms

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, the minimum corridor connection problem can be viewed as a special case of the generalized geometric Steiner tree problem, where given a

⋆ This work was supported by the Netherlands Organization for Scientific Research NWO (project *Tree-width and Combinatorial Optimization*).

⋆⋆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.

∗ Corresponding author. Tel.: +31 43 3883853; fax: +31 43 3884874

Email addresses: hansb@cs.uu.nl (Hans L. Bodlaender), c.feremans@ke.unimaas.nl (Corinne Feremans), a.grigoriev@ke.unimaas.nl (Alexander Grigoriev), penninkx@cs.uu.nl (Eelko Penninkx), r.sitters@tue.nl (René Sitters), thomas.wolle@nicta.com.au (Thomas Wolle).

URLs: http://people.cs.uu.nl/hansb/ (Hans L. Bodlaender), http://www.personeel.unimaas.nl/a.grigoriev/ (Alexander Grigoriev), http://www.win.tue.nl/~sitters/ (René Sitters).

[1] NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

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 Euclidean (or geometric) *traveling salesman problem* (TSP). Assume that a salesman has to visit n customers. Each customer has a set of specified locations (points) 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, with respect to Euclidean norm, closed salesman tour visiting each customer. This generalization is known as the geometric covering salesman problem [1], or Euclidean TSP with neighborhoods [7,11,27], or the generalized Euclidean TSP [16]. Notice that a region, as in the problem definition, is not necessarily connected. If each region is a single point, the problem becomes classic Euclidean TSP.

Similarly, one can define the generalized geometric minimum Steiner tree problem (given n regions in the plane, find the shortest network connecting them), the generalized geometric minimum spanning tree problem (given n regions in the plane, find the shortest tree such that each vertex of the tree is a point in a region and each region is hit by the tree), and many other generalized 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 6 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 the minimum corridor connection problem, appear also in constructions where, e.g., wiring has to be installed along the walls, floors and ceilings of a 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]. Until recently nothing was known on the 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 the generalized Euclidean TSP it is known that the problem cannot be efficiently approximated within $(2-\varepsilon)$ unless P=NP, see [31]. On the positive side, Mata and Mitchell [26] gave a polynomial time $O(\log n)$-approximation algorithm. Constant factor approximations were developed for the special cases where neighborhoods are disjoint convex fat objects [7,12], disjoint unit disks [1], and intersecting unit disks [11]. Dumitrescu and Mitchell [11] have developed a polynomial time approximation scheme (PTAS) for the case of the generalized Euclidean TSP with regions given by nearly disjoint nearly-unit disks, i.e., the disks become pairwise disjoint if all of them are decreased in size by a constant factor while keeping the center points the same, and there is a constant upper bound on the ratio of radii of any two disks. Recently, Mitchell [27] drastically improved this result showing that the problem restricted to general fat objects already admits a PTAS.

For the generalized Steiner tree problem, Helvig, Robins and Zelikovsky [22] developed a polynomial time n^{ε}-approximation algorithm where $\varepsilon > 0$ is any fixed constant. Finally, many exact search methods and heuristics for the generalized geometric problems were 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 two talks [14,6] presented at the European Workshop on Computational Geometry EWCG 2005 and the Workshop on Approximation and Online Algorithms WAOA 2006, respectively. 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, our reduction relies on a class of instances of the minimum corridor connection problem that is completely different from the instances in [20]. Also, Gonzalez-Gutierrez and Gonzalez [20] require in the definition of the problem that the corridor includes a point of the outer boundary of the polygon, while we do not have this restriction.

In Section 3 we present a subexponential time exact algorithm for the minimum corridor connection 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 the minimum corridor connection with fat rooms having nearly the same size, that partially solves another open question from CCCG 2000 on the approximability of the problem, see [8]. In Section 5 we show how the algorithm for the generalized Euclidean TSP from Elbassioni et al [12] can be used to derive a polynomial time constant approximation algorithm for the minimum corridor connection problems with fat rooms of varying sizes. This complements our partial answer on the open question from CCCG 2000 on the approximability of the problem, see [8]. Finally, in Section 6 we discuss the applications of our algorithms to the other generalized geometric problems.

2 Complexity of the minimum corridor connection problem

In this section, we show that the decision version of the minimum corridor connection problem is strongly NP-complete.

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

PROOF. Clearly, the decision version of the minimum corridor connection problem belongs to NP. To prove NP-completeness, we use a transformation from the *connected vertex cover problem* for planar graphs with maximum degree four. In this latter problem, we are given a planar graph $G = (V, E)$ such that each vertex in V has degree at most 4, and a positive integer $R \leq |V|$; and the question is: does there exist 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 $\{u, v\} \in E$? The connected vertex cover problem for planar graphs with maximum degree four is NP-complete, see [18,19]. We transform the instance of the connected vertex cover problem into an instance of the minimum corridor connection problem in a number of steps.

(1) *Make a rectilinear embedding of G.*
Let $n = |V|$. Use the algorithm of Biedl and Kant [4] to find a rectilinear embedding of G in an $n \times n$ grid such that each edge has at most two bends. This rectilinear embedding assigns to each vertex of V and to each bend point a point in the plane with integer coordinates in $\{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$; each edge is represented by a sequence of horizontal and vertical segments, and the length of the edge, denoted by ℓ_e ($e \in E$), is the total length of the segments.

(2) *Enlarge the drawing.*
Let $L = \max_{e \in E} \ell_e$ and 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.*
In this step, we transform each edge into a very narrow room in the following way.

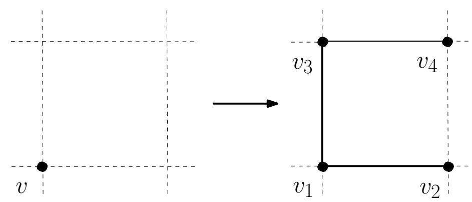

Fig. 2. Replacing vertices

We replace each vertex by four vertices at the corners of a one by one square, i.e., if $v = (i, j)$ is a vertex of G in the constructed embedding, let $v_1 = (i, j)$, $v_2 = (i + 1, j)$, $v_3 = (i, j + 1)$, and $v_4 = (i + 1, j + 1)$ be the corresponding four vertices. Introduce three additional edges on three sides of the unit square: (v_1, v_2), (v_1, v_3) and (v_3, v_4). Call these edges *short walls*, see Fig. 2.

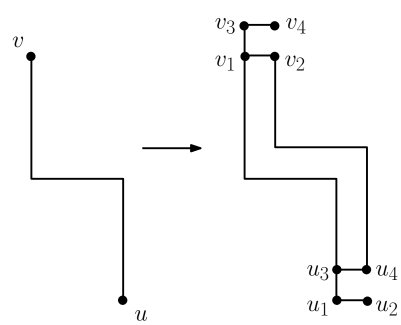

Fig. 3. Transforming edges to rooms

Alongside the original path representing an edge, we take two parallel paths with distance one from each other such that these two paths and two short walls at the vertices form a face in the new drawing. Call these paths *long walls*, see Fig. 3.

We now distinguish two types of faces: *edge rooms* (faces of "width one" representing edges in G) and *original rooms* (all other faces in the new drawing).

(4) *Make long walls have equal length.*
The total length of a long wall is at most $2KL + 4$: by definition of L, edges were represented by a path of length at most 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). Notice also that the total length of a long wall is at least $2K$.

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

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 walls have the required length $2KL + 3$ or $2KL + 4$.

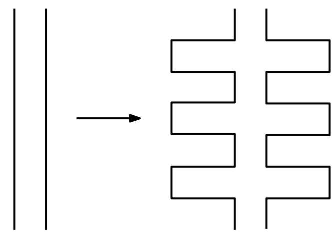

Fig. 4. Making edges longer

Also, note that the step does not change the faces, except for the lengths of their boundary edges.

For an example of complete transformation see Fig. 5. Here, in (a) we have the input graph. A rectilinear embedding of this graph is illustrated in (b). In (c) we schematically show how edge rooms are created, where "schematically" means that we do not properly enlarge/stretch the drawing as described in Step 2. Here, the figure is provided only to illustrate the faces in the new drawing. Finally, in (d) we also schematically show how to make long walls of nearly equal length.

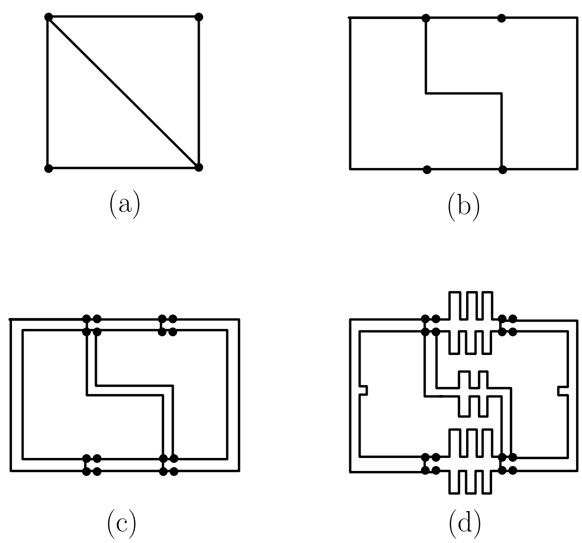

Fig. 5. An example of complete transformation

Consider the resulting diagram. It is a rectilinear decomposition of a rectilinear polygon. We claim that there is a tree along the walls of the subdivided polygon, such that each room is incident with a vertex of the tree and the tree has total length of 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 G such that $|W| \leq R$. 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 rooms in the instance of the minimum corridor connection problem. For each edge in T, take one of the two long walls of the corresponding edge room, and add that to T'. To this, we add for each vertex in W, the three corresponding short walls. The total length hence is at most $(R-1)(2KL+4)+3|W| \leq 2KL(R-1)+7R$. Tree T' hits all the rooms. Indeed, consider any edge room. As W is a cover in G, at least one of the endpoints of the corresponding edge is in W. By construction, all short walls corresponding to this endpoint are in T'. Therefore, the edge room is incident to a vertex in T'. An original room is hit: look at one of its incident edges; at least one endpoint of the edge belongs to W and hence contains a vertex in T' that is incident to the room.

Suppose we have a tree T' along the edges of the subdivided polygon such that each room is incident with a vertex of the tree, and T' 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 walls of the edge 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 F contributes at least $2K+3$ to the length of T'. Here, 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 edge 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 $K_{1,n-1}$.

To complete the proof of strong NP-completeness, we observe that all numbers in our constructed instance are integers bounded by a polynomial in n, because $L = O(n)$ and hence $K = O(n^2)$. And we also observe that the steps of the transformation can be carried out in polynomial time. □

3 Exact algorithms with branch-width

In this section, we discuss how the problem can be solved exactly exploiting the notion of branch-width 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 *branch-width* of a graph is the minimum width over all branch decompositions. For an example illustrating some of these notions, see Fig. 6.

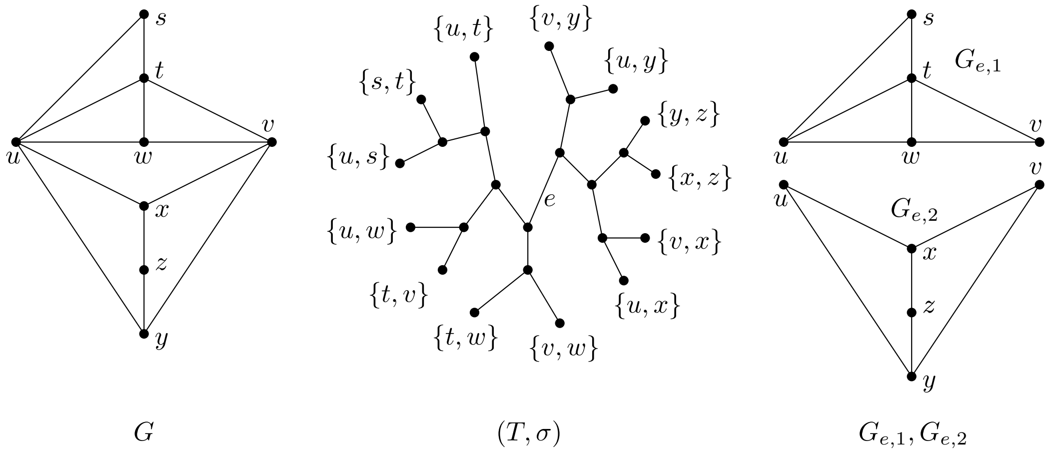

Fig. 6. A graph G with 8 vertices and 13 edges; a branch decomposition of G where σ is given by the labeling of the leaves of T; the graphs $G_{e,1}$ and $G_{e,2}$ as determined by e; the middle set of e is $\{u, v\}$, and hence the width of (T, σ) is at least 2.

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.

For a planar graph on n vertices, a sphere cut decomposition of width equal to the branch-width of the graph 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 minimum corridor connection problem, we consider a generalization, which we call the *face cover tree problem*. In the face cover tree problem, we are given a planar graph $G = (V, E)$, with edge weights $w : E \to \mathbf{N}$, and the problem is to find a subtree T of G of minimum total weight such that each face of G is incident with at least one vertex from T.

Interestingly, a solution for the face cover tree problem does not depend on the embedding of the graph, because 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 leave the proof of this fact as a simple exercise for a reader.

We now give an algorithm that solves the face cover tree problem using a sphere cut decomposition of G.

Theorem 2 *Suppose a planar graph $G = (V, E)$ is given together with a sphere cut decomposition (T, σ) of G of width at most k. Then the face cover tree problem can be solved in $O((3+\sqrt{5})^{2k} \cdot k \cdot n)$ time.*

PROOF. To obtain this result, we use known approaches based on dynamic programming on the decomposition tree T; and for the analysis, we use non-standard techniques from Dorn et al. [10].

The basic idea is to build a table for each edge of T representing (partial) solutions to the problem (of a subgraph) of G. We start with tables that correspond to edges incident to a leaf of T. Then with dynamic programming, we computes all other tables using previously computed tables.

To make this more precise, let some arbitrary node r be the root of T. Now, to each edge e of T, let E_e be the set of edges of G, associated with leaves that are below e in the tree T; and let G_e 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 branch decompositions.

Slightly abusing notation, we consider a forest as a collection of trees. Now, let T' be a forest 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 tree 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 the forest T' has only one tree that touches each face of G, or each tree in the forest 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.

We *characterize* forests T' in G_e by three criteria: the set of vertices in the middle set of e that belong to T', the equivalence relation on these vertices w.r.t. 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 tree 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 tree problem. For each edge e in T, we compute a table of all characterizations of forests T' of G_e, such that each tree in the forest 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 store the minimum weight over all forests T' of G_e that have this characterization.

In other words, the characterization of a forest of G_e can be expressed by a triple (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. A triple (S, R, X) is *realized* by a forest T' of G_e, if S is the set of vertices in the middle set that belong to T', R is the relation on S that represents which vertices are connected in T', and X is the set of faces intersecting the noose of e that are touched by e. In our dynamic programming algorithm, we tabulate for each edge e in T, for each triple (S, R, X) that is realized by at least one forest, the minimum total weight over all forests T' that realizes that triple.

Lemma 3 *If e in T 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 plane), 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. A 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 $\{1_[, 1_], 1_\square, 1_m, 0_t, 0_n\}^*$. 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 $\{1_[, 1_], 1_\square, 1_m, 0_t, 0_n\}^*$ 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 $C_3 = \{\{1_[, 1_], 1_\square, 1_m\}$. 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 $3+\sqrt{5}$. □

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 characterization. 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. □

Note that $(3+\sqrt{5})^2 = 14 + 6\sqrt{5} \leq 2^{4.7770}$, i.e., we have an algorithm for the face cover tree problem that uses $O(2^{4.7770k}kn)$ time.

With a more detailed analysis, it is probably possible to reduce the running time. 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 where 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 planar 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 planar embedding with at most k non-empty layers. It is well known that a k-outerplanar graph has branch-width at most $2k$; this can be proved in the same way as the proof in [5] that k-outerplanar graphs have tree-width 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 branch-width (and tree-width) $O(\sqrt{n})$. (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 branch-width at most $\sqrt{4.5 \cdot n}$.*

Thus we have the following consequences.

Corollary 5 *The face cover tree problem, and hence also the minimum corridor connection 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 tree-width techniques allow us to solve the face cover tree problem and the minimum corridor connection problems in $O(n)$ time for k-outerplanar graphs, when k is fixed.

4 A PTAS for the problem with fat rooms of nearly the same size

We construct a polynomial time approximation scheme for the minimum corridor connection problem based on Arora's algorithm for Euclidean traveling salesman problem [2,3].

We assume that the corner points of rooms have integer coordinates and that that the area of each room it at least q^2, for some $q \geq 1$. Since rooms are rectilinear, the perimeter of each room will then be at least $4q$. We will also assume that the perimeter of each room is at most cq for some $c \geq 4$. Note that under these assumptions the smallest enclosing square of a room has side length strictly less than $cq/2$. We will assume that $n \geq 8c^2$. We may do this since otherwise the optimal tree is easily found in polynomial time if c is constant, for example by guessing a connection point for each room and then computing a minimum spanning tree on these points. Notice, while c is considered a constant, the number q is part of the input.

4.1 Grid lines and grid curves

We define an axis-aligned, square *bounding box* $\mathcal{B}$ containing all rooms. Since all rooms are connected and the perimeter of a room is at most cq, the smallest bounding box has side length at most $cqn/2$.

Let $d = 4c^3q$ and let the side length l of the bounding box be in the interval $[cqn/2, cqn]$ such that l/d is a power of 2, say $l = 2^D d$. Note that $d = 4c^3q \leq cqn/2 \leq l$. Hence, D is a non-negative integer and $2^D = l/d \leq cqn/(4c^3q) = n/(4c^2)$. The box is divided into 2^{2D} squares by adding $2^D - 1$ horizontal and $2^D - 1$ vertical *grid lines* with inter distance d. We number these two sets of lines from 1 to $2^D - 1$, starting in the upper left corner.

A grid line may cut a room into two or more parts. To facilitate the dynamic programm we replace every grid line by a *grid curve*, or simply a *curve*. Consider any horizontal grid line and consider the (finite) set of points where it intersect a vertical boundary of a room. For any two consecutive points there is room whose boundary contains both points. So we can walk from one intersection point to the next by following the boundary of a single room. (See Figure 7.) Notice that the obtained curve may go over a room boundary more than once. We shortcut the curve and obtain a simple path partitioning the set of rooms in an upper and lower set. No two horizontal curves intersect

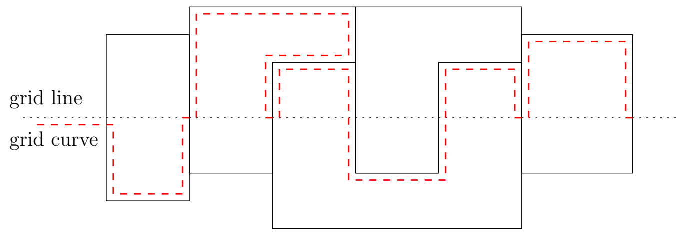

Fig. 7. A grid line and a possible grid curve (before short cutting).

since at any point on the curve, the deviation from the grid line is strictly less than $cq/2 < d/2$.

We have to be careful with the definition of vertical grid curves. First, we construct vertical grid curves similar to the horizontal curves. Again, these vertical curves do not intersect. The intersection of a horizontal curve and a vertical curve is not necessarily a single point or even a simple path. It may be a disconnected set of line segments. Let $\mathcal{H}_i$ be the i-th horizontal grid curve and let $\mathcal{V}_j$ be the j-the vertical grid curve. We change each vertical curve $\mathcal{V}_j$ to a new curve $\widehat{\mathcal{V}}_j$ in the following way. Let p_{ij} be the first point of intersection between $\mathcal{H}_i$ and $\mathcal{V}_j$ when walking down along $\mathcal{V}_j$, and let q_{ij} be the last point of intersection. We replace the part of $\mathcal{V}_j$ between p_{ij} and q_{ij} by the part of $\mathcal{H}_i$ between p_{ij} and q_{ij}. Now, any intersection is a simple path. We call this a *shared path*; see the left part of Figure 8.

Next, we prove that no two vertical curves $\widehat{\mathcal{V}}_j$ and $\widehat{\mathcal{V}}_k$ intersect. We do this by showing that, at any point p on a vertical curve $\widehat{\mathcal{V}}_j$, the deviation from the j-th grid line is less than $d/2$. We need the following lemma.

Lemma 6 *Let p and p' be two points on a horizontal curve such that the abscissa of p and the abscissa of p' differ by at most x. Then the distance on the curve between p and p' is at most $c^2x + c^3q$.*

PROOF. The path from p to p' follows the boundary of rooms that intersect the corresponding grid line. Since the height and width of any room is less than $cq/2$, all these rooms fit in a rectangle of height cq and width $x + cq$. Since the area of a room is at least q^2, the number of rooms in the rectangle is at most $cq(x+cq)/q^2 = cx/q + c^2$. Since the perimeter of a room is at most cq, we conclude that the distance on the path is bounded by $(cx/q + c^2)cq = c^2x + c^3q$. □

Let p be a point on vertical curve $\widehat{\mathcal{V}}_j$. If p lies on the original curve $\mathcal{V}_j$, then by the same arguments we used for horizontal lines, the deviation from the j-th grid line is less than $d/2$. Now, assume p is on a horizontal curve $\mathcal{H}_i$. The difference of p_{ij}-abscissa and q_{ij}-abscissa is at most cq since the deviation from the j-th vertical grid line is less than $cq/2$ for each point. By Lemma 6, the length of the shared path from p_{ij} to q_{ij} is less than $2c^3q$. Hence, in any point on the added path the deviation from the j-th vertical grid line is less than $cq/2 + c^3q < d/2$. Therefore, no two vertical curves intersect.

4.2 The dissection tree

The dissection tree is defined in almost the same way as in [2]. We define a *root* box B_0 with side length $2l$ that covers the bounding box $\mathcal{B}$ (with side length l) completely.

Denote by $(0, 0)$ the upper left corner of $\mathcal{B}$ and by (i, j) the crossing of the i-th horizontal grid line with the j-th vertical grid line. The midpoint of B_0 is (a, b) for some $a, b \in \{0, 1, \dots, 2^D - 1\}$. We partition the root box B_0 into four squares of equal size by drawing a horizontal and vertical line through (a, b). Next, each of the four squares is partitioned into four squares of equal size by adding two horizontal and two vertical lines. This process continues until the smallest squares have side length d.

We say that the root box is of *level* zero. The two lines through (a, b) and the four corresponding squares are of level 1, and so on until there are 2^D horizontal and 2^D vertical lines of level $D + 1$ that, together with the other lines, give $2^{2(D+1)}$ level-D+1 squares. We can picture this dissection as a *quadtree*. The root corresponds to B_0 and the four children are the level-1 squares. In general, the level of a square is the depth in the quadtree.

Now we replace the line segments that lie inside $\mathcal{B}$ by the curved lines and map each square (node) of the tree to its corresponding polygon, which we call a *node polygon* of the *curved dissection tree*.

4.3 Portal respecting trees

The intersection of a horizontal and a vertical dissection curve is either a single point or a simple path from p_{ij} to q_{ij}. To simplify the dynamic program we let, in the latter case, point p_{ij} be the *crossing point* of the two curves and separate the curves on the shared path. (See Figure 8.) This separation naturally defines four *sides* of each node polygon. Further, we separate each level-i line in 2^i *sections*. Each level-i section is the side of two level-i node polygons; one on each side.

Let us compute the maximum length of a level-i section. The side length of a level-i square is exactly $2l/2^i = d2^{D+1}/2^i$. The difference of p_{ij}-abscissa

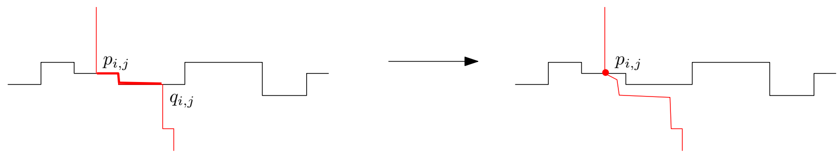

Fig. 8. The shared path of a horizontal and vertical curve is replaced by a point.

and $p_{i,j+1}$-abscissa is at most cq larger than this. Applying Lemma 6 with $x = d2^{D+1}/2^i + cq$ and using $d = 4c^3q$ gives a maximum length of a horizontal level-i section of

$$c^2\left(d\frac{2^{D+1}}{2^i} + cq\right) + c^3q = c^3q\left(4c^2\frac{2^{D+1}}{2^i} + 2\right) < 5c^5q\frac{2^{D+1}}{2^i}. \tag{1}$$

For a vertical section of level i, this length is longer by at most the number of crossings with horizontal curves times the maximum length of a crossing (shared path), i.e., $2^{D+1}/2^i$ times $2c^3q$. Adding this value $2c^3q2^{D+1}/2^i$ to the left of (1) does not violate the inequality. Thus, (1) ia an upper bound on the length of a vertical section as well.

On each level-i section we place m evenly distributes points. Hence, a level-i curve receives 2^im points. Each such point defines a *portal* on the boundary of both adjacent node polygons. Furthermore, we place one portal on each of the four corners of a node polygon. We choose $m = 40c^6D/\varepsilon = O(\log n)$. From (1) we see that the *interportal distance* on a level-i section is at most

$$5c^5q\frac{2^{D+1}}{2^i} \cdot \frac{\varepsilon}{40c^6D} = \frac{\varepsilon q 2^D}{cD2^{i+2}} \tag{2}$$

To make the dynamic programming work we assume that a segment of the tree that coincides with a dissection curve, can only connect rooms on one side of the curve. To serve rooms at the other side it has to cross the curve. (See Figure 9.) We call a feasible tree *portal respecting* if it crosses the curves only at portals.

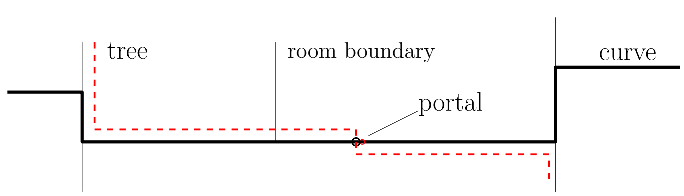

Fig. 9. To serve rooms on the other side of the curve, the tree has to cross at a portal.

4.4 The algorithm

We construct the bounding box $\mathcal{B}$ with the grid curves as in Section 4.1. Next, we choose $a, b \in \{0, 1, \dots, 2^D - 1\}$ independently at random and construct the dissection tree as in Section 4.2. We remove from the 4-ary tree all branches consisting of empty node polygons and derive a tree having at most $O(n)$ leaves and $O(n \log n)$ node polygons. Then we define the portals as in Section 4.3.

Starting from the leaves of the dissection tree, in a bottom-up way we update the dynamic programming table. An entry in the table is given by a node polygon and a *partition* $S_1, \dots, S_p$ of a subset of the portals on the boundary. The *value* of an entry is the minimum sum of the lengths of trees $T_1, \dots, T_p$ such that T_i connects the portals in S_i and each room inside the node polygon is connected to at least one tree. To facilitate the computation we do not require that the p trees are disjoint.

Let us compute the size of the table. The number of nodes in the tree is $O(n \log n)$. However, the number of partitions of the portals of a node polygon is not polynomially bounded. We say that a partition $S_1, \dots, S_p$ is *valid* if there are corresponding trees $T_1, \dots, T_p$ which are disjoint. The number of valid partition is polynomially bounded, as can be seen as follows. We may represent a valid partition by walking around the polygon and label a portal '0' if it is not used, label it '1' if it is used and is connected to a set S_i which we did not see before in this walk, and give it a label '2' if the portal is used and is connected to a set S_i that we have seen before in this walk. Conversely, for any labeling constructed in this way, the original partition is uniquely defined. Hence, the number of valid partition is bounded by 3^{4m+4}, which is polynomial in n. We conclude that the size of the table is polynomially bounded.

For any leaf of the dissection tree we can compute all entries in the table efficiently since the number of rooms in a leaf is $O(c^6)$, i.e. bounded by a constant. We may compute a minimum spanning tree for each subset S_i by guessing one corner point for each room and guess to which subset S_i it is connected.

For any non-leaf node polygons $\mathcal{P}$ of level i we can compute all values of its entries from the values of its four level-i+1 children as follows. Given a labeling $\sigma_{\mathcal{P}}$ of $\mathcal{P}$, we check if it corresponds to a valid partition. If so, we try all labelings σ of its four children, such that σ matches $\sigma_{\mathcal{P}}$ on the common sections. For each such labeling σ we check if it corresponds to a valid partitioning for each of the four children and check if they match with the partitioning defined by $\sigma_{\mathcal{P}}$. If so, we let $v(\sigma)$ be the sum of the four values, which we can read from the table. We try all labelings and take the one with minimum value $v(\sigma)$. Say this is σ', then the value of entry $(\mathcal{P}, \sigma_{\mathcal{P}})$ is $v(\sigma)$. For the root polygon B_0 we add the restriction that the partitions of the four children together correspond to a connected graph.

4.5 Performance guarantee

A simple packing argument shows that the value OPT of the optimal tree is $\Omega(qn)$.

Lemma 7 *$OPT \geq \frac{q}{2c}n$, for $n \geq c^2$.*

PROOF. If a room is connected to the optimal tree in point p, then the L_1-distance of p to any point of the room is at most $cq/2$. Now consider the region that consists of all points in the plane that are at L_1-distance at most $cq/2$ from some point in the optimal tree. The area of this region is at most $cqOPT + (cq)^2/2$. Note that this is tight if the tree is a straight line. Each room covers an area of at least q^2. Therefore, the number of rooms is $n \leq (cqOPT + (cq)^2/2)/q^2 = cOPT/q + c^2/2$, implying $OPT \geq nq/c - cq/2 = \frac{q}{c}(n - c^2/2)$. □

Lemma 8 *There is an optimal tree that crosses the grid curves at most $4n-4$ times.*

PROOF. Consider the planar graph that captures the combinatorial structure of the instance, i.e., the graph $G = (V, E)$ defined in the obvious way by the corner points of degree three and four. Note that four is the maximum degree. Each room is an (internal) face of G. The number of edges is $|E| \geq 3|V|/2$ and Euler's formula states that $|V| + n = |E| + 1$. This implies that $|V| + n \geq 3|V|/2 + 1$. Therefore, the number of vertices $|V|$ is at most $2n - 2$.

Of course, we may assume that all crossings of the tree with the grid curves (paths in the graph) are done at the corner points of degree three or four. In each vertex we cross at most one horizontal curve and at most one vertical curve. Hence, the total number of crossings is at most $4n - 4$. □

Let OPT be the length of the optimal tree and let $OPT_{a,b}$ be the length of the smallest portal respecting tree given the values $a, b \in \{0, 1, \ldots, 2^D - 1\}$.

Theorem 9

$$E[OPT_{a,b}] \leq (1 + \varepsilon)OPT.$$

PROOF. Consider an optimal tree and a random dissection. The tree follows the boundary of rooms and whenever this coincides with a dissection curve we specify which of the two sides of the curve it follows. Lemma 8 says that we can do this with at most $4n-4$ crossings. We make the tree portal respecting by moving each crossing of a curve to the nearest portal on that curve. From (2) we know that the interportal distance of a level-i section is at most $\frac{\varepsilon q 2^D}{cD2^{i+2}}$. The length of the detour is at most half this distance on each side. It is important to note that the detour may add many crossing in other polygon nodes, but it only does so at (corner) portals.

The probability that a grid line (curve) will be a level-i dissection line (curve) is $2^{i-2}/2^D$, for $2 \leq i \leq D+1$, and $1/2^D$ for $i=1$. Hence, this is at most $2^{i-1}/2^D$, for any i. Therefore, the expected length of a detour is at most

$$\sum_{i=1}^{D+1} \frac{2^{i-1}}{2^D} \cdot \frac{\varepsilon q 2^D}{cD2^{i+2}} = \frac{\varepsilon q}{8c}$$

By Lemma 7 and 8 and by linearity of expectations, the total length of all detours is at most

$$4(n-1)\frac{\varepsilon q}{8c} = \frac{\varepsilon q(n-1)}{2c} < \varepsilon OPT$$

This completes the proof. □

Summarizing Theorem 9 and the analysis of the algorithm, we derive the following result.

Theorem 10 *The described algorithm returns in time $n^{O(1/\epsilon)}$ a feasible tree of length at most $(1+\varepsilon)OPT$.*

To derandomize the algorithm we can simply go through all possible choices for a and b.

5 An approximation algorithm for the minimum corridor connection with rooms of varying sizes

Elbassioni et al. [12] give a simple constant factor approximation algorithm for the generalized Euclidean TSP, 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 the minimum corridor connection problem.

For any room R_i, $i \in \{1, \dots, n\}$, 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 whose 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 $p_i \in R_i$, $i \in \{1, \dots, n\}$, 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 the minimum corridor connection problem 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 11 *Algorithm* Greedy *gives an* $(n-1)$*-approximate solution for the minimum corridor connection problem.*

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 $i \in \{2, \dots, n\}$. 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 12 *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 intersects 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 the minimum corridor connection problem 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. Output the minimum of the obtained tree and the tree constructed by algorithm GREEDY.

Theorem 13 *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. Assume $n-1 \leq 16/\alpha-1$. Then, by Lemma 11, GREEDY guarantees a $(16/\alpha-1)$-approximation. Now, let $n \geq 16/\alpha$. 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 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 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 a minimum spanning tree 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 12 we know

$$t_i \geq \gamma\rho_i, \text{ for all } i \in F. \tag{3}$$

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 12 that $t_i \geq \gamma\rho_{h(i)}$. Combining the two bounds we get

$$t_i \geq \max\{\gamma\rho_{h(i)}, x_i - 2\rho_{h(i)}\} \geq \frac{\gamma}{\gamma+2}x_i, \text{ for all } i \in H. \tag{4}$$

Combining (3) and (4) we see that the minimum spanning tree 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 $(16/\alpha - 1)OPT$. □

6 Applications and extensions of the framework

To conclude the paper we briefly discuss applications of our techniques to the other generalized geometric problems. In fact, the randomized curved dissection presented in Section 4 can be seen as a general framework to construct PTASs for the generalized geometric versions of TSP, minimum Steiner tree, and minimum spanning tree problems restricted to disjoint fat regions of nearly the same size. Consider, for instance, generalized geometric TSP with disjoint regions where a disk of radius q can be inscribed in each region and the perimeter of each region is bounded from above by cq where c is a constant.

Notice that constructing the PTAS in Section 4, we did not use much of the specific structure of the minimum corridor connection problem. All our derivations are straightforwardly applicable to many generalized geometric problems, including TSP. Using literally the same algorithm analysis as in Arora [2] applied to the curved dissection introduced in Section 4, we straightforwardly construct $n(\log n)^{O(1/\varepsilon)}$ time approximation scheme for this version of the generalized geometric TSP. Compared to the recent approximation schemes of Dumitrescu and Mitchell [11] and Mitchell [27] for generalized geometric TSP, this is a significant improvement as their schemes run in $n^{O(1/\varepsilon)}$ time.

Moreover, Dumitrescu and Mitchell in [11] pointed out that in their approximation scheme for the generalized Euclidean TSP only some of the arguments for disjoint disks 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. [2,3,29], that Arora's algorithm for Euclidean TSP is applicable also in higher fixed dimensional spaces. Again, using literally the same argumentation as in [3] applied to the curved dissection, one easily derives that the generalized Euclidean TSP with disjoint fat regions of nearly the same size admits a PTAS even if regions are given in any space of fixed dimension (though, we have to be careful here properly defining the regions, fatness and size conditions). This resolves the open question from [11].

It is noticeable, however, that the recent approximation scheme for generalized geometric TSP by Mitchell [27] has a significant advantage compared to our algorithm. The algorithm of Mitchell can be applied to regions of varying size while we restrict the problem to the regions of nearly the same size.

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. Approximation and Online Algorithms (WAOA 2006), Lecture Notes in Computer Science 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. Canadian Conference on Computational Geometry (CCCG 2000), (Fredericton, New Brunswick, Canada, 2000).

[9] F. Dorn, Dynamic programming and fast matrix multiplication, in: Proc. European Symposium on Algorithms (ESA 2006), 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. European Symposium on Algorithms (ESA 2005), Lecture Notes in Computer Science, Vol. 3669 (Springer, Berlin, 2005) 95-106.

[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. International Colloquium on Automata, Languages and Programming (ICALP 2005), 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. European Workshop on Computational Geometry (EWCG 2005), (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) 72-103.

[21] Q. Gu and H. Tamaki, Optimal branch-decomposition of planar graphs in $O(n^3)$ time, in: Proc. International Colloquium on Automata, Languages and Programming (ICALP 2005), 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. ACM Symposium on Computational Geometry (SoCG 1995), (ACM, 1995) 360-369.

[27] J.S.B. Mitchell, A PTAS for TSP with neighborhoods among fat regions in the plane, in: Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), (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. ACM Symposium on Theory of Computing (STOC 1998), (ACM, 1998) 540-550.

[30] G. Reich and P. Widmayer, Beyond Steiner’s problem: a VLSI oriented generalization, in: Proc. Workshop on Graph-Theoretic Concepts in Computer Science (WG 1989), 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. European Symposium on Algorithms (ESA 2003), 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.