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

(Online Extended Abstract)**

Hans Bodlaender[1], Corinne Feremans[2], Alexander Grigoriev[2], Eelko Penninkx[1], René Sitters[3], and Thomas Wolle[4]

[1] Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands
{hansb,penninkx}@cs.uu.nl

[2] Department of Quantitative Economics, Maastricht University, P.O.Box 616, 6200 MD Maastricht, The Netherlands
{c.feremans,a.grigoriev}@ke.unimaas.nl

[3] Department of Algorithms and Complexity, Max-Planck-Institute for Computer Science, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
sitters@mpi-inf.mpg.de

[4] National ICT Australia Ltd***, Locked Bag 9013, Alexandria NSW 1435, Australia
thomas.wolle@nicta.com.au

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 an subexponential time exact algorithm. For the special case of k-outerplanar graphs the running time becomes $O(n^3)$. 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

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 Figure 1 which is borrowed from [6]. Let us refer to this problem as 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

* This work was supported by the Netherlands Organisation for Scientific Research NWO (project *Treewidth and Combinatorial Optimisation*)

** This article is also published in the WAOA 2006 conference proceedings, which appeared as a volume in Springer's Lecture Notes in Computer Science.

*** National ICT Australia is funded through the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

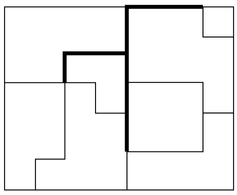

Fig. 1. Minimum length tree along the corridors to connect the rooms

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 described generalization of ETSP is known as the Generalized ETSP [12], or Group-ETSP, or ETSP with neighborhoods [5, 8], or the Geometric Covering Salesman Problem [1]. For short, we shall refer to this problem as 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, and 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 [10], Feremans, Labbé and Laporte [11], Mitchell [23], Reich and Widmayer [25].

3D-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 a multistory building such that each room has electricity, phone lines, etc.

Related Work. To the best of our knowledge, before this article nothing was known on complexity and approximability of the minimum corridor connection problem; see list of open problems from the 12th Canadian Conference on Computational Geometry CCCG 2000 [6].

For GTSP it is known that the problem cannot be efficiently approximated within $(2 - \varepsilon)$ unless P=NP, see [26]. Constant factor approximations for GTSP were developed for the special cases where neighborhoods are disjoint convex fat objects [5, 9], for disjoint unit discs [1], and for intersecting unit discs [8]. For the general GTSP, Mata and Mitchell [22] gave an $O(\log n)$-approximation. The closest related work to this article is the paper by Dumitrescu and Mitchell [8], where the authors have investigated the case of GTSP with regions given by pairwise disjoint unit disks, and they developed a polynomial time approximation scheme (PTAS) for this problem.

For the general GSTP, Helvig, Robins, and Zelikovsky [17] 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 [28], and Feremans, Labbé and Laporte [11].

Our results and paper organization. The remainder of this extended abstract is organized as follows. 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 [6].

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. (We follow [20] for the definition of "subexponential".)

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 [6]. More precisely, we consider the problem where a square of side length q can be inscribed in each room and the 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 *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 framework for PTASs presented in this paper is based on Arora's algorithm for ETSP [2]. 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 by Dumitrescu and Mitchell [8] for GTSP on disjoint unit discs 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 [8]. 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 [8].

Finally, in Section 5 we show how the algorithm for GTSP from Elbassioni et al. [9] 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 [6].

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 there 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 for each edge $\{u, v\} \in E$, $u \in W$ or $v \in W$? It is well known that CONNECTED VERTEX COVER for planar graphs with maximum degree four is NP-complete, see [14, 15]. Now we state the main result of this section. Because of space constraints, we will omit proofs in this extended abstract.

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

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 that are 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 the plane 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 of G 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 a plane graph of minimum width can be found in $O(n^3)$ time with the ratcatcher algorithm of Seymour and Thomas [27], see [7]. See also [16, 18, 19] for a necessary improvements to the original algorithm and implementation issues.

Dynamic programming with a branch decomposition. Instead of the MCC problem, we consider a small generalization, which we call FACE COVER TREE: given a plane graph $G = (V, E)$, with edge weights $w : E \to \mathbf{N}$, find a subtree T of G of minimum total weight such that each interior face has at least one vertex on T.

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 $O((3+\sqrt{5})^k k \cdot n)$ time.*

To obtain this result, we use techniques from Dorn et al. [7]. The basic idea is that we build a table for each edge in the branch decomposition. Assuming a root for T, we associate to each edge $e \in E(T)$, the subgraph formed by the edges of G associated with the leaves in T that are below e in the tree. This is one of the subgraphs $G_{e,1}$ or $G_{e,2}$; w.l.o.g., we will assume that this is always $G_{e,1}$. A forest T' that is a subgraph of $G_{e,1}$ can be extended to a solution of the FACE COVER TREE if each face of $G_{e,1}$ that does not intersect the noose is touched by T' and each subtree of T' contains at least one vertex in the middle set of e. We can characterize such forests of the second type by the set of vertices in the middle set that belong to the forest, an equivalence relation on these vertices which are connected by the forest, the information which faces that intersect the noose are touched by the forest, and (of course), the total length of all edges in the forest. Having this information is also sufficient to see how the forest can be extended.

Thus, 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 $G_{e,1}$ 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.

Using counting techniques from [7], we can show that for a middle set of size ℓ, such a table contains at most $(3+\sqrt{5})^\ell$ entries. (For instance, let R form a non-crossing partition on S. We only need to distinguish whether faces are touched whose two incident middle set vertices do not belong to S.)

It is trivial to compute the table for an edge in T incident to a leaf. 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 $O((14+6\sqrt{5})^k \cdot k)$ 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 $O((14+6\sqrt{5})^k \cdot k \cdot n)$ time. Note that $14 + 6\sqrt{5} = 2^{4.7770}$.

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 planar 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 [4] 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 well 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 [4, 21].) The best known bound to our knowledge is the following.

Theorem 3 (Fomin and Thilikos [13]). *A planar graph with n vertices has branchwidth at most $\sqrt{4.5 \cdot n}$.*

Thus we have the following consequences, where we expect that the actual running times of these algorithms will be better in practice.

Corollary 1. *The* FACE COVER TREE*, and hence also the MCC problem can be solved in $O(n^3 + 2^{9.5539k})$ time on k-outerplanar graphs, and in $O^*(2^{10.1335\sqrt{n}})$ time on planar graphs.*

4 A PTAS for MCC with geographic clustering

To construct a polynomial time approximation scheme for MCC, we modify Arora's algorithm for ETSP [2, 3]. 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 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, 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 side length at most cqn. Let the size of the bounding box be $L \in [cqn, 2cqn]$ such that L/cq is a power of 2. 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 Figures 2 and 3.) 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 $cq/2$.

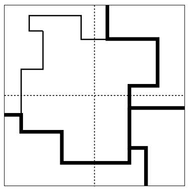

Fig. 2. Curved dissection

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 Figure 2, 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.

4.2 Portals and portal respecting trees

Let a level i dissection curve have $2^i m$ special points equally spaced on that curve. 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 *portals* and to m as *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 Figure 3.) 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 the *side* of the 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.

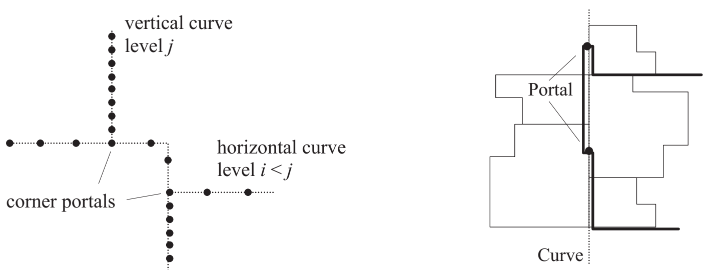

Fig. 3. Portals and a feasible portal respecting tree.

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 [3]. By removing from the 4-ary tree all branches consisting of empty node polygons, we obtain 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 at the leaves of the dissection tree in a bottom-up way, we update the dynamic programming table. For each node polygon, for 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 4 (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.

The proof is omitted and is slightly more complicated than in Arora [3]. Taking $m = O(\frac{\log n}{\varepsilon})$ and $k = O(\frac{1}{\varepsilon})$ we derive the following result.

Theorem 5. *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 [24]. In fact, a straightforward adaption of a more careful analysis presented in [24] can also significantly improve the running times presented in this extended abstract. For two dimensional space this would even imply an $O(n \log n)$ time and $O(n)$ space PTAS for MCC and other geometric problems with geographic clustering.

4.5 Extensions of the PTAS

As in Arora [2, 3] we did not use much of the specifics 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 only differ by 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 the 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 minimum size over all rooms of the maximum inscribed square or ball and c is a fixed constant).

Dumitrescu and Mitchell in [8] 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., [2, 3, 24], that Arora's algorithm for ETSP is applicable also in higher fixed dimensional spaces. Using literally the same argumentation as in [2] and our construction for MCC with geographic clustering, one can derive the following theorem.

Theorem 6. *If the corner points of the rooms are in $\mathcal{R}^d$, the MCC with geographic clustering admits a 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 of $n^{d+1}(\log n)^{(O(\sqrt{d}/\varepsilon))^{d-1}}$.*

The same holds for GTSP, GSTP and GMST. This resolves the open question from Dumitrescu and Mitchell [8].

5 An approximation algorithm for MCC with rooms of varying sizes

Elbassioni et al. [9] 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 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. Note 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 1. *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 2. *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 $2\rho_{\min}$ move along the corridor P. The total area A covered by the moving square is at most $(2\rho_{\min})^2 + 2\rho_{\min} \cdot d(P)$. Assume 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 $d(P) \geq \rho_{\min}(k\alpha/2 - 2)$, which 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 7. *Algorithm* Connect *gives a* $(16/\alpha - 1)$*-approximate solution for the minimum corridor connection problem in which the fatness of every room is at least* α*.*

Proof. If $n - 1 \leq 16/\alpha - 1$ then Greedy guarantees the approximation ratio for smaller values of n. 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* 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 into 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. Note 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 2 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 2 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 $k = 8/\alpha$. Furthermore, it is strictly smaller for any value in between. Hence, there is an integer $k \in [8/\alpha - 1, 8/\alpha]$ such that $k(k-1)/(k-4/\alpha) \leq 16/\alpha - 2$. Note 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 $(16/\alpha - 1)OPT$. □

Acknowledgments

We thank Joe Mitchell, Sándor Fekete, and Mark de Berg for useful discussions on the GTSP with geographic clustering, and also thanks to Sergio Cabello.

References

1. E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55:197–218, 1994.
2. S. Arora. Nearly linear time approximation schemes for Euclidean TSP and other geometric problems. *Journal of the ACM*, 45:1–30, 1998.
3. S. Arora. Approximation schemes for NP-hard geometric optimization problems: A survey. *Mathematical Programming*, 97:43–69, 2003.
4. H. L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209:1–45, 1998.
5. M. de Berg, J. Gudmundsson, M. Katz, C. Levcopoulos, M. Overmars, and A. van der Stappen. TSP with neighborhoods of varying size. *Journal of Algorithms*, 57:22–36, 2005.
6. E. D. Demaine and J. O'Rourke. Open problems from CCCG 2000. `http://theory.lcs.mit.edu/~edemaine/papers/CCCG2000Open/`, 2000.
7. F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *Algorithms - ESA 2005, 13th Annual European Symposium*, pages 95–106. LNCS 3669, Springer, October 2005.
8. A. Dumitrescu and J. S. B. Mitchell. Approximation algorithms for TSP with neighborhoods in the plane. *Journal of Algorithms*, 48:135–159, 2003.
9. K. Elbassioni, A. V. Fishkin, N. H. Mustafa, and R. Sitters. Approximation algorithms for Euclidean group TSP. In *Automata, Languages and Programming: 32nd International Colloquium, ICALP*, pages 1115–1126. LNCS 3580, Springer, July 2005.
10. C. Feremans. *Generalized Spanning Trees and Extensions*. PhD thesis, Université Libre de Bruxelles, Brussels, 2001.
11. C. Feremans, M. Labbé, and G. Laporte. Generalized network design problems. *European Journal of Operational Research*, 148:1–13, 2003.
12. M. Fischetti, J. J. Salazar, and P. Toth. A branch-and-cut algorithm for the symmetric generalized traveling salesman problem. *Operations Research*, 45:378–394, 1997.
13. F. V. Fomin and D. M. Thilikos. New upper bounds on the decomposability of planar graphs. *Journal of Graph Theory*, 51:53–81, 2006.
14. M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal of Applied Mathematics*, 32:826–834, 1977.
15. M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.
16. Q. Gu and H. Tamaki. Optimal branch-decomposition of planar graphs in $O(n^3)$ time. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005*, pages 373–384. LNCS 3580, Springer, July 2005.
17. C. S. Helvig, G. Robins, and A. Zelikovsky. An improved approximation scheme for the Group Steiner Problem. *Networks*, 37:8–20, 2001.
18. I. V. Hicks. Planar branch decompositions I: The ratcatcher. *INFORMS Journal on Computing*, 17:402–412, 2005.
19. I. V. Hicks. Planar branch decompositions II: The cycle method. *INFORMS Journal on Computing*, 17:413–421, 2005.
20. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
21. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math*, 36:177–189, 1979.
22. C. S. Mata and J. S. B. Mitchell. Approximation algorithms for geometric tour and network design problems. In *ACM SoCG 1995: Vancouver, BC, Canada*, pages 360–369. ACM, 1995.
23. J. S. B. Mitchell. *Handbook of Computational Geometry*, chapter Geometric shortest paths and network optimization, pages 633–701. Elsevier, North-Holland, Amsterdam, 2000.
24. S. Rao and W. D. Smith. Approximating geometric graphs via "spanners" and "banyans". In *ACM STOC 1998: Dallas, Texas, USA*, pages 540–550. ACM, 1998.

25. G. Reich and P. Widmayer. Beyond Steiner's problem: a VLSI oriented generalization. In *Graph-Theoretic Concepts in Computer Science, 15th International Workshop, WG '89*, pages 196–210. LNCS 411, Springer, 1990.
26. S. Safra and O. Schwartz. On the complexity of approximating TSP with neighborhoods and related problems. In *Algorithms - ESA 2003, 11th Annual European Symposium*, pages 446–458. LNCS 2832, Springer, September 2003.
27. P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14:217–241, 1994.
28. M. Zachariasen and A. Rohe. Rectilinear group Steiner trees and applications in VLSI design. *Mathematical Programming*, 94:407–433, 2003.