Approximation Schemes for Minimum-Cost k-Connectivity Problems in Geometric Graphs *

Artur Czumaj
Department of Computer Science
New Jersey Institute of Technology
Newark, NJ 07102-1982, USA
E-mail: czumaj@cis.njit.edu

Andrzej Lingas
Department of Computer Science
Lund University
Box 118, S-22100 Lund, Sweden
E-mail: Andrzej.Lingas@cs.lth.se

1 Introduction

1.1 Multi-connectivity problems

We survey the recent progress in the design of approximation schemes for geometric variants of the following classical optimization problem: for a given undirected weighted graph, find its minimum-cost subgraph that satisfies a priori given multi-connectivity requirements. We present the approximation schemes for various geometric minimum-cost k-connectivity problems and for geometric survivability problems, giving a detailed tutorial of the novel techniques developed for these algorithms. We also shortly discuss extensions to include planar graphs.

A classical multi-connectivity graph problem is as follows: for a given undirected weighted graph, find its minimum-cost subgraph that satisfies a priori given connectivity requirements.

Multi-connectivity graph problems are central in algorithmic graph theory and have numerous applications in computer science and operation research, see, e.g., [1, 22, 18, 34, 35]. They also play very important role in the design of networks that arise in practical situations, see, e.g., [1, 22, 30]. Typical application areas include telecommunication, computer and road networks. Low degree connectivity problems for geometrical graphs in the plane can often closely *approximate* such practical connectivity problems (see, e.g., the discussion in [22, 32, 35]). For instance, they can be used to model the design of low-cost telephone networks that can "survive" some types of edge and node failure. In such a model, the cost of the edge corresponds to the cost of lying a fiber-optic cable between the endpoints of the edge plus the planned cost of the service of the cable. Furthermore, the minimum connectivity requirement for a pair

*Research supported in part by NSF ITR grant CCR-0313219 and by VR grant 621-2002-4049.

of vertices corresponds to the minimum number of edge and/or node failures that must occur in the network before the pair is completely disconnected. In practice, the latter value tends to be quite low, usually no more that 2, since failures are assumed to be isolated accidents, such as fires at nodes [32, 35]. Note that the cost of lying a fiber-optic cable between two points is roughly proportional to the length of the link (see, e.g., [32]).

In this work we survey approximation results for these problems restricted to geometric graphs and planar graphs.

The most classical problem we study is the *(Euclidean) minimum-cost* k*-vertex connected spanning subgraph (*k*-VCSS) problem*. We are given a set S of n points in the Euclidean space $\mathbb{R}^d$ and the aim is to find a minimum-cost k-vertex connected Euclidean graph spanning points in S (i.e., a subgraph of the complete Euclidean graph on S).

Throughout the paper we shall assume that the cost of the graph is equal to the sum of the costs of the edges of the graph. Furthermore, in the geometric case, the cost of an edge connecting a pair of points $x, y \in \mathbb{R}^d$ is equal to the Euclidean distance between points x and y, that is, $\sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$, where $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$. More generally, the distance could be defined using other norms, such as ℓ_p norms for any $p > 1$; all results discussed in this survey can be extended from the Euclidean case to other ℓ_p norms.

By substituting the requirement of k-edge connectivity for that of k-vertex connectivity, we obtain the corresponding *(Euclidean) minimum-cost* k*-edge connected spanning subgraph (*k*-ECSS) problem*. We term the generalization of the latter problem which allows for parallel edges in the output graph spanning S as the *(Euclidean) minimum-cost* k*-edge connected spanning sub-multigraph (*k*-ECSSM) problem*.

The concept of minimum-cost k-connectivity naturally extends to include that of *Euclidean Steiner* k*-connectivity* by allowing the use of additional vertices, called *Steiner points*. For a given set S of points in $\mathbb{R}^d$, we say that a geometric graph G is a *Steiner* k*-VCSS (or, Steiner* k*-ECSS) for* S if the vertex set of G is a *superset* of S and for every pair of points from S there are k internally vertex-disjoint (edge-disjoint, respectively) paths connecting them in G. The problem of *(Euclidean) minimum-cost Steiner* k*-vertex- (or,* k*-edge-) connectivity* is to find a minimum-cost Steiner k-VCSS (or, Steiner k-ECSS) for S. For $k = 1$, it is simply the *Steiner minimal tree* (SMT) problem, which has been very extensively studied in the literature (see, e.g., [14, 26] and Chapter R-30).

In a more general formulation of multi-connectivity graph problems, non-uniform connectivity constraints have to be satisfied. The *survivable network design problem* is defined as follows: for a given weighted undirected graph $G = (V, E)$ and a connectivity requirement function $r : V \times V \to \mathbb{N}$, find a minimum-cost subgraph of G such that for any pair of vertices $x, y \in V$ the subgraph has $r_{x,y}$ internally vertex-disjoint (or edge-disjoint, respectively) paths between x and y. Also in that case, the output may be allowed to be a multigraph [35]. The survivable network design problem arises in many aforementioned applications, e.g., in telecommunication, communication network design, VLSI design, etc.

In many applications of this problem, often regarded as the most interesting ones [16, 22], the connectivity requirement function is specified with the help of a one-argument function which assigns to each vertex v its connectivity type $r_v \in \mathbb{N}$. Then, for any pair of vertices $v, u \in V$, the connectivity requirement $r_{u,v}$ is simply given as $\min\{r_u, r_v\}$ [20, 21, 22, 22, 32, 35]. Following the literature, we assume this standard simplification of the connectivity requirements function in this paper. Note that, in particular, this includes the *Steiner tree problem* (see, e.g., [2]), in which $r_v \in \{0, 1\}$ for any vertex $v \in V$. It also includes the most widely applied variant of the survivability problem in which $r_v \in \{0, 1, 2\}$ for any vertex $v \in V$ (see, e.g., [22, 32, 35]).

Since all the aforementioned k-connectivity problems are known to be $\mathcal{NP}$-hard when restricted to even two-dimensions for $k \geq 2$ [17], we focus on efficient constructions of good approximations. We aim at developing a *polynomial-time approximation scheme*, a *PTAS*. This is a family of algorithms $\{\mathcal{A}_\varepsilon\}$ such that, for each fixed $\varepsilon > 0$, $\mathcal{A}_\varepsilon$ runs in time polynomial in the size of the input and produces a $(1+\varepsilon)$-approximation (see Chapter R-10 and [24]).

1.2 History of the multi-connectivity problems

For a very extensive presentation of results concerning problems of finding minimum-cost k-vertex- and k-edge-connected spanning subgraphs, non-uniform connectivity, connectivity augmentation problems, and geometric problems, we refer the reader to [1, 28, 34] and to various chapters of [24], especially to [18, 27]. Here we discuss mostly the work related to geometric graphs.

All the multi-connectivity problems discussed in this survey are known to be $\mathcal{NP}$-hard not only for general graphs, but also for several non-trivial classes of graphs. For general graphs, the multi-connectivity problems are even known to be APX-hard, that is, they do not have a PTAS unless $\mathcal{P} = \mathcal{NP}$ (see [9]). Despite the practical relevance of the multi-connectivity problems for geometrical graphs and the vast amount of practical heuristic results reported (see, e.g., [21, 22, 32, 35]), very little theoretical research has been done towards developing efficient approximation algorithms for these problems until a few years ago. This contrasts with the very rich and successful theoretical investigations of the corresponding problems in general metric spaces and for general weighted graphs. And so, until 1998, even for the simplest and most fundamental multi-connectivity problem, that of finding a minimum-cost biconnected graph spanning a given set of points in the Euclidean plane, obtaining approximations achieving better than a $\frac{3}{2}$ ratio had been elusive (the ratio $\frac{3}{2}$ is the best polynomial-time approximation ratio known for general graphs whose weights satisfy the triangle inequality [15]; for other results, see e.g., [6, 27]).

For many years the algorithmic community has believed that TSP and the multi-connectivity problems discussed in this survey were also APX-hard for geometric and planar graphs, and hence they do not have a PTAS unless $\mathcal{P} = \mathcal{NP}$. However, the situation changed dramatically with the seminal works of Arora [2] and Mitchell [31], who showed that the TSP problem in geometric graph has a PTAS; soon after, a PTAS for TSP in planar graphs has been also developed by Arora et al. [4]. These results gave a hope that also the multi-connectivity problems in geometric and planar graphs can have a PTAS. This has been proven affirmatively in a series of papers by Czumaj and Lingas [8, 9, 10] in the case of geometric graphs. The case of planar graphs seems to be more challenging, and as for today, we know only a partial solution for 2-connectivity problems, where a quasi-polynomial time approximation scheme (running-time of the form $n^{\widetilde{\mathcal{O}}(\log n/\varepsilon)}$) has been recently developed [5, 7].

1.3 Overview of the results

In this survey, we overview the recent polynomial-time approximation schemes for multi-connectivity problems in geometric graphs, as developed in a series of papers by Czumaj, Lingas, and Zhao in [8, 9, 10, 13] (see also [11, 12] for full versions of the papers). Besides presenting the specific results, we give a detailed tutorial of techniques developed during the design of polynomial-time approximation schemes for various k-connectivity problems in geometric graphs; we also emphasize the difference between these algorithms and the recent PTASs for TSP and related problems. In addition, we discuss lower bounds on approximability of these problems in higher dimensions [9] and extensions to include the survivability problems [13] and planar graphs [5, 7].

2 Preliminaries

In this section we introduce basic technical definitions and notions used in this survey. For simplicity of presentation we shall assume that the quality of approximation ε satisfies $n^{-1/4} < \varepsilon \le 0.1$. Furthermore, we shall aim at achieving an approximation of $(1 + \mathcal{O}(\varepsilon))$ rather than $(1 + \varepsilon)$. Both these assumptions can be easily relaxed.

All algorithms for geometric graphs that we discuss in this survey are randomized. Even though the algorithms can be derandomized, and the final results are stated in deterministic versions as well, the randomized versions of the algorithms are more natural to present and are simpler.

For a given undirected graph G, a *traveling salesman tour* (TST) is any Hamiltonian cycle in G; a *traveling salesman path* is any Hamiltonian path in G. For a given set S of points, a *traveling salesman tour* (TST) is any traveling salesman tour for the complete graph on S. For a given graph G with cost on the edges, or for a set of points S in a metric space, the *traveling salesman problem (TSP)* is to find a TST in G or for S, respectively, that has a minimum total cost of the edges. For simplicity of our presentation, we define a TST for a set of *two* points to be the edge connecting these points.

We use term L^d-*cube* with $L \in \mathbb{R}$ to denote any axis-parallel d-dimensional cube in $\mathbb{R}^d$ of side-length L in all d dimensions. A *bounding box* of the input multiset of points in $\mathbb{R}^d$ is any L^d-cube in $\mathbb{R}^d$ enclosing these points.

The perturbation. In our algorithms for multi-connectivity problems, we first *perturb* the input instance so that each node lies on the unit grid and every inter-node distance is at least 8. We begin with rescaling the input so that the smallest bounding box L^d has $L = \mathcal{O}(n^3)$ being a power of two. Next, we move every point to the nearest point

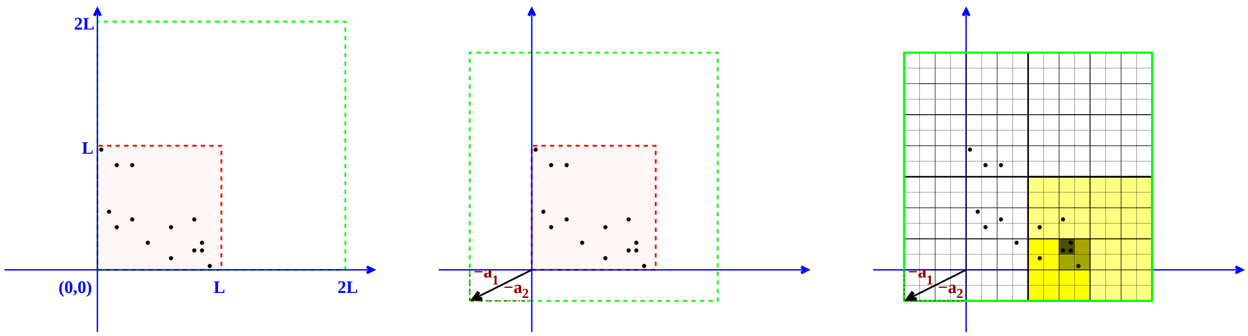

Figure 1.1: Shifted dissection of a set of points in the bounding box $[0, L]^d$ in $\mathbb{R}^2$ (with $\mathbf{a} = (a_1, a_2)$).

on the unit grid whose all coordinates are multipliers of 8 (what may merge some points). Then, it is easy to see that the perturbation ensures that any k-VCSS for the original input instance is now mapped into a k-VCSS whose cost (after rescaling) differs by at most an ε fraction. Therefore, if we can find a $(1+\mathcal{O}(\varepsilon))$-approximation for the k-VCSS problem for the perturbed instance, then we can directly obtain an $(1+\mathcal{O}(\varepsilon))$-approximation for the k-VCSS problem for the original instance. Because of that, from now on we assume that all input points have integer coordinates, lie in the cube $[0, L]^d$ with $L = \mathcal{O}(n^3)$ being a power of two, and the distance between any two points is either 0 or is at least 8. (We chose $L = \mathcal{O}(n^3)$ for convenience only; a much smaller L would be enough.)

The dissection. The concept of space partitioning via *dissections* (quadtrees) and *shifted dissections* plays the key role in all our algorithms. Following [2], we define the geometric partitioning of a bounding box as follows. A *(2^d-ary) dissection* of the bounding box L^d of a set of points in $\mathbb{R}^d$ is its recursive partitioning into smaller sub-cubes, called *regions*. Each *region* U^d of volume larger than 1 is recursively partitioned into 2^d regions $(U/2)^d$. A 2^d*-tree* with respect to the (2^d-ary) dissection is a tree whose root corresponds to the bounding box, and whose other non-leaf nodes correspond to the regions containing at least two points from the input multiset. For a non-leaf node v of the tree, the nodes corresponding to the 2^d regions partitioning the region corresponding to v are the children of v in the tree. Note that the dissection has $\Theta(L^d)$ regions and its recursion depth is logarithmic in L. Furthermore, if L is a power of two, the boundaries of all regions in the dissection have integer coordinates, and thus they are in the unit grid.

For any d-vector $\mathbf{a} = (a_1, \dots, a_d)$, where all a_i are $0 \le a_i \le L$, the **a**-*shifted dissection* [2] of a set X of points in the bounding box $[0, L]^d$ in $\mathbb{R}^d$ is the result of shifting all the regions in the dissection of X in the bounding box $[0, 2L]^d$ by the vector $(-\mathbf{a})$. The **a**-*shifted* 2^d*-ary tree* with respect to the **a**-shifted dissection is defined analogously. A *random shifted dissection* of a set of points X in a cube L^d in $\mathbb{R}^d$ is an **a**-shifted dissection of X with $\mathbf{a} = (a_1, \dots, a_d)$ and the elements $a_1, \dots, a_d$ chosen independently and uniformly at random from $\{0, 1, \dots, L\}$.

3 First approach: polynomial-time "pseudo-approximation" schemes

After the development of polynomial-time approximation schemes for the TSP problem due to Arora [2] and Mitchell [31], it seemed to be almost a straightforward task to extend their schemes to obtain a PTAS for multi-connectivity problems, or at least for the most basic 2-VCSS and 2-ECSS problems. However, it turned out that the schemes, which work very well for TSP and for some number of related problems, including Minimum Steiner Tree, Min-Cost Perfect Matching, k-TSP, and k-MST, could not be extended in a simple way. The reason was that in all these approximation schemes, the key step was to find a low cost solution which uses Steiner points. While (by the triangle inequality) a Steiner point can be removed from a TST without any increase of its cost, such a transformation is impossible for k-VCSS and k-ECSS problems: e.g., a minimum-cost 2-VCSS for a point set S in $\mathbb{R}^2$ can have cost as much as $\frac{\sqrt{3}}{2}$ times larger than a Steiner 2-VCSS for S [25].

Despite this difficulty, Czumaj and Lingas [8] showed that one can apply the approach developed by Arora [2] to design a "pseudo-approximation scheme": an algorithm that finds a Steiner k-VCSS for a point set S whose cost is at most $(1+\varepsilon)$ times larger than the cost of a minimum-cost S-VCSS for S. In other words, the algorithm finds a solution with Steiner points that has cost not much larger than an optimal solution that uses no Steiner points. Even though this is only a pseudo-approximation scheme and not a PTAS, in this section we shall present this algorithm in details, because the underlying ideas of this approach are used later in all other algorithms we discuss in this survey.

On a very high level, the approach of Arora [2] (see also [31] and [3, 33]) is a clever combination of the divide-and-conquer method with the dynamic programming approach, and as such, it follows a design of many classical PTASs. For the multi-connectivity problems, similarly as Arora, we hierarchically partition the cube containing the input points (via random shifted dissection) into regions, and then prove the key technical result, that there is an approximate solution to the problem which can cross the boundaries of each region only in pre-specified points a bounded number of times (*Structure Theorem*). The Structure Theorem states that for any problem instance there is a $(1+\mathcal{O}(\varepsilon))$-approximation that satisfies some basic local property: it is m-*portal respecting* and r-*light*, see the definitions below. The Structure Theorem is proven by taking an optimal solution to the problem and applying a sequence of transformations that increases the cost of the resulting graph and at the same time makes it m-portal respecting and r-light. Once the Structure Theorem is proved, a dynamic programming procedure finds in a polynomial-time an almost optimal solution that satisfies the basic local property. The dynamic programming procedure combines optimal partial solutions within regions into an optimal global solution under the crossing restrictions. In order to combine solutions efficiently, we derive a k-connectivity characteristic of a spanning subgraph within a region solely in terms of the set of pre-specified points on the region boundary included in its vertex set. In our crucial theorem, we show that the connectivity characteristic of a union of two adjacent subgraphs can be computed from the connectivity characteristics of the subgraphs. This allows us to set up a dynamic programming procedure computing a $(1+\mathcal{O}(\varepsilon))$-approximation

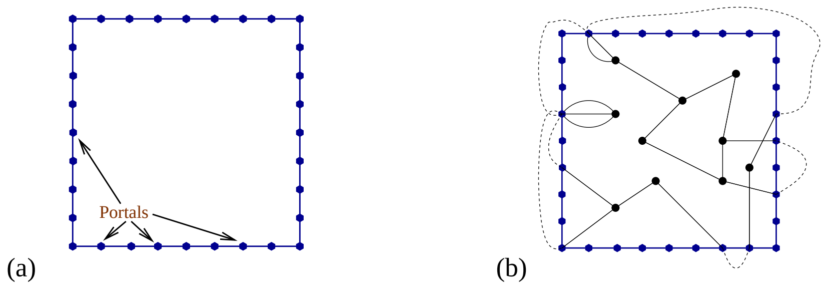

Figure 1.2: Portals and a portal-respecting graph.

of minimum-cost Euclidean graph which is k-vertex or k-edge connected and obeys the crossing restrictions.

Below we discuss this approach in more details; we focus only on the k-VSCC problem and note that the extension to the k-ECSS problem is straightforward.

3.1 Special forms of geometric graphs

3.1.1 m-portal-respecting graphs

For every integer m, an m-*regular* set of *portals* in a $(d-1)$-dimensional region facet U^{d-1} is an orthogonal lattice of m points in the facet where the spacing between the portals is $(U+1) \cdot m^{-1/(d-1)}$. A graph is m-*portal-respecting* (with respect to a shifted dissection) if whenever it crosses a facet in the dissection, it does so at a portal. Observe that this restriction forces us to assume that an m-portal-respecting graph may have to *bend* some of its edges, that is, an edge may deviate from being a straight-line connecting its endpoints and be rather a *straight-line path* between the endpoints. If we are allowed to bend the ends, then for any graph in a dissection it is easy to make it m-portal-respecting by moving every crossing of every facet to its nearest portal. Arora [2] proved the following result that transforms a graph into an m-portal-respecting one at a small cost increase and without changing the connectivity.

Lemma 1.1 *Let* G *be a geometric graph in* $\mathbb{R}^d$ *for a set of (perturbed) points contained in a bounding box* L^d*. Pick a random shifted dissection of* L^d*. Then, one can transform* G *into an* m*-portal-respecting graph by moving each crossing of each facet to its nearest portal so that the expected increase of the cost of the resulting graph is at most* $\mathcal{O}(d \log L\, m^{-1/(d-1)}) \cdot \mathrm{cost}(G)$.

Proof: Pick any edge (v, u). By the definition of the dissection, edge (v, u) crosses the facets in the dissection at most $\mathcal{O}(\sqrt{d}) \cdot \mathfrak{c}(v, u)$ times, where $\mathfrak{c}(v, u)$ is the cost of the edge (v, u).[1] To make this edge m-portal-respecting, we move each crossing of a facet to the nearest portal, which involves bending the edge that might increase its length. If the facet has side-length $L/2^i$, then this increases the distance by at most $\mathcal{O}(\sqrt{d}\, L/2^i)\, m^{-1/(d-1)}$, since the inter-

[1]The number of crossings of the facets in the dissection is upper bounded by a constant times the ℓ_1 distance between v and u, and this is upper bounded by $\mathcal{O}(\sqrt{d})$ times the ℓ_2 distance between v and u, that is, $\mathcal{O}(\sqrt{d}) \cdot \mathfrak{c}(v, u)$.

portal distance is $\mathcal{O}(L/2^i)\, m^{-1/(d-1)}$. Because we have chosen the dissection at random, the probability that a given facet has side-length $L/2^i$ is $\mathcal{O}(2^i/L)$. Hence, the expected increase of the cost of a given edge (v, u) is

$$\sum_{i=0}^{\log L} \mathcal{O}(\sqrt{d}\, 2^i/L) \cdot \mathcal{O}(L/2^i)\, m^{-1/(d-1)} \quad = \quad \mathcal{O}(\sqrt{d}\, \log L \cdot m^{-1/(d-1)}) \ .$$

The same arguments can be applied to all of at most $\mathcal{O}(\sqrt{d}) \cdot \mathfrak{c}(v, u)$ dissection crossings by any edge (v, u). Therefore, the expected increase of the cost of the entire graph is at most

$$\sum_{(v,u)} \Big(\mathcal{O}(\sqrt{d}) \cdot \mathfrak{c}(v, u)\Big) \cdot \mathcal{O}(\sqrt{d}\, \log L \cdot m^{-1/(d-1)}) \quad = \quad \mathcal{O}(d\, \log L \cdot m^{-1/(d-1)}) \cdot \mathrm{cost}(G) \ . \qquad \square$$

Note that in our applications we require this error term to be at most an $\mathcal{O}(\varepsilon)$ factor of the cost of the optimal solution, and therefore we set $m = (\mathcal{O}(d\, \log L/\varepsilon))^{d-1}$. Using the transformation from Lemma 1.1, from now on, we assume that we consider a geometric graph that is m-portal-respecting with $m = (\mathcal{O}(d\, \log L/\varepsilon))^{d-1}$.

Special forms of geometric graphs: r-light graphs. We say a geometric graph is r-*light* (with respect to a shifted dissection) if for each region in the dissection there are at most r edges *crossing any of its facets*.

3.2 Dynamic programming and finding an optimal m-portal-respecting r-light solution

In our presentation, we begin from the end and discuss first the goal of our analysis. In the following section we show that for any set S of n (perturbed) points in $\mathbb{R}^d$ that are contained in a bounding box L^d, if we choose a random shifted dissection of L^d, then with a good probability there is an m-portal-respecting r-light (for the dissection chosen) Steiner k-VCSS for S whose cost is at most $(1 + \mathcal{O}(\varepsilon))$ times the cost of the optimal k-VCSS for S, for appropriated values of m and r. How can we use this existential result? The key observation is that if we restrict ourself to m-portal-respecting r-light graphs then we can use dynamic programming to actually find an almost optimal Steiner k-VCSS efficiently! In what follows, we briefly discuss main ideas of this result; see [8] for more details.

The key idea is that the subproblem (finding an optimal Steiner k-VCSS) inside a region in the dissection can be solved independently of the subproblems in other regions provided that we know which portals are used by the edges of the graph and the structure of the external k-connectivity properties outside that region. External k-connectivity properties outside a region are defined in terms of the portals used: if a portal is used by the graph outside the current region, we want to know which connections with other portals it supports. The concept of *connectivity characteristic* developed in [8] aims at maintaining this structural properties of graphs contained in any region.

Let us define an *internal interface* of a region Q to be any multiset of at most m portals, such that for every facet of Q, the total multiplicity of all portals (in the multiset) is upper bounded by r. Since our goal is to find an

m-portal-respecting and r-light Steiner k-VCSS with low cost, we do not know its structure in advance (except that it is m-portal-respecting and r-light) and hence in our algorithm we have to consider all possible internal interfaces. Note that for m-portal-respecting and r-light graphs the number of internal interfaces of a region is at most $m^{\mathcal{O}(d\,r)}$.

Next, for any region Q and any given internal interface of Q, we define a *connectivity characteristic* to be a description of routing properties within the region and requirements on the routing properties in the complementary graph from the point of view of portals needed to preserve k-connectivity. Let P be the multiset of points in the portals used in the internal interface. The connectivity characteristic consists of three parts corresponding to different aspects of k-vertex connectivity requirements for the graph:

- requirements for *internal connectivity*: what configurations of external disjoint paths ought to be outside the region to make any pair of vertices within region k-vertex connected; since for any pair of points in Q, all sets of disjoint paths leaving Q must traverse through the portals, each such a set of vertex-disjoint paths can be encoded by a matching in the complete graph on P
- requirements for *internal/external connectivity*: what configurations of disjoint paths ought to be inside and outside Q to ensure that any vertex inside region Q has k vertex-disjoint paths to any vertex outside the region; this can be encoded by a set of pairs consisting of a matching in the complete graph on P and a subset of portals (that is used to encode the parts of the paths from a vertex inside Q to the first portals, before they leave Q);
- requirements for *external connectivity*: what configurations of internal disjoint paths ought to be inside Q to ensure that any pair of vertices outside Q are connected by k vertex-disjoint paths; this can be encoded by matchings in the complete graph on P.

One can show that for a given region and its internal interface, there are at most $2^{(d\,r)^{\mathcal{O}(d\,r)}}$ connectivity characteristics.

The goal of the dynamic programming procedure is to determine for each region Q, for each possible internal interface of Q, and for each possible connectivity characteristic of Q, an (almost) optimal m-portal-respecting r-light graph within the region using given internal interface and having given connectivity characteristic. We maintain a lookup table that, for each region, each internal interface, and each connectivity characteristic, stores the optimal way to solve the subproblem inside the region. The lookup table is created bottom-up and the efficiency of this procedure relies on the efficiency of computing the connectivity characteristic for a region from its 2^d subregions one level down in the 2^d-dissection tree. One can find a minimum-cost graph within region Q having a given characteristic by combining minimum-cost graphs within subregions of Q, and this can be done in time $m^{d\,2^d\,r} \cdot 2^{(d\,r)^{\mathcal{O}(d\,r)}}$. This approach has to be refined for regions corresponding to the leaves in the 2^d-dissection tree, where we have to find an optimal graph directly. Unfortunately, since we do not know the locations of Steiner points in an optimal solution, we can only find an approximate solution within every leaf region. Still, this is enough to conclude with the following result (see [8] for more details):

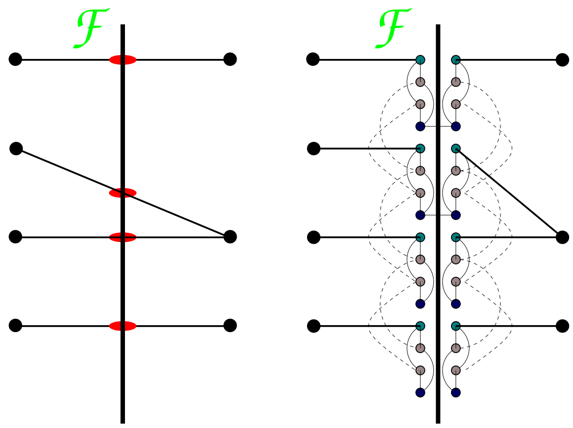

Figure 1.3: Graph H^* constructed in the Patching Lemma. Dotted lines corresponds to the traveling salesman paths. In this example $d = 2, \ell = 4$, and $k = 2$.

Lemma 1.2 *Let* S *be a (perturbed) point set in* $\mathbb{R}^d$ *contained in a bounding box* L^d *and with minimum nonzero inter-distance at least* 8*. Let* m *and* r *be integer parameters. Then, in time* $n \cdot \log L \cdot m^{\mathcal{O}(d\,2^d\,r)} \cdot 2^{(d\,r)^{\mathcal{O}(d\,r)}}$ *one can find a* $(1+\mathcal{O}(\varepsilon))$*-approximation of a minimum-cost* m*-portal-respecting* r*-light Steiner* k*-VCSS for* S*.* □

3.3 Patching Lemma: reducing number of crossings using Steiner points

In this section we discuss a *patching* procedure (initially used by Arora [2] for TSP), which is a key ingredient of our result that for any set of points and for a random shifted dissection of its bounding box, there is always an m-portal-respecting r-light Steiner k-VCSS for S whose cost is low. The patching procedure takes any facet crossed by more than k edges and patches the crossings to reduce the number of crossings to at most k, by augmenting the original graph with new Steiner vertices and new edges (line segments).

Lemma 1.3 (Patching Lemma) *Let* $\mathcal{F}$ *be a* $(d-1)$*-dimensional facet of side length* W *and let* H *be any Steiner* k*-VCSS (for some point set* S*) that crosses* $\mathcal{F}$ *exactly* ℓ *times,* $\ell > k$*. Then, one can break edges of* H *in all but* k *of the crossings and add to* H *new Steiner vertices (that lie infinitesimally close to* $\mathcal{F}$*) and line segments of total cost at most* $\mathcal{O}(k \cdot W \cdot \ell^{1-1/(d-1)})$ *such that* H *changes into a* k*-VCSS* H^* *for* S *that crosses* $\mathcal{F}$ *at most* k *times.*

Proof: Let $x_1, \dots, x_\ell$ be the points at which H crosses the $(d-1)$-dimensional facet W^{d-1}-cube $\mathcal{F}$. For each i, $1 \le i \le \ell$, break the edge (y_i, z_i) crossing $\mathcal{F}$ at x_i into two parts, one on each side of $\mathcal{F}$; we assume that all vertices $y_1, \dots, y_\ell$ are one the same side of $\mathcal{F}$. We consider $2k+4$ copies of each x_i, denoted by $x^+_{i,j}$, and $x^-_{i,j}$ with $0 \le j \le k+1$; $k+2$ copies for each side of $\mathcal{F}$. We assume that all copies are at distance zero from each other.

Now, we define H^*. H^* is obtained from H by removing all the edges crossing $\mathcal{F}$, and inserting the vertices $\{y_1, \dots, y_\ell\} \cup \{z_1, \dots, z_\ell\} \cup \bigcup_{1 \le i \le \ell \,\&\, 0 \le j \le k+1} \{x^+_{i,j}\} \cup \bigcup_{1 \le i \le \ell \,\&\, 0 \le j \le k+1} \{x^-_{i,j}\}$ and eight groups of edges:

(i) two halves of each edge crossing $\mathcal{F}$ in H in the form of the edges $\{y_i, x^+_{i,0}\}$ and $\{x^-_{i,0}, z_i\}$, for all $1 \le i \le \ell$,

(ii) edges crossing $\mathcal{F}$ that connect $x^+_{i,k+1}$ with $x^-_{i,k+1}$, for all $1 \le i \le k$,

(iii) k edges connecting $x^+_{i,0}$ with $x^+_{i,j}$, for all $1 \le i \le \ell$, $1 \le j \le k$,

(iv) edges connecting $x^+_{i,k+1}$ with $x^+_{i,j}$, for all $1 \le i \le \ell$, $1 \le j \le k$,

(v) edges connecting $x^-_{i,0}$ with $x^-_{i,j}$, for all $1 \le i \le \ell$, $1 \le j \le k$,

(vi) edges connecting $x^-_{i,k+1}$ with $x^-_{i,j}$, for all $1 \le i \le \ell$, $1 \le j \le k$,

(vii) edges of a traveling salesman path for $\bigcup_{1 \le i \le \ell}\{x^+_{i,j}\}$, for all $1 \le j \le k$, and

(viii) edges of a traveling salesman path for $\bigcup_{1 \le i \le \ell}\{x^-_{i,j}\}$, for all $1 \le j \le k$.

(Observe that all edges in groups (ii)–(vi) have cost zero (infinitesimally small), because we assumed that for every i and j, $1 \le i < \ell$, $0 \le j \le k+1$, all nodes $x^+_{i,j}$, and $x^-_{i,j}$ are at distance zero from each other.)

It is easy to see that the cost of the non-zero length edges in $H^* \setminus H$ is bounded from above by the cost of the edges in H plus the cost of $2\,k$ traveling salesman paths for the point sets $\bigcup_{1 \le i \le \ell}\{x^+_{i,j}\}$, $\bigcup_{1 \le i \le \ell}\{x^-_{i,j}\}$, $j = 1, \dots, k$, respectively. Now, a well-known result about geometric TSP (see, e.g., Chapter 6 in [29]) implies that for any set of ℓ points contained in a $(d-1)$-dimensional W^{d-1} cube, there is a traveling salesman path of total length smaller than $\mathcal{O}(W\,\ell^{1-\frac{1}{d-1}})$. Therefore, we can conclude that the total additional cost is bounded by $\mathcal{O}(k\,W\,\ell^{1-\frac{1}{d-1}})$.

Finally, it is not hard to show that H^* satisfies the vertex-connectivity requirements. □

3.4 Structure Theorem: there is always a good r-light Steiner k-VCSS

Now, we are ready to present the first Structure Theorem for the k-VCSS problem. This theorem compares the cost of an m-portal-respecting r-light Steiner k-VCSS for a set of points with the cost of an optimal k-VCSS for this set of points, where the optimal solution is not allowed to use Steiner points.

Theorem 1.4 (Structure Theorem) *Let S be a (perturbed) point set in $\mathbb{R}^d$ contained in a bounding box L^d and with minimum nonzero inter-distance at least 8. Pick a random shifted dissection of L^d. Then with probability at least 0.9, there is an m-portal-respecting r-light Steiner k-VCSS for S whose cost is at most $(1+\mathcal{O}(\varepsilon))$-time the optimal k-VCSS for S, where $m = (\mathcal{O}(d\,\log L/\varepsilon))^{d-1}$ and $r = (\mathcal{O}(\sqrt{d}\,k/\varepsilon))^{d-1}$.*

The proof of the Structure Theorem follows from the Patching Lemma above by repeatedly patching the original graph in an appropriated order of facets, following the original approach of Arora. This part of the analysis is technical and subtle, and we only sketch it here; a reader interested in more details is refereed to [8] or [2, 3].

Sketch of the proof: The idea is to transform an optimal k-VCSS for S into a r-light Steiner k-VCSS for S of low cost by applying the Patching Lemma 1.3 to every facet which is crossed too often. Lemma 1.3 ensures that the resulting graph is a r-light Steiner k-VCSS for S. However, since its every application increases the cost of the resulting graph, it is crucial to show that the expected cost of the resulting graph is at most $(1+\mathcal{O}(\varepsilon))$-time the optimal k-VCSS for S. If we prove this claim, then the lemma follows by applying Lemma 1.1 and by Markov inequality.

We bound the total cost of the new edges resulting from invoking the Patching Lemma by charging their cost to grid hyperplanes. For every facet in the dissection we charge the cost of removing the excess of the edges crossing the facet to the grid hyperplane that contains the facet. We show that the expected cost charged to a grid hyperplane $\mathcal{H}$ is at most $\varepsilon\, t(\mathcal{H})/(2\sqrt{d})$, where $t(\mathcal{H})$ is the number of crossings of the hyperplane $\mathcal{H}$ by the optimal k-VCSS for S. Now, the result follows by the linearity of expectations and by the fact that $\sum_{\mathcal{H}} t(\mathcal{H})$ is at most $2\sqrt{d}$ times the cost of the optimal k-VCSS for S (this result is obtained by well-known relation between the ℓ_1 and ℓ_2 norms).

Let us fix a grid hyperplane $\mathcal{H}$ perpendicular to some coordinate axis. Note that within the bounding box L^d, $\mathcal{H}$ forms a L^{d-1} cube. We apply the Patching Lemma to all facets of the dissection that belong to $\mathcal{H}$. We first begin with the smallest facets, and then consider the facets in the increasing order of their sizes. Let c_j be the number of facets in $\mathcal{H}$ of side length $L/2^j$ for which patching has been invoked. For $\ell \le c_j$, let $t_{j,\ell}$ be the number of crossings of the ℓth facet of side length $L/2^j$ for which patching has been applied. Observe that for the ℓth facet of side length $L/2^j$ for which patching has been applied, the total cost of the new edges added by the Patching Lemma is upper bounded by $\mathcal{O}(k\,(L/2^j)\,(t_{j,\ell})^{1-1/(d-1)})$. Therefore, if the largest facet in the hyperplane $\mathcal{H}$ has side-length $L/2^i$, then the total cost of the new edges added by applying the Patching Lemma to all facets in $\mathcal{H}$ is upper bounded by:

$$\mathcal{O}\left(\sum_{j=i}^{\log L}\sum_{\ell=1}^{c_j} k\,(L/2^j)\,(t_{j,\ell})^{1-1/(d-1)}\right) \quad . \tag{1.1}$$

Next, we study the expected cost as above, where the expectation is taken over shifts chosen in the random shifted dissection. Let us assume that the grid hyperplane $\mathcal{H}$ is perpendicular to the sth coordinate axis. Let us fix the random vector $\mathbf{a} = (a_1, \dots, a_d)$ used to determine the random shifted dissection in which all elements are fixed with the exception of a_s, which is kept random. We observe that the random shift in the dissection depends only on the value of a_s, and therefore, if $a_1, \dots, a_{s-1}, a_{s+1}, \dots, a_d$ are fixed, then the probability that the largest facet in the hyperplane $\mathcal{H}$ has side-length $L/2^i$ is $\mathcal{O}(2^i/L)$. Furthermore, one can show that the values of c_j and $t_{j,\ell}$ are independent of a_s. Therefore, the expected cost of all edges added by applying patching to all facets in $\mathcal{H}$ is at most:

$$\begin{aligned}\sum_{i=0}^{\log L}\mathcal{O}(2^i/L)\cdot\mathcal{O}\left(\sum_{j=i}^{\log L}\sum_{\ell=1}^{c_j} k\,(L/2^j)\,(t_{j,\ell})^{1-1/(d-1)}\right) &\le \mathcal{O}\left(\sum_{j=0}^{\log L}\sum_{\ell=1}^{c_j} k/2^j\,(t_{j,\ell})^{1-1/(d-1)}\sum_{i=0}^{j}2^i\right)\\ &= \mathcal{O}(k)\cdot\sum_{j=0}^{\log L}\sum_{\ell=1}^{c_j}(t_{j,\ell})^{1-1/(d-1)} \quad .\end{aligned}$$

Since $t_{j,\ell} \ge r+1$, the bound above is maximized when each $t_{j,\ell} = r+1$, and therefore it is bounded by $\mathcal{O}(k)\cdot(r+1)^{1-1/(d-1)}\cdot\sum_{j=0}^{\log L} c_j$. Now, we need a good upper bound for $\sum_{j=0}^{\log L} c_j$. Since each application of the

Patching Lemma reduces the number of crossings of $\mathcal{H}$ by at least $r + 1 - k$, the definition of $t(\mathcal{H})$ yields:

$$\sum_{j=0}^{\log L} c_j \leq \frac{t(\mathcal{H})}{r+1-k} .$$

Therefore, the expected cost of all edges added by applying patching to all facets in $\mathcal{H}$ is upper bounded by:

$$\mathcal{O}(k) \cdot (r+1)^{1-1/(d-1)} \cdot \sum_{j=0}^{\log L} c_j \leq \mathcal{O}\left(\frac{k\,(r+1)^{1-1/(d-1)}\, t(\mathcal{H})}{r+1-k}\right) .$$

We set $r = (\mathcal{O}(\sqrt{d}\,k/\varepsilon))^{d-1}$ to upperbound this by $\varepsilon\, t(\mathcal{H})/(2\sqrt{d})$. By our arguments above, this implies that the expected cost of all edges added by applying the Patching Lemma (which results in a transformation of the graph into an r-light one) to an optimal k-VCSS for S is at most ε times the cost of the optimal k-VCSS for S.

Finally, we have to transform the graph into m-portal-respecting. We apply the construction presented in Lemma 1.1 with the value of $m = (\mathcal{O}(d\, \log L/\varepsilon))^{d-1}$. Since, by Lemma 1.1, this construction increases in expectation the cost of the graph by at most a factor of $\mathcal{O}(\varepsilon)$, the final result follows. □

3.5 Final result: "pseudo-approximation" schemes for multi-connectivity problems

The results from the previous sections (Lemma 1.2 and Theorem 1.4) are summarized in the following theorem.

Theorem 1.5 *Let k and d be any integers, $k, d \geq 2$, and let ε be any positive real. Let S be a set of n points in $\mathbb{R}^d$. There is a randomized algorithm which finds a Steiner k-VCSS for S, whose cost is at most $(1+\varepsilon)$-time the optimal k-VCSS for S, in time $n \cdot (\log n)^{(\mathcal{O}(\sqrt{d}\,k/\varepsilon))^{d-1}} \cdot 2^{(d\,k/\varepsilon)^{(\mathcal{O}(\sqrt{d}\,k/\varepsilon))^{d-1}}}$ with probability at least 0.9 .*

Furthermore, within the same running time one can find a Steiner k-ECSS for S whose cost is at most $(1+\varepsilon)$-time the optimal k-ECSS for S. Also, all these algorithms can be derandomized in polynomial time.

Observe that when all d, k, and ε are constant, the running time of the randomized algorithm is $n \cdot (\log n)^{\mathcal{O}(1)}$. When d is a constant and k and ε are arbitrary, then the running time is $n \cdot (\log n)^{(k/\varepsilon)^{\mathcal{O}(1)}} \cdot 2^{2^{(k/\varepsilon)^{\mathcal{O}(1)}}}$.

4 PTAS for geometric multi-connectivity problems

The results from the previous section are certainly not fully satisfactory, and a natural question arises if we can obtain a similar result *without using Steiner points* in the solution. In this section, we discuss in details how one can modify the approach from Section 3 to obtain a PTAS for geometric multi-connectivity problems. Even if this method can be seen as a generalization of the approach developed initially by Arora [2], the details of the new construction are significantly different than those used for TSP and related problems. The material in this section is based on [11], an updated and improved version of [9, 10].

The main idea of the PTAS is similar to that from the previous section: we want to prove a result of the form similar to that from Structure Theorem 1.4. However, this time we want to make sure that no new Steiner points difficult to remove are created. We achieve this goal by aiming at a variant of the Structure Theorem that does not require the resulting graph to be r-light but only r-*locally-light*, see the definition below. The difference between these two requirements is insignificant for the dynamic programming phase, but it is critical in our analysis: as we show in our main theorem, there is always an almost optimal k-VCSS for a set of points in $\mathbb{R}^d$ that is m-portal-respecting and r-locally-light for small values of m and r. Before we proceed on, we begin with introducing some new notation.

Relevant crossings and vital edges. A crossing of an edge with a region facet of side-length W in a dissection is called *relevant* if it has exactly one endpoint in the region and its length is at most $2\sqrt{d}\,W$. For a given region Q in a shifted dissection, any edge having exactly one endpoint in Q is called *vital* (for Q).

Special forms of geometric graphs: r-gray and r-locally-light graphs. We say a geometric graph is r-*gray* (with respect to a shifted dissection) if for each region in the dissection there are at most r *relevant crossings*. A graph is r-*locally-light* (with respect to a shifted dissection) if each region in the dissection has at most r vital edges.

Augmented traveling salesman tours. A k*th power* of a graph G is obtained by augmenting G by the edges whose endpoints are connected by paths consisting of at most k edges in G. For any set S and ℓ, an ℓ-*augmented traveling salesman tour on* S is either a clique on S if $|S| \le 2\ell$, or the ℓth power of some TST on S if $|S| \ge 2\ell + 1$.

4.1 Transformation lemmata

In this section we present a variant of the Structure Theorem designed to deal with the problem of finding a minimum-cost k-VCSS for a set of points in $\mathbb{R}^d$. Our goal is to obtain a similar claim as the Structure Theorem 1.4 but without the assumption that the promised graph has Steiner points. We prove this new Structure Theorem in three steps. We take an optimal solution for the minimum-cost k-VCSS problem and we modify it to a suitable form to obtain a graph that is still k-VCSS and whose cost is just slightly larger than that of the minimum-cost. In the first two steps we remove some number of edges (and thus, we do not increase the cost of the graph) to ensure that the resulting graph is first r-gray and then r-locally-light. In the third step we add replacement of the removed edges to ensure that the obtained graph is k-VCSS. The first and the third steps are randomized and they show that in expectation the cost of the resulting graph is at most $(1 + \mathcal{O}(\varepsilon))$ times the minimum-cost k-VCSS.

4.1.1 Local Decomposition Lemma

In this section we discuss our first key result in the analysis, the so-called Local Decomposition Lemma. The Local Decomposition Lemma aims at reducing the number of *relevant* crossings of any given facet to at most k. This procedure is very similar in the spirit to the Patching Lemma 1.3. However, unlike the previously known approaches, the Local Decomposition Lemma *does not use any Steiner points*. The key feature of this construction is that it *only removes edges* and the decision which new edges should be inserted to ensure the connectivity requirements is *delayed*. Instead, a description of properties the new edges must satisfy is provided and these edges are inserted only at the very end of the algorithm (using the TST Covering Lemma 1.11). [2]

To streamline maintaining the connectivity properties of the missing edges, we always describe missing edges in a form of k-augmenting TSTs. The idea is that in order to ensure that a set of points is k-connected it is enough to maintain its TST and then observe that the k-augmenting TST is k-connected. Furthermore, by controlling the cost of a minimum-cost TST for that set of points, we can also control an upper bound for a minimum-cost k-augmenting TST for these points. This will be important in our analysis.

Lemma 1.6 (Local Decomposition Lemma) *Let G be an Euclidean graph on a multiset S of points in $\mathbb{R}^d$. Let $\mathcal{F}$ be a $(d-1)$-dimensional facet of side length W in a dissection of the bounding box of S. If the edges of G form ℓ relevant crossings of $\mathcal{F}$, then there exist a subgraph G^* of G, and two disjoint subsets S_1 and S_2 of S, such that*

- *there are at most $2k^2$ relevant crossings of $\mathcal{F}$ in G^*,*
- *there are a TST on S_1 and a TST on S_2 such that the cost of each is upper bounded by $\mathcal{O}(d\,W\,\ell^{1-\frac{1}{d}})$, and*
- *if G is a k-VCSS on S, then the graph H^* resulting from the graph G^* by adding **any** k-augmented TST on S_1 and **any** k-augmented TST on S_2, is a k-VCSS on S.*

Remark 1.7 *There are three key differences between the Local Decomposition Lemma and the Patching Lemma 1.3: (i) the Local Decomposition Lemma does not introduce any new points to the obtained graph, (ii) it reduces only the number of relevant crossings, leaving the number of arbitrary crossings possibly arbitrarily large, and (iii) it does not produce a k-VCSS on S, but rather it says that one can build one by adding some additional edges.*

Remark 1.8 *For a given TST $\mathcal{T}$ on X it is easy to construct a k-augmented TST $\mathfrak{T}^{\langle k\rangle}$ on X such that the cost of $\mathfrak{T}^{\langle k\rangle}$ is at most $\binom{k+1}{2} \le 2k^2$ times larger than the cost of $\mathcal{T}$ and each hyperplane $\mathcal{H}$ (which does not contain any edge from $\mathcal{T}$) is crossed by the edges of $\mathfrak{T}^{\langle k\rangle}$ at most $\binom{k+1}{2} \le 2k^2$ times more than it is crossed by the edges of $\mathcal{H}$.*

Proof: We can assume $\ell > 2k^2$. We first construct the subgraph G^* and the subsets S_1 and S_2, and then briefly argue about their properties.

[2] One can ask why do we delay inserting the new edges: e.g., in a similar situation in Arora's PTAS for TSP [2], the new edges are inserted at once, as we also do in the analysis of the Structure Theorem 1.4. Note however that Arora [2] and others were always able to place the new edges on the facet for which the Patching Lemma is applied, which facilitates dealing with the new crossings. In the case discussed here, we do not want to create Steiner points and therefore we need to add new edges in arbitrary locations.

Let $\mathcal{E}$ be the set of the ℓ edges of G forming the ℓ *relevant* crossings with $\mathcal{F}$. We define $S_1 = \{x_1, \dots, x_\ell\}$ as the set of endpoints of the edges in $\mathcal{E}$ in the first half-space induced by $\mathcal{F}$ and $S_2 = \{y_1, \dots, y_\ell\}$ as the corresponding set of endpoints of these edges in the other half-space. Next we define G^*. G^* is obtained by removing from G a subset of the edges in $\mathcal{E}$. Let $\mathbb{M}$ be a maximum cardinality subset of $\mathcal{E}$ such that no two edges in the subset are incident. Let $q = \min\{k, |\mathbb{M}|\}$. Then, we define the set $\mathcal{E}^*$ of edges in $\mathcal{E}$ that will remain in G^* to consist of

- the first q edges of $\mathbb{M}$, and
- if $q < k$, then, additionally, for each endpoint v of each edge from $\mathbb{M}$ we add to $\mathcal{E}^*$ $\min\{k-1, \mathrm{deg}_{\mathcal{E}}(v) - 1\}$ edges in $\mathcal{E} \setminus \mathbb{M}$ incident to v, where $\mathrm{deg}_{\mathcal{E}}(v)$ is the number of edges in $\mathcal{E}$ incident to v.

Now, the graph G^* is obtained from G by removing the edges in $\mathcal{E} \setminus \mathcal{E}^*$.

To complete the proof, we must show that G^*, S_1 and S_2 satisfy the properties promised in the lemma. Clearly, $\mathcal{E}^*$ is of size at most $2\,k^2$, and hence there are at most $2\,k^2$ relevant crossings of $\mathcal{F}$ in G^*. Furthermore, each of S_1 and S_2 consists of at most ℓ vertices that are contained in a bounding box of size $\mathcal{O}(\sqrt{d}\,W)$. (Indeed, since the vertices in S_1 and S_2 are endpoints of relevant crossings $\mathcal{F}$, their distance from $\mathcal{F}$ is bounded by $2\sqrt{d}\,W$.) Thus, there is a TST on each of S_1 and S_2 of total length smaller than $\mathcal{O}(d\,W\,\ell^{1-\frac{1}{d}})$ (see, e.g., Section 6 in [29])[3]. The remaining properties can be also easily shown, see [9, 11] for details. □

4.1.2 Weak (too weak) version of Structure Theorem: Global Decomposition Lemma

With the Local Decomposition Lemma above, we can provide a weak version of the Structure Theorem that uses similar arguments as those used in the proof of Theorem 1.4. Since this formulation is too weak for our applications, our goal in the following sections will be to extend it to obtain a stronger result.

Lemma 1.9 (Global Decomposition Lemma) *Let S be a (perturbed) point set in $\mathbb{R}^d$ contained in a bounding box L^d and with minimum nonzero inter-distance at least 8. Pick a random shifted dissection of L^d. Then, there is an r-gray graph G on S and a collection $\mathbb{S}$ of (possible intersecting) subsets of S such that:*

- *the cost of G is not larger than the minimum cost of k-VCSS for S,*
- $r = (\mathcal{O}(k^2\, d^{3/2}/\varepsilon))^d$,
- *there is a graph H consisting of (possible non-disjoint) TSTs on every set $X \in \mathbb{S}$ whose expected (over the choice of the random shifted dissection) total cost is at most $\mathcal{O}(\varepsilon/k^2)$ times the minimum cost of k-VCSS for S, and*
- *the graph resulting from G by adding **any** k-augmented TSTs on each $X \in \mathbb{S}$ is a k-VCSS on S.*

Proof: The proof of this result mimics the proof of the Structure Theorem 1.4 with the exception of a few modifications that are caused by a different form of the Local Decomposition Lemma 1.6. We take a minimum-cost k-VCSS

[3]Note that we need here the assumption that each of S_1 and S_2 is included in a bounding box of size $\mathcal{O}(\sqrt{d}\,W)$. In contrast, in the Patching Lemma 1.3 the points could be arbitrarily far away from each other and thus, for example, there could be no TSP on S_1 of length $o(n)$.

G_{opt} for S and apply a sequence of the Local Decomposition Lemma to make this graph r-gray. Since each application of the Local Decomposition Lemma only removes the edges from G_{opt}, the obtained graph G is a subgraph of G_{opt} and hence its cost is not larger than the minimum cost of k-VCSS for S. Furthermore, the resulting graph is r-gray by Lemma 1.6. This lemma ensures also that if we define $\mathbb{S}$ as the family of sets returned by all calls to the Local Decomposition Lemma, then by adding to G any k-augmented TSTs on all $X \in \mathbb{S}$ we obtain a k-VCSS on S.

What remains to prove is that for the sets $X \in \mathbb{S}$, the total expected costs of minimum-cost TSTs on the sets $X \in \mathbb{S}$ is at most $\mathcal{O}(\varepsilon/k^2)$ times the cost of G_{opt}. The proof of this fact mimics the analysis of the Structure Theorem 1.4. We charge the cost of invoking the Local Decomposition Lemma to a facet contained in a grid hyperplane to that hyperplane. Then, a similar analysis implies that the expected cost of all minimum-cost TSTs on all sets $X \in \mathbb{S}$ resulting from applying the Local Decomposition Lemma to all facets contained in $\mathcal{H}$ is upper bounded by:

$$\mathcal{O}\left(\frac{d \cdot (r+1)^{1-1/d} \cdot t(\mathcal{H})}{r+1-2k^2}\right) .$$

Now, if we set $r = (\mathcal{O}(k^2\, d^{3/2}/\varepsilon))^d$, then the same arguments as those used in the proof of the Structure Theorem 1.4 imply that the expected (over the choice of the random shifted dissection) total cost of minimum-cost TSTs on all sets $X \in \mathbb{S}$ is upper bounded by $\mathcal{O}(\varepsilon/k^2)$ times the minimum cost of k-VCSS for S. □

4.1.3 Filtering Lemma

The Global Decomposition Lemma 1.9 transforms an arbitrary Euclidean graph G into an r-gray graph, so that certain properties of optimal k-vertex connected graphs induced by these graphs are satisfied. There are however stronger requirements for the transformed graph in order to get a PTAS. Even if after applying the Global Decomposition Lemma each facet in an r-gray graph has only $\mathcal{O}(r)$ relevant crossings, many other (longer) crossings are possible. The Filtering Lemma below transforms any r-gray graph into an r^*-locally-light one by removing a set of edges of total small cost, with the parameter r^* just slightly bigger than r.

Lemma 1.10 (Filtering Lemma) *Let $r \geq 1$ and let S be a (perturbed) point set in $\mathbb{R}^d$ contained in a bounding box L^d and with minimum nonzero inter-distance at least 8. For a given shifted dissection, let $G = (S, E)$ be any r-gray graph on S. Then, we can find a subgraph G^* of G that is r^*-locally-light for $r^* = \mathcal{O}(r\, d\, \log(d\, k/\varepsilon))$, and such that the total cost of the edges in $G \setminus G^*$ is at most $\mathcal{O}(\varepsilon/k^2) \cdot \mathrm{cost}(G)$.*

The proof of the Filtering Lemma explores the property that if a graph is r-gray, then for every region Q of side length L in the dissection there are at most $2\, d\, r$ vital edges for Q whose length is in the interval $(2^j \sqrt{d}\, L, 2^{j+1} \sqrt{d}\, L]$ for every value of j. This implies that if there are many vital edges crossing any single facet then most of them (all but a small number of the heaviest edges) have small cost. Therefore, one can transform any r-gray graph into an

r^*-locally-light one by deleting some number of short edges whose total cost (by careful charging arguments) is low.

4.1.4 TST Covering Lemma

In the previous subsections we have transformed an Euclidean graph into the one that possesses fewer edges crossing each facet in the dissection. The key feature of the Global Decomposition Lemma and the Filtering Lemma is that after the graph transformations we are left with some (possible intersecting) sets of nodes that are to be connected is some way (either in pairs by edges or into k-augmented TSTs). The main reason of such construction was to postpone immediately connecting the nodes within each set because this could introduce many new crossings and might destroy the r-locally-lightness of the graph. The TST Covering Lemma below shows how to connect the nodes within each set without increasing the cost of the graph too significantly and without introducing too many crossings of any facet.

We need a definition of a *cover* of a superset that can be seen as a way of connecting multiple TSTs. Let $\mathbb{S}$ be a collection of (not necessarily disjoint) sets. A collection $\mathbb{S}^*$ is called a *cover* of $\mathbb{S}$ if (i) for every $X \in \mathbb{S}$ there is a $Y \in \mathbb{S}^*$ such that $X \subseteq Y$ and (ii) $\bigcup_{X \in \mathbb{S}} X = \bigcup_{Y \in \mathbb{S}^*} Y$. Now, we are ready to state the TST Covering Lemma.

Lemma 1.11 (TST Covering Lemma) *Let S be a (perturbed) point set in $\mathbb{R}^d$ contained in a bounding box L^d and with minimum nonzero inter-distance at least 8. Pick a random shifted dissection of L^d. Let $\mathbb{S}$ be a collection of (possibly non-disjoint) subsets of S. Suppose there is a graph G on S that is a union of TSTs, one for each $X \in \mathbb{S}$, of total cost* $\text{cost}(G)$. *Then, there is a graph G^* such that*

- G^* *is r-light with respect to the dissection, where $r = (\mathcal{O}(\sqrt{d}))^{d-1}$,*
- *there is a cover $\mathbb{S}_{G^*}$ of $\mathbb{S}$ such that G^* is the union of TSTs for each $Y \in \mathbb{S}_{G^*}$, and*
- *the expected (over the choice of the random shifted dissection) cost of G^* is at most $\mathcal{O}(\text{cost}(G))$.* □

The proof of this lemma is an extension of the PTAS for Euclidean TSP by Arora [2] and uses ideas similar to those underlined in the proof of the Structure Theorem 1.4. (Observe that since now we need to find TSTs the appearance of Steiner points in the approach of Arora [2] does not cause any problems.)

4.1.5 Concluding: Structure Theorem for k-vertex connectivity

We conclude with a Structure Theorem for k-vertex connectivity that shows the existence of a low-cost locally-light graph. This theorem is obtained by combining Lemma 1.1, the Global Decomposition Lemma, the Filtering Lemma, and the TST Covering Lemma, when applied to a minimum-cost k-VCSS G for the input point set.

Theorem 1.12 (Structure Theorem II) *Let S be a (perturbed) point set in $\mathbb{R}^d$ contained in a bounding box L^d and with minimum nonzero inter-distance at least 8. Pick a random shifted dissection of L^d. Then with probability at least 0.9, there is an m-portal-respecting r-light k-VCSS for S whose cost is at most $(1 + \mathcal{O}(\varepsilon))$-time the optimal k-VCSS for S, where $m = (\mathcal{O}(d \log L/\varepsilon))^{d-1}$ and $r = (\mathcal{O}(k^2 d^{3/2}/\varepsilon))^d \log(k/\varepsilon)$.*

4.2 PTAS for Euclidean k-vertex and k-edge connectivity

Now, with the Structure Theorem II 1.12 at hand, we are ready to present the "real" PTAS for the minimum-cost k-VCSS problem in geometric graphs. In Section 3.2, we showed how to find a $(1+\mathcal{O}(\varepsilon))$-approximation of a minimum-cost m-portal-respecting r-light Steiner k-VCSS for a set of points, see Lemma 1.2. Similar result holds also for finding a minimum-cost m-portal-respecting r-locally-light k-VCSS for a set of points. The running time of the appropriated dynamic programming scheme is the same as that promised in Lemma 1.2, but this time we can even find an optimal solution (not an $(1+\mathcal{O}(\varepsilon))$-approximation, as in Lemma 1.2). Therefore, we can combine this result with the Structure Theorem II 1.12 to obtain the following result.

Theorem 1.13 *Let k and d be any integers, $k, d \geq 2$, and let ε be any positive real. Let S be a set of n points in $\mathbb{R}^d$. There is a randomized algorithm that in time $n \cdot (\log n)^{(k\,d/\varepsilon)^{\mathcal{O}(d)}} \cdot 2^{2^{(k\,d/\varepsilon)^{\mathcal{O}(d)}}}$ with probability at least 0.9 finds a k-VCSS for S whose cost is at most $(1+\varepsilon)$-time the optimal k-VCSS for S.*

Furthermore, within the same running time one can find a k-ECSS for S whose cost is at most $(1+\varepsilon)$-time the optimal k-ECSS for S. Also, all these algorithms can be derandomized in polynomial time.

When the parameters ε, k, and d are constants, then the running time of the randomized algorithm is $n \cdot \log^{\mathcal{O}(1)} n$. When d and ε are constant and k is arbitrary, the running time becomes $n \cdot (\log n)^{k^{\mathcal{O}(1)}} \cdot 2^{2^{k^{\mathcal{O}(1)}}}$; when ε is arbitrary, it is $n \cdot (\log n)^{(1/\varepsilon)^{\mathcal{O}(1)}} \cdot 2^{2^{(1/\varepsilon)^{\mathcal{O}(1)}}}$. In particular, for a constant dimension d, our scheme leads to a PTAS for the minimum-cost k-VCSS and k-ECSS problems for all k such that $k \leq (\log\log n)^c$ for certain positive constant $c < 1$.

5 Faster PTAS for Euclidean k-ECSSM and 2-connected graphs

Czumaj and Lingas [10] showed that the approximation schemes from Section 4 can be improved in the special case when $k = 2$ and for the minimum-cost k-ECSSM problem. The main source of the improvement is the observation that if we knew a graph/multigraph that contains an optimal or near optimal k-VCSS (k-ECSS, k-ECSSM), then we would be able to apply similar transformations as those described in Section 4 to transform this graph into an r-locally-light one. Comparing to the result from the Structure Theorem II 1.12, we would gain by not having to make the graph m-portal-respecting, because dynamic programming would not have to "guess" the locations of crossings of the facets. This would potentially eliminate term m in the analysis (see Lemma 1.2), and thus greatly improve the running time.

A geometric graph G on a set of points in $\mathbb{R}^d$ is called a t-*spanner* of S, $t \geq 1$, if for any pair of points $p, q \in S$ there is a path in G from p to q of length at most t times the distance between p and q. Gudmundson et al. [23] showed that for any set S of n points in $\mathbb{R}^d$ and for any positive ε, in time $\mathcal{O}((d/\varepsilon)^{\mathcal{O}(d)} \cdot n + d \cdot n \cdot \log n)$ one can find a $(1+\varepsilon)$-spanner of S with maximum degree $(d/\varepsilon)^{\mathcal{O}(d)}$ and with the total cost at most $(d/\varepsilon)^{\mathcal{O}(d)} \cdot \textsc{mst}(S)$.

For a given multigraph H, the *graph induced* by H is the graph obtained by reducing the multiplicity of each edge of H to one. The following lemma formally describes the intuition that a t-spanner contains (implicitly) a t-approximation of the minimum-cost k-ECSSM.

Lemma 1.14 *Let* G *be a* t*-spanner for a point set* S *in* $\mathbb{R}^d$ *and let* k *be an arbitrary positive integer. Then, there exists a* k*-edge-connected multigraph* H *on* S *such that (i) the graph induced by* H *is a subgraph of* G*, (ii) the total cost of* H *is at most* t *times larger than the minimum-cost* k*-edge-connected multigraph on* S*, and (iii) there are no parallel edges in* H *of multiplicity exceeding* k*.*

Now, with a good spanner at hand and with Lemma 1.14, we can proceed with the approach sketched before. This approach is partly inspired by the recent use of spanners to speed-up PTAS for Euclidean versions of TSP due to Rao and Smith [33]. The analysis relies on a series of transformations of a low cost and sparse $(1+\mathcal{O}(\varepsilon))$-spanner for the input point set into an r-locally-light k-edge connected *multigraph* spanning the input set and having nearly optimal cost. With some modifications of the analysis from the Structure Theorem II, one can get the following theorem.

Theorem 1.15 (Structure Theorem III) *Let* S *be a (perturbed) set of* n *points in* $\mathbb{R}^d$ *contained in a bounding box* L^d *and with minimum nonzero inter-distance at least* 8*. Let* G *be a* $(1+\varepsilon)$*-spanner for* S *that has* $n\,(d/\varepsilon)^{\mathcal{O}(d)}$ *edges and has total cost* $(d/\varepsilon)^{\mathcal{O}(d)} \cdot \textsc{mst}(S)$*. Choose a shifted dissection uniformly at random. Then, one can transform* G *into a graph* G^* *on* S *such that with probability (over the random choice of the shifted dissection) at least* 0.9*,*

- G^* *is* r*-locally-light with respect to the shifted dissection,* $r = k\,d^{\mathcal{O}(d)} + \mathcal{O}(k\,d^2\,\log(d/\varepsilon)) + (d/\varepsilon)^{\mathcal{O}(d^2)}$*, and*
- *there exists a* k*-edge-connected multigraph* $\mathbb{H}$ *which is a spanning subgraph of* $\mathbb{G}$ *with possible parallel edges (of multiplicity at most* k*) whose cost is upper bounded by* $(1+\mathcal{O}(\varepsilon))$ *times the minimum-cost* k*-ECSSM for* S*.*

Moreover, the transformation can be done in time $n \cdot 2^{(\mathcal{O}(\sqrt{d}))^{d-1}} + n \cdot (d/\varepsilon)^{\mathcal{O}(d)}\,\log n$.

Once we have the transformation defined in the Structure Theorem III 1.15, we can use dynamic programming, similar to that described in Lemma 1.2 and in Section 4.2, to obtain the following lemma.

Lemma 1.16 *Let* S *be a set of* n *points in* $\mathbb{R}^d$ *contained in a bounding box* L^d *and with minimum nonzero inter-distance at least* 8*. Consider an arbitrary shifted dissection and assume that the* 2^d*-ary dissection tree of* S *is given. Let* G *be an* r*-locally-light graph on* S*, where* $r \geq 1$ *is arbitrary. Then, a minimum-cost* k*-ECSSM* G^* *on* S *for which the induced graph is a subgraph of* G *can be found in time* $n \cdot 2^{d+(k\,r)^{\mathcal{O}(k\,r)}}$.

Therefore, if we combine the Structure Theorem III 1.15 with Lemma 1.16, we directly obtain the following theorem.

Theorem 1.17 *Let* k *and* d *be any integers,* $k, d \geq 2$*, and let* ε *be any positive real. Let* S *be a set of* n *points in* $\mathbb{R}^d$*. There is a randomized algorithm that in time* $n \cdot \log n \cdot (d/\varepsilon)^{\mathcal{O}(d)} + n \cdot 2^{2^{(k^{\mathcal{O}(1)} \cdot (d/\varepsilon)^{\mathcal{O}(d^2)})}}$*, with probability at* *least 0.9 finds a k-ECSSM for S whose cost is at most $(1+\varepsilon)$-time the optimal k-ECSSM for S. The algorithm can be derandomized in polynomial time.*

Observe that when all d, k, and ε are constant, the running time of the randomized algorithm is $\mathcal{O}(n \log n)$. When d and k are constant and ε is arbitrary, the running time becomes $n \log n \, (1/\varepsilon)^{\mathcal{O}(1)} + n \, 2^{2^{(1/\varepsilon)^{\mathcal{O}(1)}}}$. When d and ε are set to constants, then the running time is $\mathcal{O}(n \log n) + n \, 2^{2^{k^{\mathcal{O}(1)}}}$.

5.1 2-connected graphs are not worse than 2-connected multigraphs

The algorithm presented in the previous section does not work for minimum-cost k-VCSS or k-ECSS problems. The reason is that no result similar to that from Lemma 1.14 holds. However, in the special case when $k = 2$, we still can use multigraph approach to obtain a fast PTAS for the minimum-cost 2-VCSS or 2-ECSS problems. Indeed, it is known than any 2-VCSS is also a 2-ECSSM. Therefore, the minimum-cost 2-ECSSM for a set of points is not bigger than the minimum-cost 2-VCSS for the same point set. The following theorem shows that actually, we can always quickly find a 2-VCSS (and hence also 2-ECSS) that has cost not larger than that of a 2-ECSSM.

Lemma 1.18 [10, 15] *A 2-edge-connected multigraph on a set of points in $\mathbb{R}^d$ can be transformed in linear time into a biconnected graph on the same set of points without increasing the total cost.*

In view of this result, we could find an $(1+\varepsilon)$-approximation for the minimum-cost 2-VCSS problem by first running an algorithm for finding a $(1+\varepsilon)$-approximation of the minimum-cost 2-ECSSM and then applying Lemma 1.18. By Theorem 1.17, such randomized algorithm for the minimum-cost 2-VCSS problem runs in time $n \cdot \log n \cdot (d/\varepsilon)^{\mathcal{O}(d)} + n \cdot 2^{2^{(\mathcal{O}(d/\varepsilon))^{\mathcal{O}(d^2)}}}$. However, as Czumaj and Lingas [10, 12] proved, one can obtain further speed-up by improving the dynamic programming scheme from Lemma 1.16 in the special case $k = 2$. For any set S of n points in $\mathbb{R}^d$ and for any Euclidean graph G on S that is r-locally-light with respect to some given shifted dissection, one can use dynamic programming to find in time $n \cdot 2^d \cdot r^{\mathcal{O}(r\,2^d)}$ a minimum-cost 2-edge-connected multigraph on S for which the induced graph is a subgraph of G. This yields the following theorem.

Theorem 1.19 *Let d be any integer $d \geq 2$, and let ε be any positive real. Let S be a set of n points in $\mathbb{R}^d$. There is a randomized algorithm which in time $n \cdot \log n \cdot (d/\varepsilon)^{\mathcal{O}(d)} + n \cdot 2^{(d/\varepsilon)^{\mathcal{O}(d^2)}}$, with probability at least 0.9 finds a 2-VCSS for S whose cost is at most $(1+\varepsilon)$-time the optimal 2-VCSS for S.*

The same holds for the minimum-cost 2-ECSS problem; these algorithms can be derandomized in polynomial time.

For constant d and arbitrary ε, the running time of the randomized algorithm is $n \log n \, (1/\varepsilon)^{\mathcal{O}(1)} + 2^{(1/\varepsilon)^{\mathcal{O}(1)}}$.

6 Lower bounds

The results discussed in previous sections show that various multi-connectivity problems have a PTAS. However, the obtained algorithms work in polynomial-time only for small values of d and k. Are these results just a sign that our methods still need to be improved or they are inherent for the multi-connectivity problems?

As for now, we still do not know if there is a PTAS for large values of k and, say, if we pick $k = \log n$ we do not know if the k-VCSS problem for geometric graphs on the plane (i.e., for $d = 2$) has a PTAS or does not. However, we know that we cannot obtain a PTAS for large values of d. Our basic tool is a powerful result of Trevisan [36] that connects the inapproximability of TSP in geometric graphs with the inapproximability of TSP in the so called so-called 1–2 graphs. A weighted undirected complete graph G is a *1–2 graph* if each of its edges has weight either 1 or 2. It is called a *1–2–Δ graph* if it is a 1–2 graph and each of its vertices is incident to at most Δ edges of weight 1. It is easy to see that in every graph TST has cost that is not smaller than the cost of a minimum-cost 2-VCSS. The following result showing that in 1–2 graphs TST and minimum-cost 2-VCSS coincide is central for our analysis.

Lemma 1.20 *In every 1–2 graph, TST is a minimum-cost 2-VCSS.* □

With this result, general inapproximability results for TST in 1–2 graphs proven by Trevisan [36] directly imply similar results for the 2-VCSS problem.

Theorem 1.21 [9] *There exist constants $\Delta_0 > 0$ and $\varepsilon > 0$ such that, given a 1-2-Δ_0 graph G on n vertices, and given the promise that either its minimum-cost 2-VCSS H has cost n, or its cost is greater than or equal to $(1+\varepsilon)\,n$, it is $\mathcal{NP}$-hard to distinguish which of the two cases holds. In particular, it is $\mathcal{NP}$-hard to approximate within $(1+\varepsilon)$ the cost of a minimum-cost 2-VCSS of a 1-2-Δ_0 graph.*

The next result is a direct application of Theorem 1.21 combined with classical results on metric embeddings.

Theorem 1.22 [9] *For any fixed $p \geq 1$ there exists a constant $\xi > 0$ such that it is $\mathcal{NP}$-hard to approximate within $1+\xi$ the minimum-cost 2-connected graph spanning a set of n points in the ℓ_p metric in $\mathbb{R}^{\log n}$.*

Corollary 1.23 *The minimum k-VCSS problem in graphs of maximum degree bounded by some constant is APX-hard and hence does not have a PTAS unless $\mathcal{P} = \mathcal{NP}$.*

One can easily modify the proofs of the theorems presented in this section in order to obtain similar inapproximability results for the problem of finding a minimum-cost k-edge-connected subgraph of a k-edge-connected graph.

7 Extensions to other related problems

The results and techniques we discussed in the previous sections can be applied to various related problems.

7.1 Pseudo-approximations and Steiner k-VCSS/ECSS

It is not hard to improve the pseudo-approximation result obtained in Theorem 1.5 by modifying the result from Theorem 1.17. We begin with finding a k-ECSSM whose cost is within $1+\varepsilon$ of the minimum using the result from Theorem 1.17. Then, we can trivially transform this multigraph into a Steiner k-VCSS by placing $k-1$ Steiner points on each input point (i.e., at the length zero from it) and forming a k-clique of zero cost out of the point and its associated $k-1$ Steiner points. The cost of the resulting graph is within $(1+\varepsilon)$ of the minimum-cost k-ECSSM for the input set, which, in turns, does not exceed $(1+\varepsilon)$ times the minimum-cost k-VCSS on the input set. Such a Steiner k-VCSS can be found in (asymptotically) the same time as required by Theorem 1.17 to find the k-ECSSM, which is significantly better than the result in Theorem 1.5. The same approach works also for Steiner k-ECSS.

7.2 Steiner k-connectivity — real approximation schemes

The techniques described in the survey can be also used to derive efficient approximation schemes for Euclidean minimum-cost Steiner k-connectivity. In contrast to the result in Section 7.1, our goal is to find a Steiner k-VCSS (or k-ECSS) for a set of points S in $\mathbb{R}^d$ whose cost is at most $(1+\varepsilon)$ times the minimum-cost Steiner k-VCSS (k-ECSS, respectively) for S; so both, the solution found and the optimal solution are allowed to use Steiner points.

The main difficulty with extending the result from Section 7.1 to a real PTAS for Steiner k-VCSS/ECSS is that the spanners used in the Structure Theorem III 1.15 and in the PTAS from Theorem 1.17 do not include Steiner points. Nevertheless, one can decompose optimal Steiner solutions for k-connectivity and combine this decomposition with the construction of banyans due to Rao and Smith [33]. The case of $k=2$ is most interesting. Extending the work of Hsu and Hu [25], Czumaj and Lingas [10] showed a new structural characterization of minimum-cost Steiner biconnected graphs that lead to a decomposition of an optimal Steiner solution into minimum Steiner trees. This opened the possibility of using the so called $(1+\varepsilon)$-banayans, for the purpose of approximating the Euclidean minimum Steiner tree problem. As the result, Czumaj and Lingas [10] obtain a PTAS for Euclidean minimum-cost Steiner biconnectivity and Euclidean minimum-cost two edge connectivity; the algorithms run in time $\mathcal{O}(n \log n)$ for any constant dimension and ε. For general d and ε, the running time is $n\ \log n\,(d/\varepsilon)^{\mathcal{O}(d)} + n\,2^{(d/\varepsilon)^{\mathcal{O}(d^2)}} + n\,2^{2^{d^{\mathcal{O}(1)}}}$.

7.3 Survivable networks

Czumaj et al. [13] extended the analysis from previous sections (in particular, Theorem 1.19) to a more general problem of survivable networks. They considered the variant of the *survivable network design problem* in which for a given set S of n points in Euclidean space $\mathbb{R}^d$ and a connectivity requirement function $r : S \to \mathbb{N}$, the goal is to find a minimum-cost graph G on S such that for any pair of points $x, y \in S$, G has $\min\{r(x), r(y)\}$ internally vertex-disjoint paths between x and y. The two most basic (and of largest practical relevance) variants of this problem are those in which $r(x) \in \{0, 1\}$ and when $r(x) \in \{0, 1, 2\}$, for any point $x \in S$.

First, for the simplest case in which $r(x) \in \{0, 1\}$ for any point $x \in S$, that is, for the *Steiner tree problem*[4], Czumaj et al. [13] designed a randomized algorithm that, for any constant d and any constant ε, in time $\mathcal{O}(n \log n)$ finds a Steiner tree whose cost is at most $(1 + \varepsilon)$ times larger than the minimum. For general d and ε, its running time is $n \log n \, (d/\varepsilon)^{\mathcal{O}(d)} + n \, 2^{(d/\varepsilon)^{\mathcal{O}(d^2)}} + n \, 2^{2^{d^{\mathcal{O}(1)}}}$.

Next, for the case when $r(x) \in \{0, 1, 2\}$ for any point $x \in S$ (this is the classical problem investigated thoroughly by Grötschel and Monma *et. al.* [20, 21, 22, 32, 35]), Czumaj et al. [13] extended algorithm for the Steiner tree problem to design an algorithm that, for any constant d and any constant ε, in time $\mathcal{O}(n \log n)$ finds a graph satisfying all the vertex connectivity requirements and having the cost at most $(1 + \varepsilon)$ times the minimum. When d and ε are allowed to be arbitrarily, its running time is $n \log n \, (d/\varepsilon)^{\mathcal{O}(d)} + n \, 2^{(d/\varepsilon)^{\mathcal{O}(d^2)}} + n \, 2^{2^{d^{\mathcal{O}(1)}}}$.

Finally, essentially the same techniques can be used to obtain a PTAS for the multigraph variant, where the edge-connectivity requirements satisfy $r(x) \in \{0, 1, \ldots, k\}$ and $k = \mathcal{O}(1)$.

All these approximation schemes are randomized, but they can be *derandomized* in a polynomial time.

7.4 Finding low-cost k-VCSS and k-ECSS in planar graphs

Recently, there has been also a progress in designing approximation schemes for the 2-VCSS and 2-ECSS problem in planar graphs [7, 5]. Similarly as for the TSP problem in planar graphs [4, 19], the first step towards an efficient approximation scheme has been achieved for unweighted graphs. Czumaj et al. [7] showed that for every positive ε, for a given undirected graph planar G with n vertices, one can find in time $n^{\mathcal{O}(1/\varepsilon)}$ a 2-VCSS (or 2-ECSS) of G whose total number of edges is at most $(1+\varepsilon)$ times the minimum number of edges in any 2-VCSS (or 2-ECSS, respectively) of G; this gives a PTAS for the unweighted version of the 2-VCSS and 2-ECSS problem in planar graphs. In fact, the approximation scheme provided in [7] works also for the weighted case, but then the running time becomes $n^{\mathcal{O}(\gamma/\varepsilon)}$, where γ is the ratio of the total edge cost to the optimum solution cost.

Soon later, Berger et al. [5] modified the scheme from [7] and obtained a *quasi-polynomial time approximation scheme* for the 2-VCSS and 2-ECSS problem in planar graphs. Their algorithm runs in time $n^{\mathcal{O}(\log n \, \log(1/\varepsilon)/\varepsilon)}$ and finds a 2-VCSS (or 2-ECSS) of G whose total cost is at most $(1 + \varepsilon)$ times the minimum-cost 2-VCSS (or 2-ECSS, respectively) of G. Furthermore, their algorithm can be extended to solve within the same runtime bounds the survivable network design problem in planar graphs in which $r(x) \in \{1, 2\}$ for any vertex.

The underlying techniques developed for the approximation schemes for the 2-VCSS and 2-ECSS problem in planar graphs were surprisingly similar to those used for geometric graphs: a combination of (new) separator theorems

[4] Note that this variant of the Steiner tree problem is different from the Steiner tree problem considered by Arora [2], for which a PTAS is also known [2, 3] (see also [31, 33]). The variant considered in this survey requires that all locations of Steiner points are given in advance (they are the points $x \in S$ with $r(x) = 0$), while in the other variant, all points in $\mathbb{R}^d$ could be used as Steiner points.

with dynamic programming, and then new constructions of *light spanners* for planar graphs. For more details, we refer interested readers to the original papers [7, 5].

8 Final comments

In this paper, we surveyed recent approximation schemes for various variants of network design problems for geometric graphs. Our main goal was not only to show the result, but also to demonstrate a variety of new techniques developed to coupe with these problems.

8.1 Interesting open questions:

In our context, perhaps the most intriguing open problem for now is whether the minimum-cost 2-VCSS and 2-ECSS problems for planar graphs has a PTAS. We conjecture that this is indeed the case, but so far, the existing techniques seem to be too weak. Further, it would be interesting to see if there is a PTAS for the k-VCSS/ECSS problem in planar graphs for $k = 3, 4$ (note that for $k \geq 5$ no planar graph can be k-vertex-connected).

Another interesting open problem is whether there exists a PTAS for the geometric minimum-cost k-VCSS and k-ECSS problems for very large values of k. The techniques presented in this survey seem to work only for the values of k up to $(\log \log n)^c$ for certain positive constant $c < 1$. What about large values of k?

Finally, and perhaps most importantly, how practical are the methods discussed in this survey? Even though, most probably any direct implementation of the PTAS for k-connectivity problems would be inferior to the existing heuristic implementations discussed, e.g., [21, 22, 32, 35], we believe that the techniques presented in this survey when combined with heuristics could lead to significant improvements in practical implementations.

Acknowledgements

We thank Hairong Zhao, Christos Levcopoulos, Sanjeev Arora and Luca Trevisan for their valuable comments and suggestions. We thank an anonymous referee for pointing out the relation between TST and 2-VCSS in 1-2-graphs.

References

[1] Ahuja, R.K., Magnanti, T.L., Orlin, J.B., and Reddy, M.R., Applications of network optimization, in *Handbooks in Operations Research and Management Science*, Vol. 7: Network Models, Ball, M.O., Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., eds., 1995, chap 1.

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

[3] Arora, S., Approximation schemes for $\mathcal{NP}$-hard geometric optimization problems: A survey, *Math. Prog., Series B*, 97(1–2), 43, July 2003.

[4] Arora, S., Grigni, M., Karger, D., Klein, P., and Woloszyn, A., A polynomial-time approximation scheme for weighted planar graph TSP, in *Proc. of SODA*, 1998, 33.

[5] Berger, A., Czumaj, A., Grigni, M., and Zhao, H., Approximation schemes for minimum 2-connected spanning subgraphs in weighted planar graphs, in *Proc. of ESA*, 2005, 472.

[6] Cheriyan, J. and Vetta, A., Approximation algorithms for network design with metric costs, in *Proc. of STOC*, 2005, 167.

[7] Czumaj, A., Grigni, M., Sissokho, P., and Zhao, H., Approximation schemes for minimum 2-edge-connected and biconnected subgraphs in planar graphs, in *Proc. of SODA*, 2004, 489.

[8] Czumaj, A., and Lingas, A., A polynomial time approximation scheme for Euclidean minimum cost k-connectivity, in *Proc. of ICALP*, 1998, 682.

[9] Czumaj, A., and Lingas, A., On approximability of the minimum-cost k-connected spanning subgraph problem, in *Proc. of SODA*, 1999, 281.

[10] Czumaj, A. and A. Lingas, A., Fast approximation schemes for Euclidean multi-connectivity problems, in *Proc. of ICALP*, 2000, 856.

[11] Czumaj, A. and Lingas, A., Polynomial time approximation schemes for Euclidean multi-connectivity problems. Manuscript, 2005.

[12] Czumaj, A. and Lingas, A., Fast approximation schemes for Euclidean biconnectivity problems and for multi-connectivity problems. Manuscript, 2005.

[13] Czumaj, A., Lingas, A., and Zhao, H., Polynomial-time approximation schemes for the Euclidean survivable network design problem, in *Proc. of ICALP*, 2002, 973.

[14] Du, D.Z. and Hwang, F.K., A proof of the Gilbert-Pollak conjecture on the Steiner ratio, *Algorithmica*, 7, 121, 1992.

[15] Frederickson, G.N. and JáJá, J., On the relationship between the biconnectivity augmentation and traveling salesman problem, *Theor. Comp. Sci.*, 19(2), 189, 1982.

[16] Gabow, H.N., Goemans, M.X., and Williamson, D.P., An efficient approximation algorithm for the survivable network design problem, *Math. Prog., Series B*, 82(1-2), 13, 1998.

[17] Garey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, New York, NY, 1979.

[18] Goemans, M.X. and Williamson, D.P., The primal-dual method for approximation algorithms and its application to network design problems, in *Approximation Algorithms for $\mathcal{NP}$-Hard Problems*, Hochbaum, D.S., ed., PWS Publishing Company, 1996, chap 4.

[19] Grigni, M., Koutsouias, E., and Papadimitriou, C., An approximation scheme for planar graph TSP, in *Proc. of FOCS*, 1995, 640.

[20] Grötschel M., and Monma, C.L., Integer polyhedra arising from certain network design problems with connectivity constraints, *SIAM J. on Disc. Math.*, 3(4), 502, 1990.

[21] Grötschel, M., Monma, C.L., and Stoer, M., Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints, *Oper. Res.*, 40(2), 309, 1992.

[22] Grötschel, M., Monma, C.L., and Stoer, M., Design of survivable networks, in *Handbooks in Operations Research and Management Science*, vol. 7, Network Models, Ball, M.O., Magnanti, T.L., Monma, C.L., and Nemhauser, G.L., eds., 1995, chap 10.

[23] Gudmundson, J., Levcopoulos, C., and Narasimhan, G., Fast greedy algorithms for constructing sparse geometric spanners, *SIAM J. on Comput.*, 31(5), 1479, 2002.

[24] Hochbaum, D.S., ed., *Approximation Algorithms for $\mathcal{NP}$-Hard Problems*, PWS Publishing Company, Boston, MA, 1996.

[25] Hsu, D.F., and Hu, X.-D., On short two-connected Steiner networks with Euclidean distance, *Networks*, 32(2), 133, 1998.

[26] Hwang, F.K., Richards, D.S., and Winter, P., *The Steiner Tree Problem*, Vol 53 of *Annals of Disc. Math.*, North-Holland, Amsterdam, 1992.

[27] Khuller, S., Approximation algorithms for finding highly connected subgraphs, in *Approximation Algorithms for $\mathcal{NP}$-Hard Problems*, Hochbaum, D.S., ed., PWS Publishing Company, Boston, MA, 1996, chap 6.

[28] Kortsarz, G. and Nutov, Z., Approximating k-node connected subgraphs via critical graphs, *SIAM J. on Comput.*, 35(1), 247, 2005,

[29] Lawler, E.L., Lenstra, J.K., Rinnooy Kan, and A.H.G., Shmoys, D.B., editors, *The Traveling Salesman Problem*, John Wiley & Sons, New York, NY, 1985.

[30] Mihail, M., Shallcross, D., Dean, N., and Mostrel, M., A commercial application of survivable network design: ITP/INPLANS CCS network topology analyzer, in *Proc. of SODA*, 1996, 279.

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

[32] Monma, C.L. and Shallcross, D.F., Methods for designing communications networks with certain two-connected survivability constraints, *Oper. Res.*, 37(4), 531, 1989,

[33] Rao, S.B. and Smith, W.D., Approximating geometrical graphs via "spanners" and "banyans", in *Proc. of STOC*, 1998, 540.

[34] Schrijver, A., *Combinatorial Optimization Polyhedra and Efficiency*, Springer-Verlag, New York, 2003.

[35] Stoer, M., *Design of Survivable Networks*, volume 1531 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 1992.

[36] Trevisan, L., When Hamming meets Euclid: the approximability of geometric TSP and Steiner Tree, *SIAM J. on Comput.*, 30(2), 475, 2000.