Approximation of Octilinear Steiner Trees Constrained by Hard and Soft Obstacles*

Matthias Müller–Hannemann and Anna Schulze

[1] Technische Universität Darmstadt, Department of Computer Science,
Hochschulstraße 10, 64289 Darmstadt, Germany,
muellerh@algo.informatik.tu-darmstadt.de,
http://www.algo.informatik.tu-darmstadt.de
[2] Zentrum für Angewandte Informatik Köln, Weyertal 80, 50931 Köln, Germany
schulze@zpr.uni-koeln.de

Abstract. The novel octilinear routing paradigm (X-architecture) in VLSI design requires new approaches for the construction of Steiner trees. In this paper, we consider two versions of the shortest octilinear Steiner tree problem for a given point set K of terminals in the plane: (1) a version in the presence of hard octilinear obstacles, and (2) a version with rectangular soft obstacles.

The interior of hard obstacles has to be avoided completely by the Steiner tree. In contrast, the Steiner tree is allowed to run over soft obstacles. But if the Steiner tree intersects some soft obstacle, then no connected component of the induced subtree may be longer than a given fixed length L. This kind of length restriction is motivated by its application in VLSI design where a large Steiner tree requires the insertion of buffers (or inverters) which must not be placed on top of obstacles.

For both problem types, we provide reductions to the Steiner tree problem in graphs of polynomial size with the following approximation guarantees.

Our main results are (1) a 2–approximation of the octilinear Steiner tree problem in the presence of hard rectilinear or octilinear obstacles which can be computed in $O(n \log^2 n)$ time, where n denotes the number of obstacle vertices plus the number of terminals, (2) a $(2 + \varepsilon)$–approximation of the octilinear Steiner tree problem in the presence of soft rectangular obstacles which runs in $O(n^3)$ time, and (3) a polynomial time $(1.55 + \varepsilon)$–approximation of the octilinear Steiner tree problem in the presence of soft rectangular obstacles.

Keywords: approximation algorithms, computational geometry, octilinear Steiner trees, obstacles, VLSI design

1 Introduction

Background and motivation. Octilinear routing is a novel routing paradigm in VLSI design, the so-called X-architecture [X], which has recently been introduced. In addition to vertical and horizontal wires, octilinear routing allows wiring in 45- and 135-degree directions. Compared to traditional and state-of-the-art rectilinear (Manhattan) routing, such a technology promises clear advantages in wire length but also in via reduction. As a consequence a significant chip performance improvement and power reduction can be obtained (with estimations being in the range of 10% to 20% improvement) [Tei02,CCK$^+$03,PWZ04]. To enable such a technology, novel algorithmic approaches for the construction of octilinear Steiner trees are needed.

* An extended abstract of this paper appears in Proceedings of the 10th Scandinavian Workshop on Algorithm Theory (SWAT 2006), LNCS 4059, pp. 242-254, 2006, Springer.

An *octilinear Steiner tree* is a tree that interconnects a set of points (*terminals*) in the plane with minimum length such that every line segment uses one of the four given orientations. Even more general routing architectures are obtained if a fixed set of uniformly oriented directions is allowed. For an integer parameter $\lambda \geq 2$, consecutive orientations are separated by a fixed angle of π/λ. A λ-*geometry* is a routing environment in which every line segment uses one of the given orientations. Manhattan routing can then be seen as the special case $\lambda = 2$ and the X-architecture (*octilinear routing*) as the case $\lambda = 4$. In this paper we focus on the octilinear case (although most of our results can be generalized to arbitrary $\lambda \geq 2$). We study approximation algorithms for the octilinear Steiner tree problem with different types of obstacles.

Hard and soft obstacles. In VLSI design preplaced macros or other circuits are obstacles. Throughout this paper, an *obstacle* is a connected region in the plane bounded by a simple polygon such that all obstacle edges lie within the 4–geometry (*octilinear obstacle*). If all boundary edges of an obstacle are rectilinear, we call such an obstacle a *rectilinear obstacle*. For a given set of obstacles $\mathcal{O}$ we require that the obstacles be disjoint, except for possibly a finite number of common points. By ∂O we denote the boundary of an obstacle O. In practice, obstacles can be assumed to be axis-parallel rectangles.

An obstacle which prohibits wiring and therefore has to be avoided completely will be referred to as a *hard obstacle*. Due to the availability of several routing layers, most obstacles usually do not block wires, but it is impossible to place a buffer (or inverter) on top of an obstacle. A large Steiner tree requires the insertion of buffers (or inverters) in such a way that no induced subtree without any buffers becomes too large. This application in VLSI design motivates and translates into our model of *soft obstacles*. In this case the Steiner tree is allowed to run over obstacles; however, if we intersect the Steiner tree with some obstacle, then no connected component of the induced subtree may be longer than a given fixed length L.

Related work. The rectilinear and the Euclidean Steiner tree problem have been shown to be NP-Hard in [GJ77] and [GGJ77], respectively. Quite recently, we have been able to prove that the octilinear Steiner tree problem is also NP-hard in the strong sense [MS05].

Most previous work on the octilinear Steiner tree problem considered the problem *without obstacles*. Exact approaches to the octilinear Steiner tree problem have been developed by Nielsen, Winter and Zachariasen [NWZ02] and Coulston [Cou03]. Nielsen et al. report the exact solution to a large instance with 10000 terminals within two days of computation time.

For rectilinear Steiner tree problems for point sets in the plane, the most successful approaches are based on transformations to the related Steiner tree problem in graphs. Given a connected graph $G = (V, E)$, a length function ℓ, and a set of terminals $K \subseteq V$, a *Steiner tree* is a tree which contains all vertices of K and is a subgraph of G. A Steiner tree T is a *Steiner minimum tree* of G if the length of T is minimum among all Steiner trees. An implementation by Althaus, Polzin and Daneshmand [APD03] is the currently strongest available exact approach for both the Steiner tree problem in graphs and the rectilinear Steiner tree problem. The best available approximation guarantee for the Steiner problem in general graphs is $\alpha = 1 + \frac{\ln 3}{2} \approx 1.55$, obtained by Robins and Zelikovsky [RZ00].

Unfortunately, in the octilinear case, the only known transformation to the Steiner tree problem in graphs is based on a generalization of the Hanan-grid and requires $O(n^{2^{O(n)}})$ many vertices [DH92,LS96,LX00]. Hence, this transformation is not polynomial. Müller-Hannemann and Schulze [MS05] recently constructed a graph of size $O(n^2/\varepsilon^2)$ which contains for every $\varepsilon > 0$ a $(1 + \varepsilon)$-approximation for the case without obstacles and with hard obstacles.

We would like to point out that the well-known approximation schemes of Arora [Aro98] and Mitchell [Mit99] are only applicable to the octilinear Steiner tree problem without obstacles, but it seems to be unknown whether polynomial time approximation schemes are possible. For the octilinear Steiner tree problem *without obstacles* heuristics have been proposed by Kahng et al. [KMZ03] and Zhu et al. [ZZJ+04].

Müller-Hannemann and Peyer [MP03] showed that the rectilinear Steiner tree problem in the presence of soft obstacles can be 2-approximated in $O(n^2 \log n)$ time, where n denotes the number of terminals plus the number of obstacle vertices. They also presented a $(1.55 + \varepsilon)$-approximation for rectangular obstacles. In this paper we generalize these result to the octilinear Steiner tree problem. However, it will turn out that the problem becomes substantially more complicated and requires novel techniques both in design and analysis. We are not aware of any other exact approaches or heuristics in the presence of obstacles.

Our methodology. We provide transformations from the octilinear Steiner tree problem in the plane with obstacles to the Steiner tree problem in graphs which contain approximate solutions. To achieve a 2–approximation our aim is to construct a *path preserving graph*, i.e., a graph which contains a shortest octilinear path between any pair of terminals. With respect to obstacles, the graph should only contain feasible paths and only feasible Steiner trees. (Note that for soft obstacles the latter does not follow from the feasibility of all paths.) These properties ensure that any approximation algorithm based on this graph for the Steiner tree problem will produce a feasible Steiner tree. In particular, we may use Mehlhorn's [Meh88] implementation of a minimum spanning tree based approximation which runs in time $O(m+n \log n)$ on a graph with n nodes and m edges. This approach yields a 2–approximation, and we can show that the analysis is asymptotically tight.

Heading for a good running time, our secondary goal is to construct small path preserving graphs. Shortest paths in the presence of polygonal obstacles have already been studied intensively. See the surveys of Mitchell [Mit00] and Lee, Yang, and Wong [LYW96]. Our construction of small path preserving graphs generalizes techniques in previous work of Wu et al. [WWSW87] and Clarkson et al. [CKV87].

To achieve a $(1 + \varepsilon)$-approximation we develop a different technique. A Steiner tree is a *full* Steiner tree if all its terminals are leaves. Any Steiner tree can be decomposed into its full components. A *t-restricted Steiner tree* is a Steiner tree where all full components have at most t terminals. The boundary of each obstacle is discretized by auxiliary vertices with a distance of at most Δ between neighboring vertices. (We show that Δ can be chosen so that we obtain a polynomial number of auxiliary vertices and still achieve the desired accuracy.) Inside obstacles, we approximate an optimal tree with the help of t-restricted Steiner trees for some constant t. Each of these trees respects the length restriction L for the obstacle. Outside obstacles, a grid-like graph through the terminals and obstacle vertices is refined by additional lines so that it contains a sufficiently close approximation. These ideas will be made precise in Section 4.

Our contribution. We summarize the main results of this paper:

- There is a 2–approximation of the octilinear Steiner tree problem in the presence of hard octilinear obstacles which can be computed in $O(n \log^2 n)$ time, where n denotes the number of obstacle vertices plus the number of terminals.
- For any integer k, we obtain a $(2 + \frac{1}{k})$–approximation which runs in $O(k^2 n^3)$ time for the octilinear Steiner tree problem with soft rectangular obstacles.
- We can show that our analysis of the minimum spanning tree heuristic is tight by constructing a class of instances for which our approximation algorithm asymptotically achieves a performance guarantee of 2.
- We construct a graph of polynomial size which contains a $(1 + \varepsilon)$-approximation of the octilinear Steiner tree problem with rectangular soft obstacles. Hence, the currently strongest

approximation guarantee by Robins and Zelikovsky for the Steiner tree problem in graphs implies a $(1.55+\varepsilon)$-approximation for this problem. This matches the best known guarantees for the rectilinear case [MP03].

Overview. The remaining part of the paper is organized as follows. In Section 2, we describe how to construct shortest path preserving graphs for hard obstacles of size $O(n \log n)$. The more complicated construction of shortest paths for soft obstacles will be explained in Section 3. Our main result is presented in Section 4, where we show how to construct a graph of polynomial size which contains a $(1+\varepsilon)$-approximation for rectangular soft obstacles. Finally, we conclude with a short summary and suggestions for future work.

2 Octilinear Shortest Paths Amidst Hard Obstacles

Throughout this section, let K be a set of points (terminals) in the plane and O be a set of octilinear obstacles. Denote by V_O the set of obstacle vertices. Let $n = |K| + |V_O|$. In this section we will show how to construct shortest path preserving graphs.

Octilinear track graphs. As a first step we construct a path preserving graph based on visibility. Our construction may be viewed as a generalization of that of Wu et al. [WWSW87], which was designed for rectilinear polygons and rectilinear paths. To simplify our discussion we add to our scene a bounding box containing all obstacles and all terminals. Clearly all desired paths will run within this bounding box.

A *track tr* generated by a point t and an orientation is a line segment that starts at t and ends when it first hits an obstacle edge or the bounding box. The generated endpoints of tracks are called *track-induced Steiner points*.

For each terminal t and each feasible orientation we construct a track in both directions from t. Similarly, we introduce tracks for each convex obstacle vertex v. More precisely, if $e_1 = (v_1, v)$ and $e_2 = (v, v_2)$ are polygon edges incident with v in clockwise order of the polygon, denote by r_1 the ray in direction from v_1 to v, and by r_2 the ray in direction from v_2 to v. We construct a track generated by v for all feasible directions which do neither go through the interior of the obstacle nor through the interior of the sector spanned by ray r_1 and r_2 in counter-clockwise order. See Fig. 1. The intersections among all tracks and their endpoints are made the vertices of the track graph. The edges are the track segments between the intersections. The construction is completed by adding edges connecting two consecutive track-induced Steiner points or polygon vertices along the boundary of each obstacle. The length of an edge in the track graph is simply the octilinear distance between its endpoints. See the middle part of Fig. 1 for a small example which illustrates this construction. The track graph consists of $O(n)$ many tracks which induce $O(n^2)$ many vertices and edges.

For rectilinear paths it is possible to restrict the track construction to so-called extreme edges [WWSW87]. An obstacle edge is *extreme* if its two adjacent edges lie on the same side of the line containing the edge. Note that such a reduction is not possible for octilinear paths.

Sparser path-preserving graphs. To improve upon the quadratic space bound of the track graph we use an idea of Clarkson et al. [CKV87] and adapt their approach to the octilinear case. We construct a sparser path-preserving graph $G = (V, E)$ as follows. The vertex set is constructed in two rounds. In the first round, we create V_1 as the union of

1. the set of all terminals K,
2. the set of all obstacle vertices V_O, and
3. the set of track-induced Steiner points for tracks induced by K and V_O.

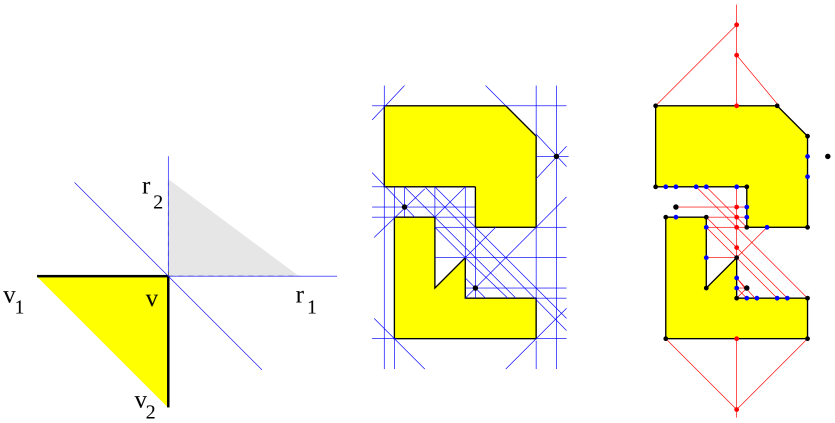

Fig. 1. Illustration of the graph construction. Left: The tracks around a convex vertex v of some obstacle. There is no track inside the shaded area. Middle: The track graph for an instance with three terminals (black dots) and two hard octilinear obstacles. Right: The first vertical cut line construction.

With respect to V_1 we create the set V_2 recursively by adding more Steiner points along vertical, horizontal and diagonal so-called *cut lines*. We explain the construction for vertical cut lines. A vertical cut line is placed at the median of the x-coordinates of all vertices. Vertices in V_1 generate projection points on the line. Projections are performed in all feasible orientations so that we may get up to three projections points on the line for each vertex in V_1 (in the rectilinear setting, Clarkson et al. need to project only orthogonally onto the cut line). Two points are mutually *visible* to each other if the straight line segment between them contains no obstacle point in its interior. All those projection points on a cut line which are visible from some inducing point in V_1 are put into a vertex set V_2. Moreover, we add the intersection points of the cut line with obstacle vertices to V_2. The following edges are inserted into E. Two consecutive Steiner points on the line are connected by an edge if these points are visible to each other. We also add edges from each vertex in V_1 to its corresponding projection points.

This procedure is repeated recursively with the vertices respectively on the left and right sides of the cut line. The union of all these vertices yields $V = V_1 \cup V_2$. See Fig. 1 for a vertical cut line on the highest level. There are $O(\log n)$ many levels of recursion, and in each level we will create $O(n)$ many vertices and edges. This gives in total $O(n \log n)$ vertices and edges. Finally, for each obstacle we have edges between consecutive vertices from V on its boundary.

Correctness of the construction. We now present the proof that the graph G has the desired properties.

Theorem 2.1. *For any two vertices from K the constructed graph G contains a shortest octilinear path.*

The validity of this theorem is based on the following three lemmas. A *segment* S of a path P is a subpath with the property that all its edges have the same orientation. Hence, any path can be thought of as composed by a sequence of inclusion-maximal segments, i.e. longest subpaths with the same orientation.

Lemma 2.2. *The track graph contains a shortest octilinear path for any two vertices from K.*

Proof. For arbitrary $s, t \in K$ choose a shortest octilinear path P which has the fewest number k of inclusion-maximal segments which do not lie in the track graph. Note that k is finite

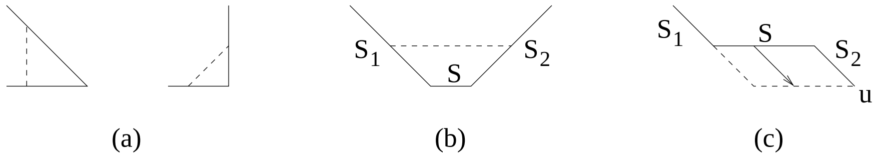

Fig. 2. If the segments meet at 45°, 90° (a) or S_1 and S_2 meet s at 135° and have different orientations (b) and all segments do not lie on the boundary of an obstacle, we can shorten the path (denoted by dashed lines). If S_1 and S_2 have the same orientation, we can slide S towards u.

since all obstacle edges lie in the 4-geometry. (If obstacle edges were allowed to have arbitrary orientations this statement would become invalid.) If $k = 0$, the path is completely contained in the track graph and the lemma holds. So assume $k > 0$. Hence, there is an inclusion-maximal segment S which does not lie on the track graph. Denote by v and w the endpoints of S. Clearly, neither v nor w are in K as all valid orientations through points of K are represented in the track graph. This implies that S is adjacent to an inclusion-maximal segment $S_1 \subset P$ through v and to an inclusion-maximal segment $S_2 \subset P$ through w. Now consider the possible angles between S_1 and S and S and S_2, respectively. We first note that neither of these angles can be 45° nor 90° as otherwise P would not be a shortest path in 4-geometry. Such angles between segments of shortest paths can only occur if both segments are incident with obstacle edges. This in turn would imply that both segments lie on the track graph. See Fig. 2.

The case that both angles have 135° and S_1 and S_2 have different orientations can be excluded quite similarly. Either we can shorten P by sliding segment S in the direction of S_1 and S_2, or an obstacle vertex incident with S would prevent us from doing so. Both possibilities lead to contradictions.

The only remaining case is that both angles have 135° and S_1 and S_2 have the same orientation. We now modify our path P without changing its length such that it contains one segment less not lying in the track graph.

Denote by u the second endpoint of S_2 and let Q be the parallelogram spanned by S and S_2. We modify P by sliding segment S towards u until it either hits a track line inside Q or u itself. This modification leads to a feasible path since as soon as we hit an obstacle, we also hit a track line.

Note that u is either a terminal, or their must be a further segment S_3 of P through u which is parallel to S. If the moved segment hits a track line, the path has the same number of segments, but one less which does not lie on the track graph. Otherwise, the parallel segments S and S_3 now form a single inclusion-maximal segment. This contradicts the minimality of k. □

Lemma 2.3. *For any two vertices $p, q \in K$ there is a shortest octilinear path (in the plane, not restricted to G) which visits a sequence of vertices $p = v_0, v_1, v_2, \ldots, v_k = q$ from V_1 and for each two subsequent vertices v_i and v_{i+1}, $i = 0, \ldots, k-1$, these vertices are connected as short as possible in 4-geometry (i.e. with the same distance as if there were no obstacles).*

Proof. Let $p, q \in K$ be an arbitrary pair of terminals. By Lemma 2.2, the track graph contains a shortest path from p to q. Among all shortest paths choose P as one with the maximum number of vertices from V_1. If P is chosen in such a way, then we claim that every inclusion-maximal segment of P is incident to a vertex from V_1. We prove this by induction on the number of inclusion-maximal segments which are not incident to a vertex from V_1. Assume that S is an inclusion-maximal segment of P not incident to a vertex from V_1. This implies that S is adjacent to an inclusion-maximal segment $S_1 \subseteq P$ and an inclusion-maximal segment

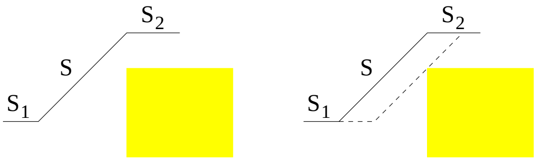

Fig. 3. We slide the segment S until it hits an obstacle or an endpoint of S_1 or S_2, respectively.

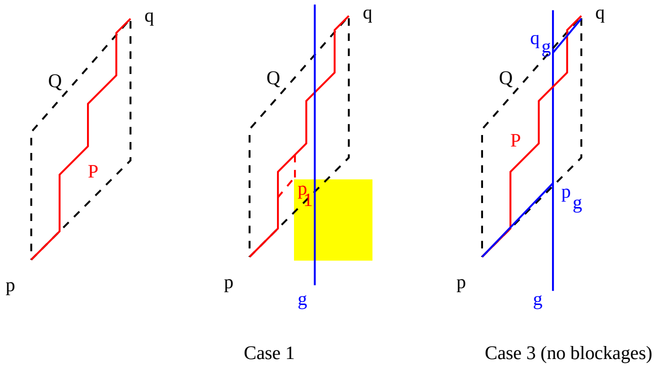

Fig. 4. Illustration of the situation in Lemma 2.4.

$S_2 \subseteq P$. If an angle between S_1 and S or between S_2 and S has either 45° or 90°, both segments must lie on the boundary of a rectangle, since otherwise the path P would not be a shortest path in 4-geometry. But if the segments lie on the boundary of a rectangle, S_1 and S or S_2 and S, respectively, meet at a common endpoint which is a vertex of an obstacle. This contradicts our assumption that no point of S belongs to V_1. The same holds if S_1 and S_2 both meet S at an angle of 135° and S_1 and S_2 have different orientations. The only remaining case is that S_1 and S_2 meet S with an angle of 135° and both segments S_1 and S_2 have the same orientation. We can slide the segment S until it hits an obstacle or an endpoint of S_1 or S_2, respectively. If S hits an obstacle, the segment that hits the obstacle is now incident to a vertex from V_1. See Fig. 3. If S hits an endpoint of S_1 or S_2, then S is now either incident to a vertex from V_1 or we have one segment less that is not incident to a vertex from V_1. This contradicts the minimality of segments which are not incident to a vertex from V_1.

Hence, for any two consecutive vertices from V_1 on P, the path bends at most once, and if it does bend, the corresponding angle is 135°. But then the distance between these vertices in P is the same as in 4-geometry. □

Lemma 2.4. *Let p and q be two vertices of V_1 such that some shortest octilinear path between p and q in the track graph does not contain any other vertex of V_1. Then the graph G contains a path from p to q of minimum length in 4-geometry.*

Proof. If two points v_i and v_{i+1} lie on a common diagonal, the shortest octilinear path between them is unique. Otherwise, we may assume that we have the following situation.

Let p and q two points in the plane with $p_x < q_x$ and $p_y + q_x - p_x < q_y$. Assume that a shortest octilinear path P from p to q goes through the parallelogram Q spanned by p, $r = (q_x, p_y + q_x - p_x)$ and q, and is x- and y-monotone. In other words: P requires no detours and consists of an alternating sequence of diagonal and vertical segments.

We use induction on the number k of vertices inside Q. (If $k = 0$, the claim trivially holds.) Let g be the vertical cut line which separates p and q.

Case 1: p is not projected in north-eastern diagonal orientation onto g.
This can only happen if some obstacle blocks the view from p to g in this direction. But then there must be some vertex $p_1 \in V$ which lies left from g, below the path P within the parallelogram Q. Choose p_1 so that it has smallest x-coordinate and among all vertices with smallest x-coordinate largest y-coordinate.

Now we can modify P so that it runs through p_1 and maintains its optimal octilinear length. After this modification the subpaths from p to q_1 and from q_1 to q contain a shortest path by induction.

Case 2: q is not projected in south-western diagonal orientation onto g.
This case is mirror-symmetric to Case 1.

Case 3: Both p and q are projected in positive diagonal orientation onto g.
Let p_g and q_g be the projections. If the vertical line from p_g to q_g is nowhere blocked, the claim holds. Otherwise, there must be again some vertex p_1 in Q which either lies below P and left from g or above P and right from g. Assume without loss of generality that we are in the first alternative. We choose p_1 as in Case 1 and proceed exactly in the same way by induction. □

Lemma 2.5. *Given a set of terminals and a set of octilinear obstacles with n vertices in total, there is a graph with $O(n \log n)$ vertices and edges which contains for every pair of terminals a shortest octilinear path.*

Proof. By Lemma 2.3, for any pair of vertices $p, q \in K$, there is a shortest octilinear path P in the plane which visits a sequence of points corresponding to vertices $p = v_0, v_1, v_2, \ldots, v_k = q$ from V_1. Any two consecutive vertices v_i and v_{i+1} are connected in P as short as possible in 4-geometry for $i = 0, \ldots, k-1$. By Lemma 2.4, the graph G contains a path from v_i to v_{i+1} of the same length as in 4-geometry for $i = 0, \ldots, k-1$. This implies that G also contains a shortest path for the whole sequence of vertices from p to q.

The size of the graph has already been analyzed when we described its construction. □

By Lemma 2.5, we can apply Mehlhorn's implementation of the spanning tree heuristic to a graph with $O(n \log n)$ vertices and edges. This immediately implies the following theorem.

Theorem 2.6. *There is a 2-approximation of the octilinear Steiner tree problem with hard octilinear obstacles. Such an approximation can be computed in $O(n \log^2 n)$ time.*

Proof. The ordering of the vertices on each cut line can be determined via sorting in $O(n \log^2 n)$. The visibility relations among the vertices can be determined in $O(n \log^2 n)$ time using a sweep line algorithm. So the graph G can be constructed in $O(n \log^2 n)$. If we apply Mehlhorn's algorithm of the spanning tree heuristic to the graph G with $O(n \log n)$ vertices and edges, we have a 2-approximation that can be computed in $O(n \log^2 n)$ time. □

3 Soft Obstacles

For soft obstacles we introduce length restrictions for those portions of a tree T which run over obstacles. Namely, for a given parameter $L \in \mathbb{R}_0^+$ we require the following for each obstacle $O \in \mathcal{O}$ and for each strictly interior connected component T_O of $(T \cap O) \setminus \partial O$: the length $\ell(T_O)$ of such a component must not be longer than the given length restriction L. Note that the intersection of a Steiner minimum tree with an obstacle may consist of more than one connected component and that our length restriction applies individually for each connected component. For ease of exposition, we restrict our presentation of soft obstacles to (axis-parallel) rectangular obstacles. Generalizations to rectilinear and octilinear soft obstacles are possible and do

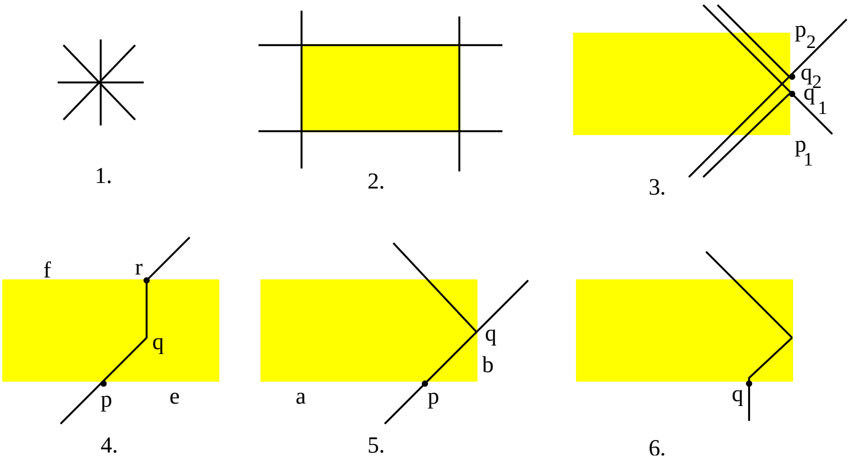

Fig. 5. The different types of tracks for soft obstacles.

not change the asymptotic size of the resulting graphs.

Track graph construction. For soft obstacles, an analogous construction of the track graph is substantially more complicated than for hard obstacles. (This is in sharp contrast to the rectilinear case). We obtain the track graph by applying the following rules inductively. See Fig. 5 for an illustration of each rule.

1. We generate track lines for all terminals and all feasible orientations. But in contrast to hard obstacles, a track does not end as soon as it hits an obstacle. It only ends at an obstacle if the intersection of the track line with the obstacle exceeds the given length restriction L. Hence, we distinguish between Steiner points which are endpoints of a track due to a length restriction, called *L-Steiner points*, and all other Steiner points generated as intersections of a track line and obstacles. The latter type of Steiner points will still be called *track-induced Steiner points*.
2. Similarly, we introduce track lines through all edges of rectangular polygons. This yields $O(n)$ track lines and may cause $O(n^2)$ many track-induced Steiner points. This already implies that the number of track-induced Steiner points will be one order of magnitude larger than for hard obstacles.
3. Additional tracks are needed to make shortcuts when an obstacle causes a deviation due to the length restriction L.
   For each edge $e = (p_1, p_2)$ of an obstacle with length $\ell(e) > L/\sqrt{2}$ we do the following. At the points q_1 and q_2 on e with distance $L/\sqrt{2}$ from the corners p_1 and p_2 we respectively generate tracks which have an angle of 45 and 135 degrees with e and run through the obstacle but do not exceed the length restriction. This yields another $O(n)$ track lines and $O(n^2)$ track-induced Steiner points.
4. Next suppose that a track tr ends at a point p of an edge e of some obstacle due to the length restriction and hits e with an angle of 45 degrees. If the edge f which is opposite to e in such a rectangle has a distance not exceeding L from e, we let the track continue inside the rectangle up to a certain point q. At q the track bends by an angle of 135 degrees and continues until it hits edge f, say at r. The point q is chosen in such a way that length of the two segments $\overline{pq}$ and $\overline{qr}$ together equals the length restriction L. Finally, at r a new track parallel to tr is created. Note that tracks generated for this item do not increase the asymptotic complexity.

5. Now consider the following situation. A track tr enters an obstacle O at some point p on edge a in an angle of 45 degrees and leaves the obstacle at some point q on an edge b of O which is adjacent to a. Furthermore, we assume that the length of b exceeds L. Then we start a new track tr_2 at q which runs orthogonally to tr through the obstacle, provided that $\ell(tr_2 \cap O) < L$ (i.e., the intersection of tr_2 with O does not exceed L; if equality holds this track has already been inserted). As a track may cross many obstacles each of which potentially induces a new track of the just described kind, and newly generated tracks in turn may induce further tracks of this kind, we have to be careful not to generate infinitely many new tracks. Therefore, the generation process is done in rounds for each track generated in Items 1-4. In each round, we create a tree of new tracks, called *track tree*. The root r of such a track tree is one of the tracks generated by Items 1-4. Every induced new track is made an immediate successor of its inducing track. A round ends if no new track is induced. To make each round finite, we add the following rule. Consider a fixed round and suppose that we have generated in step i of this round a track tr_i from a Steiner point on rectangle side e. If in a later step $j > i$ we would have to insert a further track tr_j from the very same rectangle side e due to Item 5 and this track would have track tr_i as a predecessor in the track tree, such a track is not necessary. This is because in such a scenario the generated tracks would form a full cycle around a rectangle, and clearly no cycle can be in a shortest path. Hence, our rule is not to generate a further track in such cases. By applying this rule, we have a finite number of tracks.
6. Suppose that a track tr ends at an obstacle O due to the length restriction and hits edge e of O orthogonally at some point q. Moreover, suppose q has a distance of less than $L/\sqrt{2}$ from some obstacle corner v on e. Then we add a segment and a new track to shortcut the way around O (the latter only if its intersection with O does not exceed L). See again Fig. 5. We handle such tracks as in the previous item.

This completes the construction of our track graph. In the same way as for hard obstacles we can prove the correctness of this construction.

Lemma 3.1. *The constructed track graph contains a shortest length-restricted path between every pair of terminals.*

Proof. For arbitrary terminals s and t choose a shortest octilinear path P which has the fewest number k of inclusion-maximal segments which do not lie completely in the track graph (as the track graph contains line segments, it is possible that parts of an inclusion-maximal segment of a path lie in the track graph and other parts do not). We assume further that P is selected in such a way that the total length of the segments not lying on the track graph is minimum.

If $k = 0$, the path is completely contained in the track graph and the lemma holds. So assume $k > 0$. Hence, there is a an inclusion-maximal segment S which does not lie completely on the track graph. Denote by v and w the endpoints of S. Clearly, neither v nor w are terminals as all valid orientations through terminals are represented in the track graph. This implies that S is adjacent to an inclusion-maximal segment $S_1 \subset P$ through v and an inclusion-maximal segment $S_2 \subset P$ through w. We may assume that S is the first segment in the order of segments from s to t on path P which does not lie on the track graph. Hence, at least one of the adjacent segments belongs to the track graph, say S_1.

Now consider the possible angles between S_1 and S and S_2 and S, respectively. We first note that neither of these angles can be 45°, as otherwise P would not be a shortest path in 4-geometry. Furthermore, if one of these angles is 90°, the meeting point of the corresponding segments lies on a side a of the obstacle and the segments lie inside the obstacle as otherwise P would not be a shortest path in 4-geometry. Moreover, such an angle can only occur if the side a is longer than the length restriction L. Since S_1 is in the track graph and so S would be

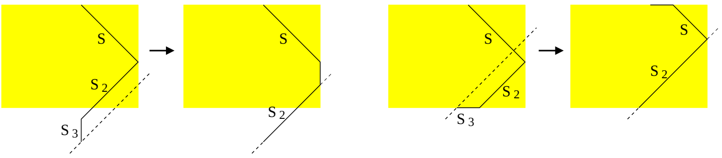

Fig. 6. Two cases where we can reduce the number of segments which do not lie on the track graph.

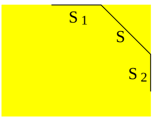

Fig. 7. This figure shows a scenario where segment S belongs to the track graph.

also in the track graph due to Item 5 or 6 of the construction, the 90° angle must be between S and S_2. If the segment S_2 is contained in the track graph then again due to Item 5 or 6 in the construction of the track graph, the segment S must be contained in the track graph, too. So assume that both S and S_2 are not contained in the track graph. Clearly the endpoints of S_2 are not terminals. Therefore, S_2 is adjacent to another inclusion-maximal segment S_3 which must meet S_2 at 135°. For both possible directions we can slide the segment S_2 to the nearest parallel edge of the track graph without violating any length restriction and resulting in a path with one segment less which does not lie on the track graph. See also Fig. 6. This contradicts the minimality of k.

The case that both angles have 135° and S_1 and S_2 have different orientations can only occur if S_1 and S_2 both lie on the boundary of an obstacle. Since P is a shortest octilinear path, the segment S must have length L and lies on an edge of the track graph by Item 3 of the construction of the track graph. See Fig. 7. This contradicts that S is not an element of the track graph.

The only remaining case is that both angles have 135° and S_1 and S_2 have the same orientation. We now modify our path P without changing its length such that it either (1) contains one segment less not lying in the track graph, or (2) reduces at least the total length of segments of P which do not lie on the track graph.

Denote by u the second endpoint of S_2. We modify P by sliding segment S towards u until it either hits an edge of the track graph parallel to S or u itself. See Fig. 8 for an illustration. Each of the four obstacles A, B, C, D in this example potentially induce a track graph edge parallel to S; obstacle A by Item 4, obstacle B by Item 3, and obstacles C and D by Item 2 of the track graph construction.

Obviously, the modified path has the same length and the parts not lying on the track graph have been reduced. Thus, we have a contradiction if we can show that the modified path respects the length restriction.

We may assume that the path P does not bend more than once inside an obstacle (as we can always bring the path into such a form without increasing its length inside any obstacle).

Suppose that v lies inside some obstacle O. Then the local configuration around v must look like in one of the three possibilities shown in Fig. 9. In case (a), we can slide S towards the dashed line without violating the length restriction by Item 4 of the track graph construction. In case (b) or (c), either the modified path is length feasible, or the pointed line (which is then part of the track graph by Item 3) can be used as a feasible alternative.

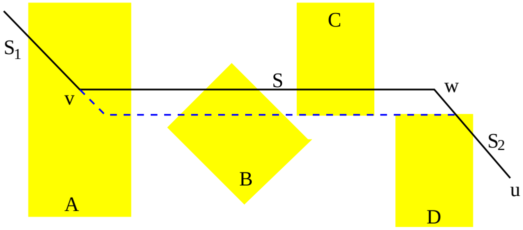

Fig. 8. The segment S is slided to the dashed line. In this example, we assume that there is no track edge parallel to S inside the parallelogram spanned by S and the dashed line.

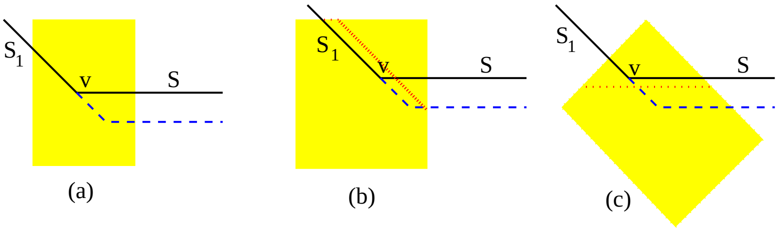

Fig. 9. Local configurations around v.

For all other obstacles which might be intersected with the path before or after the modification, no length violation can occur by our choice of how far we slide segment S (see once more Fig. 8). □

Approximate shortest paths. The track graph as described above may have exponential size. With a smarter construction one can bound the size of the track graph by $O(n^3)$ (but the proof then becomes quite complicated). In this paper, we therefore prefer a simpler construction which uses approximate shortest paths. For any integer k, we obtain $(1 + \frac{1}{k})$–approximate shortest paths. The idea is to leave out Items 5 and 6 of the track graph construction (which are responsible for the blow up in the graph size). Instead, we insert $k-1$ additional tracks for each corner of an obstacle. These tracks "cut off" the corner and are placed in distance $\frac{j \cdot L}{\sqrt{2} \cdot k}$ from the corner for $j = 1, \ldots, k-1$. See Fig. 10.

This construction induces $O(kn)$ many new tracks which are responsible for $O(kn)$ new track-induced Steiner points per obstacle. Next we apply the same sparsification technique as for hard obstacles and make sure that every path in our graph is feasible with respect to our length restriction. We do this in two steps. In the first step, we use the modified cut line approach from Section 2 on the set of original vertices, terminals and all track-induced Steiner points on the boundary of obstacles. In this step we treat all obstacles as hard obstacles (i.e., projection onto some cut line is only possible from some visible point). The overall number of track-induced Steiner points is $O(kn^2)$. Hence, the sparsification technique outside obstacles yields $O(kn^2 \log(kn))$ many vertices and edges.

In the second step, we add connections between vertices and Steiner points on the boundary of obstacles. In the previous discussion we observed that we may have $O(kn)$ many track-induced Steiner points lying on the boundary of an obstacle O. Locally these Steiner points can be regarded as terminals which have to be connected pairwise without violating the length bound L.

Lemma 3.2. *Let O be a rectilinear obstacle with t terminals on its boundary. Then we need $O(t^2)$ many edges for a graph which has (1) to represent shortest paths between any pair of*

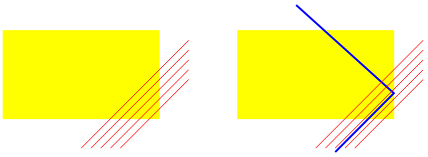

Fig. 10. The extra tracks inserted at a corner of some obstacle to approximate shortest paths (left). Clearly, the "thick" path is only slightly longer than an approximation using one of the extra tracks (right).

terminals respecting the length restriction L, and (2) does not contain any path exceeding the length restriction L inside some obstacle.

Proof. For every pair of terminals, we add an edge if and only if their octilinear distance is less or equal to L. □

Thus, we can now apply Lemma 3.2 with $t = O(kn)$ and get $O(k^2n^2)$ edges inside a single obstacle, for a total of $O(k^2n^3)$ edges inside all obstacles.

It is easy to see that shortest paths between terminals in this modified graph will be at most a factor of $(1 + \frac{1}{k})$ longer than shortest paths.

Lemma 3.3. *There is a graph for soft rectangular obstacles with $O(kn^2 \log(kn))$ many vertices and $O(k^2n^3)$ many edges which contains a $(1+\frac{1}{k})$–approximative shortest path between any pair of terminals for any integer k. Moreover, all paths in this graph respect the length restriction L inside obstacles. The graph can be constructed in time proportional to its size.*

Proof. The size of the graph follows directly from our explanations in the main section. It is also clear that it can be constructed in the same time.

Thus it remains to prove the approximation guarantee. Denote the track graph according to Item 1-6 by G_1. By Lemma 3.1, the constructed track graph contains a shortest length-restricted path between every pair of terminals. We analyze next the effect of inserting $k - 1$ additional tracks for each corner instead of applying Item 5 and 6 of the track graph construction. Denote the track graph after this modification by G_2. For two arbitrary terminals s and t, consider a shortest length-restricted path P in G_1 from s to t. We may assume that this path P has the fewest number of bends among all shortest paths between s and t.

We embed this path in G_2 as follows. All segments of P which are also in G_2 remain unchanged. All remaining segments are embedded successively. As long as the segments in G_2 are not connected, denote by S_1 the last inclusion-maximal segment of P (in the given orientation from s to t) which is also represented in G_2, and by S_2 the first inclusion-maximal segment not represented in G_2. Then, the common point of S_1 and S_2 must be a boundary point p of some obstacle O, and S_2 must lie on a track line introduced by Item 5 or 6. Let P_2 denote the subpath of P which starts at p with segment S_2 and ends at a point q which is chosen as follows. The point q is the first point on P when traversing the path from p towards t where P enters an edge which also belongs to G_2 or where it bends once more and enters another track inserted by Item 5 or 6 in the track graph construction. (Clearly, such a q exists since the last bending point before arriving at t is always a candidate.)

This subpath P_2 is replaced by a sequence of track lines $S'_1, S'_2, \ldots, S'_t$ in G_2 which are parallel to and at most a distance of $a < \frac{L}{\sqrt{2} \cdot k}$ away from corresponding segments of P_2. These segments are linked to the rest of P by at most two short segments of length a on the boundary of obstacles as shown in Fig. 11. Note that the replacement exists, i.e., none of the necessary track lines $S'_1, S'_2, \ldots, S'_t$ is stopped because of a length violation inside some obstacle. This is

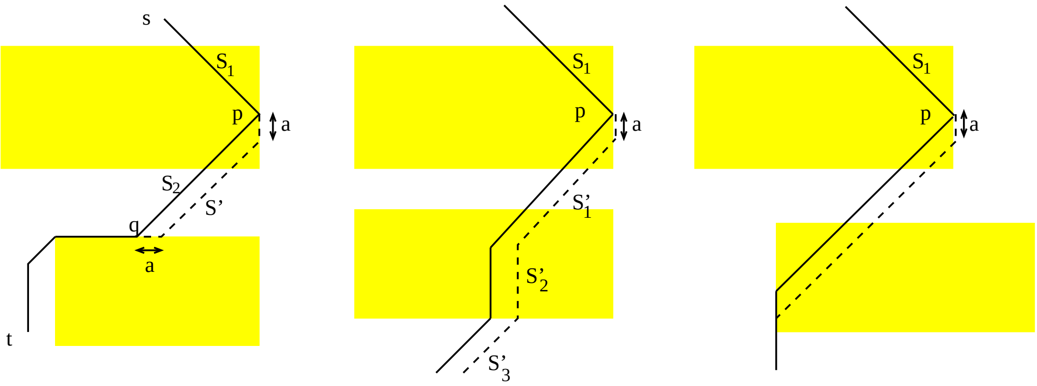

Fig. 11. Approximation of a path in the track graph G_1. Parts of the original solid path not in G_1 are replaced by the dashed path which belongs to G_2.

true since the intersection of some S'_i with some obstacle cannot exceed the length restriction L as otherwise either the corresponding segment of P would already have been infeasible or some other track line would be nearer to P contradicting the choice of our replacement (the right case in Fig. 11).

Suppose, we have to apply such a modification m times. In each case, the original path goes through an obstacle O and the intersection of the path with the obstacle must be at least

$$\ell(P \cap O) \geq \sqrt{2}L,$$

as otherwise no track for Item 5 or 6 would have been inserted. Hence, the length of P is lower bounded by $\sqrt{2}Lm$. The length of a is certainly smaller than the distance between two inserted additional track lines, hence $a \leq \frac{L}{\sqrt{2} \cdot k}$ and $\ell(S') \leq \ell(S_2)$ for each modified segment S'.

Thus the modified path P' satisfies

$$\begin{aligned} \ell(P') &\leq \ell(P) + \frac{2Lm}{\sqrt{2} \cdot k} \\ &\leq \ell(P) + \frac{\ell(P)}{k} \\ &\leq (1 + \frac{1}{k}) \cdot \ell(P). \end{aligned}$$

We finally note that the sparsification technique applied to G_2 does not further change path lengths. In the same way as we proved the correctness for the sparsification technique for hard obstacles, one can show that the distance between any two track-induced Steiner points remains unchanged by the sparsification. Details are left to the reader.

The final graph contains only paths which respect the length restriction inside obstacles since we deleted for each obstacle O the whole subgraph of G_2 with edges and vertices “inside” O, and replaced these subgraphs by length-feasible direct connections between points on the boundary. □

As the obtained graph contains only length-feasible paths, we can apply Mehlhorn’s implementation of the minimum spanning tree heuristic to construct a Steiner tree. We finally obtain:

Theorem 3.4. *For any fixed k, we can find a $(2 + \frac{1}{k})$-approximation of the octilinear Steiner tree problem with soft rectangular obstacles in time $O(n^3)$.*

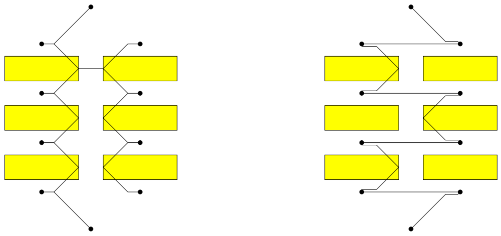

Fig. 12. An example of our instance class for which the analysis is tight. We show an octilinear Steiner tree T (left) and a minimum spanning tree T_{OPT}^{span} based on shortest length-restricted paths.

We conclude this section by mentioning that our analysis is tight. It is possible to construct a class of instances for which our approximation algorithm asymptotically achieves a performance guarantee of 2.

Lemma 3.5. *There is a class of instances for which the minimum spanning tree based approximation of the octilinear Steiner tree problem subject to soft rectangular obstacles asymptotically achieves a performance guarantee of* 2.

Proof. We construct a class of instances for the octilinear Steiner tree problem with soft obstacles as follows.

Let L be the parameter for the length restriction. We embed $k = 2r$ terminals $p_1, p_2, \ldots, p_k$ to the following positions in the plane: $p_1 = (0, (L+3)/2 - 1)$, $p_{2i} = (-rL, iL + 3i)$, $p_{2i+1} = (rL, iL + 3i)$ for $i = 1, \ldots, r-1$, and $p_k = (0, rL + 3r - (L+3)/2 + 1)$. We add $k - 4$ rectangular obstacles $O_1, O_2, \ldots, O_{k-4}$. The left lower and the right upper corner of O_i are denoted by lc_i and rc_i, respectively. There coordinates are given by $lc_{2i-1} = (-2rL, iL+3i+1)$, $rc_{2i-1} = (-1, iL + 3i + L + 2)$, $lc_{2i} = (1, iL + 3i + 1)$, and $rc_{2i} = (2rL, iL + 3i + L + 2)$ for $i = 1, 2, \ldots, r-2$. See also Fig. 12 for an illustration with $r = 5$.

The length of the feasible octilinear Steiner tree T as shown in Fig. 12 is given by

$$l(T) = 2r^2L + (2\sqrt{2} - 3)rL + (6\sqrt{2} - 5)r + (2 - 3\sqrt{2})L + 12 - 7\sqrt{2}.$$

The length of a minimum spanning tree T_{OPT}^{span} based on shortest length-restricted paths is

$$l(T_{OPT}^{span}) = 4r^2L + (\sqrt{2} - 6)rL + (3\sqrt{2} - 10)r + (2 - \sqrt{2})L + 10 - \sqrt{2}.$$

As the optimum length-restricted Steiner tree T_{OPT} is not longer then $\ell(T)$, we obtain

$$2 \geq \lim_{r\to\infty} \frac{l(T_{OPT}^{span})}{l(T_{OPT})} \geq \lim_{r\to\infty} \frac{l(T_{OPT}^{span})}{l(T)} = 2.$$

Thus, we can conclude that the approximation guarantee is asymptotically tight. □

4 Improved Approximation Guarantee

In this section we show how to construct a graph of polynomial size which contains a $(1+\varepsilon)$-approximation for the octilinear Steiner tree problem with soft rectangular obstacles. The graph construction requires the following five steps:

Step 1: The very first step is to compute an axis-parallel box which contains an optimal Steiner tree. Everything outside such a box can then be safely ignored in the subsequent steps. For the later analysis it is important that the side length b of this box can be bounded by a constant times the length of an optimal Steiner tree T_{opt}.

To achieve this goal, we can run the minimum spanning tree based approximation. Let us assume that this approximation yields a tree of length $\ell(T_{MST})$. Denote by $BB(K)$ the bounding box of the given terminal set, that is, the smallest axis-parallel rectangle which includes all terminals. Let bb be the maximal side length of $BB(K)$. Now we can define $b := bb + 2\ell(T_{MST})$. Clearly, an axis-parallel box B of side length b centered at the barycenter of $BB(K)$ is large enough to contain an optimal Steiner tree. Since the minimum spanning tree yields a 2-approximation and $bb \leq \ell(T_{opt})$, we also have

$$b \leq 5 \cdot \ell(T_{opt}). \tag{1}$$

Step 2: We build a refinement of a Hanan-like grid graph restricted to the area of B. This refinement is parameterized by some parameter k (to be determined later). More specifically, we subdivide the boundary of box B equidistantly with k points into $k+1$ segments and add for each subdivision point additional lines in all four feasible orientations of the octilinear geometry. To this set of lines we add lines through each terminal and each vertex of an obstacle in all feasible directions. Let G be the graph induced by intersections of these lines restricted to the area inside B (including the boundary of B).

Step 3: The resulting graph may allow subtrees inside obstacles which violate the length restriction L. Therefore, we delete all nodes and edges which lie strictly inside some obstacle.

Step 4: Let $t \in \mathbb{N}$ be another parameter which will be chosen as a constant depending on ε but independent from the given instance. For each obstacle O and for each subset S of at most t vertices on the boundary of O compute an optimal Steiner tree for S which respects the length restriction L inside O. We add each such Steiner tree to the current graph and identify common boundary vertices. Since t is a constant, there is only a polynomial number of these small Steiner tree instances and each of these trees can be computed in constant time as we will show later.

Step 5: Finally, we want that our graph contains a feasible almost shortest octilinear path between any pair of vertices on the boundary of obstacles. More specifically, we require that these paths approximate the true shortest paths by a factor of $1 + 1/(k+1)$. We can compute these paths and their lengths by the methods from Section 3 and add them to the graph.

On the resulting graph $G = G(k,t)$, parameterized by k and t, we can then solve the Steiner tree problem for the given terminal set K.

The parameter t will be chosen as a constant and the parameter $k = O(n)$. This immediately implies that the constructed graph has polynomial size.

It remains to show that Step 4 can be done efficiently. Therefore, we have to show the following lemma.

Lemma 4.1. *Let S be a set of at most t terminals on the boundary of some rectangle O and L be some length restriction inside O. If t is a constant, then the octilinear Steiner tree problem for S with length restriction L can be solved in constant time.*

The proof of this theorem requires some well-known properties concerning the structure of optimal octilinear trees, see for example [LS96].

Property 4.2 ([LS96]). There exists an octilinear Steiner minimum tree T_{opt} such that the degree of any Steiner point is either three or four. There exists a Steiner minimum tree such that every degree-4 Steiner point is adjacent to four terminals which form a cross (i.e., all four

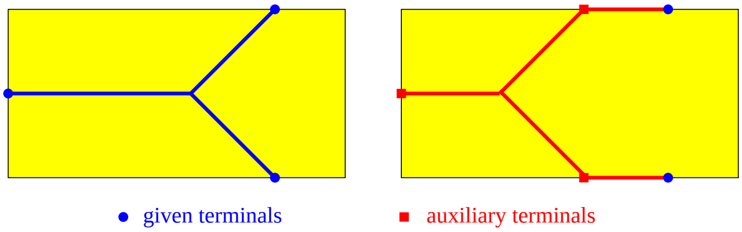

Fig. 13. In this example, the unrestricted optimal Steiner tree is shown on the left, whereas the optimal solution subject to a length restriction is shown on the right.

angles around a degree-4 Steiner point are $\frac{\pi}{2}$). Furthermore, the three angles around a degree-3 Steiner point of T_{opt} are $\frac{\pi}{2}$, $\frac{3\pi}{4}$, $\frac{3\pi}{4}$ (in some order).

The *topology* of a Steiner tree merely refers to the graph structure, i.e., it includes the terminals and Steiner points as vertices and specifies the connections between these vertices as edges. However, the topology does not include the geometric embedding in the plane.

Proof of Lemma 4.1. For a given set S of terminals we first compute an optimal octilinear Steiner tree *without* considering the length restriction L.

To this end, we simply enumerate over all possible tree topologies and finally take the shortest tree. Since every tree can be decomposed into its full components, we restrict our attention only to full trees. By Property 4.2 the possible tree topologies are restricted. Their number is obviously finite. Brazil et al. [BTWZ02] have shown that, for each given topology one can construct a Steiner minimum tree (that means find an optimal embedding) in linear time in t.

If the optimum tree for S also satisfies the length restriction L, we are done. However, if this tree exceeds the length restriction and is therefore infeasible, we need some more work.

In such a case, the optimal feasible tree is composed by one or more full trees of *exactly length* L inside the obstacle O and some segments on its boundary which connect the full tree with the given terminals. See Fig. 13 for a small example.

The precise position of the full tree can be computed by linear programming. Again we restrict our attention to the case of a single full component inside O.

An octilinear (L, k)*-tree in a rectangular obstacle* O is a full Steiner tree of length L with k terminals which are located on the obstacle's boundary.

Assume that $S = \{t^{(1)}, t^{(2)}, \ldots, t^{(k)}\}$ are the given terminals, for $k \leq t$. For each given terminal $t^{(i)}$, we associate an auxiliary terminal $t'^{(i)}$ (this mapping is, in general, not injective. Two given terminals may be mapped to the same auxiliary terminal). These auxiliary terminals shall be the terminals of an (L, k)–tree. The coordinates of these auxiliary terminals have to be determined so as to minimize the segment lengths on the boundary.

Let us fix the tree topology including the orientation of its edges of an (L, k)–tree. We also fix the counterclockwise order of given terminals and auxiliary terminals around the boundary of O and their assignment to the four rectangle sides of the obstacle O.

Our objective is to minimize the overall length of the segments connecting the auxiliary terminals with the given terminals. This is a linear function in the unknown coordinates $t'^{(i)}_x, t'^{(i)}_y$ (for $1 \leq i \leq k$) subject to several linear side constraints. Assume that the origin of our coordinate system is the left bottom corner of the rectangle O.

We require that

- for a rectangular obstacle of dimension $a \times b$, all vertical coordinates are in the range $[0, a]$ and all horizontal coordinates are in the range $[0, b]$.
- The length of the full Steiner tree is exactly L. The length of an (L, k)–tree T can be expressed as a function of the coordinates of its terminals and Steiner points. Let us consider all tree edges $e = (u, v)$ as directed edges, namely horizontal edges as directed from left to right, all vertical edges directed from top to bottom and all diagonals from left to right. Then we obtain the linear equality

$$L \equiv \sum_{e \in E(T)} ||e|| = \sum_{\substack{e \in E(T)\\ \text{vertical}}} ||e|| + \sum_{\substack{e \in E(T)\\ \text{horizontal}}} ||e|| + \sum_{\substack{e \in E(T)\\ \text{diagonal}}} ||e||$$

$$= \sum_{\substack{e=(u,v) \in E(T)\\ \text{vertical}}} (u_x - v_x) + \sum_{\substack{e=(u,v) \in E(T)\\ \text{horizontal}}} (v_y - u_y) + \sqrt{2} \cdot \sum_{\substack{e=(u,v) \in E(T)\\ \text{vertical}}} (v_y - u_y).$$

- All tree edges have nonnegative length. This gives one linear inequality for the coordinates of each edge.
- The distance between pairs of auxiliary terminals on opposite sides of the tree must be exactly the corresponding side length of the rectangle. For each such pair we obtain a linear equality.
- The given ordering of auxiliary terminals and terminals is not violated. This gives one or two additional linear inequalities per auxiliary terminal.

Hence, finding an optimal embedding of an (L, k)–tree for a fixed topology amounts to solving a linear programming problem of constant dimension. As these can be solved in constant time, our lemma follows by enumerating over all possible tree topologies. □

4.1 Analysis of the Approximation

For the analysis, we fix some optimal Steiner tree T_{opt}. To bound the approximation achieved by our graph G, we partition T_{opt} into several parts which are analyzed independently. To this end we define how to *cover a Steiner tree T by a set of axis-parallel rectangles.* This set $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$ is obtained as follows. Denote by R_1 the set of obstacles which include at least one Steiner point of T_{opt} in its interior. For each Steiner point s of T not covered by an obstacle, the set $\mathcal{R}_2$ contains a smallest rectangle including s with horizontal and vertical edges from G. In the degenerate case that s lies on a vertex or an edge of G we add no rectangle. We also add a smallest enclosing rectangle for each point p where an edge of T bends. Degenerate cases are handled as with Steiner points. For each straight-line segment of T not covered by previous rectangles we independently add to $\mathcal{R}_2$ a smallest enclosing rectangle bounded by vertical and horizontal edges from G. Thus, we finally have the following partition of the Steiner tree: $T = \cup_{R \in \mathcal{R}} (T \cap R)$.

The constructed graph G contains an approximative tree T_{app} which we obtain as follows. The general idea is to replace portions of the optimal tree by trees contained in G. From the union of all these trees we eliminate in a postprocessing step the longest edge of each cycle which may occur and all leaves and incident edges of the resulting tree which are not terminals.

Replacing portions covered by $\mathcal{R}_1$. Let T_R be some inclusion-maximal connected component of T_{opt} which lies strictly inside $R \in \mathcal{R}_1$ except for a finite set of points on the boundary of R. Denote by K_R this set of boundary points of T_R.

In our graph G, the boundary of each obstacle within the box B has been discretized. For any point p on the boundary there is a point in the discretized set with distance at most

$$\Delta \leq \frac{b}{k+1} \leq \frac{5 \cdot \ell(T_{opt})}{k+1} \tag{2}$$

units from p. Denote by K'_R the set of vertices in G such that for every point in K_R there is one in K'_R with distance at most Δ. Then, an optimal Steiner tree T'_R for the set K'_R satisfies

$$\ell(T'_R) \leq \ell(T_R) + |K_R| \cdot \Delta.$$

If $|K'_R| > t$, the tree T'_R may not be contained in G. However, since we have included in G optimal trees for any t-element subset of boundary vertices, we have an approximation by t-restricted Steiner trees available.

The exact approximation ratio of t-restricted Steiner trees under the octilinear metric and length restrictions has not yet been determined. However, this ratio cannot be worse than the ratio for t-restricted Steiner trees in graphs. For the latter, it is known that the ratio is $r_t = \frac{(r+1)2^r+\ell}{r2^r+\ell}$ for $t = 2^r + \ell$ [BD95]. Obviously, $r_t \geq 1$ is monotonously decreasing and converges to 1 for large t. Hence, we may choose t such that $r_t - 1 \leq \frac{\varepsilon}{2}$ for any given $\varepsilon > 0$.

For each $R \in \mathcal{R}_1$ we have

$$\ell(T_{app} \cap R) - \ell(T_{opt} \cap R) \leq (r_t - 1) \cdot \ell(T_{opt} \cap R) + r_t \cdot |K_R| \cdot \Delta.$$

Since $\sum_{R \in \mathcal{R}_1} |K_R| \leq 3n - 6$ (as we have at most $n-2$ rectangles in $\mathcal{R}_1$ and each Steiner point has 3 incident edges which may contribute to some K_R) and clearly $r_t \leq 2$, we obtain by (2)

$$\sum_{R \in \mathcal{R}_1} (\ell(T_{app} \cap R) - \ell(T_{opt} \cap R)) \leq \frac{\varepsilon}{2} \cdot \ell(T_{opt}) + \frac{10(3n-6)}{k+1} \cdot \ell(T_{opt}).$$

Replacing portions covered by $\mathcal{R}_2$. For rectangles in $\mathcal{R}_2$ the following technical lemma was shown in [MS05].

Lemma 4.3 ([MS05]). *For each $R \in \mathcal{R}_2$, the following bound holds:*

$$\ell(T_{app} \cap R) - \ell(T \cap R) \leq (4 - \sqrt{2})\frac{b}{k+1}.$$

In the presence of (soft) obstacles, edges between terminals and/or Steiner points may be forced to bend several times. Hence, in general, an edge $e = (p, q)$ consists of a certain number of straight line segments, say $s_1, s_2, \ldots s_{w-1}, s_w$, and hits a number of obstacles. Let p_1 be the first common point of such an edge with some obstacle and p_2 be the last, respectively. Denote by p'_1 and p'_2 the nearest points in our graph G belonging to the same obstacle as p_1 and p_2, respectively. Since we have an almost shortest path between p'_1 and p'_2 in our graph (by step 5 of the graph construction), the path from p_1 to p_2 can be approximated by taking two short segments of length at most Δ on the boundary of the first and last obstacle plus this almost shortest path. Thus, this path is at most $2\Delta + \ell(T_{opt})/(k+1) \leq 11\ell(T_{opt})/(k+1)$ longer than the corresponding one in T_{opt}. The very first segment s_1 and the one or two last segments s_{w-1} and s_w plus the possible corner point between s_{w-1} and s_w are covered by up to four rectangles from $\mathcal{R}_2$. Hence by Lemma 4.3, the total error contributed by a single edge e is upper bounded by

$$4 \cdot (4 - \sqrt{2})\frac{b}{k+1} + \frac{11\ell(T_{opt})}{k+1} \leq \frac{71\ell(T_{opt})}{k+1}.$$

As there are at most $2n-3$ edges in total, the overall error contributed by edges which are covered by rectangles in $\mathcal{R}_2$ is upper bounded by $\frac{71(2n-3)\ell(T_{opt})}{k+1}$. There are at most $n-2$ Steiner points covered by rectangles in $\mathcal{R}_2$. These may contribute an additional error of $\frac{15(n-2)\ell(T_{opt})}{k+1}$.

Summing up, the total error can be bounded by

$$\ell(T_{app}) - \ell(T_{opt}) \leq \frac{\varepsilon}{2}\ell(T_{opt}) + \frac{10(3n-6)}{k+1}\ell(T_{opt}) + \frac{71(2n-3)\ell(T_{opt})}{k+1} + \frac{15(n-2)\ell(T_{opt})}{k+1},$$

which simplifies to

$$\ell(T_{app}) - \ell(T_{opt}) \leq \frac{\varepsilon}{2}\ell(T_{opt}) + \frac{(187n-303)\cdot\ell(T_{opt})}{k+1}.$$

If we choose $k := \lceil \frac{2\cdot(187n-303)}{\varepsilon} \rceil$, our graph contains a $(1+\varepsilon)$–approximation and has polynomial size. Thus, we have shown the following theorem.

Theorem 4.4. *Let α denote the approximation guarantee for an algorithm solving the Steiner tree problem in graphs. Given a terminal set K, a set of rectangular soft obstacles O with length restriction L, and some $\varepsilon > 0$, there is an $(\alpha+\varepsilon)$-approximation of the octilinear Steiner tree problem with length restriction L inside obstacles.*

5 Conclusion and Future Work

In this paper we have studied approximation algorithms for the octilinear Steiner tree problem in the presence of hard and soft obstacles.

We have shown how to construct relatively small graphs which are guaranteed to contain shortest paths between any pair of terminals. This construction leads to efficient 2-approximations by means of the minimum spanning tree heuristic.

Less practical, but of theoretical importance is our main result. Namely, the best approximation bound we can prove comes arbitrarily close to the best available approximation algorithm for the Steiner tree problem in graphs which achieves an approximation guarantee of $\alpha = 1.55$.

Our asymptotically best approximations of the octilinear Steiner tree problem uses t-restricted Steiner trees. To achieve a polynomial running time, it was sufficient that the t-restricted Steiner ratio converges to 1 for large t. However, the convergence rate for the Steiner ratio in general graphs (which we used) is very slow. We conjecture that the true convergence should be much faster. For comparison, we note that the t-restricted Steiner ratio in the rectilinear plane is $\frac{2t}{2t-1}$ for $t \geq 4$ [BR94,BDGW98]. Hence, it is an interesting open problem to find tighter or even exact bounds on the t-restricted Steiner ratio for octilinear Steiner trees.

As a major challenge it remains to find and to analyze an approximation algorithm for the octilinear Steiner tree problem subject to obstacles which beats the approximation bound α for the Steiner tree problem in general graphs.

Acknowledgment

The first author was partially supported by the DFG Focus Program 1126 “Algorithmic Aspects of Large and Complex Networks”, grant Mu1482/2-2.

References

[APD03] E. Althaus, T. Polzin, and S.V. Daneshmand. Improving linear programming approaches for the Steiner tree problem. Research Report MPI-I-2003-1-004, Max-Planck-Institut für Informatik, Saarbrücken, Germany, March 2003.

[Aro98] S. Arora. Polynomial time approximation schemes for the Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45:753–782, 1998.

[BD95] A. Borchers and D.-Z. Du. The k-Steiner ratio in graphs. In *STOC '95: Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pages 641–649, New York, NY, USA, 1995. ACM Press.

[BDGW98] A. Borchers, D.-Z. Du, B. Gao, and P. Wan. The k-Steiner ratio in the rectilinear plane. *Journal of Algorithms*, 29:1–17, 1998.

[BR94] P. Berman and V. Ramaiyer. Improved approximations for the Steiner tree problem. *Journal of Algorithms*, 17:381–408, 1994.

[BTWZ02] M. Brazil, D.A. Thomas, J.F. Weng, and M. Zachariasen. Canonical forms and algorithms for Steiner trees in uniform orientation metrics. Technical Report TR-02/22, DIKU, Department of Computer Science, Copenhagen, Denmark, 2002.

[CCK$^+$03] H. Chen, C.-K. Cheng, A. B. Kahng, I. Măndoiu, and Q. Wang. Estimation of wire-length reduction for λ-geoemtry vs. Manhattan placement and routing. In *Proceedings of SLIP'03*, pages 71–76. ACM Press, 2003.

[CKV87] K.L. Clarkson, S. Kapoor, and P.M. Vaidya. Rectilinear shortest paths through polygonal obstacles in $O(n(\log n)^2)$ time. In *Proceedings of the 3rd Annual ACM Symposium on Computational Geometry*, pages 251–257, 1987.

[Cou03] C. Coulston. Constructing exact octagonal Steiner minimal trees. In *ACM Great Lakes Symposium on VLSI*, pages 1–6, 2003.

[DH92] D.-Z. Du and F.K. Hwang. Reducing the Steiner problem in a normed space. *SIAM Journal on Computing*, 21(6):1001–1007, 1992.

[GGJ77] M.R. Garey, R.L. Graham, and D.S. Johnson. The complexity of computing Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 32:835–859, 1977.

[GJ77] M.R. Garey and D.S. Johnson. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32:826–834, 1977.

[KMZ03] A.B. Kahng, I.I. Măndoiu, and A.Z. Zelikovsky. Highly scalable algorithms for rectilinear and octilinear Steiner trees. *Proceedings 2003 Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 827–833, 2003.

[LS96] D.T. Lee and C.-F. Shen. The Steiner minimal tree problem in the λ-geometry plane. In *Proceedings 7th International Symposium on Algorithms and Computations (ISAAC 1996)*, volume 1178 of *Lecture Notes in Computer Science*, pages 247–255. Springer, 1996.

[LX00] G.-H. Lin and G. Xue. Reducing the Steiner problem in four uniform orientations. *Networks*, 35(4):287–301, 2000.

[LYW96] D.T. Lee, C.D. Yang, and C.K. Wong. Rectilinear paths among rectilinear obstacles. *Discrete Applied Mathematics*, 70:185–215, 1996.

[Meh88] K. Mehlhorn. A faster approximation algorithm for the Steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.

[Mit99] J.S.B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[Mit00] J.S.B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier, 2000.

[MP03] M. Müller-Hannemann and S. Peyer. Approximation of rectilinear Steiner trees with length restrictions on obstacles. In *8th Workshop on Algorithms and Data Structures (WADS 2003)*, volume 2748 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2003.

[MS05] M. Müller-Hannemann and A. Schulze. Hardness and approximation of octilinear Steiner trees. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC 2005), Hainan, China*, volume 3827 of *Lecture Notes in Computer Science*, pages

256–265. Springer, 2005. Journal version: http://www.algo.informatik.tu-darmstadt.de/muellerh/hardness_approx.pdf.

[NWZ02] B.K. Nielsen, P. Winter, and M. Zachariasen. An exact algorithm for the uniformly-oriented Steiner tree problem. In *10th Annual European Symposium on Algorithms (ESA 2002)*, volume 2461 of *Lecture Notes in Computer Science*, pages 760–772. Springer, 2002.

[PWZ04] M. Paluszewski, P. Winter, and M. Zachariasen. A new paradigm for general architecture routing. *Proceedings of the 14th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, pages 202–207, 2004.

[RZ00] G. Robins and A. Zelikovsky. Improved Steiner tree approximation in graphs. *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.

[Tei02] S. L. Teig. The X architecture: not your father's diagonal wiring. In *SLIP '02: Proceedings of the 2002 international workshop on System-level interconnect prediction*, pages 33–37. ACM Press, 2002.

[WWSW87] Y.F. Wu, P. Widmayer, M.D.F. Schlag, and C.K. Wong. Rectilinear shortest paths and minimum spanning trees in the presence of rectilinear obstacles. *IEEE Transactions on Computing*, pages 321–331, 1987.

[X] http://www.xinitiative.org.

[ZZJ$^+$04] Q. Zhu, H. Zhou, T. Jing, X. Hong, and Y. Yang. Efficient octilinear Steiner tree construction based on spanning graphs. *Proceedings 2004 Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 687–690, 2004.