In Search of the Boundaries of Intractability for Euclidean Degree-$k$ MST Problems

Semester Thesis
Andrea Francke
September 2008 – April 2009

Advisor: Dr. Michael Hoffmann
Department of Computer Science, ETH Zürich
Abstract

This report documents the semester thesis carried out by Andrea Francke under supervision by Michael Hoffmann at the Institute for Theoretical Computer Science, ETH Zürich, from September 2008 to April 2009. The main result of the thesis is a proof for NP-hardness of the Euclidean degree-4 minimum spanning tree problem. Additionally to the proof, which can also be found in [FH09], the main contents of this report are sections on related problems, on other attempts at the proof and on open questions, as well as an appendix giving the coordinates of a vertex gadget for an analogous proof for the Euclidean degree-3 minimum spanning tree problem.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contents</td>
<td>iii</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2 The Problem</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Problem Definition</td>
<td>3</td>
</tr>
<tr>
<td>2.2 Related Work</td>
<td>4</td>
</tr>
<tr>
<td>3 The Proof</td>
<td>9</td>
</tr>
<tr>
<td>3.1 The Reduction</td>
<td>9</td>
</tr>
<tr>
<td>3.2 The Gadgets</td>
<td>11</td>
</tr>
<tr>
<td>3.3 Proof of Equivalence</td>
<td>18</td>
</tr>
<tr>
<td>4 Other Approaches Taken</td>
<td>23</td>
</tr>
<tr>
<td>5 Open Questions</td>
<td>31</td>
</tr>
<tr>
<td>6 Conclusion</td>
<td>35</td>
</tr>
<tr>
<td>A Task Description</td>
<td>37</td>
</tr>
<tr>
<td>B Additional Figures and Tables</td>
<td>39</td>
</tr>
<tr>
<td>C A Vertex Gadget for $k = 3$</td>
<td>45</td>
</tr>
<tr>
<td>Bibliography</td>
<td>47</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

On the quest to explore the realm of geometrical problems, a primary aim is to be able to draw a map, i.e., to locate problems with regard to their complexity: Which problems are easy, which are difficult? Which ones can be efficiently approximated and which resist these attempts? And where precisely, for families of similar problems, lie the boundaries between easy and difficult, approximable and inapproximable?

An example for such a family of similar problems are the Euclidean degree-$k$ minimum spanning tree (MST) problems. They ask for the minimum length spanning tree on a set of points in the plane, where each node in the tree is allowed to have degree at most $k$. Such trees are of interest because the tree’s maximum degree is a relevant parameter for many applications that use minimum spanning tree constructions as an intermediary step; the running time of some of these algorithms even depends exponentially on the spanning tree’s maximum degree [RS95].

The complexity of the decision problems corresponding to the Euclidean degree-$k$ MST problems has long been known for all but one $k$: For $k = 2$, the smallest sensible choice for $k$, the Euclidean degree-$k$ MST problem is equivalent to the path version of the Euclidean TSP, that is well known to be NP-hard [Pap77]. On the other hand, as no Euclidean minimum spanning tree contains a vertex with degree larger than 6, the Euclidean degree-$k$ MST problem corresponds to the unbounded Euclidean MST problem and is therefore, in its optimization variant, polynomial-time solvable for $k \geq 6$. The same holds for $k = 5$ [MS92], whereas for $k = 3$, it is NP-hard [PV84] (for details on the proofs, see section 2.2). For $k = 4$, which also figures as “Problem 48” [Fek03] in “The Open Problems Project” [DMO], the complexity of the problem was conjectured to be NP-hard in 1984 [PV84], but the question was
open since. As uncharted territory between two related famous computational geometry problems that reside in differing complexity classes, it constitutes an open problem that is worth contemplating.

The semester thesis this report documents had the Euclidean degree-4 minimum spanning tree problem as its topic. Of all open questions about the problem (see task description, Appendix A), I focused on the complexity question. Fortunately, the question could be settled in the course of the semester thesis: The Euclidean degree-4 MST problem is NP-hard, as conjectured by Papadimitriou and Vazirani. Before presenting the proof, which can also be found in [FH09], in Chapter 3, this report provides a formal definition of the problem (Section 2.1) and an overview on related work (Section 2.2). Following the proof, Chapter 4 lists a number of simpler and less successful attempts at the proof conducted before the reduction presented in Chapter 3 was found, illustrating why the elaborateness of the latter might be necessary. Before concluding, a chapter on open questions (Chapter 5) highlights interesting related issues that are left open. Appendix B collects large figures and tables that are necessary for the completeness of the proof in Chapter 3 but not crucial to its understanding, and Appendix C gives the coordinates for a vertex gadget necessary to perform a proof for NP-hardness of the Euclidean degree-3 MST problem analogous to the one presented in Chapter 3.
Chapter 2

The Problem

2.1 Problem Definition

We start out with a definition of the notions and abbreviations we need during the course of the report, followed by a definition of the problem that this thesis sets out to solve, as well as a definition of the decision variant of the Euclidean degree-4 MST problem. The last paragraph outlines the necessary steps to answer our main question.

Notation. Consider a set \( P \) of \( n \) points in the plane. For any geometric graph \( G = (P, E) \) defined on \( P \), let its weight \( w(G) \) be \( w(G) = \sum_{\{p,q\} \in E} d(p,q) \), where \( d(p,q) \) denotes the Euclidean distance between two points \( p = (p_x, p_y) \) and \( q = (q_x, q_y) \) in \( \mathbb{R}^2 \): \( d(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \). A spanning tree for \( P \) is a connected and acyclic graph whose vertex set is \( P \). A Euclidean minimum spanning tree (EMST) for \( P \) is a spanning tree of minimum weight among all spanning trees for \( P \). In general, a point set does not have a unique EMST. Denote the weight of an EMST for \( P \) by \( \psi(P) \). An edge that appears in some EMST for \( P \) is called an EMST-edge for \( P \).

For \( k \geq 2 \) a maximum degree \( k \) spanning tree (Dk-ST) for \( P \) is a spanning tree in which no vertex has degree larger than \( k \). A Euclidean degree-\( k \) minimum spanning tree (E-Dk-MST) for \( P \) is a spanning tree of minimum weight among all Dk-STs for \( P \). Denote the weight of an E-Dk-MST for \( P \) by \( \psi_k(P) \).

Problem Definition. As mentioned in the introduction, the question this thesis focuses on is whether the E-D4-MST problem is NP-hard or not – colloquially expressed. More formally, when we are talking about NP-hardness of the E-D4-MST problem, what is meant is NP-hardness of the corresponding
2. The Problem

decision problem, as the notion is only defined for decision problems, not for
optimization problems as, e.g., the E-D4-MST problem.

The decision version of the E-D4-MST problem is the following question:

Problem 2.1 (Euclidean degree four MST) Given a set \( P \) of \( n \) points in the
Euclidean plane and a positive integer \( w \), is there a spanning tree of the
points in \( P \) with no vertex degree exceeding four that has weight \( \leq w \)?
It was conjectured to be NP-hard in [PV84].

Task Outline. In order to settle the complexity of Problem 2.1, we either need
to provide a polynomial time algorithm that decides the problem and thus
disproves the above-mentioned conjecture, or prove it by showing that Problem
2.1 is NP-hard. In order to prove NP-hardness, it is sufficient to show that
a known NP-hard problem \( L \) can be polynomial-time reduced to the problem
at hand, i.e., we need to provide a polynomial-time construction which trans-
forms an arbitrary input to \( L \) into an input to Problem 2.1, guaranteeing that
an oracle to Problem 2.1 can efficiently solve any instance of \( L \).

2.2 Related Work

This section gives an overview on problems related to the E-D4-MST problem,
ranging from the most closely related ones, which would be the E-Dk-MST
problems mentioned in the introduction, in more detail, to less closely related
problems in less detail.

E-Dk-MST Problems. Recall from the introduction that the E-Dk-MST prob-
lem is polynomial-time solvable for \( k \geq 5 \) and NP-hard for \( k \leq 3 \). Here are
short outlines of the corresponding proofs:

For \( k \geq 6 \), the polynomial-time solvability follows from the fact that no EMST
contains a vertex of degree larger than six. Assume an EMST contains a ver-
tex \( p \) with degree larger than six (see Figure 2.1). Then at least one of the
angles spanned by two edges incident to the vertex, say \( \{p, q\} \) and \( \{q, r\} \), is
\(< 60^\circ \). It is thus not the largest angle in \( \triangle pqr \), which is equivalent to the
edge opposite to it, \( \{q, r\} \), not being the longest edge in \( \triangle pqr \). So one of the
EMST edges \( \{p, q\} \) and \( \{q, r\} \) must be the largest edge in \( \triangle pqr \), which contra-
dicts with the fact that an EMST never contains the largest edge in a triangle.
So a tree in which \( \{p, q\} \) is replaced by \( \{q, r\} \) is shorter than the initial tree.
Thus, an EMST never contains a vertex of degree larger than six, and finding an E-Dk-MST for \( k \geq 6 \) is just equivalent to finding a non-degree-restricted EMST, and therefore polynomial-time solvable [PV84].

Figure 2.1: On the left, a spanning tree containing a vertex \( p \) of degree larger than six. \( p \) has an incident angle \( \angle qpr < 60^\circ \). On the right, a shorter spanning tree in which \( \{p, q\} \) is replaced by \( \{q, r\} \). The tree on the left is therefore not an EMST.

If an EMST contains a vertex \( v \) of degree 6, its incident edges are all of equal length and spaced by angles of 60° each (see Figure 2.2). By connecting two of the neighbours of \( v \) by an edge and removing one of the other sides of the newly created triangle, we remove one degree six vertex from the tree. As our former degree six vertex cannot be adjacent to vertices of degree five, no new vertex of degree six will be created in the process. Thus, by switching edges as described above, we obtain a spanning tree of equal length with one vertex of degree six less. It follows that an E-D5-MST can be obtained in polynomial time by first determining an EMST and then removing its degree six vertices in time linear in their number [MS92].

Figure 2.2: If an EMST contains a vertex of degree six (left), its incident edges are all of equal length and spaced by angles of 60° each, and an EMST of equal length with one vertex of degree six less (right) is readily determined.
2. The Problem

The most involved proof of all hitherto known cases is needed for $k = 3$ and can be found in [PV84] (see also Chapter 4). It reduces the Hamilton circuit problem for planar directed graphs with indegree $v$ + outdegree $v = 3$ for every vertex $v$ to the E-D3-MST problem by embedding the corresponding graph on the grid and transforming it in such a way that every vertex has room for an additional “suburb” vertex in direction of the empty neighbouring place on the grid. “suburb” hereby means that in every EMST, the corresponding vertex will be connected to its “city center” vertex as a leaf. These suburbs can be placed in such a way that the E-D3-MST of the obtained pointset consists, if there is a Hamiltonian path, of this path and one link to the additional suburb per vertex; it has a well-defined length $L$. If conversely the original graph does not have a Hamiltonian path, the E-D3-MST of the pointset must be longer than $L$.

For $k = 2$, the E-D2-MST problem is, as mentioned in the introduction, equivalent to the path-version of the traveling salesman problem.

Approximating E-Dk-MSTs. Given the known or conjectured NP-hardness of the E-Dk-MST problems for $k \leq 4$, an evident aim is to approximate the spanning trees in polynomial time, or even to devise approximation schemes.

Denote by $\tau_k(P) = \psi_k(P)/\psi(P)$ the ratio of a minimum bounded degree spanning tree to a minimum unrestricted spanning tree for $P$. It is known that $\tau_2 = 2[FKK+97]$ and $\tau_5 = 1 [MS92]$. Khuller, Raghavachari et al. [KRY96] showed how to compute for any set $P$ of $n$ points in the plane an E-D4-MST of weight at most $1.25\psi(P)$, thereby proving $\tau_4(P) \leq 1.25$. They also obtained a corresponding bound for the degree three case: $\tau_3(P) \leq 1.5$. The algorithms are based on local transformations to reduce the degree of high degree vertices starting from an EMST. Chan [Cha04] improved these bounds to $\tau_3(P) \leq 1.402$ and $\tau_4(P) \leq 1.143$, also using local transformations starting from an EMST. In turn, the analysis of Chan’s algorithm was improved by Jothi and Raghavachari [JR04] who proved that it actually computes an E-D4-MST of weight at most $\frac{2+\sqrt{2}}{3}\psi(P)$ and thus $\tau_4(P) \leq (2+\sqrt{2})/3 < 1.1381$.

On the lower bound side, Fekete et al. [FKK+97] gave examples of point sets for which $\tau_4(\cdot) = (2\sin(\pi/5) + 4)/5 > 1.035$ and $\tau_3(\cdot) = (3+\sqrt{2})/4 > 1.1035$, respectively.

Arora and Chang [AC04] gave a quasi-polynomial time approximation scheme (QPTAS) to obtain a $(1 + \epsilon)$-approximation for an E-Dk-MST in $n^{O(\log^2 n)}$ time. The well known PTASs for the Euclidean Traveling Salesman Prob-
lem [Aro98, Mit99] do not seem to generalize to the bounded degree scenario, contrary to what was claimed in one of the early papers [Aro96].

Other Bounded Degree Spanning Tree Problems. A natural extension of the E-Dk-MST problems is to consider the same problem and its approximation in higher dimensions [AC04, Cha04, KRY96, FKK+97], for different metrics [FKK+97], especially for L_p metrics [RS95], or for graphs: Goemans [Goe06] approximates a degree-k minimum spanning tree with weight w on graphs in the sense that he presents an algorithm that finds a tree with maximum degree k + 2 with weight at most w.

Low Degree Structures and Low Degree Spanning Tree Problems. In a larger context, the E-Dk-MST problems lie at the intersection of two families of well studied problems: minimum spanning trees problems on one hand, degree-constrained structures on the other. An overview on the latter can be found in Raghavachari's survey [Rag96]. Other problems that are part of both group of problems are e.g. the one treated by Fürer and Raghavachari in [FR92]: They consider the problem of constructing a spanning tree on a graph whose maximal degree \( \Delta^* \) is minimal among all spanning trees of the graph, which is NP-hard, and its approximation: A tree with maximal degree \( O(\Delta^* + 1) \) can be found in polynomial time.

\[1\] Note that there is a number of names for degree bounded structures: e.g., "degree-constrained", "degree-restricted" and "bounded-degree".
Chapter 3

The Proof

In this chapter, our aim is to show that Problem 2.1 is NP-hard; The reduction described in the following is from vertex cover in cubic graphs. Section 3.1 outlines the reduction. Section 3.2 presents how the vertex and edge gadgets, the building blocks of our construction, function and look like in detail and finally, Section 3.3 finalizes the proof by bringing together the gadget’s properties and the reduction’s structure.

3.1 The Reduction

Given a graph \( G = (V, E) \), a vertex cover for \( G \) is any subset \( V^* \subseteq V \) that contains at least one endpoint from every edge in \( E \). We start from the following restricted version of the vertex cover problem that is known to be NP-complete \[GJ77, GJ79\].

**Problem 3.1 (VC in cubic planar graphs)** Given a planar graph \( G = (V, E) \) with each vertex incident to exactly three edges and a positive integer \( w \), does there exist a vertex cover \( V^* \subseteq V \) for \( G \) satisfying \( |V^*| \leq w \)?

Let \( G_1 = (V_1, E_1) \) be a cubic planar graph on \( n \) vertices. In the following, we exhibit a series of steps which construct, based on \( G_1 \), a set \( P \) of points on an \( O(n) \times O(n) \) integer grid with the following property: \( P \) admits an E-D4-MST of weight \( \leq \psi(P) + k \cdot \varepsilon \) if and only if \( G_1 \) has a vertex cover of size \( \leq k \), where \( \varepsilon \) is a constant (independent of \( n \) and \( k \)). Observe that for points on an \( O(n) \times O(n) \) integer grid, \( \psi(P) = O(n^2) \).

**Overview of the Construction.** First, path-embed \( G_1 \) on the rectilinear grid \( G \), the graph whose vertices are all points with integer coordinates and in which two vertices are adjacent if and only if their distance is one. That is, we map
vertices of $G_1$ to vertices of $G$ and edges of $G_1$ to pairwise internally vertex-disjoint paths on $G$. In other words, we find a subdivision of $G_1$ in $G$. Such an embedding of a graph $G$ into $G$ is called an orthogonal drawing of $G$. As the degree in $G_1$ is bounded by three, an orthogonal drawing $G_2 = (V_2, E_2)$ of $G_1$ into an $O(n) \times O(n)$ subgrid of $G$ can be obtained in linear time \cite{TT89, PT98}. Observe that $G_2$ in general contains many more vertices than $G_1$, but due to the bounded gridsize, $|V_2| = O(n^2)$. In the following, we make a distinction between those vertices $V_c \subseteq V_2$ that correspond to vertices in $G_1$ and those vertices $V_\text{c} = V_2 \setminus V_c$ that were added along the connecting paths. As an example, Figure 3.1 shows an orthogonal drawing of $K_4$ in which vertices of $V_c$ are marked black.

In the second step, construct a graph $G_3 = (V_3, E_3)$ from $G_2$. The vertices of $G_3$ will serve as an input point set to the E-D4-MST problem eventually. Each vertex from $V_c$ is represented in $G_3$ by a vertex gadget that is discussed in detail below. For now it is enough to know that this gadget is of constant size and thus we could fit it into the drawing $G_2$ by blowing it up by a constant factor. More formally, blowing up $G_2$ by a factor of $c$ means to replace each vertex with coordinates $(x, y) \in \mathbb{Z}^2$ by a vertex with coordinates $(cx, cy)$.

These vertex gadgets are connected in $G_3$ by edge gadgets that come in two different types. Choose (in $O(|V_1| + |E_1|)$ time) an arbitrary spanning tree $T$ for $G_1$. For any two vertices of $G_1$ that are adjacent in $T$, the corresponding vertex gadgets in $G_3$ are connected by a double-edge gadget that consists of two parallel paths. For any two vertices that are connected by an edge in $G_1$ that is not in $T$, the corresponding vertex gadgets are connected by a single-edge gadget in $G_3$ that consists of a single path. The idea is the following: Out of any two parallel paths one should be forced into any EMST and thus provide an overall connectivity among the vertex gadgets in the same way as $T$ connects the vertices in $G_1$. The other path, just like any single path, must be connected in every EMST to at least one of the two incident vertex gadgets, just like for each edge in $G_1$, a vertex cover must contain at least one of the two incident vertices.

Finally, we disregard the edges of $G_3$ and consider its vertices as a set $P$ of points in the plane. We will show that $P$ has a short E-D4-MST if and only if $G_1$ has a small vertex cover.
3.2 The Gadgets

This section describes the basic components the graph $G_3$ is built from. For each vertex of $G_1$, there is a corresponding vertex gadget in $G_3$, and for each edge $\{u, v\}$ of $G_1$, there is a corresponding edge gadget in $G_3$ that is attached to the vertex gadgets of $u$ and $v$.

**Vertex Gadget.** Each vertex gadget consists of 27 points that we denote by $p_1, \ldots, p_{27}$, aligned as shown in Figure 3.2. Here is a rough intuition behind the construction: In order to obtain a point set $G_3$ for which an E-D4-MST yields a minimum vertex cover for $G_1$, we need vertex gadgets that, when the maximally allowed vertex degree in a spanning tree is reduced from five to four, take on the same role as vertices in a vertex cover. That is, under a small additional cost, such a gadget should provide connections to all incident optional connections (that need such a connection at at least one of their endpoints, as described above). Without this additional cost, none of these connections should be possible. As you will see in the following two sections, this is achieved by degree five vertices that are adjacent in the EMST of $G_3$, such as $p_{17}$ and $p_{20}$. Replacing the edge that connects the two vertices $(p_{17}p_{20})$ by a slightly more expensive one $(p_{16}p_{19})$ allows for two outgoing connections $(p_{15}p_{17}, p_{20}p_{23})$ in the degree four setting that are both not available if the edge that causes degree five twice in the EMST is in the D4-ST.
Appendix another pair of degree five vertices (p8 and p11) to an end of the first pair expands this effect for three instead of two connections. Additional degree five vertices (p4, p24) are needed in order to obtain three optional edges of equal length (p2p4, p11p12, p24p26).

For a more formal examination of the vertex gadget, observe that for a vertex gadget in isolation, there is a small set of possible EMSTs only.

**Proposition 3.2** Any EMST T for the 27 points listed in Figure 3.2 contains all edges shown by a solid line there. Furthermore, T contains exactly one edge from each of the following three pairs (shown by a dotted line in Figure 3.2): \{p4p6, p6p8\}, \{p8p11, p15p17\}, and \{p20p23, p23p24\}.

**Proof** By inspection.¹ (Any EMST can be obtained greedily, that is, by iteratively adding a shortest edge that does not yield a cycle to an initially empty forest.)

Note that the longest EMST-edge within a vertex gadget has length |p2p4| = |p11p12| = |p24p26| = \sqrt{769} < 27.731. Also observe that any EMST for a vertex gadget has at least two degree five vertices.

As there are n copies of this vertex gadget in G3, formally we should write p^j_i to denote the point p_i in the j-th vertex gadget. However, as usually it is clear from the context which p_i is meant (possibly all p_i’s collectively), we refrain from cluttering the presentation with indices.

**Edge Gadget.** Vertex gadgets are connected by edge gadgets that have a very simple form: either they consist of a single path (single-edge gadget) or of two parallel paths (double-edge gadget). The three points p2, p12, and p26 of a vertex gadget are referred to as its bond vertices because they serve as an optional connection to the attached edge gadget. In fact, one should think of the bond vertices being part of attached edge gadget rather than of the vertex gadget. The set of vertices of a vertex gadget that are not bond vertices are called its internal vertices.

The vertices p1, p3, p10, p13, p25, and p27 of a vertex gadget are called its glue vertices because they serve as a possible connection to the “forced” path of a double-edge gadget. (The optional/forced terminology is just for intuition, the paths themselves do not differ in any way. The “force” is implicit and caused by the fact that, say, edge p3p4 is shorter than edge p2p4.)

¹In the appendix we give a table of all pairwise distances.
The bond vertices in Figure 3.2 are located at the top, right, and bottom side of the gadget. Depending on where the three edges incident to a vertex from \( V \) are located, we use an appropriately rotated and/or reflected version of the gadget to represent that vertex. In order to give the different variations of the vertex gadget a uniform appearance, wrap it into a box of size 248 × 248 in which the points with coordinates as listed in Figure 3.2 are translated by the vector \((89, 74)\).

The path of an edge gadget connecting to the vertex gadget enters the box around the center of the side from where it approaches. For instance, a path from above (connecting to one of \( p_1 \), \( p_2 \), or \( p_3 \)) enters the box at either \((110, 248)\) or \((138, 248)\). We call these two points the top portals of the gadget. Symmetrically, the bottom portals are \((110, 0)\) and \((138, 0)\), the left portals are \((0, 110)\) and \((0, 138)\), and the right portals are \((248, 110)\) and \((248, 138)\). Only some of the portals are vertices of \( G_3 \), depending on the type and location of the three attached edge gadgets.

Observe that the distance between any two portals is at least 28 which is larger than the length of any EMST-edge within a vertex gadget. We will keep this spacing of at least 28 in routing the edge-gadget paths to their destination vertex within the box. In case of a single-edge gadget, only one of the two portals is used. We simply route the path along the grid using at most two bends and such that any bend has distance at least 14 to the box boundary and at least 28 to the 27 points of the vertex gadget. Here “routing along the grid” means to insert every vertex along the path into the edge gadget and thereby into \( G_3 \). In figures, these paths appear as thick lines because the points along them are spaced so densely that they cannot be distinguished visually.

For a double-edge gadget, we can choose which of the two glue vertices on this side to connect to, which gives us the freedom to choose the order of the forced path and the optional path along the boundary of the box in a way that is consistent with the corresponding order for the second vertex gadget the edge gadget is attached to. The two different types of routings are shown in Figure 3.3. For the connection to the right side, the two different orders can be obtained by a horizontal reflection. As the routings to the top and to the bottom are independent, together with an appropriate rotation all possible degree three vertices combined with any forced/optional portal order can be realized.

This completes the description of \( G_3 \) and the point set \( P = V_3 \). For illustration let us consider the graph \( G_1 = K_4 \) as an example. As a first step, take the orthogonal drawing \( G_2 \) given in Figure 3.1. For the second step choose as a
3. The Proof

![Diagram showing connection routing for two different glue vertices.](image)

Figure 3.3: Connection routing for the two different glue vertices.

spanning tree $T_v$, the tree induced by the vertices of $V_v$ in $G_2$ (marked black in Figure 3.1). The resulting point set $P$ is shown in Figure 3.4

Analysis. We will now analyze some properties of the gadgets introduced above and the way they interact with each other, in particular related to EMSTs for $P$. Let us first characterize how an EMST for $P$ looks like.

**Proposition 3.3** An EMST-edge for $P$ has length at most $\sqrt{769} < 27.731$.

**Proof** Proposition 3.2 shows how to connect each vertex gadget locally using edges of length at most $\sqrt{769}$. Edge gadgets are built from induced paths on $G$, that is, unit length edges. Consider the graph $G$ on $P$ formed by all these edges. Each path of an edge gadget is connected in $G$ to two bond vertices of vertex gadgets. The set of vertex gadgets is connected by construction via the paths that correspond to edges of $T_v$. In summary, $G$ is a spanning subgraph for $P$ in which all edges have length at most $\sqrt{769}$. The claim follows noting that any EMST can be constructed greedily.

The following proposition ensures that we can argue about spanning trees for vertex gadgets locally.

**Proposition 3.4** Let $T$ be an EMST for $P$ and consider the graph $H$ induced by $T$ on the internal vertices of a vertex gadget. Then $H$ is a spanning tree.

**Proof** Clearly $H$ is acyclic as a subgraph of an acyclic graph. It remains to show that $H$ is spanning. Denote the set of internal vertices of the gadget by
3.2. The Gadgets

Figure 3.4: Construction of $P$ for $G_1 = K_4$. Solid edges are part of any EMST for $P$. Dotted edges form pairs such that any EMST for $P$ contains exactly one edge from each pair.

According to Proposition 3.3, no point is connected in $T$ to a point that is in distance larger than $\sqrt{769}$. For a vertex gadget, by construction the only such points are the 27 points of the gadget itself plus the points along the edge gadget paths attached to a bond or glue vertex.

The points along any edge gadget path—including the bond or glue vertex the path is attached to—are connected in $T$ using unit length edges only. Therefore, at most one point from any edge gadget path is connected in $T$ to a vertex from $I$. Moreover, for any edge gadget path there is a unique shortest edge to connect it to a vertex from $T$: for a path to $p_2$ it is the edge $p_2p_4$, for a path to $p_{12}$ it is the edge $p_{11}p_{12}$, and for $p_{26}$ it is the edge $p_{24}p_{26}$. However, these three edges are the uniquely longest EMST-edges within a vertex gadget. That is, $I$ can be connected using edges of length strictly less than $\sqrt{769}$.

Denote by $G$ the graph on $P$ that contains all edges from $\binom{P}{2}$ that have length strictly less than $\sqrt{769}$. Recall that the forced paths of the double-edge gadgets correspond to the edges of $T_*$ in $G_1$. Therefore, along these paths the internal vertices of all vertex gadgets collectively are connected in $G$. But as $T_*$ is acyclic, there is no cycle in $G$ that passes through points from at least two different vertex gadgets. As $T$ can be constructed greedily, it induces a spanning tree on each connected component of $G$. It follows that $H$ is spanning. \qed
3. The Proof

Ultimately, we are interested in an E-D4-MST for \( P \). Recall that any EMST for an isolated vertex gadget contains some degree five vertices. It is not yet clear whether this is also true when considering the whole set \( P \) as opposed to a single gadget in isolation. But as it turns out, there is a certain number (to be quantified precisely later) of degree five vertices in any EMST for \( P \). In other words, every E-D4-MST for \( P \) contains some non-EMST-edges.

We measure the impact of an edge \( e \in \binom{P}{2} \) on the weight of a spanning tree \( T = (P, E) \) by defining its insertion cost \( \text{ins}_T(e) = w(T') - w(T) \), where \( T' = (P, E') \) is a minimum spanning tree for \( P \) such that \( \{e\} \subseteq E' \subseteq E \cup \{e\} \). Analogously for an acyclic set \( F \subseteq \binom{P}{2} \) of edges define \( \text{ins}_T(F) = w(T') - w(T) \), where \( T' \) is a minimum spanning tree for \( P \) such that \( F \subseteq E(T') \subseteq E(T) \cup F \).

Clearly, if \( T \) is an EMST, then \( \text{ins}_T(e) \geq 0 \) and \( \text{ins}_T(e) = 0 \) if and only if \( e \) is an EMST-edge. Insertion costs are super-additive in the following sense.

**Proposition 3.5** \( \text{ins}_T(F) \geq \sum_{f \in F} \text{ins}_T(f) \).

**Proof** Insertion of an edge \( f \in F \) into \( T \) yields a tree \( T' \) that either coincides with \( T \) (if \( f \in E(T) \)) or differs from \( T \) in exactly two edges \( f \in E(T') \setminus E(T) \) and \( e \in E(T) \setminus E(T') \). In the latter case, the possible choices for \( e \) are the edges along the unique cycle in \( T \cup f \) except \( f \)—and a minimum weight tree \( T' \) is obtained by selecting \( e \) to be a longest of these edges. The sum on the right-hand side corresponds to a selection of a longest edge individually for each edge in \( F \), which certainly is a lower bound for the overall cost of inserting all edges from \( F \) collectively. But altogether any single edge from \( T \) can be removed at most once. Therefore, not all of these individual choices may be compatible, which may increase the weight on the left-hand side. \( \square \)

Finally, the following two lemmata quantify how close in weight a D4-ST for a gadget can be compared to an unrestricted EMST.

**Lemma 3.6** Let \( I \) be the set of internal vertices of some vertex gadget, and let \( B \subseteq \{p_2, p_{12}, p_{26}\} \) be any non-empty subset of its bond vertices. Then any spanning tree for \( A = I \cup B \) that contains a non-EMST-edge has weight at least \( \psi(A) + \epsilon \), where \( \epsilon = d(p_{16}, p_{19}) - d(p_{17}, p_{20}) \in [0.07680, 0.07681] \).

**Proof** Proposition 3.2 describes a spanning tree using edges of length at most \( \sqrt{769} \) for \( A \). Therefore we need to consider non-EMST-edges of length at most \( \sqrt{769} + \epsilon < 27.81 \) only. It is easily verified\(^2\) that this leaves us with the candidates listed in Table \( \text{Table 3.1} \). By Proposition 3.5 we only have to check the insertion cost \( \text{ins}_T(e) \) into an EMST \( T \) for each candidate edge \( e \). Compute

\( ^2 \) The corresponding entries are marked with a dark background in Table \( \text{Table 3.1} \) in the appendix.
3.2. The Gadgets

\(\text{ins}_T(e)\) by checking which of the edges listed in Proposition 3.2 is an edge \(f(e)\) that is a longest edge on a cycle through \(e\). As shown in Table 3.1, there is exactly one edge, \(p_{16}p_{19}\), with insertion cost \(\epsilon\), and the insertion cost for all other edges is strictly greater.

\[
\begin{array}{cccccccc}
  e & p_{1}p_{5} & p_{3}p_{6} & p_{4}p_{7} & p_{4}p_{8} & p_{4}p_{9} & p_{5}p_{6} & p_{5}p_{7} \\
  f(e) & p_{1}p_{4} & p_{3}p_{4} & p_{4}p_{6} & p_{4}p_{6} & p_{4}p_{6} & p_{5}p_{6} & p_{5}p_{7} \\
  \text{ins}_T(e) > & 0.18 & 0.25 & 6 & 5 & 9 & 0.082 & 0.3 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  e & p_{5}p_{9} & p_{6}p_{7} & p_{6}p_{9} & p_{6}p_{15} & p_{7}p_{9} & p_{7}p_{10} & p_{7}p_{11} \\
  f(e) & p_{5}p_{8} & p_{6}p_{8} & p_{6}p_{8} & p_{9}p_{15} & p_{7}p_{8} & p_{8}p_{11} & p_{8}p_{11} \\
  \text{ins}_T(e) > & 6 & 8 & 0.28 & 2.7 & 7 & 0.92 & 0.4 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  e & p_{8}p_{15} & p_{9}p_{11} & p_{9}p_{14} & p_{14}p_{15} & p_{14}p_{18} & p_{16}p_{19} & p_{17}p_{19} \\
  f(e) & p_{9}p_{15} & p_{8}p_{11} & p_{8}p_{11} & p_{8}p_{11} & p_{14}p_{17} & p_{17}p_{20} & p_{17}p_{20} \\
  \text{ins}_T(e) > & 7 & 0.24 & 0.24 & 0.3 & 0.95 & \boxed{\epsilon} & 0.4 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  e & p_{17}p_{21} & p_{18}p_{21} & p_{19}p_{21} & p_{19}p_{22} & p_{19}p_{23} & p_{19}p_{24} & p_{20}p_{24} \\
  f(e) & p_{17}p_{20} & p_{17}p_{20} & p_{19}p_{20} & p_{20}p_{22} & p_{20}p_{23} & p_{20}p_{23} & p_{20}p_{23} \\
  \text{ins}_T(e) > & 0.24 & 0.24 & 7.8 & 0.3 & 8 & 6 & 5 \\
\end{array}
\]

\[
\begin{array}{cccccccc}
  e & p_{21}p_{22} & p_{21}p_{23} & p_{21}p_{24} & p_{22}p_{23} & p_{22}p_{25} & p_{23}p_{27} \\
  f(e) & p_{20}p_{22} & p_{20}p_{23} & p_{20}p_{23} & p_{20}p_{23} & p_{24}p_{25} & p_{24}p_{27} \\
  \text{ins}_T(e) > & 6 & 0.28 & 10 & 0.082 & 0.18 & 0.25 \\
\end{array}
\]

Table 3.1: Lower bound for \(\text{ins}_T(e)\), together with replaced edge \(f(e)\), for edges \(e\) in \(A\) of length \(\leq \sqrt{769} + \epsilon < 27.81\) (cf. Lemma 3.6).
3. **The Proof**

![Figure 3.5: Spanning trees of \( A \) in Lemma 3.7](image)

**Lemma 3.7** Under the same assumptions as in Lemma 3.6, \( A \) has a D4-ST of weight \( \psi(A) + \varepsilon \).

**Proof** Assume without loss of generality that all three bond vertices are in \( A \). Consider the tree \( T(A) \) depicted in Figure 3.5a and the tree \( T'(A) \) depicted in Figure 3.5b. We know from Proposition 3.2 that \( T(A) \) is an EMST of \( A \). As \( T'(A) = (T(A) \setminus \{p_{17}p_{20}\}) \cup \{p_{16}p_{19}\} \), we have \( w(T'(A)) = w(T(A)) + \varepsilon = \psi(A) + \varepsilon \). Clearly \( T'(A) \) is a D4-ST of \( A \). □

### 3.3 Proof of Equivalence

We are now ready to complete the proof by quantifying the correspondence between a small vertex cover in \( G_1 \) and a D4-ST of small weight for \( P \). Roughly speaking, the idea is the following: In \( P \) we have to connect all optional paths of edge gadgets to one of the two attached vertex gadgets. However, by doing so, we force at least one degree five vertex into any EMST for that gadget. Fortunately, there is a reasonably cheap way to get rid of all degree five vertices in a vertex gadget by paying an extra cost of \( \varepsilon \). So how can we minimize the extra cost to be paid? By choosing as few vertices as possible, but still covering all edges. A minimum vertex cover...

**Lemma 3.8** \( G_1 \) has a vertex cover of size \( \leq k \) \( \iff \exists \) EMST on \( P \) with \( \leq k \) pairwise non-adjacent vertices of degree five.

**Proof** \("\Rightarrow\): Suppose \( G_1 = (V_1, E_1) \) has a vertex cover \( U \subseteq V_1 \) with \( |U| \leq k \). Construct an EMST \( S \) for \( P \) as follows. Consider the vertices of \( U \) one after another: In each gadget that corresponds to a vertex \( u \in U \) we include into \( S \) all EMST-edges except \( p_{4}p_{6}, p_{8}p_{11}, \) and \( p_{23}p_{24} \) (Figure 3.6a). Any of the three edges \( p_{2}p_{4}, p_{11}p_{12}, \) and \( p_{24}p_{26} \) is included only if the attached edge
gadget path is not already connected to the respective other vertex gadget that corresponds to the second endpoint of the edge. As U forms a vertex cover, in this way the optional path of each edge gadget is connected to exactly one vertex gadget.

Finally, in each gadget that corresponds to a vertex \( v \in V_1 \setminus U \) we include into \( S \) all EMST-edges except \( p_2p_4, p_6p_8, p_{11}p_{12}, p_{15}p_{17}, p_{20}p_{23} \), and \( p_{24}p_{26} \) (Figure 3.6b).

\( S \) is connected because the vertex gadgets are connected via the forced edge gadget paths corresponding to the tree \( T \) in \( G_1 \). The only vertices of degree greater than four in \( S \) appear in vertex gadgets corresponding to a vertex in the cover. However, the number of these gadgets is \( |U| \leq k \) and in each of them the two degree five vertices (\( p_{17} \) and \( p_{20} \)) are adjacent.

\( \Leftarrow \): Suppose there exists an EMST \( S \) on \( P \) that contains at most \( k \) pairwise non-adjacent vertices of degree five. Consider a vertex gadget corresponding to some vertex \( v \in V(G_1) \). We claim that if any of the three edges \( p_2p_4, p_{11}p_{12}, \) or \( p_{24}p_{26} \) is in \( S \), then \( S \) contains at least one degree five vertex within this gadget. Suppose the claim holds. Then every vertex gadget that connects in \( S \) to at least one of the attached optional edge gadget paths induces at least one degree five vertex in \( S \). However, as \( S \) is spanning, each optional edge gadget path has to connect to at least one of its two neighboring vertex gadgets. In other words, the set of vertices from \( G_1 \) whose gadgets connect to at least one attached optional edge gadget path forms a vertex cover in \( G_1 \). As each such gadget induces a degree five vertex in \( S \) and no two vertices from different vertex gadgets can be adjacent in \( S \) (cf. Proposition [3.3]), and by assumption \( S \) does not contain more than \( k \) pairwise non-adjacent degree five vertices, it follows that \( G_1 \) has a vertex cover of size \( \leq k \).

It remains to prove the claim. By Proposition [3.4] \( S \) is connected on the internal vertices of each vertex gadget. Thus \( S \) contains all local EMST-edges,
except for the three pairs \( \{p_4p_6, p_6p_8\}, \{p_8p_{11}, p_{15}p_{17}\}, \) and \( \{p_{20}p_{23}, p_{23}p_{24}\} \), each of which contributes exactly one edge to \( S \). If \( p_{15}p_{17} \in E(S) \) then \( \deg_S(p_{17}) = 5 \) and there is nothing to show. Otherwise, \( p_{8}p_{11} \in E(S) \) and if \( p_6p_8 \in E(S) \) then \( \deg_S(p_8) = 5 \). Hence suppose that \( p_4p_6 \in E(S) \). If \( p_2p_4 \in E(S) \) then \( \deg_S(p_4) = 5 \). Similarly, if \( p_{11}p_{12} \in E(S) \) then \( \deg_S(p_{11}) = 5 \). Finally, if \( p_{20}p_{23} \in E(S) \) then \( \deg_S(p_{20}) = 5 \). Otherwise, \( p_{23}p_{24} \in E(S) \) and \( \deg_S(p_{24}) = 5 \), unless \( p_{24}p_{26} \notin E(S) \), which completes the proof. \( \square \)

**Lemma 3.9** \( P \) has a Euclidean minimum spanning tree with \( \leq k \) pairwise non-adjacent vertices of degree five if and only if it has a D4-ST of weight \( \leq \psi(P) + k \cdot \epsilon \).

**Proof** \( \Rightarrow \): Let \( T \) be an EMST of \( P \) with \( \leq k \) pairwise non-adjacent vertices of degree five. All of these are placed within vertex gadgets and cannot spread over more than \( k \) such gadgets. From Lemma 3.7 we know that an additional cost of \( \epsilon \) per vertex gadget suffices to lift all degree five vertices in that gadget. It follows that if \( P \) has a EMST with \( \leq k \) pairwise non-adjacent vertices of degree five, it also has an D4-ST of weight \( \leq \psi(P) + k \cdot \epsilon \).

\( \Leftarrow \): Assume \( P \) has no EMST with \( \leq k \) pairwise non-adjacent degree five vertices. Let \( T \) be a D4-ST, and let \( E \) be the set of edges in \( T \) that are EMST-edges. \( E \) can be completed to an EMST \( T' \), which, as we know from the assumption, has \( > k \) pairwise non-adjacent vertices of degree five. Therefore, in order to transform \( T' \) back into \( T \), we need to exchange a set \( F \) of more than \( k \) edges. We know from Lemma 3.6 that the cheapest insertion cost among all non-EMST-edges is \( \epsilon \). Therefore, by Proposition 3.5 \( T \) has weight at least \( \psi(P) + \text{ins}_{T'}(F) \geq \psi(P) + \sum_{f \in F} \text{ins}_{T'}(f) > \psi(P) + k \cdot \epsilon \). \( \square \)

Putting Lemma 3.8 and Lemma 3.9 together, it follows: \( G_1 \) has a vertex cover of size \( \leq k \) if and only if \( P \) has a D4-ST of weight \( \leq \psi(P) + k \cdot \epsilon \). This completes the proof of the following theorem.

**Theorem 3.10** The Euclidean degree four minimum spanning tree problem is strongly NP-hard.
Figure 3.7: An E-D4-MST for the example $G_1 = K_4$. The corresponding minimum vertex cover consists of all vertices but the central one.
Chapter 4

Other Approaches Taken

Papadimitrou and Vazirani state in [PV84] that “It seems that considerably stronger techniques [than those used for the NP-hardness proof for the degree-3 MST problem] would be required to show that the degree-4 MST problem is NP-complete (sic!)” Nevertheless, it does not seem clear at first glance why the reduction could not be adapted to the degree four case. During the first six weeks of my thesis, I mainly followed this approach. This section documents dead ends met during this phase and might serve as an illustration why the technicality and elaborateness of the proof presented in Chapter 3 might be necessary.

As outlined in Section 2.2, the reduction in [PV84] embeds an input to the Hamilton path problem on the integer grid in such a way that around every node, at least one adjacent place on the grid is unused, such that a “suburb” vertex can be added in direction of that empty neighboring place, “suburb” meaning that in every EMST, the vertex will be connected to its “city center” vertex as a leaf. In this way, if the maximally allowed vertex degree is reduced to three, one out of the possible three connections is used by the suburb, only leaving two for overall connectivity, forcing the E-D3-MST to contain the Hamilton path of the main vertices, if there is one. If there is not, connections between suburbs must be made, and the suburbs are placed in such a way that these connections are longer than center-center- or center-suburb-connections. Thus, if the input graph has a Hamilton path, the E-D3-MST of the vertices of the resulting graph has the embedded version of this Hamilton path as a subgraph – it is the spanning tree minus the newly added suburbs. That is, an E-D3-MST with the well-known length $l$ of an EMST on the same pointset is possible exactly if the input graph has a Hamilton path. So by deciding whether the resulting pointset has an E-D3-MST of length $l$ or less determines whether the input graph has a Hamilton path or not.
In order to adapt this reduction to degree four, we would want to embed the input graph in such a way that in an EMST, two suburbs would need to be connected to every vertex instead of one as described above. In this manner, two possible connections per vertex would be used up by the suburbs, while again two remain to provide overall connectivity. Four approaches to do so spring to mind: Embedding the input graph on the grid as before, but adding two instead of one suburb to every vertex on the grid; Embedding the input graph on a tessellation of the plane other than the rectangular grid, which could leave more space for an additional suburb; Slightly changing the grid structures in order to create more free space for two suburbs; and finally, structures with more than two points, using an elaborate system of suburbs and sub-suburbs that would force two additional connections onto every vertex of the grid. A description of each approach can be found in the paragraphs further below.

We also shortly considered other NP-hard problems as starting points for a reduction, more precisely, the traveling salesman problem and planar 3SAT; These approaches are left out in the following account. As for reasons of efficiency, we followed all approaches not to a point where we could prove that they are dead ends but only to the point where we could tell that they most probably were, many of the statements in the following paragraphs are inherently “handwavy”\(^1\) and vague.

\[\text{Figure 4.1: Two suburbs for a vertex on the grid?}\]

\footnotesize Two Suburbs for Every Vertex on the Grid. In the graph that results from embedding an input graph to the Hamilton path problem on the integer grid, \(G'_4\) in [PV84], every vertex has degree at most 3. This leaves us with a range of 180' degrees in which we can place suburbs, for now ignoring that the next neigh-

\(^1\)i.e., lacking details and rigor; cf. [http://en.wikipedia.org/wiki/Handwaving](http://en.wikipedia.org/wiki/Handwaving)
bour would need space for suburbs, too. Is this enough room for two suburbs, such that a local EMST with edges as drawn in Figure 4.1 is possible?

Let \( b \) be the vertex to which we want to add suburbs \( p \) and \( q \), and \( a \) and \( c \) \( b \)'s neighbours on the grid such that \( \{ a, b \} \) and \( \{ b, c \} \) form an angle of 180°. Let \( p \) be the suburb closer to \( a \) than to \( c \), while \( q \) is closer to \( c \) than to \( a \). In order to be connected to \( b \) in an EMST, while the connections \( \{ a, b \} \) and \( \{ b, c \} \) both also remain in the EMST, \( p \) and \( q \) must fulfill the following conditions:

1. \( p \) must be closer to \( b \) than to \( a \); \( q \) must be closer to \( b \) than to \( c \), as otherwise, \( p \) will rather be connected to \( a \), and \( q \) to \( c \). This means that the suburbs may not be placed in the area shaded grey in Figure 4.2.

2. In the triangle \( \triangle(abp) \), the edge \( \{ a, b \} \) may not be the longest one, as otherwise, connections \( \{ a, p \} \) and \( \{ p, b \} \) would replace \( \{ a, b \} \) in any EMST (recall that an EMST never contains the longest edge in a triangle), and we want \( \{ a, b \} \) to remain in any EMST. In \( \triangle(bcq) \), \( \{ b, c \} \) may not be the longest edge for analogous reasons.

3. From 1 and 2 follows: In \( \triangle(abp) \), \( \{ a, p \} \) must be the unique longest edge. Especially, it must hold that \( w(a, p) > w(a, b) \), so \( p \) is banned from a circular area with radius \( w(a, b) \) around \( a \). Similarly, in \( \triangle(bcq) \), \( \{ c, q \} \) must be the unique longest edge. This means that the suburbs
4. Other Approaches Taken

my not be placed in the area shaded grey in Figure 4.3. Note that the unshaded area does not allow the placement of two vertices $p$, $q$ such that $\angle pbq > 60^\circ$, as $\angle xby = 60^\circ$.

4. In $\triangle (bpq)$, $\{p, q\}$ must be the unique longest edge, as otherwise, the suburb farther away from $b$, say $q$, would rather be connected to $p$ than to $b$ in any EMST.

As in a triangle, a unique longest edge is always opposite an angle $> 60^\circ$, all three angles $\angle pbq$, $\angle abp$ and $\angle cbq$ must be $> 60^\circ$. Which contradicts with $\angle abc = 180^\circ$. Thus forcing additional degree two onto a vertex of the grid by squeezing two suburbs into $180^\circ$ is not possible.

Tessellations Other than the Rectangular Grid. For our reduction, could we also start with the same variant of the Hamilton path problem as in [PV84], but embed the input graph into another tessellation of the plane than the rectangular grid? Tessellations with vertices of degree five or higher would maybe provide us with more space to place suburbs. We limited our considerations thereby to regular and semi-regular tessellations, that is, tilings of the plane either consisting of one or several types of regular polygons, respectively.

We quickly abandoned this approach for the following two reasons: Every embedding similar to the one described in [PV84] most probably uses scaling as a way of creating enough place for suburbs and constructs such as those called dumbbells and tentacles in [PV84]. Scaling though is only possible in an easy way in tessellations with grid structure – i.e., in the rectangular grid, the one used in [PV84], and the regular tessellation made of equilateral triangles, that is shown in Figure 4.4.

![Image of regular tessellation consisting of equilateral triangles](http://commons.wikimedia.org/wiki/File:Tiling.Regular.3-6_.Triangular.svg)

Figure 4.4: The regular tessellation consisting of equilateral triangles.\(^2\)

\(^2\)Image source: [http://commons.wikimedia.org/wiki/File:Tiling.Regular.3-6_.Triangular.svg](http://commons.wikimedia.org/wiki/File:Tiling.Regular.3-6_.Triangular.svg), image licensed under the Creative Commons Attribution ShareAlike 3.0 license.
In the latter though, placing suburbs on grid edges in between two grid vertices is not possible without interfering with neighbouring grid edges. Placing suburbs on the free places of the grid instead makes us face the problem that a suburb cannot be clearly assigned to a grid vertex anymore, as it might be in equal distance to multiple grid vertices. Therefore, using the regular tessellation consisting of triangles did not seem to be a promising approach. We did not follow the idea of considering semiregular tessellations instead of grids, incorporating degree-five vertices, as shown in Figure 4.5— which might provide more space to place suburbs while not containing problematic 60° angles, but might make scaling more complicated than a grid structure— any further.

Weakening the Grid Structures – Mutated Tentacles. As the attempts of placing two suburbs per vertex on the rectangular grid have shown, 180° might be just marginally not enough room to place two suburbs. As most vertices of the grid-embedded graph are part of ladder-like “tentacles,” would it be of use to add dents to this tentacles such that a range of slightly more than 180° around every tentacle vertex would be free of neighbours? Dents can be constructed by placing additional vertices in the interior of the tentacle deflecting the original outer edges. In order to preserve the equal length of return path and cross path through tentacles (cf. [PV84]), not each exterior tentacle edge could be dented, but both sides would have to be dented alternately, as shown in Figure 4.6. This approach fails as the interior edges of tentacles only allow for relatively small dents; the resulting neighbour-free space does not suffice for the placement of two suburbs per vertex, see Figure 4.7.

Figure 4.5: The three semiregular tessellations including vertices of degree $\geq 5$. ³

³Image sources: http://commons.wikimedia.org/wiki/File:Tiling_Semiregular_3-3-3-3-6_Snub_Hexagonal.svg, http://commons.wikimedia.org/wiki/File:Tiling_Semiregular_3-3-4-3-4_Snub_Square.svg, http://commons.wikimedia.org/wiki/File:Tiling_Semiregular_3-3-3-4-4_Elongated_Triangular.svg, images licensed under the Creative Commons Attribution ShareAlike 3.0 license.
4. Other Approaches Taken

(a) Traditional "tentacles" with return (middle) and cross path (right)

(b) "Mutated tentacles" with return (middle) and cross path (right)

Figure 4.6: Old and considered new form of ladder-like connection structures, the latter using additional points inside the structure to dent every second outer line, thereby creating more space for suburbs on the outside of the structure.

Figure 4.7: Dents cannot make sufficient room for two suburbs. The dents possible are relatively small (left): Point d can only be placed as far in the tentacle structure as that angles $\alpha = \beta$ do not become smaller than or equal to 75°. The new area in which suburbs can be placed, the unshaded area on the right, still does not allow for a triangle $\triangle bpq$ in which $\{p, q\}$ is the longest edge.
Recursive Structures Enforcing Additional Degree Two for Every Vertex on the Grid.

All approaches up to here only relied on EMST properties when adding suburbs and did not make use of the degree restriction in E-D4-MSTs. What if we added three sub-suburbs to each suburb? Could we avoid that the suburbs would connect to each other in an E-D4-MST, even if they were closer to each other than to the grid vertex by blocking three of the four connection "sockets" by sub-suburbs that are even closer? And thus place two suburbs in a range smaller than 60°?

Let b be the grid vertex, a and c its neighbours on the grid, p and q b’s suburbs, s₁, s₂, s₃ p’s sub-suburbs and t₁, t₂, t₃ q’s sub-suburbs, for example as depicted in Figure 4.8.

![Figure 4.8: Two suburbs for a vertex on the grid with the aid of sub-suburbs?](image)

1. p, q, s₁, s₂, s₃, t₁, t₂, t₃ all have to be at a large enough distance to a and c, which implies that they all must lie within a range of 60° around b, more precisely in the unshaded area depicted in Figure 4.3.

2. For i, j ∈ {1, 2, 3}, i ≠ j, all sᵢ have to have a larger distance to any sⱼ, so that in any EMST, they are a leaf connected to p; the same holds for q and all tᵢ, tⱼ, i, j ∈ {1, 2, 3}, i ≠ j.

3. For i, j ∈ {1, 2, 3}, all sᵢ have to have a larger or equal distance to any tⱼ than p has to q.

4. For i ∈ {1, 2, 3}, all sᵢ have to have a larger distance to b than p has; the same holds for all tᵢ and q.

5. The cost of connecting p and q both with b instead of connecting them directly, i.e. \( \max\{w(b, p), w(b, q)\} - w(p, q) \), must be cheaper than connecting a pair of sub-suburbs, w.l.o.g. s₁, s₂, i.e. \( w(s₁, s₂) - \max\{w(p, s₁), w(p, s₂)\} \). Otherwise, if \{s₁, s₂\} was the cheaper connection, it would cancel degree 4 at either p or q and allow the connection \{p, q\}.

29
Because of Condition 1, it seems that Condition 5, given the other conditions, is almost impossible to fulfill: the larger the distances between the sub-suburbs are made in order to increase $w(s_i, s_j) - \max\{w(p, s_i), w(p, s_j)\}$, the more space the suburbs need in the 60° range. This requires $\{p, q\}$ to become shorter, which in turn makes $\max\{w(b, p), w(b, q)\} - w(p, q)$ larger. We did not find a setting that met all requirements.
Open Questions

As mentioned in the introduction, our aim was to close the gap between the known and differing complexity class membership for E-Dk-MST problems for \( k = 3 \) and \( k = 5 \). On the one hand, this newly gained knowledge raises a couple of questions that would be interesting to investigate further and that were beyond the scope of this semester thesis. On the other hand, an important question that we encountered during our work is the one whether the decision versions of the E-Dk-MST problems are in NP.

Membership in NP. In the first version of our proof, we claimed that the E-D4-MST problem "obviously is in NP". It turned out that this is not the case, and it is actually not even the case for the decision version of the EMST problem. While for the optimization variant of the EMST problem, it is easy to verify that a given Euclidean spanning tree is indeed the shortest possible spanning tree of a point set – to this aim, we only need to sort distances between pairs of points, which can be done by comparing the squared distances, that are, being sums of squares of integers, easy to compare – it is not known whether it is easy or not to compare the sum of distances, i.e. the weight of the spanning tree, with an integer, which is required in the decision version of the problem. The problem that underlies this is better known as the "Sum of Square Roots Problem", Problem 33 [DMO06] in "The Open Problems Project" [DMO]. If one wants to compare the sum of \( k \) square roots of integers no larger than \( n \) with an integer, it is generally not known how many bits of the solution need to be computed in order to decide which number is larger, and particularly, it is not known whether the number of necessary bits is bounded above by a polynomial in \( k \) and \( \log n \). It is therefore an open question whether the E-Dk-MST problems lie in NP.
5. Open Questions

Existence of a PTAS. Given that the E-D4-MST problem is NP-hard, the next question in terms of complexity is whether E-D4-MSTs can be efficiently approximated. Can Arora’s and Chan’s QPTAS (AC04, see also Section 2.2) be improved to a PTAS? The same question is open for the E-D3-MST problem. If the answer is “no” to both questions: What properties do make the E-Dk-MST problem inapproximable for \( k = 3 \) and \( k = 4 \), while there exists a PTAS for \( k = 2 \)? The E-D3-MST and the E-D4-MST problem seem structurally quite similar. The reduction presented in this report works for both, \( k = 3 \) and \( k = 4 \); see Appendix C for a vertex gadget that can be used to produce a proof for the case \( k = 3 \) analogous to the one for \( k = 4 \). For \( k = 2 \), the reduction does not work, though. It would be surprising thus if the answer to the above question differed for the E-D3-MST and the E-D4-MST problem, say if there existed a PTAS for the E-D3-MST but not for the E-D4-MST problem. It would then of course be intriguing to find out what tells apart E-D3-MSTs from E-D4-MSTs.

Our reduction maps cubic planar graphs with a vertex cover of size \( \leq k \) to E-D4-MSTs of size \( \leq \psi(P) + k \cdot \varepsilon \), while it maps cubic planar graphs with no vertex cover of size \( \leq k \) to E-D4-MSTs of size \( \geq \psi(P) + (k + 1) \cdot \varepsilon \), thereby producing a gap between “yes” and “no” instances of the reduced decision problem of size \( \geq \varepsilon \), i.e. a gap of constant size, independent of the size of a solution to a “yes” instance. This means that the rate of approximation that can be shown to be impossible to achieve in polynomial time with this proof sketch decreases with the size of the problem instance. A reduction that proves inapproximability to a constant factor in turn would have to map an instance of a decision problem to which the answer is “yes” to an optimization problem instance with solution of size \( \leq \ell \), while mapping an instance with answer “no” to an instance with solution of size \( \geq (1 + c) \cdot \ell \), i.e., the reduction would have to produce a gap of size proportional to the size of the solution to a “yes” instance. Our reduction thus does not trivially yield an inapproximability proof. Note that the same holds for the reduction presented in [PV84], which also produces a gap of constant size, so even a reduction based on the approaches presented in Chapter 4 would not yield a proof of inapproximability to a constant factor.

Approximation Algorithms and Preciser (Worst Case) Bounds for \( \tau_k \). Recall that \( \tau_k(P) = \psi_k(P)/\psi(P) \), where \( \psi(P) \) is the weight of an EMST and \( \psi_k(P) \) the weight of an E-Dk-MST for a point set \( P \) in the plane (cf. Section 2.2). The currently known best upper bounds for \( \tau_3 \) and \( \tau_4 \) are \( \tau_3 \leq 1.143 \) and \( \tau_4 \leq 1.402 \). In terms of lower bounds for the worst-case ratio of the weight of the minimum degree-K spanning tree to the weight of the minimum spanning tree, Fekete et al. showed in [FKK97] that there exist point sets such that \( 1.1035 < \tau_3 \) and \( 1.035 < \tau_4 \). They conjectured that these lower bounds are tight, and an inspection of the corresponding constructions does make this conjecture seem
plausible. Can the upper bounds be improved? Or even the lower bounds? If the E-D4-MST problem and/or the E-D3-MST problem turn out to be inapproximable to a constant factor, what are the best possible polynomial time approximation algorithms?

Boundary of Intractability in Three and More Dimensions. Our proof has shown that in two dimensions, the boundary between polynomial time solvable (in the optimization variant) and NP-hard lies exactly between E-Dk-MSTs that have the same length as normal EMSTs – e.g. E-D5-MSTs – and E-Dk-MSTs that can have different length – e.g. E-D4-MSTs. Where does this boundary lie for E-Dk-MSTs in higher dimensions, especially in three dimensions? Are E-Dk-MSTs in higher dimensions easy to find if they have the length of an EMST, and hard to find if not?
Chapter 6

Conclusion

After presenting the Euclidean degree-4 minimum spanning tree problem, a proof for its NP-harness, an overview on other, less successful proof approaches and a couple of ideas what questions might be interesting to pursue from here, this report comes to an end with a few concluding remarks.

The E-D4-MST problem turned out to be a great choice for a semester thesis topic, as it provided me with a very interesting and industrious time, with a good experience – trying to keep one’s nerve while not having any results during the first six out of (intended) eight weeks – and a surprising and rewarding outcome. I also was lucky with, and would like to thank first and foremost my supervisor, who not only coached me during numerous hours but also contributed largely to the proof. I also would like to thank the Center for Algorithms, Discrete Mathematics and Optimization (CADMO), part of the Institute for Theoretical Computer Science at ETH, for providing its master students with great workspace, my team at IBM Research, my internship company, for letting me take most of my free days already in my first month of employment in order to finish our SoCG submission [FH09] in time, Professor Welzl for giving me the great opportunity to go to SoCG 2009, Thomas Rast for the CADMO thesis template and, finally, my family, friends and flat mates for top-notch moral support.
Appendix A

Task Description

Semester / Bachelor / Diploma / Master Thesis

Agoraphobia: Well connected with only four neighbors?

Suppose there are \( n \) sites that should be connected in such a way that within the network every site can reach any other site - not necessarily directly but possibly via a sequence of sites. As every link between two sites incurs some cost to set up, we should build the network with as few links as possible and therefore select a set of links which form a spanning tree on the sites. Any such spanning tree \( T \) consists of \( n - 1 \) edges and there are \( n^{n-2} \) possible choices for \( T \). However, in reality the costs for setting up a link between two sites are usually not the same for every pair of sites. One possible model is to think of the sites as points in the Euclidean plane and take their Euclidean distance as a cost measure for setting up the corresponding link in the network.

One of the classical results in computational geometry is that for any set \( P \) of \( n \) points in the plane some spanning tree which minimizes the sum of the Euclidean distances along its edges appears as a subgraph of any Delaunay triangulation for \( P \). In other words, a Euclidean minimum spanning tree for \( n \) points can be obtained in \( O(n \log n) \) time. But what if there are additional requirements on the network, other than just being cheap and connected? For example, it may be desirable to avoid vertices of large degree because such a vertex is very likely to become a bottleneck.

On one hand it is known that every planar point set admits a Euclidean minimum spanning tree of maximum vertex degree five [2]. On the other hand it is NP-complete to decide whether a given set of points has a Euclidean minimum spanning tree of maximum vertex degree three [3]. However, the case of maximum vertex degree four is still widely open. Chan [1] gave approximative
A. Task Description

Figure A.1: A Euclidean minimum spanning tree with maximum vertex degree four.

Bounds on how much worse a minimum spanning tree of bounded degree is compared to the unrestricted tree, which is at most a factor of 1.402 times worse for degree three and at most 1.143 times worse for degree four.

Goal. The goal of this project is to gain some insight into the problem of constructing Euclidean minimum spanning trees of bounded degree. Ideally the complexity of the degree four case can be settled or some progress can be made regarding the approximation ratios given by Chan, either by improving the upper bound or by giving better lower bound constructions. These objectives may turn out to be rather ambitious, hence it may be a good idea to also look at variations of the problem. For example, does it help if we allow a certain number of high degree vertices?

Prerequisites. Proficient knowledge in computational geometry, design and analysis of algorithms, and theory of computing.

References.


Contact Person: Michael Hoffmann, lastname@inf.ethz.ch.
Figure B.1: Approximate distances along Delaunay edges in a vertex gadget.
Figure B.2: Insertion costs of Delaunay edges in a vertex gadget.
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d^2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.28</td>
<td>0.48</td>
<td>0.27</td>
<td>0.38</td>
<td>0.38</td>
<td>0.29</td>
<td>0.37</td>
<td>0.44</td>
<td>0.78</td>
</tr>
<tr>
<td>2</td>
<td>0.80</td>
<td>0.28</td>
<td>0.77</td>
<td>0.33</td>
<td>0.37</td>
<td>0.43</td>
<td>0.45</td>
<td>0.44</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>2.29</td>
<td>0.79</td>
<td>0.27</td>
<td>0.33</td>
<td>0.36</td>
<td>0.27</td>
<td>0.45</td>
<td>0.45</td>
<td>0.39</td>
</tr>
<tr>
<td>4</td>
<td>0.73</td>
<td>0.77</td>
<td>0.50</td>
<td>0.71</td>
<td>1.08</td>
<td>1.17</td>
<td>1.08</td>
<td>2.22</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.48</td>
<td>1.15</td>
<td>1.13</td>
<td>0.14</td>
<td>1.16</td>
<td>1.11</td>
<td>1.04</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.85</td>
<td>1.89</td>
<td>2.04</td>
<td>0.34</td>
<td>0.13</td>
<td>0.42</td>
<td>0.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.38</td>
<td>2.05</td>
<td>1.54</td>
<td>0.29</td>
<td>0.13</td>
<td>0.46</td>
<td>0.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2.00</td>
<td>2.44</td>
<td>1.52</td>
<td>0.48</td>
<td>0.32</td>
<td>0.53</td>
<td>0.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.83</td>
<td>4.05</td>
<td>5.19</td>
<td>2.04</td>
<td>1.48</td>
<td>2.19</td>
<td>0.72</td>
<td>1.25</td>
<td>1.66</td>
</tr>
<tr>
<td>10</td>
<td>2.93</td>
<td>4.81</td>
<td>4.19</td>
<td>1.80</td>
<td>1.27</td>
<td>1.39</td>
<td>0.69</td>
<td>0.67</td>
<td>0.68</td>
</tr>
<tr>
<td>11</td>
<td>5.06</td>
<td>8.50</td>
<td>8.40</td>
<td>4.52</td>
<td>3.62</td>
<td>4.13</td>
<td>2.42</td>
<td>2.75</td>
<td>2.90</td>
</tr>
<tr>
<td>12</td>
<td>6.62</td>
<td>9.07</td>
<td>7.43</td>
<td>4.58</td>
<td>3.78</td>
<td>3.60</td>
<td>2.85</td>
<td>2.56</td>
<td>2.27</td>
</tr>
<tr>
<td>13</td>
<td>4.80</td>
<td>5.69</td>
<td>3.70</td>
<td>2.33</td>
<td>1.92</td>
<td>1.42</td>
<td>1.65</td>
<td>1.06</td>
<td>0.68</td>
</tr>
<tr>
<td>14</td>
<td>3.06</td>
<td>2.81</td>
<td>1.21</td>
<td>0.85</td>
<td>0.80</td>
<td>0.29</td>
<td>1.07</td>
<td>0.46</td>
<td>2.05</td>
</tr>
<tr>
<td>15</td>
<td>6.40</td>
<td>4.66</td>
<td>1.69</td>
<td>2.82</td>
<td>3.03</td>
<td>1.84</td>
<td>3.84</td>
<td>2.60</td>
<td>1.96</td>
</tr>
<tr>
<td>16</td>
<td>6.56</td>
<td>6.08</td>
<td>3.07</td>
<td>3.04</td>
<td>2.90</td>
<td>1.85</td>
<td>3.14</td>
<td>2.05</td>
<td>1.38</td>
</tr>
<tr>
<td>17</td>
<td>9.09</td>
<td>9.66</td>
<td>6.22</td>
<td>5.25</td>
<td>4.77</td>
<td>3.70</td>
<td>4.50</td>
<td>3.40</td>
<td>2.60</td>
</tr>
<tr>
<td>18</td>
<td>8.79</td>
<td>4.50</td>
<td>5.72</td>
<td>5.78</td>
<td>4.12</td>
<td>6.46</td>
<td>4.79</td>
<td>3.77</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>5.57</td>
<td>6.37</td>
<td>6.28</td>
<td>4.62</td>
<td>6.73</td>
<td>5.06</td>
<td>3.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>6.20</td>
<td>6.61</td>
<td>6.41</td>
<td>4.80</td>
<td>6.68</td>
<td>5.06</td>
<td>3.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>6.19</td>
<td>7.59</td>
<td>7.61</td>
<td>5.70</td>
<td>8.29</td>
<td>6.04</td>
<td>5.19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>7.48</td>
<td>8.42</td>
<td>8.28</td>
<td>6.37</td>
<td>8.71</td>
<td>6.82</td>
<td>5.55</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>7.29</td>
<td>8.87</td>
<td>8.90</td>
<td>6.82</td>
<td>9.63</td>
<td>7.57</td>
<td>6.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>6.84</td>
<td>9.89</td>
<td>9.38</td>
<td>7.17</td>
<td>8.52</td>
<td>7.24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table B.1: Point to point distances within a vertex gadget (Part 1). Above the diagonal the Euclidean distance $d(p_i, p_j)$ is shown, approximated up to three digits after the decimal point; below the diagonal the exact squared Euclidean distance $d^2(p_i, p_j)$ is shown. If the distance between two points is 100 or more, the corresponding entry is marked $\infty$. Entries shown in boldface correspond to EMST-edges. Entries with a shaded background correspond to non-EMST edges that are no longer than the longest EMST-edge ($\sqrt{769}$) plus the smallest insertion cost $\varepsilon$ (cf. Lemma 3.6).
Table B.2: Point to point distances within a vertex gadget (Part 2).
Above the diagonal the Euclidean distance $d(p_i, p_j)$ is shown, approximated up to three digits after the decimal point; below the diagonal the exact squared Euclidean distance $d^2(p_i, p_j)$ is shown. If the distance between two points is 100 or more, the corresponding entry is marked $\infty$. Entries shown in boldface correspond to EMST-edges. Entries with a shaded background correspond to non-EMST edges that are no longer than the longest EMST-edge ($\sqrt{\epsilon}$) plus the smallest insertion cost $\epsilon$ (cf. Lemma 3.6).

<table>
<thead>
<tr>
<th></th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>42.755</td>
<td>54.203</td>
<td>71.168</td>
<td>81.400</td>
<td>69.339</td>
<td>55.317</td>
<td>80.056</td>
<td>81.043</td>
<td>95.383</td>
</tr>
<tr>
<td>d</td>
<td>66.370</td>
<td>69.375</td>
<td>92.195</td>
<td>95.273</td>
<td>75.451</td>
<td>53.075</td>
<td>68.308</td>
<td>78.006</td>
<td>98.310</td>
</tr>
<tr>
<td>d</td>
<td>72.069</td>
<td>64.776</td>
<td>91.678</td>
<td>86.209</td>
<td>60.876</td>
<td>34.785</td>
<td>41.146</td>
<td>55.443</td>
<td>78.892</td>
</tr>
<tr>
<td>d</td>
<td>45.188</td>
<td>42.426</td>
<td>67.268</td>
<td>67.675</td>
<td>48.270</td>
<td>29.154</td>
<td>53.150</td>
<td>55.154</td>
<td>72.470</td>
</tr>
<tr>
<td>d</td>
<td>38.470</td>
<td>35.693</td>
<td>60.207</td>
<td>61.465</td>
<td>43.863</td>
<td>28.284</td>
<td>55.108</td>
<td>53.851</td>
<td>69.065</td>
</tr>
<tr>
<td>d</td>
<td>46.861</td>
<td>37.336</td>
<td>64.288</td>
<td>60.016</td>
<td>37.735</td>
<td>17.088</td>
<td>42.953</td>
<td>43.081</td>
<td>60.876</td>
</tr>
<tr>
<td>d</td>
<td>26.925</td>
<td>26.400</td>
<td>49.010</td>
<td>53.338</td>
<td>40.706</td>
<td>32.756</td>
<td>62.032</td>
<td>56.080</td>
<td>67.119</td>
</tr>
<tr>
<td>d</td>
<td>35.355</td>
<td>26.529</td>
<td>50.596</td>
<td>32.649</td>
<td>21.587</td>
<td>51.078</td>
<td>45.276</td>
<td>58.309</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>0</td>
<td>25.019</td>
<td>29.068</td>
<td>47.010</td>
<td>50.159</td>
<td>54.918</td>
<td>84.905</td>
<td>72.801</td>
<td>75.166</td>
</tr>
<tr>
<td>d</td>
<td>626</td>
<td>0</td>
<td>27.730</td>
<td>27.202</td>
<td>25.495</td>
<td>37.443</td>
<td>65.764</td>
<td>49.658</td>
<td>50.159</td>
</tr>
<tr>
<td>d</td>
<td>845</td>
<td>769</td>
<td>0</td>
<td>28.160</td>
<td>47.634</td>
<td>65</td>
<td>92.314</td>
<td>73.246</td>
<td>65.459</td>
</tr>
<tr>
<td>d</td>
<td>2210</td>
<td>740</td>
<td>793</td>
<td>0</td>
<td>28.460</td>
<td>53.235</td>
<td>74.732</td>
<td>51.865</td>
<td>38.470</td>
</tr>
<tr>
<td>d</td>
<td>2516</td>
<td>650</td>
<td>2269</td>
<td>810</td>
<td>0</td>
<td>26.305</td>
<td>46.529</td>
<td>25.612</td>
<td>26.570</td>
</tr>
<tr>
<td>d</td>
<td>3016</td>
<td>1402</td>
<td>4225</td>
<td>2834</td>
<td>692</td>
<td>0</td>
<td>30.016</td>
<td>26</td>
<td>45.453</td>
</tr>
<tr>
<td>d</td>
<td>7209</td>
<td>4325</td>
<td>8522</td>
<td>5585</td>
<td>2165</td>
<td>901</td>
<td>0</td>
<td>25.079</td>
<td>50.089</td>
</tr>
<tr>
<td>d</td>
<td>5300</td>
<td>2466</td>
<td>5365</td>
<td>2690</td>
<td>656</td>
<td>676</td>
<td>629</td>
<td>0</td>
<td>25.495</td>
</tr>
<tr>
<td>d</td>
<td>5650</td>
<td>2516</td>
<td>4285</td>
<td>1480</td>
<td>706</td>
<td>2066</td>
<td>2509</td>
<td>650</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>9797</td>
<td>5785</td>
<td>9810</td>
<td>5725</td>
<td>2665</td>
<td>2305</td>
<td>680</td>
<td>697</td>
<td>1657</td>
</tr>
<tr>
<td>d</td>
<td>9616</td>
<td>5482</td>
<td>9005</td>
<td>4874</td>
<td>2372</td>
<td>2600</td>
<td>1201</td>
<td>676</td>
<td>1066</td>
</tr>
<tr>
<td>d</td>
<td>9169</td>
<td>5069</td>
<td>8168</td>
<td>4141</td>
<td>2089</td>
<td>2725</td>
<td>1618</td>
<td>689</td>
<td>689</td>
</tr>
<tr>
<td>d</td>
<td>22</td>
<td>$\infty$</td>
<td>7184</td>
<td>$\infty$</td>
<td>6596</td>
<td>3554</td>
<td>3490</td>
<td>1413</td>
<td>1274</td>
</tr>
<tr>
<td>d</td>
<td>23</td>
<td>$\infty$</td>
<td>6980</td>
<td>$\infty$</td>
<td>5800</td>
<td>3370</td>
<td>3946</td>
<td>2125</td>
<td>1394</td>
</tr>
<tr>
<td>d</td>
<td>24</td>
<td>$\infty$</td>
<td>8298</td>
<td>$\infty$</td>
<td>7442</td>
<td>4328</td>
<td>4372</td>
<td>1961</td>
<td>1800</td>
</tr>
<tr>
<td>d</td>
<td>25</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>6370</td>
<td>5018</td>
<td>1745</td>
<td>2938</td>
</tr>
<tr>
<td>d</td>
<td>26</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>8725</td>
<td>8345</td>
<td>4274</td>
<td>4813</td>
</tr>
<tr>
<td>d</td>
<td>27</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>$\infty$</td>
<td>9738</td>
<td>7092</td>
<td>8144</td>
<td>5069</td>
</tr>
<tr>
<td>d² \ d</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td>-------</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>2</td>
<td>93.776</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>3</td>
<td>67.119</td>
<td>74.632</td>
<td>78.746</td>
<td>78.714</td>
<td>86.533</td>
<td>85.428</td>
<td>78.638</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>4</td>
<td>75.663</td>
<td>79.812</td>
<td>81.344</td>
<td>87.132</td>
<td>91.760</td>
<td>94.201</td>
<td>94.762</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>5</td>
<td>76.059</td>
<td>79.246</td>
<td>80.081</td>
<td>87.281</td>
<td>91.005</td>
<td>94.339</td>
<td>96.881</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>6</td>
<td>64.195</td>
<td>67.911</td>
<td>69.289</td>
<td>75.538</td>
<td>79.812</td>
<td>82.607</td>
<td>84.723</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>7</td>
<td>80.399</td>
<td>82.054</td>
<td>81.743</td>
<td>91.087</td>
<td>93.343</td>
<td>98.045</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>8</td>
<td>69.260</td>
<td>71.175</td>
<td>71.196</td>
<td>80.024</td>
<td>82.607</td>
<td>87.005</td>
<td>92.347</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>9</td>
<td>61.400</td>
<td>63.126</td>
<td>63.150</td>
<td>72.062</td>
<td>74.545</td>
<td>79.025</td>
<td>85.094</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>10</td>
<td>98.979</td>
<td>98.061</td>
<td>95.754</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>11</td>
<td>76.059</td>
<td>74.040</td>
<td>71.196</td>
<td>84.758</td>
<td>83.546</td>
<td>91.093</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>12</td>
<td>99.045</td>
<td>94.894</td>
<td>90.376</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
<td>∞</td>
</tr>
<tr>
<td>13</td>
<td>75.663</td>
<td>69.814</td>
<td>64.350</td>
<td>81.215</td>
<td>76.157</td>
<td>86.267</td>
<td>∞</td>
<td>∞</td>
<td>98.681</td>
</tr>
<tr>
<td>14</td>
<td>51.623</td>
<td>48.703</td>
<td>45.705</td>
<td>59.615</td>
<td>58.051</td>
<td>65.787</td>
<td>79.812</td>
<td>93.407</td>
<td>84.214</td>
</tr>
<tr>
<td>15</td>
<td>48.010</td>
<td>50.990</td>
<td>52.201</td>
<td>59.076</td>
<td>62.817</td>
<td>66.121</td>
<td>70.837</td>
<td>91.350</td>
<td>90.244</td>
</tr>
<tr>
<td>16</td>
<td>26.076</td>
<td>34.655</td>
<td>40.224</td>
<td>37.589</td>
<td>46.097</td>
<td>44.283</td>
<td>41.773</td>
<td>65.375</td>
<td>71.196</td>
</tr>
<tr>
<td>17</td>
<td>25.400</td>
<td>26</td>
<td>26.248</td>
<td>35.693</td>
<td>37.336</td>
<td>42.426</td>
<td>54.203</td>
<td>69.375</td>
<td>64.776</td>
</tr>
<tr>
<td>18</td>
<td>40.706</td>
<td>32.649</td>
<td>26.248</td>
<td>43.863</td>
<td>37.735</td>
<td>48.270</td>
<td>69.339</td>
<td>75.451</td>
<td>60.876</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>11.180</td>
<td>19.026</td>
<td>11.704</td>
<td>20.615</td>
<td>18.681</td>
<td>29.068</td>
<td>43.566</td>
<td>45.177</td>
</tr>
<tr>
<td>20</td>
<td>125</td>
<td>0</td>
<td>8.062</td>
<td>11.401</td>
<td>12.083</td>
<td>17.088</td>
<td>37.121</td>
<td>44.777</td>
<td>49.396</td>
</tr>
<tr>
<td>21</td>
<td>362</td>
<td>65</td>
<td>0</td>
<td>18.027</td>
<td>12.369</td>
<td>22.022</td>
<td>44.777</td>
<td>49.396</td>
<td>39.051</td>
</tr>
<tr>
<td>22</td>
<td>137</td>
<td>130</td>
<td>325</td>
<td>0</td>
<td>12.165</td>
<td>7.071</td>
<td>27.202</td>
<td>33.837</td>
<td>33.615</td>
</tr>
<tr>
<td>23</td>
<td>425</td>
<td>146</td>
<td>153</td>
<td>148</td>
<td>0</td>
<td>12.083</td>
<td>38.470</td>
<td>37.802</td>
<td>27.459</td>
</tr>
<tr>
<td>24</td>
<td>349</td>
<td>292</td>
<td>485</td>
<td>50</td>
<td>146</td>
<td>0</td>
<td>27.018</td>
<td>27.730</td>
<td>27.202</td>
</tr>
<tr>
<td>25</td>
<td>845</td>
<td>1378</td>
<td>2005</td>
<td>740</td>
<td>1480</td>
<td>730</td>
<td>0</td>
<td>28.301</td>
<td>47.853</td>
</tr>
<tr>
<td>26</td>
<td>1898</td>
<td>2005</td>
<td>2440</td>
<td>1145</td>
<td>1429</td>
<td>769</td>
<td>801</td>
<td>0</td>
<td>28.160</td>
</tr>
<tr>
<td>27</td>
<td>2041</td>
<td>1544</td>
<td>1525</td>
<td>1130</td>
<td>754</td>
<td>740</td>
<td>2290</td>
<td>793</td>
<td>0</td>
</tr>
</tbody>
</table>

Table B.3: Point to point distances within a vertex gadget (Part 3). Above the diagonal the Euclidean distance $d(p_i, p_j)$ is shown, approximated up to three digits after the decimal point; below the diagonal the exact squared Euclidean distance $d^2(p_i, p_j)$ is shown. If the distance between two points is 100 or more, the corresponding entry is marked ∞. Entries shown in boldface correspond to EMST-edges. Entries with a shaded background correspond to non-EMST edges that are no longer than the longest EMST-edge ($\sqrt{769}$) plus the smallest insertion cost $\varepsilon$ (cf. Lemma 3.6).
A Vertex Gadget for $k = 3$

Figure C.1: A point set with coordinates that serves as a vertex gadget if one wants to construct an analogous proof to the one presented in Chapter 3 for $k = 3$ instead of $k = 4$. All solid black edges will appear in every EMST of the vertex gadget. The two dotted edges are of equal length and equivalent choices in any EMST. The blue vertices are bond vertices, serving as a possible connection point to optional paths. The red vertices are glue vertices, providing connection points to forced paths. Shown in grey is a possible setting of optional and forced paths surrounding the vertex gadget, the slanted edges providing a fixed and large enough distance between forced and optional paths. Finally, the dashed red edge is the cheapest insertion of a non-EMST edge into the EMST, and this insertion is also sufficient to remove degree three locally (see Figure C.3).
C. A Vertex Gadget for $k = 3$

Figure C.2: The gadget for $k = 3$ in action. If no connection to a optional path needs to be provided, there are no degree three vertices in the EMST locally. Thus, the E-D3-MST is locally identical to the EMST.

Figure C.3: The gadget for $k = 3$ in action. If at least one connection to an optional path is needed, there are degree three vertices in the EMST in the gadget at hand. These degree three vertices, induced from up to three connections to optional paths, above marked in blue, can be removed by one local deviation from the EMST: the edge marked in red. Above shown is the resulting E-D3-MST.
Bibliography


