BRNO UNIVERSITY OF TECHNOLOGY

Faculty of Mechanical Engineering

RNDr. Ing. Miloš Šeda, Ph.D.

STEINER TREE PROBLEMS AND APPROXIMATION METHODS FOR THEIR SOLUTION

HABILITATION THESIS

VUTIUM

Brno, October 2001

Keywords: Steiner tree, minimum spanning tree, Steiner ratio, NP-completeness, approximation, heuristic

ISBN 80-214-1988-1
ISSN 1213-418X

Contents

Although this may seem a paradox, all exact science is dominated by the idea of approximation.
Bertrand Russell (1872-1970)

1 Introduction

In practice, we frequently solve problems based on finding minimal networks. The criterion of minimality can be represented by the total costs for the implementation of the network or by the total length of connections. These problems include, e.g. the problem of finding optimal location of a source with respect to given sinks. An important class of optimization problems is represented by problems minimizing the total lengths of connections among vertices of a given graph [58]. These connections can include additional points in the plane (or vertices of a graph, respectively) if it leads to better solution. These problems are called *Euclidean Steiner tree problem* and *Steiner tree problem in graphs* (also called *network Steiner tree problem*). A special case of the Euclidean problem, called (*rectilinear Steiner tree problem*), requires all connections leading only in the horizontal or vertical direction.

This paper focuses on the Steiner problems. They were selected because of their many practical applications [53] and also for their NP-completeness. This implies the necessity of using the approximation of heuristic approaches to get good solutions for larger instances. The natural way of approximation is finding the minimum spanning tree and the process of its gradual improvement. Besides approximation algorithms, heuristics (mainly stochastic heuristics) may be used. They do not guarantee bounds for the worst case as the approximation methods do but seem to provide good results, especially for the network Steiner tree problem.

In this paper, we propose a hybrid combination of approximation and heuristic approaches to obtain solutions guaranteeing a closeness to optimal solutions.

1.1 The aims of the paper

The aims of the habilitation work [61] was a detailed study of the Steiner problems. It includes:

- characterisation of their complexity,
- specification of Steiner tree properties
- formulation of the mathematical model for the network Steiner tree problem,
- verification of the proposed model by a professional software optimization tool,
- description of approximation approaches and proofs of bounds for the approximation by minimum spanning tree, and
- a discussion of computational results achieved by stochastic heuristic methods.

2 Network Steiner tree problem

The *Network Steiner tree problem* (NSTP) (or *Steiner tree problem in graphs*) [12], [28], [68] is concerned with connecting a subset of vertices at a minimal cost. More precisely, given an undirected connected graph $G = (V, E)$ with vertex set V, edge set E, nonnegative weights associated with the edges, and a subset B of V (called *terminals* or *customer vertices*), the problem is to find a subgraph T that connects the vertices in B so that the sum of the weights of the edges in T is minimised. It is obvious that the solution is always a tree and it is called a *Steiner minimum tree* for B in G.

Applications of the NSTP are frequently found in the layout of connection structures in networks and circuit design [7], [9], [16]. Their common feature is that of connecting together a set of terminals (communications sites or circuits components) by a network of the minimal total length.

If $|B| = 2$ then the problem reduces to the *shortest path problem* and can be solved by Dijkstra's algorithm. In the case of $B = V$ the NSTP reduces to the *minimum spanning tree* (MST) *problem* and can be solved by Jarník's (Prim's), Borůvka's or Kruskal's algorithm. All these algorithms are polynomial. However, in the general case the NSTP is NP-complete [34], [44] and therefore it cannot be solved exactly for larger instances, i.e. heuristic or approximation methods must be used. Normally a Steiner minimum tree is not a minimum spanning tree only, it can also span some nonterminals, called *Steiner vertices*, as shown in Fig. 1.

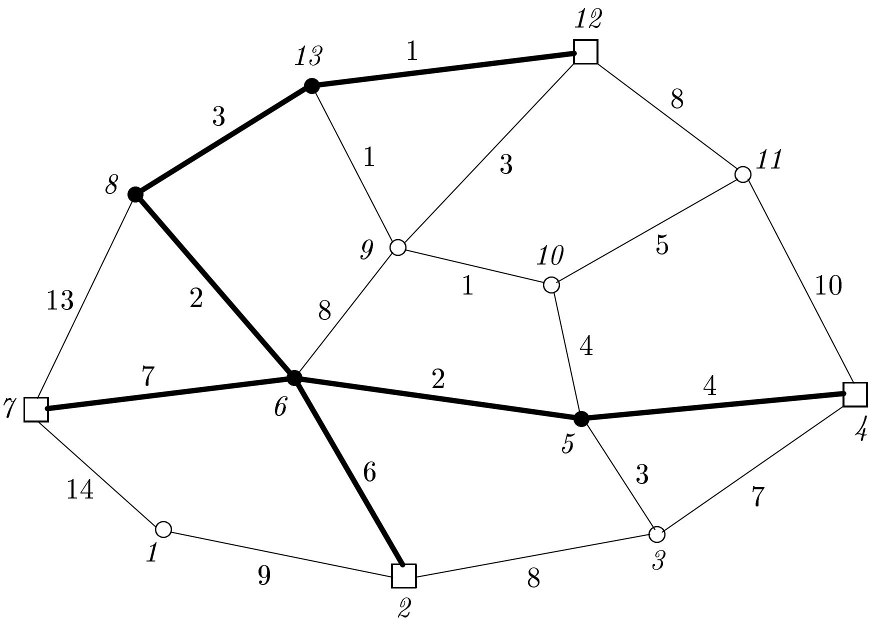

Figure 1: An example of the network Steiner tree problem □ = terminals, • = Steiner vertices)

2.1 Mathematical formulation

Let $V = \{1, 2, \ldots, n\}$ and S be a set of Steiner vertices. For every edge (i, j), c_{ij}, $c_{ij} \geq 0$ is a weight of the edge. The aim is to find a connected graph $G' = (B \cup S, E')$ (Steiner tree), $E' \subset E$, for the sum of weights to be minimal.

In other words, the Steiner minimum tree problem can be described as a problem of finding a set of edges that connects terminals. Therefore we can define a bivalent variable x_{ij} for each edge $(i, j) \in E$ indicating whether the edge (i, j) is included into the Steiner tree ($x_{ij} = 1$) or not ($x_{ij} = 0$) and similarly a bivalent variable f_i indicating whether vertex i is included in the Steiner tree ($f_i = 1$) or not ($f_i = 0$). For terminals, $i \in B$, it is satisfied $f_i = 1$, and $f_i \in \{0, 1\}$ for the other vertices, $i \in (V - B)$. In [55], we derived the the model based on a network flow formulation of the NSTP as follows. The variable y_{ij} represents a flow through the edge $(i, j) \in E'$.

$$\text{Minimise} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$
\begin{array}{rcl}
& r := \min\{k \mid k \in B\} & \\
\forall j \in (V - \{r\}) : \ x_{rj} & = & 0 \\
\forall i \in (V - \{r\}) : \ \sum_{j=1}^{n} x_{ij} & = & f_i \\
\forall i \in (V - \{r\}) : \ \sum_{j=1}^{n} y_{ij} - \sum_{j=1, j \neq r}^{n} y_{ji} & = & f_i \\
\forall i, j \in V : \ y_{ij} & \leq & (n-1) x_{ij} \\
\forall i, j \in V : \ y_{ij} & \in & \mathbf{Z}_+ \\
\forall i, j \in V, c_{ij} = 0 : \ x_{ij} & := & 0 \\
\forall i, j \in V : \ x_{ij} & \in & \{0, 1\} \\
\forall i \in B : \ f_i & = & 1 \\
\forall i \in (V - B) : \ f_i & \in & \{0, 1\}
\end{array}
$$

where $\mathbf{Z}_+$ denotes the set of nonnegative integers.

Another formulations can be found in [35] and [43]. A survey of Steiner tree problem formulations is presented in [21].

2.2 Preprocessing routines

Before executing the algorithm for solving the network Steiner tree problem, we will attempt to reduce the size of the given problem by using standard graph reduction techniques. These preprocessing routines always include the following simple ones:

1. An edge with its weight larger than a shortest path between its end vertices can be removed.
2. Any Steiner vertex of degree 1 can be removed along with the edges incident with it.

2.3 Approximation algorithms

As was mentioned above, the network Steiner tree problem is NP-complete. It means that solving the problem requires a computational time that grows (in the worst case) *exponentially* with the problem size. Therefore we must use approximation or heuristic approaches for large scaled instances. Many approaches are based on a reformulation of the problem, branch and bound method, dynamic programming, etc. [2], [13], [14], [39].

Aproximation algorithms are algorithms that guarantee that the "distance" of their solutions from the optimum is in the worst case restricted by a mathematically proved upper bound. The quality of an approximation algorithm is mostly measured by its *performance ratio*, which is given by the ratio of then achieved solution and the optimum. If a Steiner tree for a set of terminals B is computed by an approximation algorithm and has a total length $ASMT(B)$, Steiner minimum tree for B is $SMT(B)$, then the Steiner tree problem performance ratio $\varrho_{\text{graph}}(B)$ is given by

$$\varrho_{\text{graph}}(B) = \frac{w(ASMT(B))}{w(SMT(B))}$$

The simplest approximation algorithms for the NSTP are *distance network approximation* [36], *minimum path approximation* [52], and *contraction approximation* [45]. We will briefly summarise first two of them.

A distance network approximation [1] can be described as follows:

[*To determine a Steiner mimimum tree in a connected weighted graph* $G = (V, E)$ *with a set of terminals* B]

1. Construct the complete graph $K_B = (B, F)$ in which the weight of $\{i, j\} \in F$ is the shortest distance between i and j in G.
2. Obtain a minimum spanning tree $MST(K_B)$ of the graph K_B.
3. Replace each edge $\{i, j\}$ of $MST(K_B)$ by a shortest path between i and j in G. (The resulting graph, G', is a Steiner subgraph of G since it is connected and contains B.)
4. Obtain a minimum spanning tree $MST(G')$ in G'. ($MST(G')$ is a Steiner tree.)
5. If v is a Steiner vertex of degree 1 in $MST(G')$, delete v from the tree $MST(G')$ with its incident edge. Continue this process by deleting one Steiner vertex at a time.

Theorem 2.1 *The distance network approximation algorithm runs in* $O(|B||V|^2)$ *time.*

Proof. Running times of the steps are:

1. If the distance graph is not known, Step 1 requires time $O(|B||V|^2)$ to compute the shortest paths from each of the $|B|$ vertices.
2. Kruskal's minimum spanning tree algorithm requires time $O(|B| \log |B|)$.
3. Each of the $|B| - 1$ edges of $MST(K_B)$ may correspond to a path in G of up to $|V| - 1$ edges. Hence, Step 3 requires time $O(|B||V|)$.
4. $O(|B||V| \log(|B||V|))$ time using Kruskal's algorithm again.
5. The final step is done in time $O(|V|)$.

Step 1 is the most expensive and gives the distance network approximation algorithm a time complexity of $O(|B||V|^2)$. •

Theorem 2.2 *Performance ratio of the distance network approximation satisfies* $\varrho_{\text{graph}}(B) \leq 2$.

Proof. See [55]. •

The shortest paths approximation [52] is an analogy of Jarník's algorithm for finding a minimum spanning tree. This time we will describe it in a pseudopascal code.

As noted in [46] the solution can be improved by two additional steps:

1. Determine a minimum spanning tree for the subnetwork of G induced by the vertices in T.
2. Delete from this minimum spanning tree nonterminals of degree 1 (one at a time) and edges incident with them. The resulting tree is the (suboptimal) solution.

There exist several different (much more complicated) approximation algorithms for the network Steiner tree problem with provably good performance ratio. Table 1 gives a survey of known algorithms with their performance ratios.

input: connected undirected graph $G = (V, E)$,
weight function,
set of terminals $B \subset V$
output: Steiner tree T for B

select arbitrary terminal in B and denote it x_1
$T := \{x_1\}$;
$k := 1$;
while $k < |B|$ **do**
  begin determine a terminal $x_{k+1} \notin T$ closest to T
    $T := T \cup \{x_{k+1}\}$;
    $T := T\cup$ {a shortest path joining T and x_{k+1}};
    $k := k + 1$
  end;
{T is an approximation of the SMT}

Figure 2: Shortest paths approximation

reference	performance ratio
[52]	2.00
[45]	2.00
[70]	$\frac{11}{6} \approx 1.834$
[32]	1.757
[33]	1.644
[26]	1.598
[48]	$1 + \frac{\ln 3}{2} \approx 1.55$

Table 1: Steiner tree approximations

2.4 Heuristics

A *heuristic algorithm* is a technique that is designed to provide "good" (i.e. near optimal) solutions at a reasonable computational cost without being able to guarantee optimality (or feasibility in certain cases), or even in many cases, to state how close to optimality a particular feasible solution is. It is generally intended for solving approximately and efficiently, large and/or difficult optimisation problems for which optimal solutions cannot be found in a reasonable amount of computing time or within the available amount of computer memory using any of the existing exact methods [47].

Metaheuristics [40] constitute a class of paradigma useful for function optimisation, often inspired by the study of natural processes. They usually update possible solutions, one or a whole set at a time, to find the optimal solution of a given problem; in this sense the naming *evolutionary algorithm* is common in the literature. Particularly efficient instantiation of evolutionary algorithms are represented by *stochastic heuristic methods*. They can be divided by their strategy into two classes [59, 54]:

1. methods using *point-based strategy* (e.g. *simulated annealing*, *hill-climbing*, *tabu-search*), which define the *neighbourhood* around the current point (representing a solution) to be the current promising region to achieve a better solution in the search space, and
2. methods using *population-based strategy* (*genetic algorithms*) which define promising regions based on these points.

These methods differ from classical random descent methods in that they make it possible (with a certain probability) to accept a worse solution in the next iteration in order that to escape from local optima. For more detail see [20], [22], [37], [38], [42], [47].

A substantial question of finding the network Steiner minimum tree using stochastic heuristics is a representation of Steiner trees [15], [23], [18], [10], [19].

Solutions in the search space can be restricted to binary strings where a 1 or 0 corresponds to whether or not a vertex from $V - B$ is included in the Steiner tree.

Thus, the network Steiner tree problem can be *reformulated* [62] as follows: Given a graph $G = (V, E)$ and $B \subseteq V$, the NSTP is to find $S \subseteq V$ so that the spanning tree of the subgraph of G induced by $B \cup S$ has a minimum total weight.

(For a given graph $G = (V, E)$ and a subset $V' \subseteq V$, the *subgraph of G induced by V'* is the graph $G' = (V', E')$, such that (1) $E' \subseteq E$, (2) $(v_i, v_j) \in E' \Rightarrow v_i, v_j \in V'$, and (3) $[v_i, v_j \in V' \wedge (v_i, v_j) \in E] \Rightarrow (v_i, v_j) \in E'$.)

The problem is, that subgraphs induced by $B \cup S$ need not be connected or contain cycles, in other words, they do not represent spanning trees. Traditional approaches (e.g. [15]) apply approximation methods to a terminal set given by $B \cup S$. As the number of iterations of stochastic heuristic methods can reach tens of thousands, this approach requires too much time. For this reason, we propose another approach. When a subgraph induced by $B \cup S$ is disconnected, we penalise it so as to minimise its chance to be selected for the next iteration. In the second case, when a connected subgraph induced by $B \cup S$ c ontains cycles, we avoid them by a simple spanning tree algorithm.

2.5 Computational results

The use of the mathematical model in GAMS [55] showed that this software package is able to find an exact solution for small instances such in Fig. 1 but, e.g. for an instance with 50 vertices, it is not able to achieve a solution in 30 minutes. We must realize that the model, described by several equations, contains, in fact, up to tens of thousands of equations because of quantifiers, e.g. equations that begin with $\forall i, j \in V$ correspond to $|V|^2$ equations.

The genetic algorithm parameters were set as follows [66]:

- population size: 100,
- crossover: uniform,
- mutation: inverting of one randomly chosen position in the binary string of Steiner vertices candidates,
- selection: binary tournament,
- number of iterations: 10000-30000
- replacement: steady-state (eliminating the worst individual).

The *fitness function* corresponds to the objective function. In traditional GAs objective function is maximised, however in our problem we wish to minimise the total length of the MST:

Simulated annealing settings [66]:

- neighbourhood operation: the same as the GA mutation
- initial temperature: 10000,
- final temperature: 1,
- temperature decrement 0.80-0.99

The ability of computing (near-)optimal solutions was checked using standard benchmarks from OR-Library (OR = Operations Research) accessible from London Imperial College Management School [4] and from www page of Berlin Konrad-Zuse-Zentrum für Informationstechnik [51]. There were two classes of benchmarks: the first with 50-100 vertices and 63-200 edges. Graphs in the second class have 500 vertices and 625-2500 edges. GAs give the optimum in 74% of cases for instances with no more 200 vertices and the SA in 80% cases [66]. The running times were similar, 7 or 5 minutes, respectively. For larger instances with 500 vertices, the results are not satisfactory, they differ up to tens of precent form the optimum.

3 Rectilinear Steiner tree problem

Let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of points in the plane called *terminals*. A *Steiner tree* is a tree in the plane that contains V. The *Steiner tree problem* is to find a Steiner tree of minimum length (StMT). There are several versions of the problem. They differ by the underlying distance metric. In the *rectilinear Steiner tree problem* (RSTP) the distance $d(v_i, v_j)$ between two points v_i and v_j is defined by the *rectilinear* (or *Manhattan*) *metric*

$$d(v_i, v_j) = |x_i - x_j| + |y_i - y_j|,$$

where (x_i, y_i) are the Cartesian coordinates of v_i. Thus the *rectilinear Steiner minimum tree* (RStMT) is the shortest network of horizontal and vertical lines connecting all the terminals of V.

The RStMT problem has numerous applications in the area of VLSI design automation as well as printed circuit board layout.

The decision form of this problem has been shown to be NP-complete [17]. For this reason, it is important to find an approximation for solving this problem in a reasonable amount of

test	$\|V\|$	$\|H\|$	$\|B\|$	optimum	min	average
b01	50	63	9	82	82	82.0
b02	50	63	13	83	83	83.9
b03	50	63	25	138	138	138.0
b04	50	100	9	59	59	60.2
b05	50	100	13	61	61	61.0
b06	50	100	25	122	122	122.0
b07	75	94	13	111	111	111.0
b08	75	94	19	104	104	104.3
b09	75	94	38	220	220	220.0
b10	75	150	13	86	86	87.7
b11	75	150	19	88	88	88.9
b12	75	150	38	174	174	174.0
b13	100	125	17	165	165	168.2
b14	100	125	25	235	235	249.5
b15	100	125	50	318	318	319.1
b16	100	200	17	127	127	129.3
b17	100	200	25	131	131	131.0
b18	100	200	50	218	218	218.0

Table 2: GA, number of iterations 10000, running time ¡ 1 min

test	$\|V\|$	$\|H\|$	$\|B\|$	optimum	min	average
b13	100	125	17	165	165	165.5
b14	100	125	25	235	235	238.6
b15	100	125	50	318	318	319.1
b16	100	200	17	127	127	128.6

Table 3: GA, number of iterations 30000, running time ¡ 7 min

test	$\|V\|$	$\|H\|$	$\|B\|$	optimum	min	prmr
c01	500	625	5	85	88	106.5
c02	500	625	10	144	148	156.0
c06	500	1000	5	55	55	66.3
c07	500	1000	10	102	118	128.6
c09	500	1000	125	707	728	733.0
c10	500	1000	250	1093	1095	1096.7

Table 4: GA, number of iterations 30000, running time ¡ 1 hour

test	$\|V\|$	$\|H\|$	$\|B\|$	optimum	min	average
b01	50	63	9	82	82	82.5
b02	50	63	13	83	83	85.8
b03	50	63	25	138	138	138.2
b04	50	100	9	59	59	64.4
b05	50	100	13	61	61	66.7
b06	50	100	25	122	122	124.8
b07	75	94	13	111	111	115.7
b08	75	94	19	104	104	109.2
b09	75	94	38	220	220	221.0
b10	75	150	13	86	86	91.8
b11	75	150	19	88	90	94.6
b12	75	150	38	174	174	177.3
b13	100	125	17	165	170	179.0
b14	100	125	25	235	235	258.7
b15	100	125	50	318	322	334.2
b16	100	200	17	127	129	141.3
b17	100	200	25	131	132	139.4
b18	100	200	50	218	218	224.6

Table 5: SA, decrement 0.80, running time 18 sec

test	$\|V\|$	$\|H\|$	$\|B\|$	optimum	min	average
b01	50	63	9	82	82	82.2
b02	50	63	13	83	83	83.0
b03	50	63	25	138	138	138.0
b04	50	100	9	59	59	59.0
b05	50	100	13	61	61	61.0
b06	50	100	25	122	122	122.0
b07	75	94	13	111	111	111.0
b08	75	94	19	104	104	104.4
b09	75	94	38	220	220	220.0
b10	75	150	13	86	86	86.6
b11	75	150	19	88	88	88.4
b12	75	150	38	174	174	174.2
b13	100	125	17	165	165	165.0
b14	100	125	25	235	235	235.0
b15	100	125	50	318	318	319.6
b16	100	200	17	127	127	127.5
b17	100	200	25	131	132	139.4
b18	100	200	50	218	218	224.6

Table 6: SA, decrement 0.99, running time 5 min

time. As a rectilinear minimum spanning tree (RMSpT) is at most 1.5 times longer than the length of an RStMT [5, 6, 25, 28], several RMSpT-based heuristics were proposed in which a rectilinear Steiner tree is obtained by refining the RMSpT. We will show that the proof of this fundamental proposition published in [25] is mistaken.

3.1 Rectilinear Steiner tree properties

In this section basic definitions and theorems, necessary for proving the Steiner ratio, are summarised [12, 24, 25, 27, 28, 69].

Theorem 3.1 (Hanan [24]) *Let the coordinates of the terminals be* $(x_1, y_1), \ldots, (x_n, y_n)$*. There exists a rectilinear Steiner minimum tree* T *where if* (x', y') *is a Steiner point of* T *then* $x' = x_i$ *and* $y' = y_j$*,* $1 \leq i, j \leq n$*.* •

Theorem 3.2 *The number of RStTs on* n *terminals equals* $2^{n-1}n^{n-2}$*.*

Proof. To find a rectilinear Steiner minimum tree, by Hanan's theorem, it is sufficient to consider only the unoccupied corners of the rectangles each pair of terminals determines. Thus each spanning tree edge determines two alternative Steiner points. By Euler's formula trees on n vertices contain $n-1$ edges, so the $n-1$ edges of a spanning tree determine $2.2. \ldots .2 = 2^{n-1}$ different combinations of Steiner points, thus 2^{n-1} rectilinear Steiner trees can be constructed from one spanning tree. By Cayley's formula [41], the number of distinct spanning trees on a complete graph of n points equals n^{n-2}. Hence we get $2^{n-1}n^{n-2}$ distinct rectilinear Steiner trees. •

Definition 3.1 Let V be a set of terminals. A rectilinear Steiner tree T for V is called a *full rectilinear Steiner tree* if all terminals are leaves in T (i.e. have degree 1). □

The following is a well-known *folk theorem* [67] (also called *decomposition theorem*) .

Theorem 3.3 *Let* V *be a set of terminals, with* $|V| \geq 2$*. If* T *is a rectilinear Steiner minimum tree for* V*, then at terminals with degree more than one,* T *can be split into edge-disjoint full rectilinear Steiner trees that have only the split terminals in common. (These trees are called full components).*

Proof. It is straightforward. In other words, every rectilinear Steiner minimum tree is composed of a number of full rectilinear Steiner trees that intersect at terminals of degree 2 or greater. Let $deg(v_i)$ be the degree of terminal v_i. The number of full components is $1 + \sum_{i=1}^{N}(deg(v_i) - 1)$ where N is the number of terminals with degree 2 or greater. •

An example of an RStMT composed from 9 full components is shown in Fig. 3.

The following Hwang's theorem characterises the topological shape of full components.

Theorem 3.4 (Hwang [27]) *Let* $n \geq 5$*. Suppose that the RStMT of a given set of* n *terminals is full. Then there exists an RStMT that has one of the two canonical topologies: either it consists of a single line with* $n-1$ *alternating incident segments, or of a corner with* $n-3$ *alternating segments incident to one leg and a single segment incident to the other leg.* •

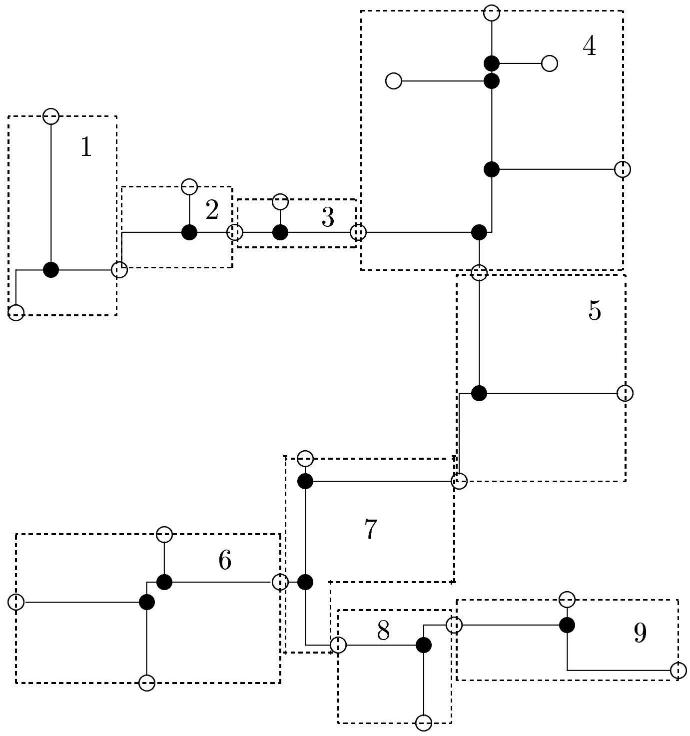

Figure 3: Decomposition of RStMT into 9 full components

3.2 Minimum spanning tree approximation

In this section we will pay attention to the quality of a rectilinear minimum spanning tree (RMSpT) as an approximation of a rectilinear Steiner minimum tree (RStMT).

Definition 3.2 An algorithm A is said to be a $\delta-$*approximation algorithm* for a minimisation problem P if, for every instance I of P, it delivers a solution that is at most δ times the optimum. □

Definition 3.3 The *rectilinear Steiner ratio*, denoted ϱ_{rect}, is the supremum over all terminal sets of the length of the rectilinear minimum spanning tree $w(RMSpT(V))$ divided by the length of the rectilinear Steiner minimum tree $w(RStMT(V))$,

$$\varrho_{\text{rect}} = \sup_{V \subset \Re^2} \frac{w(RMSpT(V))}{w(RStMT(V))} \tag{1}$$

□

Hwang showed that the rectilinear Steiner ratio is $\frac{3}{2}$. Since the original proof of this proposition is rather complicated, other proves have been proposed. Here, we use the strategy presented in [25], but we will modify it to avoid the mistake included in it.

We may assume that the rectilinear Steiner minimum tree is a rectilinear full Steiner tree (otherwise, by Theorem 3.3, we might split T into smaller full Steiner trees and apply the theorem for the Steiner ratio inductively).

Theorem 3.5 *Let $RStMT(V)$ be a rectilinear Steiner minimum tree for V, $RMSpT(V)$ be a rectilinear minimum spanning tree for V and $w(T)$ be the total length of T. Then, for every set of terminals V, $\varrho(V) = \frac{w(RMSpT(V))}{w(RStMT(V))}$ satisfies $\varrho(V) \leq \frac{3}{2}$.*

Proof. For $n = 2$ or all n terminals collinear, the rectilinear Steiner minimum tree is the rectilinear minimum spanning tree. In the sequel, we will assume that all n terminals are not collinear.

Let $n = 3$. It is known [24] that, in this case, the Steiner point is located at the medians of the x and y coordinates of the given terminals and the length of a rectilinear Steiner minimum tree is given by $\frac{1}{2}w(R)$, where $w(R)$ is the length of the rectangle enclosing the terminals. If we leave in this rectangle the connection of a pair of adjacent terminals that is the longest, i.e. a corner with a length $\geq \frac{1}{3}w(R)$, we get a rectilinear spanning tree with a length $w(RMSpT(V)) \leq \frac{2}{3}w(R)$. Hence we get

$$\varrho(V) = \frac{w(RMSpT(V))}{w(RStMT(V))} \leq \frac{4}{3} < \frac{3}{2}$$

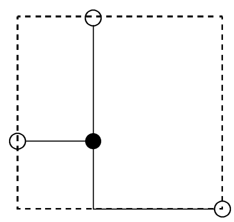

Figure 4: Rectilinear Steiner minimum tree for 3 terminals

If $n = 4$, then T is a tree in Fig. 5.

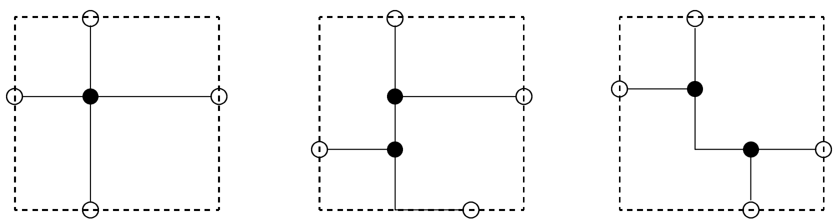

Figure 5: Rectilinear Steiner minimum trees for 4 terminals

The length of the rectilinear Steiner minimum tree from the left-hand side of Fig. 5 is greater than half of the enclosing rectangle R for a set V, the tree on the right-hand side has a length equal to $\frac{1}{2}w(R)$. Therefore, both cases satisfy $w(RStMT(V)) \geq \frac{1}{2}w(R)$. By deleting the longest link of the enclosing rectangle R in Fig. 5, i.e. the link with a length $\geq \frac{1}{4}w(R)$, we obtain a rectilinear spanning tree whose length satisfies $w(RMSpT(V)) \leq \frac{3}{4}w(R)$ and thus

$$\varrho(V) = \frac{w(RMSpT(V))}{w(RStMT(V))} \leq \frac{3}{2}$$

Now let $n \geq 5$. In Fig. 6 the part of the full component of a canonical form bounded by the two opposite local minima h_1 and h_2 of the horizontal lines whose distance is at least three vertical segments is called a Steiner segment. By a local minimum we mean a horizontal line shorter than its two neighbours on the same side of the backbone (only one neighbour is at the end of the full component). This part of the full component we will call In the second and third part of Fig. 6, two rectilinear spanning trees $RMSpT_1$ and $RMSpT_2$ are shown. They connect the terminals of the Steiner segment. The $RMSpT_2$ looks like the $RMSpT_1$ excluding the area between the maximal horizontal lines H_1, H_2 from both sides of the backbone. The terminals of the $RMSpT_2$ are linked by a chain alternating both sides of the backbone. Now both spanning trees contain a parallel segment that extends the Steiner segment. However, these extensions are disjoint and correspond to a subset of the Steiner segment edges. Evidently, the sum of both spanning tree lenghts includes no extensions from the outside horizontal lines h_1 and h_2, which compensates the extension caused by the fact that neighbouring Steiner segments share the outside horizontal lines. Therefore the sum of the lenghts of the spanning trees $RMSpT_1$ and $RMSpT_2$ does not exceed the length of the Steiner segment times three. It implies that the shorter of these two spanning trees has a length of at most $\frac{3}{2}$ of the Steiner segment length. Since it holds for every Steiner segment, and thus for the whole full component, too, the theorem is proved. •

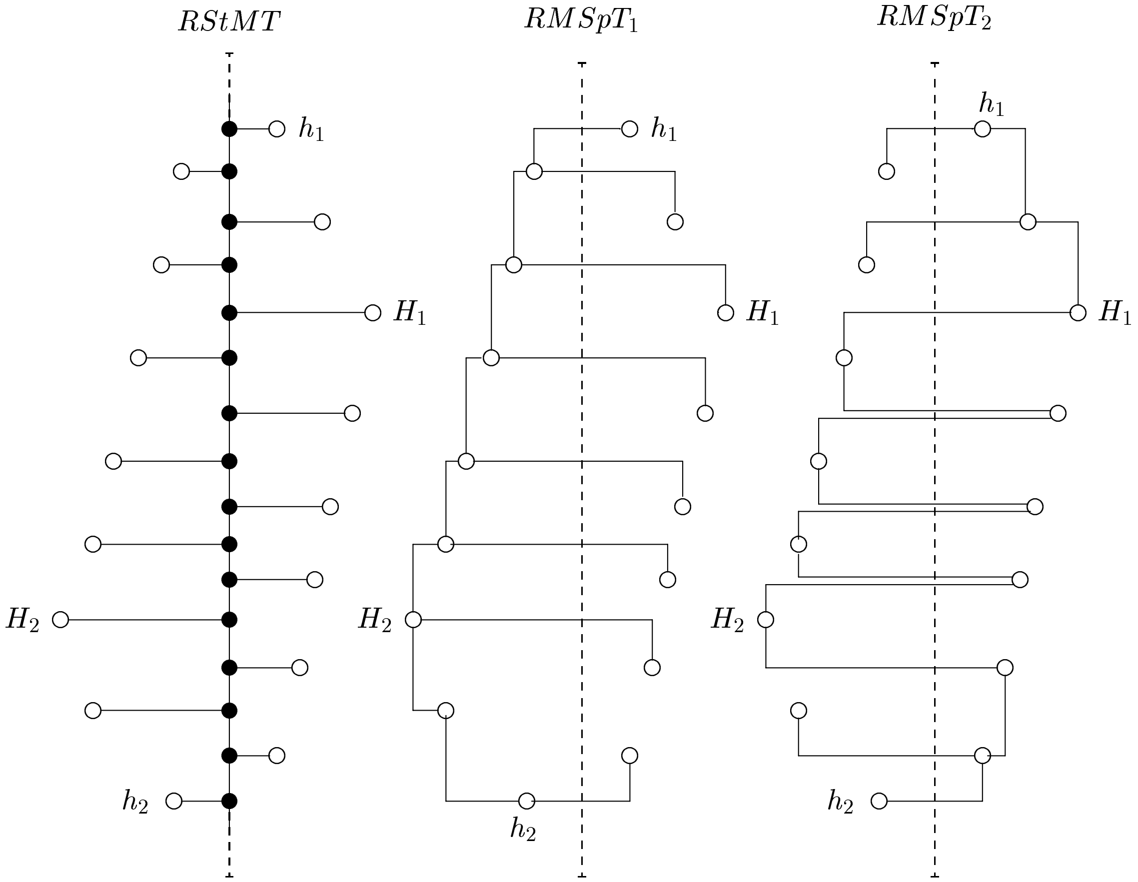

Figure 6: Steiner segment

Note 3.1 The proof of Theorem 3.5 was inspired by a graphical proof from [25]. However, it contains a mistake [56]. It is applied to the whole full component of a canonical form. In the first spanning tree, all terminals from the left-hand side are connected and terminals of the adjacent horizontal lines from the right-hand side of the backbone are joined with them. In the second spanning tree, the mirrored strategy is used. This situation is demonstrated in Fig. 7. Evidently, parallel coverings of the backbone are disjoint between both spanning trees and the total length of both spanning tree includes three times the backbone length, which is in accoradance with the assertion of Theorem 3.5. The total length of the horizontal lines of both spanning trees can be longer than three times the total length of the full component. Let us consider a case where the horizontal line lengths are $h_1 = h_5 = h_9 = h_4 = h_8 = 10$ and $h_3 = h_7 = h_2 = h_6 = h_{10} = 1$ and the lengths of the backbone vertical segments equal 1. Then the total length of the horizontal lines in the tree T equals $5 \cdot 10 + 5 \cdot 1 = 55$, the total length of the horizontal lines of the spanning tree $RMSpT_1$ is $(8 \cdot 10 - 2 \cdot 1) + (4 \cdot 1 + 3 \cdot 10) = 112$, the total length of the horizontal lines of $RMSpT_2$ is $(3 \cdot 10 + 2 \cdot 1) + (1 \cdot 1 + 6 \cdot 10 - 2 \cdot 1) = 91$. The total length of the horizontal lines in $RMSpT_1$ and $RMSpT_2$ equals $112 + 91 = 203$, which is more than $3 \cdot 55 = 165$. It is due to the fact that both spanning trees have the maximal horizontal lines on the same side and they are doubled when connecting a local minimum with adjacent maxima (see the connection of the lower three terminals on the left-hand side of $RMSpT_1$. Let us evaluate the lengths of T, $RMSpT_1$ and $RMSpT_2$, let us denote them by $w(T)$, $w(RMSpT_1)$ and $w(RMSpT_2)$. For the given data and the total lengths of the horizontal lines we get: $w(T) = 55 + 10 = 65$, $w(MSpT_1) = 112 + 10 + 4 = 126$ and $w(MSpT_2) = 91 + 10 + 5 = 106$. That means that the $RMSpT_2$ is shorter. The ratio $\frac{w(MSpT_2)}{w(T)} = \frac{106}{65} = 1.63 > \frac{3}{2}$, which contradicts the proof of Theorem 3.5 in [25]. $\diamondsuit$

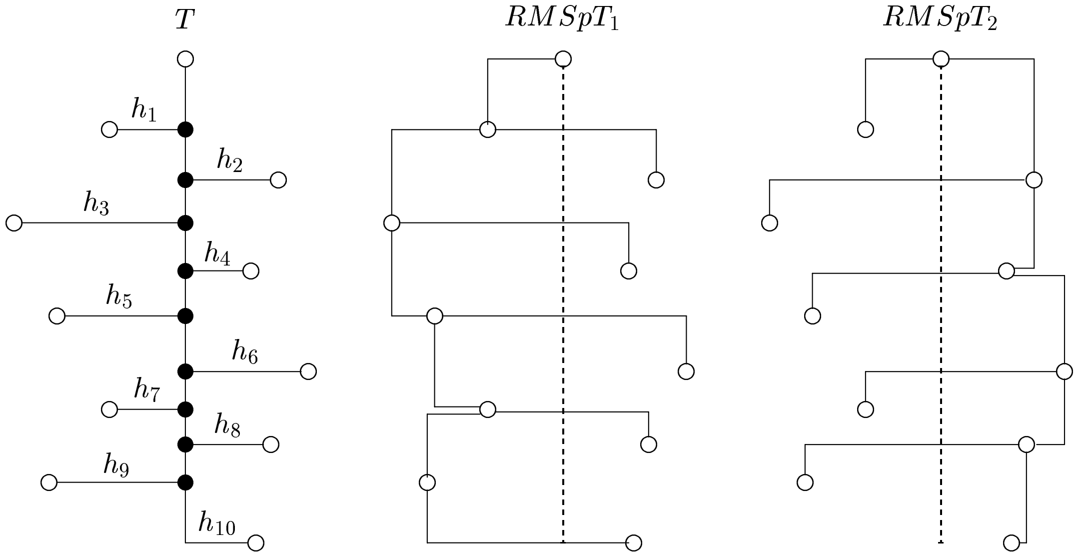

Figure 7: Illustration of the incorrect proof from [25]

The theorem 3.5 guarantees that the performance ratio when approximating the rectilinear Steiner minimum tree by a rectilinear minimum spanning tree does not exceed 1.5. The 1.5 bound is attained in the graph shown in Fig. 8 where all edges are of equal lengths or in graphs that consist of copies of Fig. 8. It implies that the rectilinear Steiner ratio equals 1.5.

3.3 Computational results

For computational tests we used an approach similar to that presented in the previous section. However, due to the high number of Steiner point candidates in locations of Hanan's grid, computations found the optimum in a reasonable amount of time (up to 30 min) only for benchmarks with no more than 20 points [64]. This agrees with literature, e.g., in [49], an algorithm is mentioned which requires one day running time for the 35-point benchmark. We can say that metaheuristics are not suitable [30, 31] for the rectilinear problem and it is better to use traditional iteration methods or to try to improve them.

4 Euclidean Steiner Tree Problem

The *Euclidean Steiner tree problem* (ESTP) is given by a set of points $V = \{v_1, v_2, \ldots, v_n\}$ in the Euclidean plane, called *terminals*, and asks for the shortest planar straight-line spanning the

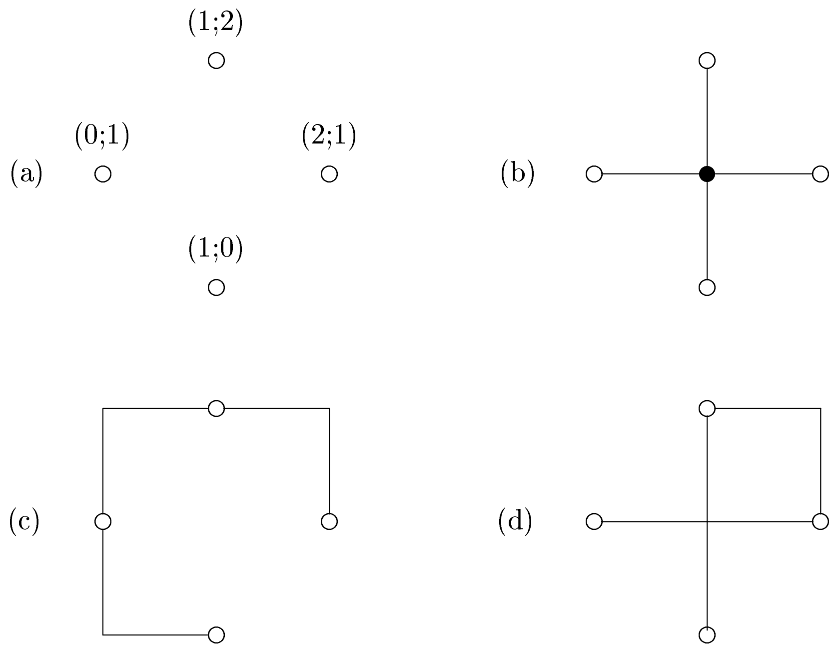

Figure 8: An example with the performance ratio $\varrho(V) = \frac{3}{2}$; (a) given set of terminals, (b) rectilinear Steiner minimum tree with length 4, (c),(d) two different rectilinear minimum spanning trees with length 6.

set V. The solution takes the form of a tree, called a *Euclidean Steiner minimum tree* (ESMT). Contrary to the *minimum spanning tree problem*, connections in ESMTs are not required to be between the terminals only. Additional intersections, called *Steiner points*, can be introduced to obtain shorter spanning networks.

The distance $d(u,v)$ between two points $u = (u_x, u_y)$ and $v = (v_x, v_y)$ in the Euclidean plane is defined by

$$d(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}.$$

Euclidean Steiner tree problem applications span a wide range, from modelling the evolution of species in biology to modelling soap films for grids of wires; from the design of collections of data to the design of heating or air-conditioning systems in buildings; and from the creation of oil and gas pipelines to the creation of communication networks, electricity distribution networks, road and railway lines [8].

The Euclidean mimimum spanning tree (EMST) can be used as an approximation Steiner tree. The supremum (over all terminal sets V) of the ratio between the minimum spanning tree length and the Steiner minimum tree length is called the *Steiner ratio*. It can be proved [28] that the Steiner ratio for the Euclidean problem is $2/\sqrt{3} \approx 1.1547$. That means that the EMST length does not exceed that of an ESMT by more than 15.47% (the average excessive length is, of course, smaller). Therefore, the EMST naturally becomes the standard, against which other approximation algorithms or heuristics are compared.

The problem of EMST construction can be reduced easily to that of searching for a minimum weighted spanning subgraph of a weighted graph. We construct the complete graph K_n by joining each pair of terminals v_i, v_j from V by a line segment and define the weight of the edge $\{v_i, v_j\}$ of the graph K_n by putting $w_{ij} = |v_i v_j|$. This approach requires $O(n^2)$ time. However, the Euclidean minimum spanning tree (EMST) can be constructed in $O(n \log n)$ time, if we use the fact that, in searching for a current edge of minimal weight in classical EMST algorithms, it suffices to go through the edges of the Delaunay triangulation [63, 57, 50].

4.1 Steiner mimimum tree properties

It is known [28, 29], that the Steiner minimum trees have the following properties:

- All Steiner points are incident with exactly 3 edges forming an angle of 120° between each other (i.e. they have degree 3 with respect to the edges used in the tree). We refer to this property as the *angle and degree conditions*.
- ESMTs for n terminals have at most $n-2$ Steiner points.
- ESMTs are unions of *full Steiner trees* (FSTs). FSTs have two Steiner points less than they have terminals and terminals spanned by an FST have degree 1. If two FSTs share a terminal z, then the two edges incident with z (one from each FST) forming and angle of at least 120° between each other. No terminal can therefore be in more than three FSTs.

If $|V| = 3$ we can directly construct the ESMT as follows: Let $V = \{a, b, c\}$.

1. If one of the angles of $\triangle abc$ is at least 120°, then the ESMT consists of simply the two edges subtending the obtuse angle.
2. If all internal angles of $\triangle abc$ are less than 120°, then we draw an equilateral triangle $\triangle abd$ and circumscribe a circle around this triangle. The Steiner point s is given by the intersection of the line cd with the circle (Fig. 9). It can be shown that the total length of segments as, bs, cs is equal to the length of segment cd, which is known as the *Simpson line* for the FST over terminals a, b, c.

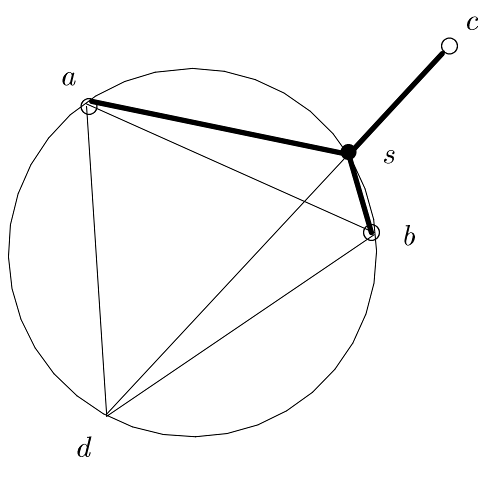

Figure 9: A 3-point ESMT algorithm

4.2 Steiner insertion heuristic

In this section, we describe a simple heuristic derived from [11]. It belongs to heuristics based on improving the EMST. It systematically inserts Steiner points between edges of the minimal spanning tree meeting at angles less than 120 degrees, performing a local optimization at the end. As the last step is not specified in [11], we replace it by another step inspired by one preprocessing rule known from the Steiner tree problem in graphs.

The algorithm can be described as follows [65]:

1. Find the Euclidean minimum spanning tree.
2. **For each** edge connecting fixed points $\{x, y\}$ **do**
   (a) Find the edge $\{y, z\}$ that meets $\{x, y\}$ at the smallest angle, where z can be either a fixed point or a Steiner point.
   (b) **If** this angle is less than 120° **then**
      i. Place a new Steiner point s_n on top of y.
      ii. Remove the edges $\{x, y\}$ and $\{y, z\}$. These edges will no longer be considered for the loop of Step 2.
      iii. Add the edges $\{x, s_n\}$, $\{y, s_n\}$ and $\{z, s_n\}$.
3. Remove all Steiner points of degree 1 along with their incident edges.

As the time complexity of Step 2 is $O(n^3)$ and is higher than the complexity of the Euclidean minimum spanning tree enumeration in Step 1, and the complexity of optimization Step 3, it determines the complexity of the algorithm.

The selection by the smallest angle is based on the fact that the smaller the angle is, the better local improvement is reached with respect to the minimum spanning tree. (Of course this angle must be less than 120°.) The best improvement is reached by inserting a Steiner point into the equilateral triangle with angles 60°.

The angle between a pair of edges meeting at end points can be easily determined using elementary geometry. If α is an angle to be calculated and Euclidean distances of the end points define the lengths of the triangle edges and they are denoted by a, b, c as shown in Figure 10, then we have:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha,$$

and hence we obtain:

$$\alpha = \arccos \frac{b^2 + c^2 - a^2}{2bc}$$

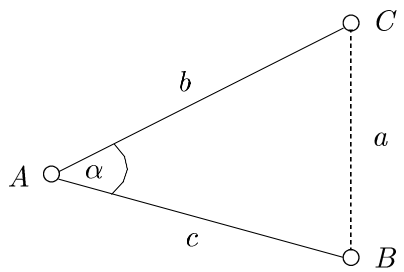

Figure 10: To the angle calculations

4.3 Computational results

Table 7 summarises the computational results of the Steiner insertion heuristic for a subset of Euclidean Steiner tree problem benchmarks published in OR-Library [3]. In several cases the optimum from OR-Library, $w(EStMT)_{\mathrm{OR}}$, is worse than a solution achieved by the Steiner insertion heuristic, $w(EStMT)$. It is caused by rounding errors when Euclidean distances are calculated. In all cases improvements against the Euclidean minimum spanning trees reached a minimum of 3%. Run times did not exceed tens of seconds. For an example with 500 points (not included in the table) the improvement was equal to 2% and the calculation required 30 minutes. The algorithm was implemented in C++ Builder and run on the IBM compatible computer with INTEL III processor, 633 MHz and an operational memory of 256 MB.

$\lvert V \rvert$	$w(EMSpT)_{\mathrm{OR}}$	$w(EStMT)_{\mathrm{OR}}$	$w(EStMT)$	number of Steiner points
10	1372	1313	1338	4
20	2088	1997	2009	8
30	2378	2326	2339	11
40	2622	2553	2553	13
50	3229	3144	3148	19
60	3223	3128	3131	20
70	3632	3530	3528	27
80	4188	4067	4067	29
90	4089	3937	3933	38
100	4296	4156	4151	37

Table 7: Computational results

5 Conclusions and future work

The aim of this paper was to describe Steiner tree problems, which are an important subset of mimimal network tasks. The only criterion that we used was total length of the network or the total costs for its creating. In spite of their broad application area, Steiner problems are not studied in the Czech literature, which is evidenced by the references. Graph theory books only note these problems as a special case of the mimimum spanning tree problem or even do not mention them at all.

We have tried to deal with he problems in this area in a clear way. It refers to tens of other papers often written in a very sketched form, e.g. proofs are very brief, incomplete or not quite general (e.g. the proof of NP-completeness of the RStMTP. Many theorems were formulated and proved by the author without an analogy to literature. The aim was to explain all Steiner tree problems in the same way as the foreign authors often specialize, e.g. Hwang, Warme and Zachariasen in the geometrical problems (rectilinear and Euclidean), Ganley and Robins in the rectilinear one, Voss and Hougardy to the Steiner problem in graphs, Zelikovsky in the rectilinear and graphical problem, etc. Further, in a simple way, it was proved that Eppstein's proof of the Steiner ratio for rectilinear problem is incorrect and its correct modification was proposed [56] based on the original strategy of two spanning trees, however, their different topology does not exceed the bound of 1.50 for the Steiner ratio.

Time complexities of many algorithms were recalculated with respect to the use of effecient data structures, e.g. the implementation of priority queues by a binary heap decreased the time complexity of Jarník's and Dijkstra's algorithm and this was projected into two approximation algorithms for Steiner tree problem in graphs.

Due to the NP-completeness of the Steiner tree problems, it was not possible to determine an exact solution for large instances. Using a new mathematical model it was shown that even the professional optimization package GAMS (probably based on Branch and Bound Method) can solve only small instances with no more than 50 vertices.

On the other hand, a general applicability of metaheuristics for all Steiner tree problems was not confirmed. They surely cannot be recommended for the rectilinear problem, but are promising for the Steiner problem in graphs where the Steiner ratio is high. As for the Euclidean Steiner tree problem its Steiner ratio $\frac{2}{\sqrt{3}}$ shows that the approximation by the Euclidean minimum spanning tree is a very good initial solution and its improvement by simple deterministic heuristic presented here is quite sufficient.

In future, we will try to generalize these problems to the fuzzy case where weights of edges will be given by fuzzy numbers. As yet an algorithm has been implemented determining the shortest path with distances described by fuzzy numbers [60], which will be used for a fuzzy variant of the graphical Steiner tree problem.

Acknowledgments

This work was supported by the research project CEZ: J22/98: 261100009 *Non-traditional methods for investigating complex and vague systems.*

References

[1] V.K. Balakrishnan. *Network Optimization.* Chapman & Hall Mathematics, London, 1995.

[2] C.D. Bateman, C.S. Helvig, G. Robins, and A. Zelikovsky. Provably Good Routing Tree Construction with Multi-Port Terminals. In *Proceedings of the ACM/SIGDA International Symposium on Physical Design*, pages 96–102, Napa Valley (Canada), 1997.

[3] J.E. Beasley. OR-Library. Report, The Management School Imperial College, London, 1996. `http://mscmga.ms.ic.ac.uk/jeb/orlib/esteininfo.html`.

[4] J.E. Beasley. OR-Library. Report, The Management School Imperial College, London, 1996. `http://mscmga.ms.ic.ac.uk/info.html`.

[5] P. Berman, U. Fössmeier, M. Karpinski, M. Kaufmann, and A. Zelikovsky. Approaching the 5/4-Approximation for Rectilinear Steiner Trees. Technical Report, International Computer Science Institute, University of California at Berkeley, September 1995. 26 pp. (*Proceedings of the Second Annual European Symposium on Algorithms ESA '94*. Utrecht (Netherlands), 1997, pp. 60-71. Springer-Verlag Berlin, LNCS 855).

[6] P. Berman and V. Ramaiyer. Improved Approximations for the Steiner Tree Problem. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms SODA '92*, pages 325–334, Orlando, 1992.

[7] D. Cavendish, A. Fei, M. Gerla, and R. Rom. On the Construction of Low Cost Multicast Trees with Bandwidth Reservation. In *Proceedings of the International Conference on High-Performance Computing and Networking HPCN*, Amsterdam, 1998. 29 pp.

[8] D. Cieslik. *Steiner Minimal Trees.* Kluwer Academic Publishers, Dordrecht, 1998.

[9] D.W. Corne, M.J. Oates, and G.D. Smith (eds.). *Telecommunications Optimization: Heuristic and Adaptive Techniques.* John Wiley & Sons, New York, 2000. ISBN 0-471-98855-3.

[10] K.A. Dowsland. Hill-Climbing, Simulated Annealing and the Steiner Problem in Graphs. *Engineering Optimisation*, 17:91–107, 1991.

[11] D.R. Dreyer and M.L. Overton. Two Heuristics for Euclidean Steiner Tree Problem. *Journal on Global Optimization*, 13:95–106, 1998.

[12] D.-Z. Du, J.M. Smith, and J.H. Rubinstein. *Advances in Steiner Trees.* Kluwer Academic Publishers, Dordrecht, 2000. ISBN 0-7923-6110-5.

[13] C.W. Duin and A. Volgenant. The Partial Sum Criterion for Steiner Trees in Graphs and Shortest Paths. *European Journal of Operational Research*, 97:172–182, 1997.

[14] C.W. Duin and S. Voss. Efficient Path and Vertex Exchange in Steiner Tree Algorithms. *Networks*, 29:89–105, 1997.

[15] H. Esbensen. Computer Near-Optimal Solutions to the Steiner Problem in a Graph Using a Genetic Algorithm. *Networks*, 26:173–185, 1995.

[16] A. Fink, G. Schneidereit, and S. Voss. *Solving General Ring Network Design Problems by Meta-Heuristics*, chapter in: Laguna, M. and Gonzlez Velarde, J.L. (eds.): *Computing Tools for Modeling, Optimization and Simulation* (*Interfaces in Computer Science and Operations Research*), pages 91–113. Kluwer Academic Publishers, Boston, 1999.

[17] M.R. Garey and D.S. Johnson. The Rectilinear Steiner Tree Problem is NP-Complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977.

[18] M. Gendreau, J.F. Larochelle, and B. Sanso. A Tabu Search Heuristic for the Steiner Tree Problem. *Networks*, 34:162–172, 1999.

[19] A. Ghanwani. Neural and Delay Based Heuristics for the Steiner Problem in Networks. *European Journal of Operational Research*, 108:241–265, 1998.

[20] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Boston, 1997.

[21] M.X. Goemans and Y. Myung. A Catalog of Steiner Tree Formulations. *Networks*, 23:19–28, 1993.

[22] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, New York, 1989.

[23] P. Guitart and J.M. Basart. A Genetic Algorithm Approach for the Steiner Problem in Graphs. In *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing EUFIT '98*, volume 1, pages 508–512, Aachen, 1998.

[24] M. Hanan. On Steiner's Problem with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 14(2):255–265, 1966.

[25] D. Hochbaum (ed.). *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, 1996. ISBN 0-534-94968-1.

[26] S. Hougardy and H.J. Prömel. Approximation Algorithm for the Steiner Problem in Graphs. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms SODA '99*, pages 448–453, Baltimore (Maryland), 1999.

[27] F.K. Hwang. On Steiner Minimal Trees with Rectilinear Distance. *SIAM Journal on Applied Mathematics*, 30(1):104–114, 1976.

[28] F.K. Hwang, D.S. Richards, and P. Winter. *The Steiner Tree Problem.* North-Holland, Amsterdam, 1992. ISBN 0-444-89098-X.

[29] A.O. Ivanov and A.A. Tuzhilin. *Minimal Networks. The Steiner Tree Problem and its Generalizations.* CRC Press, Boca Raton, 1994. ISBN 0-8493-8642-X.

[30] B.A. Julström. A Genetic Algorithm for the Rectilinear Steiner Problem. In *Proceedings of the Fifth International Conference on Genetic Algorithms ICGA '93*, pages 474–480, Urbana Champaign (Illinois), 1993. Morgan Kaufmann.

[31] B.A. Julström. Representing Rectilinear Steiner Trees in Genetic Algorithms. In *Proceedings of the ACM Symposium on Applied Computing SAC '96*, pages 245–250, Philadelphia (Pennsylvania), 1996. ACM Press.

[32] M. Karpinski and A. Zelikovsky. 1.757 and 1.267-Approximation Algorithms for the Network and Rectilinear Steiner Tree Problem. Technical Report TR-95-010, International Computer Science Institute, University of California at Berkeley, March 1995. 13 pp.

[33] M. Karpinski and A. Zelikovsky. New Approximation Algorithms for the Steiner Tree Problems. *Journal of Combinatorial Optimization*, (1):47–65, 1997.

[34] S. Khuller. Design and Analysis of Algorithms. Lecture Notes, University of Maryland, Department of Computer Science, 1994. 112 pp.

[35] T. Koch and A. Martin. Solving Steiner Tree Problems in Graphs to Optimality. *Networks*, 32:207–232, 1998.

[36] L. Kou, G. Markowsky, and L. Berman. A Fast Algorithms for Steiner Trees. *Acta Informatica*, 15:141–145, 1981.

[37] V. Kvasnička, L. Beňušková, J. Pospíchal, I. Farkaš, P. Tiňo, and A. Král. *Úvod do teórie neurónových sietí.* IRIS, Bratislava, 1997.

[38] V. Kvasnička, Pospíchal J., and P. Tiňo. *Evolučné algoritmy.* STU, Bratislava, 2000.

[39] A. Lucena and J.E. Beasley. A Branch and Cut Algorithm for the Steiner Problem in Graphs. *Networks*, 31:39–59, 1998.

[40] V. Maniezzo, A. Colorni, and M. Dorigo. Algodesk: An Experimental Comparison of Eight Evolutionary Heuristics Applied to the Quadratic Assignment Problem. *European Journal of Operational Research*, 81(1):188–205, 1995.

[41] J. Matoušek and J. Nešetřil. *Invitation to Discrete Mathematics.* Clarendon Press, Oxford, 1998. ISBN 0-19-850208-7.

[42] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer-Verlag, Berlin, 1994.

[43] J. Pelikán. *Diskrétní modely.* VŠE, fakulta informatiky a statistiky, Praha, 1999.

[44] J. Plesník. *Grafové algoritmy.* VEDA, vydavateľstvo Slovenskej akadémie vied, Bratislava, 1983.

[45] J. Plesník. Heuristics for the Steiner Problem in Graphs. *Discrete Applied Mathematics*, 37/38:451–463, 1992.

[46] V.J. Rayward-Smith and A. Clare. On Finding Steiner Vertices. *Networks*, 16:283–294, 1986.

[47] C.R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications, Oxford, 1993.

[48] G. Robins and A. Zelikovsky. Improved Steiner Tree Approximation in Graphs. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms SODA 2000*, pages 770–779, San Francisco (California), 2000.

[49] J.S. Salowe and D.M. Warme. Thirty-Five Point Rectilinear Steiner Minimal Trees in a Day. *Networks*, 25:69–87, 1995.

[50] M.I. Shamos and D. Hoey. Closest Point Problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science FOCS '75*, pages 151–162, Berkeley (California), 1975.

[51] G. Skorobohatyj. Testsets. Report, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2000. `ftp://ftp.zib.de/pub/Packages/mp-testdata/index.html`.

[52] H. Takahashi and A. Matsuyama. An Approximate Solution for the Steiner Problem in Graphs. *Mathematica Japonica*, 24(6):573–577, 1980.

[53] M. Šeda. Aplikace Steinerových stromů v síťové optimalizaci. In *Proceedings of the 4th International Scientific-Technical Conference PROCESS CONTROL 2000 (ŘÍP 2000)*, Brno, 2000. 10 str. na CD-ROMu `\Rip2k\Sbornik\Texty\Rip044.pdf`; (Šeda, M.: Applications of Steiner Trees in Network Optimization, pp. 53).

[54] M. Šeda. Aproximativní a heuristické algoritmy pro řešení Steinerova problému. In *Sborník z 9. semináře Moderní matematické metody v inženýrství* 3μ, pages 154–158, Dolní Lomná u Jablunkova, 2000.

[55] M. Šeda. Computing Near-Optimal Solutions to the Network Steiner Tree Problem Using Approximate and Heuristic Techniques. In *Proceedings of the 13th International Conference PROCESS CONTROL '01*, Štrbské pleso (Slovakia), 2001. 11 pp. on CD-ROM `\Papers\Session4\p106.pdf`; (Summaries Volume, pp. 121).

[56] M. Šeda. Eppstein's Proof of the Steiner Ratio for Rectilinear Steiner Trees is Mistaken and How to Correct it. In *Proceedings of the 7th International Conference on Soft Computing MENDEL 2001*, pages 221–227, Brno, 2001.

[57] M. Šeda. Insertion Heuristic for the Euclidean Steiner Tree Problem. In *Proceedings of the XXIInd International Colloquium Advanced Simulation of Systems ASIS 2000*, pages 91–96, Sv. Hostýn - Bystřice pod Hostýnem, 2000.

[58] M. Šeda. On Finding Minimal Networks. In *Proceedings of the 4th Japan-Central Europe Joint Workshop on Energy and Information in Non-Linear Systems*, Brno, 2000. 4 pp, in print.

[59] M. Šeda. Solving Steiner Tree Problem Using Local Search Methods. In *Proceedings of the 22nd International Conference Telecommunications and Signal Processing TSP '99*, pages 102–105, Brno, 1999.

[60] M. Šeda. Solving the Shortest Path Problem on a Graph with Fuzzy Edge Lengths. In *Proceedings of the 9th Fuzzy Colloquium*, pages 247–253, Zittau (Germany), 2001.

[61] M. Šeda. Steinerovy problémy a přibližné metody jejich řešení. Habilitační práce, Vysoké učení technické v Brně, fakulta strojního inźenýrství, 2001. 153 str.

[62] M. Šeda. Stochastic Heuristic Methods for the Steiner Tree Problem in Graphs. In *Abstracts of the European Operational Research Conference EURO 2001*, page 74, Rotterdam (Netherlands), 2001.

[63] M. Šeda. Voronoi Diagrams and their Applications. In *Proceedings of the XXVIth Seminar ASR 2001 Instruments & Control*, pages 65/1–65/8, Ostrava, 2001.

[64] M. Šeda and Z. Lebiš. Heuristic Approaches for Rectilinear Steiner Tree Problem. In *Proceedings of the 6th International Conference on Soft Computing MENDEL 2000*, pages 151–156, Brno, 2000.

[65] M. Šeda and P. Nečas. Insertion Heuristic for the Euclidean Steiner Tree Problem. In *Proceedings of the XXIIIrd International Colloquium Advanced Simulation of Systems ASIS 01*, pages 71–76, Velké Losiny, 2001.

[66] M. Šeda and D. Smutek. Metaheuristics for the Graphical Steiner Tree Problem. In *Proceedings of the 12th International DAAAM Symposium*, Jena (Germany), 2001. 2 pp, in print.

[67] D.M. Warme. A New Exact Algorithm for Rectilinear Steiner Trees. In *Proceedings of the INFORMS Conference*, San Diego, 1997. 42 pp.

[68] P. Winter. The Steiner Problem in Networks: A Survey. *Networks*, 17:129–167, 1987.

[69] M. Zachariasen. Rectilinear Full Steiner Tree Generation. *Networks*, 33:125–143, 1999.

[70] A. Zelikovsky. An 11/6 Approximation Algorithm for the Network Steiner Problem. *Algorithmica*, 9:463–470, 1993.

Shrnutí

Cílem práce bylo podat výklad Steinerových problémů, které tvoří významnou podmnožinu úloh zaměřených na hledání minimálních sítí. Jediným kritériem, které zde bylo použito, byla délka (popř. náklady na zřízení) výsledné sítě. Přes velký význam je problematika Steinerových stromů v české literatuře zatím zcela opomíjena, jak je patrné i ze seznamu odkazů. V učebnicích teorie grafů se informace nejčastěji omezují pouze na několikařádkovou poznámku popisující Steinerův problém jako variantu problému hledání minimální kostry anebo zcela chybí.

Záměrem bylo podat srozumitelným způsobem ty partie zkoumané problematiky, které se v originále odkazují na desítky jiných pramenů, resp. jsou uvedeny velmi zhuštěným způsobem (např. v důkazu věty je uvedeno, že tvrzení plyne z věty 3, lemmat 4,5, věty 6 a důsledku 7, popř. jsou zcela ponechány na laskavém čtenáři) anebo v nedostatečně obecné podobě (např. důkaz NP-úplnosti rektilineárního problému).

Řada vět, lemmat a tvrzení byla autorem formulována a dokázána bez přímé analogie na studovanou literaturu, a to tak, aby výklad Steinerova problému v grafech, rektilineárního Steinerova problému a euklidovského Steinerova problému byl strukturálně jednotný. Renomovaní autoři jsou totiž velmi často specializováni, např. Hwang, Warme a Zachariasen na geometrické varianty problému (rektilineární a euklidovský), Ganley a Robins na rektilineární problém, Voss a Hougardy na problém v grafech, Zelikovsky na rektilineární problém a problém v grafech apod. Mimo jiné na jednoduchém příkladě byl vyvrácen Eppsteinův důkaz pro určení Steinerova poměru při aproximaci rektilineárního Steinerova minimálního stromu rektilineární minimální kostrou a byla navržena jeho modifikace, která vychází z jeho strategie konstrukce dvou koster, jejich odlišně navržená topologie však v žádném případě neporušuje mez 1.50 pro poměr chování.

Časová složitost některých algoritmů byla "přepočítána" s přihlédnutím k použitým efektivním datovým strukturám. Konkrétně se to týká použití prioritní fronty implementované binární haldou, což umožnilo zmenšit časovou náročnost Jarníkova a Dijkstrova algoritmu, a to se pak promítlo do dvou aproximativních algoritmů pro hledání Steinerova minimálního stromu v grafu.

Protože, rozhodovací verze Steinerových problémů patří do třídy NP-úplných problémů, přesné řešení bylo možné očekávat pouze pro malé rozsahy dat. Na autorem navrženém matematickém modelu se prokázalo, že i profesionální softwarový nástroj GAMS (patrně založený na metodě větví a mezí) není schopen v dostupném čase přesně vyřešit Steinerův problém v grafech s více jak 50 vrcholy. Na druhé straně (poněkud v rozporu s očekáváním autora po zkušenostech s jejich nasazeních v problémech rozvrhování) se nepotvrdila obecná použitelnost metaheuristik pro všechny Steinerovy problémy. Rozhodně je nelze doporučit pro rektilineární problém. Svůj význam mají především u grafické verze Steinerova problému, kde Steinerův poměr i těch nejlepších aproximativních algoritmů je nepříznivý. U euklidovského Steinerova problému jeho Steinerův poměr $\frac{2}{\sqrt{3}}$ ukazuje, že aproximace minimální kostrou je natolik dobrým přiblížením Steinerově minimálnímu stromu, že stačí výchozí řešení zlepšit jednoduchou deterministickou heuristikou.

Budoucím cílem bude zobecnit Steinerovy problémy na případ, kdy vzdálenosti jsou dány fuzzy čísly. Zatím byl implementován algoritmus pro hledání nejkratší cesty s délkami hran zadanými fuzzy čísly [60], který bude použit pro fuzzy variantu Steinerova problému v grafech.