# Faster Exact Algorithms for Computing Steiner Trees in Higher Dimensional Euclidean Spaces

Marcus Brazil [*] Rasmus Fonseca [†] Pawel Winter [‡] Martin Zachariasen [§]

**Abstract**

Given a set of $n$ points (known as terminals) in $d$-dimensional space...[TODO]

**Keywords:** Steiner tree problem, $d$-dimensional Euclidean space, exact algorithm, computational study.

## 1 Introduction

Given a finite set of points $N$ in a $d$-dimensional space ($d \geq 2$), the Euclidean Steiner tree problem (ESTP) asks for a shortest possible interconnection of the points in $N$. The Euclidean Steiner tree problem in the plane ($d = 2$) has a history that goes back more than two centuries [**?**], while the generalisation to more than two dimensions first appears to be mentioned by Bopp [2] in 1879; more recent mathematical treatments can be found in [10, 8]. The problem in 3-space has a number of applications in computational biology [14].

While numerous exact algorithms have been proposed for the problem in the plane — and problem instance with several thousand points ca be solved to optimality [**?**] — significantly less improvement has been made for the problem in higher dimensions ($d \geq 3$). Currently, only one main algorithmic approach exists for the problem — proposed by Smith [13] in 1992.

,by the only realistic approach

**Background and Motivation.**

[Preliminary/old notes on the 2d problem]

The Euclidean Steiner tree problem is known to be NP-hard [7], even when the terminals are restricted to lie on two parallel lines [12]. The Melzak construction [**?**] forms the building block

[*]Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia, brazil@unimelb.edu.au

[†]Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen Ø, Denmark, rfonseca@di.ku.dk

[‡]Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen Ø, Denmark, pawel@di.ku.dk

[§]Department of Computer Science, University of Copenhagen, DK-2100 Copenhagen Ø, Denmark, martinz@di.ku.dk

of the most successful algorithm for the problem — the GeoSteiner algorithm ([**?**], [16], [17] and [19]). Using the GeoSteiner algorithm, optimal solutions to the Euclidean Steiner tree problem for thousands of terminals can be computed in reasonable time. Approximate solutions can be computed efficiently in theory and practice; the so-called polynomial-time approximation scheme of [**?**] provides solutions that are a factor of $1 + \epsilon$ away from optimum in polynomial-time for any fixed $\epsilon > 0$.

Euclidean Steiner minimal trees (ESMTs) are unions of Euclidean full Steiner trees (EFSTs) whose terminals are incident with one FST-edge each. An EFST spanning $k$ terminals, $1 \leq k \leq n$, has $k - 2$ Steiner points. Steiner points in EFSTs have three incident edges meeting at $120^o$.

First exact algorithms for the ESTP in $R^2$ are based on the following common framework [**?**]. Subsets of terminas are considered one by one. For each subset, all its EFSTs are determined and the shortest is retained. Several test can be applied to these retained EFSTs in order to decide if they can be in an ESMT. Surviving EFSTs are then concatenated in all possible ways to obtain trees spanning all terminals. The shortest of them is an ESMT.

The bottleneck of this approach is the generation of EFSTs. It has been observed [**?**] that substantial improvements can be obtained if EFSTs are generated simultaneously across various subsets of terminals. Very powerful geometrical tests for the identification of non-optimal EFSTs can then be applied not after but *during* the generation of EFSTs. As a consequence, the concatenation of EFST became a bottleneck of EFST-based algorithms for the ESTP.

An improvement of the concatenation [**?**] was based on the observation that the concatenation of EFSTs can be formulated as a problem of finding a minimum spanning tree in a hypergraph with terminals as vertices and subsets spanned by EFSTs as hyperedges. This problem can be solved using branch-and-cut.

The dramatical improvements of both the generation and concatenation of EFST led to the development of GeoSteiner [] which can routinely solve problem instances with thousands of terminals in a reasonable amount of time [**?**]. Similar methodology has been applied to the rectilinear Steiner tree problem (also a part of GeoSteiner).

The methodology of GeoSteiner seems at first sight to be applicable to the ESTP in $d$-dimensional spaces, $d \geq 3$. Unfortunately, this is not the case. First of all, the generation of EFSTs is far from as simple as in the plane. Any geometrical construction approach for $d \geq 3$ would require solving eigth-degree polynomials [**?**]. As a consequence, numerical approaches are the only way to approximate optimal EFSTs for a given subset of terminals. Such numerical approaches seem to block the generation of (optimal) EFSTs across various subsets of terminals (as is the case in the plane). Furthermore , the geometrical non-optimality tests are much weaker than in the plane.

While the generation of EFST seems to be very troublesome for $d \geq 3$, the concatenation can be applied without any significant modifications. However, the problem is of course that the lack of geometrical tests in the generation phase will permit a huge number of non-optimal EFSTs to survive causing the branch-and-cut algorithm to choke. These congestion problems become more and more serious as $d$ grows.

A numerical approach to solve ESTP is available [**?**]. It determines EFSTs for the entire set of terminals (rather than for subsets of terminals). As a consequence, the concatenation phase becomes obsolete. In order to determine the EFSTs for the entire set, all its full Steiner topologies must be generated. For each such topology, a numerical method is used to obtain a *relatively minimal tree*. Such a tree may have some Steiner points overlapping with terminals or with each other. The generation of full Steiner topologies is achieved by a straightforward expansion procedure. Assume that a full Steiner topology $\mathcal{T}_k$ for terminals $t_1, t_2, , t_k$, $3 \leq k < n$, is given. Note that for $k = 3$, only one such full Steiner topology exists. Expand $\mathcal{T}_k$ into $2k5$ full Steiner topologies of $k + 1$ terminals by inserting a new Steiner point (adjacent to $t_{k+1}$) into every edge of $\mathcal{T}_k$. This expansion process stops when $k = n$. It can be shown that it generates every full Steiner topology of $n$ terminals exactly once.

Given a full Steiner topology for all $n$ terminals, arbitrary initial positions are assigned to its $n-2$ Steiner points. The positions of Steiner points are then recomputed iteratively by solving a system of $n-2$ equations with $n-2$ unknown (corresponding to the locations of Steiner points). It has been shown [**?**] that the length of the tree reduces with each iteration and it converges to the unique (possible degenerate) *relatively minimal tree* (RMT). The iterative process terminates if all pairs of incident edges meet at Steiner points at angles within the interval $[2\pi/3 - \epsilon, 2\pi/3 + \epsilon]$ for an arbitrarily small constant $\epsilon > 0$.

This approach is quite slow and practical only for $n \leq 12$. An improved version [**?**] is still slow but it works for $n \leq 16$.

The approach suggested in this paper can be seen as a compromise between the approach used by GeoSteiner and the numerical approach. Rather than generating all full Steiner topologies for $n$ terminals, we generate topologies of *branches* involving subsets of $n$ terminals. Consider a full Steiner topology with $n$ terminals. Remove one of its Steiner points. Then the full Steiner topology breaks into 3 *branches*. As will be seen below that every full Steiner tree contains a Steiner point where none of the branches has more than $\lceil n/2 \rceil$ terminals. This significanly reduces the number of branches that need to be generated. For each branch, a reasonable lower bound on its length can be determined. More precisely, consider a full Steiner topology obtained from the topology of a a particular branch (with at least 2 terminals) by bypassing its Steiner point of degree 2. The iterative procedure of the approximation algorithm can be used to approximate (from below) the length of the RMT with this full Steiner topology. Some of the generated branches can be pruned away if these lower bounds and some (mild) geometrical tests exclude the possibility that they can occur in the ESMT. Once all not pruned branches are generated, triplets of them are picked up. Branches in each triplet must span different terminals and all terminals must be spanned by them. If the sum of lower bounds of the selected triplet of branches is less than the best solution found so far, the iterative prcedure of the approximation algorithm can be used to approximate the RMT with the full Steiner topology given by the triplet of branches (spliced together via a Steiner point). Using this approach problem instances with up to 20? terminals can be solved in a reasonable amount of time.

## 2 Preliminaries

The well-known Euclidean Steiner minimal tree problem asks for a network, $T^*$, of minimum total length interconnecting a given finite set of points in $d$-dimensional Euclidean space. A Steiner tree

By the *length* of a tree $T$, we mean the sum of the lengths of the edges of $T$. A tree $T^*$ that interconnects a given set of points, $N$, with length $|T^*|$, where $|T^*|$ is as small as possible, is called a Steiner minimal tree (SMT) for $N$. The given points $N$ are called the *terminals* of $T$.

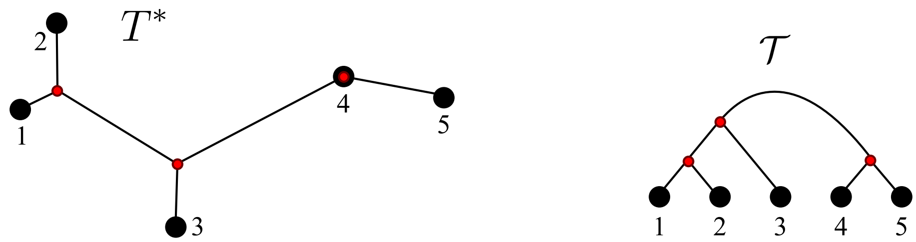


Figure 1: The Steiner minimal tree of five two-dimensional terminals (black). Steiner points are shown as smaller red circles. The topology $\mathcal{T}$ of $T^*$ is shown to the right.

The topology of a tree $T$ is a graph that indicates how terminals and Steiner points are connected. It can be shown [ref] that any SMT can be represented by a topology where all terminals have degree 1 and all Steiner points degree exactly 3. In this case Steiner points might overlap with terminals as shown in Figure 1. In practice, $T$, is approximated as the relatively minimal tree obtained from the terminal positions and the topology.

Since all Steiner points have degree 3, the removal of a Steiner point will cause the topology to split into three binary trees, $\mathcal{T}_1, \mathcal{T}_2$ and $\mathcal{T}_3$ that will be denoted *branches*. Each branch has an unconnected half-edge pointing from its root. Note that the root of a branch can be removed by edge-contracting either its left- or right-edge such that the resulting graph is a topology. In the following sections a lower bound computation is described that relies on assigning a length to a branch. For convenience, we therefore let $\mathrm{RMT}(\mathcal{T})$ denote the the relatively minimal tree of the topology that results from edge-contracting away the root of $\mathcal{T}$ (see Figure 2). Any branch, $\mathcal{T}$, for which $\mathrm{RMT}(\mathcal{T}) = T^*$ is denoted $\mathcal{T}^*$.

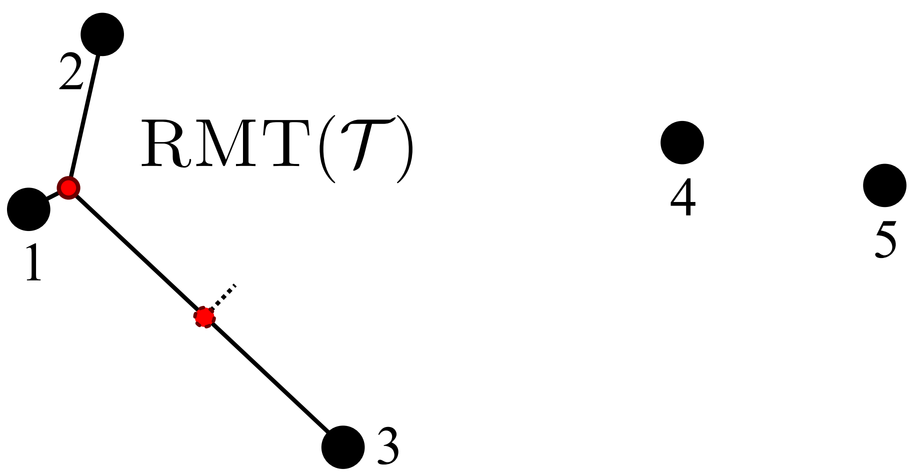


Figure 2: A branch $\mathcal{T}$ and $\mathrm{RMT}(\mathcal{T})$.

# 3 GeoSteiner approach

GeoSteiner identifies $T^*$ in three phases (see Algorithm 1). The first is a *preprocessing phase* that computes SMTs for subsets of terminals. The second phase *generates branches* containing up to $\lceil \frac{n}{2} \rceil$ terminals while trying to prune as many as possible that can not be part of $\mathcal{T}^*$. The third phase *concatenates branches* so they containing all $n$ terminals and evaluates the full topologies they correspond to.

---

**Algorithm 1** Generalized GeoSteiner FST algorithm

---

**Input:** Set of $d$-dimensional terminals, $N$

**Output:** The FST $\mathcal{T}^*$ containing the full component of the minimum Steiner tree for $N$

1: Let $\Gamma_p$ be the set of FSTs for all $S \subset N, |S| = \min(8, n/2)$.

2: Let $\Gamma_i, i = 1 \ldots \lceil \frac{n}{2} \rceil$ be sets of branches containing $i$ terminals such that $\mathcal{T}^*$ is guaranteed to be a concatenation of three branches from $\Gamma_i$.

3: Test all concatenations of branches from $\Gamma_i$ that has exactly $n$ non-overlapping terminal sets and return the one corresponding to the minimum Steiner tree.

---

The purpose of preprocessing subsets of terminals is to strengthen the pruning using lower bound computations in the branch generation phase.

**Lemma 3.1** *If the branch $\mathcal{T}$ is a subtree of $\mathcal{T}^*$ then $|\mathrm{RMT}(\mathcal{T})|$ is a lower bound on $|T^*|$.*

**Proof** ..

**Lemma 3.2** *Let $\mathcal{T}_S$ be a branch containing the terminals $S$ and let $S' \subseteq N \backslash S$. Adding the lengths of $\mathrm{RMT}(\mathcal{T}_S)$ and the SMT of $S'$ gives a lower bound on $|T^*|$.*

**Proof** ..

Following Lemma 3.2 a branch can be discarded if its smallest length added to the length of a preprocessed tree containing a different set of terminals exceeds an upper bound on $|T^*|$. The preprocessed SMTs are used in a similar way to prune branches during the concatenation phase.

## 3.1 Preprocessing phase

[Choice of $s$ (right now $\min(8, \lfloor n/2 \rfloor)$). ]

## 3.2 Generate branches

Branches are generated by increasing number of contained terminals, $i$, and placed in corresponding lists, $\Gamma_i$. A terminal is considered a branch containing 1 terminal and therefore placed in $\Gamma_1$. The branches in a particular list, $\Gamma_k,\ k \geq 2$, are generated by combining all branches from level $l$ with all from level $k - l$ for $l = 1 \ldots \lfloor k/2 \rfloor$. There are three criteria that can be used to reject a branch $\mathcal{T}_k = (\mathcal{T}_l, \mathcal{T}_{k-l})$.

**Algorithm 2** Generate optimal FSTs for all subsets of $N$ of size $s$

**Input:** Set of $d$-dimensional terminals, $N$
**Input:** A subset size $s \in \mathbb{N}$
**Output:** The optimal FSTs, $\Gamma_p$, for all subsets $N_s \subset N$ of size $s$ sorted in descending order

```
Γ_p ← {}
for N_s ⊂ N, |N_s| = s do
    Let 𝒯_s be the optimal FST for N_s.
    Γ_p.add(𝒯_s)
end for
Sort Γ_p in descending order by length of FST.
return Γ_p
```

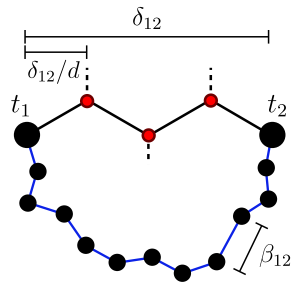


Figure 3: Euclidean distance, $\delta_{12}$ and bottleneck distance $\beta_{12}$, between two terminals connected by $d = 4$ edges. If the smallest value of the longest Steiner tree edge ($\frac{\delta_{12}}{d}$) exceeds $\beta_{12}$ then the branch can not be part of the optimal tree.

First, if $\mathcal{T}_l$ and $\mathcal{T}_r$ contain the same terminal then their combination will not be an FST and can be rejected.

Second, if two terminals are connected by a path in the minimum spanning tree of $N$ consisting of edges that are all shorter than the longest edge between the terminal in $\mathrm{RMT}(\mathcal{T})$ then $\mathcal{T}$ can not be part of $T^*$. [Expand a bit]. This check can be performed without computing the relative minimal tree. The bottleneck distance, $\beta_{12}$, between two terminals $t_1$ and $t_2$ is defined as the length of the longest edge on the path from $t_1$ to $t_2$ in the minimum spanning tree (see Figure 3). The euclidean distance is denoted $\delta_{12}$. Bottleneck distances are computed for each pair of terminals before the main loop starts. Assume that combining branches $\mathcal{T}_1$ and $\mathcal{T}_2$ will result in $d$ edges between terminals $t_1$ and $t_2$. The shortest possible length of the longest edge connecting them will be $\frac{\delta_{12}}{d}$. If this edge is longer than $\beta_{12}$ then it would be possible to remove it and replace it with the bottleneck edge in the minimum spanning tree.

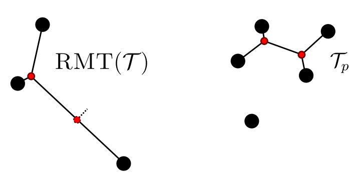


Figure 4: The length of RMT(T) added to the MST of any subset of terminals disjoint from $\mathcal{T}$ is a lower bound on $|T^*|$

Third, if $\mathcal{T}$ has a lower bound which is higher than the length of any previously observed network connecting $N$ it can be pruned. As previously noted, the length of RMT(T) added to one of the SMTs computed in the preprocessing phase gives a lower bound on $|T^*|$. As an upper bound it is possible to use the length of the minimum spanning tree, but since it is crucial to prune as many branches as early as possible we use a heuristic [ref heur paper] to find an even lower upper bound.

**Algorithm 3** Generate branches containing up to $\lceil \frac{n}{2} \rceil$ terminals

**Input:** Set of $d$-dimensional terminals, $N$
**Input:** Sorted list, $\Gamma_p$, of optimal FSTs for subsets of $N$.
**Input:** An upper bound, $U$, on $\mathrm{RMT}(\mathcal{T}^*)$
**Output:** $\Gamma_i, i = 1 \ldots \lceil \frac{n}{2} \rceil$

**for** $k = 1$ **to** $\lceil \frac{n}{2} \rceil$ **do**
  **for** $l = 1$ **to** $\lfloor \frac{k}{2} \rfloor$ **do**
    **for all** $\mathcal{T}_1 \in \Gamma_l$ **do**
      **for all** $\mathcal{T}_2 \in \Gamma_{k-l}$ **do**
        **for all** pairs of terminals $(t_1, t_2)$ where $t_1 \in \mathcal{T}_1, t_2 \in \mathcal{T}_2$ **do**
          **if** $t_1 = t_2$ **then**
            The concatenation of $\mathcal{T}_1$ and $\mathcal{T}_2$ can not be part of $\mathcal{T}^*$
          **end if**
          Let $d_1$ denote the depth of $t_1$ in $\mathcal{T}_1$, and vice versa for $d_2$
          Let $\delta_{12}$ denote the euclidean distance between $t_1$ and $t_2$
          Let $\beta_{12}$ denote the bottleneck distance between $t_1$ and $t_2$
          **if** $\frac{2}{\sqrt{3}} \frac{\delta_{12}}{\beta_{12}} > d_1 + d_2 + 1$ **then**
            The concatenation of $\mathcal{T}_1$ and $\mathcal{T}_2$ can not be part of $\mathcal{T}^*$
          **end if**
        **end for**
        Let $\mathcal{T}$ be the concatenation of $\mathcal{T}_1$ and $\mathcal{T}_2$
        Locate the largest FST, $\mathcal{T}_p \in \Gamma_p$, that doesn't share a terminal with $\mathcal{T}$
        **if** $|\mathrm{RMT}(\mathcal{T})| + |\mathrm{RMT}(\mathcal{T}_p)| > U$ **then**
          The concatenation of $\mathcal{T}_1$ and $\mathcal{T}_2$ can not be part of $\mathcal{T}^*$
        **end if**
        If none of the above tests failed put $\mathcal{T}$ in $\Gamma_k$
      **end for**
    **end for**
  **end for**
**end for**
**return** $\Gamma_i, i = 1 \ldots \lceil \frac{n}{2} \rceil$

### 3.3 Concatenate branches

Consider an FST $\mathcal{T}$ with $n$ terminals, $n \geq 3$. Let $s$ be any of its $n-2$ Steiner points. When $s$ (and its three incident edges) are removed, $\mathcal{T}$ splits into three subtopologies $\mathcal{T}_i$, $\mathcal{T}_j$ and $\mathcal{T}_k$ with respectively $n_i$, $n_j$, and $n_k$ terminals, $n = n_i + n_j + n_k$. Assume that $n_i \geq n_j \geq n_k$.

**Lemma 3.3** *$\mathcal{T}$ has a splitting Steiner point $s$ such that $n_i \leq n_j + n_k$.*

**Proof**. Assume that $n_i > n_j + n_k$ for every Steiner points in $\mathcal{T}$. Pick a Steiner point $s$ minimizing $n_i$. Let $s' \in \mathcal{T}_i$ denote the Steiner point adjacent to $s$ in $\mathcal{T}$. It exists since $n_i \geq 2$. Let $n'_i, n'_j$, and $n'_k$ denote the number of terminals obtained by splitting $\mathcal{T}$ at $s'$. Then $n'_i = n_j + n_k$, $n'_j = x$ for some $x$, $0 < x < n_i$, and $n'_k = n_i - x$. Hence, $n'_i$, $n'_j$, and $n'_k$ are all less than $n_i$, contradicting the choice of $s$.

A split of any FST $\mathcal{T}$ with $n_i \geq n_j \geq n_k$, $n = n_i + n_j + n_k$ and $n_i \leq n_j + n_k$ is called a *canonical split*.

**Lemma 3.4** $\lceil \frac{n}{3} \rceil \leq n_i \leq \lfloor \frac{n}{2} \rfloor$ *in a cananical split of any* $FST$ *with* $n$ *terminals.*

**Proof**. To obtain the first inequality, observe that $n = n_i + n_j + n_k \Rightarrow n \leq 3n_i \Rightarrow \lceil \frac{n}{3} \rceil \leq n_i$. To obtain the second inequality, observe that $n_i \leq n_j + n_k = n - n_i \Rightarrow 2n_i \leq n \Rightarrow n_i \leq \lfloor \frac{n}{2} \rfloor$.

**Lemma 3.5** *An FST* $\mathcal{T}$ *with* $n$ *terminals has exactly one canonical split unless* $n$ *is even and* $n_i = n/2$ *in which case* $\mathcal{T}$ *has two canonical splits at adjacent Steiner points.*

**Proof**. Let $\mathcal{T}_i$, $\mathcal{T}_j$ and $\mathcal{T}_k$ denote three subtopologies of the canononical split at a Steiner point $s$. Consider another Steiner point $s' \in \mathcal{T}$. Assume first that $s' \in \mathcal{T}_j$. Let $n'_i \geq n'_j \geq n'_k$ denote the number og terminals in the subtopologies of this split at $s'$. Hence, $n'_i \geq n_i + n_k$ and $n'_j + n'_k \leq n_j$. Now $n'_i \geq n_i + n_k > n_j \geq n'_j + n'_k$ implies that the split is not canonical. Similar argument applies if $s' \in \mathcal{T}_k$. Assume finally that $s' \in \mathcal{T}_i$. Hence, $n'_i \geq n_j + n_k$. Now $n'_i \geq n_j + n_k \geq n_i \geq n'_j + n'_k$ implies that this split is canonical iff $n'_i = n'_j + n'_k$. This can only happen iff $n'_i = n_i = n/2$. Hence, $n$ has to be even and $s$ and $s'$ must be adjacent in $\mathcal{T}$.

It is therefore only necessary to concatenate branch triples $(\mathcal{T}_i, \mathcal{T}_j, \mathcal{T}_k)$ that don't contain the same terminal twice, have $n = n_i + n_j + n_k$, and whose individual sizes are between $\lceil \frac{n}{3} \rceil \leq n_i \leq \lfloor \frac{n}{2} \rfloor$.

For implementational convenience the FST is represented as a branch. Since the Steiner point at the root of the branch is removed, RMT($\mathcal{T}$) simply results in the relatively minimal tree of the FST.

# 4 Pruning methods

BSD

Lower bounds (WS + eq-points)

# 5 Computational experience

# 6 Conclusions

### Acknowledgements

The first author was partially supported in this research by an Australian Research Council Discovery Grant.

**Algorithm 4** Concatenate triples of branches and determine optimal

**Input:** Sets of branches $\Gamma_i, i = 1 \ldots \lfloor \frac{n}{2} \rfloor$
**Input:** An upper bound, $U$, on $\mathrm{RMT}(\mathcal{T}^*)$
**Output:** The optimal FST, $\mathcal{T}^*$

$T^* \leftarrow$ NIL
**for** $i = \lfloor \frac{n}{2} \rfloor$ **to** $\lceil \frac{n}{3} \rceil$ **do**
  **for** $j = i$ **to** $\lceil \frac{n-i}{2} \rceil$ **do**
    $k \leftarrow n - i - j$
    **if** $k < 1$ **then**
      **continue**
    **end if**
    **for all** distinct triples $(\mathcal{T}_i, \mathcal{T}_j, \mathcal{T}_k)$ where $\mathcal{T}_i \in \Gamma_i, \mathcal{T}_j \in \Gamma_j$ and $\mathcal{T}_k \in \Gamma_k$ **do**
      **if** $\mathcal{T}_i, \mathcal{T}_j$ or $\mathcal{T}_k$ contain the same terminal twice **then**
        **continue**
      **end if**
      Let $\mathcal{T}$ be the branch $((\mathcal{T}_i, \mathcal{T}_j), \mathcal{T}_k)$.
      **if** $|\mathrm{RMT}(\mathcal{T})| < |T^*|$ **then**
        $T^* \leftarrow \mathrm{RMT}(\mathcal{T})$
      **end if**
    **end for**
  **end for**
**end for**
**return** $T^*$

| $n$ | SMITH #RMT | SMITH $\lvert RMT \rvert$ | SMITH Time | SMITH* #RMT | SMITH* $\lvert RMT \rvert$ | SMITH* Time | BRANCHENUMERATION #RMT | BRANCHENUMERATION $\lvert RMT \rvert$ | BRANCHENUMERATION Time |
|---|---|---|---|---|---|---|---|---|---|
| 11 | 186 293 | 9.6 | 8.6 | 23 522 | 9.3 | 1.8 | 29 287 | 7.1 | 1.7 |
| 11 | 181 860 | 9.8 | 8.2 | 16 024 | 9.0 | 1.1 | 24 512 | 7.0 | 2.3 |
| 11 | 5 375 525 | 10.7 | 340.1 | 156 317 | 9.8 | 11.6 | 226 952 | 8.6 | 19.4 |
| 12 | 534 748 | 10.6 | 29.8 | 112 722 | 10.1 | 9.3 | 185 885 | 9.7 | 24.5 |
| 12 | 540 271 | 10.7 | 25.2 | 139 255 | 9.4 | 6.2 | 58 864 | 8.3 | 12.7 |
| 12 | 1632 120 | 10.0 | 80.0 | 15 616 | 9.1 | 1.3 | 71 889 | 9.0 | 12.2 |
| 12 | 7494 024 | 10.9 | 603.6 | 51 688 | 10.0 | 4.0 | 42 074 | 7.3 | 11.4 |
| 12 | 1 994 940 | 10.6 | 94.5 | 168 303 | 10.4 | 14.5 | 114 185 | 9.3 | 25.3 |
| 13 | 75 851 228 | 12.5 | 4926.9 | 2 284 379 | 11.2 | 195.1 | 3 969 103 | 9.7 | 524.0 |
| 13 | 8 988 538 | 11.0 | 750.5 | 163 987 | 10.8 | 15.0 | 267 858 | 8.8 | 29.4 |
| 13 | 3 724 158 | 11.0 | 185.0 | 399 432 | 10.7 | 32.3 | 202 386 | 8.7 | 40.8 |
| 14 | 14 034 712 | 11.0 | 735.1 | 958 368 | 11.4 | 87.6 | 749 897 | 9.3 | 204.0 |
| 14 | 10 508 107 | 11.7 | 674.0 | 6399672 | 11.7 | 572.4 | 521 997 | 10.4 | 196.7 |
| 15 | 45 785 568 | 13.0 | 4315.8 | 8 665 008 | 13.4 | 843.8 | 1 431 567 | 10.6 | 411.2 |
| 15 | 385 590 983 | 13.2 | 22570.6 | 2 121 425 | 11.3 | 180.9 | 2 676 981 | 9.5 | 472.8 |
| 16 | 1 221 022 | 13.3 | 76.9 | 101 937 | 11.2 | 9.6 | 154 846 | 8.4 | 1536.8 |
| 16 | | | | 3 139 934 | 11.6 | 250.7 | 179 104 | 8.6 | 3477.8 |
| 17 | | | | 67 278 935 | 13.5 | 4569.6 | 22 755 368 | 10.3 | 10168.0 |
| 17 | | | | | | | 136 355 832 | 10.7 | 54223.6 |
| 17 | | | | 80 744 814 | 13.9 | 5353.5 | 16 903 799 | 10.6 | 9398.9 |
| 17 | | | | 92 124 614 | 12.4 | 5334.4 | 64 095 211 | 10.5 | 20378.2 |
| 17 | | | | 19 222 828 | 13.2 | 1179.9 | 24 975 144 | 10.3 | 13548.2 |
| 18 | | | | | | | 4 959 344 | 10.3 | 10090.0 |

Table 1: Properties of algorithms run on 3D problem instances from the Carioca set.

# References


[1] J. E. Beasley. OR-library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072, 1990.

[2] K. Bopp. *Üeber das kürzeste Verbindungssystem zwischen vier Punkten*. PhD thesis, Universität Göttingen, 1879.

[3] M. Brazil, R. L. Graham, D. A. Thomas, and M. Zachariasen. On the history of the Euclidean Steiner tree problem. *Archive for History of Exact Sciences*, 68:327–354, 2014.

[4] M. Brazil, D. A. Thomas, B. K. Nielsen, P. Winter, C. Wulff-Nilsen, and M. Zachariasen. A novel approach to phylogenetic trees: $d$-dimensional geometric Steiner trees. *Networks*, 53(2):104–111, 2009.

[5] L. L. Cavalli-Sforza and A. W. F. Edwards. Phylogenetic analysis: Models and estimation procedures. *Evolution*, 21:550–570, 1967.

[6] M. Fampa and K. M. Anstreicher. An improved algorithm for computing Steiner minimal trees in euclidean $d$-space. *Discrete Optimization*, 5:530–540, 2008.

[7] M. R. Garey, R. L. Graham, and D. S. Johnson. The complexity of computing Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 32(4):835–859, 1977.

[8] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM Journal on Applied Mathematics*, 16(1):1–29, 1968.

[9] F. K. Hwang, D. S. Richards, and P. Winter. *The Steiner Tree Problem*. Annals of Discrete Mathematics 53. Elsevier Science Publishers, Netherlands, 1992.

[10] V. Jarník and M. Kössler. O minimálních grafeth obeahujících n daných bodú. *Cas. Pest. Mat. a Fys.*, 63:223–235, 1934.

[11] J. W. Van Laarhoven and K. M. Anstreicher. Geometric conditions for Euclidean Steiner trees in $\mathbb{R}^d$. *Computational Geometry*, 46:520–531, 2013.

[12] J. H. Rubinstein, D. A. Thomas, and N. C. Wormald. Steiner trees for terminals constrained to curves. *SIAM Journal on Discrete Mathematics*, 10(1):1–17, 1997.

[13] W. D. Smith. How to Find Steiner Minimal Trees in Euclidean $d$-Space. *Algorithmica*, 7(2/3):137–177, 1992.

[14] C. Stanton and J. M. Smith. Steiner trees and 3-d macromolecular conformation. *INFORMS Journal on Computing*, 16:470–485, 2004.

[15] D. M. Warme. A New Exact Algorithm for Rectilinear Steiner Minimal Trees. Technical report, System Simulation Solutions, Inc., Alexandria, VA 22314, USA, 1997.

[16] D. M. Warme, P. Winter, and M. Zachariasen. Exact solutions to large-scale plane Steiner tree problems. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 979–980, 1999.

[17] D. M. Warme, P. Winter, and M. Zachariasen. GeoSteiner 3.1. Department of Computer Science, University of Copenhagen (DIKU), http://www.diku.dk/geosteiner/, 2001.

[18] P. Winter. An Algorithm for the Steiner Problem in the Euclidean Plane. *Networks*, 15:323–345, 1985.

[19] P. Winter and M. Zachariasen. Euclidean Steiner minimum trees: An improved exact algorithm. *Networks*, 30:149–166, 1997.