A specialized branch-and-bound algorithm for the Euclidean Steiner tree problem in n-space

Marcia Fampa
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, Brazil
fampa@cos.ufrj.br

Jon Lee
University of Michigan
Ann Arbor, Michigan, USA
jonxlee@umich.edu

Wendel Melo
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ, Brazil
wendelmelo@cos.ufrj.br

ABSTRACT

We present a specialized branch-and-bound (B&B) algorithm for the Euclidean Steiner tree problem (ESTP) in $\mathbb{R}^n$ and apply it to a convex mixed-integer nonlinear programming (MINLP) formulation of the problem, presented by Fampa and Maculan. Our main emphasis is on isomorphism pruning, in order to prevent the algorithm to solve equivalent subproblems corresponding to isomorphic Steiner trees. The concept of representative Steiner topologies is introduced, which allows pruning these subproblems and fixing variables. A more general procedure is also proposed to improve the efficiency of the B&B, that may be extended to the solution of a variety of problems. Computational results demonstrate substantial gains compared to the standard B&B for convex MINLP.

KEYWORDS: Euclidean Steiner tree problem, branch-and-bound, isomorphism pruning.

Main area: PM - Mathematical Programming

1 Introduction

The Euclidean Steiner tree problem (ESTP) is to find a tree of minimal Euclidean length spanning a given set of points in $\mathbb{R}^n$, using or not additional points in its construction. Given points are called *terminals*, and additional points are *Steiner points*. The ESTP is NP-Hard (Hwang et al. 1992) and is notoriously difficult to solve in dimension greater than 2. An optimal solution of ESTP, known as a Steiner minimal tree (SMT), satisfies some well-known properties (Hwang et al. 1992): (i) an SMT on p terminals has at most $p - 2$ Steiner points, (ii) each terminal has degree between one and three, and (iii) each Steiner point has degree three.

The *topology* of a Steiner tree is an important concept in solution methods for the ESTP and is defined as the tree where the connections between terminals and Steiner points are specified, but the locations of the Steiner points are not. A full Steiner topology (FST) for p terminals is a topology with $p-2$ Steiner points, where all the terminals have degree one. Any SMT with non-full topology can be related to a FST where some edges have zero length.

Only a few papers have considered the exact solution for the ESTP in $\mathbb{R}^n$, when $n \geq 3$ (see (Gilbert and Pollak 1968, Smith 1992, Maculan et al. 2000, Fampa and Maculan 2004, Fampa and Anstreicher 2008, Van Laarhoven and Anstreicher 2013)). Mathematical-programming formulations were only presented in (Maculan et al. 2000, Fampa and Maculan 2004). In (Maculan et al. 2000) the problem is formulated as a non-convex mixed-integer nonlinear programming (MINLP) problem, and in (Fampa and Maculan 2004) the model is convexified using a Big-M parameter (we use the standard terminology of *convex MINLP* for an MINLP that has a convex continuous relaxation). No computational results based on the formulations were presented.

Our objective is to consider the formulation for the ESTP in (Fampa and Maculan 2004), presented in §2, and take advantage of recent advances convex MINLP to solve the ESTP. Numerical experiments with the model, where we applied the branch-and-bound (B&B) algorithms for convex MINLP described in (Melo et al. 2014), revealed the difficulty in solving even small instances of the problem, and motivated us to develop a B&B algorithm specialized for the ESTP. The MINLP B&B algorithms in (Melo et al. 2014) were coded in the solver `Muriqui`, and the specialized B&B algorithm presented in this work is named `SAMBA` (Steiner Adaptations on Muriqui B&b Algorithm).

The main difficulty observed in the experiments with `Muriqui` comes from the large number of isomorphisms among the trees that span terminals and Steiner points, leading the B&B algorithm to solve equivalent or symmetric subproblems several times. We present in this work a procedure to avoid the consideration of isomorphic trees in the B&B enumeration scheme, based on the idea of representatives for classes of isomorphic Steiner topologies. The difficulty related to symmetry in the solution of integer-programming problems is well known (Margot 2009), and our approach is based on the idea of representatives for classes of isomorphic subproblems introduced in (Margot 2002).

In the remaining of the paper, we discuss how representative topologies are used to improve the B&B algorithm. We propose a procedure to fix variables in a preprocessing phase of the B&B and a pruning-by-isomorphism strategy. Still in preprocessing, we use geometric conditions for the SMT to reduce the number of representative topologies. Besides the procedures related to the concept of representative topologies, we present a more general procedure, called Dynamic Constraints Set (DCS); it eliminates nonlinear redundant constraints from subproblems and was very effective in our experiments. The DCS strategy can be extended to a variety of MINLPs having disjunctive constraints. We propose a heuristic procedure to improve the upper bound during the B&B execution and show how we deal with the non-differentiability of the Euclidean norm function at points where the solution degenerates. Finally, we present computational results demonstrating the effectiveness of `SAMBA`.

2 An MINLP formulation

In (Fampa and Maculan 2004), Fampa and Maculan formulate the ESTP as the MINLP problem

$$
\begin{array}{llcll}
\text{FM:} & \min & \sum_{[i,j]\in E} d_{ij}, & & (1)\\
 & \text{s.t.:} & d_{ij} \geq \|x^i - x^j\| - M(1-y_{ij}), & [i,j]\in E_1, & (2)\\
 & & d_{ij} \geq \|a^i - x^j\| - M(1-y_{ij}), & [i,j]\in E_2, & (3)\\
 & & \sum_{j\in S} y_{ij} = 1, & i\in P, & (4)\\
 & & \sum_{i\in P} y_{ij} + \sum_{k<j,k\in S} y_{kj} + \sum_{k>j,k\in S} y_{jk} = 3, & j\in S, & (5)\\
 & & \sum_{i<j,i\in S} y_{ij} = 1, & j\in S-\{p+1\}, & (6)\\
 & & y_{ij}\in\{0,1\}, d_{ij}\in\mathbb{R}, & [i,j]\in E_1\cup E_2, & (7)\\
 & & x^i\in\mathbb{R}^n, & i\in S, & (8)
\end{array}
$$

where $P=\{1,\dots,p\}$ and $S=\{p+1,\dots,2p-2\}$ are indices for terminals $a^1,\dots,a^p$ and Steiner points $x^{p+1},\dots,x^{2p-2}$. $[i,j]$ is an edge of the graph $G=(P\cup S, E_1\cup E_2)$, with $E_1=\{[i,j]|i,j\in S, i<j\}$ and $E_2=\{[i,j]|i\in P, j\in S\}$.

Variable y_{ij} indicates if $[i,j]$ is in the SMT. $\|\cdot\|$ is the Euclidean norm. Variable d_{ij} is the length of the edge $[i,j]$, and constraints (2–3) ensure that the length is only considered if the edge is in the tree. The Big-M parameter is an upper bound on the distance between two nodes in a SMT; $M=\max_{i,j\in P}\{\|a^i-a^j\|\}$. Constraints (4–6) model a FST, constraints (4–5) enforce degree 1 for terminals and degree 3 for Steiner points, and constraint (6) prevents subcycles.

3 Representative topologies and the B&B algorithm

Due to the large number of isomorphic FSTs, the B&B algorithm, when applied to formulation FM of the ESTP, generally solves equivalent subproblems many times and, because of this, the solution of even moderate-sized instances may become very challenging. In Figure 1 we show two isomorphic FSTs for $p=4$ that belong to the feasible set of FM, where nodes 5 and 6 are Steiner points and the other nodes are terminals.

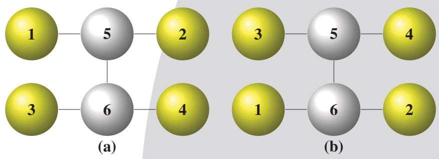

Figure 1: Two isomorphic FSTs for $p=4$

We implemented a procedure for generating non-isomorphic representatives for all FSTs of FM. We code the non-isomorphic FSTs and save them in a lexicographic order in a set Ψ. We construct the FSTs one by one and save a FST if it is not isomorphic to any FST already saved. Checking if two FSTs are isomorphic is done in polynomial time (Kelly 1957, Köber et al. 1993), and although the number of non-isomorphic FSTs grows exponentially with the number of terminals p, the representatives do not depend on the instance data of the ESTP. Therefore, the procedure is independent of the B&B algorithm, and once we save the set for a given p, we may apply it to the solution of any instance of the same size. Furthermore, in the construction of representative FSTs with $p'>p$ terminals, we can make use of the representative FSTs with p terminals, in case it is not possible to save the representatives for all p' terminals. We next describe the main procedures added to the NLP B&B algorithm implemented in the MINLP solver `Muriqui` in order to specialize it to solve more efficiently the ESTP.

4 The preprocessing phase

Geometric conditions satisfied by a SMT may be used to eliminate topologies from the list of representative FSTs in a preprocessing phase of our B&B algorithm. Let η_i be the distance from terminal a^i to the nearest other terminal, i.e.,

$$\eta_i := \min_{j \in P,\, j \neq i} \{\|a^i - a^j\|\}, \text{ for all } i \in P.$$

Let T be a minimum-length spanning tree on terminals a^i, $i \in P$, and β_{ij} be the length on the longest edge on the unique path between a^i and a^j in T, for $i, j \in P$. Two conditions on SMTs based on these parameters are known (see (Van Laarhoven and Anstreicher 2013)):

1. Terminals a^i and a^j may be connected to a common Steiner point only if
$$\|a^i - a^j\| \leq \eta_i + \eta_j.$$
2. Terminals a^i and a^j may be connected by two or fewer Steiner points only if
$$\|a^i - a^j\| \leq \eta_i + \eta_j + \beta_{ij}.$$

Considering these two conditions, we eliminate from Ψ, all FSTs that do not satisfy them.

We also use the set Ψ to fix variables, still in the preprocessing phase of the B&B algorithm.

Let E^+ be the set of edges of G that belong to every topology in Ψ, and E^- be the set of edges that do not belong to any topology in Ψ. If $[i, j] \in E^+$, we set $y_{ij} = 1$, and if $[i, j] \in E^-$, we set $y_{ij} = 0$.

5 Branching

Our B&B algorithm is divided into two phases. In phase 1 (phase 2), the branching variable is selected among $y_{ij} \in (0, 1)$ in the solution of the subproblem continuous relaxation, such that $[i, j]$ belongs to E_1 (E_2).

In phase 1, the spanning tree connecting the Steiner points is built, and the selection of the branching variable follows the standard strategy regarding the distance between y_{ij} and an integer value. We choose the variable corresponding to the greatest value of $\min\{y_{ij}, 1 - y_{ij}\}$. In phase 2, terminals are connected to the partially constructed tree, and the branching variable y_{ij} corresponds to the farthest terminal a^j to the other terminals already connected to the tree, attempting to increase the lower bounds rapidly.

Once the branching variable y_{ij} is selected at a node of the B&B tree, the descendants of the node represent different possibilities for connecting the Steiner point x^j (in phase 1) or the terminal a^j (in phase 2) to the tree, dividing the feasible region of FM according to constraint (6) or (4), resp.

6 Pruning by isomorphism

At each node of the B&B tree, we use the set Ψ to prune all descendant nodes where the variables fixed at 1 do not correspond to edges in any representative FST. This is how we avoid solution of subproblems corresponding to isomorphic trees.

7 Dynamic Constraint Set (DCS)

Constraints (2-3) are a typical way of modeling the requirement of considering the constraints

$$d_{ij} \geq \|x^i - x^j\|$$

and

$$d_{ij} \geq \|a^i - x^j\|$$

only when the corresponding y_{ij} are equal to 1.

The Big-M parameter is chosen to make the constraints redundant when $y_{ij} = 0$.

Although the these redundant inequalities do not modify the solution of B&B subproblems, it does increase considerably the running time. Therefore, we have added to our B&B algorithm, a strategy named “Dynamic Constraint Set” (DCS) — the idea is to dynamically change the set of constraints of the subproblems, eliminating the redundant constraints. Constraints (2-3) corresponding to y_{ij} fixed at 0, are accordingly removed from the model.

8 A heuristic

We propose a heuristic procedure to improve the upper bound on the optimal value of the problem, which can be applied at any node of the B&B enumeration tree.

Let $\bar{y}_{ij}$ be the value of the binary variable y_{ij} at the optimal solution of the relaxation solved at a node of the B&B tree. If the solution does not satisfy the integrality constraints, we consider the graph $G = (V, E)$ defined in Section 2 and assign to each edge $[i, j]$ of E, the weight given by $w_{ij} := 1 - \bar{y}_{ij}$. Considering these weights, we then search for a minimum spanning tree of G having a FST.

For the computation of this tree, we propose a simple greedy heuristic that starts considering a tree with the first two Steiner points $p + 1$ and $p + 2$ and then iteratively connects the next Steiner point to a node already in the tree, such that the new edge added to the tree has minimum possible weight and the degree condition on a FST is not violated, i.e., no Steiner point receives more than three neighbors. When the Steiner points are all in the tree, the process continues with the terminal nodes, taking into account that terminal nodes should be connected to Steiner points and can be connected to any one of them.

Once the FST is obtained by the heuristic, we locate the Steiner points by solving the convex problem obtained by fixing all integer variables in (FM). The solution obtained is feasible for the ESTP, and therefore the upper bound on its optimal value is possibly updated. Furthermore, since this is a best-possible solution for the given topology, the following cut can be added to all subproblems still open in the B&B enumeration tree:

$$\sum_{[i,j]\in E} \hat{y}_{ij} y_{ij} \leq 2p - 4,$$

where $\hat{y}$ is the characteristic vector of the tree, i.e. $\hat{y}_{ij} = 1$ if edge $[i, j]$ belongs to the tree, and $\hat{y}_{ij} = 0$, otherwise.

9 Dealing with the non-differentiability

The continuous relaxation of FM, as well as the problems obtained when we fix the values of all binary variables y_{ij} in FM, are convex NLP problems that we can solve to optimality by general-purpose NLP solvers (e.g. `Ipopt`, `Mosek`). Most NLP solvers, however, require all functions in the problem to be twice continuously differentiable, which is not the case for the problems

Table 1: Average results

Dimension	Muriqui B&B		SAMBA	
	gap (%)	cpu time (sec)	gap (%)	cpu time (sec)
$n = 3$	84	14400.38	0	240.96
$n = 4$	84	14401.12	0	740.87
$n = 5$	82	14400.34	0	1301.54

mentioned above, due to the non-differentiability of the Euclidean norm at points where the solution degenerates. There are different ways that we can deal with the particular non-differentiability that we face. The approach used in this work was to approximate $\sqrt{w}$ by $h(w) := \sqrt{w+\delta} - \sqrt{\delta}$ for some small $\delta > 0$.

10 Computational results and conclusions

We compare the performance of the B&B algorithm of `Muriqui` with the specialized B&B algorithm implemented as `SAMBA` — both are implemented in `C++` and all runs were on a 3.60 GHz core i7-4790 CPU, 8 MB, 16 GB, running under Linux, with a time limit of 4 hrs. `MOSEK` (Andersen and Andersen 1999) was used to solve the subproblems relaxations in B&B. Instances considered were the same used in (Fampa and Anstreicher 2008): 30 instances with 10 terminals, 10 in each dimension $n = 3, 4, 5$. No instances could be solved by `Muriqui` within the time limit, with an average duality gap of 83%=100%·(upper bound - lower bound)/upper bound, endorsing the well-known difficulty of the ESTP. We also applied the MINLP solver `Bonmin` (Bonami et al. 2008), with similar results to those of `Muriqui`; no instance could be solved in the time limit. In contrast, we solved all 30 instances to optimality with `SAMBA`, as shown in Table 1.

The results indicate significant gains obtained by the procedures we propose, turning practical the application of B&B to the convex MINLP formulation of the ESTP.

Although the reduction on the number of representative FSTs, when compared to the number of FSTs in the feasible solution of FM, is very impressive, the number of representatives still grows extremely fast, and it would not be possible to save all representative FSTs for a significantly larger number of terminals. We note, however, that in case we are solving problems with p' terminals, and we have only the list of non-isomorphic FSTs with p terminals, where $p < p'$, it is trivial to construct the list of non-isomorphic spanning trees on the $p' - 2$ Steiner points, taking as input the list of representatives for the spanning trees on the $p - 2$ Steiner points already constructed. This pre-saved list may still be used to initiate the B&B algorithm, and then, during the execution of the B&B, when the terminals are added to the Steiner trees, we can construct the list of representative FSTs as they are iteratively generated by the algorithm, as suggested in (Margot 2002). This is subject of our future research.

Finally, we note that our DCS strategy decreased the running time of B&B by 57% on average. DCS was very effective and can be applied to other MINLPs having disjunctive constraints modeled with Big-M parameters.

References

E. Andersen and K. Andersen. The mosek interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High Performance Optimization*, volume 33 of *Appl. Opt.* Kluwer, 1999.

Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, Gérard Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, François Margot, Nicolas Sawaya, and Andreas Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optim.*, 5(2): 186–204, 2008.

Marcia Fampa and Kurt M. Anstreicher. An improved algorithm for computing Steiner minimal trees in Euclidean d-space. *Discrete Optimization*, 5(2):530–540, 2008.

Marcia Fampa and Nelson Maculan. Using a conic formulation for finding Steiner minimal trees. *Numerical Algorithms*, 35(2-4):315–330, 2004.

E.N. Gilbert and H.O. Pollak. Steiner minimal trees. *SIAM J. Applied Math.*, 16:1–29, 1968.

F.K. Hwang, D.S. Richards, and W. Winter. *The Steiner tree problem*, volume 53, Annals of Discrete Math. Elsevier, Amsterdam, 1992.

P.J. Kelly. A congruence theorem for trees. *Pacific J. Math.*, 7:961–968, 1957.

J. Köber, U. Schöning, and J. Torán. *The graph isomorphism problem: its structural complexity*. Birkhauser Verlag Basel, Switzerland, 1993.

N. Maculan, P. Michelon, and A.E. Xavier. The Euclidean Steiner tree problem in R^n: A mathematical programming formulation. *Annals of Operations Research*, 96:209–220, 2000.

F. Margot. Pruning by isomorphism in branch-and-cut. *Math. Prog.*, 94:71–90, 2002.

François Margot. Symmetry in integer linear programming. In M. Jünger, Th.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, Reinelt, G. G., Rinaldi, and L.A. Wolsey, editors, *50 Years of Integer Programming*. Springer, 2009.

Wendel Melo, Marcia Fampa, and Fernanda Raupp. Integrating nonlinear branch-and-bound and outer approximation for convex mixed integer nonlinear programming. *Journal of Global Optimization*, 60:373–389, 2014.

Warren D. Smith. How to find Steiner minimal trees in Euclidean d-space. *Algorithmica*, 7(1-6): 137–177, 1992.

Jon W. Van Laarhoven and Kurt M. Anstreicher. Geometric conditions for Euclidean Steiner trees in R^d. *Comput. Geom.*, 46(5):520–531, 2013.