# Building Steiner Trees on Supercomputers

## Massively Parallel Solution of Hard Steiner Tree Problems

***Thorsten Koch**, Zuse Institute Berlin*

### In Short

- Finding optimal Steiner trees is a famous problem in mathematical optimization and computer science.
- Steiner trees can also be found in many real-world applications, ranging from VLSI design to computational biology.
- We develop a massively parallel solver to efficiently solve especially hard Steiner tree problem instances on supercomputes.

"A very simple but instructive problem was treated by Jacob Steiner, the famous representative of geometry at the University of Berlin in the early nineteenth century. Three villages A,B,C are to be joined by a system of roads of minimum length." Such reads an early description of the *Steiner tree problem in graphs* in the classic "What is mathematics?" by Courant and Robbins from 1941. Indeed, the roots of this problem are much older, and can be traced back to Pierre de Fermat's famous treatise *Methodus ad disquirendam maximam et minimam*. The problem was later rediscovered by the likes of Carl Friedrich Gauß and Vojtěch Jarník. In the modern, general version of the Steiner tree problem, we are given an arbitrary number of cities, and the length of the roads can be any non-negative number. Mathematically, the problem is defined as follows. Given an undirected graph with non-negative edge weights and a subset of vertices called *terminals*, the Steiner tree problem in graphs (SPG) is to find a tree of minimum weight that contains all terminals. This seemingly simple problem has proven to be notoriously hard to solve, both in theory and in practice. Moreover, it is one of the most studied problems in combinatorial optimization. This large interest is also motivated by the various real-world applications that can be modeled by using Steiner trees.

For a theoretical mathematician the solution of the Steiner tree problem might seem astonishingly trivial: just check each of the finite number of Steiner trees and select one of minimum length. For practical purposes, however, such an approach requires a good deal of patience. Even for small problems with a few thousand vertices, this plain enumeration might take billions of years even on the fastest supercomputers that are currently available. However, years of mathematical research have led to more targeted and more powerful algorithms that allow the solution of many real-world SPG instances in a matter of seconds on a regular desktop computer, in defiance of the exponential worst-case complexity.

While there are practical applications that lead to a pure SPG formulation, it is far more common to encounter applications that require slight variations. As an example, consider a telecommunication network problem, where one needs to interconnect customers. For a company it may not pay off to connect every possible customer. So instead of aiming to include all customers, one might associate a non-negative prize with each customer and search the corresponding graph for a tree that maximizes the sum of its prizes minus its length. Indeed, this so-called prize-collecting Steiner tree problem has been used for planning fiber-optic networks in German cities. The various fields where one encounters problems that are closely related to the SPG range from computational biology and cancer research to computer chip design, computer vision, and even the deployment of drones.

One of the fastest Steiner tree solvers available worldwide today is the SCIP-Jack software, which can handle not only classic SPG, but also 12 related problems. In order to handle the large variety and complexity of Steiner tree problems, SCIP-Jack incorporates a diverse set of algorithmic components, which closely interact. The overall solution algorithm follows the branch-and-bound paradigm, an advanced divide- and-conquer approach that explores the space of solutions by dividing it into increasingly smaller regions. Over- and underestimates of the objective function value are used to identify and exclude entire regions that do not contain optimal solutions as early as possible. Although the branch-and-bound paradigm is guaranteed to converge to a global optimal solution in finite time, this approach alone suffers from the same exponential worst-case behavior as plain enumeration. Hence, SCIP-Jack implements many supplementary techniques including graph transformations to a beneficial standard form, heuristics for finding primal solutions and dual bounds on the best possible objective value, domain propagation, cutting planes, and branching rules. Over the years, it has also become the fastest solver worldwide for most of the thirteen problem classes it can handle. Just recently, the strong performance of SCIP-Jack was, for instance, demonstrated at the international PACE Challenge 2018 3. This competition evaluated solvers for a subclass of the SPG, so-called fixed-parameter tractable SPGs, for which

more efficient algorithms than for the general SPG are known. Even though SCIP-Jack does not include specialized algorithms for this problem class, it was the best overall solver at the challenge. The current (development) version of SCIP-Jack encompasses almost 100 000 lines of code. SCIP-Jack is freely available for academic research as part of the SCIP Optimization Suite 1. SCIP-Jack has already be licensed for commercial use, but has also been used in several academic research projects.

Most of the recent advances in solving Steiner tree problems have focused on improving the performance of sequentially executed methods. However, in the age of increasingly available data sources, the growing complexity and dimension of real-world instances pose new challenges to sequential solvers. Even after executing today's sophisticated preprocessing techniques, the reduced problems may require an extremely expensive branch- and-bound search to compute a provably optimal Steiner tree solution. Motivated by this bottleneck (which also occurs in many other hard optimization problem), researchers at ZIB 4 have developed one of the world's leading software frameworks that accelerates branch-and-bound searches by distributing the search effort using massively parallel hardware: UG 5. This framework has been combined with SCIP-Jack to allow the parallelization of its branch-and-bound search. Furthermore, several (both technical and mathematical) extensions have been made to accelerate this combined solver. In this way, several Steiner tree problem instances have already been solved to proven optimality. The largest configuration used encompassed up to 43 000 cores 6.

Certainly, the merely linear increase in computing power provided by supercomputers is only a limited remedy against the exponential worst-case complexity of combinatorial optimization problems. Nevertheless, the capability to momentarily boost the performance of algorithms by large- scale parallelization, combined with the highly sophisticated algorithms integrated in SCIP-Jack can lead to the solution of otherwise intractable problems.

In this project, we further improve the interplay of SCIP-Jack and UG. Additionally, we plan to add an internal shared-memory parallelization to SCIP-Jack in order to improve its scalability. So far, we have been able to solve five well-known Steiner tree problem instances to optimality by using supercomputers. Prior, these problems had remained unsolved for more than 15 years. We expect to be able to solve further of such notoriously hard problems within this project

## WWW

https://www.zib.de/members/koch

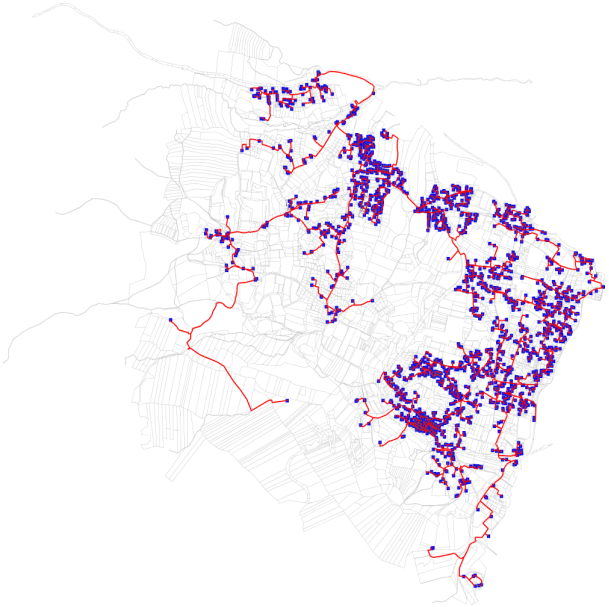

***Figure 1:*** *A real-world telecommunication network from a city in Austria with an optimal Steiner tree as a solution marked in red 2.*

## More Information


[1] SCIP Optimization Suite (including SCIP-Jack), https://www.scipopt.org/.

[2] Steiner tree problems in the real-world, https://homepage.univie.ac.at/ivana.ljubic/research/STP/.

[3] *PACE Challenge*, https://pacechallenge.org/.

[4] Zuse Institute Berlin, https://zib.de.

[5] *UG: Ubiquity Generator framework*, http://ug.zib.de/.

[6] Y. Shinano, D. Rehfeldt, and T. Koch, *Building Optimal Steiner Trees on Supercomputers by Using up to 43,000 Cores*, in Integration of Constraint Programming, Artificial Intelligence, and Operations Research. CPAIOR (2019).

[7] D. Rehfeldt, and T. Koch, *Combining NP-Hard Reduction Techniques and Strong Heuristics in an Exact Algorithm for the Maximum-Weight Connected Subgraph Problem*, in SIAM Journal on Optimization (2019).

[8] D. Rehfeldt, T. Koch, and Y. Shinano, *SCIP-Jack: An exact high performance solver for Steiner tree problems in graphs and related problems*, in Modeling, Simulation and Optimization of Complex Processes (accepted for publication).


## Funding


Forschungscampus Modal
MATH+