Combinatorial Optimization Contest 2019:
Steiner tree in disguise

Combinatorial Optimization Team

http://industrialinformatics.fel.cvut.cz/

May 21, 2019

Steiner tree problem

- Classical $\mathcal{NP}$-hard problem
- Benchmark instances selected from `http://steinlib.zib.de/steinlib.php`
  - For many of them, optimality of the best-so-far solution has not been proved yet

Instances

- Diversity:
  - Instances with unit costs
  - Instances with non-unit costs
  - Polynomially solvable instances
  - Difficult instances
  - Special instances (grid graphs with rectangular holes)

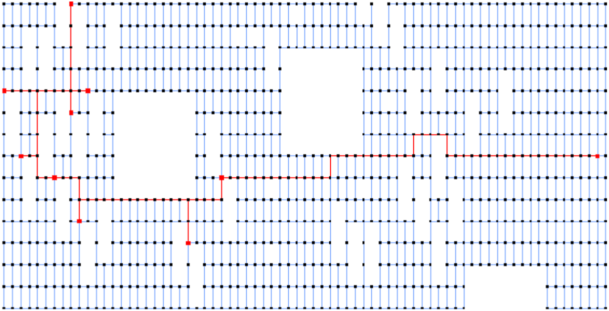


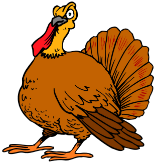
Figure: Example of a special instance of the Steiner tree problem

Reference solution

Several simple algorithms implemented to set some reasonable thresholds. One example:

1. Preprocess the graph (prune non-terminals of degree 1)
2. Find a minimum spanning tree
3. While time remains, greedily improve the current solution:
    - Try to remove some non-terminal vertex from the current spanning tree and find a new minimum spanning tree using the original graph without the removed vertex

Other strategies can be used too, e.g., splitting the tree (removing some non-terminal vertices) and trying to connect it again *somehow*.



State-of-the-art

Instance	Complexity	State-of-the-art	Gramopav	Gap [%]
bipa2p	??	35326	36147	2.3
cc3-12u	??	185	187	1.1
cc7-3u	??	549	573	4.3
cc9-2p	??	17199	17465	1.5
hc9p	NP?	30242	30475	0.8
hc12u	??	2262	2352	4.0
msm0709	LP	**884**	884	0.0
msm2152	PS	**1590**	1590	0.0
msm4312	PS	**2016**	2016	0.0
w23c23	NP?	**689**	699	1.4

Table: Comparison of the state-of-the art solutions and student's solutions

Note: LP (solvable by local preprocessing), PS (solvable by polynomial algorithms), NP (no polynomial algorithm is currently known), ?? (currently not known).