Semester projects

Thomas Rothvoß

DisOpt, EPFL

thomas.rothvoss@epfl.ch

February 9, 2009

Contents

Chapter 1

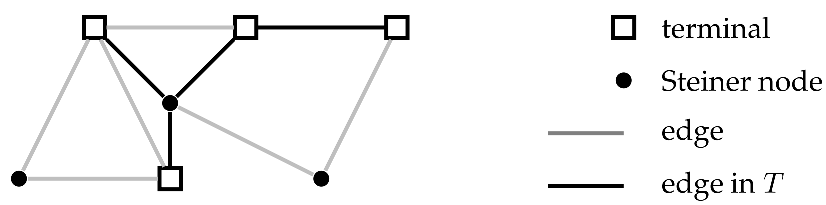
Integrality gap of the bidirected Steiner Tree Relaxation

Introduction

A very classical **NP**-hard optimization problem is the *Steiner Tree* problem, already introduced by Gauss in a letter to Schumacher. A formal definition is as follows

> STEINER TREE
> Given: An undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{Q}^+$ and distinguished *terminals* $R \subseteq V$.
> Find: A tree $T \subseteq E$, connecting all terminals, while minimizing the cost $\sum_{e \in T} c(e)$.

An example looks as follows:



For $\alpha \geq 1$, an α-approximation algorithm is an algorithm, which always computes (in time polynomial in the input size) a solution, that is of cost at most α times the cost of the best solution. Steiner tree is a generalization of the so called *Spanning Tree* problem, in which all nodes have to be connected, thus $R = V$. The difference for Steiner Tree is that here one is allowed to include vertices $V \backslash R$ into the tree to make the tree cheaper. The nodes $V \backslash R$ are called *Steiner nodes*. While the Spanning Tree problem can be solved optimally even with a simple greedy algorithm, Steiner Tree can be shown to be **NP**-hard [BP89]. In fact it is even **NP**-hard to compute a solution which is at most 1% more costly than the optimum tree [CC02]. On the other hand the best known approximation algorithm gives a 1.55-approximation [RZ00].

Steiner Tree is a special case of a bunch of other combinatorial optimization problems [KM00, GKR03, EGOS05, SK04, GKK$^+$01, EG05, Vaz01].

The undirected LP-relaxation

But how to derive an approximation algorithm with a small approximation ratio? One standard approach is to consider an *integer linear program* for the problem. Therefore let us introduce decision variables

$$x_e = \begin{cases} 1 & \text{if } e \text{ is in the Steiner tree} \\ 0 & \text{otherwise} \end{cases}.$$

and consider

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \qquad (IP) \\ \sum_{e \in \delta(S)} x_e \geq 1 & \qquad \forall S \subseteq V \text{ with } S \cap R \neq \emptyset \text{ and } (V \backslash S) \cap R \neq \emptyset \\ x_e \geq 0 & \qquad \forall e \in E \\ x_e \in \mathbb{Z} & \qquad \forall \in E \end{aligned}$$

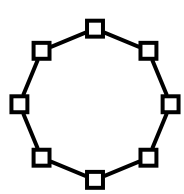
Thereby $\delta(S) = \{\{u, v\} \in E \mid u \in S, v \notin S\}$ is the *cut* induced by S. The inequality $\sum_{e \in \delta(S)} x_e \geq 1$ guarantees that all terminals must be connected to each other. Unfortunately such integer programs cannot be solved in polynomial time. But what can be solved (for example using *ellipsoid method* [GLS81]) is the *linear programming relaxation*

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \qquad (LP) \\ \sum_{e \in \delta(S)} x_e \geq 1 & \qquad \forall S \subseteq V \text{ with } S \cap R \neq \emptyset \text{ and } (V \backslash S) \cap R \neq \emptyset \\ x_e \geq 0 & \qquad \forall e \in E \end{aligned}$$

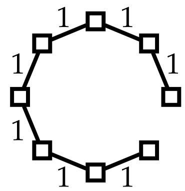
emerging from (IP) by omitting the integrality constraint.

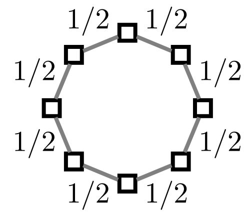
Let OPT be the optimum value of (IP) and OPT_f the optimum value for (LP). Clearly $OPT_f \leq OPT$. The maximum value of $\frac{OPT}{OPT_f}$ over all instances is called the *integrality gap* of this relaxation.

We claim that the integrality gap of (LP) is at least 2. Therefor consider the following circle graph, with $R = V$, n nodes and unit cost on the edges



Then an integer solution necessarily contains $n-1$ edges, thus $OPT = n-1$ (left picture below). But for a fractional solution it suffices to install $x_e = 1/2$ units of capacity on each edge (right picture below).





Thus $OPT_f = n/2$. Consequently the integrality gap is at least 2 (for $n \to \infty$).

Typically an LP relaxation of $\alpha \geq 1$ admits an α-approximation algorithm for the corresponding problem. For the above LP (and even more general problems), one can always derive an integer solution of cost $2 \cdot OPT_f$, thus the integrality gap for (LP) is in fact 2 [GW95].

The bi-directed relaxation

To derive better approximation algorithms it lies on hand to think about stronger relaxations - with smaller integrality gap.

One well known approach is the so called *bi-directed cut relaxation*. Therefor choose an arbitrary terminal $r \in R$ as root. Furthermore replace each undirected edge $u, v \in E$ by 2 directed edges (u,v) and (v,u) (with cost $c(u,v) = c(v,u) = c(\{u,v\})$). We again use x_e as decision variable, whereby e now is a directed edge. Denote $\delta^{\text{out}}(S) = \{(u,v) \in E \mid u \in S,\ v \notin S\}$ as the edges, leaving the set $S \subseteq V$. Then

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \qquad (IP') \\ \sum_{e \in \delta^{\text{out}}(S)} x_e \geq 1 & \qquad \forall S \subseteq V : S \cap R \neq \emptyset \text{ and } r \notin S \\ x_e \geq 0 & \qquad \forall e \in E \\ x_e \in \mathbb{Z} & \qquad \forall e \in E \end{aligned}$$

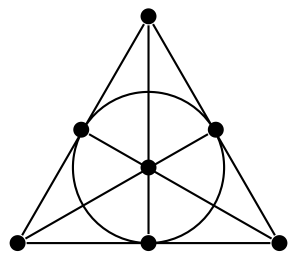
is again an integer linear program, solving Steiner Tree. It can be considered as follows: Install enough capacity, s.t. one can send one unit of flow (unsplitted) from each terminal to the root.

The outcome of (IP') will be exactly the same as (IP). But the linear programming relaxation

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \qquad (LP') \\ \sum_{e \in \delta^{out}(S)} x_e \geq 1 & \qquad \forall S \subseteq V : S \cap R \neq \emptyset \text{ and } r \notin S \\ x_e \geq 0 & \qquad \forall e \in E \end{aligned}$$

is different from the undirected case (LP). For example, in case of the circle graph above one has $OPT = OPT_f$. This always holds if $R = V$ [Vaz01]. Note that the choice of the root does not affect OPT_f. Unfortunately also this relaxation is not "perfect", i.e. in general $OPT_f < OPT$.

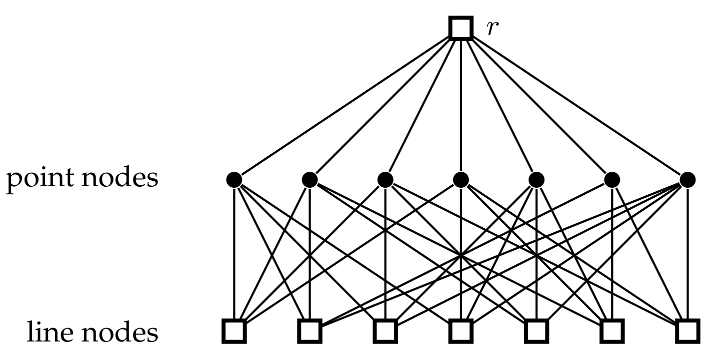
We want to give a graph, for which $\frac{OPT}{OPT_f} = \frac{8}{7}$. Therefor consider the so called *Fano plane*, containing 7 points and 7 "lines" (we consider the circle also as a line, since the Fano plane cannot be embedded into $\mathbb{R}^2$ with real lines).



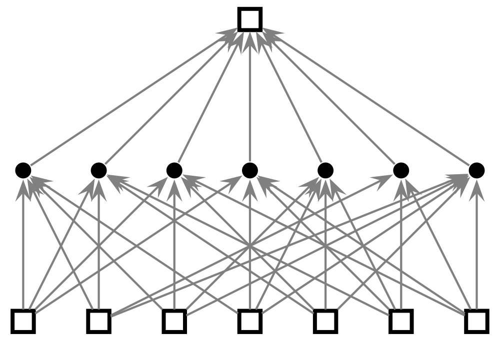
It has the following nice properties:

- Each point lies on 3 lines
- Each line contains 3 point
- 2 points lie on exactly one common line
- 2 lines share exactly one common point

From this we construct a graph with a root terminal r, 7 "point" nodes and 7 "line" terminals. Each point node is connected to r with an unit cost edge. Let i be a point node and j be a line node. Then edge (i, j) shall exist if and only if point i does not lie on line j (in the Fano plane).



From the properties of the Fano plane we see that each node (except of the root) has degree 4. If we install capacities of $1/4$ on each edge (directed upwards), then clearly the solution is feasible

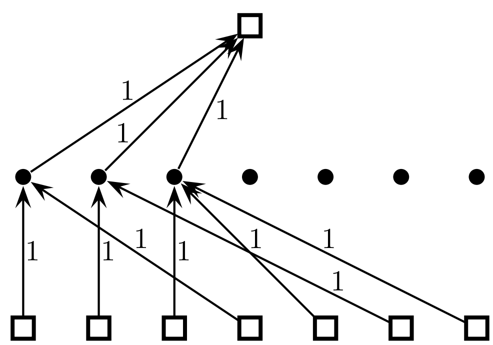


$\forall e \in E : x_e = \frac{1}{4}$

since each valid cut $\delta^{\text{out}}(S)$ contains at least 4 edges. The cost of this solution is

$$OPT_f = \frac{1}{4} \cdot (7 + 7 \cdot 4) = \frac{35}{4}$$

Now let us think, how an optimum integer solution may look like. Clearly we need to buy 7 edges to connect the terminal nodes on the lower level to the Steiner nodes. Furthermore each Steiner node that we need has to be paid with 1. Thus the question is, how many Steiner nodes do we need to include in the tree. Suppose for contradiction that 2 "point" nodes i, i' are sufficient. But for each pair of points there is a line j in the Fano plane, containing both points, i and i'. Consequently this node j is not connected to i or i' (in the graph). We conclude that we need at least 3 Steiner nodes (which is sufficient), thus $OPT = 7 + 3 = 10$. The optimum solution looks as follows



Consequently the integrality gap of this instance is $\frac{10}{35/4} = \frac{8}{7}$. However, this is the best known lower bound on the integrality gap of the bi-directed cut relaxation. On the other hand the best known upper bound is still 2, as it is for the undirected relaxation. Only in case of *quasi-bipartite graphs* (Steiner nodes are non adjacent) the upper bound can be improved to $3/2$ [RV99].

The task

The example graph above was just recently discovered by Skutella (see also [KPT07]), while before only an infinite family of graphs was known for which the integrality converges to $8/7$ (see [Vaz01]). This motivates examination of the worst case integrality gap even for small graphs. Define

$$I(n) = \max\left\{ \frac{OPT(G,c,R)}{OPT_f(G,c,R)} \mid (G,c,R) \text{ Steiner tree instance with } G = (V,E),\ |V| = n \right\}$$

be the worst case integrality gap for graphs with n nodes. Here $OPT(G, c, R)$ denotes the cost of an optimum Steiner tree in graph G with cost function c and terminals R; similar for $OPT_f(G, c, R)$.

It is task of the project, to determine $I(n)$ exactly for small n (say $n = 5, 6, 7, 8, \ldots ?$). This might give hints about the "real" integrality gap and could help to clarify, whether $I(n) \leq 8/7$ for all n as some researchers conjecture.

Clearly enumerating all graphs, sets $R \subseteq V$ and rational vectors $c \in \mathbb{Q}^{|E|}$ is not doable. Therefore observe that given a graph G, we may add not-existing edges e to make G complete. If we choose $c(e)$ high enough, the optimum solutions won't change. Thus we only need to consider a complete graph. Let n be fixed and define

$$P = \left\{ x \in \mathbb{Q}^{|E|} \mid \sum_{e \in \delta^{\text{out}}(S)} x_e \geq 1 \ \forall S \subseteq V : S \cap R \neq \emptyset, r \notin S; \ 0 \leq x_e \leq 1 \right\}$$

as the polytope of all solutions. Then we can rewrite the integrality gap as

$$I(n) = \max_{c \in \mathbb{Q}_+^{|E|}} \frac{\min\{c^T x \mid x \in P \cap \mathbb{Z}^{|E|}\}}{\min\{c^T x \mid x \in P\}}$$

Let $P_I = \text{conv}(P \cap \mathbb{Z}^{|E|})$ be the convex hull of the integer solutions. We know that for each c, both minimums are attained at a vertex of P_I and P, respectively. Suppose we are given these vertices $x^I \in P_I$ and $x^f \in P$, then we had

$$I(n) = \max \left\{ \frac{c^T x_I}{c^T x_F} \mid c \in \mathbb{Q}_+^{|E|}; \ x_I, x_F \text{ are optimal w.r.t } c \right\}$$

Since we are maximizing this expression we do not have to enforce optimality of x_F; we just have to ensure that x_I is an optimum integer solution. Furthermore the optimum is invariant under scaling c, thus we may add the constraint $c^T x_F = 1$. Consequently Algorithm 1 determines $I(n)$.

Clearly this approach is not very efficient. Thus one should think about possible improvements. Do we really have to consider all pairs of vertices, or can we exclude some in advance?

Summary

The tasks of the student are

- Implement above algorithm using the OR software/library of your choice (and a programming language of your choice). Recommended are C/C++ or Java. For example CPLEX, Qhull, cddlib might be useful.
- Compute $I(n)$ for as many n's as possible.

In case of interest, just talk to Thomas Rothvoß, MA C1 557 (`thomas.rothvoss@epfl.ch`).

Algorithm 1 Algorithm for computing $I(n)$

INPUT: n

OUTPUT: $I(n)$

1. $I := 0, V = \{1, \ldots, n\}$
2. FOR ALL $k := 1$ TO n DO
   (a) $s := 1, R := \{1, \ldots, k\}$
   (b) Enumerate vertices V_F of
   $$P = \{x \in \mathbb{Q}^{|E|} \mid \sum_{e \in \delta^{\text{out}}(S)} x_e \geq 1 \; \forall S \subseteq V : S \cap R \neq \emptyset, r \notin S;\ 0 \leq x_e \leq 1\}$$
   (c) Enumerate solutions $V_I = P \cap \mathbb{Z}^{|E|}$
   (d) FOR EACH $x_I \in V_I$ DO
      i. FOR EACH $x_F \in V_F$ DO
         A. Solve the following LP (denote its optimum value by (LP))
         $$\begin{array}{rcl} \max c^T x_I & & \\ c^T x_I & \leq & c^T x \quad \forall x \in V_I \\ c^T x_F & = & 1 \\ c & \geq & 0 \end{array}$$
         B. IF $(LP) > I$ THEN $I := (LP)$
3. RETURN I

Chapter 2

Connected layer families

Suppose we have symbols $\{1,\dots,n\}$. A set of d symbols is called a *vertex*. A set of vertices is called a layer $\mathcal{L}_i \subseteq \binom{1,..,n}{d}$. We call a set of layers $\mathcal{L}_1,\dots,\mathcal{L}_\ell$ a *connected layer family* of height ℓ if the following conditions hold.

1. *Disjointness*: $\forall i \neq j : \mathcal{L}_i \cap \mathcal{L}_j = \emptyset$.

2. *Connectivity*: $\forall 1 \leq i < j < k \leq \ell$ and $u \in \mathcal{L}_i, v \in \mathcal{L}_k$ there is a $w \in \mathcal{L}_j$ such that $u \cap v \subseteq w$.

The parameter d is termed the *dimension* of the layer family.

$$
\begin{aligned}
\mathcal{L}_1 &= \{\{1,6\}\} \\
\mathcal{L}_2 &= \{\{1,2\},\{2,6\}\} \\
\mathcal{L}_3 &= \{\{2,5\},\{1,3\},\{\mathbf{4},6\}\} \\
\mathcal{L}_4 &= \{\{2,\mathbf{4}\},\{1,5\},\{3,6\}\} \\
\mathcal{L}_5 &= \{\{2,3\},\{1,\mathbf{4}\},\{5,6\}\} \\
\mathcal{L}_6 &= \{\{\mathbf{4},5\},\{3,\mathbf{4}\}\} \\
\mathcal{L}_7 &= \{\{3,5\}\}
\end{aligned}
$$

Above is an example of a 2-dimensional connected layer family with $n = 6$ symbols and 7 layers. A symbol s is *active* on a layer $_i$ if there exists a vertex of $\mathcal{L}_i$ containing s. In our example, we highlight the symbol 4 and, due to condition 2) the layers on which 4 is active are consecutive. This holds for each symbol and thus the example above is a 2-dimensional connected layer family.

The importance of this combinatorial structure is that $D(d,n)$ is an upper bound on the diameter of any polyhedron $P = \{x \in \mathbb{R}^d \mid a_1 x \leq b_1, \dots, a_n x \leq b_n\}$ in d dimensions, described by n inequalities. For more than 60 years it is an outstanding open problem, whether the diameter of a polyhedron is always bounded y a polynomial in d and n. For our abstraction one can show upper bounds of $D(d,n) \leq n^{1+\log d}$ and $D(d,n) \leq 2^{d-1} \cdot n$, while we found the lower bound $D(d,n) \geq \Omega(d \cdot n)$ (for $d \leq \sqrt{n/6}$).

Since there is a huge gap between lower and upper bounds we are strongly interested in determining $D(d,n)$ exact at least for small values of d and n.

Objectives

1. Read and understand the paper [EHR09].
2. For parameters $d, n, \ell \in \mathbb{N}$ come up with a SAT formula which is satisfiable if and only if there is a connected layer family of height ℓ, dimension d with n symbols.
3. Write a program (in a programming language of your choice), that outputs this formula in a file, that is readable for `zchaff`, the most powerful available SAT solver.
4. Find symmetry braking constraints, which help to solve larger instances like: "W.l.o.g. the vertex $\{1, \ldots, d\}$ must lie on the first level"
5. Find exact values of $D(d, n)$ for as many d, n as possible.

Bibliography

[BP89] M. Bern and P. Plassmann. The steiner problem with edge lengths 1 and 2,. *Inf. Process. Lett.*, 32(4):171–176, 1989.

[CC02] Chlebik and Chlebikova. Approximation hardness of the steiner tree problem on graphs. In *SWAT: Scandinavian Workshop on Algorithm Theory*, 2002.

[EG05] F. Eisenbrand and F. Grandoni. An improved approximation algorithm for virtual private network design. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA 05)*, volume 16, pages 928–932, Vancouver, Canada, 2005. SIAM.

[EGOS05] F. Eisenbrand, F. Grandoni, G. Oriolo, and M. Skutella. New approaches for virtual private network designs. In *Annual International Colloquium on Automata, Languages and Programming*, volume 3580 of *Lecture Notes in Computer Science*, pages 1151–1162, Lisboa, Portugal, 2005. Springer.

[EHR09] F. Eisenbrand, N. Haehnle, and T. Rothvoss. Diameter of polyhedra: Limits of abstraction. *Submitted to SoCG09*, 2009.

[GKK+01] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 389–398, New York, USA, 2001. ACM Press.

[GKR03] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *ACM Symposium on Theory of Computing (STOC)*, 2003.

[GLS81] M. Grötschel, L. Lovasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[GW95] Goemans and Williamson. A general approximation technique for constrained forest problems. *SICOMP: SIAM Journal on Computing*, 24, 1995.

[KM00] D. R. Karger and M. Minkoff. Building steiner trees with incomplete global knowledge. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, page 613, Washington, USA, 2000. IEEE Computer Society.

[KPT07] Jochen Könemann, David Pritchard, and Kunlun Tan. A partition-based relaxation for steiner trees. *CoRR*, abs/0712.3568, 2007. informal publication.

[RV99] Rajagopalan and Vazirani. On the bidirected cut relaxation for the metric steiner tree problem. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1999.

[RZ00] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. *Proc. of ACM/SIAM Symposium on Discrete Algorithms (SODA)*, 2000.

[SK04] C. Swamy and A. Kumar. Prima-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

[Vaz01] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.