k-Truss Minimization by Bilevel Optimization

Arie M.C.A. Koster[1], Jonas Kreyer[2], Ivana Ljubić[3], and Matthis Penz[2]

[1]Discrete Optimization, RWTH Aachen University, Aachen, Germany, ✉ koster@math2.rwth-aachen.de
[2]RWTH Aachen University, Aachen, Germany, ✉ jonas.kreyer@rwth-aachen.de, matthis.penz@rwth-aachen.de
[3]ESSEC Business School, Paris, France, ✉ ljubic@essec.edu

Graphs can be used to model connections between actors in all kinds of (social) networks. Dense subgraphs are often of interest and hence their identification has been studied for many decades. Like the concepts of k-plexes, k-clubs and k-cores, the concept of k-trusses generalizes cliques in a graph and can be exploited to identify dense (but not completely dense) subgraphs. Moreover, it is worth to investigate which edges are crticial for the stability of the structure. Stated otherwise, if you are allowed to destroy b edges, which ones would you select in order to affect the structure the most? In this talk, we study this general problem in the context of k-trusses and focus on exact solution approaches.

A clique is a subset of vertices, where every pair of vertices is adjacent. Stated otherwise, if the clique has k vertices, every edge in the induced subgraph is contained in $k-2$ *triangles*, induced K_3s. Now, a k-truss is an induced subgraph, where every edge is contained in $k-2$ triangles [4]. A maximum k-truss in a graph $G=(V,E)$ can be computed by iteratively removing isolated vertices and all edges that are not contained in $k-2$ triangles until all edges satisfy this property. Hence, the maximum k-truss is unique and can be solved in polynomial time [4].

Given a budget $b \in \mathbb{N}$, the problem of k-Truss Minimization [2] asks for which b edges to remove such that the maximum k-truss in the remaining graph is minimized (regarding the number of edges). We can assume that the input graph is a k-truss as all other edges vanish without actively removing any edge. As an earlier proof (presented in [2]) turns out to be incorrect, we first show that this problem is NP-complete for all $k \geq 3$.

To solve k-Truss Minimization, we present a bilevel optimization model, inspired by [1] for the collapsed k-core problem. At the leader level, we decide which b edges are removed from the graph. At the follower level, the maximum k-truss in the remaining graph is determined. The objective of the leader is to select the edges in such a way that the maximum k-truss in the end is as small as possible.

To model the follower problem, we first model the maximum k-truss problem as an integer linear program. Let x_{ij} be binary variables denoting whether edge $ij \in E$ is part of the maximum k-truss. Moreover, let $\Delta(G) = \{(i,j,\ell) \in V : G[\{i,j,\ell\}] = K_3\}$ be the set of triangles in G and binary variables $c_{ij l} \in \{0,1\}$ denoting whether triangle $(i,j,\ell) \in \Delta(G)$ is part of the maximum k-truss. Then, the maximum k-truss problem reads

$$\max \sum_{ij \in E} x_{ij} \tag{1}$$

$$s.t.\ (k-2)x_{ij} \leq \sum_{(i,j,\ell) \in \Delta(G)} c_{ij\ell} \qquad \forall ij \in E \tag{2}$$

$$c_{ij\ell} \leq x_{ij}, x_{i\ell}, x_{j\ell} \qquad \forall (i,j,\ell) \in \Delta(G) \tag{3}$$

$$x_{ij} \in \{0,1\}, c_{i,j,\ell} \in \{0,1\} \tag{4}$$

Here, constraints (2) model that an edge can only be part of a k-truss if at least $k-2$ triangles exist and constraints (3) require that all three edges exist for a triangle to remain in the graph.

Although the constraint matrix is obviously not totally unimodular, we show that the LP relaxation always returns an integral optimal solution (this in contrast to the integer program for k-core [1]). By this, we can apply dualization of the follower problem (treating the leader variables as constants) to obtain a single level reformulation. In this reformulation, the dual variables of the follower problem are multiplied with decision variables $w_{ij} \in \{0,1\}$ (denoting whether an edge selected for removal) of the leader, resulting in a quadratic optimization problem.

Alternatively, a single level reformulation can be obtained by following the same approach as considered for the collapsed k-core problem in [1]: Consider a subset of edges $T \subseteq E$ that induces a k-truss in G. If none of these edges are removed, the remaining k-truss has size at least $|T|$, hence with variable $z \in \mathbb{Z}$ representing the objective of the leader, we obtain inequalities

$$z \geq |T| \left(1 - \sum_{ij \in T} w_{ij} \right) \tag{5}$$

for every k-truss inducing subset T.

A second alternative is obtained by considering subsets of removed edges $W \subset E$ with $|W| = b$ (again, inspired by the same idea in [1]). In this case, the k-truss in $G - W$ has to be determined, say $t(W)$. We obtain

$$z \geq t(W) \left(\sum_{ij \in W} w_{ij} - (b-1) \right) \tag{6}$$

for every subset W of size b. Both sets of inequalities can be combined to improve the performance.

Finally, the bilevel model can be directly handed to a general-purpose bilevel optimization solver like the one presented in [3]. Despite its quadratic nature, our computational results show that the dualization approach outperforms both the bilevel solver and the single level reformulation by exponentially many constraints (5) and (6).

References

[1] M. Cerulli, D. Serra, C. Sorgente, C. Archetti, and I. Ljubić. Mathematical programming formulations for the collapsed k-core problem. *European Journal of Operational Research*, 311:56–72, 2023.

[2] C. Chen, M. Zhang, R. Sun, X. Wang, W. Zhu, and X. Wang. Locating pivotal connections: the k-truss minimization and maximization problems. *World Wide Web*, 25:899–926, 2022.

[3] M. Fischetti, I. Ljubić, M. Monaci, and M. Sinnl. A new general-purpose algorithm for mixed-integer bilevel linear programs. *Operations Research*, 65(6):1615–1637, 2017.

[4] J. Wang and J. Cheng. Truss decomposition in massive networks. In *Proceedings of the VLDB Endowment*, volume 5, pages 812–823. VLDB Endowment, 2012.