ROUNDING OUT STRUCTURAL ROUNDING

Madison Cooley, Blair D. Sullivan, Brian Lavallee

SIAM Workshop on Network Science 2020
July 9–10 · Toronto

Summary

Structural rounding is a framework developed for designing polynomial-time approximation algorithms on graphs close to a structural class [3]. Structural rounding works by editing the original graph into a well-structured class, solving the specified problem on the edited graph, and then lifting the solution to the original graph. This allows us to expand the scope of class-specific approximation algorithms to all graphs 'close' to that class without sacrificing approximation guarantees. Recent work established that structural rounding is both practical and effective in a limited setting [6], however several key barriers remain to making the framework broadly applicable in real-world network analysis.

Motivation

Real-world networks are inherently noisy due to many factors, some of which include measurement error, uncertainty, or variations from an underlying model. This additional noise makes running structure-based algorithms infeasible and often hinders approximation quality in algorithms that rely on greedy choices. Fortunately, there is evidence [7, 4] that many real-world networks are noisy representations of intrinsically structured data. Specifically, these networks can be transformed into a graph from a structural class via a short sequence of edit operations (e.g., vertex/edge deletion or edge contraction). Structural rounding [3] guarantees polynomial-time approximation algorithms in exactly this setting; further, in contrast to prior work [7, 2, 1], the approximation guarantees are relative to the optimal solution on the (noisy) original network not the edited graph.

Recent work applied the structural rounding framework to solve Vertex Cover in near-bipartite graphs [6], including a Python implementation. This work empirically established that algorithms from the framework can achieve both practical runtimes and high-quality solutions, competitive even with heavily-adopted heuristics, while maintaining approximation guarantees. While this illustrated the promise of structural rounding, in order for it to become a reliable tool in the network science toolkit, we must demonstrate scalability and practicality for additional problems and classes.

Structural Rounding Framework

The framework consists of three steps: first, edit the input graph into the desired class, then solve the optimization problem using an existing class-specific algorithm, and finally lift the partial solution from the edited instance to a solution on the original graph. See Figure 1.

To be specific, we define a graph G to be $\gamma - close$ to a specified structural class C, under edit operation ψ, if the graph $G' \in C$ can be obtained from G via some sequence of $k \leq \gamma$ edits of type ψ. An algorithm for $(C_\lambda, \psi) - Edit$, where λ is the class parameter, gives a (bicriteria) $(\alpha, \beta) - approximation$ if the number of edits is guaranteed to be at most α times the optimal number of edits into the class C_λ and the edited graph G' is in the class $C_{\beta\lambda}$. For a problem Π to be amenable to the structural rounding framework, it must simultaneously satisfy two properties– stability and structural liftability. Structural rounding supports both minimization and maximization problems–we outline the minimization case here. A minimization problem Π is *stable* under edit operation ψ with constant c' if $OPT(G') \leq OPT(G) + c'\gamma$. Additionally, a (minimization) problem Π can be *structurally lifted* with respect to ψ with constant c if given a solution S' to Π on G', a solution S to Π on G can be found in polynomial time such that $Cost_\Pi(S) \leq Cost_\Pi(S') + ck$. When the class C has a ρ-approximation algorithm for Π and $\gamma \leq \delta OPT_\Pi$, where $\delta = \delta(\epsilon, \alpha) > 0$, structural rounding results in a $((1 + c'\alpha\delta) \cdot \rho(\beta\lambda) + c\alpha\delta)$-approximation for Π on graphs γ-close to C_λ (Theorem 4.1 in [3]).

Many common graph optimization problems satisfy the conditions of structural rounding, including Vertex Cover, Feedback Vertex Set, Independent Set, Minimum Maximal Matching, Chromatic Number, Dominating Set, and Max-Cut. While the framework is agnostic to the choice of graph class C, many well-studied classes (e.g. bounded degeneracy, bounded genus,

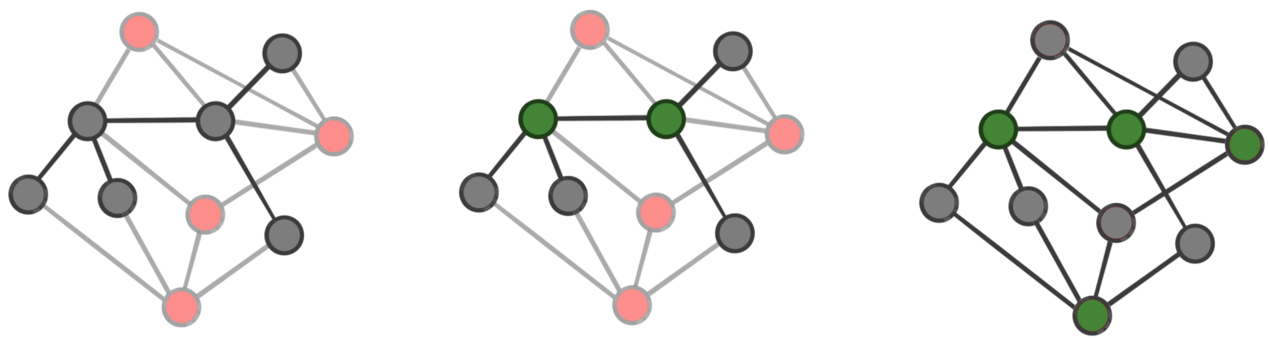

Figure 1: Illustration of applying structural rounding to solve VERTEX COVER by editing to bounded treewidth. In step 1, we edit to treewidth 1 (a forest), marking deleted vertices in red. In step 2, we solve vertex cover on the edited graph, marking nodes in our solution with green. Finally, in step 3, we lift our partial solution to the original graph, marking the full vertex cover with green.

and bounded treewidth) are promising targets due to a rich body of exact and high-quality approximation algorithms for graphs in the class.

Applicability

A recent C++ implementation[1] of [6] has increased the runtime performance significantly. Initial experiments show that the C++ implementation runs approximately 10 times faster than the Python version. This performance increase ensures that structural rounding can efficiently deliver high-quality solutions backed by theoretical guarantees at scale, easily handling graphs with more than 200 million edges.

While finding a class with efficient approximation algorithms may be easy, editing into that class is often hard (e.g. the best-known approximation for editing to planar is polylogarithmic [5]). Because of the strong dependence of the structural rounding approximation factor on the number of edits, this creates a worrisome barrier for networks near many natural classes.

Therefore, in this work, we examine potential extensions to the structural rounding framework which would accommodate the difficulty of editing to specific classes. For example, the current approach requires editing algorithms to be problem-agnostic; could it be adapted to allow problem-specific editing strategies that provided guarantees on minimizing the change in optimal solution value? Alternatively, is an approximation on the maximal subgraph from a class sufficient? If so, are these algorithms easier to design and implement?

[1] Code available at https://github.com/TheoryInPractice/structural-rounding

References

[1] N. Bansal, D. Reichman, and S. W. Umboh. Lp-based robust algorithms for noisy minor-free and bounded treewidth graphs. *SODA*, 2017.

[2] T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete Comput. Geom.*, 2012.

[3] E. D. Demaine, T. D. Goodrich, K. Kloster, B. Lavallee, Q. C. Liu, B. D. Sullivan, A. Vakilian, and A. van der Poel. Structural rounding: Approximation algorithms for graphs near an algorithmically tractable class. *ESA*, 2019.

[4] E. D. Demaine, F. Reidl, P. Rossmanith, F. S. Villaamil, S. Sikdar, and B. D. Sullivan. Structural sparsity of complex networks: Bounded expansion in random models and real-world graphs. *J. Comput. Syst. Sci.*, 2019.

[5] K. Kawarabayashi and A. Sidiropoulos. Polylogarithmic approximation for minimum planarization (almost). *FOCS*, 2017.

[6] B. Lavallee, H. Russell, B. D. Sullivan, and A. van der Poel. Approximating Vertex Cover using Structural Rounding. *ALENEX*, 2019.

[7] A. Magen and M. Moharrami. Robust algorithms for max independent set on minor-free graphs based on the sherali-adams hierarchy. *APPROX*, 2009.