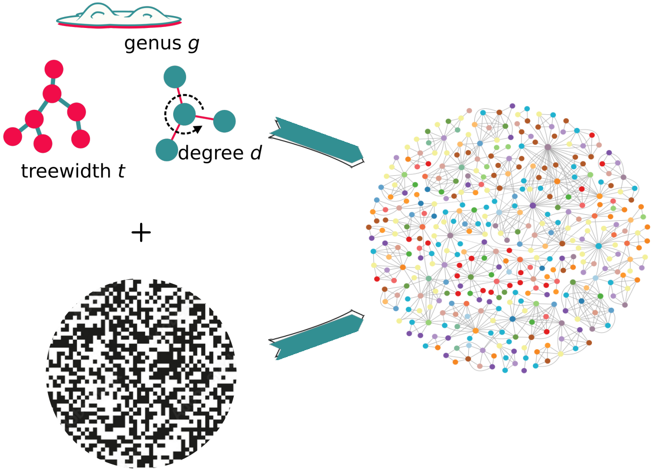
Structural Rounding

Brian Lavallee

18 October 2022

Approximations for Real-World Networks

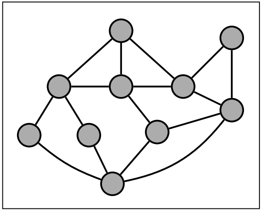


- **Structural rounding** gives better approximations by assuming networks are *close* to a structural class.

Outline

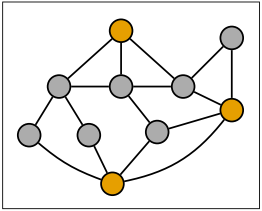
- Structural Rounding Framework
- Extending to Realistic Classes
  - Editing to Bounded Expansion
- Empirical Evaluation
  - Vertex Cover in Near-Bipartite Graphs
  - Dominating Set in Near-Bounded Treewidth Graphs
- Summary and Open Questions

Structural Rounding



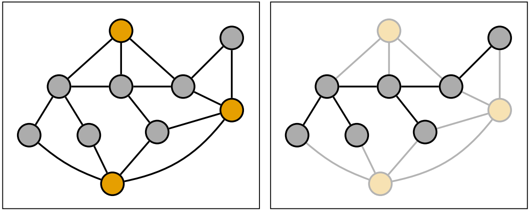
Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$.

Structural Rounding



Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$.

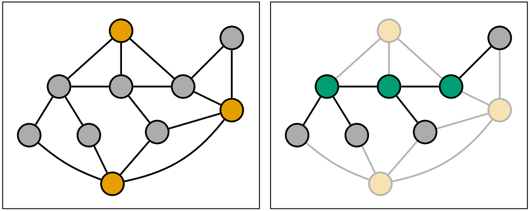
Structural Rounding



Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$.

Solve: Run an exact or approximate algorithm on the edited instance in polynomial time.

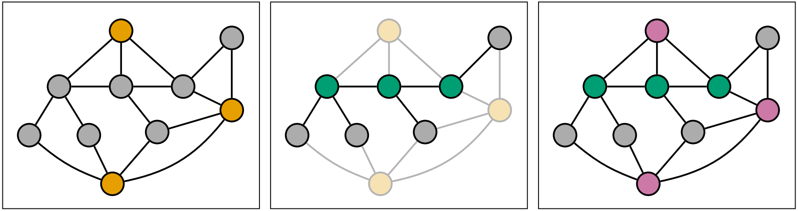
Structural Rounding



Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$.

Solve: Run an exact or approximate algorithm on the edited instance in polynomial time.

Structural Rounding

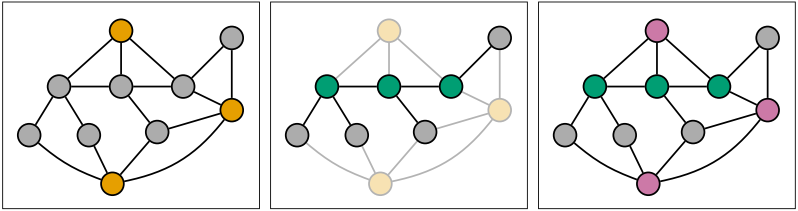


Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$.

Solve: Run an exact or approximate algorithm on the edited instance in polynomial time.

Lift: Extend the partial solution to G.

Structural Rounding

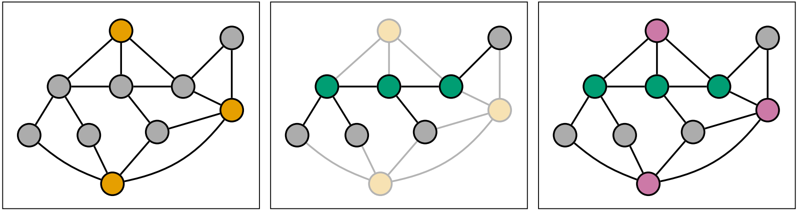


Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$. A problem is c'-**stable** w.r.t. ψ if $\psi_1, \ldots, \psi_d$ cannot increase $\mathrm{opt}(G)$ by more than $c' \cdot d$.

Solve: Run an exact or approximate algorithm on the edited instance in polynomial time.

Lift: Extend the partial solution to G.

Structural Rounding

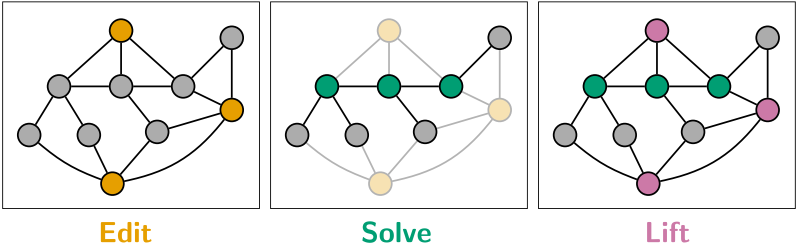


Edit: Use ψ to edit G to be in $\mathcal{C}_\lambda$. A problem is c'-**stable** w.r.t. ψ if $\psi_1, \ldots, \psi_d$ cannot increase $\text{opt}(G)$ by more than $c' \cdot d$.

Solve: Run an exact or approximate algorithm on the edited instance in polynomial time.

Lift: Extend the partial solution to G. A problem can be c-**lifted** w.r.t. ψ if any partial solution S can be extended to G by undoing $\psi_1, \ldots, \psi_d$ without increasing $\text{cost}(S)$ by more than $c \cdot d$.

Structural Rounding



Theorem (Demaine, **L** et al, 2019)

Let Π *be a problem*

- *that is* c'*-stable under* ψ *and*
- *that can be* c*-lifted w.r.t.* ψ.

If Π *has a poly-time* $\rho(\lambda)$*-approximation in* $\mathcal{C}_\lambda$ *and* $(\mathcal{C}_\lambda, \psi)$-Editing *has a poly-time* (α, β)*-approximation, then there is a poly-time* $((1 + c'\alpha\delta) \cdot \rho(\beta\lambda) + c\alpha\delta)$*-approximation for* Π *on graphs that are* $(\delta \cdot \mathrm{opt}_\Pi(G))$*-close to* $\mathcal{C}_\lambda$.

Extending to Realistic Classes

- Editing to Bounded Expansion
- Generalized Coloring Numbers

Why Editing?

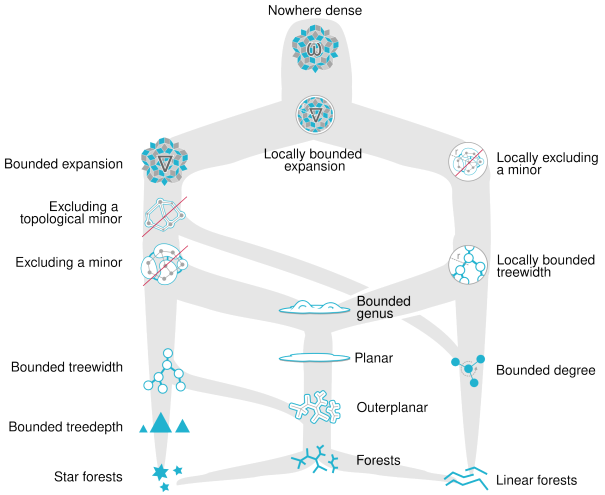


image credit: Felix Reidl

Why Editing?

Smaller Edit Sets

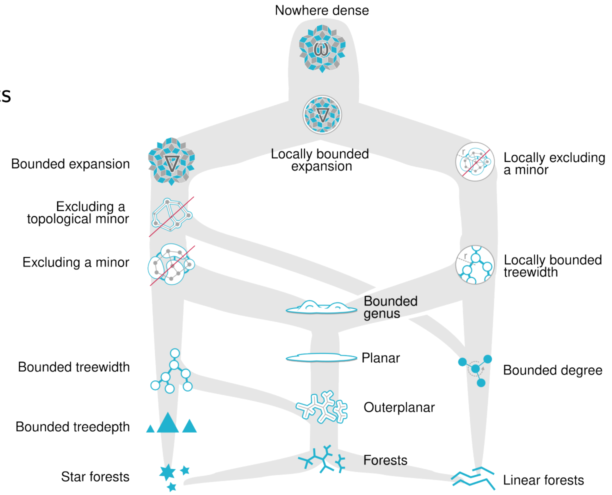


image credit: Felix Reidl

Why Editing?

Smaller Edit Sets

Faster/Better Approximations

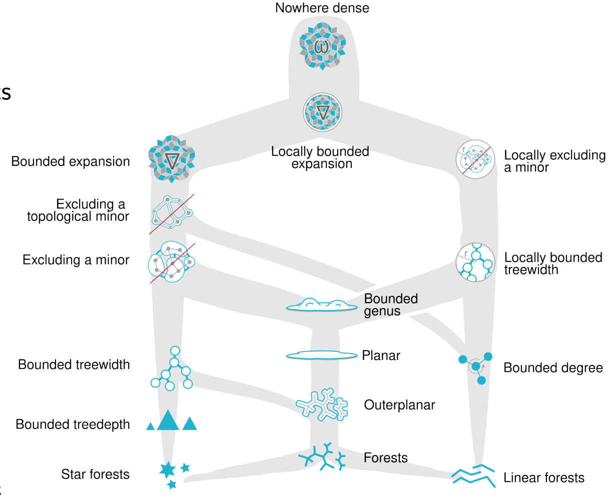


image credit: Felix Reidl

Why Bounded Expansion?

Smaller Edit Sets

Faster/Better Approximations

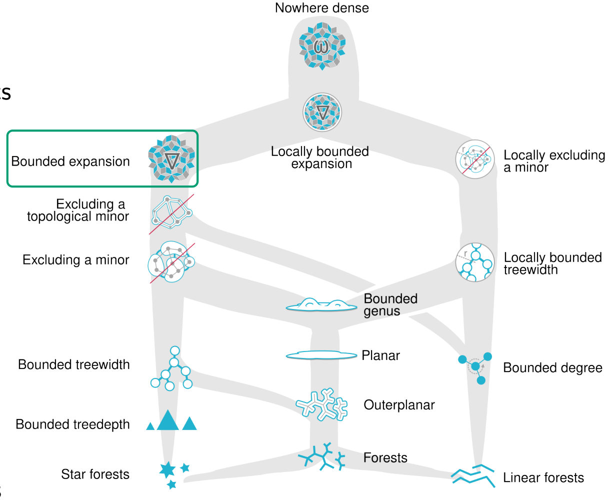


image credit: Felix Reidl

Why Bounded Expansion?

Smaller Edit Sets

Faster/Better Approximations

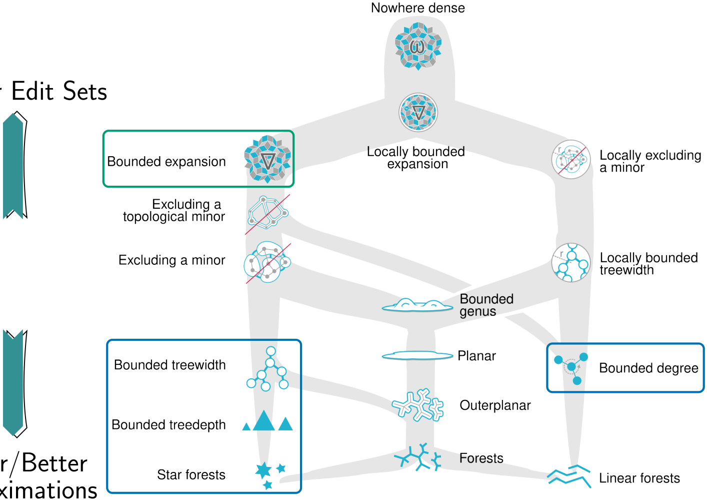


image credit: Felix Reidl

Why Bounded Expansion?

Smaller Edit Sets

Faster/Better Approximations

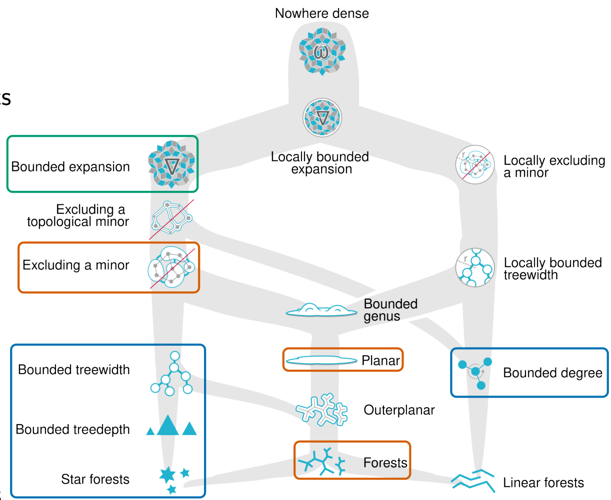
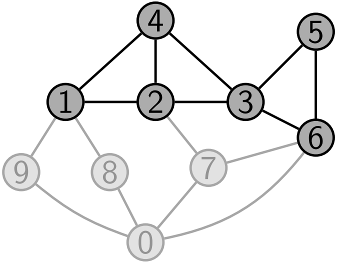


image credit: Felix Reidl

Generalized Coloring Numbers

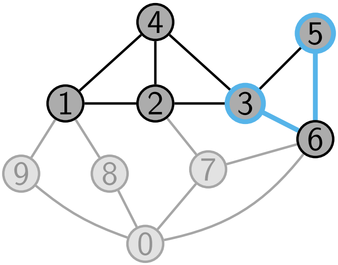
- extend degeneracy to larger r-neighborhoods
- classes with bounded expansion have $\mathrm{col}_r(G) \leq f(r)$



$$\sigma = [9, 8, 7, 0, \mathbf{6}, 5, 3, 4, 2, 1]$$

Generalized Coloring Numbers

- extend degeneracy to larger r-neighborhoods
- classes with bounded expansion have $\mathrm{col}_r(G) \leq f(r)$

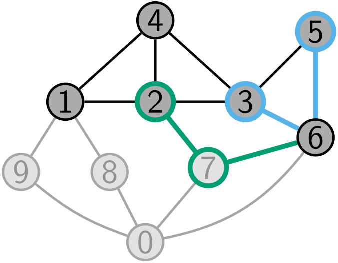


$\sigma = [9, 8, 7, 0, \mathbf{6}, 5, 3, 4, 2, 1]$

- $\mathrm{R}_1(6, G_\sigma) = \{3, 5\}$

Generalized Coloring Numbers

- extend degeneracy to larger r-neighborhoods
- classes with bounded expansion have $\mathrm{col}_r(G) \leq f(r)$

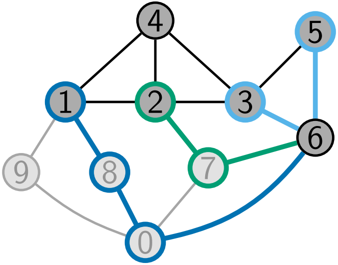


$\sigma = [9, 8, 7, 0, \mathbf{6}, 5, 3, 4, 2, 1]$

- $\mathrm{R}_1(6, G_\sigma) = \{3, 5\}$
- $\mathrm{R}_2(6, G_\sigma) = \{2, 3, 5\}$

Generalized Coloring Numbers

- extend degeneracy to larger r-neighborhoods
- classes with bounded expansion have $\mathrm{col}_r(G) \leq f(r)$

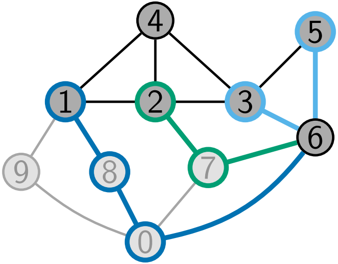


$\sigma = [9, 8, 7, 0, \mathbf{6}, 5, 3, 4, 2, 1]$

- $\mathrm{R}_1(6, G_\sigma) = \{3, 5\}$
- $\mathrm{R}_2(6, G_\sigma) = \{2, 3, 5\}$
- $\mathrm{R}_3(6, G_\sigma) = \{1, 2, 3, 5\}$

Generalized Coloring Numbers

- extend degeneracy to larger r-neighborhoods
- classes with bounded expansion have $\mathrm{col}_r(G) \leq f(r)$



$\sigma = [9, 8, 7, 0, \mathbf{6}, 5, 3, 4, 2, 1]$

- $\mathrm{R}_1(6, G_\sigma) = \{3, 5\}$
- $\mathrm{R}_2(6, G_\sigma) = \{2, 3, 5\}$
- $\mathrm{R}_3(6, G_\sigma) = \{1, 2, 3, 5\}$

$$\mathrm{col}_r(G) := \min_{\sigma \in \pi} \max_{u \in V} |\,\mathrm{R}_r(u, G_\sigma)|$$

Generalized Coloring Numbers

Theorem (Breen-McKay, **L**, Sullivan, 2022)

There is a greedy $(k-1)^{r-1}$ *approximation for* $\mathrm{col}_r(G)$ *in graphs with* r*-admissibility at most* k*.*

- $\mathrm{adm}_r(G) \leq \mathrm{col}_r(G)$

Theorem (Breen-McKay, **L**, Sullivan, 2022)

It is NP-hard to determine if $\mathrm{col}_r(G) \leq 6$ *when* $r \geq 2$*.*

Implications:

- No polynomial time $\frac{7}{6}$-approximation.
- No $O(n^{f(t)})$ time algorithm to check if $\mathrm{col}_r(G) \leq t$.

Coloring Number Deletion

r-Coloring Number Deletion

Input: a graph $G = (V, E)$ and a target t.

Problem: find a set of vertices $X \subseteq V$ of minimum size such that $\mathrm{col}_r(G \setminus X) \leq t$.

Theorem (Breen-McKay, Kloster, **L**, Sullivan, 2019)

There exists a constant c such that it is NP-hard to approximate r-Coloring Number Deletion within a factor of $c \cdot \log \frac{n}{r+t}$.

Coloring Number Deletion

r-Coloring Number Deletion

Input: a graph $G = (V, E)$ and a target t.
Problem: find a set of vertices $X \subseteq V$ of minimum size such that $\mathrm{col}_r(G \setminus X) \leq t$.

Theorem (Breen-McKay, Kloster, **L**, Sullivan, 2019)

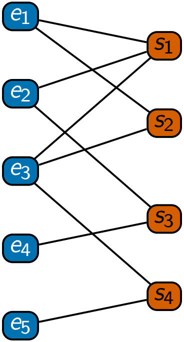
There exists a constant c such that it is NP-hard to approximate r-Coloring Number Deletion *within a factor of $c \cdot \log \frac{n}{r+t}$.*

Set Cover

Input: a universe of elements $\mathcal{U}$ and a family of sets $\mathcal{F}$.
Problem: find a set $S \subseteq \mathcal{F}$ of minimum size such that $\bigcup_{s_i \in S} s_i = \mathcal{U}$.

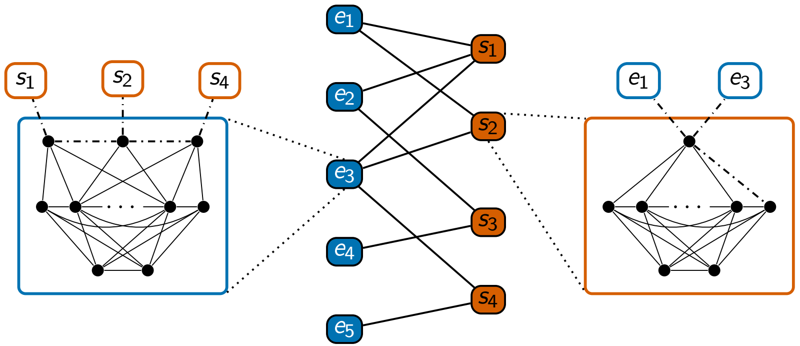
Coloring Number Deletion

$$\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5\} \; \mathcal{F} = \Big\{(e_1, e_2, e_3), (e_1, e_3), (e_2, e_4), (e_3, e_5)\Big\}$$



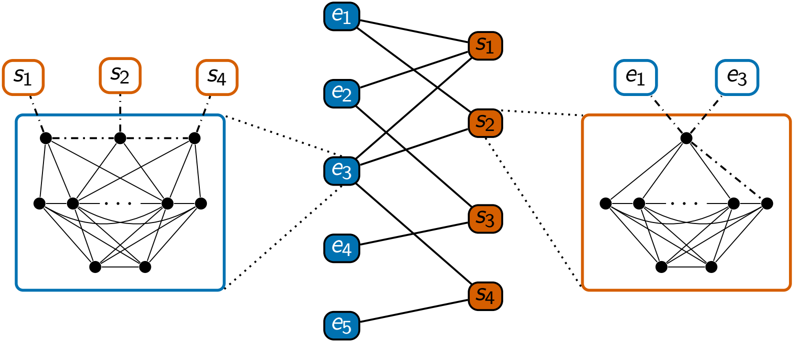
Coloring Number Deletion

$$\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5\} \; \mathcal{F} = \Big\{(e_1, e_2, e_3), (e_1, e_3), (e_2, e_4), (e_3, e_5)\Big\}$$



Coloring Number Deletion

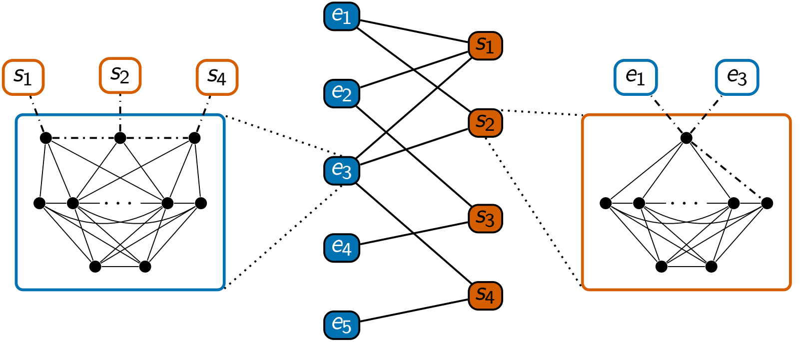
$$\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5\} \ \mathcal{F} = \Big\{(e_1, e_2, e_3), (e_1, e_3), (e_2, e_4), (e_3, e_5)\Big\}$$



- Implies Degeneracy Deletion is $o(\log \frac{n}{t})$ inapproximable.

Coloring Number Deletion

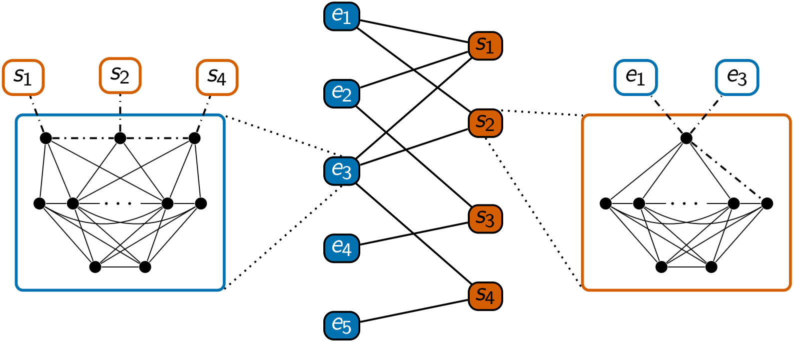
$$\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5\} \ \mathcal{F} = \Big\{(e_1, e_2, e_3), (e_1, e_3), (e_2, e_4), (e_3, e_5)\Big\}$$



Caveat: The NP-hardness of computing $\mathrm{col}_r(G)$ implies that no multiplicative approximation exists.

Coloring Number Deletion

$$\mathcal{U} = \{e_1, e_2, e_3, e_4, e_5\} \ \mathcal{F} = \Big\{(e_1, e_2, e_3), (e_1, e_3), (e_2, e_4), (e_3, e_5)\Big\}$$



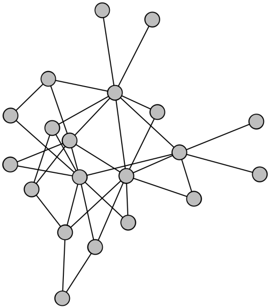
Optimistic view: Still applies when $\mathrm{col}_r(G) > t$ is guaranteed.

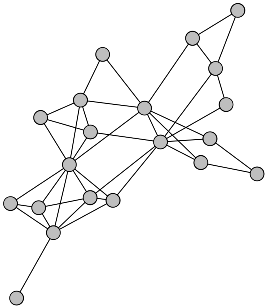
Structural Rounding in Practice

- Vertex Cover in Near-Bipartite Graphs
- Dominating Set in Near-Bounded Treewidth Graphs

Graph Corpuses

- `vc-synth`: 1,620 graphs generated with preferential attachment restricted by random labels (E, L, R)
- `ds-synth`: 750 graphs generated by taking a random subgraph of a random *k*-tree





Graph Corpuses

- `vc-synth`: 1,620 graphs generated with preferential attachment restricted by random labels (E, L, R)
- `ds-synth`: 750 graphs generated by taking a random subgraph of a random k-tree
- `pace`: 200 graphs used by 2019 PACE Challenge to assess exact VERTEX COVER algorithms
- `networks`: 207 graphs collected from scientific domains

Graph Corpuses

- `vc-synth`: 1,620 graphs generated with preferential attachment restricted by random labels (E, L, R)
- `ds-synth`: 750 graphs generated by taking a random subgraph of a random *k*-tree
- `pace`: 200 graphs used by 2019 PACE Challenge to assess exact VERTEX COVER algorithms
- `networks`: 207 graphs collected from scientific domains

corpus	vertices	edges
`vc-synth`	10K	39K - 4M
`ds-synth`	4K - 10K	6K - 4M
`pace`	153 - 138K	1K - 454K
`networks`	9 - 935K	16 - 17M

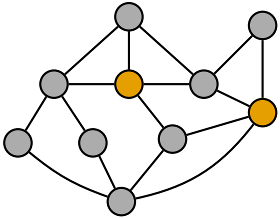
Structural Rounding in Practice

- Vertex Cover in Near-Bipartite Graphs
- Dominating Set in Near-Bounded Treewidth Graphs

Vertex Cover in Near-Bipartite Graphs

Edit: Odd Cycle Transversal

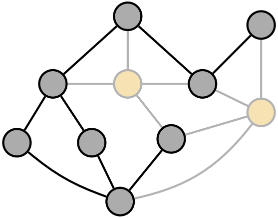
- `greedy-edit`: builds from maximal cut.
- `indset-edit`: builds from two maximal independent sets.
- `bfs-edit`: builds from layers of BFS tree.



Vertex Cover in Near-Bipartite Graphs

Edit: Odd Cycle Transversal

- `greedy-edit`: builds from maximal cut.
- `indset-edit`: builds from two maximal independent sets.
- `bfs-edit`: builds from layers of BFS tree.



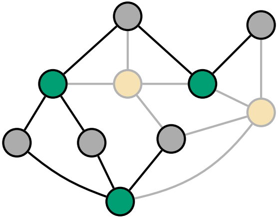
Vertex Cover in Near-Bipartite Graphs

Edit: Odd Cycle Transversal

- `greedy-edit`: builds from maximal cut.
- `indset-edit`: builds from two maximal independent sets.
- `bfs-edit`: builds from layers of BFS tree.

Solve: Vertex Cover

- `Hopcroft-Karp`: exact algorithm for VC in bipartite graphs.



Vertex Cover in Near-Bipartite Graphs

Edit: Odd Cycle Transversal

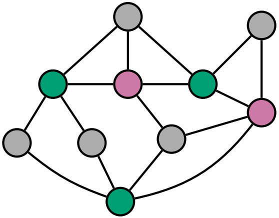
- `greedy-edit`: builds from maximal cut.
- `indset-edit`: builds from two maximal independent sets.
- `bfs-edit`: builds from layers of BFS tree.

Solve: Vertex Cover

- `Hopcroft-Karp`: exact algorithm for VC in bipartite graphs.

Lift: VC-Lift (minimum completion of a partial solution)

- `greedy-lift`
- `apx-lift`
- `oct-lift`
- `bip-lift`
- `recursive-lift`



Vertex Cover in Near-Bipartite Graphs

Edit: Odd Cycle Transversal

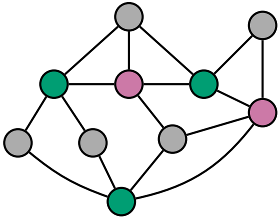
- `greedy-edit`: builds from maximal cut.
- `indset-edit`: builds from two maximal independent sets.
- `bfs-edit`: builds from layers of BFS tree.

Solve: Vertex Cover

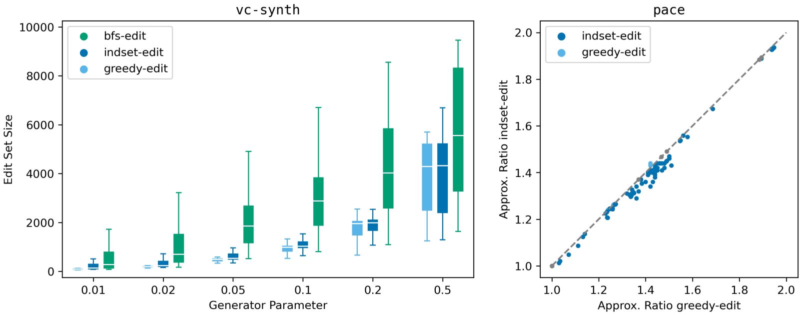
- `Hopcroft-Karp`: exact algorithm for VC in bipartite graphs.

Lift: VC-Lift (minimum completion of a partial solution)

- `greedy-lift`
- `apx-lift`
- `oct-lift`
- `bip-lift`
- `recursive-lift`

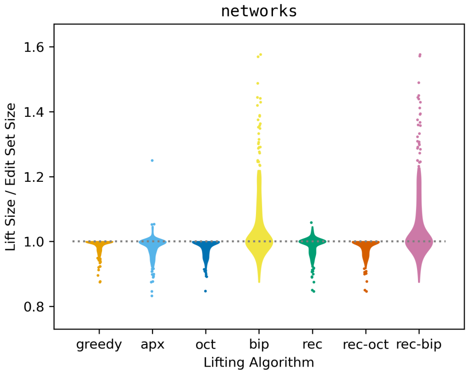


Vertex Cover in Near-Bipartite Graphs



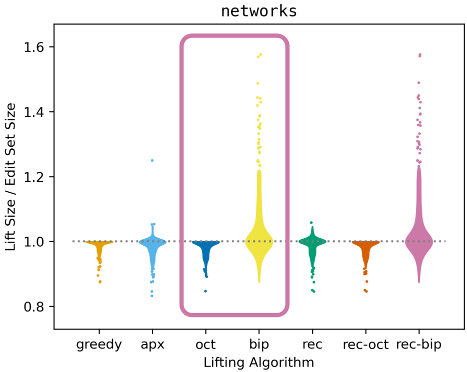
- `indset-edit` results in better solutions even though it finds larger edit sets.

Vertex Cover in Near-Bipartite Graphs



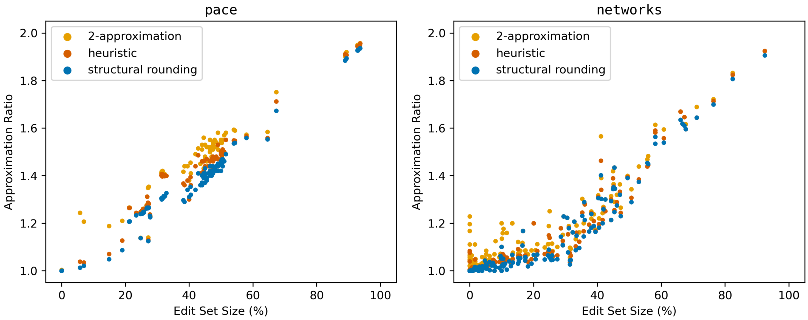
- Guaranteeing the lifting constant matters.

Vertex Cover in Near-Bipartite Graphs



- Guaranteeing the lifting constant matters.

Vertex Cover in Near-Bipartite Graphs



- Better performance even with large edit sets ($> 40\%$).
- About 1.8x slower than standard approximations/heuristics.
- Real-world networks aren't exactly *close* to bipartite.

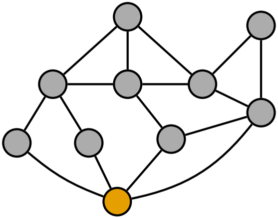
Structural Rounding in Practice

- Vertex Cover in Near-Bipartite Graphs
- Dominating Set in Near-Bounded Treewidth Graphs

Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

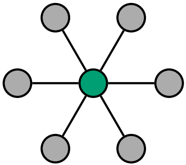
- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\mathrm{tw}(G) \leq t$.

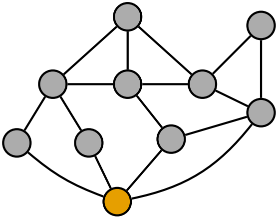


Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\text{tw}(G) \leq t$.

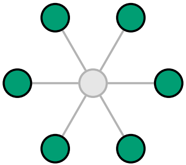


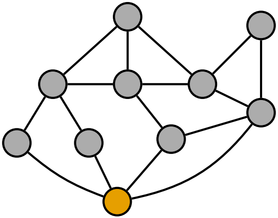


Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\mathrm{tw}(G) \leq t$.

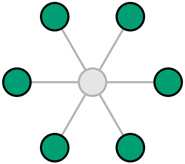


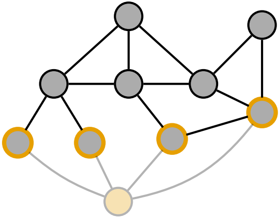


Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\mathrm{tw}(G) \leq t$.





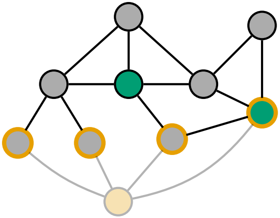
Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\mathrm{tw}(G) \leq t$.

Solve: Dominating Set

- `DP`: $O(4^t n)$ exact algorithm for DS in graphs with treewidth t.



Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

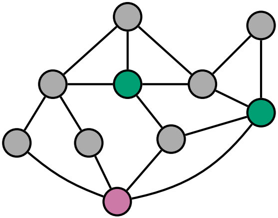
- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\text{tw}(G) \leq t$.

Solve: Dominating Set

- `DP`: $O(4^t n)$ exact algorithm for DS in graphs with treewidth t.

Lift: DS-Lift (minimum completion of a partial solution)

- `greedy-lift`
- `apx-lift`



Dominating Set in Near-Treewidth Graphs

Edit: Treewidth Deletion

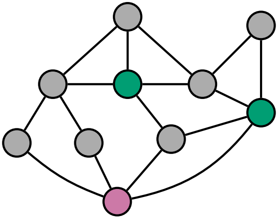
- `greedy-edit`: finds a tree decomposition and removes vertices until it has width t.
- `root-edit`: removes vertex separators until $\mathrm{tw}(G) \leq t$.

Solve: Dominating Set

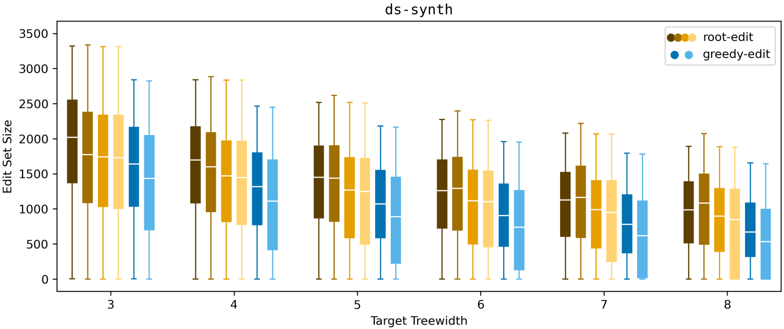
- `DP`: $O(4^t n)$ exact algorithm for DS in graphs with treewidth t.

Lift: DS-Lift (minimum completion of a partial solution)

- `greedy-lift`
- `apx-lift`

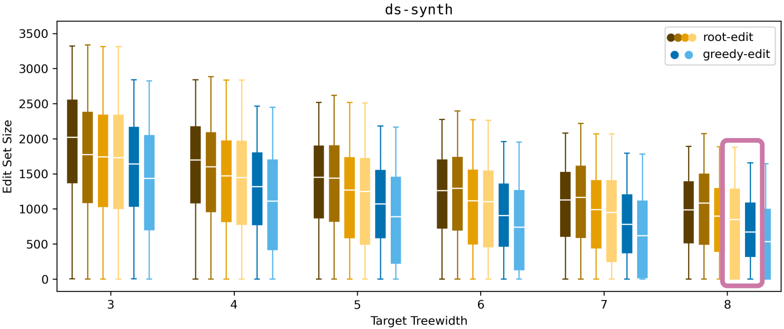


Dominating Set in Near-Bounded Treewidth Graphs



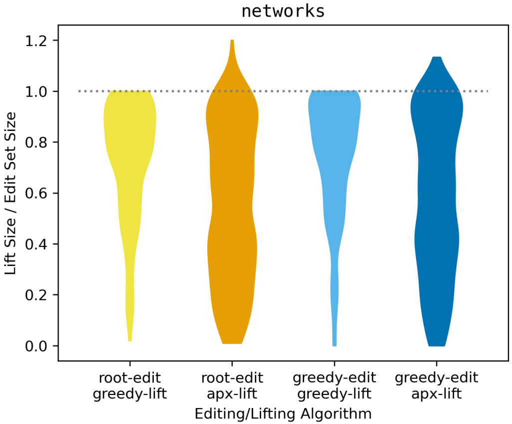
- `greedy-edit` produces smaller edit sets than `root-edit`.

Dominating Set in Near-Bounded Treewidth Graphs



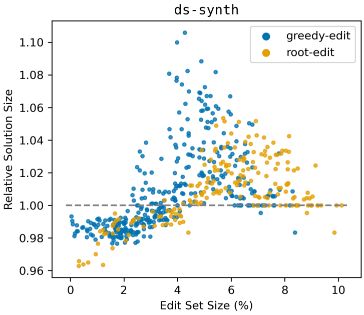
- `greedy-edit` produces smaller edit sets than `root-edit`.
- final algorithm has runtime quadratic in treewidth of input.

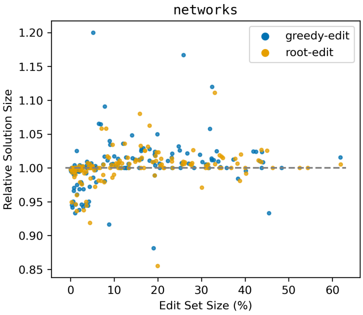
Dominating Set in Near-Bounded Treewidth Graphs



- Guaranteeing the lifting constant still matters.

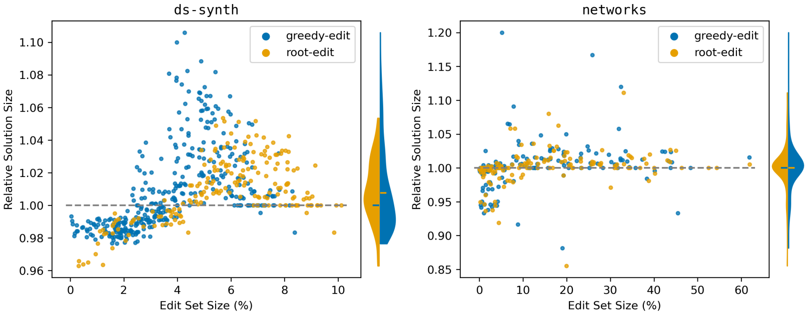
Dominating Set in Near-Bounded Treewidth Graphs





- Improved performance for small ($< 5\%$) edit sets.
- Comparable performance overall, but 60x slower.
- Real-world networks are closer to treewidth 8 than bipartite.

Dominating Set in Near-Bounded Treewidth Graphs



- Improved performance for small ($< 5\%$) edit sets.
- Comparable performance overall, but 60x slower.
- Real-world networks are closer to treewidth 8 than bipartite.

Summary

- Structural Rounding is a viable technique for approximating NP-hard problems in real-world networks.
  - SR outperforms standard VERTEX COVER algorithms with only 1.8x slowdown.
  - SR finds small dominating sets in graphs close to tw-8 ($< 5\%$).

Summary

- Structural Rounding is a viable technique for approximating NP-hard problems in real-world networks.
  - SR outperforms standard Vertex Cover algorithms with only 1.8x slowdown.
  - SR finds small dominating sets in graphs close to tw-8 ($< 5\%$).
- r-Coloring Number Deletion is $o(\log \frac{n}{r+t})$ inapprox.

Summary

- Structural Rounding is a viable technique for approximating NP-hard problems in real-world networks.
  - SR outperforms standard Vertex Cover algorithms with only 1.8x slowdown.
  - SR finds small dominating sets in graphs close to tw-8 ($< 5\%$).

- r-Coloring Number Deletion is $o(\log \frac{n}{r+t})$ inapprox.

- It is NP-hard to decide $\mathrm{col}_r(G) \leq 6$.
  - No $\frac{7}{6}$ approximation.
  - No $n^{f(t)}$ algorithm to check $\mathrm{col}_r(G) \leq t$.

Summary

- Structural Rounding is a viable technique for approximating NP-hard problems in real-world networks.
  - SR outperforms standard VERTEX COVER algorithms with only 1.8x slowdown.
  - SR finds small dominating sets in graphs close to tw-8 ($< 5\%$).
- r-COLORING NUMBER DELETION is $o(\log \frac{n}{r+t})$ inapprox.
- It is NP-hard to decide $\text{col}_r(G) \leq 6$.
  - No $\frac{7}{6}$ approximation.
  - No $n^{f(t)}$ algorithm to check $\text{col}_r(G) \leq t$.
- The generalized coloring numbers admit a $(k-1)^{r-1}$ approximation in graphs with r-admissibility at most k.

Open Questions

- Performance of structural rounding using edge deletions
  - DOMINATING SET has better stability and lifting constant w.r.t. edge deletion

Open Questions

- Performance of structural rounding using edge deletions
  - Dominating Set has better stability and lifting constant w.r.t. edge deletion
- Precise approximation bounds for $\mathrm{col}_r(G)$
  - Large gap between $\frac{7}{6}$ and $(k-1)^{r-1}$
  - Constant factor approximation for $\mathrm{col}_2(G)$?

Open Questions

- Performance of structural rounding using edge deletions
  - DOMINATING SET has better stability and lifting constant w.r.t. edge deletion
- Precise approximation bounds for $\mathrm{col}_r(G)$
  - Large gap between $\frac{7}{6}$ and $(k-1)^{r-1}$
  - Constant factor approximation for $\mathrm{col}_2(G)$?
- Bicriteria inapproximability
  - Inapproximability results for editing to classes with NP-hard parameters when class parameter is approximate

Thanks!

- Blair Sullivan
- Drew van der Poel
- Mike Breen-McKay
- Kyle Kloster
- Hayley Russell
- Madi Cooley
- Cole Perschon
- Erik Demaine
- Ali Vakilian
- Quanquan Liu
- Tim Goodrich
- Felix Reidl
- Mike O'Brien
- Andrew Fraser

GORDON AND BETTY MOORE FOUNDATION