Structural Rounding: Approximation Algorithms for Graphs Near an Algorithmically Tractable Class

Erik D. Demaine	MIT CSAIL
Timothy D. Goodrich	NC State University
Kyle Kloster	NC State University
Brian Lavallee	NC State University
Quanquan C. Liu	MIT CSAIL
Blair D. Sullivan	NC State University
Ali Vakilian	MIT CSAIL
Andrew van der Poel	NC State University

European Symposium on Algorithms (ESA 2019)

Noisy Graphs

- Real-world networks tend to be **sparse and have a certain structure but are noisy**

- Hard problems (independent set, dominating set, minimum maximal matching…etc.) can be solved in polytime in graphs with certain structure (**bounded treewidth, bounded degeneracy**…etc.)

- Even approximation is hard for some problems in general graphs

- <u>Our Results:</u> Solve such problems using approx. algs. and PTASes in **graphs close to structured graph classes**

Graph Editing and Solving Hard Problems

- **Strategy:**
  - **Editing algorithms:** Given a graph **γ-close** to a structured graph class $\mathcal{C}$, find series of $f(\gamma)$ edits
  - **Structural rounding algorithms:** Convert ρ-approximate solutions on the edited graph in $\mathcal{C}$ to **$g(\rho, \gamma)$-approximate solutions** in the original graph

Related Works

- **Using Editing to Approximate Optimization Problems**

  - Magen and Moharrami '09: approx. *size* of maximum independent sets (IS) in graphs δn away from a minor-closed graph family for sufficiently small δ

  - Chan and Har-Peled '12: PTAS for IS in noisy planar graphs

  - Bansal et. al. '17 [BRU17]: LP-based approach for noisy minor-closed IS and noisy Max k-CSPs for *edge edits*; uses techniques similar to our structural rounding framework

Related Works

- **Editing Approximation Algorithms**
  - Fomin et al. '12 [FLMS12]: vertex editing to planar $\mathcal{H}$-minor-free graphs; randomized $c_{\mathcal{H}}$-approx. alg.
  - Aggrawal et al. '18 [ALMSZ18]: $O(\log^{1.5} n)$-approximation for Weighted Planar $\mathcal{F}$-Vertex Deletion via a randomized algorithm
  - Bansal et al. '17 [BRU17]: bicriteria LP-based algorithm for edge treewidth editing with $\big(O(\log n \log\log n), O(\log w)\big)$-approx.

Related Works

- **Editing FPT Algorithms**
  - Fomin et al. '12 [FLMS12]: vertex editing to planar $\mathcal{H}$-minor-free graphs an FPT algorithm parameterized by the size of the edit set
  - Guruswami and Lee '17: FPT $O(\log k)$-approximation algorithms for $\mathcal{H}$-Vertex-Deletion when $\mathcal{H}$ is a star or a simple path
  - Gupta et al. '18 [GLLMW18]: parameterized $O(\log w)$-approximation algorithm for editing to bounded treewidth with running time $2^{O(w^3 \log^3 w)} \cdot n \log n + n^{O(1)}$

Our Approximation Results

Problem	Edit Type	Approximation Factor
Independent Set	Vertex Deletion	$\frac{1}{r+1}(1-6\delta)$; degeneracy
Annotated Dominating Set	Vertex* Deletion	$r+6\delta$; degeneracy
Independent Set	Vertex Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\sqrt{\log\log n}}\right)$
Annotated Dominating Set	Vertex* Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\sqrt{\log\log n}}\right)$
Annotated (ℓ)-Dominating Set	Vertex* Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log_\ell n}{\sqrt{\log\log n}}\right)$
Connected Dominating Set	Vertex Deletion	PTAS $O\left(\frac{OPT}{3\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O(1)$
Vertex Cover	Vertex Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\sqrt{\log\log n}}\right)$
Feedback Vertex Set	Vertex Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\sqrt{\log\log n}}\right)$
Minimum Maximal Matching	Vertex Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\sqrt{\log\log n}}\right)$
Chromatic Number	Vertex Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log n}{\log\log n\sqrt{\log\log n}}\right)$
Independent Set	Edge Deletion	$\frac{1}{r+1}-6\delta$; degeneracy
Dominating Set	Edge Deletion	$r(1+6\delta)$; degeneracy
(ℓ)-Dominating Set	Edge Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log_\ell n}{\sqrt{\log\log n}}\right)$
Edge (ℓ)-Dominating Set	Edge Deletion	PTAS $O\left(\frac{OPT}{2\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log_\ell n}{\sqrt{\log\log n}}\right)$
Max-Cut	Edge Deletion	PTAS $O\left(\frac{OPT}{\epsilon\cdot\log^{1.5} n}\right)$-far from treewidth $O\left(\frac{\log_\ell n}{\sqrt{\log\log n}}\right)$

Table of Editing Results

Graph Family $\mathcal{C}_\lambda$	Edit Operation ψ: Vertex Deletion	Edit Operation ψ: Edge Deletion
Bounded Degree (d)	**d-BDD-V** $O(\log d)$-approx. [30] $(\ln d - C \cdot \ln\ln d)$-inapprox.	**d-BDD-E** Polynomial time [47]
Bounded Degeneracy (r)	**r-DE-V** $O(r \log n)$-approx. $\left(\frac{4m-\beta rn}{m-rn}, \beta\right)$-approx. $\left(\frac{1}{\varepsilon}, \frac{4}{1-2\varepsilon}\right)$-approx. ($\varepsilon < 1/2$) $o(\log(n/r))$-inapprox.	**r-DE-E** $O(r \log n)$-approx. – $\left(\frac{1}{\varepsilon}, \frac{4}{1-\varepsilon}\right)$-approx. ($\varepsilon < 1$) $o(\log(n/r))$-inapprox.
Bounded Weak c-Coloring Number (t)	**t-BWE-V-c** – $o(t)$-inapprox. for $t \in o(\log n)$	**t-BWE-E-c** – $o(t)$-inapprox. for $t \in o(\log n)$
Bounded Treewidth (w)	**w-TW-V** $(O(\log^{1.5} n), O(\sqrt{\log w}))$-approx. $o(\log n)$-inapprox. for $w \in \Omega(n^{1/2})$	**w-TW-E** $(O(\log n \log\log n), O(\log w))$-approx. [4] –
Bounded Pathwidth (w)	**w-PW-V** $(O(\log^{1.5} n), O(\sqrt{\log w} \cdot \log n))$-approx. –	**w-PW-E** $(O(\log n \log\log n), O(\log w \cdot \log n))$-approx. [4] –
Star Forest	**SF-V** 4-approx. $(2-\varepsilon)$-inapprox. (**UGC**)	**SF-E** 3-approx. APX-complete

Table of Editing Results

Graph Family $\mathcal{C}_\lambda$	Edit Operation ψ: Vertex Deletion	Edit Operation ψ: Edge Deletion
Bounded Degree (d)	**d-BDD-V** $O(\log d)$-approx. [30] $(\ln d - C \cdot \ln \ln d)$-inapprox.	**d-BDD-E** Polynomial time [47]
Bounded Degeneracy (r)	**r-DE-V** $O(r \log n)$-approx. $\left(\frac{4m-\beta rn}{m-rn}, \beta\right)$-approx. $\left(\frac{1}{\varepsilon}, \frac{4}{1-2\varepsilon}\right)$-approx. ($\varepsilon < 1/2$) $o(\log(n/r))$-inapprox.	**r-DE-E** $O(r \log n)$-approx. – $\left(\frac{1}{\varepsilon}, \frac{4}{1-\varepsilon}\right)$-approx. ($\varepsilon < 1$) $o(\log(n/r))$-inapprox.
Bounded Weak c-Coloring Number (t)	**t-BWE-V-c** – $o(t)$-inapprox. for $t \in o(\log n)$	**t-BWE-E-c** – $o(t)$-inapprox. for $t \in o(\log n)$
Bounded Treewidth (w)	**w-TW-V** $(O(\log^{1.5} n), O(\sqrt{\log w}))$-approx. $o(\log n)$-inapprox. for $w \in \Omega(n^{1/2})$	**w-TW-E** $(O(\log n \log\log n), O(\log w))$-approx. [4] –
Bounded Pathwidth (w)	**w-PW-V** $(O(\log^{1.5} n), O(\sqrt{\log w} \cdot \log n))$-approx. –	**w-PW-E** $(O(\log n \log\log n), O(\log w \cdot \log n))$-approx. [4] –
Star Forest	**SF-V** 4-approx. $(2-\varepsilon)$-inapprox. (**UGC**)	**SF-E** 3-approx. APX-complete

Only Talk about These Results!

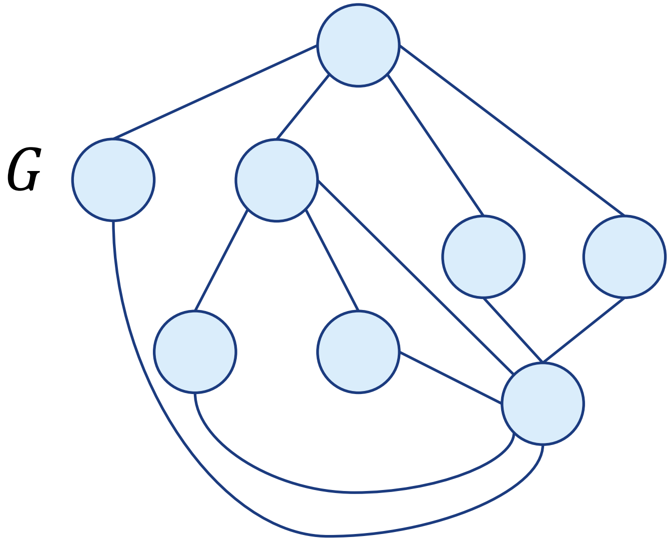
Structural Rounding

Framework for approximating solutions in close to structured graphs

Structural Rounding

(Minimization) Problem $\mathcal{P}$ with optimal solution $OPT(G)$

Framework for approximating solutions in close to structured graphs

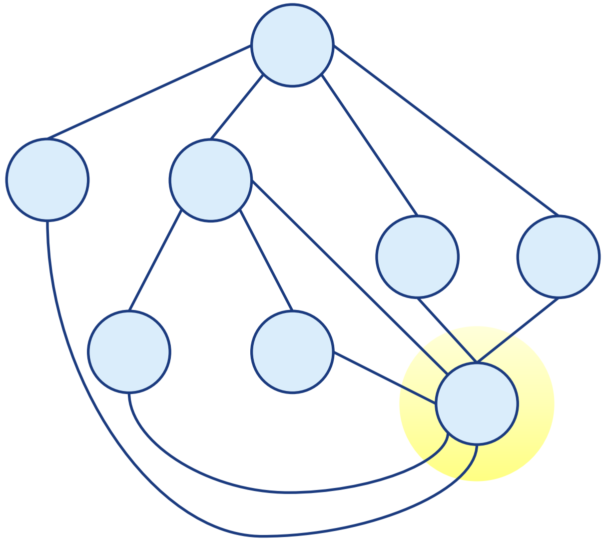


Structural Rounding

(Minimization) Problem $\mathcal{P}$ with optimal solution $OPT(G)$

Framework for approximating solutions in close to structured graphs

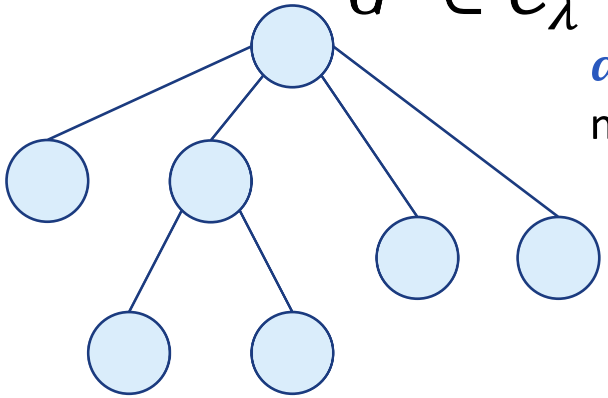
G



Edit to a Structured Graph Class

(α, β)-approximation

$G' \in \mathcal{C}_\lambda$



α-approx. min number of edits X

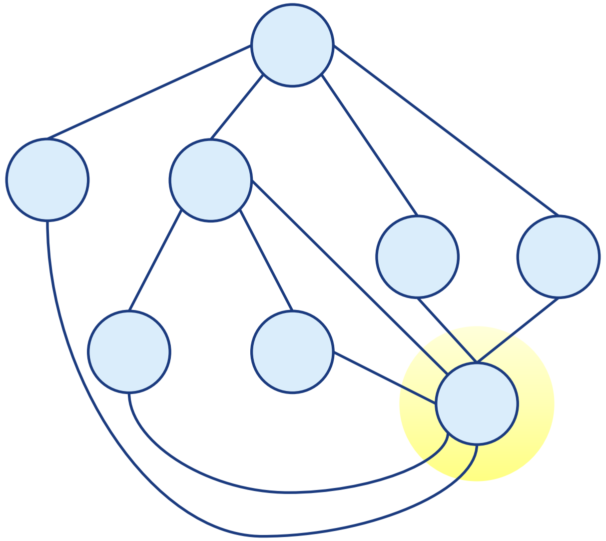
β-approx. on $\mathcal{C}_\lambda$
e.g. $\lambda \leq \beta \cdot \lambda^*$

Structural Rounding

(Minimization) Problem $\mathcal{P}$ with optimal solution $OPT(G)$

Framework for approximating solutions in close to structured graphs

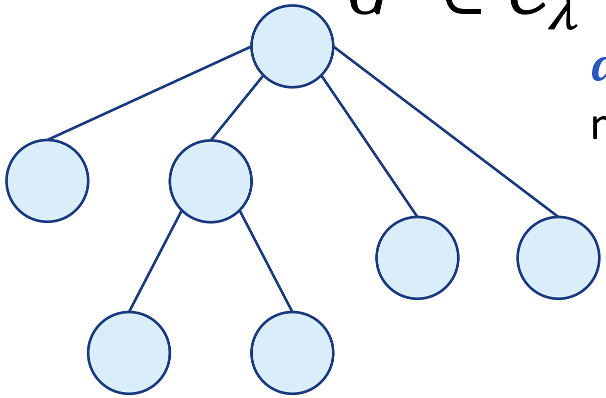
G



Edit to a Structured Graph Class

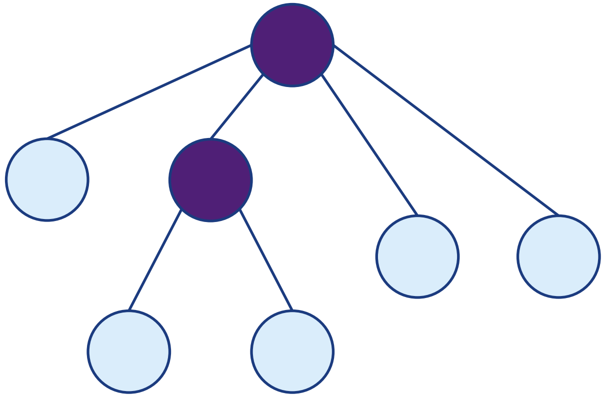
$(\boldsymbol{\alpha}, \boldsymbol{\beta})$**-approximation**

$G' \in \mathcal{C}_\lambda$



$\boldsymbol{\alpha}$**-approx.** min number of edits X

$\boldsymbol{\beta}$**-approx.** on $\mathcal{C}_\lambda$
e.g. $\lambda \leq \beta \cdot \lambda^*$



G'

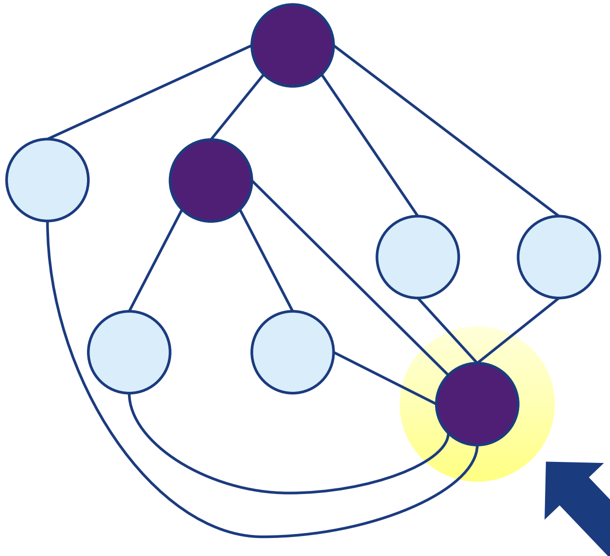
Solve the problem in the edited graph G': $OPT(G')$ where we assume $OPT(G') \leq OPT(G)$

Structural Rounding

(Minimization) Problem $\mathcal{P}$ with optimal solution $OPT(G)$

Framework for approximating solutions in close to structured graphs

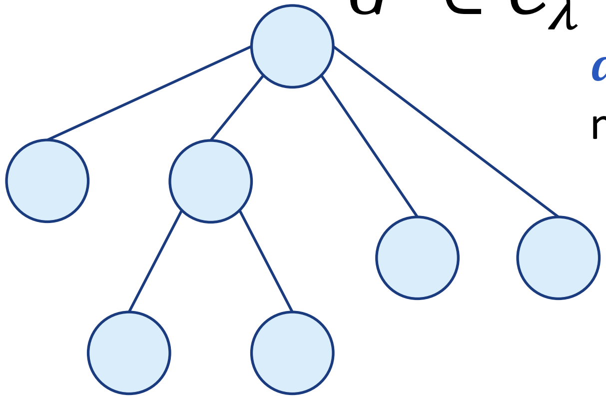
G



Edit to a Structured Graph Class

$(\boldsymbol{\alpha}, \boldsymbol{\beta})$**-approximation**

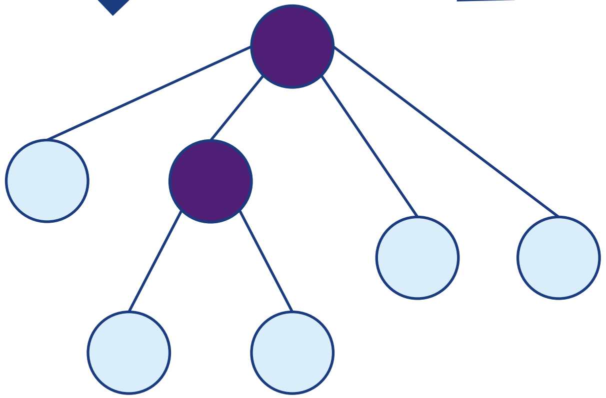
$G' \in \mathcal{C}_\lambda$



$\boldsymbol{\alpha}$**-approx.** min number of edits X

$\boldsymbol{\beta}$**-approx.** on $\mathcal{C}_\lambda$ e.g. $\lambda \leq \beta \cdot \lambda^*$

Solve the problem in the edited graph G': $OPT(G')$ where we assume $OPT(G') \leq OPT(G)$



G'

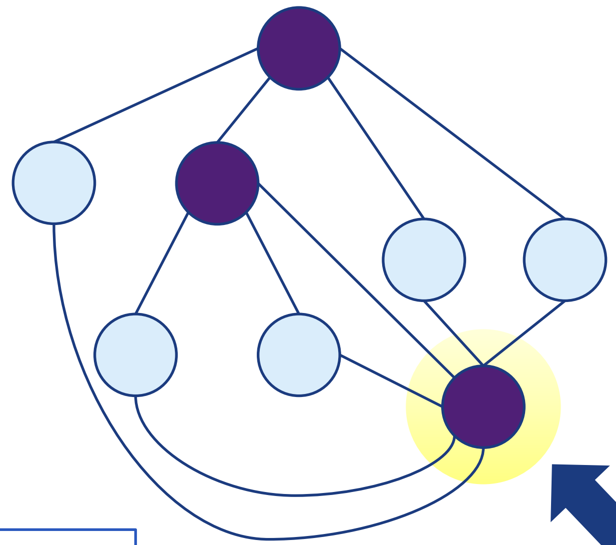
Lift the solution to the original graph with **loss** $\boldsymbol{c}$ i.e. need to add $c \cdot |X|$ to the solution

Structural Rounding

(Minimization) Problem $\mathcal{P}$ with optimal solution $OPT(G)$

Framework for approximating solutions in close to structured graphs

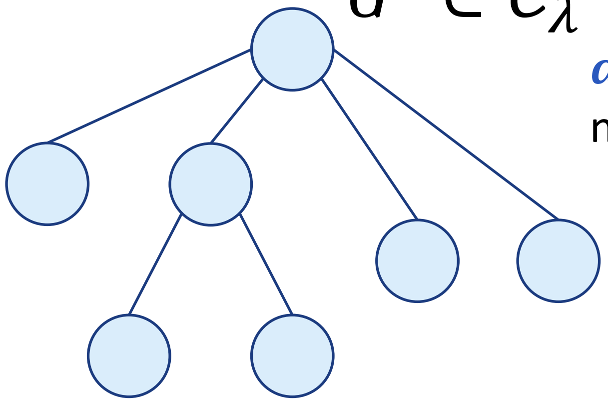
G



Edit to a Structured Graph Class

(α, β)-approximation

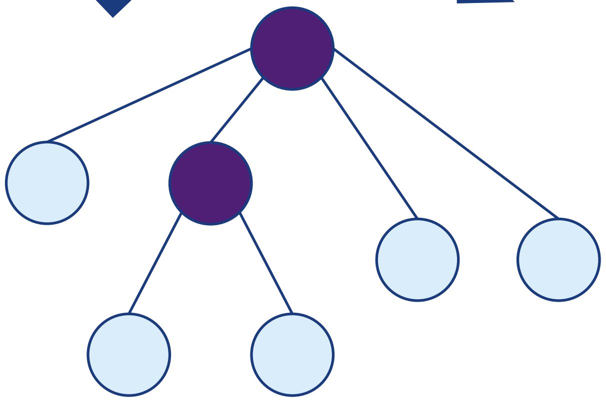
$G' \in \mathcal{C}_\lambda$



α-approx. min number of edits X

β-approx. on $\mathcal{C}_\lambda$ e.g. $\lambda \leq \beta \cdot \lambda^*$

Solve the problem in the edited graph G': $OPT(G')$ where we assume $OPT(G') \leq OPT(G)$



G'

Lift the solution to the original graph with **loss c** i.e. need to add $c \cdot |X|$ to the solution

$S \leq OPT(G') + c \cdot \alpha \cdot X$

$(1 + \epsilon)$-approx. when $\gamma \leq \frac{\mathrm{OPT(G)}}{\mathrm{c} \cdot \alpha \cdot \epsilon}$

Properties of Problems

- Stability
  - A graph minimization problem (resp. maximization) is **stable** if $OPT_{\Pi}(G') \leq OPT_{\Pi}(G) + c'\gamma$ (resp. $OPT_{\Pi}(G') \geq OPT_{\Pi}(G) - c'\gamma$)
- Structural Liftability
  - A graph minimization problem (resp. maximization) can be **structurally lifted** if given γ edits for a graph G that transforms it into graph G', we can convert a solution S' in G' into a solution S in G with cost $Cost_{\Pi}(S) \leq Cost_{\Pi}(S') + c \cdot \gamma$ (resp. $Cost_{\Pi}(S) \geq Cost_{\Pi}(S') - c \cdot \gamma$)

Editing Algorithms: Degeneracy and Treewidth

Editing to Degeneracy r

- LP-based bicriteria (6, 6)-approximation for vertex edits
- $O(r \log n)$-approximation based on a simple greedy algorithm and using the bicriteria constant approx. to edit to degen. r

(6, 6)-Approximation Vertex Editing to Degeneracy r

- LP-relaxation for minimizing number of required vertex deletions to family of r-degenerate graphs

<u>DegenVertexEdit-LP</u>

Input: $G = (V, E), w, r$

Minimize $\sum_{v \in V} y_v \, w_v$

s.t. $x_{\overrightarrow{vu}} + x_{\overrightarrow{uv}} \geq 1 - y_u - y_v \quad \forall uv \in E.$

$\sum_{u \in N(v)} x_{\overrightarrow{vu}} \leq r \quad \forall v \in V.$

$x_{\overrightarrow{uv}} \geq 0 \quad \forall uv \in E.$

(6, 6)-Approximation Vertex Editing to Degeneracy r

- LP-relaxation for minimizing number of required vertex deletions to family of r-degenerate graphs

$y_v = 1$ if $v \in X$

<u>DegenVertexEdit-LP</u>

Input: $G = (V, E), w, r$

$$\text{Minimize} \quad \sum_{v \in V} \boxed{y_v} w_v$$

$$\text{s.t.} \quad x_{\overrightarrow{vu}} + x_{\overrightarrow{uv}} \geq 1 - \boxed{y_u} - \boxed{y_v} \qquad \forall uv \in E.$$

$$\sum_{u \in N(v)} x_{\overrightarrow{vu}} \leq r \qquad \forall v \in V.$$

$$x_{\overrightarrow{uv}} \geq 0 \qquad \forall uv \in E.$$

(6, 6)-Approximation Vertex Editing to Degeneracy r

- LP-relaxation for minimizing number of required vertex deletions to family of r-degenerate graphs

$x_{vu} = 1$ and $x_{uv} = 0$ if edge oriented from v to u

<u>DegenVertexEdit-LP</u>

Input: $G = (V, E), w, r$

$$\text{Minimize} \quad \sum_{v \in V} \boxed{y_v} w_v$$

$$\text{s.t.} \quad \boxed{x_{\overrightarrow{vu}}} + \boxed{x_{\overrightarrow{uv}}} \geq 1 - \boxed{y_u} - \boxed{y_v} \qquad \forall uv \in E.$$

$$\sum_{u \in N(v)} \boxed{x_{\overrightarrow{vu}}} \leq r \qquad \forall v \in V.$$

$$\boxed{x_{\overrightarrow{uv}}} \geq 0 \qquad \forall uv \in E.$$

$y_v = 1$ if $v \in X$

(6, 6)-Approximation Vertex Editing to Degeneracy r

- LP-relaxation for minimizing number of required vertex deletions to family of r-degenerate graphs

$x_{vu} = 1$ and $x_{uv} = 0$ if edge oriented from v to u

<u>DegenVertexEdit-LP</u>

Input: $G = (V, E), w, r$

$$\text{Minimize} \quad \sum_{v \in V} \boxed{y_v} w_v$$

$$\text{s.t.} \quad x_{\overrightarrow{vu}} + x_{\overrightarrow{uv}} \geq 1 - \boxed{y_u} - \boxed{y_v} \qquad \forall uv \in E.$$

$$\sum_{u \in N(v)} x_{\overrightarrow{vu}} \leq r \qquad \forall v \in V.$$

$$x_{\overrightarrow{uv}} \geq 0 \qquad \forall uv \in E.$$

$y_v = 1$ if $v \in X$

Check degeneracy less than r

(6, 6)-Approximation Vertex Editing to Degeneracy r

- LP-relaxation for minimizing number of required vertex deletions to family of r-degenerate graphs

$x_{vu} = 1$ and $x_{uv} = 0$ if edge oriented from v to u

Minimize weight of vertices removed; $w_v = 1$ if unweighted

<u>DegenVertexEdit-LP</u>

Input: $G = (V, E), w, r$

$$\text{Minimize} \quad \sum_{v \in V} y_v w_v$$

$$\text{s.t.} \quad x_{\overrightarrow{vu}} + x_{\overrightarrow{uv}} \geq 1 - y_u - y_v \qquad \forall uv \in E.$$

$$\sum_{u \in N(v)} x_{\overrightarrow{vu}} \leq r \qquad \forall v \in V.$$

$$x_{\overrightarrow{uv}} \geq 0 \qquad \forall uv \in E.$$

$y_v = 1$ if $v \in X$

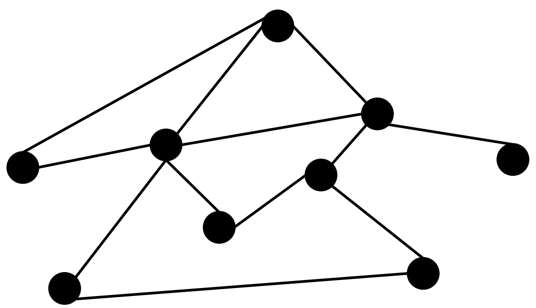
Check degeneracy less than r

$O(r \log n)$-approximation to Degen. r

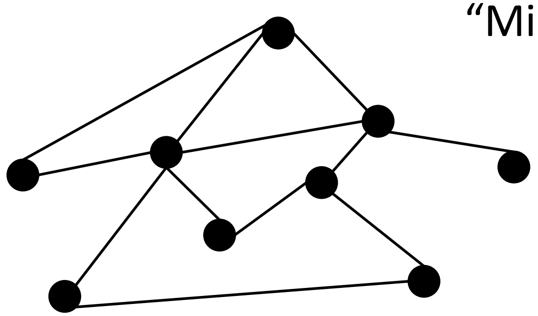
Matula and Beck ’83: Degeneracy Algorithm Produces Min-Degree Removal Ordering

- Sort vertices by degree, breaking ties arbitrarily.
- Remove the vertex with the smallest degree. Let d be this degree.
- Re-sort the vertices by degree and continue to remove the vertex with the smallest degree. Update d to indicate the vertex with the largest removed degree.
- Perform the above procedure until all vertices are removed.
- d is the degeneracy after all vertices have been removed.

$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy

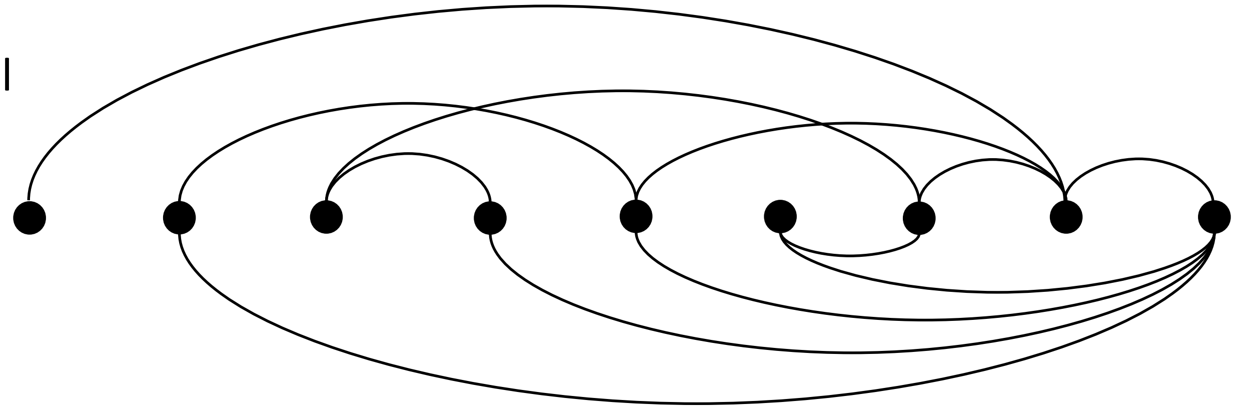


$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy

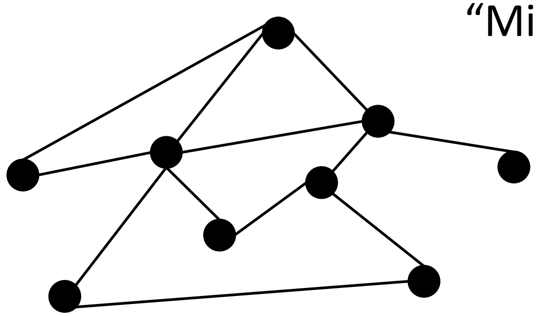


"Min-degree" removal ordering

Edit from degeneracy $r + 1$ to r

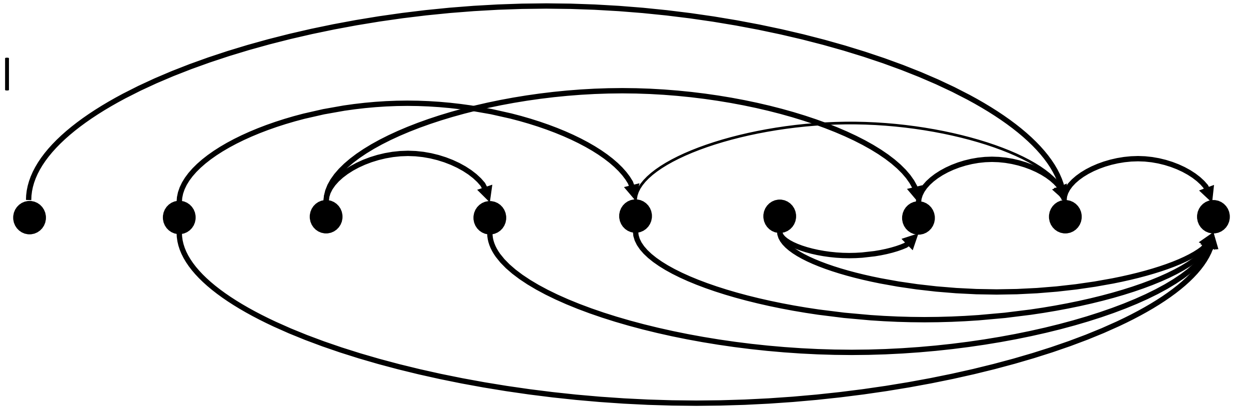


$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy

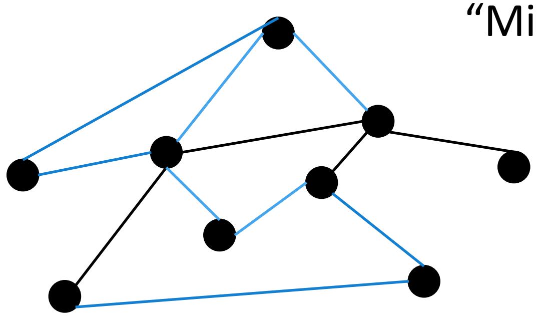


"Min-degree" removal ordering

Edit from degeneracy $r + 1$ to r

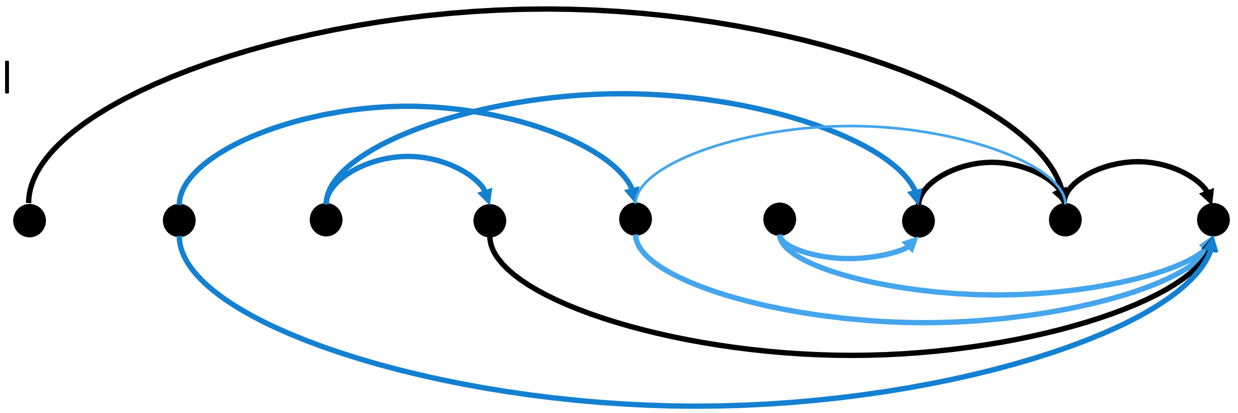


$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy

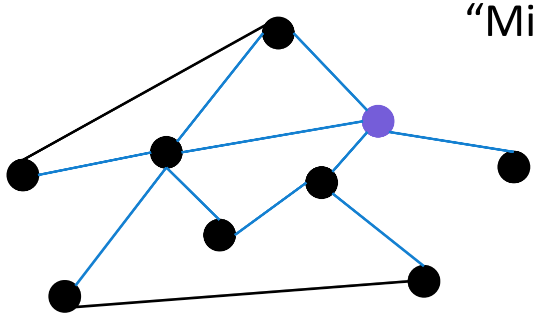


“Min-degree” removal ordering

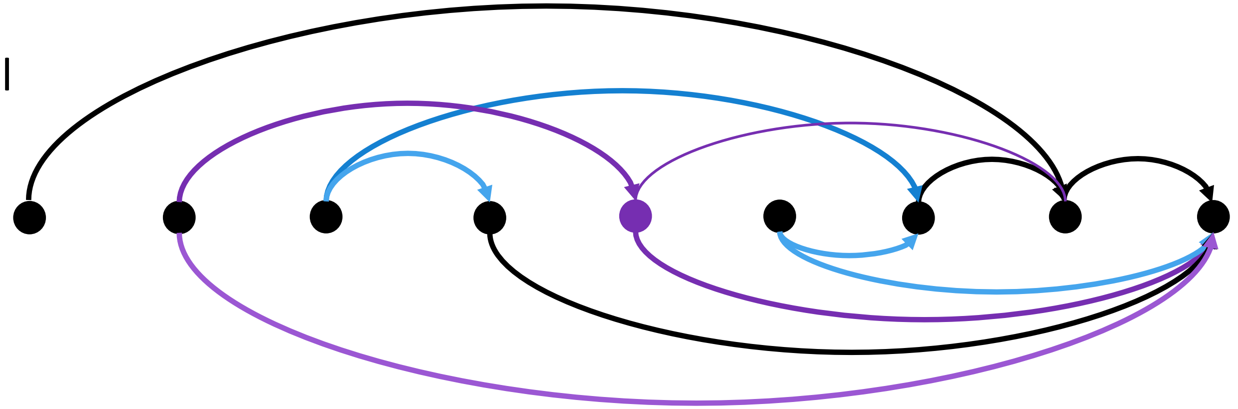
Edit from degeneracy $r+1$ to r



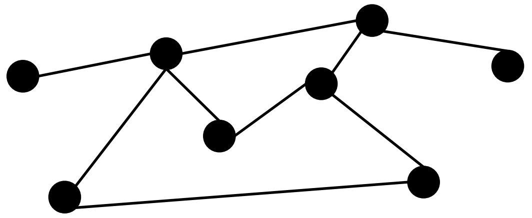
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



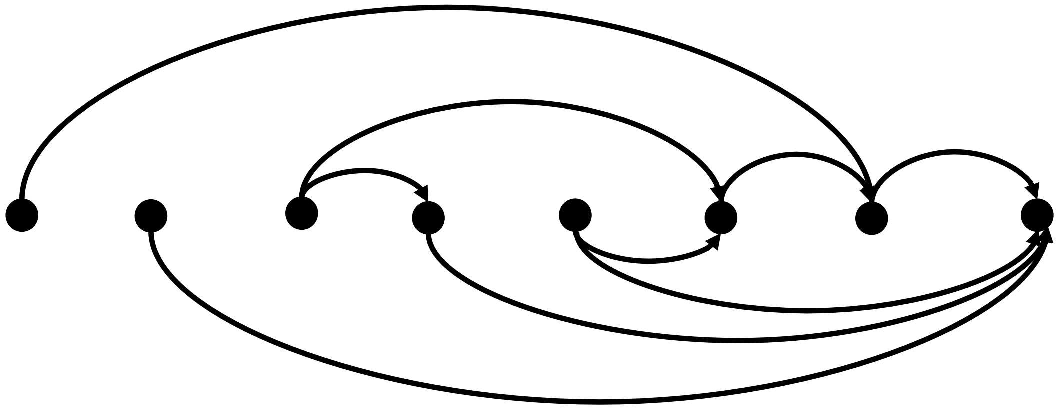
“Min-degree” removal ordering



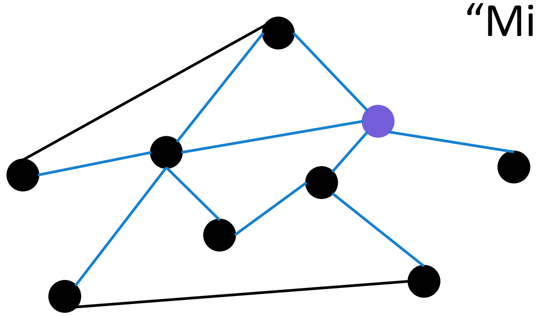
Edit from degeneracy $r + 1$ to r



New Min-degree removal ordering

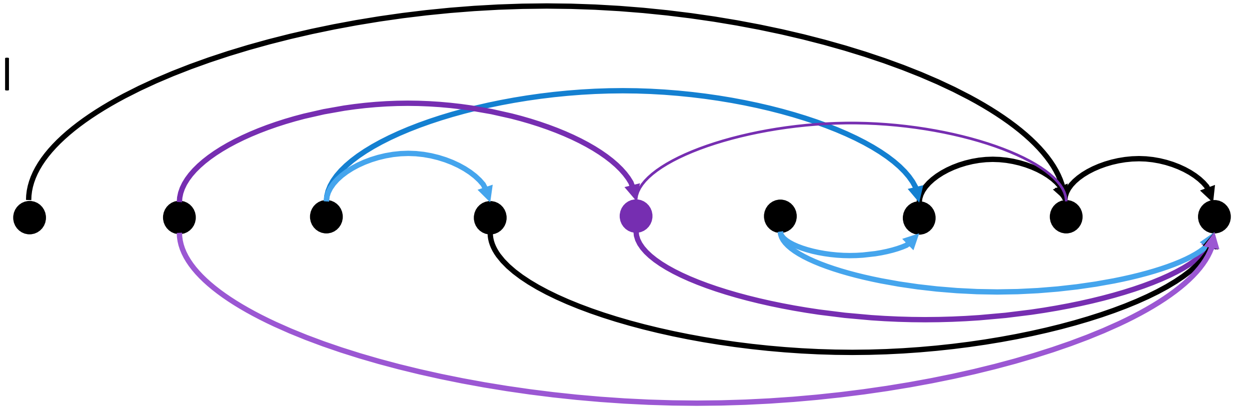


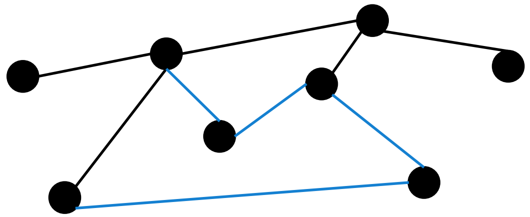
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



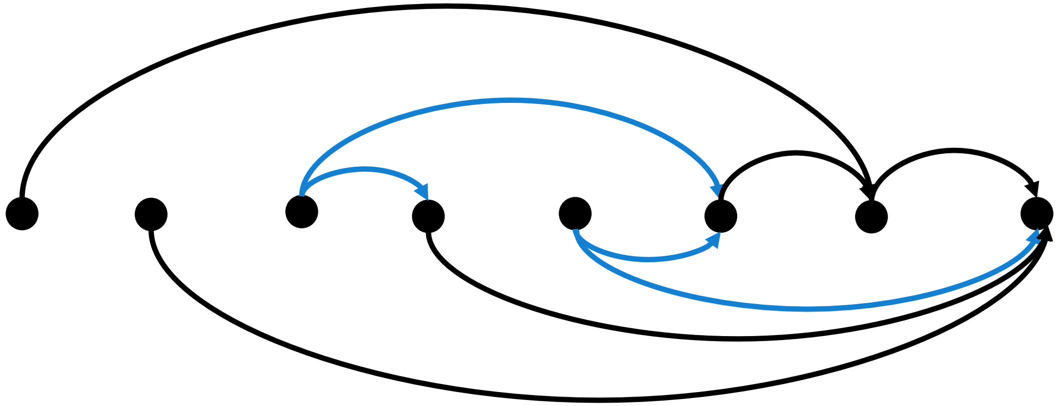
“Min-degree” removal ordering

Edit from degeneracy $r+1$ to r

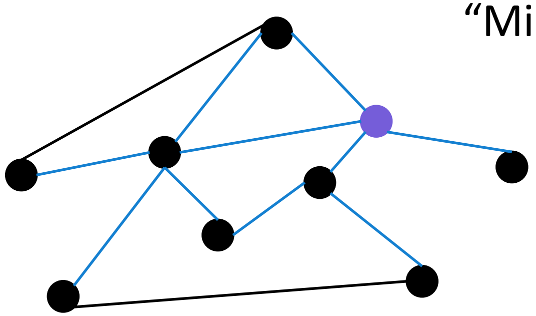




New Min-degree removal ordering

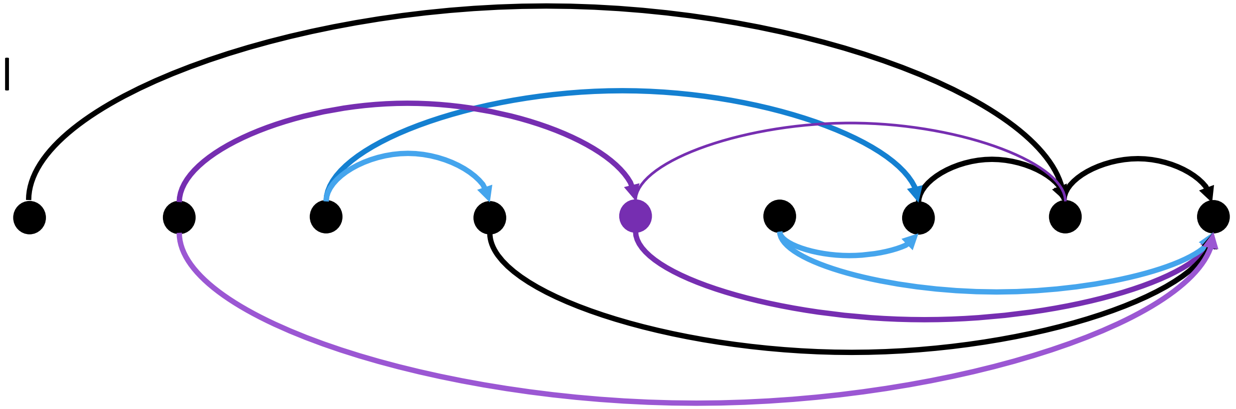


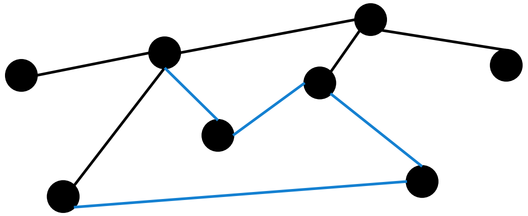
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



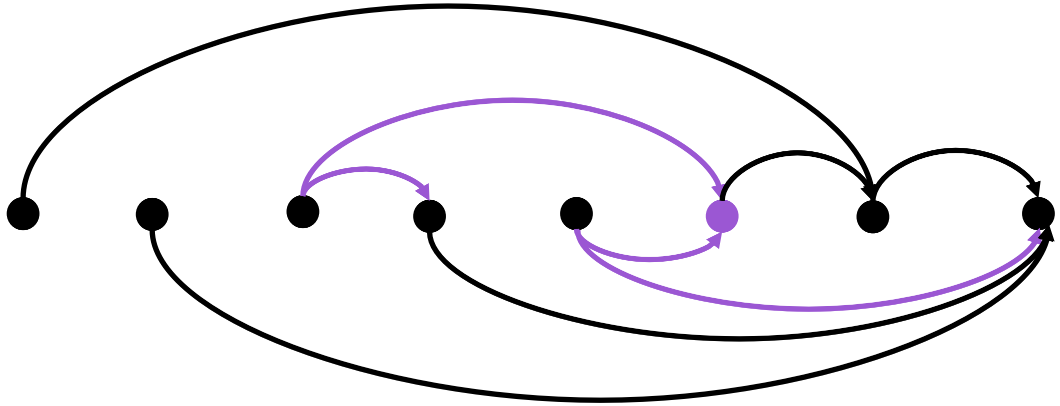
"Min-degree" removal ordering

Edit from degeneracy $r+1$ to r

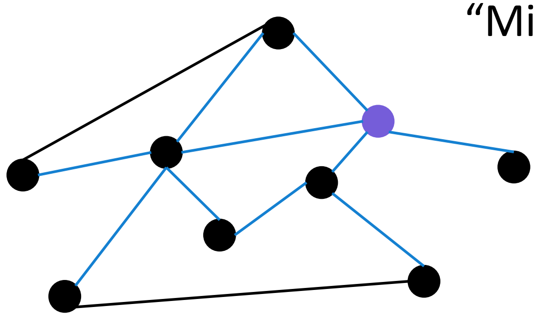




New Min-degree removal ordering

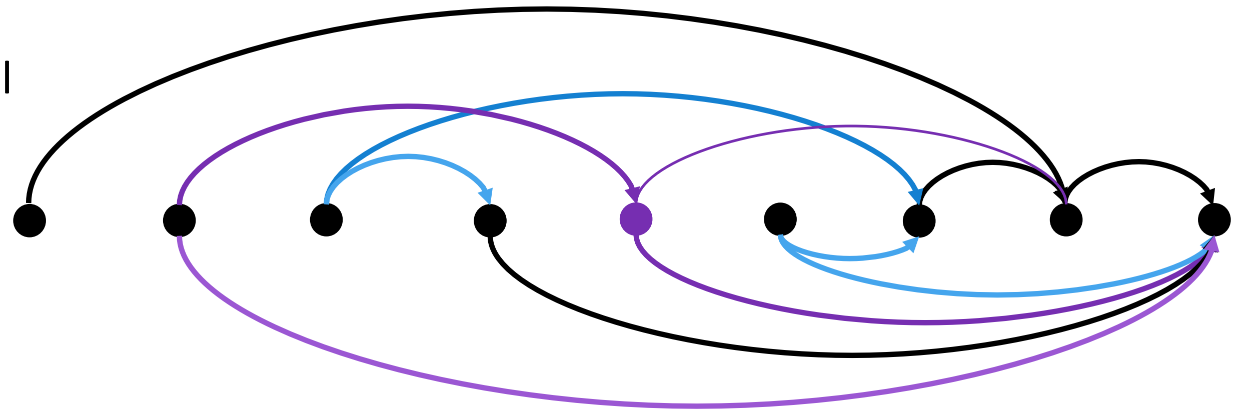


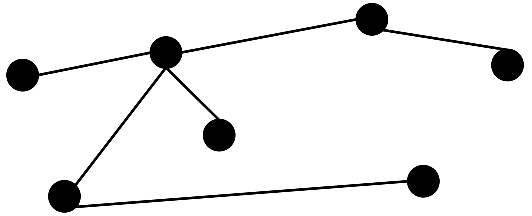
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



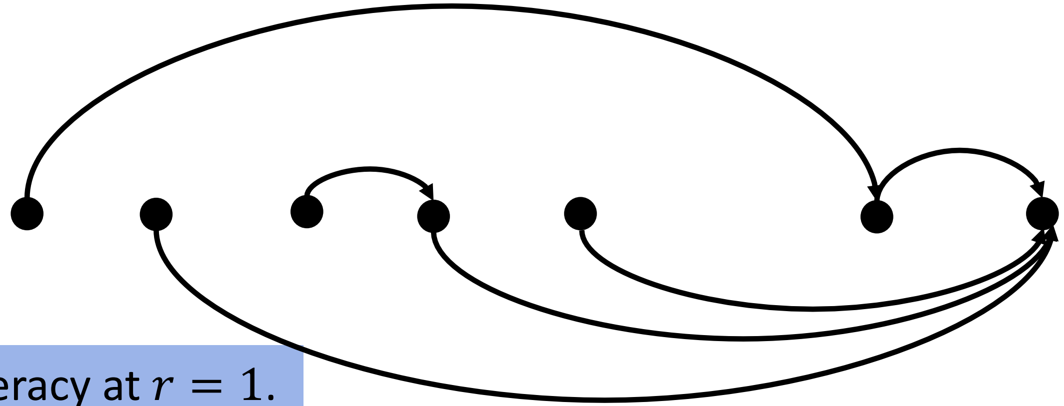
“Min-degree” removal ordering

Edit from degeneracy $r + 1$ to r



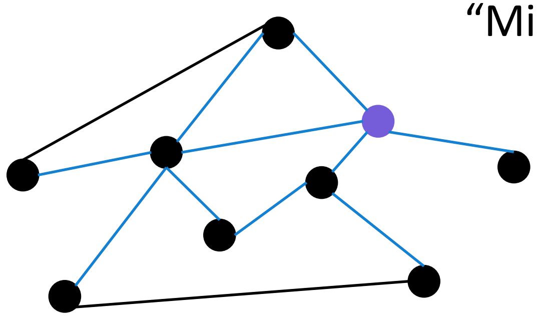


New Min-degree removal ordering



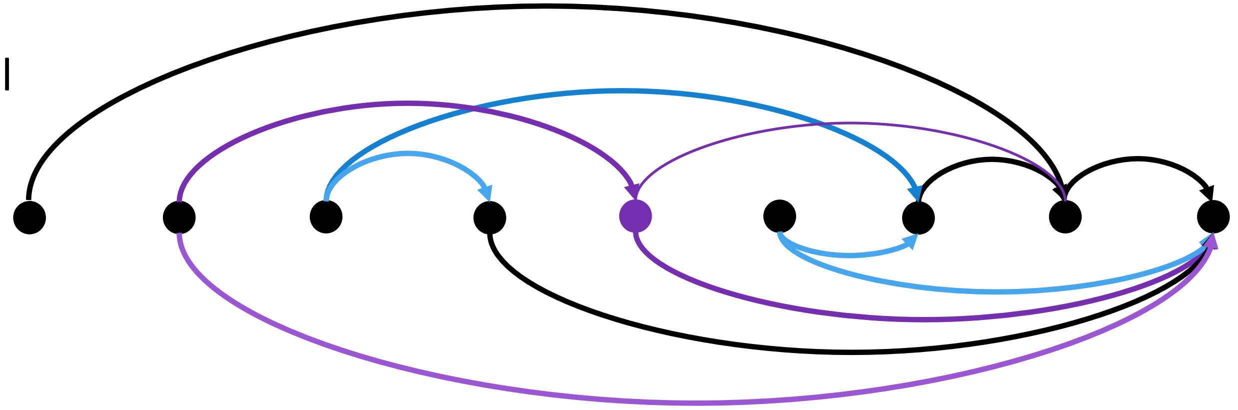
Correct Degeneracy at $r = 1$.

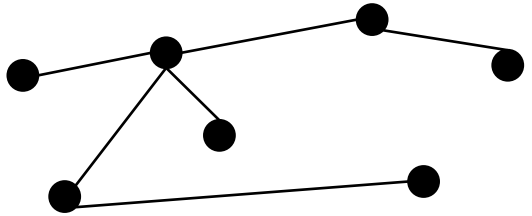
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



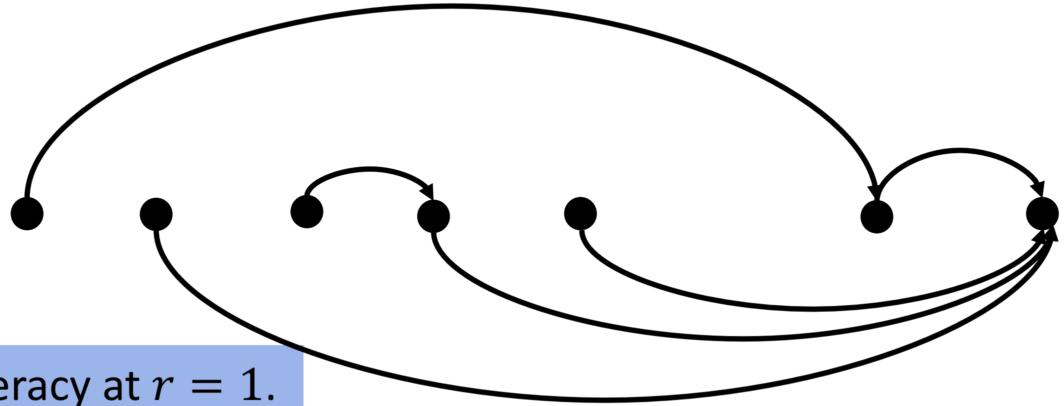
"Min-degree" removal ordering

Edit from degeneracy $r + 1$ to r





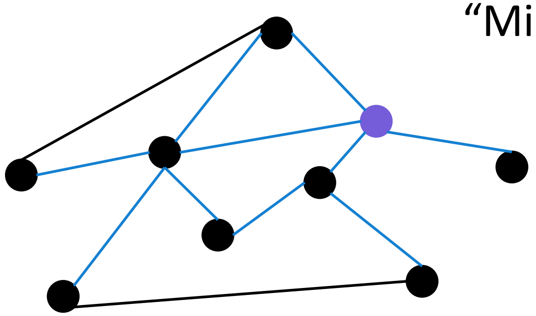
New Min-degree removal ordering



Correct Degeneracy at $r = 1$.

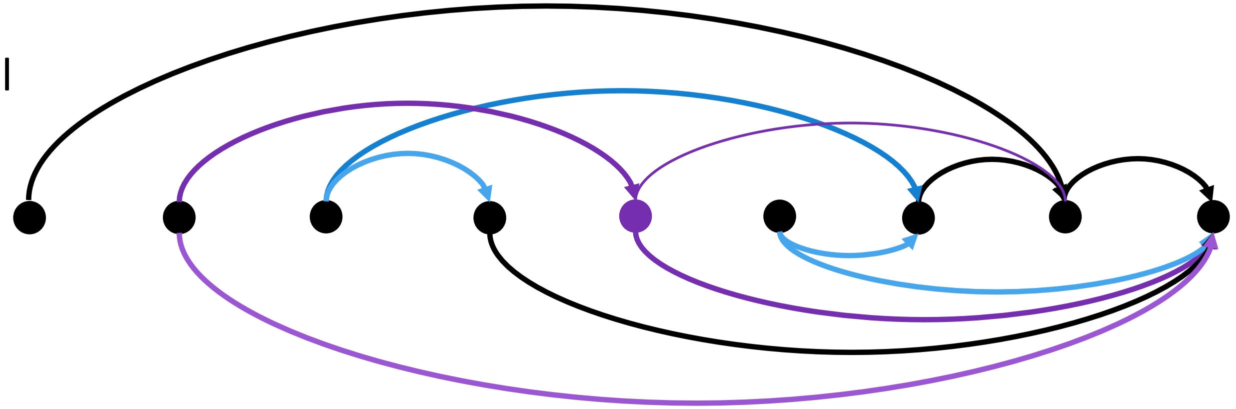
$O\big((r_1 - r_2) \log n\big)$-approximation

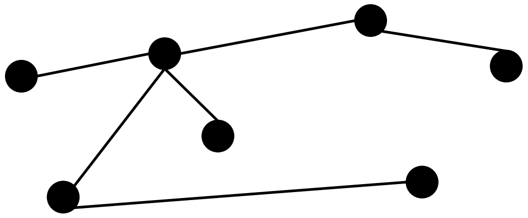
$O(r \log n)$-approximation for Vertex Editing to r-Degeneracy



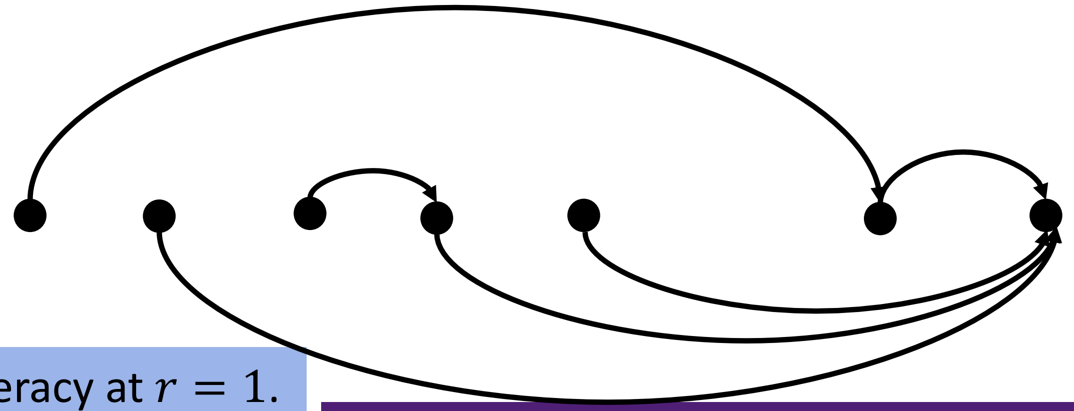
"Min-degree" removal ordering

Edit from degeneracy $r + 1$ to r





New Min-degree removal ordering



Correct Degeneracy at $r = 1$.

$O\big((r_1 - r_2) \log n\big)$-approximation

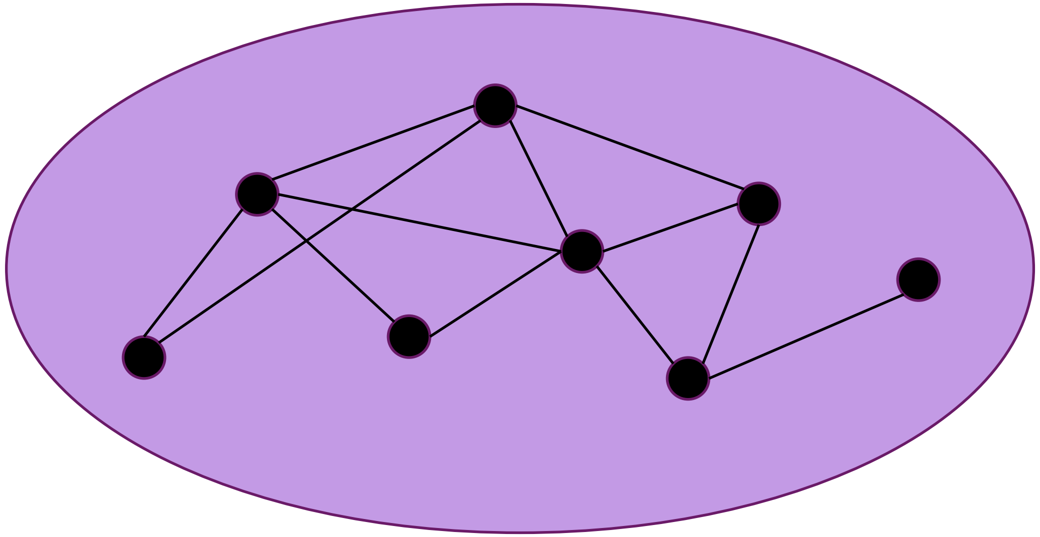
Combined with bicriteria result obtain $O(r \log n)$-approximation

$\left(O(\log^{1.5} n), O\left(\sqrt{\log w}\right)\right)$-approximation editing to Treewidth w

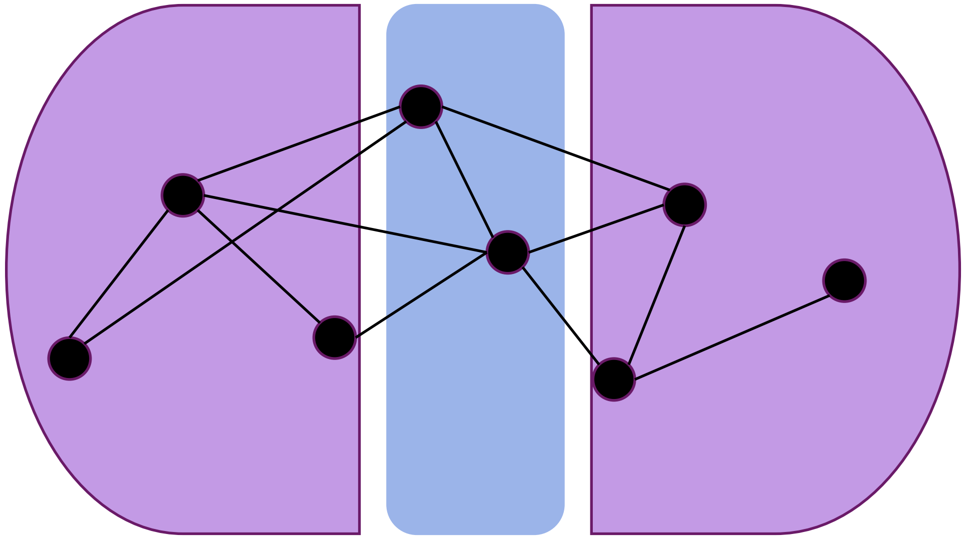
Editing to Bounded Treewidth

- Bicriteria approximation on the number of edits and the treewidth
- Uses (3/4)-*vertex separators* algorithm by Feige et al. [FHL08] which provides $O\left(\sqrt{\log w}\right)$-approximation
- Combined with ideas from Bodlaendar et al. [BGHK95] $O(\log n)$-approx. for treewidth
- The treewidth of a graph is always upper bounded by $O(|S|)$ where S is a minimum vertex separator

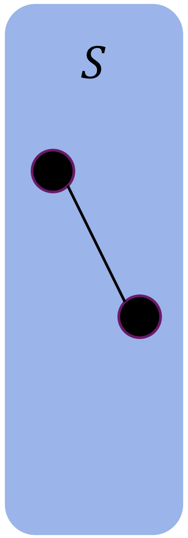
Editing to Bounded Treewidth



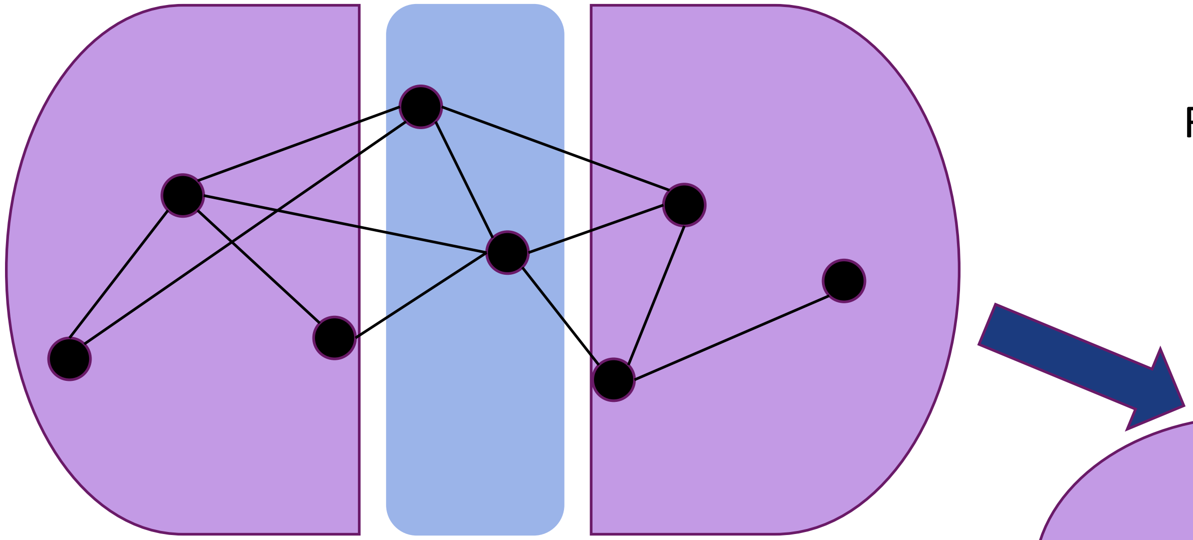
Editing to Bounded Treewidth



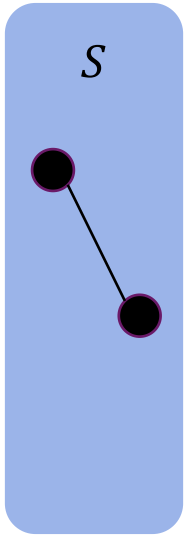
Find a $\left(\frac{3}{4}\right)$-vertex separator and remove the separator, keep separator vertices in S.

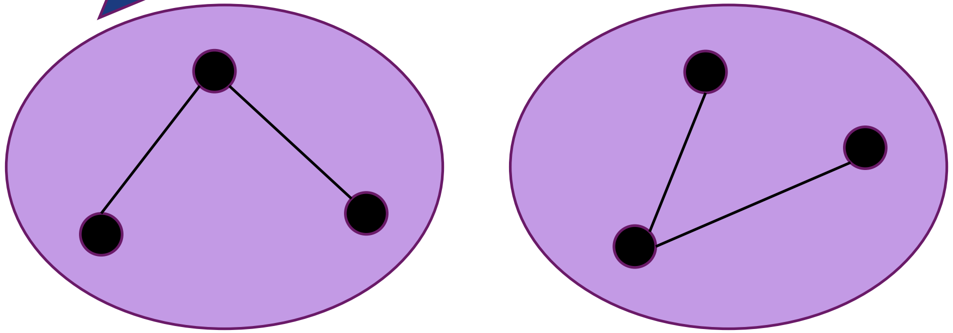


Editing to Bounded Treewidth



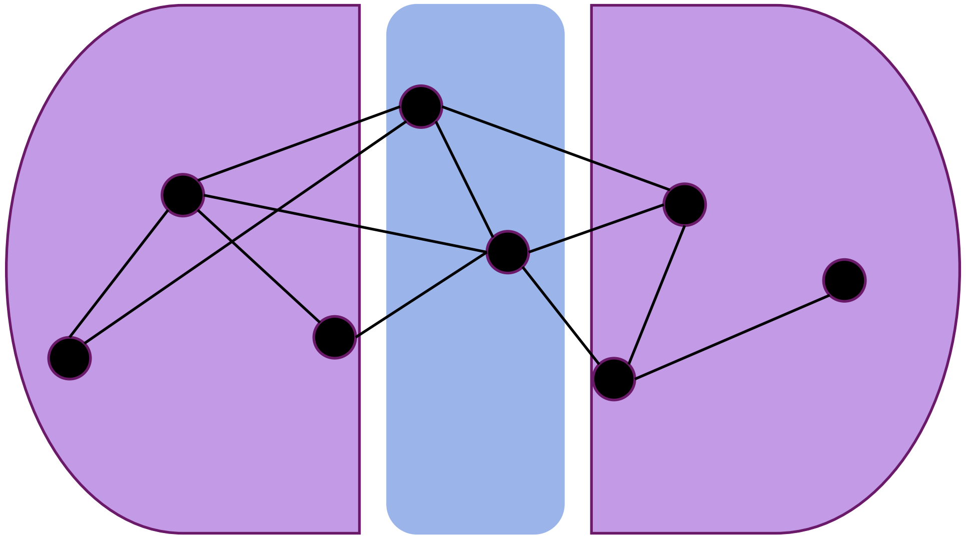
Find a $\left(\frac{3}{4}\right)$-vertex separator and remove the separator, keep separator vertices in S.



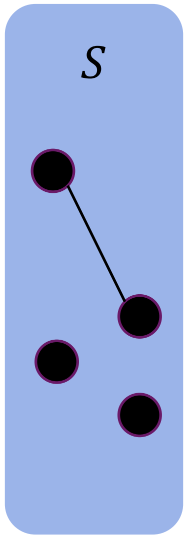


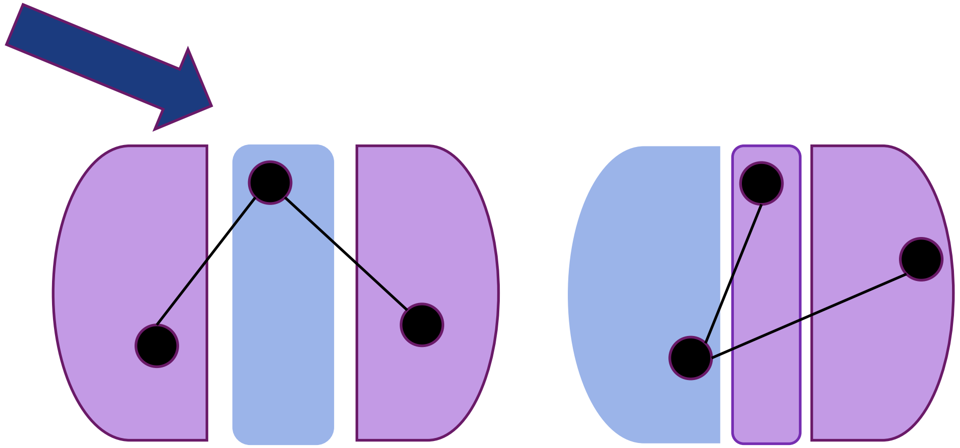
Recursively perform the same process on the subgraphs

Editing to Bounded Treewidth



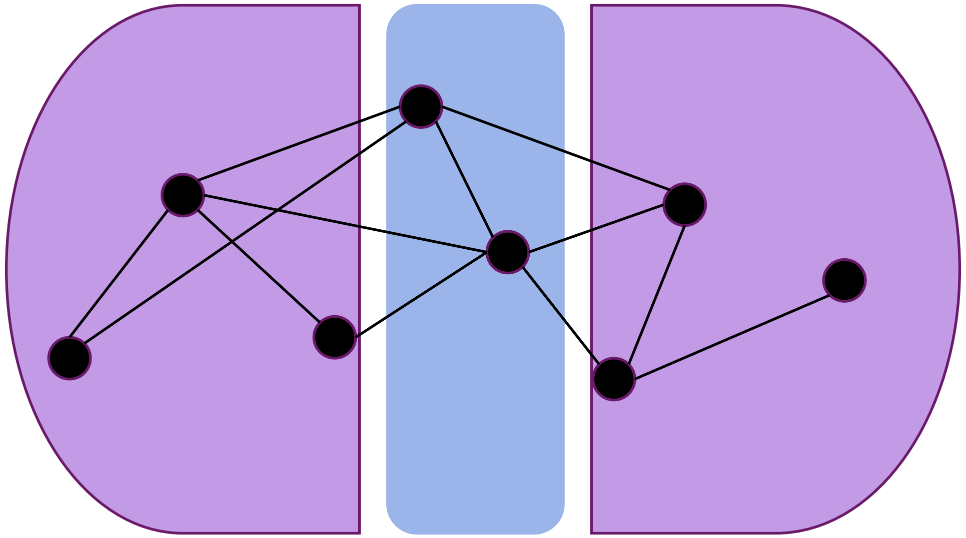
Find a $\left(\frac{3}{4}\right)$-vertex separator and remove the separator, keep separator vertices in S.



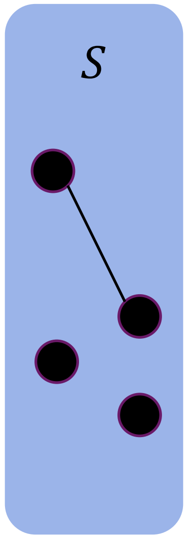


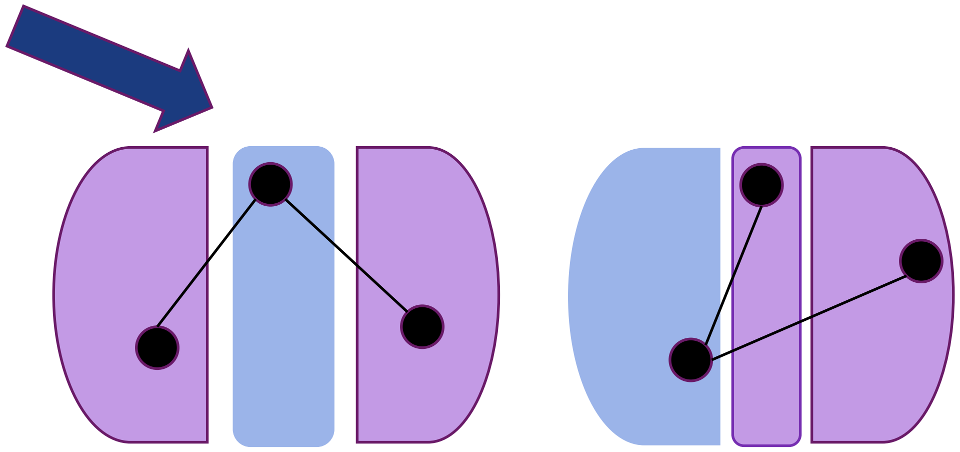
Recursively perform the same process on the subgraphs

Editing to Bounded Treewidth

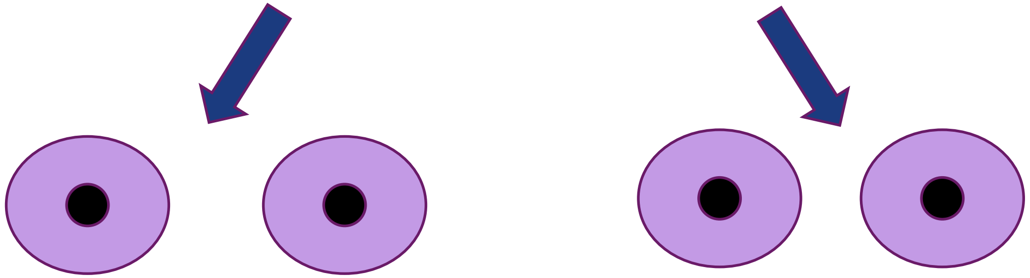


Find a $\left(\frac{3}{4}\right)$-vertex separator and remove the separator, keep separator vertices in S.



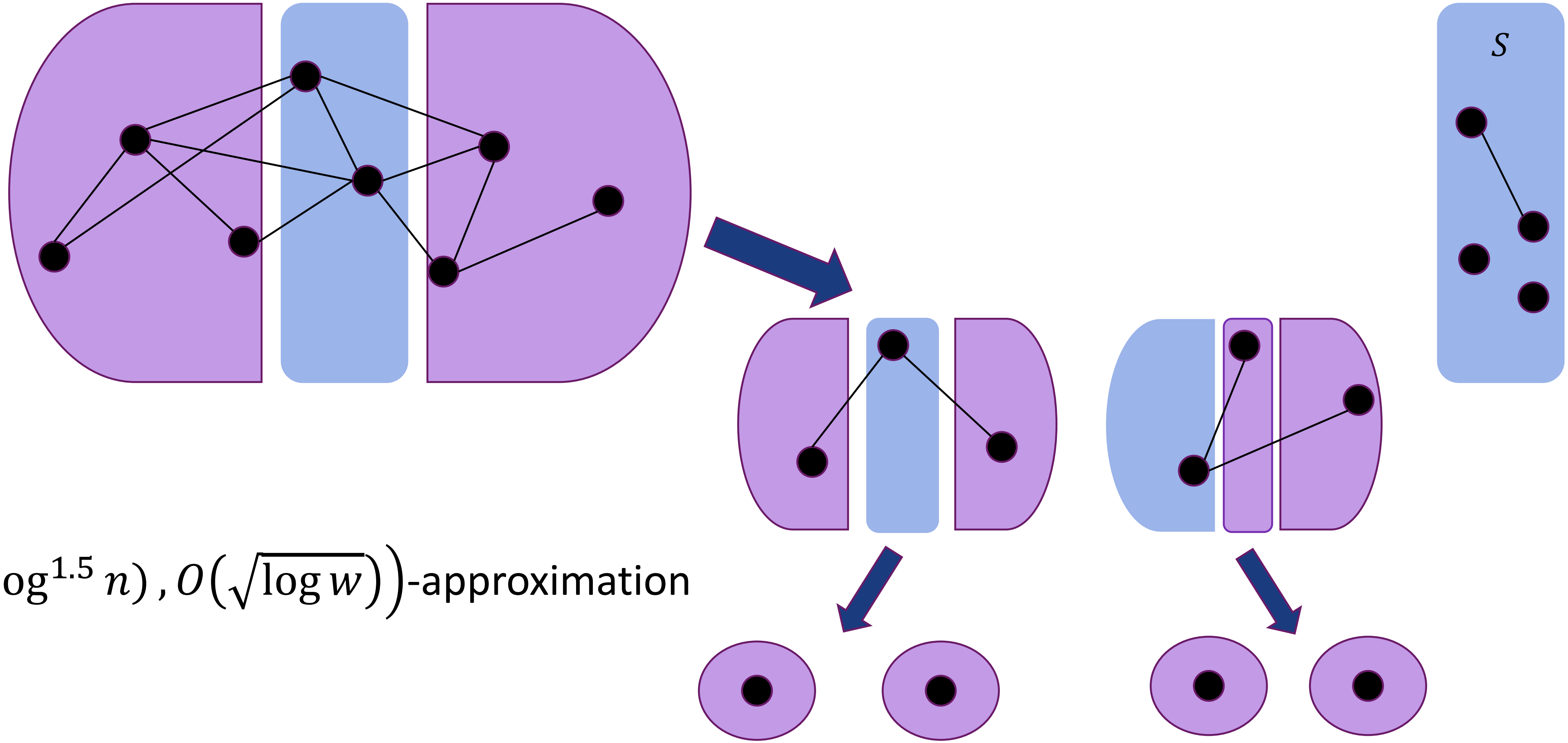


Recursively perform the same process on the subgraphs



Until resulting subgraphs have desired treewidth w

Editing to Bounded Treewidth



$\left(O(\log^{1.5} n), O\left(\sqrt{\log w}\right)\right)$-approximation

Our New Work

- Work with Daniel Lokshtanov combines [GLLMW18] with [FLS18] to obtain PTASes for graphs $O(OPT(G))$ away from $\mathcal{H}$-minor-free

- This supersedes our treewidth results for certain cases

- Note: Applying Gupta et al. [GLLMW18] directly for the treewidth editing only gets you to the class of $\left((\log_2 n)^{\frac{1}{3}}\right)$-treewidth graphs for PTASes so slightly smaller structured treewidth class than what our structural rounding algorithm achieves ($\left(\frac{\log_2 n}{\log \log n}\right)$-treewidth graphs in our case)
  - But using their algorithm we get a better δ for δ-close to the target treewidth

Open Problems

- More problems that can benefit from structural rounding
- Better approximation for editing to r-degeneracy
- Better bicriteria approximation algorithms for bounded treewidth editing
  - Might be as difficult as approximating treewidth
- Extend structural rounding to account for more noise (beyond only $O(\mathrm{OPT}(G))$ noise)
- Extend framework to other types of edits (e.g. edge insertions, edge contraction…etc.)