Twin-width

Yan Garito

Carleton University yangarito@cmail.carleton.ca

Context

Problem: Many problems are NP-hard on general graphs:

Carleton University yangarito@cmail.carleton.ca

Context

Problem: Many problems are NP-hard on general graphs:

- k-Independent graph;

Carleton University yangarito@cmail.carleton.ca

Context

Problem: Many problems are NP-hard on general graphs:

- k-Independent graph;
- k-Dominating Set;

Context

Problem: Many problems are NP-hard on general graphs:

- k-Independent graph;
- k-Dominating Set;
- k-Coloring;

Context

Problem: Many problems are NP-hard on general graphs:

- k-Independent graph;
- k-Dominating Set;
- k-Coloring;
- The list goes on...

Solution: FPT algorithms

Carleton University yangarito@cmail.carleton.ca

Context

Problem: Many problems are NP-hard on general graphs:

- k-Independent graph;
- k-Dominating Set;
- k-Coloring;
- The list goes on...

Solution: FPT algorithms...but what parameter do we fix?

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)
  - Useful even on dense graphs

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)
  - Useful even on dense graphs
  - Unknown whether we can recognize graphs with clique-width k for constant k

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)
  - Useful even on dense graphs
  - Unknown whether we can recognize graphs with clique-width k for constant k
- **Twin-width** (Bonnet, Kim, et al. 2021)

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)
  - Useful even on dense graphs
  - Unknown whether we can recognize graphs with clique-width k for constant k
- **Twin-width** (Bonnet, Kim, et al. 2021)
  - Useful even on dense graphs

Carleton University yangarito@cmail.carleton.ca

Some parameters to fix

- Treewidth (Bertele and Brioschi 1972; Halin 1976; Robertson and Seymour 1984)
  - Linked with tree decompositions
  - Only useful for sparse graphs
  - Constant k: can recognize graphs with treewidth k in polynomial time (Arnborg, Corneil, and Proskurowski 1987)
- Clique-width (Courcelle, Engelfriet, and Rozenberg 1993)
  - Useful even on dense graphs
  - Unknown whether we can recognize graphs with clique-width k for constant k
- **Twin-width** (Bonnet, Kim, et al. 2021)
  - Useful even on dense graphs
  - Can mostly be computed (NP-hard still) : SAT solvers (Bergé, Bonnet, and Déprés 2022; Schidler and Szeider 2022)

Carleton University yangarito@cmail.carleton.ca

What is twin-width?

Definition (Trigraph)

Triple $G = (V, E, R)$:

- V vertex set;
- $E \subseteq V \times V$ **black** edge set
- $R \subseteq V \times V$ **red** edge set
- $E \cap R = \varnothing$

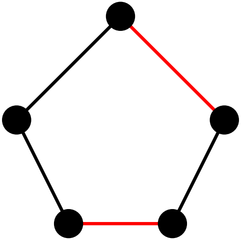
Carleton University yangarito@cmail.carleton.ca

What is twin-width?

Definition (Trigraph)

Triple $G = (V, E, R)$:

- V vertex set;
- $E \subseteq V \times V$ **black** edge set
- $R \subseteq V \times V$ **red** edge set
- $E \cap R = \varnothing$



What is twin-width?

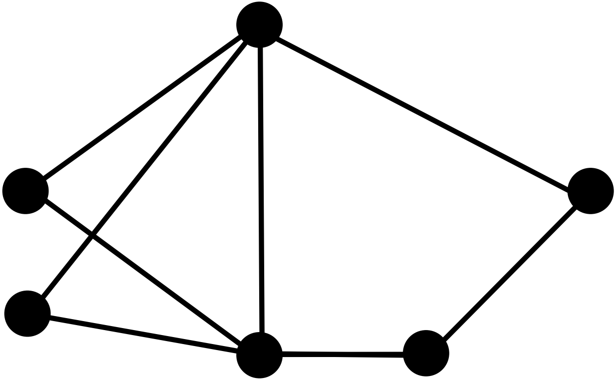
Definition (Contraction sequence)

G (tri)graph on n vertices. Contraction sequence $G = G_n, G_{n-1}, ..., G_2, G_1 = K_1$ if for all $i \in \{1, 2, ..., n-1\}$, G_{n-i} is obtained from G_{n-i+1} by merging two vertices u and v into a vertex uv as follows:

- If u and v both have a black edge with w, then uv has a black edge with w
- If neither u nor v have an edge with w, then uv has no edge with w
- Otherwise, uv has a red edge with w

What is twin-width?

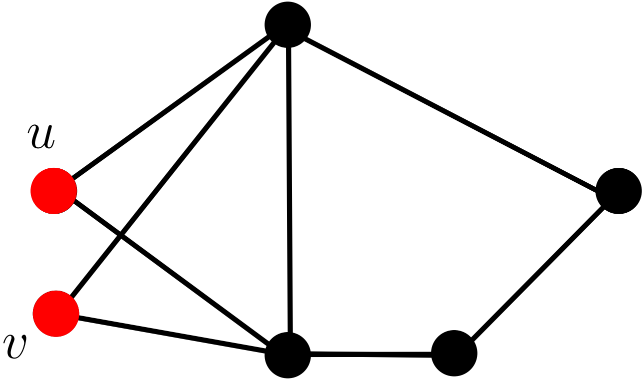
uw and vw black $\Rightarrow (uv)w$ black
uw and vw missing $\Rightarrow (uv)w$ missing
otherwise $(uv)w$ red



Carleton University yangarito@cmail.carleton.ca

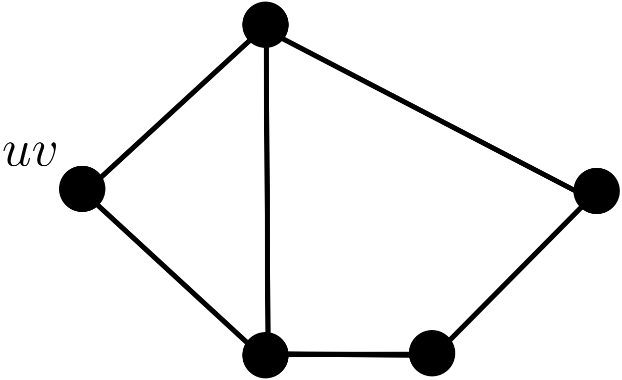
What is twin-width?

uw and *vw* black ⇒ (*uv*)*w* black
uw and *vw* missing ⇒ (*uv*)*w* missing
otherwise (*uv*)*w* red



What is twin-width?

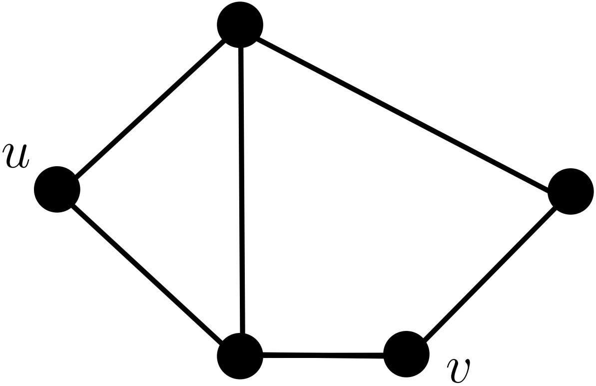
uw and vw black $\Rightarrow$ $(uv)w$ black
uw and vw missing $\Rightarrow$ $(uv)w$ missing
otherwise $(uv)w$ red



Carleton University yangarito@cmail.carleton.ca

What is twin-width?

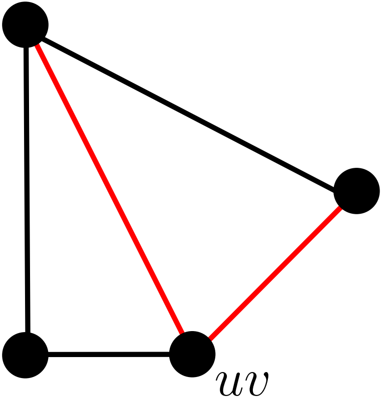
uw and vw black $\Rightarrow (uv)w$ black
uw and vw missing $\Rightarrow (uv)w$ missing
otherwise $(uv)w$ red



Carleton University yangarito@cmail.carleton.ca

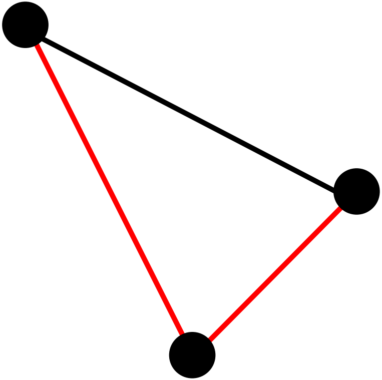
What is twin-width?

uw and vw black $\Rightarrow$ $(uv)w$ black
uw and vw missing $\Rightarrow$ $(uv)w$ missing
otherwise $(uv)w$ red



What is twin-width?

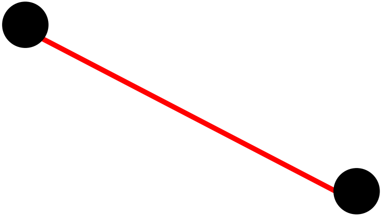
uw and vw black $\Rightarrow$ $(uv)w$ black
uw and vw missing $\Rightarrow$ $(uv)w$ missing
otherwise $(uv)w$ red



Carleton University yangarito@cmail.carleton.ca

What is twin-width?

uw and vw black $\Rightarrow (uv)w$ black
uw and vw missing $\Rightarrow (uv)w$ missing
otherwise $(uv)w$ red



Carleton University yangarito@cmail.carleton.ca

What is twin-width?

uw and vw black $\Rightarrow$ $(uv)w$ black
uw and vw missing $\Rightarrow$ $(uv)w$ missing
otherwise $(uv)w$ red

What is twin-width?

Definition (Twin-width)

Let $G = (V, E)$. Denote by $\Delta_r(H)$ the maximum red degree of a trigraph H.

$$\text{tww}(G) = \min_{G_n,\ldots,G_1 \text{ contraction sequence of } G} \{ \max_{i=1,\ldots,n} \{\Delta_r(G_i)\}\}$$

Carleton University yangarito@cmail.carleton.ca

What is twin-width?

Definition (Twin-width)

Let $G = (V, E)$. Denote by $\Delta_r(H)$ the maximum red degree of a trigraph H.

$$\text{tww}(G) = \min_{G_n,\ldots,G_1 \text{ contraction sequence of } G} \{ \max_{i=1,\ldots,n} \{\Delta_r(G_i)\}\}$$

Definition ($u(G)$)

If $u \in G_i$, a trigraph in a contraction sequence, then $u(G) =$ the set of vertices of G that were contracted to form u

k-Independent Set

k-Independent Set

Definition (Independent Set)

An independent set I in a graph $G = (V, E)$ is a set such that $I \subseteq V$ and for any $u, v \in I, uv \notin E$. We denote by $\alpha(G)$ the size of the largest independent set of G

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Definition (Independent Set)

An independent set I in a graph $G = (V, E)$ is a set such that $I \subseteq V$ and for any $u, v \in I, uv \notin E$. We denote by $\alpha(G)$ the size of the largest independent set of G

Problem (k-Independent Set)

Input: *a graph* $G = (V, E)$*, an integer* $k \geq 1$
Output: *yes if there exists an independent set* $I \subseteq V$ *with* $|I| \geq k$ *and no otherwise*

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Problem (k-Independent Set)

Input: *a graph $G = (V, E)$, an integer $k \geq 1$*
Output: *yes if there exists an independent set $I \subseteq V$ with $|I| \geq k$ and no otherwise*

How does twin-width help?

k-Independent Set

Idea: Use the contraction sequence to progressively build up partial solutions (Bonnet, Geniet, et al. 2021)

k-Independent Set

Idea: Use the contraction sequence to progressively build up partial solutions (Bonnet, Geniet, et al. 2021)

Definition (Partial solution)

A partial solution in the trigraph G_i is a pair (T, S):

- $T \subseteq V(G_i)$ is a connected subset of vertices in the red graph of G_i
- $S \subseteq V(G)$ is an independent set of G such that $S \subseteq \cup_{u \in T} u(G)$ and for every $u \in T, S \cap u(G) \neq \varnothing$

k-Independent Set

Idea: Use the contraction sequence to progressively build up partial solutions (Bonnet, Geniet, et al. 2021)

Definition (Partial solution)

A partial solution in the trigraph G_i is a pair (T, S):

- $T \subseteq V(G_i)$ is a connected subset of vertices in the red graph of G_i
- $S \subseteq V(G)$ is an independent set of G such that $S \subseteq \cup_{u \in T} u(G)$ and for every $u \in T, S \cap u(G) \neq \varnothing$

Definition (Compatible partial solutions)

(T, S) and (T', S') are compatible if there is no edge (no matter the color) uu' in G_i such that $u \in T$ and $u' \in T'$

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Idea: Use the contraction sequence to progressively build up partial solutions (Bonnet, Geniet, et al. 2021)

Definition (Compatible partial solutions)

(T, S) and (T', S') are compatible if there is no edge (no matter the color) uu' in G_i such that $u \in T$ and $u' \in T'$

Definition (Union of compatible partial solutions)

$(T, S) \cup (T', S') = (T \cup T', S \cup S')$

Remark

The union of two compatible partial solutions is not a partial solution

k-Independent Set

Definition (Realizable set)

A set $T \subseteq V(G_i)$ is realizable (in G_i) if there exists a set $S \subseteq V(G)$ such that (T, S) is a partial solution in G_i

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Definition (Realizable set)

A set $T \subseteq V(G_i)$ is realizable (in G_i) if there exists a set $S \subseteq V(G)$ such that (T, S) is a partial solution in G_i

Definition (dec)

Let $X \subseteq V(G_i)$. The set $\text{dec}(X)$ is the set with one partial solution in G_i for each connected component of X in its red graph. If one connected component is not realizable, then $\text{dec}(X) = \text{None}$.
$\cup\,\text{dec}(X) = \text{None}$ if two of the partial solutions in X are not compatible or if $\text{dec}(X) = \text{None}$

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$
- Let $u, v \in G_{i+1}$ be contracted to $z \in G_i$. Initialize $\mathcal{S}_i$ as all the solutions in $\mathcal{S}_{i+1}$ not intersecting $\{u, v\}$

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$
- Let $u, v \in G_{i+1}$ be contracted to $z \in G_i$. Initialize $\mathcal{S}_i$ as all the solutions in $\mathcal{S}_{i+1}$ not intersecting $\{u, v\}$
- Consider all realizable sets $T \ni z$ in G_i

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$
- Let $u, v \in G_{i+1}$ be contracted to $z \in G_i$. Initialize $\mathcal{S}_i$ as all the solutions in $\mathcal{S}_{i+1}$ not intersecting $\{u, v\}$
- Consider all realizable sets $T \ni z$ in G_i
    - For each such T, add the partial solution with largest S from among $\cup\text{dec}(T\backslash\{z\} \cup \{u, v\}), \cup\text{dec}(T\backslash\{z\} \cup \{u\})$ and $\text{dec}(T\backslash\{z\} \cup \{v\})$ to the set of partial solutions $\mathcal{S}_i$

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$
- Let $u, v \in G_{i+1}$ be contracted to $z \in G_i$. Initialize $\mathcal{S}_i$ as all the solutions in $\mathcal{S}_{i+1}$ not intersecting $\{u, v\}$
- Consider all realizable sets $T \ni z$ in G_i
  - For each such T, add the partial solution with largest S from among $\cup\,\text{dec}(T \backslash \{z\} \cup \{u, v\}), \cup\,\text{dec}(T \backslash \{z\} \cup \{u\})$ and $\text{dec}(T \backslash \{z\} \cup \{v\})$ to the set of partial solutions $\mathcal{S}_i$
  - Before adding a partial solution to $\mathcal{S}_i$, we check if its S has size $\geq k$. If so, we return yes

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Algorithm: Input: $G = (V, E)$ a graph, $k \geq 1$ an integer and a contraction sequence $(G_n, ..., G_1)$ of G

- Initialize the partial solutions $\mathcal{S}_n = \{(\{v\}, \{v\}), v \in V(G)\}$
- Let $u, v \in G_{i+1}$ be contracted to $z \in G_i$. Initialize $\mathcal{S}_i$ as all the solutions in $\mathcal{S}_{i+1}$ not intersecting $\{u, v\}$
- Consider all realizable sets $T \ni z$ in G_i
  - For each such T, add the partial solution with largest S from among $\cup \text{dec}(T \backslash \{z\} \cup \{u, v\}), \cup \text{dec}(T \backslash \{z\} \cup \{u\})$ and $\text{dec}(T \backslash \{z\} \cup \{v\})$ to the set of partial solutions $\mathcal{S}_i$
  - Before adding a partial solution to $\mathcal{S}_i$, we check if its S has size $\geq k$. If so, we return yes
- If we build $\mathcal{S}_1$ and still have not found a solution whose S has size $\geq k$, we return no

k-Independent Set

Theorem

This algorithm is correct

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Theorem

This algorithm is correct

- By induction, we show that for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ such that either $|S| = \alpha(G[\cup_{u \in T} u(G)])$ or $|S| \geq k$. We will always assume the former, as the algorithm terminates in the latter case

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:
  - $z \notin T$: Then we inherit S from $\mathcal{S}_{i+1}$ and by the induction hypothesis, it satisfies $|S| = \alpha(G[\cup_{u \in T} u(G)])$

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:
  - $z \in T$:
    - Let S' be a maximum independent set of $G[\cup_{u \in T} u(G)]$.

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:
  - $z \in T$:
    - Let S' be a maximum independent set of $G[\cup_{u \in T} u(G)]$.
    - Build $\varnothing \neq I \subseteq \{u, v\}$ by adding u (resp. v) to I if $S' \cap u(G) \neq \varnothing$ (resp. $S' \cap v(G) \neq \varnothing$)

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:
  - $z \in T$:
    - Let S' be a maximum independent set of $G[\cup_{u \in T} u(G)]$.
    - Build $\varnothing \neq I \subseteq \{u, v\}$ by adding u (resp. v) to I if $S' \cap u(G) \neq \varnothing$ (resp. $S' \cap v(G) \neq \varnothing$)
    - By the existence of S', the connected components of $T \backslash \{z\} \cup I$ are all realizable

Carleton University yangarito@cmail.carleton.ca

k-Independent Set

Theorem

This algorithm is correct

- IH: for every realizable set $T \subseteq V(G_i)$, we have $(T, S) \in \mathcal{S}_i$ with $|S| = \alpha(G[\cup_{u \in T} u(G)])$
- Let u, v be contracted into z from G_{i+1} to G_i
- Let T be realizable in G_i. There are two cases:
  - $z \in T$:
    - Let S' be a maximum independent set of $G[\cup_{u \in T} u(G)]$.
    - Build $\varnothing \neq I \subseteq \{u, v\}$ by adding u (resp. v) to I if $S' \cap u(G) \neq \varnothing$ (resp. $S' \cap v(G) \neq \varnothing$)
    - By the existence of S', the connected components of $T \backslash \{z\} \cup I$ are all realizable
    - By the induction hypothesis, dec$(T \backslash \{z\} \cup I)$ contains optimal partial solutions, so its union is optimal

k-Independent Set

Theorem

This algorithm runs in time $O(k^2 d^{2k} n)$ where d is the maximum red degree of a graph in the contraction sequence

k-Independent Set

Theorem

This algorithm runs in time $O(k^2 d^{2k} n)$ where d is the maximum red degree of a graph in the contraction sequence

- Enumerating the T's takes time $O(d^{2k})$
- Checking compatibility of partial solutions takes time $O(k^2)$

k-Dominating Set

k-Dominating Set

Definition (Dominating Set)

A dominating set D in a graph $G = (V, E)$ is a set such that $D \subseteq V$ and for any $u \in V$, either $u \in D$ or there is $v \in D$ such that $uv \in E$

k-Dominating Set

Definition (Dominating Set)

A dominating set D in a graph $G = (V, E)$ is a set such that $D \subseteq V$ and for any $u \in V$, either $u \in D$ or there is $v \in D$ such that $uv \in E$

Problem (k-Dominating Set)

Input: *a graph* $G = (V, E)$*, an integer* $k \geq 1$
Output: *yes if there exists a dominating set* $D \subseteq V$ *with* $|D| \leq k$ *and no otherwise*

Carleton University yangarito@cmail.carleton.ca

k-Dominating Set

(Bonnet, Geniet, et al. 2021)

Definition (Profile of a partial solution)

A profile of a partial solution in G_i is a triple (T, D, M) of subsets of $V(G_i)$:

- T forms a connected subset in the red graph of G_i
- $D, M \subseteq T$
- $\cup_{x \in D} B_i^2(x) \subseteq T$ where $B_i^2(x)$ is the ball of radius 2 centered at x in the red graph of G_i

A profile such that $|D| \leq k$ is called a k-profile

k-Dominating Set

Definition (Realizable profile)

(T, D, M) is realizable with $S \subseteq V(G)$ if:

1. $S \subseteq \cup_{x \in T} x(G)$
2. for all $x \in V(G_i)$, $x \in D$ iff $x(G) \cap S \neq \varnothing$
3. for all $x \in V(G_i)$, $x \in M$ iff $x(G)$ is dominated by S

Carleton University yangarito@cmail.carleton.ca

k-Dominating Set

Assume that u, v are contracted to z to form G_i from G_{i+1}

k-Dominating Set

Definition (Consistency of profiles)

A profile (T, D, M) such that $z \in T$ is consistent with a set $\{(T_1, D_1, M_1), ..., (T_\ell, D_\ell, M_\ell)\}$ if the following holds. Let $T' = T \backslash \{z\} \cup \{u, v\}, D' = \cup_{j=1}^{\ell} D_j$ and $M' = \cup_{j=1}^{\ell} M_j$.

1. $T_1, ..., T_\ell$ are the red components of T' in G_{i+1}
2. $D \backslash \{z\} = D' \backslash \{u, v\}$
3. $z \in D$ iff $u \in D'$ or $v \in D'$
4. For every $x \in T \backslash \{z\}$, $x \in M$ iff $x \in M'$ or there is $y \in D'$ such that xy is a black edge in G_{i+1}
5. $z \in M$ iff for each $x \in \{u, v\}$, either $x \in M'$ or there is $y \in D'$ such that xy is a black edge in G_{i+1}

Carleton University yangarito@cmail.carleton.ca

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, \ldots, G_1)$ of G with maximum red degree d

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Initialize a map τ_n that takes as input profiles (T, D, M) with $|T| < (d^2 + 1)k$ and outputs either None if the profile is not realizable, or outputs a set $S \subseteq_{x \in T} x(G)$ that realizes (T, D, M). Initially, $\tau_n(\{v\}, \{v\}, \{v\}) = \{v\}$, $\tau_n(\{v\}, \varnothing, \varnothing) = \varnothing$ and None on every other input.

Carleton University yangarito@cmail.carleton.ca

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \rightarrow G_i$ into z

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \rightarrow G_i$ into z
- Let (T, D, M) be a profile. There are two cases:

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \to G_i$ into z
- Let (T, D, M) be a profile. There are two cases:
  - $z \notin T$: then $\tau_i(T, D, M) = \tau_{i+1}(T, D, M)$

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \to G_i$ into z
- Let (T, D, M) be a profile. There are two cases:
  - $z \in T$: inspect every set $\mathcal{P}$ of k-profiles consistent with (T, D, M)

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \to G_i$ into z
- Let (T, D, M) be a profile. There are two cases:
  - $z \in T$: inspect every set $\mathcal{P}$ of k-profiles consistent with (T, D, M)
  - Put $\tau_i(T, D, M) =$ the best $\cup_{P \in \mathcal{P}} \tau_{i+1}(P)$ (the smallest one). If all these unions are None, $\tau_i(T, D, M) =$ None

k-Dominating Set

Algorithm: Input: a graph $G = (V, E)$, an integer $k \geq 1$ and a contraction sequence $(G_n, ..., G_1)$ of G with maximum red degree d

- Let u, v be the vertices contracted in $G_{i+1} \to G_i$ into z
- Let (T, D, M) be a profile. There are two cases:
  - $z \in T$: inspect every set $\mathcal{P}$ of k-profiles consistent with (T, D, M)
  - Put $\tau_i(T, D, M) =$ the best $\cup_{P \in \mathcal{P}} \tau_{i+1}(P)$ (the smallest one). If all these unions are None, $\tau_i(T, D, M) =$ None
  - If there is a $\mathcal{P}$ with a profile (T, D, M) with $|T| \geq (d^2 + 1)k$, since τ_{i+1} is undefined on this profile, we instead pick $v \in T \backslash (\cup_{x \in D} B^2_{i+1}(x))$ and make the query at $(T \backslash \{v\}, D \backslash \{v\}, M \backslash \{v\})$

Carleton University yangarito@cmail.carleton.ca

k-Dominating Set

Theorem

This algorithm is correct

k-Dominating Set

Theorem

This algorithm is correct

Theorem

This algorithm runs in time $O(d^{2(d^2+1)k-2}2^{2(d^2+1)k})$ *where* d *is the maximum red degree of the trigraphs in the contraction sequence*

Carleton University yangarito@cmail.carleton.ca

Conclusion

Thank you for listening!

Carleton University yangarito@cmail.carleton.ca

Bibliography I

Arnborg, Stefan, Derek G. Corneil, and Andrzej Proskurowski (1987). "Complexity of Finding Embeddings in a k-Tree." In: *SIAM Journal on Algebraic Discrete Methods* 8.2, pp. 277–284. DOI: 10.1137/0608024. eprint: https://doi.org/10.1137/0608024. URL: https://doi.org/10.1137/0608024.

Bibliography II

Bergé, Pierre, Édouard Bonnet, and Hugues Déprés (2022). "Deciding Twin-Width at Most 4 Is NP-Complete." In: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Ed. by Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff. Vol. 229. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 18:1–18:20. ISBN: 978-3-95977-235-8. DOI: 10.4230/LIPIcs.ICALP.2022.18. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2022.18.

Bertele, Umberto and Francesco Brioschi (1972). *Nonserial Dynamic Programming*. USA: Academic Press, Inc. ISBN: 0120934507.

Carleton University yangarito@cmail.carleton.ca

Bibliography III

- Bonnet, Édouard, Colin Geniet, et al. (2021). *Twin-width III: Max Independent Set, Min Dominating Set, and Coloring*. arXiv: 2007.14161 [cs.DS]. URL: https://arxiv.org/abs/2007.14161.
- Bonnet, Édouard, Eun Jung Kim, et al. (Nov. 2021). "Twin-width I: Tractable FO Model Checking." In: *J. ACM* 69.1. ISSN: 0004-5411. DOI: 10.1145/3486655. URL: https://doi.org/10.1145/3486655.

Bibliography IV

- Courcelle, Bruno, Joost Engelfriet, and Grzegorz Rozenberg (1993). "Handle-rewriting hypergraph grammars." In: *Journal of Computer and System Sciences* 46.2, pp. 218–270. ISSN: 0022-0000. DOI: https://doi.org/10.1016/0022-0000(93)90004-G. URL: https://www.sciencedirect.com/science/article/pii/002200009390004G.
- Halin, Rudolf (1976). "S-functions for graphs." In: *Journal of Geometry* 8, pp. 171–186. URL: https://api.semanticscholar.org/CorpusID:120256194.

Carleton University yangarito@cmail.carleton.ca

Bibliography V

Robertson, Neil and P.D Seymour (1984). "Graph minors. III. Planar tree-width." In: *Journal of Combinatorial Theory, Series B* 36.1, pp. 49–64. ISSN: 0095-8956. DOI: https://doi.org/10.1016/0095-8956(84)90013-3. URL: https://www.sciencedirect.com/science/article/pii/0095895684900133.

Schidler, André and Stefan Szeider (2022). "A SAT Approach to Twin-Width." In: *2022 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pp. 67–77. DOI: 10.1137/1.9781611977042.6. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611977042.6. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611977042.6.