Treewidth (III)

Yijia Chen

Shanghai Jiao Tong University

May 3, 2012

Definition

Let $\mathcal{G}$ and $\mathcal{H}$ be two graphs. A function $\underline{f} : V(\mathcal{G}) \to V(\mathcal{H})$ is an isomorphism if

(GI1) f is a bijection;

(GI2) for every $u, v \in V(\mathcal{G})$ we have $\{u, v\} \in E(\mathcal{G})$ if and only if $\{f(u), f(v)\} \in E(\mathcal{H})$.

If such an f exists, then $\mathcal{G}$ and $\mathcal{H}$ are isomorphic.

Graph Isomorphism (GI) problem

GI	
Input:	Two graphs $\mathcal{G}$ and $\mathcal{H}$.
Problem:	Decides whether $\mathcal{G}$ and $\mathcal{H}$ are isomorphic.

Remark.

1. GI *is in* **NP**.
2. GI *is not* **NP***-complete, unless Polynomial Hierarchy collapses.*
3. *We don't know whether* GI *is* **P***-hard.*
4. *Some people believe* GI *is in* **P***, but we don't even have a quantum polynomial time algorithm.*

GI on bounded treewidth graphs

Theorem (Bodlaender, 1990)

Let $k \in \mathbb{N}$. Then there is a polynomial time algorithm which decides GI on graphs $\mathcal{G}$ with $\mathsf{tw}(\mathcal{G}) \leq k$.

I will present an algorithm deciding the problem

Input:	Two graphs $\mathcal{G}$ and $\mathcal{H}$ and a smooth tree decomposition of $\mathcal{G}$ of width k.
Problem:	Decides whether $\mathcal{G}$ and $\mathcal{H}$ are isomorphic.

in time

$$(|V(\mathcal{G})| + |V(\mathcal{H})|)^{O(k)}.$$

Isomorphism via connected components

Let $\mathscr{C}_{\mathcal{G}}$ be the set of connected components of $\mathcal{G}$ and $\mathscr{C}_{\mathcal{H}}$ the set of connected components of $\mathcal{H}$.

Then $\mathcal{G}$ and $\mathcal{H}$ are isomorphic if and only if there is a bijection $h : \mathscr{C}_{\mathcal{G}} \to \mathscr{C}_{\mathcal{H}}$ such that $\mathcal{G}[C]$ and $\mathcal{H}[h(C)]$ are isomorphic for every $C \in \mathscr{C}_{\mathcal{G}}$.

This is equivalent to that there is a perfect matching in the following bipartite graph.

1. The left part is $\mathscr{C}_{\mathcal{G}}$ and the right part $\mathscr{C}_{\mathcal{H}}$.
2. There is an edge between a $C \in \mathscr{C}_{\mathcal{G}}$ and a $C' \in \mathscr{C}_{\mathcal{H}}$ if $\mathcal{G}[C]$ and $\mathcal{H}[C']$ are isomorphic.

Isomorphism via separators

Let $S \subseteq V(\mathcal{G})$ and

$$\mathscr{C}_{\mathcal{G}\setminus S} := \{C \mid C \text{ a connected component of } \mathcal{G} \setminus S\}.$$

Then $\mathcal{G}$ and $\mathcal{H}$ are isomorphic if and only if there is a set $S' \subseteq V(\mathcal{H})$, a function $h : \mathscr{C}_{\mathcal{G}\setminus S} \to \mathscr{C}_{\mathcal{H}\setminus S'}$ and functions $f_C : S \cup C \to S' \cup h(C)$ for all $C \in \mathscr{C}_{\mathcal{G}\setminus S}$ such that

1. $|S| = |S'|$;
2. h is a bijection;
3. f_C is an isomorphism between $\mathcal{G}[S \cup C]$ and $\mathcal{H}[S' \cup h(C)]$ for every $C \in \mathscr{C}_{\mathcal{G}\setminus S}$, and $f_C(S) = S'$;
4. $f_{C_1} \restriction S = f_{C_2} \restriction S$ for every $C_1, C_2 \in \mathscr{C}_{\mathcal{G}\setminus S}$.

The sets $\mathscr{C}_t$

Let $\big(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})}\big)$ be a smooth tree decomposition of width k for the graph $\mathcal{G}$. Again we choose an arbitrary root r in $\mathcal{T}$.

For every $t \in V(\mathcal{T})$ we define

$$\underline{\mathscr{C}_t} := \left\{ C \;\middle|\; C = \emptyset \text{ or } C \text{ a connected component of } \mathcal{G}_{\leq t} \setminus B_t \right\}.$$

Connected components via tree decompositions (1)

Lemma

Every nonempty $C \in \mathscr{C}_t$, i.e., a connect component in $\mathcal{G}_{\leq t} \setminus B_t$, is a connected component of $\mathcal{G} \setminus B_t$.

Proof.

Clearly there is a connected component C' in $\mathcal{G} \setminus B_t$ with $C \subseteq C'$.

Assume that $C' \setminus C \neq \emptyset$. Then there is and edge $\{u, v\} \in E(\mathcal{G})$ with $u \in V(\mathcal{G}_{\leq t}) \setminus B_t$ and $v \in V(\mathcal{G}) \setminus V(\mathcal{G}_{\leq t})$.

But then, $\{u, v\}$ is not contained in any bag of the tree decomposition. $\square$

Lemma

Let t_1 be a child of t. Then for every nonempty $C_1 \in \mathscr{C}_{t_1}$ there is a unique $C \in \mathscr{C}_t$ with $C_1 \subseteq C$, and $C_1 \cap C' = \emptyset$ for all other $C' \in \mathscr{C}_t$.

Proof.

Let C_1 be a connected component of $\mathcal{G}_{\leq t_1} \setminus B_{t_1}$.

Observe that

$$\mathcal{G}_{\leq t_1} \setminus B_{t_1} \subseteq \mathcal{G}_{\leq t} \setminus B_t,$$

so C_1 is connected in $\mathcal{G}_{\leq t} \setminus B_t$, and the result follows. $\square$

Lemma

Let t be a node in $\mathcal{T}$ with children $t_1, \ldots, t_n$. And let $C \in \mathscr{C}_t$ be nonempty. Then, there is a unique $i \in [m]$ such that

$$C \subseteq \bigcup \mathscr{C}_{t_i} \cup \{v\} \qquad \textit{where } \{v\} = B_{t_i} \setminus B_t.$$

Intuitively, C is shattered, i.e., broken into several smaller connected components, by the bag of exactly one child of t.

Lemma

Let t_1, t_2 be two distinct children of t. For every $i \in [2]$, let v_i be the vertex in $\mathcal{G}$ with $\{v_i\} = B_{t_i} \setminus B_t$; and $C_i \in \mathscr{C}_{t_i}$. Then for every $C \in \mathscr{C}_t$

$$(C_1 \cup \{v_1\}) \cap C = \emptyset \quad \textit{or} \quad (C_2 \cup \{v_2\}) \cap C = \emptyset.$$

Connected components via tree decompositions (5)

Proof.

It is easy to see

$$\big(C_1 \cup \{v_1\}\big) \cap \big(C_2 \cup \{v_2\}\big) = \emptyset.$$

Assume $(C_1 \cup \{v_1\}) \cap C \neq \emptyset \neq (C_2 \cup \{v_2\}) \cap C$. Then there is a path P from $C_1 \cup \{v_1\}$ to $C_2 \cup \{v_2\}$ in C. Without loss of generality, we can assume that all vertices on P are in

$$\big(C_1 \cup \{v_1\}\big) \cup \big(C_2 \cup \{v_2\}\big).$$

Then there is an edge between $C_1 \cup \{v_1\}$ and $C_2 \cup \{v_2\}$, which cannot be contained in any bag of the tree decomposition. □

Decompose $\mathcal{H}$

Let $\mathcal{H}$ be a second graph for which we want to decide whether $\mathcal{G}$ and $\mathcal{H}$ are isomorphic.

We define (the set of pairs of separators and connected components)

$$\underline{\mathscr{S}\mathscr{C}(\mathcal{H})} := \Big\{(S, C) \;\Big|\; S \subseteq V(\mathscr{H}) \text{ with } |S| = k + 1 \\ \text{and } (C = \emptyset \text{ or } C \text{ a connected component of } \mathcal{H} \setminus S)\Big\}$$

Definition

Let $t \in V(\mathcal{T})$, $S_1 := B_t$, and $C_1 \in \mathscr{C}_t$. Moreover, let $(S_2, C_2) \in \mathscr{S}\mathscr{C}(\mathcal{H})$. We say (S_1, C_1) and (S_2, C_2) are *f-isomorphic* for a function $f : S_1 \to S_2$, denoted by $\underline{(S_1, C_1) \equiv^f (S_2, C_2)}$, if there is a function $F : S_1 \cup C_1 \to S_2 \cup C_2$ such that

(F1) $F \restriction S_1 = f$;

(F2) for every $u, v \in S_1 \cup C_1$ we have $\{u, v\} \in E(\mathcal{G})$ if and only if $\{F(u), F(v)\} \in E(\mathcal{H})$.

That is, F is an isomorphism between $\mathcal{G}[S_1 \cup C_1]$ and $\mathcal{H}[S_2 \cup C_2]$ which extends f.

Our goal is to compute for each $t \in V(\mathcal{T})$ the set

$$\underline{\mathscr{F}_t} := \big\{(f, B_t, C_1, S_2, C_2) \;\big|\; (B_t, C_1) \equiv^f (S_2, C_2) \\ \text{where } C_1 \in \mathscr{C}_t \text{ and } (S_2, C_2) \in \mathscr{S}\mathscr{C}(\mathcal{H})\big\}.$$

using dynamic programming.

Let t be a leaf of $\mathcal{T}$.

Then $\mathscr{C}_t = \{\emptyset\}$. Hence,

$$\mathscr{F}_t := \Big\{(f, B_t, \emptyset, S_2, \emptyset) \;\Big|\; (B_t, \emptyset) \equiv^f (S, \emptyset) \\ \text{where } S_2 \subseteq V(\mathcal{H}) \text{ with } |S_2| = k+1\Big\}.$$

This can be computed in time

$$(k+1)! \cdot |V(\mathcal{H})|^{O(k)}.$$

Non-leaves (1)

Let t be a node in $\mathcal{T}$ with children $t_1, \ldots, t_m$ for some $m \geq 1$.

Now let $C_1 \in \mathscr{C}_t$. By Lemma 3, there is a unique $i \in [m]$ such that

$$C_1 \subseteq \bigcup \mathscr{C}_{t_i} \cup \{v\} \qquad \text{where } \{v\} = B_{t_i} \setminus B_t.$$

For every $(S_2, C_2) \in \mathscr{S}\mathscr{C}(\mathcal{H})$ and every $f : B_t \to S_2$ we want to check whether $(B_t, C_1) \equiv^f (S_2, C_2)$.

Non-leaves (2)

$(B_t, C_1) \equiv^f (S_2, C_2)$ if and only if for some $v' \in V(\mathcal{H}) \setminus S_2$ and $u \in S_2$ if we let

- $S_2' := S_2 \cup \{v'\} \setminus \{u\}$,
- $\mathscr{C}_1^* := \{C^* \mid C^*$ a connected component of $\mathcal{G} \setminus B_{t_i}$ with $C^* \subseteq C_1\}$ and
  $\mathscr{C}_2^* := \{C^* \mid C^*$ a connected component of $\mathcal{H} \setminus S_2'$ with $C^* \subseteq C_2\}$,
- $f' : B_{t_i} \to S_2'$ defined by

$$f'(w) = \begin{cases} v' & \text{if } w = v \\ f(w) & \text{otherwise,} \end{cases}$$

then

(N1) every connected component of $\mathcal{H} \setminus S_2'$ is either contained in or disjoint with C_2;

(N2) $C_2 \subseteq \bigcup \mathscr{C}_2^* \cup \{v'\}$;

(N3) there is a bijection $h : \mathscr{C}_1^* \to \mathscr{C}_2^*$ such that for every $C^* \in \mathscr{C}_1^*$

$$(B_{t_i}, C^*) \equiv^{f'} (S_2', h(C^*)).$$

Non-leaves (3)

(N1) and (N2) can be checked in polynomial time.

To verify (N3) we create a bipartite graph $\mathcal{B}$:

1. the left part is $\mathscr{C}_1^*$ and the right part $\mathscr{C}_2^*$;
2. there is an edge between $C_1^* \in \mathscr{C}_1^*$ and $C_2^* \in \mathscr{C}_2^*$ if $(B_{t_i}, C_1^*) \equiv^{f'} (S_2', C_2^*)$.

Then (N3) holds if and only if there is a perfect matching in $\mathcal{B}$, which can be decided in polynomial time.

The final step

$\mathcal{G}$ and $\mathcal{H}$ are isomorphic if and only if for some $S_2 \subseteq V(\mathcal{H})$ with $|S_2| = k + 1$ and $f : B_r \to S_2$ there is a perfect matching in the following bipartite graph.

1. The left part is $\mathscr{C}_r$ and the right part $\mathscr{C}^* := \left\{ C_2 \;\middle|\; (S_2, C_2) \in \mathscr{S}\mathscr{C}(\mathcal{H}) \right\}$.
2. There is an edge between a $C_1 \in \mathscr{C}_r$ and a $C_2 \in \mathscr{C}^*$ if $(B_r, C_1) \equiv^f (S_2, C^*)$.

Questions

Without $\left(\mathcal{T}, (B_t)_{t \in V(\mathcal{T})}\right)$

Can we modify the algorithm so that it doesn't need a smooth tree decomposition as a part of the input? And even without computing such a tree decomposition inside the algorithm?

Thank you