Advanced Algorithms (III)

Yijia Chen
Fudan University

Review

Partial k-Trees

k-trees and partial k-trees.

Definition

Let $k \in \mathbb{N}$. Then the set of k-trees is defined as follows.

(K1) A complete graph $\mathcal{K}_{k+1}$ is a k-tree.

(K2) Let $\mathcal{G}$ be a graph and $v \in V$ such that

- $\mathcal{N}^{\mathcal{G}}[v]$ is isomorphic to $\mathcal{K}_{k+1}$, where $\mathcal{N}^{\mathcal{G}}[v]$ is the induced subgraph of $\mathcal{G}$ on

$$N^{\mathcal{G}}[v] := \{u \in V(\mathcal{G}) \mid \{u, v\} \in E(\mathcal{G})\} \cup \{v\}$$

- $\mathcal{G}[V(\mathcal{G}) \setminus \{v\}]$ is a k-tree.

Then $\mathcal{G}$ is a k-tree.

Definition

A graph is a partial k-tree if it is a subgraph of a k-tree.

Partial k-trees and bounded treewidth

Theorem

A graph $\mathcal{G}$ is a partial k-tree if and only if $\text{tw}(\mathcal{G}) \leq k$.

Lemma

Let $\mathcal{G}$ be a subgraph of $\mathcal{H}$, i.e., $V(\mathcal{G}) \subseteq V(\mathcal{H})$ and $E(\mathcal{G}) \subseteq E(\mathcal{H})$. Then $\text{tw}(\mathcal{G}) \leq \text{tw}(\mathcal{H})$.

Theorem

1. *Every graph of treewidth $\leq k$ is a partial k-tree.*
2. *Every k-tree has a tree decomposition of width $\leq k$.*

Algebraic Construction

k-terminal graphs

Definition

Let $k \in \mathbb{N}$. A k-terminal graph $(\mathcal{G}, v_1, \ldots, v_k)$ is a pair of a graph and a tuple of its k pairwise distinct vertices, called terminals.

Ordinary graphs are obtained as 0-terminal graphs.

The algebra $\mathbf{B}$

Definition

For every $k \in \mathbb{N}$, we use $\mathbf{B}_k$ to denote a set of terms defined recursively as follows.

1. $\mathbf{0}$ is a term in $\mathbf{B}_0$.
2. $\mathbf{e}^2$ is a term in $\mathbf{B}_2$.
3. If s_1 and s_2 are two $\mathbf{B}_k$-terms, then $s_1 \oplus_k s_2$ is also $\mathbf{B}_k$-term.
4. Let $i < k$. If s is a $\mathbf{B}_k$-term, then $\sigma_k^i(s)$ is also a $\mathbf{B}_k$-term.
5. Let $i \leq k$. If s is a $\mathbf{B}_{k-1}$-term, then $\ell_k^i(s)$ is a $\mathbf{B}_k$-term.
6. If s is a $\mathbf{B}_k$-term, then $r_k(s)$ is a $\mathbf{B}_{k-1}$-term.

Let

$$\mathbf{B} := \bigcup_{k \geq 0} \mathbf{B}_k.$$

The mapping ψ

Definition

ψ maps every $\mathbf{B}_k$-term to a k-terminal graph in the following way:

1. $\psi(\mathbf{e}^2)$ is an edge with two terminals.
2. $\psi(\mathbf{0})$ is the empty graph.
3. $\psi(s_1 \oplus_k s_2)$ is a parallel composition i.e., fuse each i-th terminal in $\psi(s_1)$ and $\psi(s_2)$ for every $i \in [k]$.
4. $\psi(\sigma_k^i(s))$ is a permutation, i.e., permute the i-th terminal and the $i+1$-th terminal in $\psi(s)$.
5. $\psi(\ell_k^i(s))$ is a lifting, i.e., insert a new isolated terminal (as a new vertex) to $\psi(s)$ at the i-th position in $k-1$ terminals.
6. $r_k(s)$ removes the last terminal from $\psi(s)$.

Hereditary $\mathbf{B}_{\leq k+1}$-terms and treewidth $\leq k$

Definition

Let $k \geq 0$ and s be a $\mathbf{B}$-term. We say that s is a hereditary $\mathbf{B}_{\leq k}$-term, if every subterm of s (including s itself) is a $\mathbf{B}_{k'}$-term for some $k' \leq k$.

Theorem (Ogawa, 2004)

Let $k \geq 0$. Then a graph $\mathcal{G}$ is isomorphic to the graph $\psi(s)$ for a hereditary $\mathbf{B}_{\leq k+1}$-term (neglecting terminals) if and only if $\mathrm{tw}(\mathcal{G}) \leq k$.

Matchings in Bipartite Graphs

Matchings

Two edges $e \neq f$ in a graph G are adjacent if they have a vertex in common. Otherwise, they are independent.

Definition

A set M of pairwise independent edges in a graph $G = (V, E)$ is a matching.

M is a matching of $U \subseteq V$ if every vertex in U is incident with an edge in M. The vertices in U are then called matched (by M); vertices not incident with any edge of M are unmatched.

Definition

A matching M in a graph $G = (V, E)$ is perfect if every vertex is matched by M. Or equivalently, $|M| = |V|/2$.

Augmenting paths

We fix a bipartite graph $G = (V, E)$ with bipartition $A|B$. That is, $V = A \cup B$, $A \cap B = \emptyset$, and every edge in G has one vertex in A and one in B.

Let M be a matching in G. A path in G which starts in A at an unmatched vertex and then contains, alternately, edges from $E \setminus M$ and from M, is an alternating path with respect to M.

An alternating path that ends in an unmatched vertex of B is an augmenting path.

Lemma

Let M be a matching in a graph G. If there is an augmenting path with respect to M, then we have a matching M' in G with $|M'| > |M|$.

Vertex covers

Definition

Let $G = (V, E)$ be a graph. Then a set $S \subseteq V$ a vertex cover of E if every edge of G is incident with a vertex in S.

König's Theorem

Theorem (König, 1931)

The maximum cardinality of a matching in G is equal to the minimum cardinality of a vertex cover of its edges.

Proof.

Let M be a matching of maximum cardinality.

Define a set S in the following way: For every $\{a, b\} \in M$ with $a \in A$, if there is an alternating path ending in b, then $b \in S$; otherwise $a \in S$. $\square$

Hall's Theorem

Theorem (Hall, 1935)

Let $G = (V, E)$ be a bipartite graph with partition $A|B$. Then G contains a matching of A if and only if

$$\left|N^G(S)\right| \geq |S|$$

for all $S \subseteq A$, where

$$N^G(S) := \left\{v \in B \;\middle|\; \text{for some } u \in S \text{ we have } \{u, v\} \in E\right\}.$$

First proof

Induction on $|A|$.

Trivial for $|A| = 1$.

If $|N(S)| \geq |S| + 1$ for every non-empty set $S \subsetneq A$, we pick an edge $\{a, b\} \in E$ and apply induction hypothesis on the graph $G' := G - \{a, b\}$.

Assume A has a nonempty proper subset A' with $|B'| = |A'|$ for $B' := N(A')$. We apply the induction hypothesis on $G[A' \cup B']$ and $G - (A' \cup B')$. $\square$

Second proof

Let M be a matching that leaves a vertex in A unmatched. We will construct an augmenting path with respect to M.

Let $a_0, b_1, a_1, b_2, a_2, \ldots$ be a maximal sequence of distinct vertices $a_i \in A$ and $b_i \in B$ satisfying the following conditions for all $i \geq 1$:

1. a_0 is unmatched;
2. b_i is adjacent to some vertex $a_{f(i)} \in \{a_0, \ldots, a_{i-1}\}$;
3. $\{a_i, b_i\} \in M$.

The sequence will end in some vertex $b_k \in B$.

Consider

$$P := b_k a_{f(k)} b_{f(k)} a_{f^2(k)} b_{f^2(k)} a_{f^3(k)} \ldots, a_{f^r(k)}$$

with $f^r(k) = 0$ is an alternating path.

It is easy to see that P is an augmenting path. □

Graph Isomorphism Problems

Graph isomorphism

Definition

Let $\mathcal{G}$ and $\mathcal{H}$ be two graphs. A function $f : V(\mathcal{G}) \to V(\mathcal{H})$ 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 (which most people do not believe).*
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.*

Graph isomorphism problems and treewidth

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* $\mathrm{tw}(\mathcal{G}) \leq k$.

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

$$\mathscr{C}_t := \big\{ C \;\big|\; C = \emptyset \text{ or } C \text{ a connected component of } \mathcal{G}_{\le t} \setminus B_t \big\}.$$

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 an 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. □

Connected components via tree decompositions (2)

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. $\square$

Connected components via tree decompositions (3)

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

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

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

Connected components via tree decompositions (4)

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 $C \in \mathscr{C}_t$

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

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. $\square$

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)

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

Partial isomorphisms

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 $(S_1, C_1) \equiv^f (S_2, C_2)$, 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.

Extending partial isomorphisms

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

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

using dynamic programming.

Leaves

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 $t_1, \dots, t_m$ for some $m \geq 1$.

Now let $C_1 \in \mathscr{C}_t$ be nonempty. 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 $f : B_t \to S_2$ with $(B_t, \emptyset) \equiv^f (S_2, \emptyset)$ 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 some for $v' \in V(\mathcal{H}) \setminus S_2$, $u \in S_2$, and

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

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

we have

(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 $(B_r, C_1) \equiv^f (S_2, C^*)$.