The parameterized complexity of k-edge induced subgraphs

Yijia Chen
Shanghai Jiaotong University

December, 2011

(Joint work with Bingkai Lin)

One message

One message

Logic can help designing combinatorial algorithms!

The k-edge induced subgraph problems

The k-edge induced subgraph problems

We have the budget for one million kilometers of high speed railway system.
But where to build?

The k-edge induced subgraph problems

We have the budget for one million kilometers of high speed railway system. But where to build?

Fix a $k \in \mathbb{N}$.

k-EDGE-INDUCED-SUBGRAPH	
Input:	A graph G.
Problem:	Does G contain an induced subgraph of exactly k-edges?

The k-edge induced subgraph problems

We have the budget for one million kilometers of high speed railway system. But where to build?

Fix a $k \in \mathbb{N}$.

k-EDGE-INDUCED-SUBGRAPH	
Input:	A graph G.
Problem:	Does G contain an induced subgraph of exactly k-edges?

Remark.

1. *For each fixed $k \in \mathbb{N}$ the k-*EDGE-INDUCED-SUBGRAPH *problem can be solved in time $n^{O(k)}$.*
2. *If k is a part of the input, then the problem is* NP*-complete.*

Our result

Our result

Theorem

For each fixed $k \in \mathbb{N}$ there is a linear time algorithm that decides the k-EDGE-INDUCED-SUBGRAPH problem.

Our result

Theorem

For each fixed $k \in \mathbb{N}$ there is a linear time algorithm that decides the k-EDGE-INDUCED-SUBGRAPH problem.

Our tools:

1. several Ramsey-type combinatorial results;
2. Gauss' Eureka Theorem;
3. efficient algorithms for checking first-order logic definable properties on graphs with nice structural properties (e.g., bounded-degree graphs, bounded-treewidth graphs), i.e., a generalization of the well-known algorithmic meta-theorem due to Frick and Grohe.

Our result (another formulation)

Our result (another formulation)

p-Edge-Induced-Subgraph

Input:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether G contains a k-edge induced subgraph.

Our result (another formulation)

p-EDGE-INDUCED-SUBGRAPH	
Input:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether G contains a k-edge induced subgraph.

Theorem

There is an algorithm that decides p-EDGE-INDUCED-SUBGRAPH in time

$$f(k) \cdot \|G\|.$$

Our result (another formulation)

p-EDGE-INDUCED-SUBGRAPH	
Input:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether G contains a k-edge induced subgraph.

Theorem

There is an algorithm that decides p-EDGE-INDUCED-SUBGRAPH in time

$$f(k) \cdot \|G\|.$$

Some bad news:

$$f(k) \geq 2^{2^{2^k}}.$$

Previously known cases

Previously known cases

1. By logic methods:
   - [Courcelle, 1990] G is of bounded treewidth;
   - [Frick and Grohe, 2001] G is of bounded local treewidth (e.g., bounded-degree, planar);
   - [Flum and Grohe, 2001] G excludes a minor;
   - [Dawar, Kreutzer, and Grohe, 2007] G locally excludes a minor;
   - [Dvorak, Král, and Thomas, 2010] G is of locally bounded expansion.

Previously known cases

1. By logic methods:
   - [Courcelle, 1990] G is of bounded treewidth;
   - [Frick and Grohe, 2001] G is of bounded local treewidth (e.g., bounded-degree, planar);
   - [Flum and Grohe, 2001] G excludes a minor;
   - [Dawar, Kreutzer, and Grohe, 2007] G locally excludes a minor;
   - [Dvorak, Král, and Thomas, 2010] G is of locally bounded expansion.

2. By combinatorial methods: [Cai, 2006] k is a triangular number $\binom{m}{2}$.

Contents

Algorithmic meta-theorems

Theorems of the form:

> *There is an efficient algorithm checking whether a graph with* a certain structural property *satisfies a sentence in* a certain logic*.*

Courcelle's Theorem

Definition

Let K be a class of graphs. If there is a constant $c \in \mathbb{N}$ such that $\text{tw}(G) \leq c$ for every $G \in K$, then K has bounded treewidth.

Courcelle's Theorem

Definition

Let K be a class of graphs. If there is a constant $c \in \mathbb{N}$ such that $\mathrm{tw}(G) \leq c$ for every $G \in K$, then K has bounded treewidth.

Theorem (Courcelle, 1990)

If K has bounded treewidth. Then for every $\varphi \in \mathrm{MSO}$ there is a linear time algorithm that decides whether a graph in K satisfies φ.

There is an algorithm which decides whether a graph G satisfies an MSO-sentence φ in time

$$f(|\varphi|, \mathrm{tw}(G)) \cdot \|G\|.$$

Frick-Grohe Theorem

Frick-Grohe Theorem

Definition

K has bounded degree if there is a constant $d \in \mathbb{N}$ such that every graph in K has degree bounded by d.

Frick-Grohe Theorem

Definition

K has bounded degree if there is a constant $d \in \mathbb{N}$ such that every graph in K has degree bounded by d.

Theorem (Frick and Grohe, 2001; Seese, 1996)

Assume K has bounded degree. Then for every $\varphi \in \mathrm{FO}$ there is a linear time algorithm that decides whether a graph $G \in K$ satisfies φ.

Frick-Grohe Theorem

Definition

K has bounded degree if there is a constant $d \in \mathbb{N}$ such that every graph in K has degree bounded by d.

Theorem (Frick and Grohe, 2001; Seese, 1996)

Assume K has bounded degree. Then for every $\varphi \in$ FO there is a linear time algorithm that decides whether a graph $G \in K$ satisfies φ.

There is an algorithm which for every $G \in K$ and every $\varphi \in$ FO decides whether G satisfies φ in time

$$f(|\varphi|) \cdot \|G\|.$$

Frick-Grohe Theorem

Definition

K has bounded degree if there is a constant $d \in \mathbb{N}$ such that every graph in K has degree bounded by d.

Theorem (Frick and Grohe, 2001; Seese, 1996)

Assume K has bounded degree. Then for every $\varphi \in$ FO there is a linear time algorithm that decides whether a graph $G \in K$ satisfies φ.

There is an algorithm which for every $G \in K$ and every $\varphi \in$ FO decides whether G satisfies φ in time

$$f(|\varphi|) \cdot \|G\|.$$

Remark. *What Frick and Grohe proved is actually on graphs with bounded local treewidth, which include bounded-degree graphs and also planar graphs.*

Our algorithms

Gauss' Eureka Theorem

Gauss' Eureka Theorem

Theorem

For every $k \in \mathbb{N}$ *there exist* $k_0, k_1, k_2 \in \mathbb{N}$ *with*

$$k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}.$$

Gauss' Eureka Theorem

Theorem

For every $k \in \mathbb{N}$ there exist $k_0, k_1, k_2 \in \mathbb{N}$ with

$$k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}.$$

Corollary

If a graph G contains three k-cliques with no edge between them, then it contains a k-edge induced subgraph.

Gauss' Eureka Theorem

Theorem

For every $k \in \mathbb{N}$ there exist $k_0, k_1, k_2 \in \mathbb{N}$ with

$$k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}.$$

Corollary

If a graph G contains three k-cliques with no edge between them, then it contains a k-edge induced subgraph.

Starting with any given graph G, our algorithm works on G in a few stages. In each stage, either G has relatively simple structure so that we can already decide the problem or G can be simplified. Eventually we will arrive in the above case.

The starting point

The starting point

Let $H = (V, E)$ be a graph. We can assume that $V = [\ell]$ for some $\ell \in \mathbb{N}$ and define

$$\mathit{induced}_H := \exists x_1 \ldots \exists x_\ell \left(\bigwedge_{1 \le i < j \le \ell} x_i \ne x_j \wedge \bigwedge_{\{i,j\} \in E} E x_i x_j \wedge \bigwedge_{\{i,j\} \in \binom{V}{2} \setminus E} \neg E x_i x_j \right).$$

The starting point

Let $H = (V, E)$ be a graph. We can assume that $V = [\ell]$ for some $\ell \in \mathbb{N}$ and define

$$\mathit{induced}_H := \exists x_1 \ldots \exists x_\ell \left(\bigwedge_{1 \leq i < j \leq \ell} x_i \neq x_j \wedge \bigwedge_{\{i,j\} \in E} E x_i x_j \wedge \bigwedge_{\{i,j\} \in \binom{V}{2} \setminus E} \neg E x_i x_j \right).$$

Then for every fixed $k \in \mathbb{N}$ we let

$$\mathit{induced}_k := \bigvee_{\substack{H \text{ has no isolated vertex} \\ \text{and } |E(H)| = k}} \mathit{induced}_H.$$

The starting point

Let $H = (V, E)$ be a graph. We can assume that $V = [\ell]$ for some $\ell \in \mathbb{N}$ and define

$$induced_H := \exists x_1 \dots \exists x_\ell \left(\bigwedge_{1 \le i < j \le \ell} x_i \neq x_j \wedge \bigwedge_{\{i,j\} \in E} Ex_ix_j \wedge \bigwedge_{\{i,j\} \in \binom{V}{2} \setminus E} \neg Ex_ix_j \right).$$

Then for every fixed $k \in \mathbb{N}$ we let

$$induced_k := \bigvee_{\substack{H \text{ has no isolated vertex} \\ \text{and } |E(H)| = k}} induced_H.$$

Lemma

$G \models induced_k$ if and only if G contains a k-edged induced subgraph.

When the degrees are small . . .

When the degrees are small . . .

By the previous lemma and Frick-Grohe theorem, we have linear time algorithms for k-EDGE-INDUCED-SUBGRAPH if G is of bounded degree.

When the degrees are small ...

By the previous lemma and Frick-Grohe theorem, we have linear time algorithms for k-Edge-Induced-Subgraph if G is of bounded degree.

We say G has sufficiently small degree (with respect to k) if there is a function $D : \mathbb{N} \to \mathbb{N}$ (independent of G) such that the degree of each vertex in G is bounded by $D(k)$.

When the degrees are small ...

By the previous lemma and Frick-Grohe theorem, we have linear time algorithms for k-EDGE-INDUCED-SUBGRAPH if G is of bounded degree.

We say G has sufficiently small degree (with respect to k) if there is a function $D : \mathbb{N} \to \mathbb{N}$ (independent of G) such that the degree of each vertex in G is bounded by $D(k)$.

We generalize Frick and Grohe's Theorem to give linear time algorithms for k-EDGE-INDUCED-SUBGRAPH if G has sufficiently small degree.

A generalization of Frick and Grohe's Theorem

A generalization of Frick and Grohe's Theorem

Theorem

Let $D : \mathbb{N} \to \mathbb{N}$ be a function. Then for every $\varphi \in \mathsf{FO}$ and $k \in \mathbb{N}$ there is a linear time algorithm that decides whether a graph with degree bounded by $D(k)$ satisfies φ.

A generalization of Frick and Grohe's Theorem

Theorem

Let $D : \mathbb{N} \to \mathbb{N}$ be a function. Then for every $\varphi \in \mathrm{FO}$ and $k \in \mathbb{N}$ there is a linear time algorithm that decides whether a graph with degree bounded by $D(k)$ satisfies φ.

There is an algorithm which for every graph G and every $\varphi \in \mathrm{FO}$ decides whether G satisfies φ in time

$$f(\deg(G), |\varphi|) \cdot \|G\|.$$

When the degrees are small . . .

When the degrees are small . . .

Let $D : \mathbb{N} \to \mathbb{N}$ be a function which will be determined later. A graph G has sufficiently small degree (with respect to k) if every vertex in G has degree at most $D(k)$.

When the degrees are small ...

Let $D : \mathbb{N} \to \mathbb{N}$ be a function which will be determined later. A graph G has sufficiently small degree (with respect to k) if every vertex in G has degree at most $D(k)$.

Corollary

For every $k \in \mathbb{N}$ there is a linear time algorithm that decides whether a graph with sufficiently small degree contains a k-edge induced subgraph.

When the degrees are either small or large . . .

When the degrees are either small or large . . .

Again let $D : \mathbb{N} \to \mathbb{N}$ be a function.

When the degrees are either small or large ...

Again let $D : \mathbb{N} \to \mathbb{N}$ be a function.

Definition

A graph G is degree-extreme (with respect to $k \in \mathbb{N}$) if every vertex in G has degree at most $D(k)$ or at least $|G| - 1 - D(k)$, e.g., stars.

When the degrees are either small or large ...

Again let $D : \mathbb{N} \to \mathbb{N}$ be a function.

Definition

A graph G is degree-extreme (with respect to $k \in \mathbb{N}$) if every vertex in G has degree at most $D(k)$ or at least $|G| - 1 - D(k)$, e.g., stars.

Again, by generalizing Frick-Grohe Theorem:

Theorem

For every $k \in \mathbb{N}$ there is a linear time algorithm that decides whether a degree-extreme graph contains a k-edge induced subgraph.

When G is not degree-extreme . . .

When G is not degree-extreme . . .

There is a vertex $v_0 \in G$ whose degree is between $D(k)$ and $|G| - 1 - D(k)$.

When G is not degree-extreme ...

There is a vertex $v_0 \in G$ whose degree is between $D(k)$ and $|G| - 1 - D(k)$.
We let

$$V_1 := \{v \mid v \text{ is adjacent to } v_0\} \quad \text{and} \quad V_2 := \{v \mid v \text{ is not adjacent to } v_0\}.$$

Both V_1 and V_2 are relatively large, of size at least $D(k)$.

When G is not degree-extreme ...

There is a vertex $v_0 \in G$ whose degree is between $D(k)$ and $|G| - 1 - D(k)$. We let

$$V_1 := \{v \mid v \text{ is adjacent to } v_0\} \quad \text{and} \quad V_2 := \{v \mid v \text{ is not adjacent to } v_0\}.$$

Both V_1 and V_2 are relatively large, of size at least $D(k)$.

There are possibly many edges between V_1 and V_2. Nevertheless, we can compute a set B of vertices such that

1. every edge between V_1 and V_2 intersects B;
2. if B is sufficiently large, then G contains a k-edge induced subgraph (several Ramsey-type arguments).

When G is not degree-extreme ...

There is a vertex $v_0 \in G$ whose degree is between $D(k)$ and $|G| - 1 - D(k)$. We let

$$V_1 := \{v \mid v \text{ is adjacent to } v_0\} \quad \text{and} \quad V_2 := \{v \mid v \text{ is not adjacent to } v_0\}.$$

Both V_1 and V_2 are relatively large, of size at least $D(k)$.

There are possibly many edges between V_1 and V_2. Nevertheless, we can compute a set B of vertices such that

1. every edge between V_1 and V_2 intersects B;
2. if B is sufficiently large, then G contains a k-edge induced subgraph (several Ramsey-type arguments).

So we can assume B is sufficiently small.

When G is a bridge . . .

When G is a bridge . . .

Recall we decompose G into $G[V_1]$ and $G[V_2]$ such that

- both V_1 and V_2 are sufficiently large;
- all the edges between V_1 and V_2 intersects a set B of bounded size.

When G is a bridge ...

Recall we decompose G into $G[V_1]$ and $G[V_2]$ such that

- both V_1 and V_2 are sufficiently large;
- all the edges between V_1 and V_2 intersects a set B of bounded size.

If furthermore

- both $G[V_1]$ and $G[V_2]$ are degree-extreme, i.e., the degree of each vertex in $G[V_1]$ and $G[V_2]$ is either sufficiently large or sufficiently small,

then G is a bridge (of two degree-extreme graphs).

When G is a bridge . . .

Recall we decompose G into $G[V_1]$ and $G[V_2]$ such that

- both V_1 and V_2 are sufficiently large;
- all the edges between V_1 and V_2 intersects a set B of bounded size.

If furthermore

- both $G[V_1]$ and $G[V_2]$ are degree-extreme, i.e., the degree of each vertex in $G[V_1]$ and $G[V_2]$ is either sufficiently large or sufficiently small,

then G is a bridge (of two degree-extreme graphs).

Again, by generalizing Frick-Grohe Theorem:

Theorem

For every $k \in \mathbb{N}$ there is a linear time algorithm that decides whether a bridge contains a k-edge induced subgraph.

When G is not a bridge . . .

When G is not a bridge . . .

Assume without loss of generality $G[V_1]$ is not degree-extreme.

When G is not a bridge . . .

Assume without loss of generality $G[V_1]$ is not degree-extreme.

By repeating the previous procedure, we decompose $G[V_1]$ into $G[V_{11}]$ and $G[V_{12}]$ such that

- both V_{11} and V_{12} are sufficiently large;
- all the edges between V_{11} and V_{12} intersects a set B_1 of bounded size.

G with three parts

G with three parts

We have decomposed G into three parts, $G[V_{11}]$, $G[V_{12}]$ and $G[V_2]$.

- All three parts are sufficiently large.
- Every edge between three parts intersects $B_1 \cup B$ of bounded size.

G with three parts

We have decomposed G into three parts, $G[V_{11}]$, $G[V_{12}]$ and $G[V_2]$.

- All three parts are sufficiently large.
- Every edge between three parts intersects $B_1 \cup B$ of bounded size.

We remove $B_1 \cup B$ from G so that the remaining three parts have no edge between them.

Three separate parts

Three separate parts

Recall "being sufficiently large" depends on our choice of the function D in the definition of degree-extreme graphs.

Three separate parts

Recall "being sufficiently large" depends on our choice of the function D in the definition of degree-extreme graphs.

We choose D so large that after removing $B_1 \cup B_2$ each of the resulting three parts has size at least

$$2^{2 \cdot k} \geq \mathscr{R}_k,$$

i.e., the Ramsey Number of k.

Three separate parts

Recall "being sufficiently large" depends on our choice of the function D in the definition of degree-extreme graphs.

We choose D so large that after removing $B_1 \cup B_2$ each of the resulting three parts has size at least

$$2^{2 \cdot k} \geq \mathscr{R}_k,$$

i.e., the Ramsey Number of k.

Therefore each part contains either a clique or an independent set of size k.

Three separate parts

Recall "being sufficiently large" depends on our choice of the function D in the definition of degree-extreme graphs.

We choose D so large that after removing $B_1 \cup B_2$ each of the resulting three parts has size at least

$$2^{2 \cdot k} \geq \mathscr{R}_k,$$

i.e., the Ramsey Number of k.

Therefore each part contains either a clique or an independent set of size k. For both cases, we can show G has a k-edge induced subgraph.

Three separate parts

Recall "being sufficiently large" depends on our choice of the function D in the definition of degree-extreme graphs.

We choose D so large that after removing $B_1 \cup B_2$ each of the resulting three parts has size at least

$$2^{2 \cdot k} \geq \mathscr{R}_k,$$

i.e., the Ramsey Number of k.

Therefore each part contains either a clique or an independent set of size k. For both cases, we can show G has a k-edge induced subgraph. Recall the clique case is based on Gauss' Eureka Theorem:

Theorem

For every $k \in \mathbb{N}$ there exist $k_0, k_1, k_2 \in \mathbb{N}$ with

$$k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}.$$

Thank You!