The PACE 2023 Challenge: Twin-Width

Anakin

Outline

What is Twin-width?

Computing Twin-width

PACE 2023

Section 1

What is Twin-width?

What is Twin-width?

A large part of combinatorics is defining useful measures of the complexity of structures

What is Twin-width?

A large part of combinatorics is defining useful measures of the complexity of structures

- How easy is it to construct the object?

What is Twin-width?

A large part of combinatorics is defining useful measures of the complexity of structures

- How easy is it to construct the object?
- Algorithmic Complexity

What is Twin-width?

A large part of combinatorics is defining useful measures of the complexity of structures

- How easy is it to construct the object?
- Algorithmic Complexity
- Efficient encodings

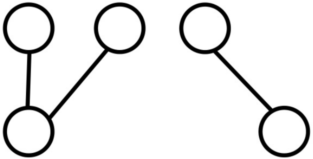
What is Twin-width?

A large part of combinatorics is defining useful measures of the complexity of structures

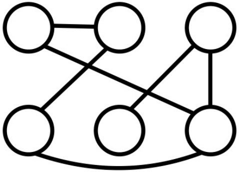
- How easy is it to construct the object?
- Algorithmic Complexity
- Efficient encodings
- *Decomposition*

Operations on Graphs: Disjoint Union

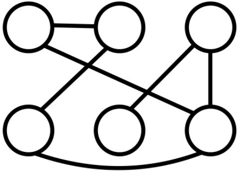
Operations on Graphs: Disjoint Union

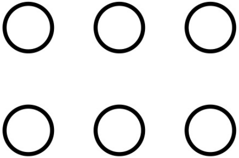


Operations on Graphs: Complement

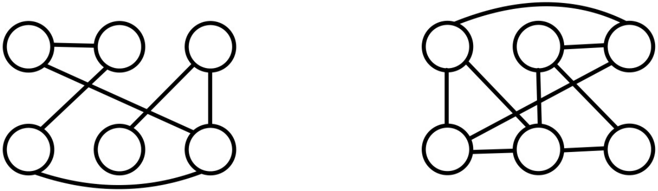


Operations on Graphs: Complement





Operations on Graphs: Complement



Cographs

We now construct the class of *cographs*

Cographs

We now construct the class of *cographs*

- K_1 is a cograph
- The disjoint union of two cographs is also a cograph
- The complement of a cograph is a cograph

Relating Other Graphs to Cographs

This efficient way of constructing and decomposing these graphs is useful for many algorithms

- Example: Finding the largest complete subgraph of a graph

Relating Other Graphs to Cographs

This efficient way of constructing and decomposing these graphs is useful for many algorithms

- Example: Finding the largest complete subgraph of a graph

However, not every graph is a cograph

Can we generalize this by adding some sort of *error* measure?

Relating Other Graphs to Cographs

This efficient way of constructing and decomposing these graphs is useful for many algorithms

- Example: Finding the largest complete subgraph of a graph

However, not every graph is a cograph

Can we generalize this by adding some sort of *error* measure?

This is called *twin-width* [BKTW20]

Contractions

Throughout this presentation, $G = (V; E)$ will be a connected undirected graph

Contractions

Throughout this presentation, $G = (V; E)$ will be a connected undirected graph

Before we can talk about twin-width, we first talk about *contractions* of a graph

Contractions

Throughout this presentation, $G = (V; E)$ will be a connected undirected graph

Before we can talk about twin-width, we first talk about contractions of a graph

- Our edges in E will have a color associated with them: black or red.
  - The red edges will be our error that we want
- Vertices are black neighbors if linked by a black edge, and red neighbors if linked by a red edge

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Contractions by picture

Formal Definition of Contraction

Suppose we are contracting two nodes u, v in G into one node called uv.

For every neighbor x of u or v:

Formal Definition of Contraction

Suppose we are contracting two nodes u, v in G into one node called uv.

- For every neighbor x of u or v:
  - If x is a black neighbor of both u and v, then x is now a black neighbor of uv
  - Otherwise, x is a red neighbor of u and v

Formal Definition of Contraction

Suppose we are contracting two nodes u, v in G into one node called uv.

For every neighbor x of u or v:

- If x is a black neighbor of both u and v, then x is now a black neighbor of uv
- Otherwise, x is a red neighbor of u and v

All other edges are left alone and maintain their color

Twin-width

Repeatedly applying the contraction operation to nodes of G produces a contraction sequence of graphs $G = G_n \to G_{n-1} \to \cdots \to G_2 \to G_1 = K_1$

Twin-width

Repeatedly applying the contraction operation to nodes of G produces a contraction sequence of graphs

$$G = G_n \to G_{n-1} \to \cdots \to G_2 \to G_1 = K_1$$

The sequence is a d-sequence if the maximum number of red edges in any of the G_i in the contraction sequence is d

Twin-width

Repeatedly applying the contraction operation to nodes of G produces a contraction sequence of graphs

$$G = G_n \to G_{n-1} \to \cdots \to G_2 \to G_1 = K_1$$

The sequence is a d-sequence if the maximum number of red edges in any of the G_i in the contraction sequence is d

The twin-width of G is the minimum d such that there exists a d-sequence of G

Observations

The graph remains connected as we do this

Observations

The graph remains connected as we do this

This process will terminate (nitely many nodes and edges)

Observations

The graph remains connected as we do this

This process will terminate (nitely many nodes and edges)

The number of red edges may increase or decrease

Why do We Care?

We can speed up certain algorithms

Why do We Care?

We can speed up certain algorithms

If we have the d-sequence for a graph, we can decompose the graph into complete bipartite graphs and do breadth- rst-search in $O(n \log n)$ time [BGK+20]

This works even if the number of edges is $\Omega(n^2)$

Section 2

Computing Twin-width

It's Hard

There are very few practical algorithms for computing the twin-width of a graph in general

It's Hard

There are very few practical algorithms for computing the twin-width of a graph in general

There are some things known for special cases

- Cographs are the graphs with twin-width zero
- *d*-dimensional graphs have twin-width $3d$ [BKTW20]
- Planar graphs have twin-width 9, and bipartite planar graphs have twin-width 6 [Hli22]
- Graphs with 4 vertices have twin-width 1 and graphs with 5 vertices have twin-width 2 [Das22]
- In general, bipartite graphs may have arbitrarily large twin-width

SAT Solvers

Given a boolean formula consisting of variables, `and` gates, and `or` gates, we want to find a satisfying assignment

SAT Solvers

Given a boolean formula consisting of variables, `and` gates, and `or` gates, we want to find a satisfying assignment

This is a problem that appears in many places, so readily available *SAT Solvers* exist

SAT Solvers

Given a boolean formula consisting of variables, `and` gates, and `or` gates, we want to find a satisfying assignment

This is a problem that appears in many places, so readily available *SAT Solvers* exist

[SS21] found a way to encode twin-width into a SAT formula

This is one of the only ways we can reasonable compute twin-widths

Section 3

PACE 2023

A Programming Competition

Parameterized **A**lgorithms and **C**omputational **E**xperiments is a long-term programming competition

Our goal will be to devise an efficient algorithm to compute the twin-width d of arbitrary graphs and their d-sequence

- Exact Track: Compute the exact sequence
- Heuristic Track: Compute an approximate sequence

The Club Submission

I only found out about this recently so we are a bit behind

We should make on submission on one track

- I was thinking about focusing on the exact track

There are 100 public test cases + 100 test cases

Score = the number of test cases we can solve in a given time limit

My goal is just to put together something and see how well we do

Bibliography

Édouard Bonnet, Colin Geniet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant.
Twin-width III: max independent set and coloring.
CoRR, abs/2007.14161, 2020.

Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant.
Twin-width I: tractable FO model checking.
CoRR, abs/2004.14789, 2020.

Kajal Das.
Computation of twin-width of graphs, 2022.

Petr Hliněný.
Twin-width of planar graphs is at most 9, and at most 6 when bipartite planar, 2022.

André Schidler and Stefan Szeider.
A SAT approach to twin-width.
CoRR, abs/2110.06146, 2021.