On local structure of cubicity 2 graphs

Dibyayan Chakraborty

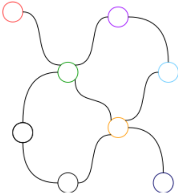
Indian Statistical Institute,Kolkata

COCOA 2016

** With Sujoy Bhore, Sandip Das, Sagnik Sen.

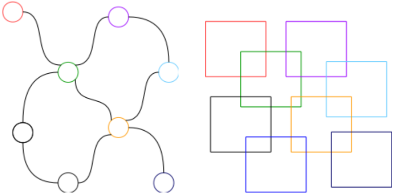
Cubicity 2 graphs

Intersection graph of unit squares on the plane



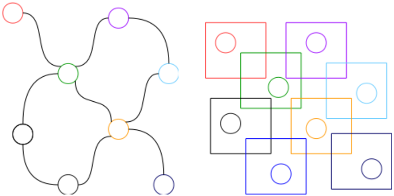
Cubicity 2 graphs

Intersection graph of unit squares on the plane



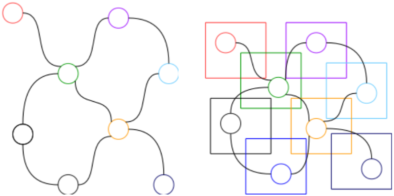
Cubicity 2 graphs

Intersection graph of unit squares on the plane



Cubicity 2 graphs

Intersection graph of unit squares on the plane



Cubicity 2 graphs

Unfortunately, deciding whether an input graph is a cubicity 2 graph, is NP-hard.

Cubicity 2 graphs

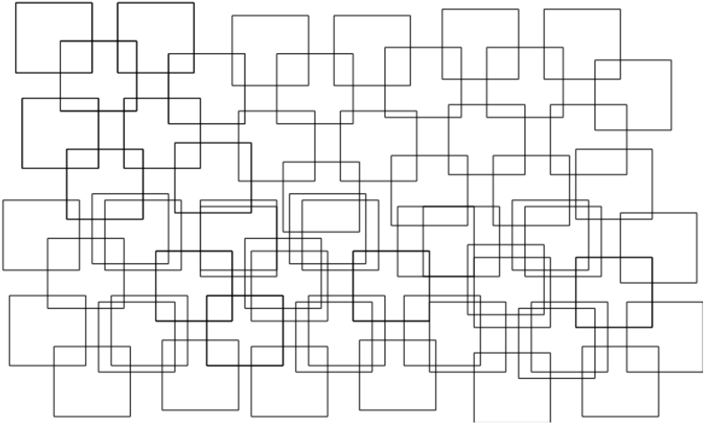
Unfortunately, deciding whether an input graph is a cubicity 2 graph, is NP-hard.

In fact, the complexity of deciding whether an input tree is a cubicity 2 graph, is unknown. (LS Chandran 2014 [1])

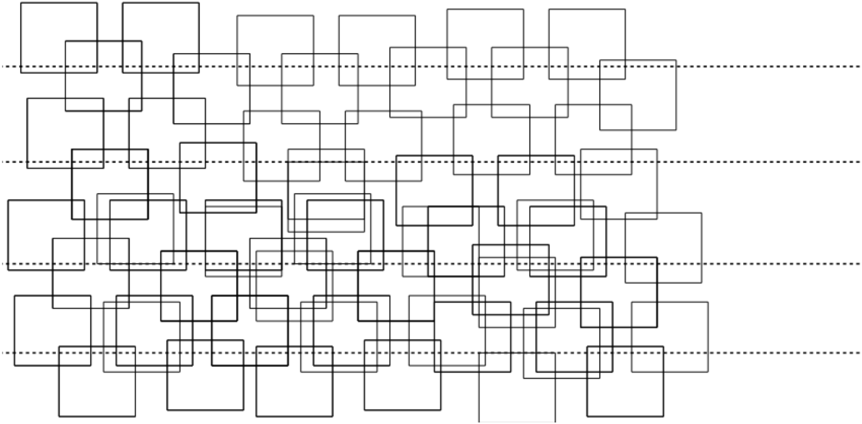
Today we shall solve the above problem "locally".

[1] Babu, Jasine, Manu Basavaraju, L. Sunil Chandran, Deepak Rajendraprasad, and Naveen Sivadasan."Approximating the cubicity of trees." arXiv preprint (2014).

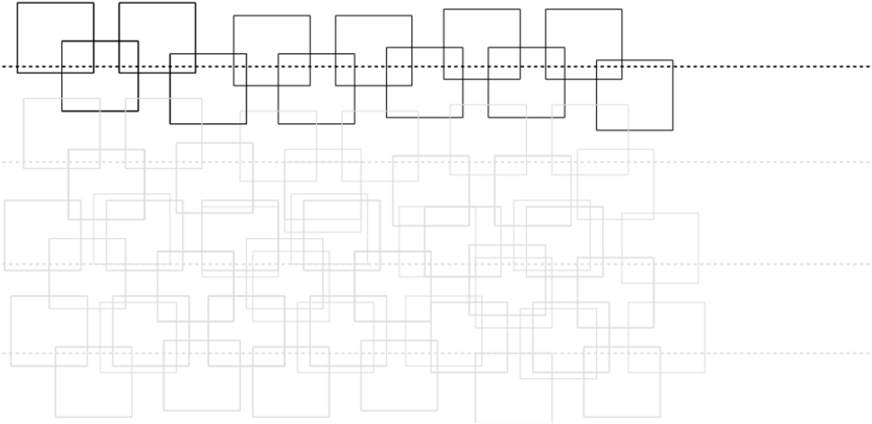
Local structure of cubicity 2 graphs



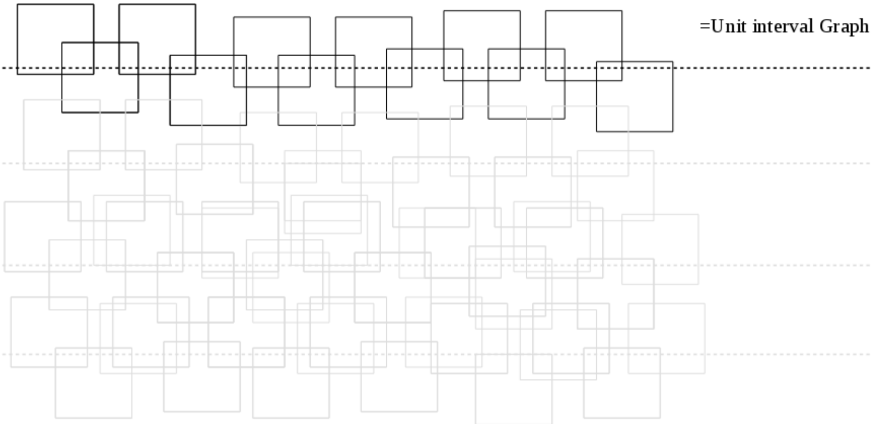
Local structure of cubicity 2 graphs



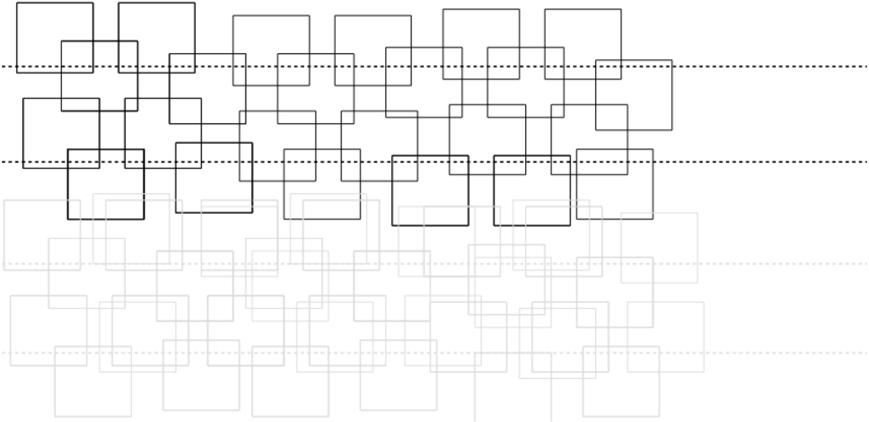
Local structure of cubicity 2 graphs



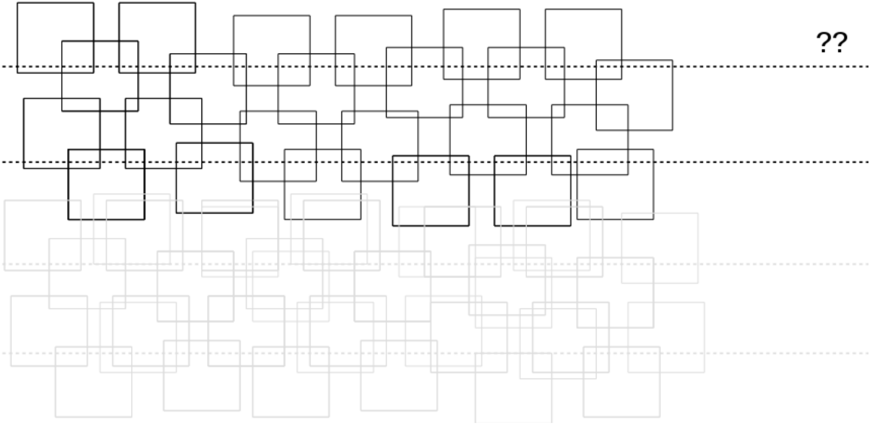
Local structure of cubicity 2 graphs



Local structure of cubicity 2 graphs



Local structure of cubicity 2 graphs



2SUIG graphs (2-stab unit interval graph)

A graph is a 2SUIG graph if it has intersection representation such that

- Two stab lines in $1 + \epsilon$ distance apart.

2SUIG graphs (2-stab unit interval graph)

A graph is a 2SUIG graph if it has intersection representation such that

- Two stab lines in $1 + \epsilon$ distance apart.
- Axes-parallel unit squares.

2SUIG graphs (2-stab unit interval graph)

A graph is a 2SUIG graph if it has intersection representation such that

- Two stab lines in $1+\epsilon$ distance apart.
- Axes-parallel unit squares.
- Each unit square intersects exactly one stab line.

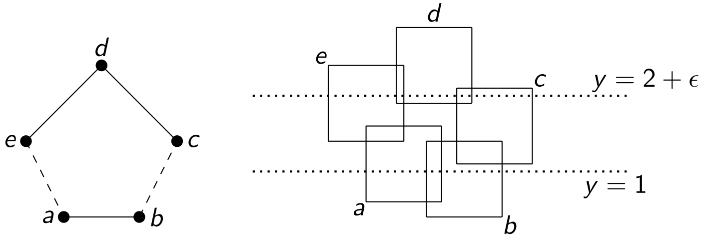


Figure: A representation (right) of a 2*SUIG* graph (left).

So, let us try to recognize Tree $\cap$ 2SUIG $=$ Tree-2SUIG graphs

In fact we prove that $\cdots$

Tree-2SUIG recognition Algorithm: Overview

Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Tree-2SUIG recognition Algorithm: Overview

Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Prove a set of necessary conditions for trees to have a 2SUIG representation.

Tree-2SUIG recognition Algorithm: Overview

Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Prove a set of necessary conditions for trees to have a 2SUIG representation.

Prove that checking those conditions can be done in linear time.

Tree-2SUIG recognition Algorithm: Overview

Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Prove a set of necessary conditions for trees to have a 2SUIG representation.

Prove that checking those conditions can be done in linear time.

If the input tree satisfies all those conditions, output a 2SUIG representation of the tree.

Tree-2SUIG recognition Algorithm: Overview

Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Prove a set of necessary conditions for trees to have a 2SUIG representation.

Prove that checking those conditions can be done in linear time.

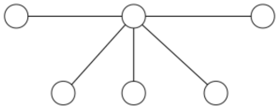
If the input tree satisfies all those conditions, output a 2SUIG representation of the tree.

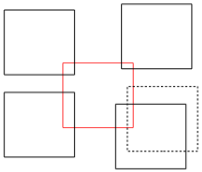
Otherwise, output a subtree that violets a particular condition.

Tree-2SUIG recognition Algorithm: Overview

Lemma

If a tree T has a 2SUIG representation then $\Delta(T) \leq 4$, $\Delta(T)$ is the maximum degree of T.





Tree-2SUIG recognition Algorithm: Overview

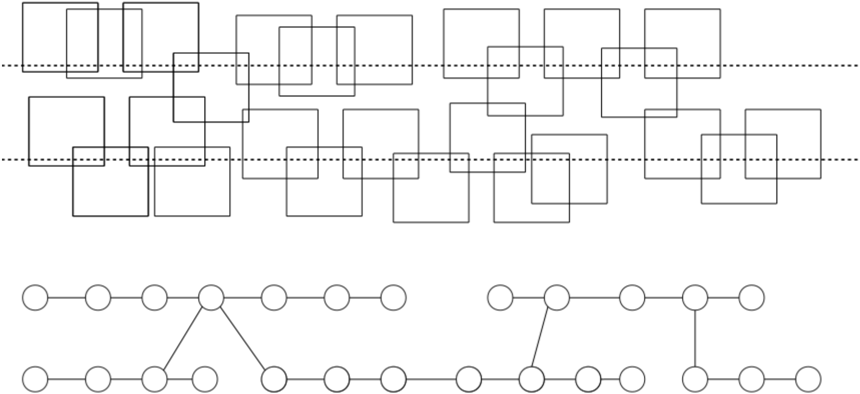


Figure: A typical tree-2SUIG.

Tree-2SUIG recognition Algorithm: Overview

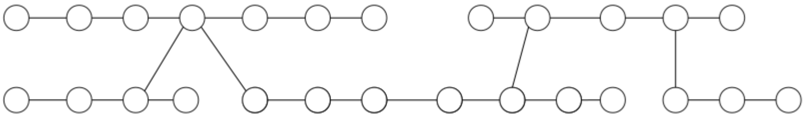
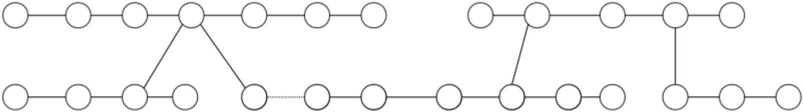


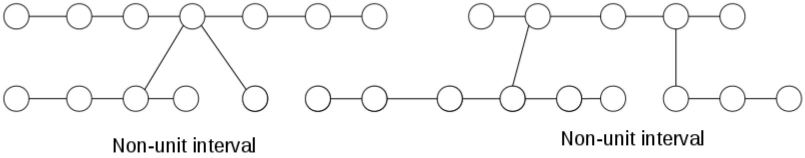
Figure: A typical tree-2SUIG.

Tree-2SUIG recognition Algorithm: Overview



Take an edge and delete it.

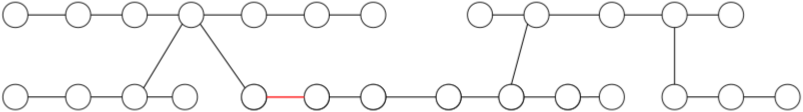
Tree-2SUIG recognition Algorithm: Overview



Take an edge and delete it.

If both sub trees are non-unit interval,

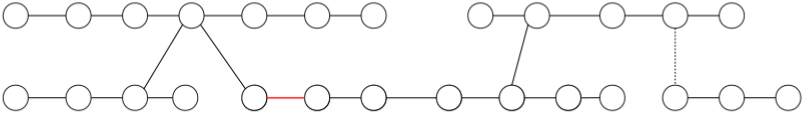
Tree-2SUIG recognition Algorithm: Overview



Take an edge and delete it.

If both sub trees are non-unit interval, call that edge as *red edge*.

Tree-2SUIG recognition Algorithm: Overview

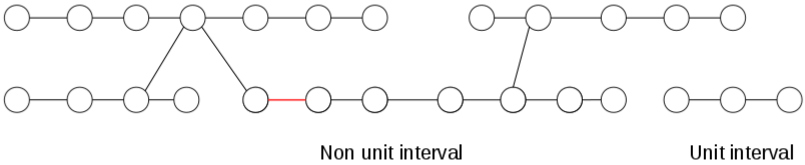


Take an edge and delete it.

If both sub trees are non-unit interval, call that edge as *red edge*.

Take another edge and delete it.

Tree-2SUIG recognition Algorithm: Overview



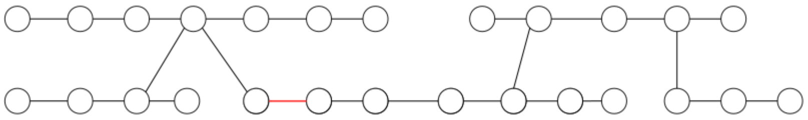
Take an edge and delete it.

If both sub trees are non-unit interval, call that edge as *red edge*.

Take another edge and delete it.

In this case,

Tree-2SUIG recognition Algorithm: Overview



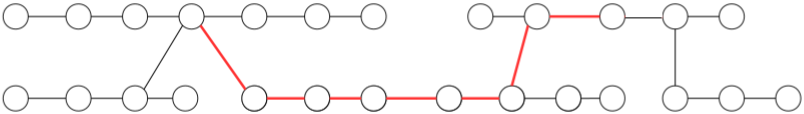
Take an edge and delete it.

If both sub trees are non-unit interval, call that edge as *red edge*.

Take another edge and delete it.

In this case, do nothing.

Tree-2SUIG recognition Algorithm: Overview



Take an edge and delete it.

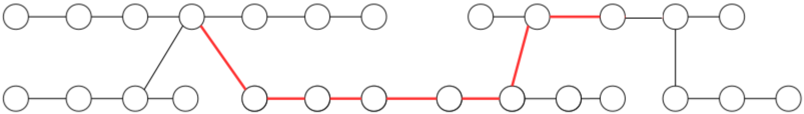
If both sub trees are non-unit interval, call that edge as *red edge*.

Take another edge and delete it.

In this case, do nothing.

Get the set of all red edges.

Tree-2SUIG recognition Algorithm: Overview



Take an edge and delete it.

If both sub trees are non-unit interval, call that edge as *red edge*.

Take another edge and delete it.

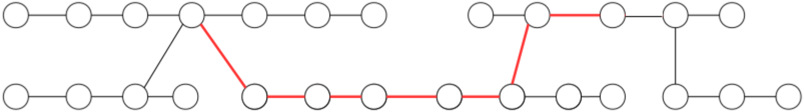
In this case, do nothing.

Get the set of all red edges.

Lemma

If a tree T has a 2SUIG representation, the set of red edges induces a connected path.

Tree-2SUIG recognition Algorithm: Overview



Delete the vertices incident to a red edge.

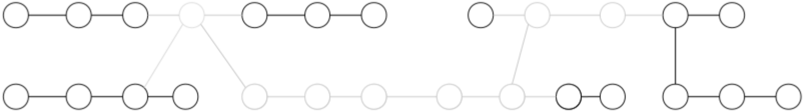
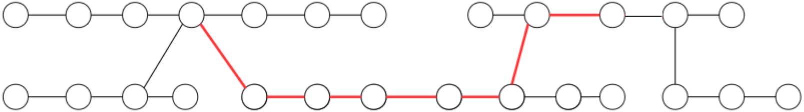


Figure: Only paths are left.

Tree-2SUIG recognition Algorithm: Overview



Delete the vertices incident to a red edge.

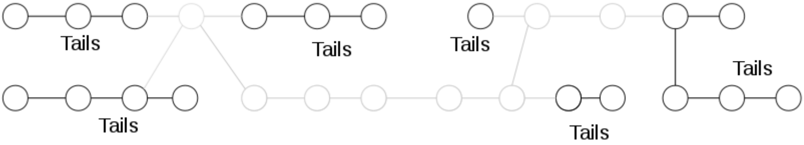


Figure: Only paths are left

Tree-2SUIG recognition Algorithm: Overview

Assuming there are red edges,

Represent the red path appropriately.

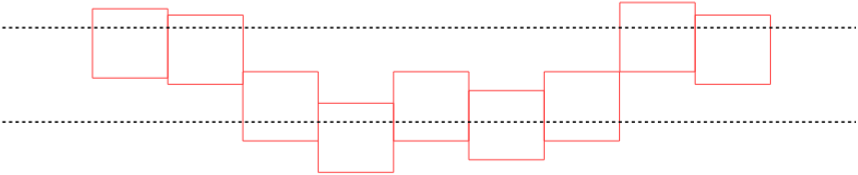


Figure: Stretched representation of red path.

Tree-2SUIG recognition Algorithm: Overview

Assuming there are red edges,

Represent the red path (Stretched representation).

Represent the tails appropriately (Shrinked representation).

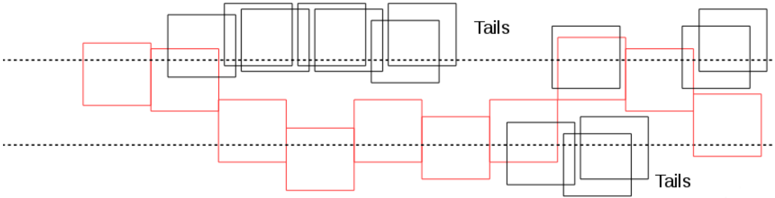


Figure: Shrinked representation of tails.

Length of the tails imposes restriction.

Tree-2SUIG recognition Algorithm: Overview

Assuming there are red edges,

Stretched representation of the red path.

$$+$$

Shrinked representation of the tails.

$$\Downarrow$$

Canonical representation of tree-2SUIG.

Tree-2SUIG recognition Algorithm: Overview

Idea is similar, even when there are no red edges.

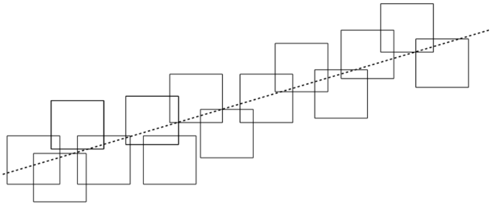
Theorem

There is a $O(n)$ time algorithm to recognize tree-2SUIG graphs.

Open problems

1. Determine the complexity of recognizing trees with cubicity 2.

2. Recognize those graphs whose intersection representation consists of squares, intersecting a common line, not necessarily axes parallel.



3. Recognize 2SUIG graphs.

Thank You