Forbidden-set connectivity - A compact-labeled approach.

Cyril Gavoille, Pierre Halftermeyer

LaBRI, Université de Bordeaux 1

November 22, 2012
1 Introduction
   ■ Forbidden-set connectivity problem
   ■ Approach
   ■ A compact labeled approach

2 Our result
   ■ Theorem
   ■ Sketch of proof

3 Conclusion
1 Introduction
   - Forbidden-set connectivity problem
   - Approach
   - A compact labeled approach

2 Our result
   - Theorem
   - Sketch of proof

3 Conclusion
A simple problem

Let $G = (V, E)$ be a graph
A simple problem

Problem

- Let $G = (V, E)$ be a graph
- Let $X$ be a subset of $V \cup E$ (called forbidden set)
A simple problem

Problem

- Let $G = (V, E)$ be a graph
- Let $X$ be a subset of $V \cup E$ (called forbidden set)
- Let $u$ and $v$ two vertices of $V$
A simple problem

Let $G = (V, E)$ be a graph
Let $X$ be a subset of $V \cup E$ (called forbidden set)
Let $u$ and $v$ two vertices of $V$
Are $u$ and $v$ in the same connected component of $V \setminus X$?
FS-connectivity problem
A natural question

- Computer network
A natural question

- Computer network
- cable cut
A natural question

- Computer network
  - cable cut
  - down router
A natural question

- Computer network
  - cable cut
  - down router
  - malicious attack
A natural question

- Computer network
  - cable cut
  - down router
  - malicious attack
- Physical network
A natural question

- Computer network
  - cable cut
  - down router
  - malicious attack
- Physical network
  - Natural disaster
A natural question

- Computer network
  - cable cut
  - down router
  - malicious attack

- Physical network
  - Natural disaster
  - Public works on roads
A natural question

- Computer network
  - cable cut
  - down router
  - malicious attack
- Physical network
  - Natural disaster
  - Public works on roads
- Etc.
If $X = \emptyset$, it’s easy to calculate in time $O(n)$ a data structure which answers the connected component of a vertex in $O(1)$ time.
If $X = \emptyset$, it’s easy to calculate in time $O(n)$ a data structure which answers the connected component of a vertex in $O(1)$ time.

If $|X| = 1$, too. By studying bridges an articulation points.
If $X = \emptyset$, it’s easy to calculate in time $O(n)$ a data structure which answers the connected component of a vertex in $O(1)$ time.

If $|X| = 1$, too. By studying bridges an articulation points.

If $|X| = k \geq 2$, the problem is harder to solve. (Almost optimally solved for forbidden sets of edges. [Patrascu, Thorup ’07]
Introduction

Approach

\(X = \emptyset, |X| = 1, |X| = 2, \ldots\)

- If \(X = \emptyset\), it’s easy to calculate in time \(O(n)\) a data structure which answers the connected component of a vertex in \(O(1)\) time.
- If \(|X| = 1\), too. By studying bridges an articulation points.
- If \(|X| = k \geq 2\), the problem is harder to solve. (Almost optimally solved for forbidden sets of edges. [Patrascu, Thorup ’07]
- We naturally aim to create a data structure that answers a connectivity-query in time \(O(k)\).
If $X = \emptyset$, it’s easy to calculate in time $O(n)$ a data structure which answers the connected component of a vertex in $O(1)$ time.

If $|X| = 1$, too. By studying bridges an articulation points.

If $|X| = k \geq 2$, the problem is harder to solve. (Almost optimally solved for forbidden sets of edges. [Patrascu, Thorup ’07]

We naturally aim to create a data structure that answers a connectivity-query in time $O(k)$.

Idea : Let’s associate a small amount of data to each vertex.
Compact labeling

What is a compact labeling scheme?

**Definition**

A compact labeling scheme for a query \( Q(x_1, x_2, \ldots, x_n) \) is composed of:

1. A labeling algorithm that computes a compact label \( L(x) \) to each vertex \( x \) of the graph.
Compact labeling

What is a compact labeling scheme?

**Definition**

A compact labeling scheme for a query \( Q(x_1, x_2, \cdots, x_n) \) is composed of:

1. A labeling algorithm that computes a compact label \( L(x) \) to each vertex \( x \) of the graph.

2. A query algorithm that answers the query from the labels of the vertices related to the query: \( L(x_1), L(x_2), \cdots, L(x_n) \).
Known result

Theorem (Courcelle et al. ’08)

The family of 3-connected planar graphs admits a compact labeling scheme in which:

- Labels are computed in time $O(n)$
Known result

Theorem (Courcelle et al. ’08)

The family of 3-connected planar graphs admits a compact labeling scheme in which:

- Labels are computed in time $O(n)$
- The size of a label is $O(\log n)$ bits
Known result

Theorem (Courcelle et al. '08)

The family of 3-connected planar graphs admits a compact labeling scheme in which:

- Labels are computed in time $O(n)$
- The size of a label is $O(\log n)$ bits
- A $(u, v)$-connectivity query is answered in time $O(\log n)$
Known result

Theorem (Courcelle et al. ’08)

The family of 3-connected planar graphs admits a compact labeling scheme in which:

- Labels are computed in time $O(n)$
- The size of a label is $O(\log n)$ bits
- A $(u, v)$-connectivity query is answered in time $O(\log n)$
- after a $O(|X|^2)$-time query-preprocessing of $X$. 
1 Introduction
   - Forbidden-set connectivity problem
   - Approach
   - A compact labeled approach

2 Our result
   - Theorem
   - Sketch of proof

3 Conclusion
Our result

**Theorem**

The family of 3-connected planar genus-\(g\) graphs admits a compact labeling scheme in which:

- Labels are computed in time \(O(n)\) \(O(n + g)\) (still optimal)
Our result

Theorem

The family of 3-connected planar genus-\(g\) graphs admits a compact labeling scheme in which:

- Labels are computed in time \(O(n)\) \(O(n + g)\) (still optimal)
- The size of a label is \(O(\log n)\) \(O(g \log n)\) bits
Our result

Theorem

The family of 3-connected planar genus-\(g\) graphs admits a compact labeling scheme in which:

- Labels are computed in time \(O(n)\) \(O(n + g)\) (still optimal)
- The size of a label is \(O(\log n)\) \(O(g \log n)\) bits
- A \((u, v)\)-connectivity query is answered in optimal time \(\Theta(\log n)\) \(O(\log \log n)\)
Our result

Theorem

The family of 3-connected planar genus-\(g\) graphs admits a compact labeling scheme in which:

- Labels are computed in time \(O(n) + O(n + g)\) (still optimal)
- The size of a label is \(O(\log n) + O(g \log n)\) bits
- A \((u, v)\)-connectivity query is answered in optimal time \(O(\log n) + O(\log \log n)\)
- after a \(O(|X|^2)\)-time \(O(|X| \cdot \log n)\)-time query-preprocessing of \(X\).
Our result

Theorem

The family of 3-connected planar genus-$g$ graphs admits a compact labeling scheme in which :

- Labels are computed in time $O(n + g)$
- The size of a label is $O(g \log n)$ bits
- A $(u, v)$-connectivity query is answered in optimal time $O(\log \log n)$
- After a $O(|X| \log n)$-time query-preprocessing of $X$.

Both theorems extend to planar graphs. Complexity analysis has not been done yet.
Let $G$ a genus-$g$ graph embedded in a genus-$g$ surface (orientable or not).

- We compute in $O(n + g)$-time a cut-graph $C$ of the surface. A cut-graph $C$ is a subgraph of $G$ such that the surface $M$ obtained by cutting along the edges of $C$ is a disc.
Our result

Let $G$ a genus-$g$ graph embedded in a genus-$g$ surface (orientable or not).

- We compute in $O(n + g)$-time a cut-graph $C$ of the surface. A cut-graph $C$ is a subgraph of $G$ such that the surface $M$ obtained by cutting along the edges of $C$ is a disc.
- We can ensure $C$ to be a bounded-degree spanning tree of $G$ to which $O(g)$ edges are added.

Cyril Gavoille, Pierre Halftermeyer
Our result

Sketch of proof

scheme in trees

Let $G$ a genus-$g$ graph embedded in a genus-$g$ surface (orientable or not).

- We compute in $O(n + g)$-time a cut-graph $C$ of the surface. A cut-graph $C$ is a subgraph of $G$ such that the surface $M$ obtained by cutting along the edges of $C$ is a disc.
- We can ensure $C$ to be a bounded-degree spanning tree of $G$ to which $O(g)$ edges are added.
- $M$'s skeleton is a planar graph whose vertices lie on the border, so it's an outerplanar graph. Its number of vertices is $O(gn)$
Our result

Sketch of proof

We construct a specific forbidden-set connectivity labeling scheme for both graphs $M$ and $C$. 
Sketch of proof

- We construct a specific forbidden-set connectivity labeling scheme for both graphs $M$ and $C$.
- For the outerplanar graph $M$, we obtain labels of $O(\log(gn)) = O(\log n)$ bits. The scheme is more efficient than Courcelle’s general planar one.
Sketch of proof

- We construct a specific forbidden-set connectivity labeling scheme for both graphs $M$ and $C$.
- For the outerplanar graph $M$, we obtain labels of $O(\log(gn)) = O(\log n)$ bits. The scheme is more efficient than Courcelle’s general planar one.
- For the cut-graph $C$ we develop an ad-hoc scheme with $O(g \log n)$-bit labels.
We construct a specific forbidden-set connectivity labeling scheme for both graphs $M$ and $C$.

For the outerplanar graph $M$, we obtain labels of $O(\log(\sqrt{n})) = O(\log n)$ bits. The scheme is more efficient than Courcelle’s general planar one.

For the cut-graph $C$ we develop an ad-hoc scheme with $O(g \log n)$-bit labels.

Both constructions proceed by reducing to a FS-connectivity labeling-scheme in trees.
Sketch of proof

- We construct a specific forbidden-set connectivity labeling scheme for both graphs $M$ and $C$.
- For the outerplanar graph $M$, we obtain labels of $O(\log(gn)) = O(\log n)$ bits. The scheme is more efficient than Courcelle’s general planar one.
- For the cut-graph $C$ we develop an ad-hoc scheme with $O(g \log n)$-bit labels.
- Both constructions proceed by reducing to a FS-connectivity labeling-scheme in trees.
- Our construction then uses a meta-scheme that allows to find the connected component of $u$ in $G \setminus X$ by querying in $M \setminus X$ and $C \setminus X$. Our meta-scheme query preprocessing of $X$ costs $O(|X| + g)$ once $X$ has been sorted.
Some details of the proof

- From outerplanar to trees.
Some details of the proof

- From outerplanar to trees.
- A scheme in trees.
outerplanar to tree
outerplanar to tree
scheme in trees
Forbidden-set connectivity - A compact-labeled approach.
Forbidden-set connectivity - A compact-labeled approach.

- Our result
- Sketch of proof
Forbidden-set connectivity - A compact-labeled approach.

Our result

Sketch of proof

scheme in trees

PREDECESSOR
ROUTING
Forbidden-set connectivity - A compact-labeled approach.

Our result

Sketch of proof

scheme in trees

- PREDECESSOR in a universe of $n$ integer is answered in time $O(\log \log n)$. 

PREDECESSOR 21 : 16
ROUTING :
scheme in trees

- PREDECESSOR in a universe of $n$ integer is answered in time $O(\log \log n)$.
- ROUTING in trees can be done in $O(1)$ time with $O(\log n)$-bit labels. [Fraignaud, Gavoille]
1 Introduction
   - Forbidden-set connectivity problem
   - Approach
   - A compact labeled approach

2 Our result
   - Theorem
   - Sketch of proof

3 Conclusion
The label length of $O(g \log n)$-bits can conceivably be improved.
Conclusion

- The label length of $O(g \log n)$-bits can conceivably be improved.
- One can show that $\Omega(\sqrt{g} + \log n)$-bit labels are required.
The label length of $O(g \log n)$-bits can conceivably be improved.

One can show that $\Omega(\sqrt{g} + \log n)$-bit labels are required.

We leave open the problem of determining the optimal label length for genus-$g$ graphs. Our meta-scheme may help.