An Exact Algorithm to Check the Existence of (Elementary) Paths and a Generalisation of the Cut Problem in Graphs with Forbidden Transitions.

Benjamin MOMÈGE [1]

Clermont-Université, Université Blaise Pascal, LIMOS, CNRS, France
momege@isima.fr

SOFSEM 2013

Coauthors : Mamadou M. KANTÉ, Chistian LAFOREST

1. B. Momège has a PhD grant from CNRS and région Auvergne.

Part 1 :
An Exact Algorithm to Check the Existence of Elementary Path between two given vertices in a Graph with Forbidden Transitions.

Definition

A transition $=$ *A couple of adjacent edges.*

Definition

A transition = *A couple of adjacent edges.*
Graph with Forbidden Transitions = *Graph + Set of transitions (called forbidden transitions).*

Definition

A transition $=$ *A couple of adjacent edges.*
Graph with Forbidden Transitions $=$ *Graph* $+$ *Set of transitions (called forbidden transitions).*
Path on ℓ **vertices** $=$ *sequence of* ℓ *vertices such that :*

Definition

A transition $=$ *A couple of adjacent edges.*
Graph with Forbidden Transitions $=$ *Graph* $+$ *Set of transitions (called forbidden transitions).*
Path on ℓ **vertices** $=$ *sequence of* ℓ *vertices such that :*
- Two consecutive vertices form an edge,

Definition

A transition $=$ *A couple of adjacent edges.*
Graph with Forbidden Transitions $=$ *Graph + Set of transitions (called forbidden transitions).*
Path on ℓ **vertices** $=$ *sequence of* ℓ *vertices such that :*
- *Two consecutive vertices form an edge,*
- *Two consecutive edges do not form a forbidden transition.*

Definition

A transition $=$ *A couple of adjacent edges.*
Graph with Forbidden Transitions $=$ *Graph + Set of transitions (called forbidden transitions).*
Path on ℓ vertices $=$ *sequence of ℓ vertices such that :*
- Two consecutive vertices form an edge,
- Two consecutive edges do not form a forbidden transition.
Elementary Path $=$ *Path with distinct vertices.*

Definition

A transition $=$ *A couple of adjacent edges.*
Graph with Forbidden Transitions $=$ *Graph + Set of transitions (called forbidden transitions).*
Path on ℓ vertices $=$ *sequence of ℓ vertices such that :*
- Two consecutive vertices form an edge,
- Two consecutive edges do not form a forbidden transition.
Elementary Path $=$ *Path with distinct vertices.*

Concrete Example :
A city in which it is prohibited to use two consecutive streets leading to the same intersection, such as a ban to turn left.

We consider the following problem :
Instance : A graph G with forbidden transitions and two vertices x and y.
Question : Is there an **elementary** path between x and y ?

We consider the following problem :
Instance : A graph G with forbidden transitions and two vertices x and y.
Question : Is there an **elementary** path between x and y ?

Theorem (Szeider, 2003)

The above problem is NP-complete

Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. Discrete Applied Mathematics, 126(2-3) :261-273, 2003.

We consider the following problem :
Instance : A graph G with forbidden transitions and two vertices x and y.
Question : Is there an **elementary** path between x and y ?

Theorem (Szeider, 2003)

The above problem is NP-complete

Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. Discrete Applied Mathematics, 126(2-3) :261-273, 2003.

We obtained the following result :

Theorem

We can solve the above problem in time $\mathcal{O}^*(2^{|V_G|})$.

We consider the following problem :
Instance : A graph G with forbidden transitions and two vertices x and y.
Question : Is there an **elementary** path between x and y ?

Theorem (Szeider, 2003)

The above problem is NP-complete

Stefan Szeider. Finding paths in graphs avoiding forbidden transitions. Discrete Applied Mathematics, 126(2-3) :261-273, 2003.

We obtained the following result :

Theorem

We can solve the above problem in time $\mathcal{O}^*(2^{|V_G|})$.

Following is dedicated to the proof of this result.

Principle of proof : Expressing the number of elementary paths according to the numbers of paths verifying some properties, which are easy to compute.

Principle of proof : Expressing the number of elementary paths according to the numbers of paths verifying some properties, which are easy to compute.

$G =$ graph with forbidden transitions,
$n = |V_G|$,
$Y \subseteq V_G$,
$v, v' \in V_G$,

$P_{\ell,Y}(v, v') =$ Paths (not necessarily elementary) on ℓ vertices, between v and v' that do not intersect Y.

Principle of proof : Expressing the number of elementary paths according to the numbers of paths verifying some properties, which are easy to compute.

$G =$ graph with forbidden transitions,
$n = |V_G|$,
$Y \subseteq V_G$,
$v, v' \in V_G$,

$P_{\ell,Y}(v, v') =$ Paths (not necessarily elementary) on ℓ vertices, between v and v' that do not intersect Y.

Lemma

For $1 \leq \ell \leq n$, we can compute $|P_{\ell,Y}(v, v')|$ in polynomial time.

Lemma

For $1 \leq \ell \leq n$, we can compute $|P_{\ell,Y}(v, v')|$ in polynomial time.

Proof

If v_1 or $v_\ell \in Y$ we have $|P_{\ell,Y}(v_1, v_l)| = 0$;

Lemma

For $1 \leq \ell \leq n$, we can compute $|P_{\ell,Y}(v, v')|$ in polynomial time.

Proof

If v_1 or $v_\ell \in Y$ we have $|P_{\ell,Y}(v_1, v_l)| = 0$; otherwise :

- *For $1 \leq \ell \leq 3$ we can test all sequences of ℓ vertices in polynomial time.*

Lemma

For $1 \leq \ell \leq n$, we can compute $|P_{\ell,Y}(v, v')|$ in polynomial time.

Proof

If v_1 or $v_\ell \in Y$ we have $|P_{\ell,Y}(v_1, v_l)| = 0$; otherwise :

- *For $1 \leq \ell \leq 3$ we can test all sequences of ℓ vertices in polynomial time.*
- *For $\ell \geq 4$ we have*

$$|P_{\ell,Y}(v_1, v_l)| = \sum_{v_{\ell-1} \in V_G \setminus Y} |P_{\ell,Y}(v_1, v_{\ell-1}, v_\ell)|$$

where for $j \geq 3$ the notation $P_{j,Y}(v_1, v_{j-1}, v_j)$ denotes the path of $P_{j,Y}(v_1, v_j)$ with v_{j-1} as penultimate vertex.

Lemma

For $1 \leq \ell \leq n$, we can compute $|P_{\ell,Y}(v, v')|$ in polynomial time.

Proof

If v_1 or $v_\ell \in Y$ we have $|P_{\ell,Y}(v_1, v_l)| = 0$; otherwise :

- *For $1 \leq \ell \leq 3$ we can test all sequences of ℓ vertices in polynomial time.*
- *For $\ell \geq 4$ we have*

$$|P_{\ell,Y}(v_1, v_l)| = \sum_{v_{\ell-1} \in V_G \setminus Y} |P_{\ell,Y}(v_1, v_{\ell-1}, v_\ell)|$$

where for $j \geq 3$ the notation $P_{j,Y}(v_1, v_{j-1}, v_j)$ denotes the path of $P_{j,Y}(v_1, v_j)$ with v_{j-1} as penultimate vertex. By induction we can compute the set

$$\left\{ |P_{\ell,Y}(v_1, v_{\ell-1}, v_\ell)| \quad | \quad (v_{\ell-1}, v_\ell) \in (V_G \setminus Y)^2 \right\}$$

from

$$\left\{ |P_{3,Y}(v_1, v_2, v_3)| \quad | \quad (v_2, v_3) \in (V(G) \setminus Y)^2 \right\}$$

in $\mathcal{O}(n^4)$ operations. □

$SP_\ell(v, v')$ = set of elementary paths on ℓ vertices between v and v',
$SP_{\ell,Y}(v, v')$ = subset of those that do not intersect Y.

$SP_\ell(v, v')$ = set of elementary paths on ℓ vertices between v and v',
$SP_{\ell,Y}(v, v')$ = subset of those that do not intersect Y.

We see that

$$|SP_\ell(v, v')| = \sum_{\substack{Y \subseteq V_G \\ |Y|=n-l}} |SP_{\ell,Y}(v, v')|.$$

$SP_\ell(v, v')$ = set of elementary paths on ℓ vertices between v and v',
$SP_{\ell,Y}(v, v')$ = subset of those that do not intersect Y.

We see that

$$|SP_\ell(v, v')| = \sum_{\substack{Y \subseteq V_G \\ |Y|=n-l}} |SP_{\ell,Y}(v, v')|.$$

With the Inclusion-Exclusion principle we obtain for $|Y| = n - \ell$

Lemma

$$\left|SP_{\ell,Y}(v, v')\right| = \sum_{X \subseteq V_G \setminus Y} (-1)^{|X|} \left|P_{\ell,Y\cup X}(v, v')\right|.$$

$SP_\ell(v, v')$ = set of elementary paths on ℓ vertices between v and v',
$SP_{\ell,Y}(v, v')$ = subset of those that do not intersect Y.

We see that

$$|SP_\ell(v, v')| = \sum_{\substack{Y \subseteq V_G \\ |Y|=n-l}} |SP_{\ell,Y}(v, v')|.$$

With the Inclusion-Exclusion principle we obtain for $|Y| = n - \ell$

Lemma

$$\left|SP_{\ell,Y}(v, v')\right| = \sum_{X \subseteq V_G \setminus Y} (-1)^{|X|} \left|P_{\ell,Y\cup X}(v, v')\right|.$$

Substituting we get :

$$\left|SP_\ell(v, v')\right| = \sum_{\substack{Y \subseteq V_G \\ |Y|=n-\ell}} \left|SP_{\ell,Y}(v, v')\right| = \sum_{\substack{Y \subseteq V(G) \\ |Y|=n-\ell}} \sum_{X \subseteq V_G \setminus Y} (-1)^{|X|} \left|P_{\ell,Y\cup X}(v, v')\right|.$$

$$\left|SP_\ell(v,v')\right| = \sum_{\substack{Y\subseteq V_G \\ |Y|=n-\ell}} \sum_{X\subseteq V_G\setminus Y} (-1)^{|X|} \left|P_{\ell,Y\cup X}(v,v')\right|.$$

Now making the change of variable

$$A \leftarrow Y \cup X$$

noting that the number of partitions $Y \sqcup X$ of A with $|Y| = n-\ell$ is

$$\binom{|A|}{n-\ell}$$

we get :

Proposition

$$\left|SP_\ell(v,v')\right| = \sum_{|A|\geq n-\ell} \binom{|A|}{n-\ell} (-1)^{|A|-(n-\ell)} \left|P_{\ell,A}(v,v')\right|$$

Proposition

$$\left|SP_\ell(v, v')\right| = \sum_{|A| \geq n-\ell} \binom{|A|}{n-\ell} (-1)^{|A|-(n-\ell)} \left|P_{\ell,A}(v, v')\right|$$

Proposition

$$\left|SP_\ell(v,v')\right| = \sum_{|A|\geq n-\ell} \binom{|A|}{n-\ell} (-1)^{|A|-(n-\ell)} \left|P_{\ell,A}(v,v')\right|$$

The number of operations required to compute $|SP_\ell(v,v')|$ is

$$\sum_{i=n-\ell}^{n} \binom{n}{i} \mathcal{O}(Poly(n))$$

Proposition

$$\left|SP_\ell(v,v')\right| = \sum_{|A|\geq n-\ell} \binom{|A|}{n-\ell} (-1)^{|A|-(n-\ell)} \left|P_{\ell,A}(v,v')\right|$$

The number of operations required to compute $|SP_\ell(v,v')|$ is

$$\sum_{i=n-\ell}^{n} \binom{n}{i} \mathcal{O}(Poly(n))$$

then to compute the number of elementary paths from v to v' are needed

$$\sum_{\ell=1}^{n} \sum_{i=n-\ell}^{n} \binom{n}{i} \mathcal{O}(Poly(n)) = \mathcal{O}(2^n.Poly(n))$$

operations.

Proposition

$$\left|SP_\ell(v,v')\right| = \sum_{|A|\geq n-\ell} \binom{|A|}{n-\ell} (-1)^{|A|-(n-\ell)} \left|P_{\ell,A}(v,v')\right|$$

The number of operations required to compute $|SP_\ell(v,v')|$ is

$$\sum_{i=n-\ell}^{n} \binom{n}{i} \mathcal{O}(Poly(n))$$

then to compute the number of elementary paths from v to v' are needed

$$\sum_{\ell=1}^{n} \sum_{i=n-\ell}^{n} \binom{n}{i} \mathcal{O}(Poly(n)) = \mathcal{O}(2^n.Poly(n))$$

operations.

Theorem

We can compute the number of elementary paths from v to v' in $\mathcal{O}^*(2^n)$ operations.

PERSPECTIVES :

PERSPECTIVES :

- Search an exact algorithm for graphs with forbidden paths of bounded length (in preparation).

PERSPECTIVES :

- Search an exact algorithm for graphs with forbidden paths of bounded length (in preparation).
- Find an exact algorithm of better complexity as $\mathcal{O}^*(2^{|V_G|})$ or a probabilistic Monte Carlo algorithm (no false positive response).

PERSPECTIVES :

- Search an exact algorithm for graphs with forbidden paths of bounded length (in preparation).
- Find an exact algorithm of better complexity as $\mathcal{O}^*(2^{|V_G|})$ or a probabilistic Monte Carlo algorithm (no false positive response).
- Watch the complexity of the problem in special classes of graphs (bounded treewidth graphs, planar graphs, ...).

Part 2 :
Generalisation of the Cut Problem in Graphs with Forbidden Transitions.

G = directed graph.
A transition in G is a triplet of vertices (a, b, c) such that ab and bc are arcs of the graph.
F_G = forbidden transitions of G.
A_G = other transitions of G = allowed transitions of G.

G = directed graph.
A transition in G is a triplet of vertices (a, b, c) such that ab and bc are arcs of the graph.
F_G = forbidden transitions of G.
A_G = other transitions of G = allowed transitions of G.

If there is a path between two vertices s and t we say that the graph is (s, t)-connected.

G = directed graph.
A transition in G is a triplet of vertices (a, b, c) such that ab and bc are arcs of the graph.
F_G = forbidden transitions of G.
A_G = other transitions of G = allowed transitions of G.

If there is a path between two vertices s and t we say that the graph is (s, t)-connected.

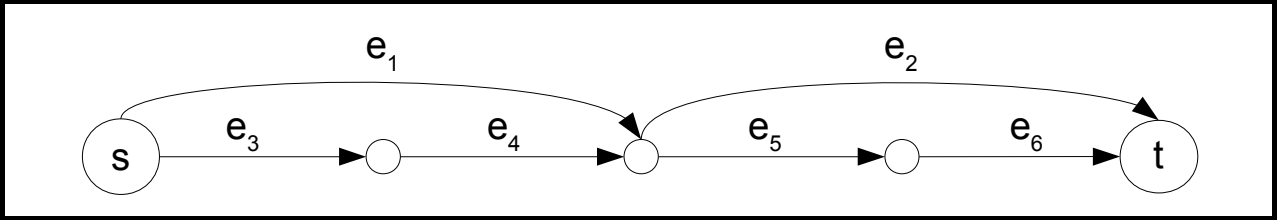
Goal : Find a minimum (cardinal) set of allowed transitions sufficient to forbid to disconnect s and t.

Definition

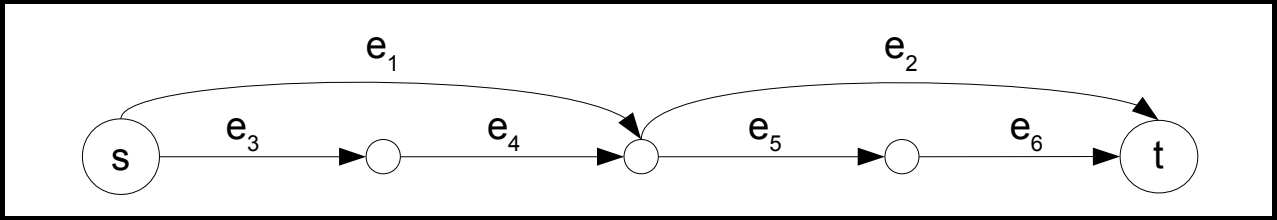
For $s, t \in V_G$, we denote by $G^*_{s,t}$ the directed graph defined by :

$$V_{G^*_{s,t}} := E_G \cup \{s'\} \cup \{t'\},$$
$$E_{G^*_{s,t}} := A_G \cup \{(s', (s, v)) \mid (s, v) \in E_G\} \cup \{((v, t), t') \mid (v, t) \in E_G\}$$

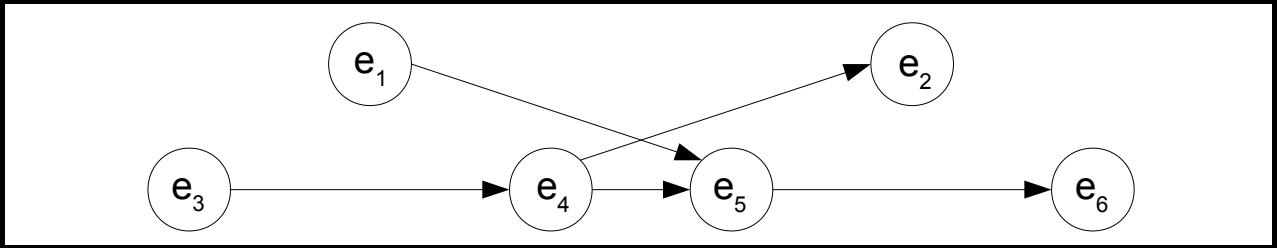
Example : With the following graph G where $F_G = \{(e_1, e_2)\}$:



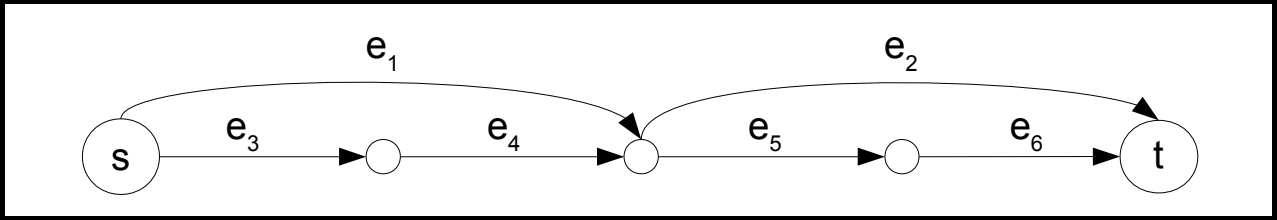
Example : With the following graph G where $F_G = \{(e_1, e_2)\}$:



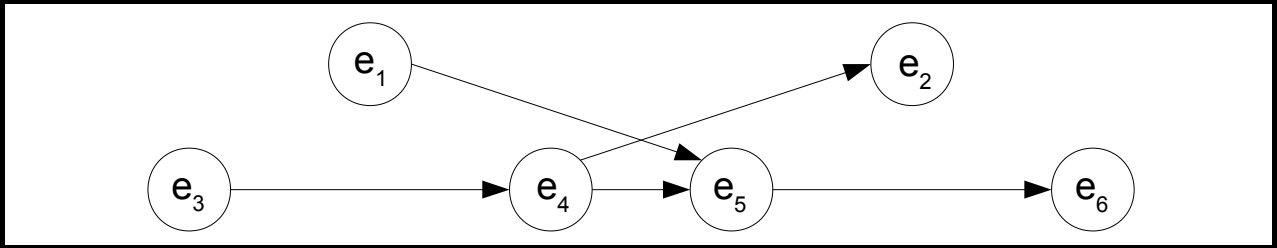
We construct an analog of the "line graph" of G where vertices are the arcs of G and the arcs are the allowed transition of G.



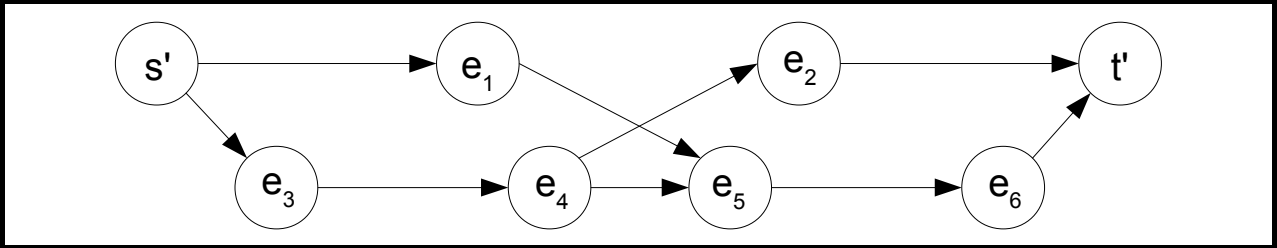
Example : With the following graph G where $F_G = \{(e_1, e_2)\}$:



We construct an analog of the "line graph" of G where vertices are the arcs of G and the arcs are the allowed transition of G.



To obtain $G^*_{s,t}$ we add two vertices s' and t' and arcs between s' and the outgoing arcs of s (in G) and between the incoming arcs of t (in G) and t'.



Proposition

The function

$$f : \begin{cases} \{(s,t)\text{-paths in } G\} & \to \{(s',t')\text{-paths in } G^*_{s,t}\} \\ (e_1 \in E_G, ..., e_k \in E_G) & \mapsto (s', e_1 \in V_{G^*_{s,t}}, ..., e_k \in V_{G^*_{s,t}}, t') \end{cases}$$

is well defined and bijective.

Proposition

The function

$$f : \begin{cases} \{(s,t)\text{-paths in } G\} & \to \{(s',t')\text{-paths in } G^*_{s,t}\} \\ (e_1 \in E_G, ..., e_k \in E_G) & \mapsto (s', e_1 \in V_{G^*_{s,t}}, ..., e_k \in V_{G^*_{s,t}}, t') \end{cases}$$

is well defined and bijective.

Corollary

G is (s,t)-connected if and only if $G^*_{s,t}$ is (s',t')-connected.

Proposition

The function

$$f : \begin{cases} \{(s,t)\text{-paths in } G\} & \to \{(s',t')\text{-paths in } G^*_{s,t}\} \\ (e_1 \in E_G, ..., e_k \in E_G) & \mapsto (s', e_1 \in V_{G^*_{s,t}}, ..., e_k \in V_{G^*_{s,t}}, t') \end{cases}$$

is well defined and bijective.

Corollary

G is (s,t)-connected if and only if $G^*_{s,t}$ is (s',t')-connected.

(s',t')-cut $=$ set of arcs disconnecting s' and t'

Proposition

The function

$$f : \begin{cases} \{(s,t)\text{-paths in } G\} & \to \{(s',t')\text{-paths in } G^*_{s,t}\} \\ (e_1 \in E_G, ..., e_k \in E_G) & \mapsto (s', e_1 \in V_{G^*_{s,t}}, ..., e_k \in V_{G^*_{s,t}}, t') \end{cases}$$

is well defined and bijective.

Corollary

G is (s,t)-connected if and only if $G^*_{s,t}$ is (s',t')-connected.

(s',t')-cut = set of arcs disconnecting s' and t'

As allowed transitions in G correspond to arcs who are neither outgoing arc of s' nor incoming arc of t' :
if we take a set of allowed transitions in G sufficient to forbid to disconnect s and t the corresponding arcs disconnect s' and t' in $G^*_{s,t}$ and thus form an (s',t')-cuts having no outgoing arc of s' and no incoming arc of t', and vice versa.

Lemma

The function that sends an allowed transition in G on the corresponding edge in $G^*_{s,t}$ induce a bijection between :
The set of sets of allowed transitions in G sufficient to forbid to disconnect s and t in G
AND
The (s', t')-cuts having no outgoing arc of s' and no incoming arc of t' in $G^*_{s,t}$

Lemma

The function that sends an allowed transition in G on the corresponding edge in $G^*_{s,t}$ induce a bijection between :
The set of sets of allowed transitions in G sufficient to forbid to disconnect s and t in G
AND
The (s', t')-cuts having no outgoing arc of s' and no incoming arc of t' in $G^*_{s,t}$

Lemma

There exists a (s', t')-cut in $G^*_{s,t}$ having no outgoing arc of s' and no incoming arc of t'. By definition the size of a cut is its number of arcs.

Lemma

The function that sends an allowed transition in G on the corresponding edge in $G^*_{s,t}$ induce a bijection between :
The set of sets of allowed transitions in G sufficient to forbid to disconnect s and t in G
AND
The (s', t')-cuts having no outgoing arc of s' and no incoming arc of t' in $G^*_{s,t}$

Lemma

There exists a (s', t')-cut in $G^*_{s,t}$ having no outgoing arc of s' and no incoming arc of t'. By definition the size of a cut is its number of arcs.

Proof

Since s and t are not adjacent in G a (s, t)-path in G takes at least one transition. By taking for each path the arc in $G^*_{s,t}$ corresponding to a transition in G we obtain a (s', t')-cut.

Theorem

Let m be the number of arcs of G^*. Assign to each arc of $G^*_{s,t}$ a capacity equal to one except for the outgoing arcs of s' and the incoming arcs of t' for which it is taken equal to m. The arcs of a (s', t')-cut of minimum capacity in $G^*_{s,t}$ correspond to a minimum set of allowed transitions in G sufficient to forbid to disconnect s and t.

Theorem

Let m be the number of arcs of G^*. Assign to each arc of $G^*_{s,t}$ a capacity equal to one except for the outgoing arcs of s' and the incoming arcs of t' for which it is taken equal to m. The arcs of a (s', t')-cut of minimum capacity in $G^*_{s,t}$ correspond to a minimum set of allowed transitions in G sufficient to forbid to disconnect s and t.

Proof

According to the second lemma, there exists a (s', t')-cut in G^* having no outgoing arc of s' and no incoming arc of t'. This cut has less than m arcs (each having a capacity equal to one) and thus admits a capacity less than m. Thus, the minimum capacity of a (s', t')-cut is strictly less than m, and therefore a (s', t')-cut of minimum capacity contains no outgoing arc of s' or incoming arc of t' because its capacity would be greater than or equal to m. Now for such a cut, the capacity is equal to its size and if its size is minimum, it corresponds to a minimum set of allowed transitions in G sufficient to forbid to disconnect s and t by the first lemma. □

Algorithm 1: GeneralizedCutProblem(G, F_G)

Data: A directed graph with forbidden transitions (G, F_G) and two non-adjacent vertices s and t.

Result: A minimum set of allowed transitions sufficient to forbid to disconnect s and t.

begin

1. Construct the graph $G^*_{s,t}$;
2. Assign to each arc of $G^*_{s,t}$ a capacity equal to one except for the outgoing arcs of s' and the incoming arcs of t' for which it is taken equal to $|E_G|$;
3. Compute a (s', t')-cut of minimum capacity in $G^*_{s,t}$;
4. **return** *Transitions of G corresponding to the arcs of the cut.*

end

Algorithm 2: GeneralizedCutProblem(G, F_G)

Data: A directed graph with forbidden transitions (G, F_G) and two non-adjacent vertices s and t.

Result: A minimum set of allowed transitions sufficient to forbid to disconnect s and t.

begin

1. Construct the graph $G^*_{s,t}$;
2. Assign to each arc of $G^*_{s,t}$ a capacity equal to one except for the outgoing arcs of s' and the incoming arcs of t' for which it is taken equal to $|E_G|$;
3. Compute a (s', t')-cut of minimum capacity in $G^*_{s,t}$;
4. **return** *Transitions of G corresponding to the arcs of the cut.*

end

Corollary

Algorithm is correct and runs in time polynomial in the size of G.

Proof

Correctness of the algorithm follows from the Theorem. Steps 1, 2, and 4 run in time $O(|E_G|^2)$ and step 3 can be done using a polynomial time algorithm which computes a (s', t')-cut of min capacity in G^* as the Edmonds-Karp algo. □

If G is an **undirected** graph with forbidden transitions.

If G is an **undirected** graph with forbidden transitions.

Apply the previous algorithm to the graph G_d which has the same paths (sequences of vertices) as G.

$$
\begin{aligned}
V_{G_d} &:= V_G, \\
E_{G_d} &:= \{(v,w),(w,v) \mid \{v,w\} \in E_G\}, \\
F_{G_d} &:= \{((v,w),(w,v)) \mid (\{v,w\},\{w,v\}) \in F_G\}.
\end{aligned}
$$

PERSPECTIVES :

PERSPECTIVES :

- Search a polynomial algorithm for the max flow problem in graphs with forbidden transitions (in preparation).

Thank you ! Questions ?