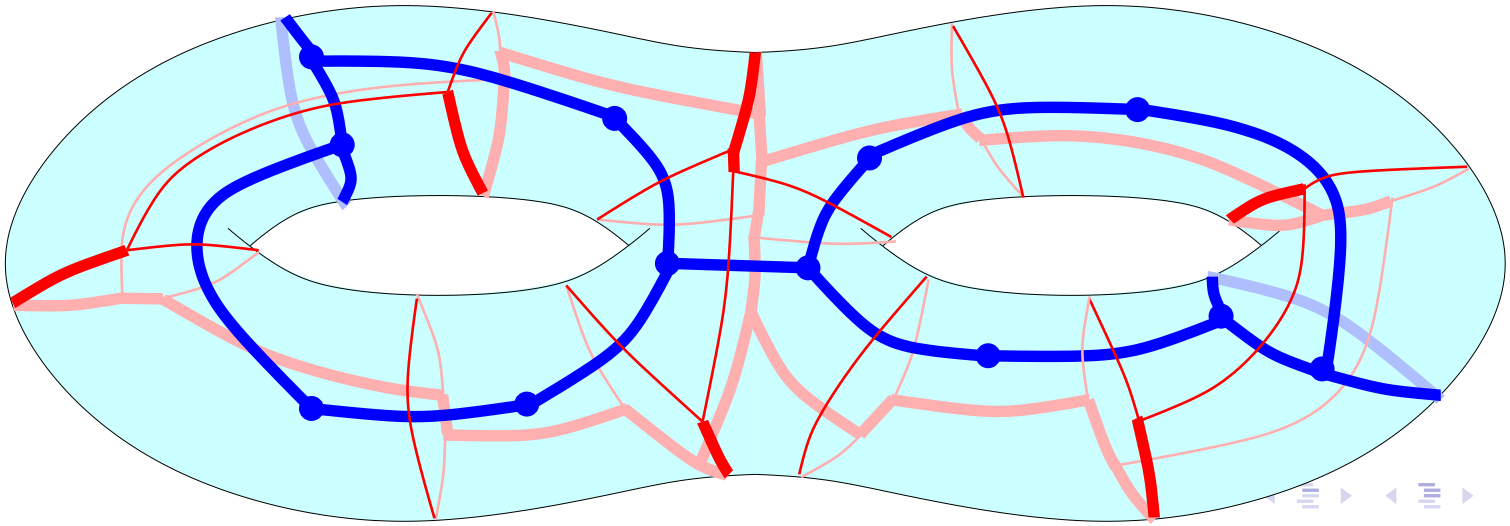
Algorithmes topologiques pour les graphes tracés sur des surfaces

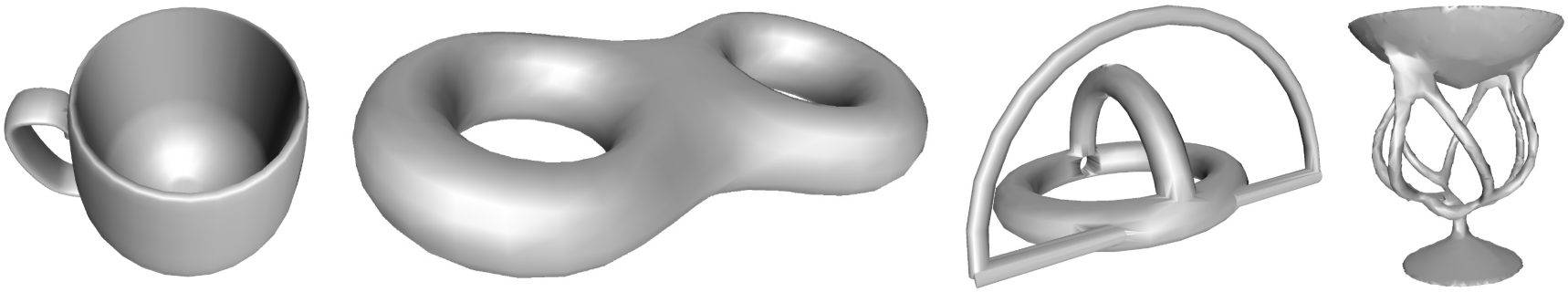
Éric Colin de Verdière

École normale supérieure (Paris) et CNRS

Journées de l'ANR EGOS, 29 octobre 2013

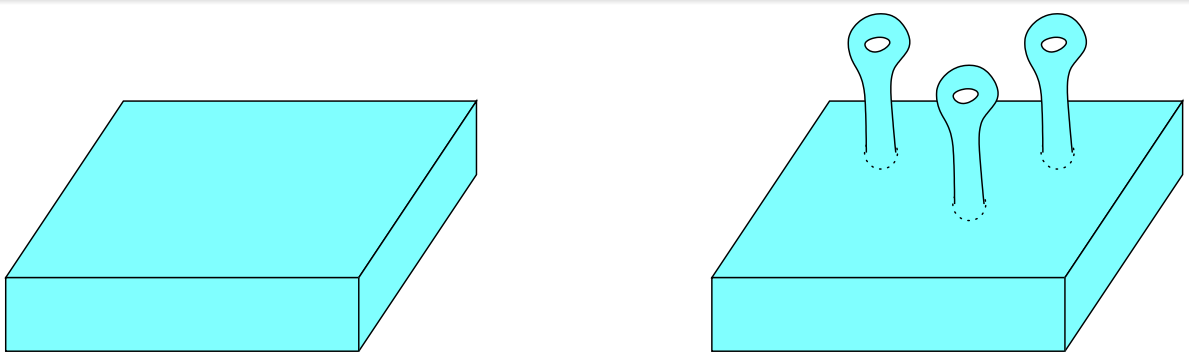


Surfaces

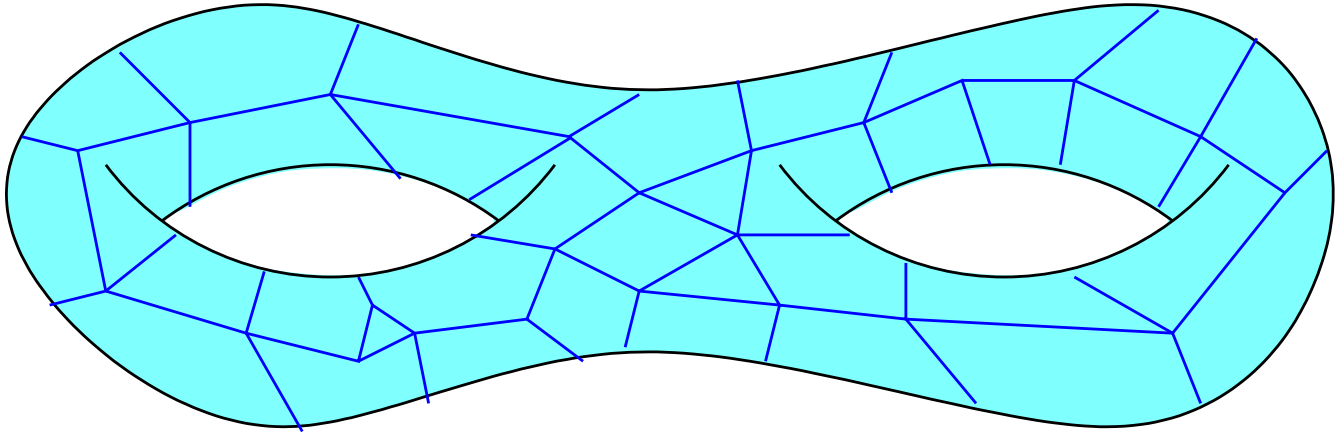


Qu'est-ce qu'une surface ?

À homéomorphisme près, $\mathscr{S}$ est une sphère à laquelle on a collé $g \geq 0$ poignées.
g est le genre de $\mathscr{S}$.

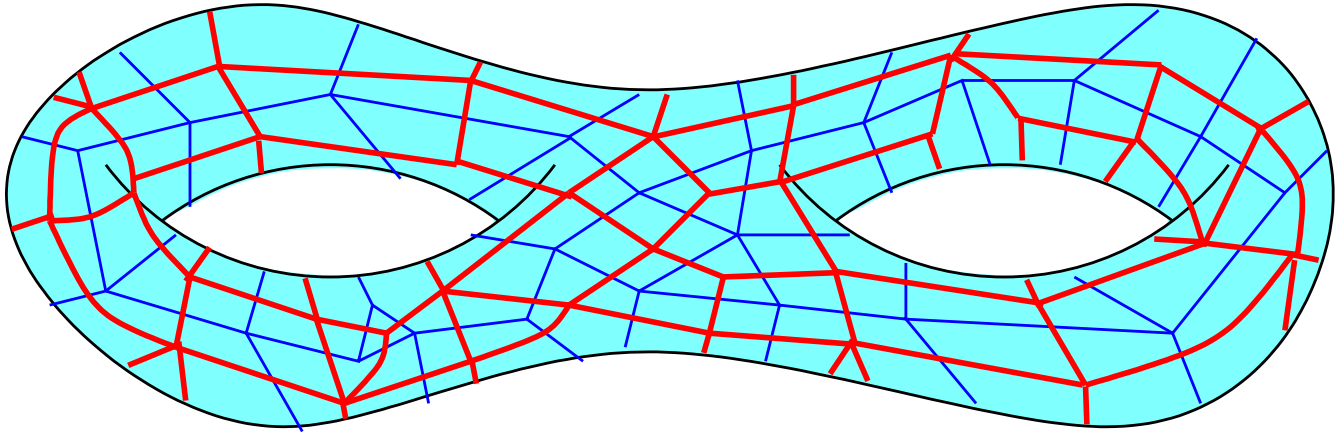


Préliminaires



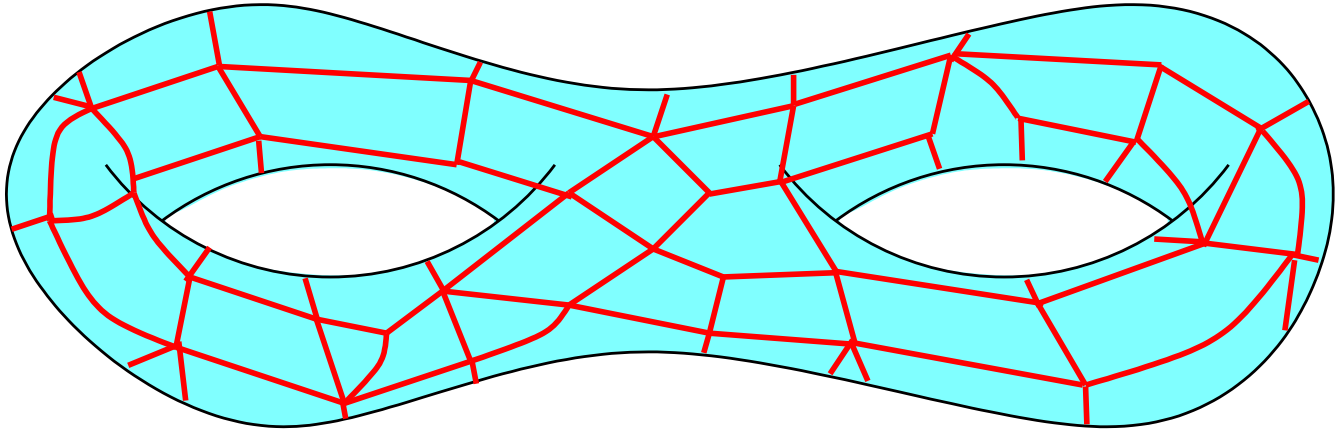
- $G = (V, E)$: graphe plongé sur $\mathscr{S}$, éventuellement à boucles et arêtes multiples.
- F : ensemble des faces de G : topologiquement, des disques ouverts.
- On supposera souvent les arêtes pondérées, à poids > 0.
- Graphe dual G^*.
- Structure de données : Graphe avec ordre cyclique des arêtes incidentes à chaque sommet.
- Courbes : *walks* dans G, éventuellement avec répétition de sommets ou d'arêtes.

Préliminaires



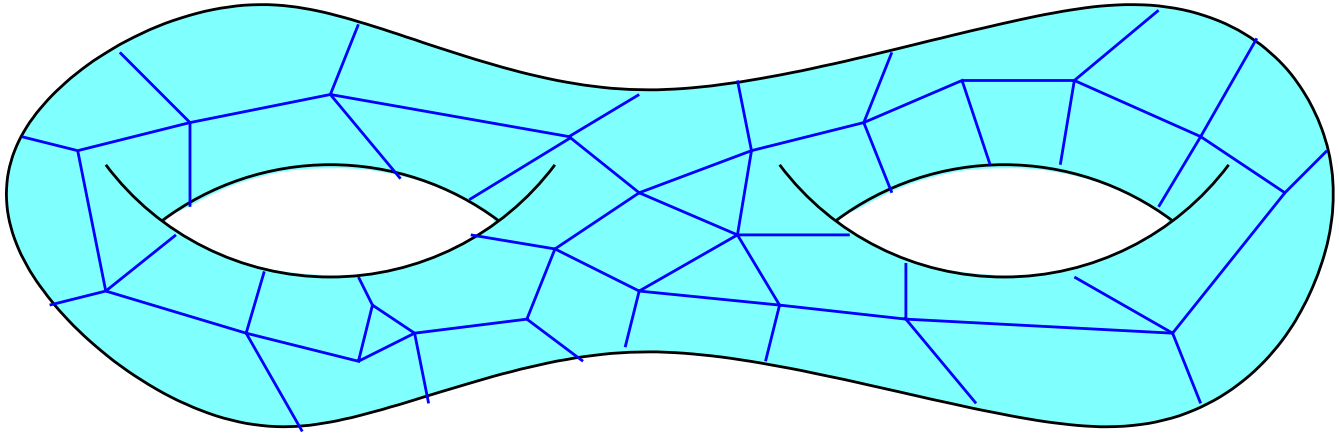
- $G = (V, E)$: graphe plongé sur $\mathscr{S}$, éventuellement à boucles et arêtes multiples.
- F : ensemble des faces de G : topologiquement, des disques ouverts.
- On supposera souvent les arêtes pondérées, à poids > 0.
- Graphe dual G^*.
- Structure de données : Graphe avec ordre cyclique des arêtes incidentes à chaque sommet.
- Courbes : *walks* dans G, éventuellement avec répétition de sommets ou d'arêtes.

Préliminaires



- $G = (V, E)$: graphe plongé sur $\mathscr{S}$, éventuellement à boucles et arêtes multiples.
- F : ensemble des faces de G : topologiquement, des disques ouverts.
- On supposera souvent les arêtes pondérées, à poids > 0.
- Graphe dual G^*.
- Structure de données : Graphe avec ordre cyclique des arêtes incidentes à chaque sommet.
- Courbes : *walks* dans G, éventuellement avec répétition de sommets ou d'arêtes.

Préliminaires

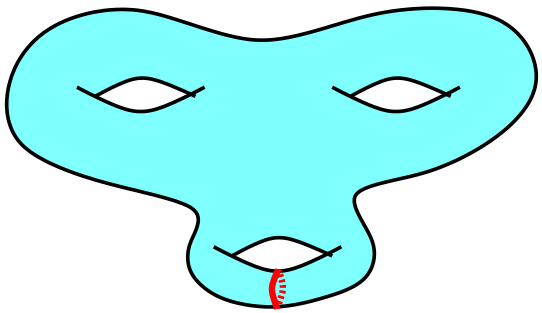


- $G = (V, E)$: graphe plongé sur $\mathscr{S}$, éventuellement à boucles et arêtes multiples.
- F : ensemble des faces de G : topologiquement, des disques ouverts.
- On supposera souvent les arêtes pondérées, à poids > 0.
- Graphe dual G^*.
- Structure de données : Graphe avec ordre cyclique des arêtes incidentes à chaque sommet.
- Courbes : *walks* dans G, éventuellement avec répétition de sommets ou d'arêtes.

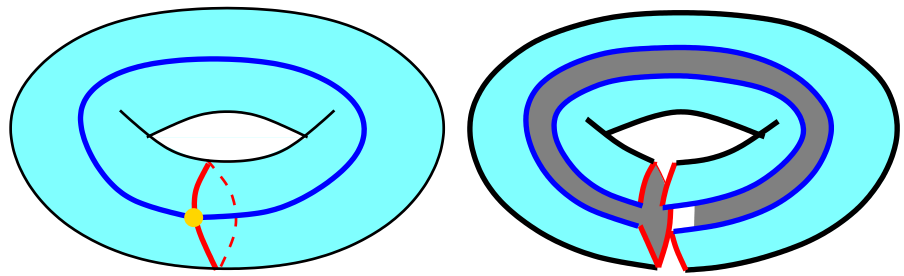
Aperçu

(a) Algorithmes efficaces pour...

1. Trouver le plus court chemin fermé dans le graphe qui forme une courbe « non triviale » sur la surface ;



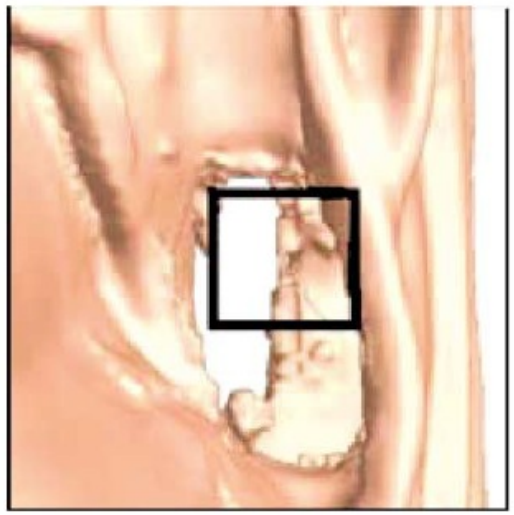
2. Trouver un sous-graphe, le plus court possible, le long duquel découper la surface pour la rendre planaire.

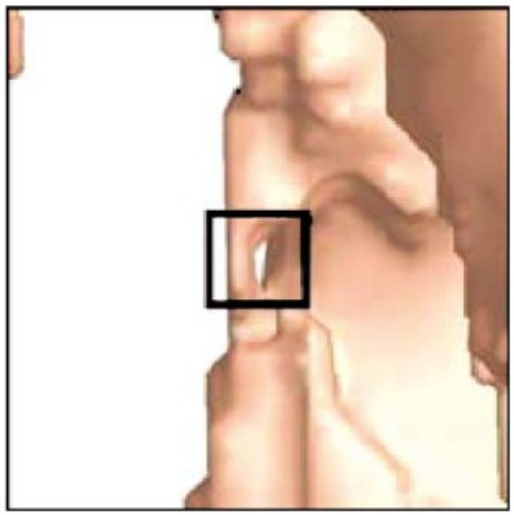


(b) Bornes pour la longueur de telles sous-structures optimales.

Applications

1. Simplification topologique, remaillage, approximation ;
2. paramétrage : plaquage de textures, compression, analyse numérique.

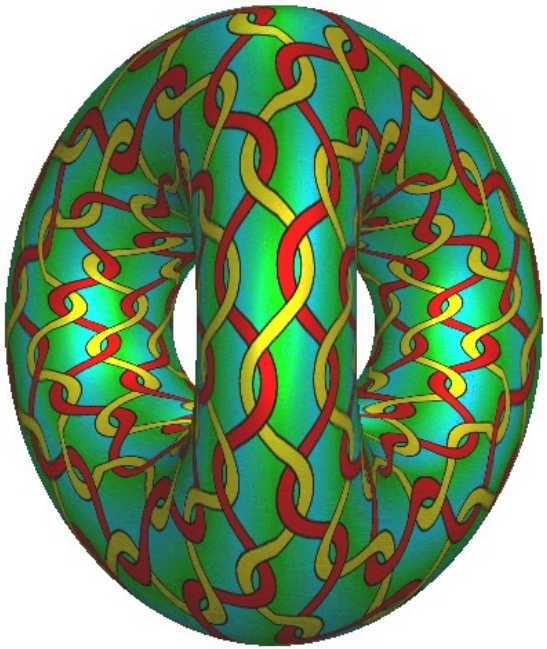




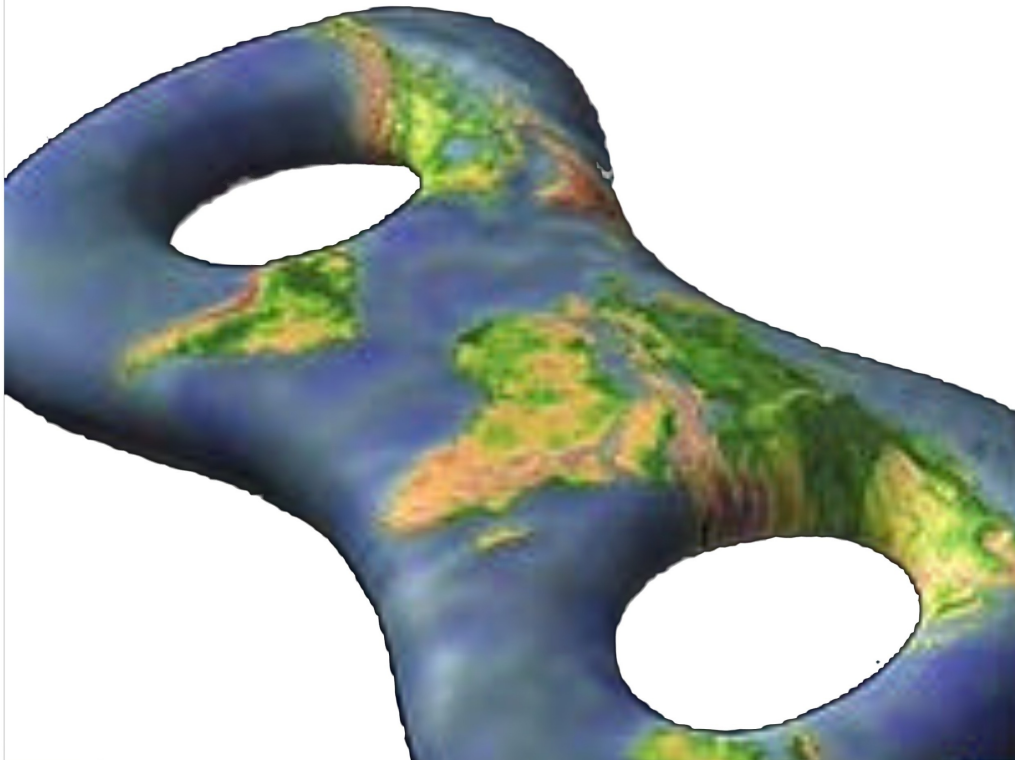
[Wood et coll., 2004]

Applications

1. Simplification topologique, remaillage, approximation ;
2. paramétrage : plaquage de textures, compression, analyse numérique.

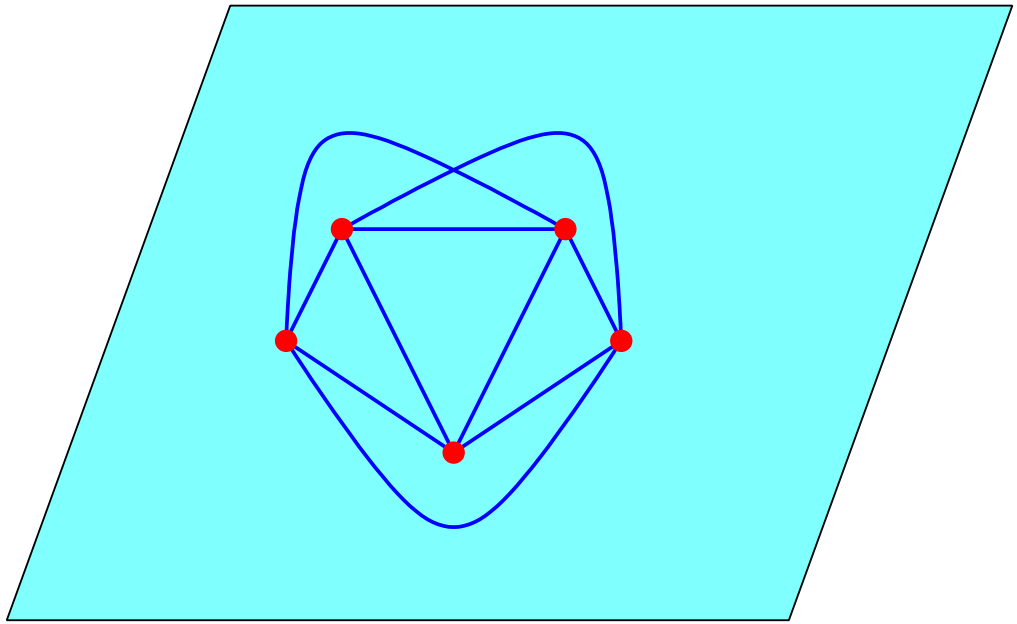


http://www.cs.berkeley.edu/~sequin/CS184/IMGS/mvs.g2.D3.gif



Théorie topologique des graphes

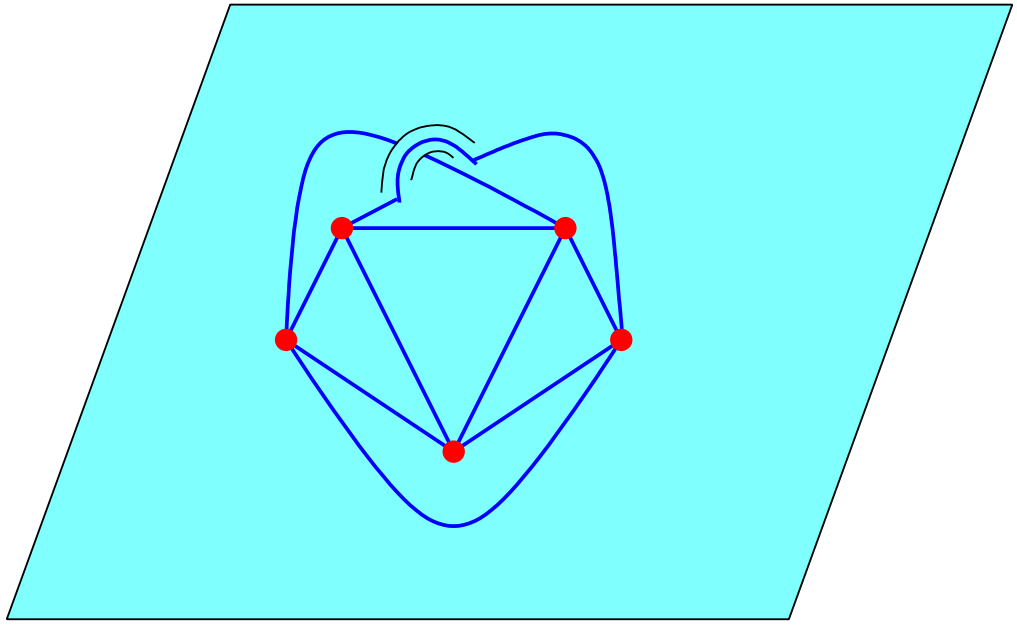
Généralisation naturelle des graphes planaires : tout graphe se plonge sur une surface.



- Plongeabilité d'un graphe sur une surface [Mohar, 1996...] ;
- *crossing number* d'un graphe dans le plan [Kawarabayashi et Reed, 2007] ;
- test d'homotopie (existence de déformation) entre courbes [Dey et Guha, 1999 ; Lazarus et Rivaud, 2012 ; Erickson et Whittlesey, 2012] ;
- raccourcissement de courbes à homotopie près [CdV, Lazarus, 2002, 2003] ; [CdV, Erickson, 2006].

Théorie topologique des graphes

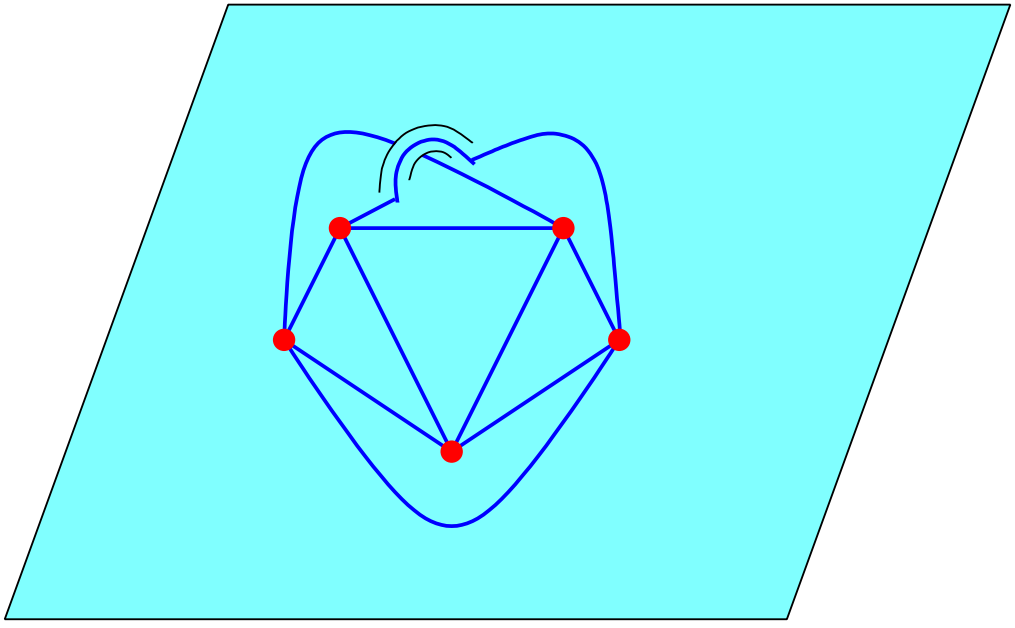
Généralisation naturelle des graphes planaires : tout graphe se plonge sur une surface.



- Plongeabilité d'un graphe sur une surface [Mohar, 1996...] ;
- *crossing number* d'un graphe dans le plan [Kawarabayashi et Reed, 2007] ;
- test d'homotopie (existence de déformation) entre courbes [Dey et Guha, 1999 ; Lazarus et Rivaud, 2012 ; Erickson et Whittlesey, 2012] ;
- raccourcissement de courbes à homotopie près [CdV, Lazarus, 2002, 2003] ; [CdV, Erickson, 2006].

Théorie topologique des graphes

Généralisation naturelle des graphes planaires : tout graphe se plonge sur une surface.



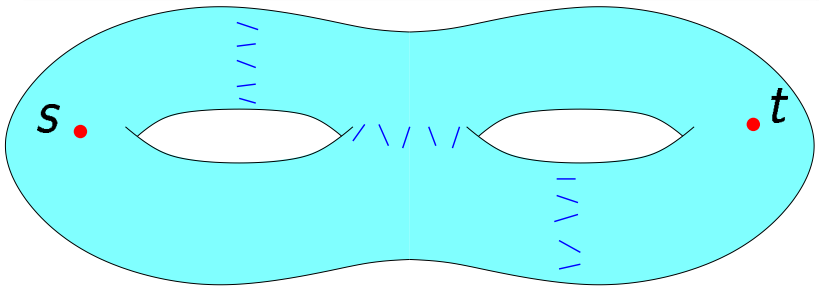
- Plongeabilité d'un graphe sur une surface [Mohar, 1996...] ;
- *crossing number* d'un graphe dans le plan [Kawarabayashi et Reed, 2007] ;
- test d'homotopie (existence de déformation) entre courbes [Dey et Guha, 1999 ; Lazarus et Rivaud, 2012 ; Erickson et Whittlesey, 2012] ;
- raccourcissement de courbes à homotopie près [CdV, Lazarus, 2002, 2003] ; [CdV, Erickson, 2006].

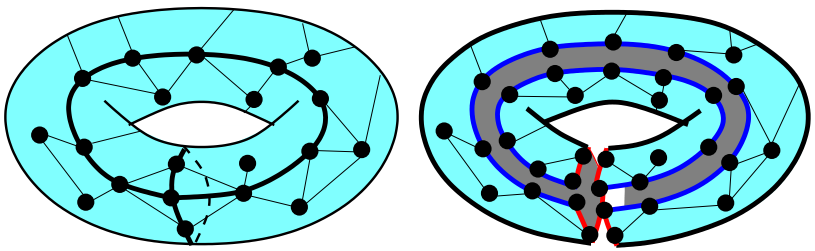
Algorithmique des graphes

Idée générale

- Prendre n'importe quel problème d'algorithmique des graphes ;
- l'étudier dans le cas spécifique des graphes plongés sur une surface.

Exemple : Coupe minimale





aussi : flots maximaux, cycles induits, isomorphismes de graphes, . . .

technique : planarisation (et/ou Robertson–Seymour, bidimensionnalité, . . .)

1. Plus courte courbe non contractile

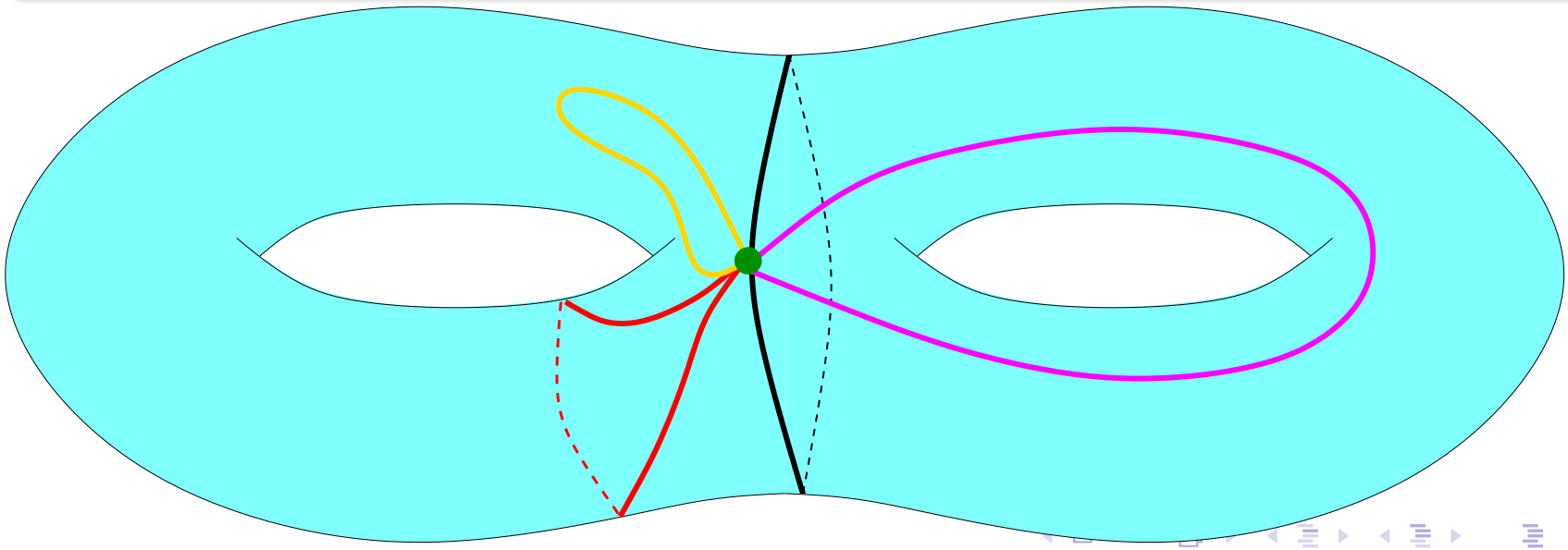
Topologie

Définitions

- Un lacet est une application continue $\ell : [0,1] \to \mathscr{S}$ avec $\ell(0) = \ell(1)$ (le point-base) ;
- ℓ est contractile si ℓ peut être déformé continûment en un lacet constant.

Problème

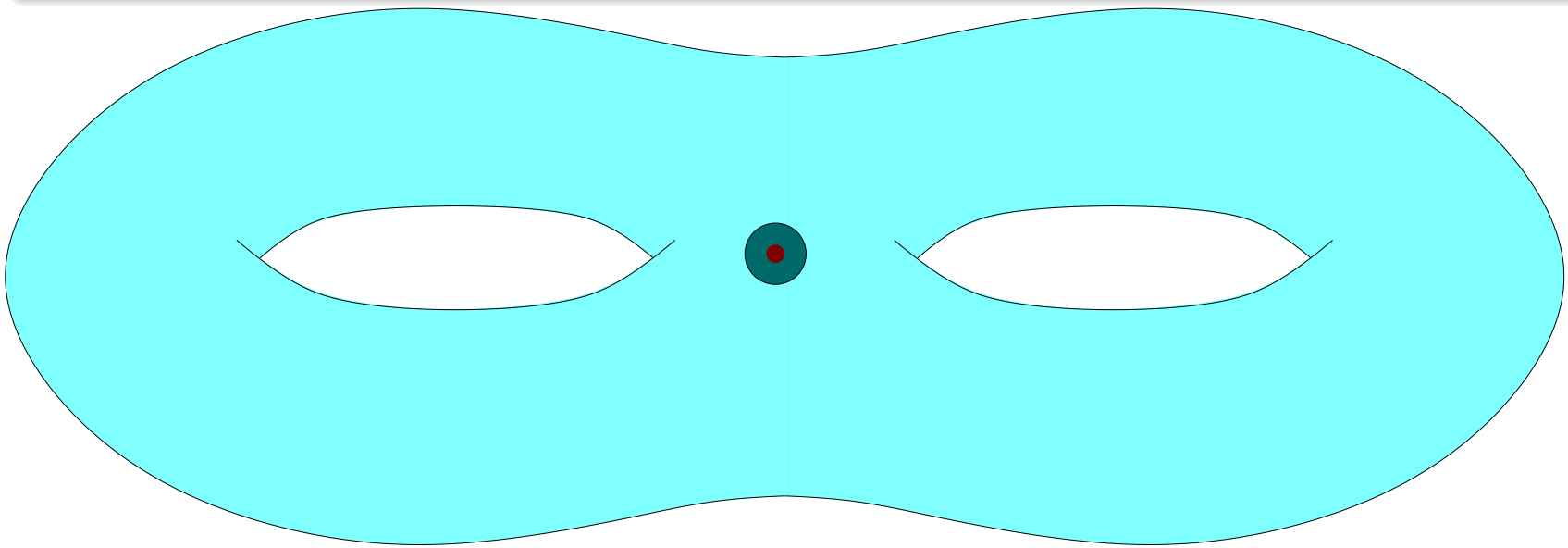
Comment calculer un plus court lacet non contractile ?
Pour l'instant, cherchons un plus court lacet non contractile à point-base b fixé.



Cut locus

Construction

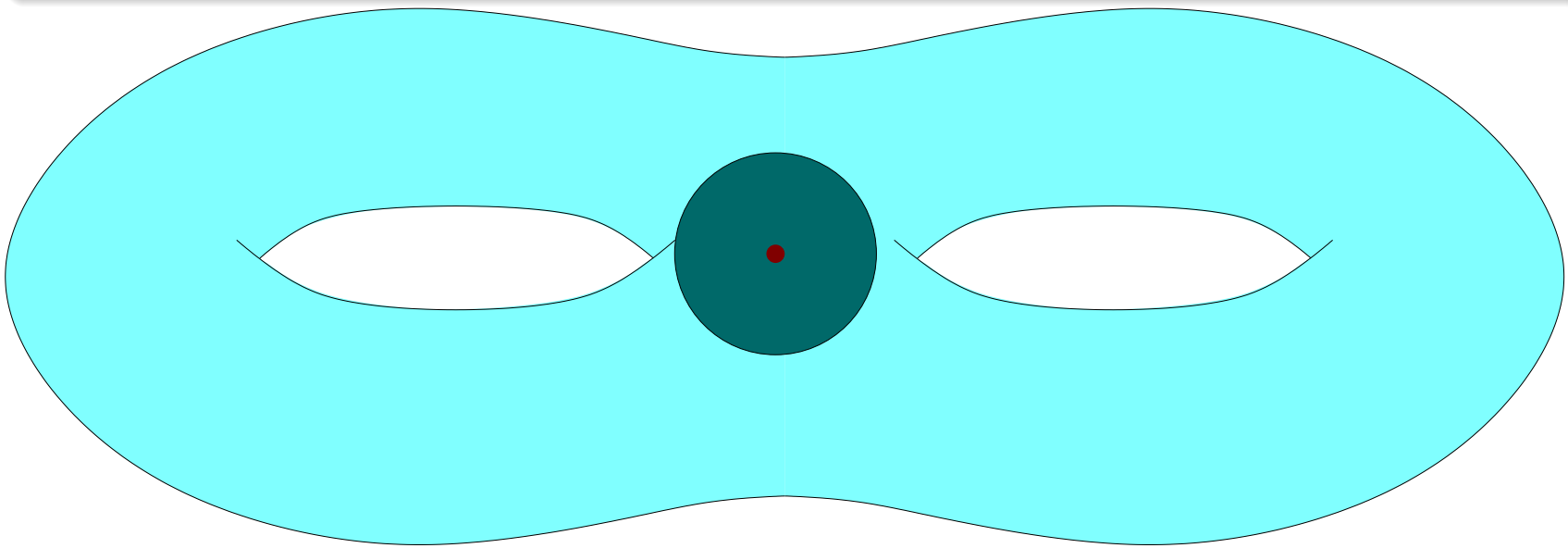
- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Cut locus

Construction

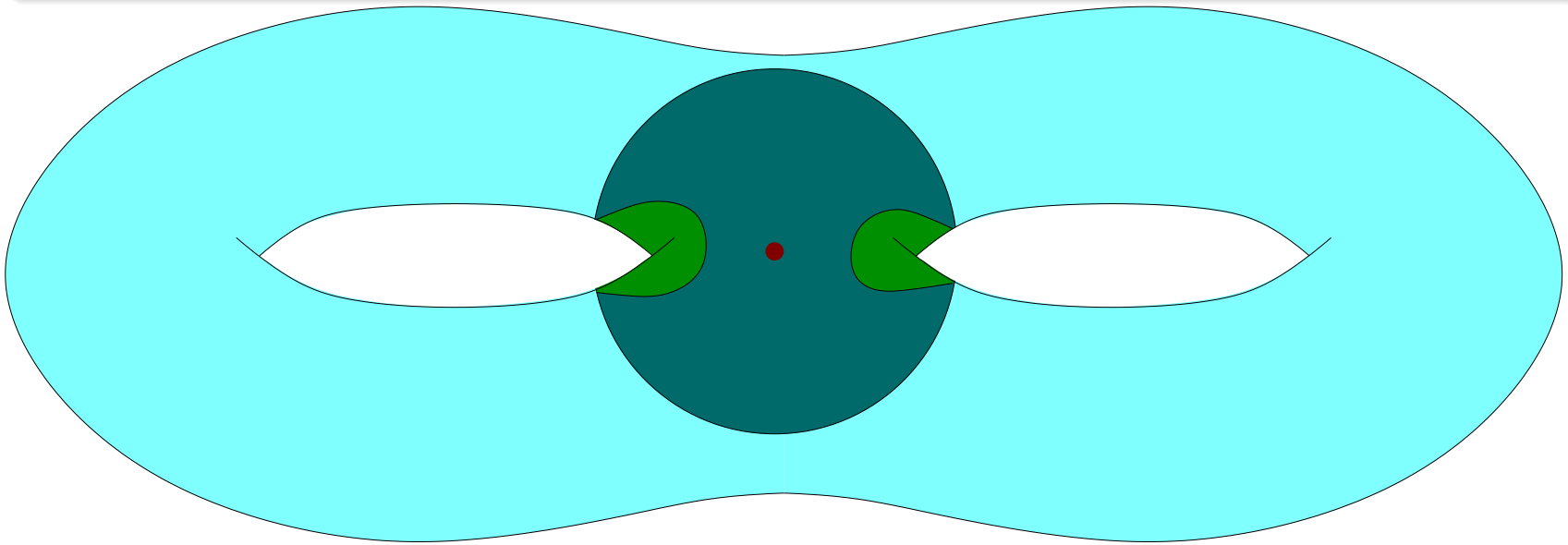
- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Cut locus

Construction

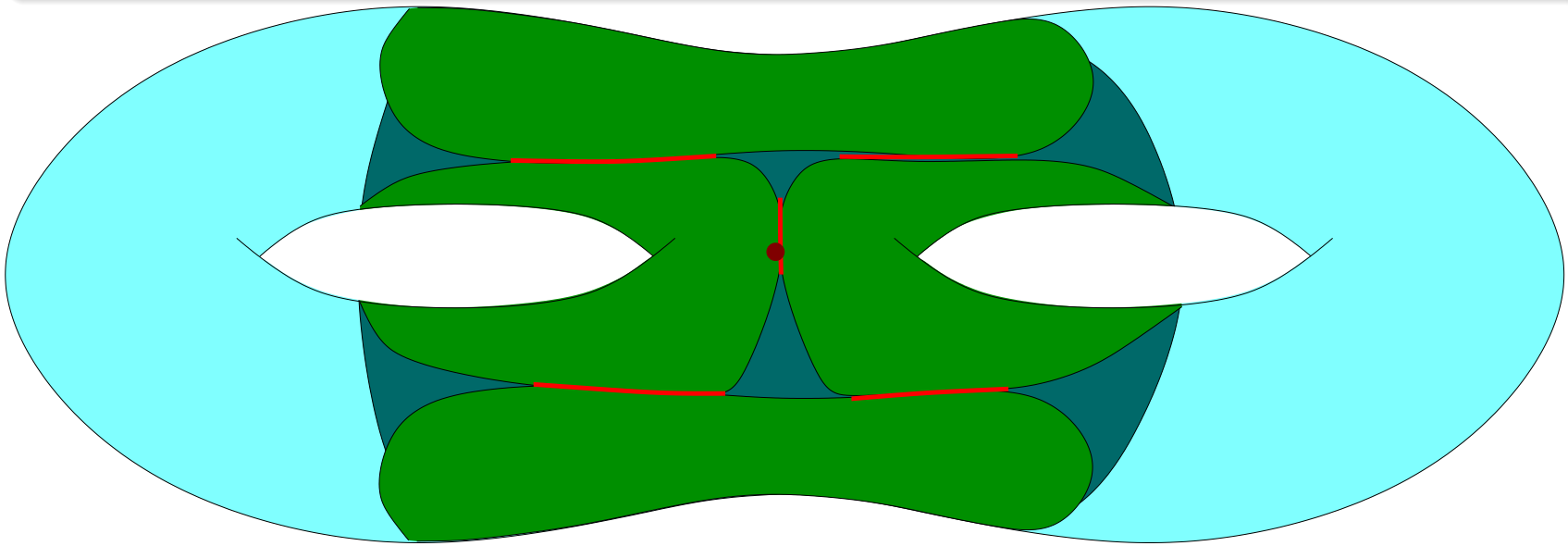
- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Cut locus

Construction

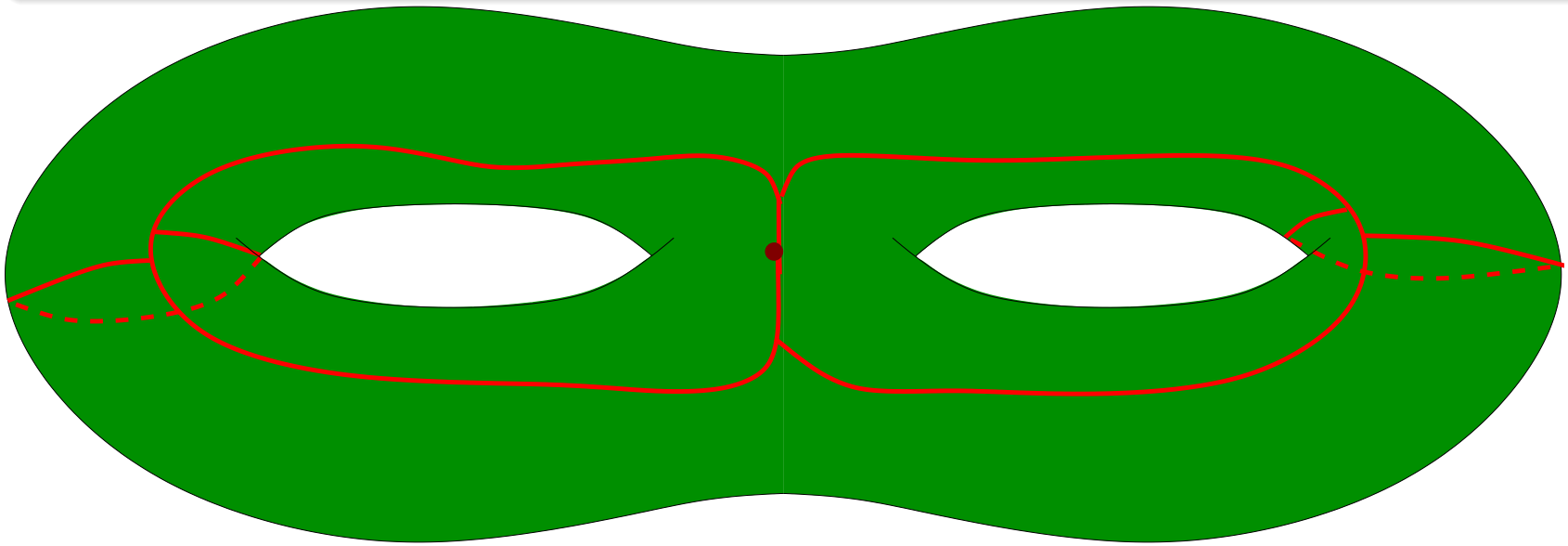
- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Cut locus

Construction

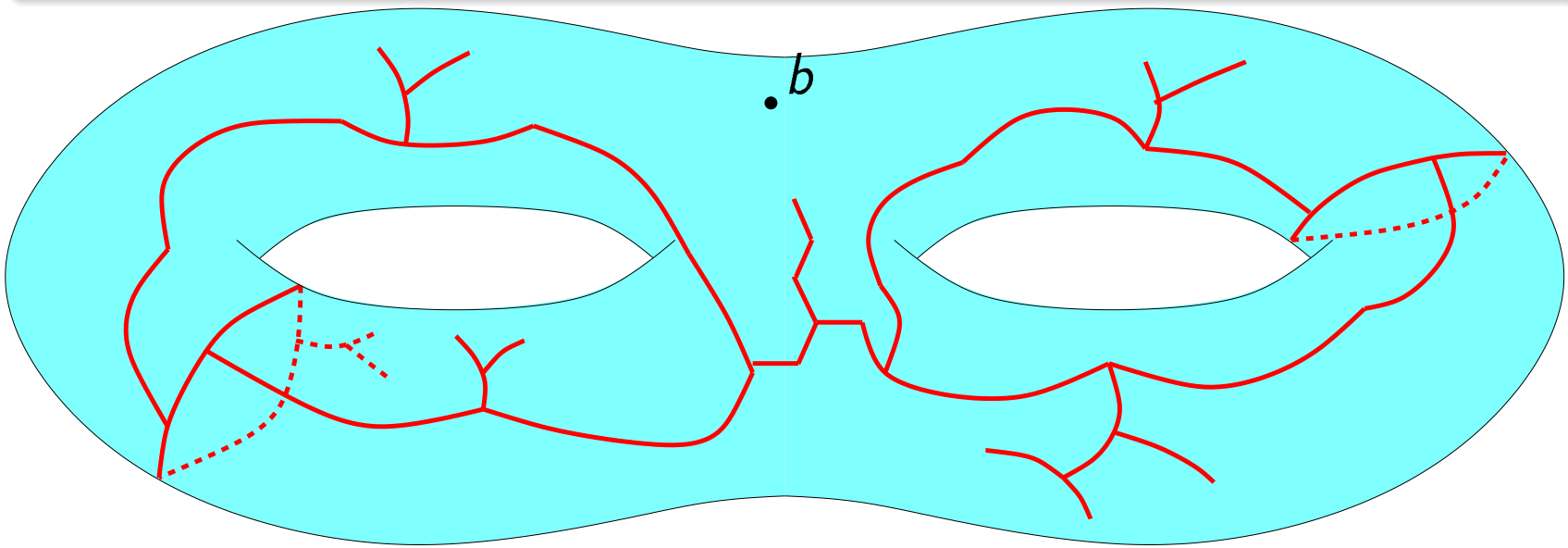
- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Cut locus

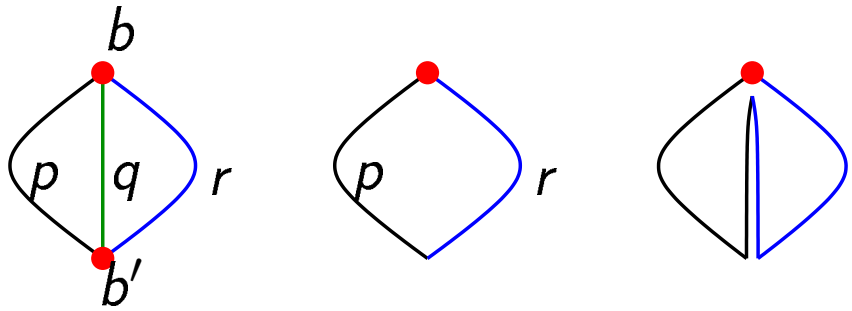
Construction

- Soit $b \in V$; soit T l'ensemble des arêtes d'un arbre de plus courts chemins dans G basé en b.
- Le cut locus de $\mathscr{S}$ par rapport à b est $C = E^* \setminus T^*$.
- C est un graphe de découpe : $\mathscr{S} \setminus C$ est (homéomorphe à) un disque ouvert.



Croisements du plus court lacet avec le cut locus

Si p, q et r sont des chemins de b à b', alors : $p \cdot \bar{q}$ et $q \cdot \bar{r}$ contractiles $\Rightarrow$ $p \cdot \bar{r}$ contractile.

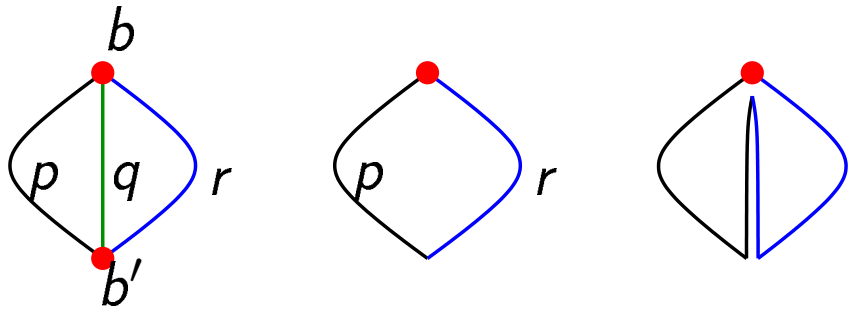


Soit ℓ un plus court lacet non contractile.

- ℓ croise le cut locus au moins une fois ;
- si ℓ croise le cut locus au moins deux fois, il y a un lacet non contractile qui est plus court ; contradiction !

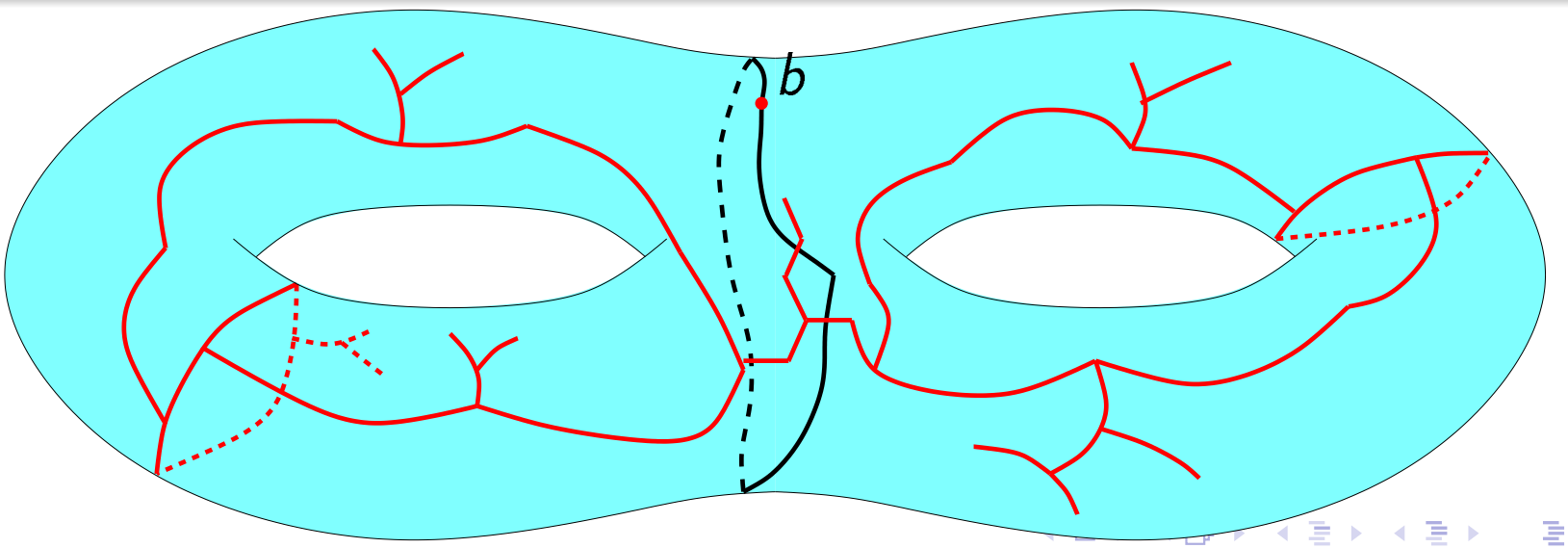
Croisements du plus court lacet avec le cut locus

Si p, q et r sont des chemins de b à b', alors : $p \cdot \bar{q}$ et $q \cdot \bar{r}$ contractiles $\Rightarrow$ $p \cdot \bar{r}$ contractile.



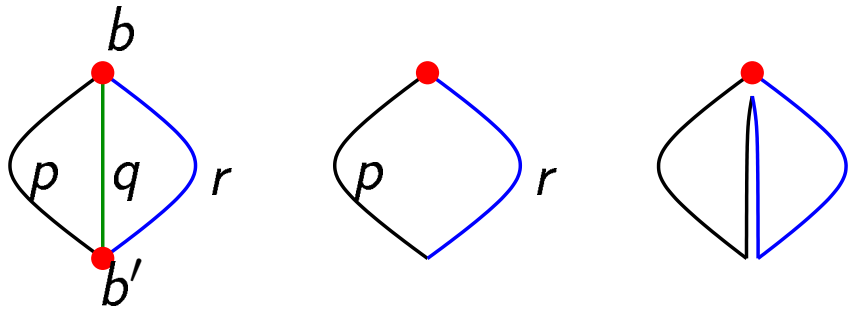
Soit ℓ un plus court lacet non contractile.

- ℓ croise le cut locus au moins une fois ;
- si ℓ croise le cut locus au moins deux fois, il y a un lacet non contractile qui est plus court ; contradiction !



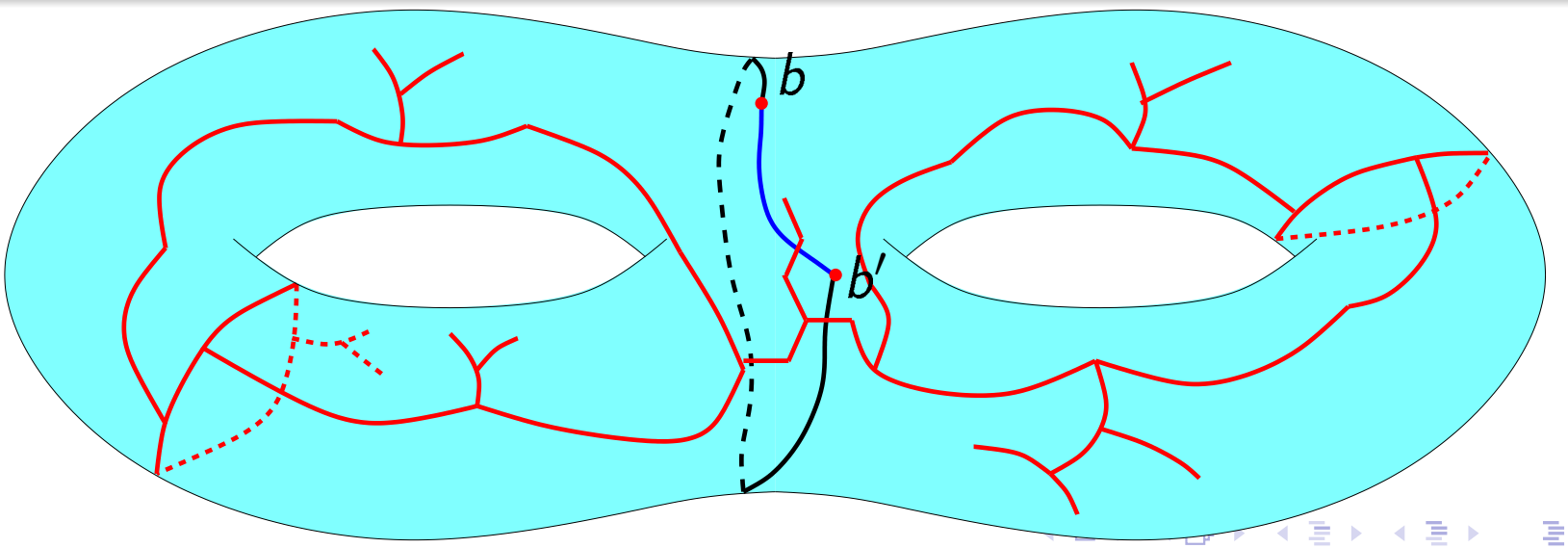
Croisements du plus court lacet avec le cut locus

Si p, q et r sont des chemins de b à b', alors : $p \cdot \bar{q}$ et $q \cdot \bar{r}$ contractiles $\Rightarrow$ $p \cdot \bar{r}$ contractile.



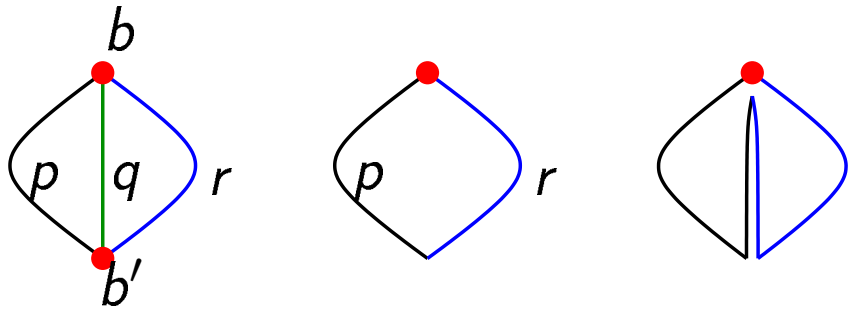
Soit ℓ un plus court lacet non contractile.

- ℓ croise le cut locus au moins une fois ;
- si ℓ croise le cut locus au moins deux fois, il y a un lacet non contractile qui est plus court ; contradiction !



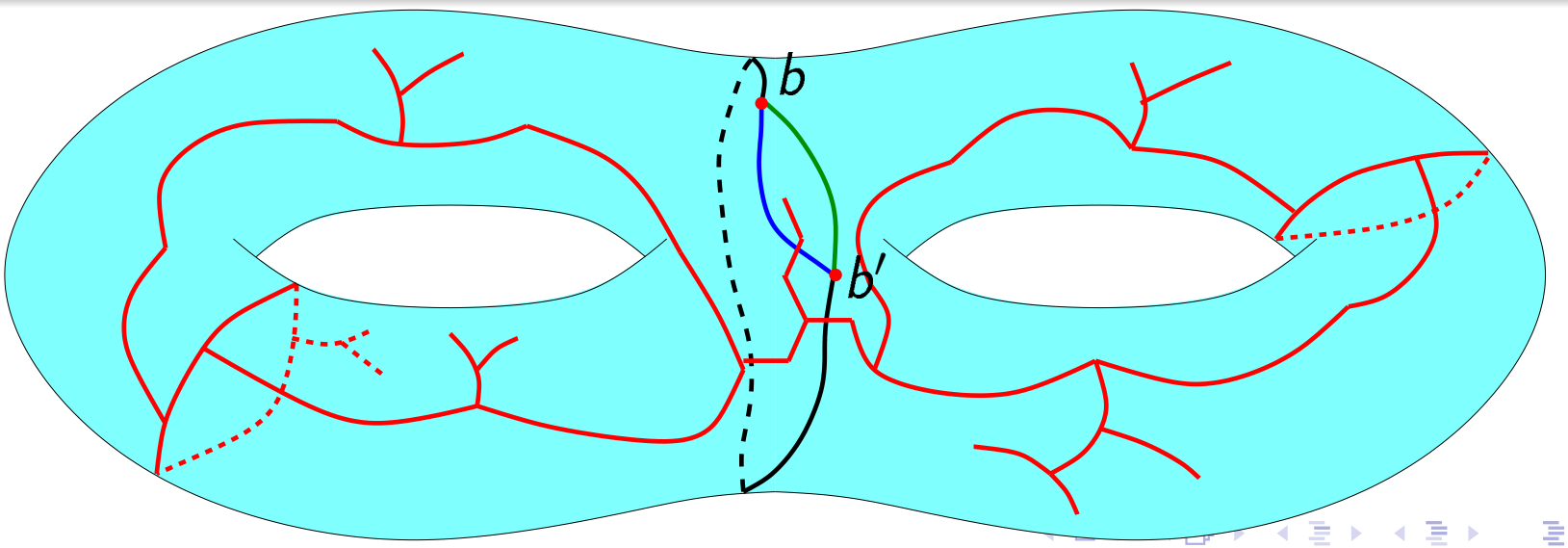
Croisements du plus court lacet avec le cut locus

Si p, q et r sont des chemins de b à b', alors : $p \cdot \bar{q}$ et $q \cdot \bar{r}$ contractiles $\Rightarrow$ $p \cdot \bar{r}$ contractile.

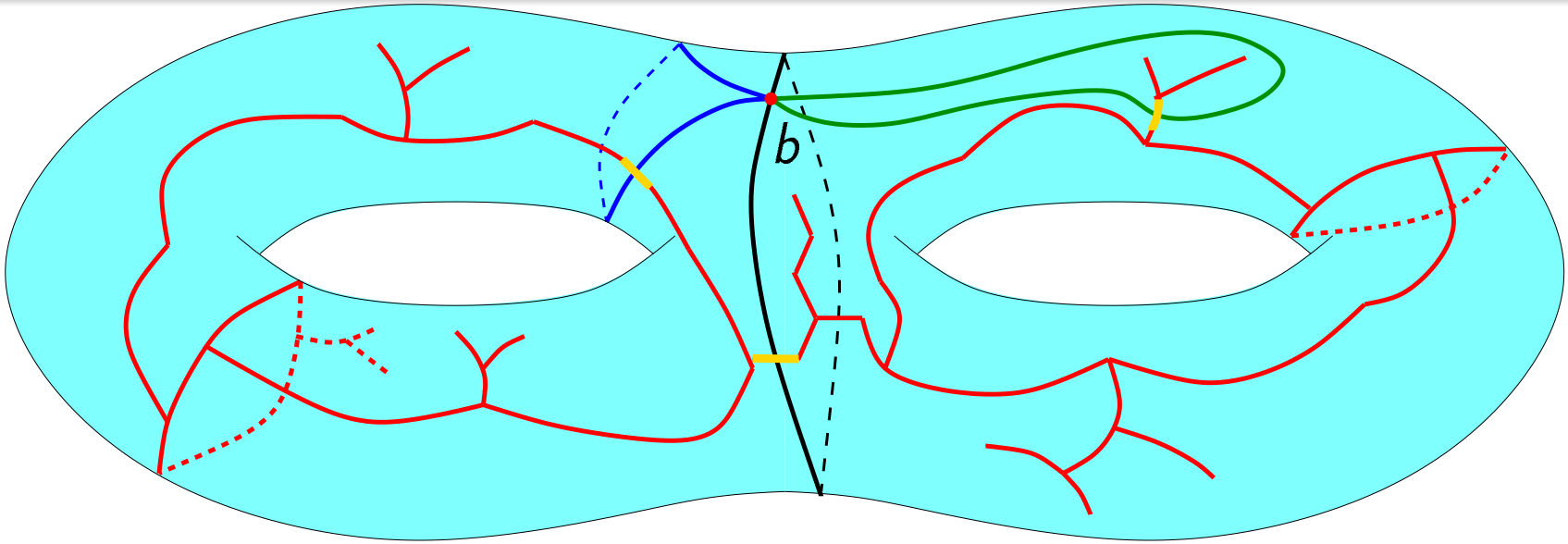


Soit ℓ un plus court lacet non contractile.

- ℓ croise le cut locus au moins une fois ;
- si ℓ croise le cut locus au moins deux fois, il y a un lacet non contractile qui est plus court ; contradiction !



Lacets croisant le cut locus exactement une fois



Définition

Si e est une arête de C, soit $e^{\perp}$ un lacet qui suit un plus court chemin partant de b, croise e et revient sur b par un plus court chemin.

Un plus court lacet non contractile est de la forme $e^{\perp}$.

Lemme

$e^{\perp}$ est contractile ssi e sépare C en deux morceaux, l'un d'entre eux étant un arbre.

Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^{\perp}$

$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^{\perp}$, avec $e \in C$ non supprimé et $e^{\perp}$ de longueur minimale

$\rightarrow$ complexité $O(n)$.

Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^{\perp}$

$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^{\perp}$, avec $e \in C$ non supprimé et $e^{\perp}$ de longueur minimale

$\rightarrow$ complexité $O(n)$.

Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^\perp$

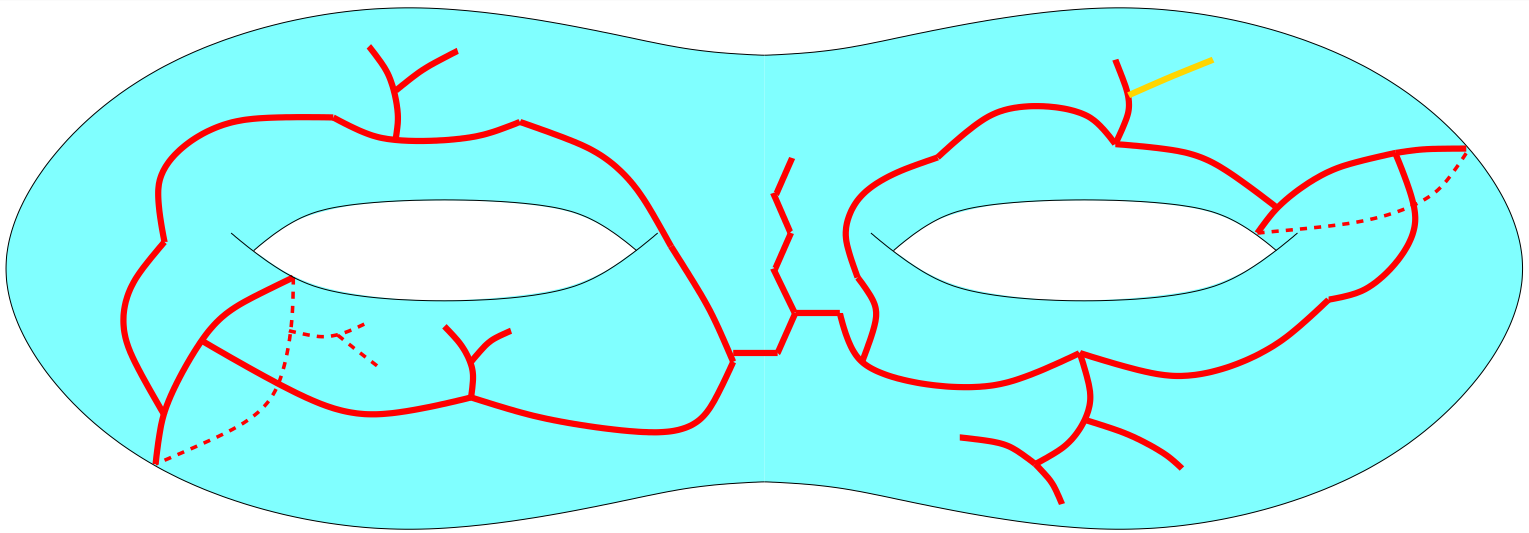
$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^\perp$, avec $e \in C$ non supprimé et $e^\perp$ de longueur minimale

$\rightarrow$ complexité $O(n)$.



Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^\perp$

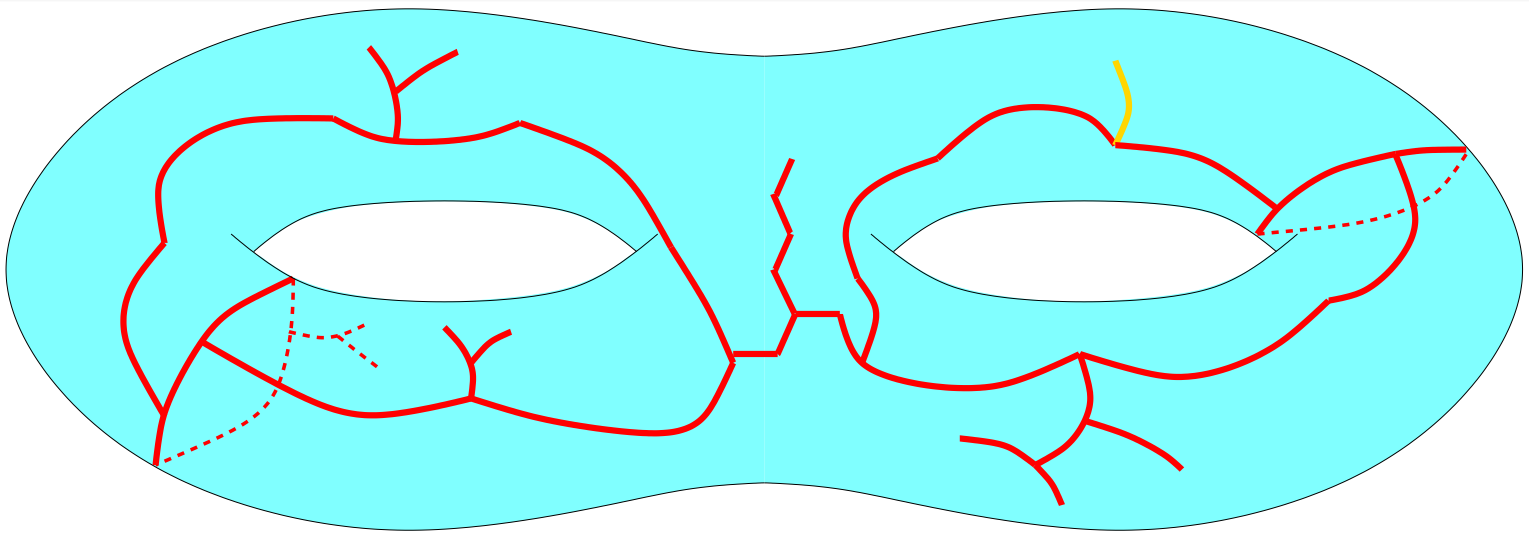
$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^\perp$, avec $e \in C$ non supprimé et $e^\perp$ de longueur minimale

$\rightarrow$ complexité $O(n)$.



Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^\perp$

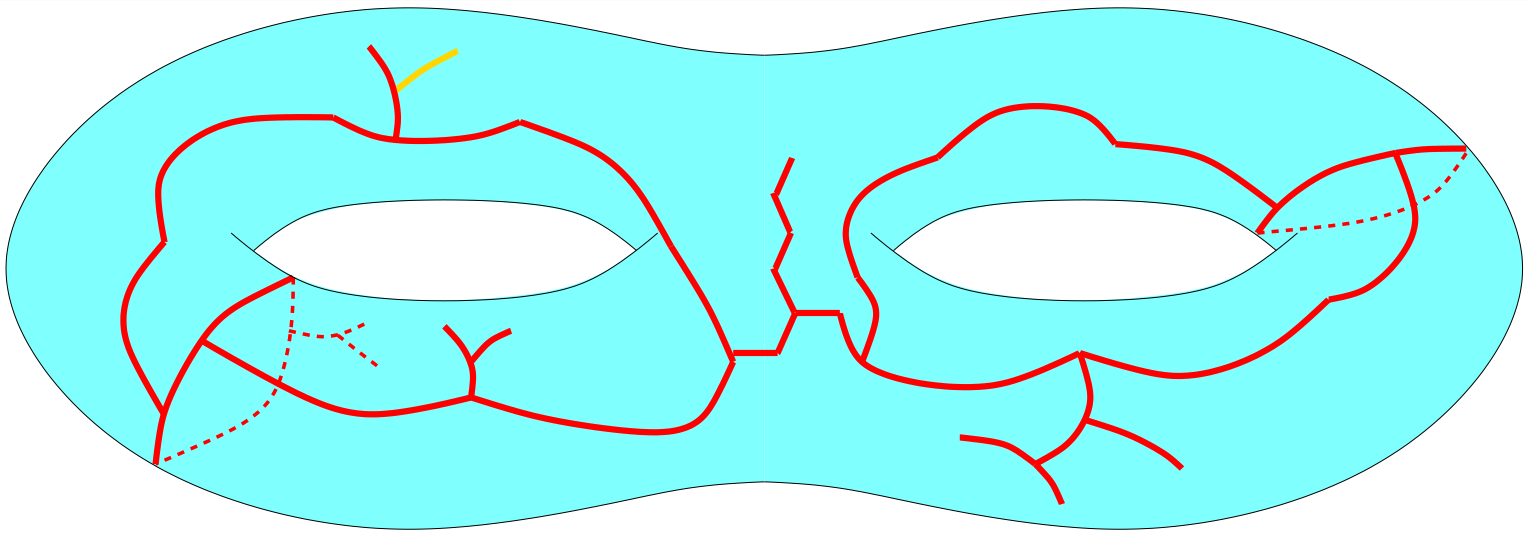
$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^\perp$, avec $e \in C$ non supprimé et $e^\perp$ de longueur minimale

$\rightarrow$ complexité $O(n)$.



Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^\perp$

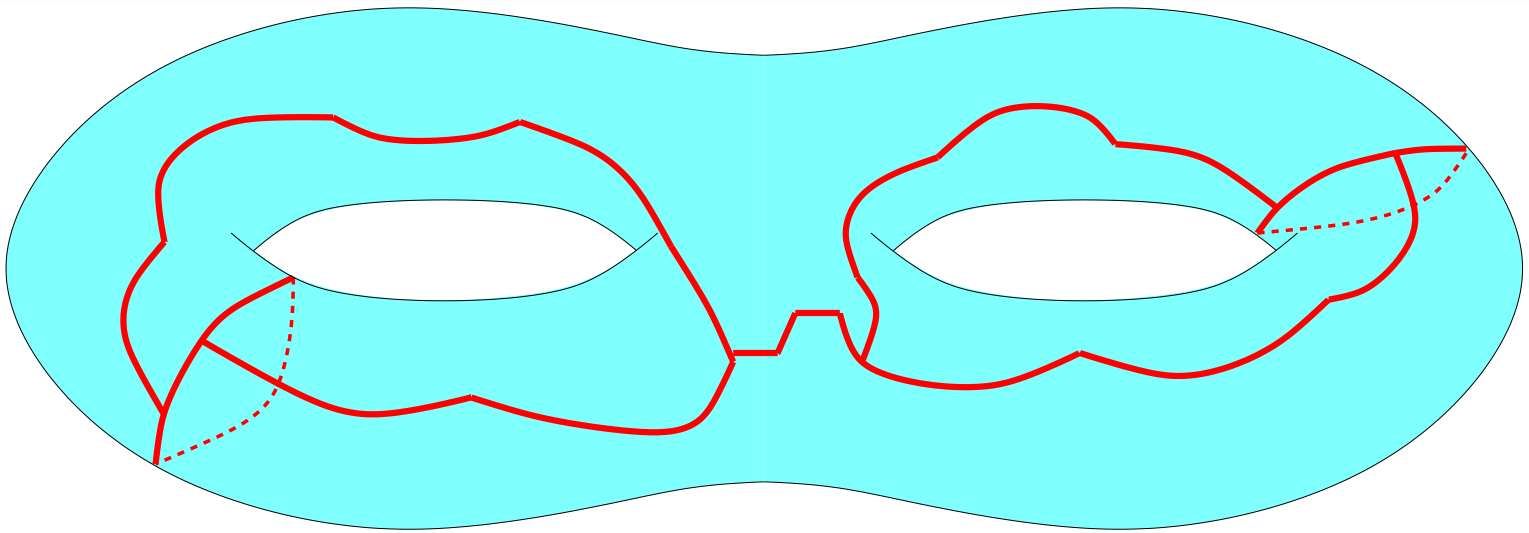
$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^\perp$, avec $e \in C$ non supprimé et $e^\perp$ de longueur minimale

$\rightarrow$ complexité $O(n)$.



Algorithme pour calculer un plus court lacet non contractile

Soit n la complexité de G.

- Pour chaque arête e de C, calculer la longueur de $e^\perp$

$\rightarrow$ arbre de plus courts chemins dans G : Dijkstra, $O(n \log n)$;

- supprimer de C les arêtes de C qui déconnectent C en deux morceaux, l'un d'entre eux étant un arbre

$\rightarrow$ complexité $O(n)$ (élagage) ;

- retourner $e^\perp$, avec $e \in C$ non supprimé et $e^\perp$ de longueur minimale

$\rightarrow$ complexité $O(n)$.

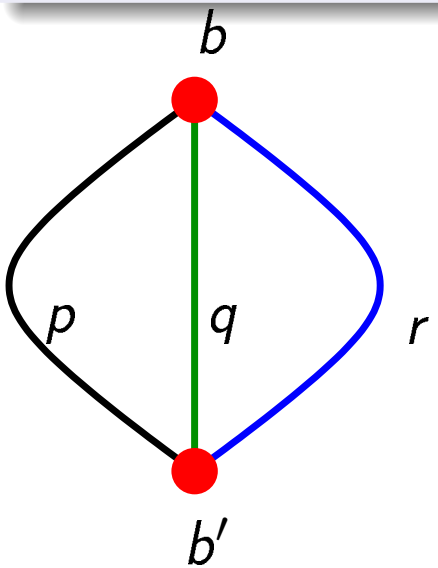
Résultat

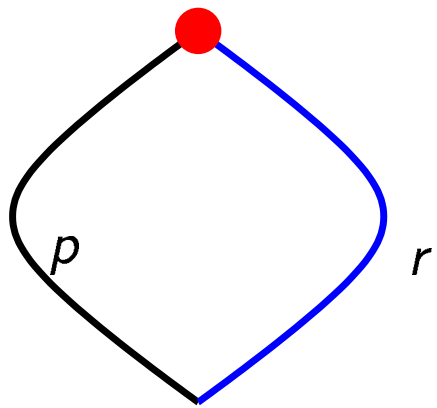
- Plus court lacet non contractile avec point-base b : complexité $O(n \log n)$ [Erickson et Har-Peled, 2004].
- Sans fixer le point-base : $O(n^2 \log n)$ en appliquant l'algorithme précédent $O(n)$ fois, plaçant b sur chaque sommet de G.
- Finalement : circuit dans G.

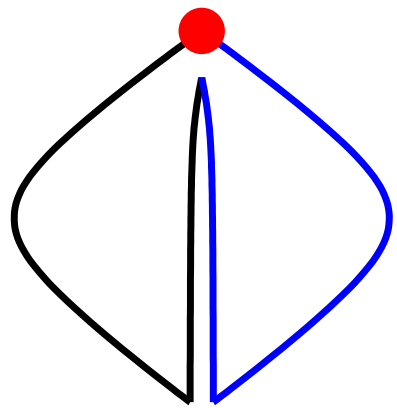
Plus court lacet non séparateur

- Un lacet simple est contractile ssi il borde un disque.
- Un lacet simple est homologiquement trivial ssi il est séparateur.

Si p, q et r sont des chemins de b à b', alors : $p \cdot \bar{q}$ et $q \cdot \bar{r}$ homologiquement trivial $\Rightarrow$ $p \cdot \bar{r}$ homologiquement trivial.

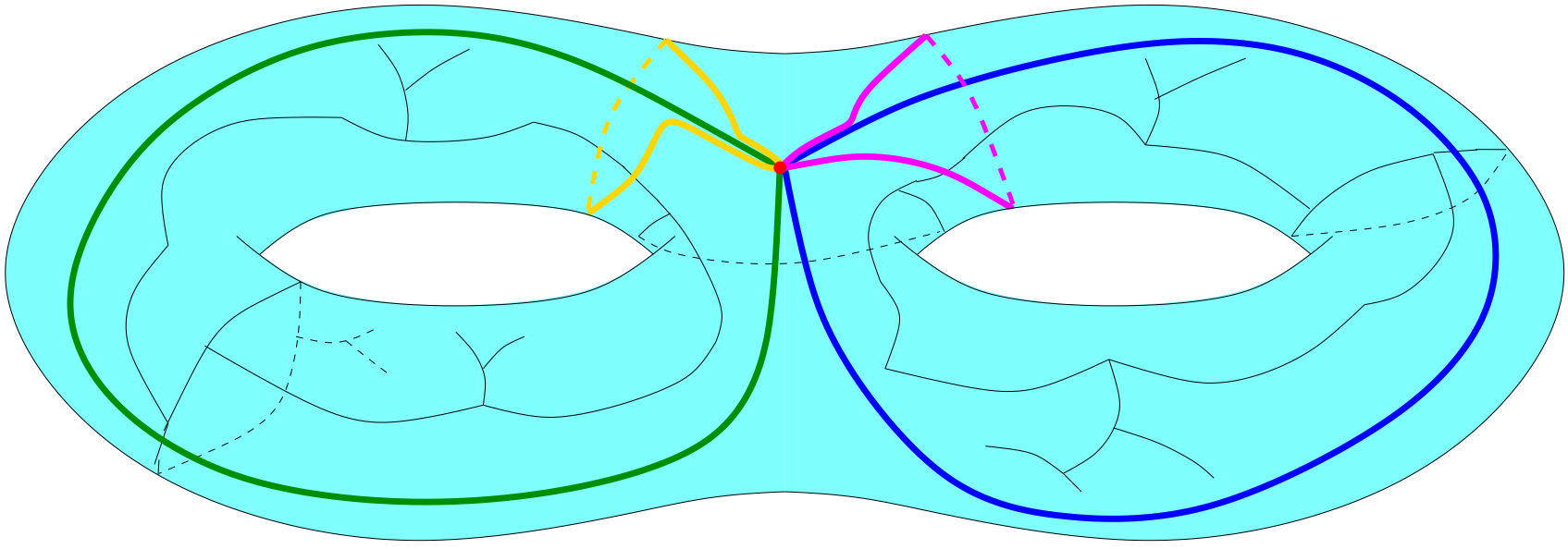






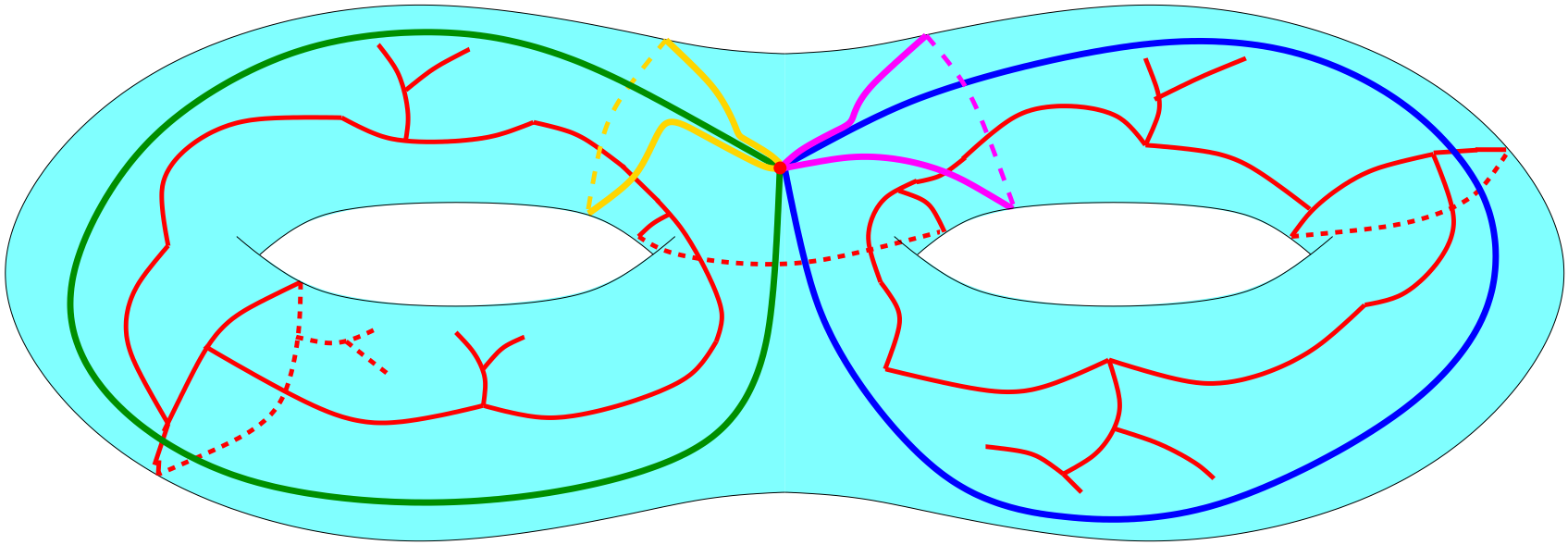
2. Graphe de découpe court

Aperçu de l'algorithme



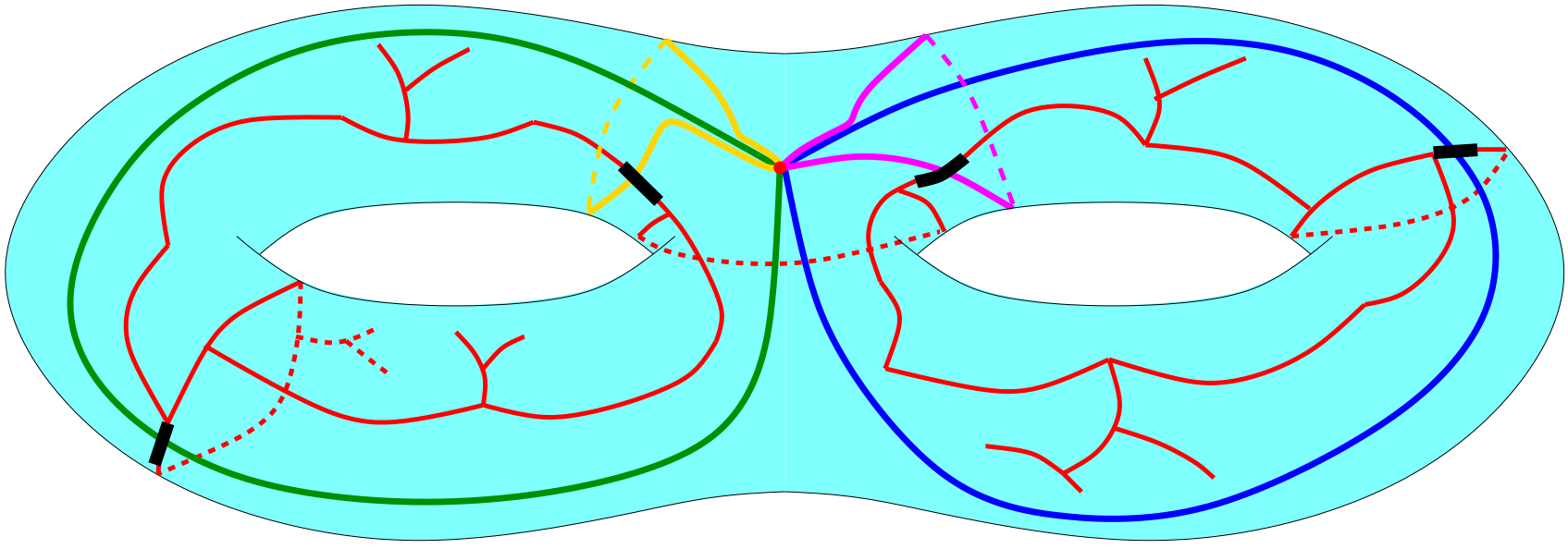
- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\{e^\perp \mid e \notin U\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

Aperçu de l'algorithme



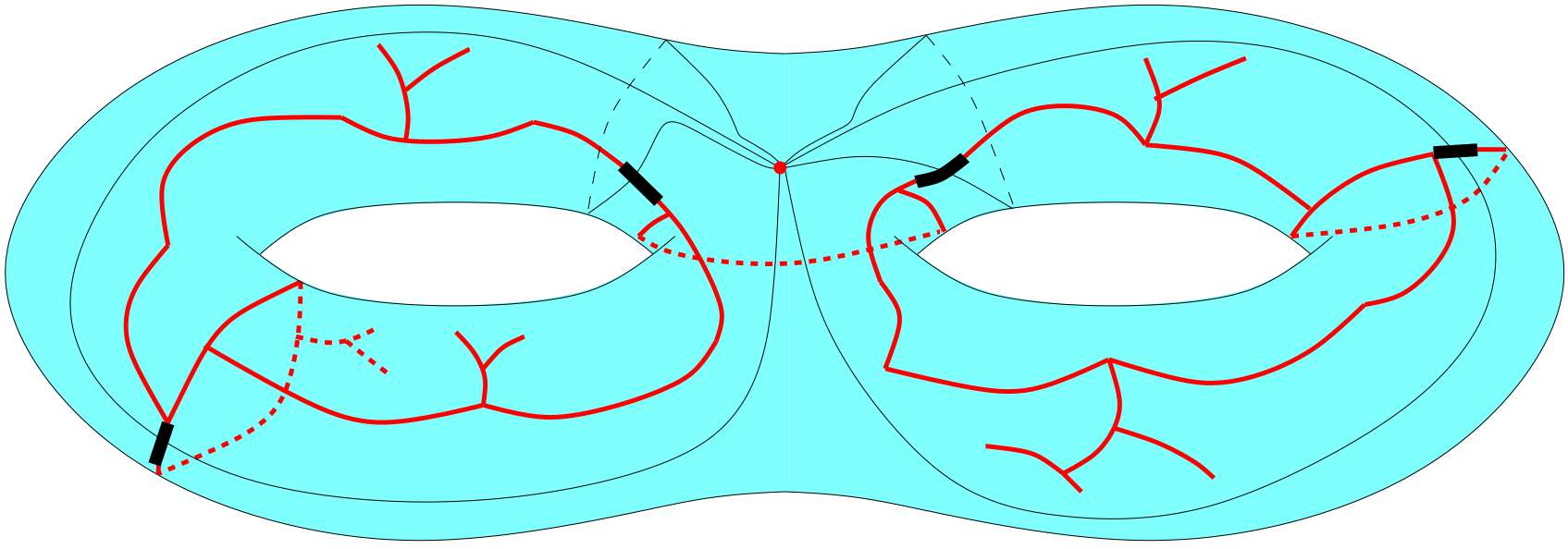
- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\{e^\perp \mid e \notin U\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

Aperçu de l'algorithme



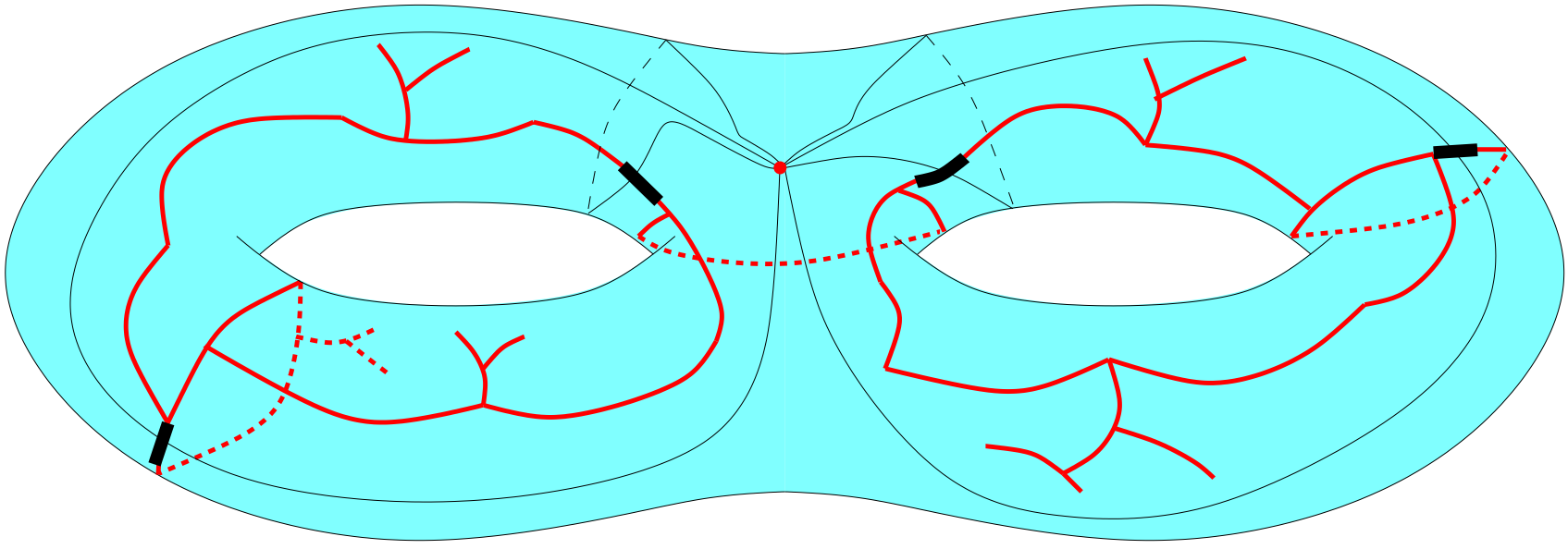
- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\{e^\perp \mid e \notin U\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

Aperçu de l'algorithme



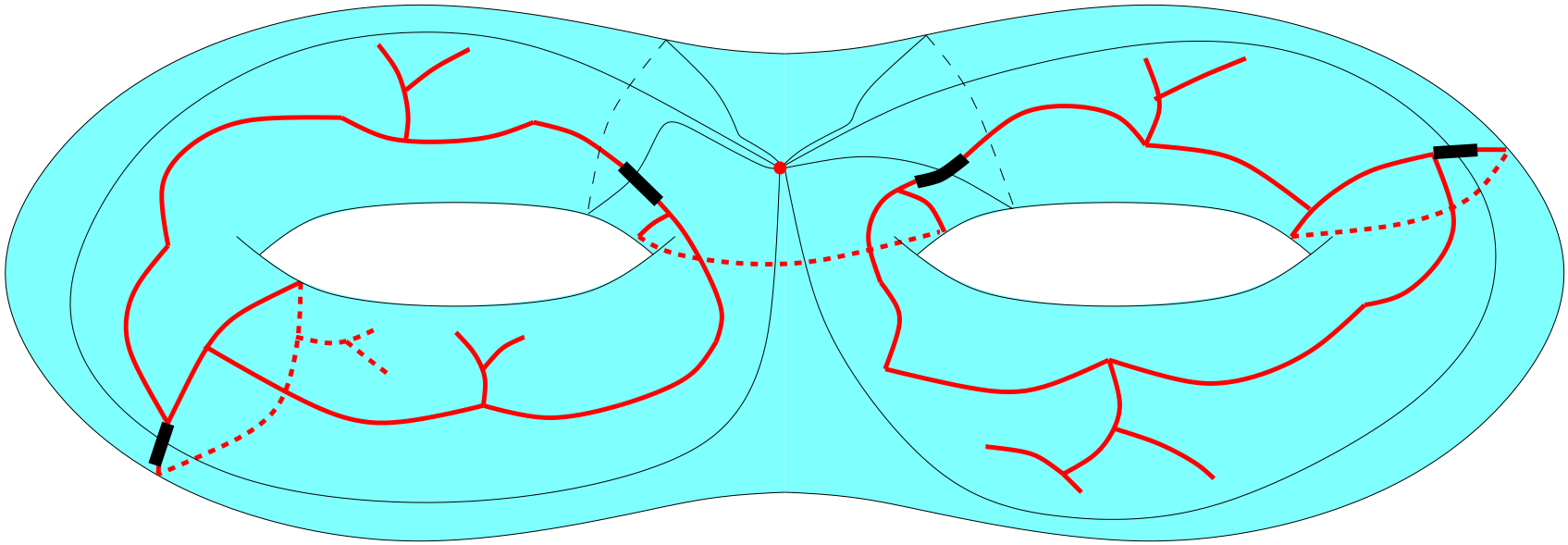
- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\{e^\perp \mid e \notin U\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

Aperçu de l'algorithme



- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\{e^\perp \mid e \notin U\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

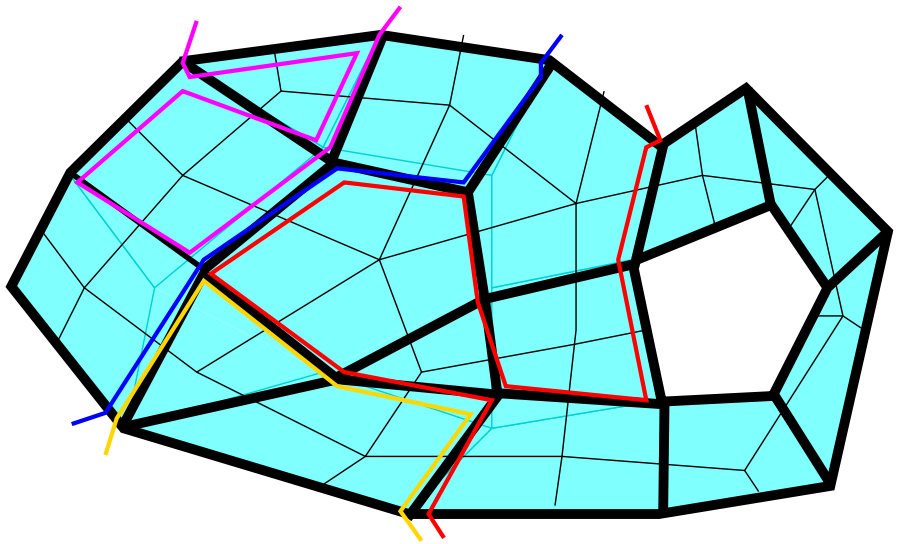
Aperçu de l'algorithme



- $A^\perp$ est un graphe de découpe ssi $C \setminus A$ est un arbre couvrant de C ;
- Soit U un arbre couvrant *max*imal de C, où le poids d'une arête e est la longueur de $e^\perp$. Retourner $\left\{e^\perp \;\middle|\; e \notin U\right\}$.
- Temps $O(n \log n + gn)$ car il y a $2g$ lacets.

Aparté : il est optimal !

Les lacets $A^{\perp}$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

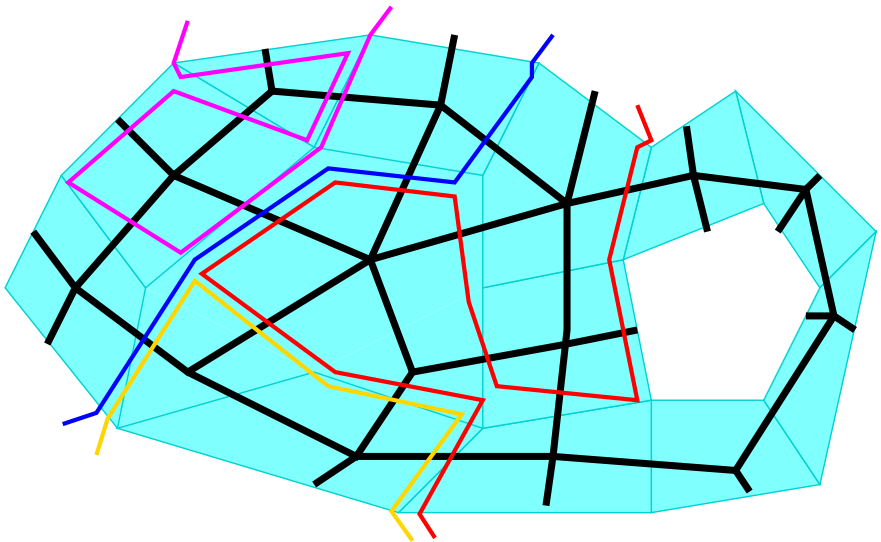
NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

Aparté : il est optimal !

Les lacets $A^\perp$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

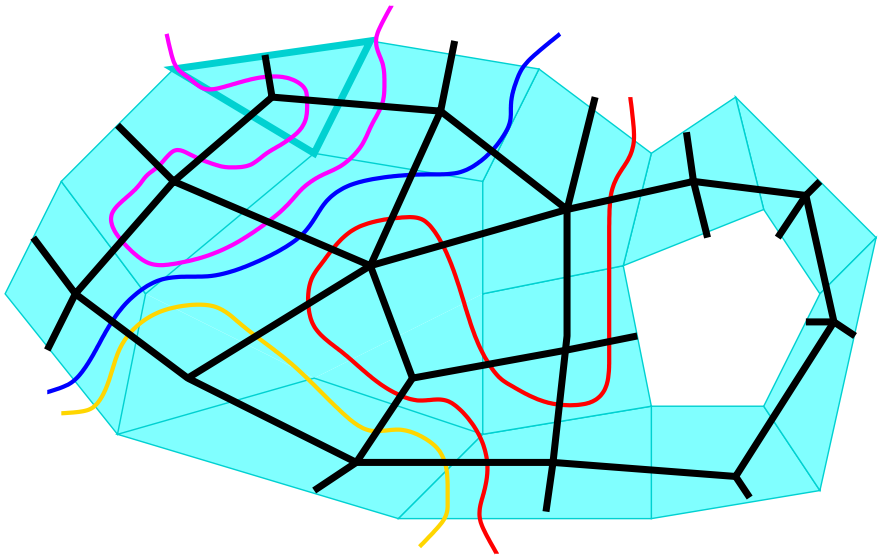
NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

Aparté : il est optimal !

Les lacets $A^{\perp}$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

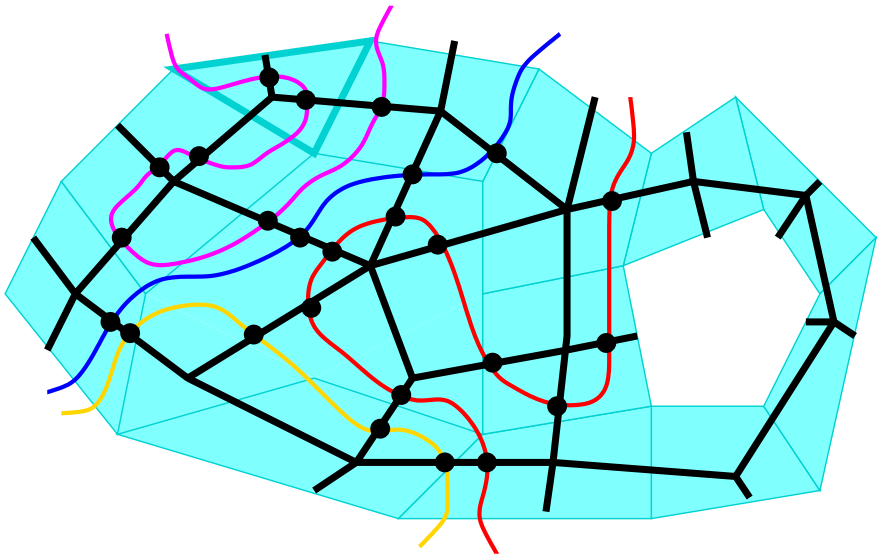
NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

Aparté : il est optimal !

Les lacets $A^{\perp}$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

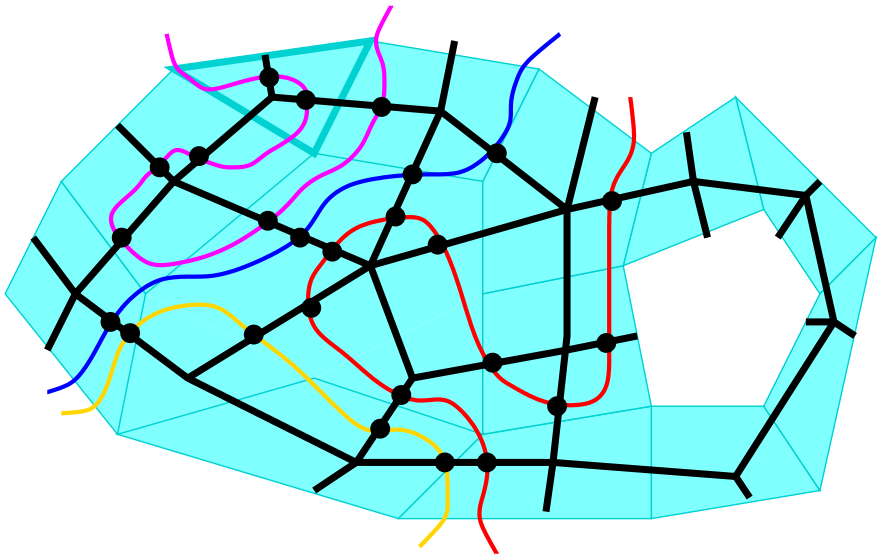
NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

Aparté : il est optimal !

Les lacets $A^{\perp}$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

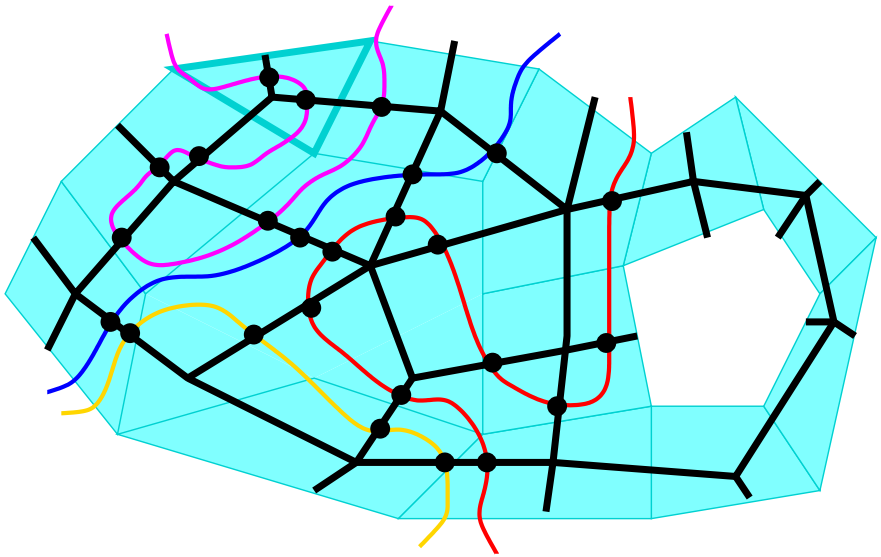
NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

Aparté : il est optimal !

Les lacets $A^{\perp}$ ne sont pas nécessairement disjoints, ni simples.
Cadre topologique dual : surface à métrique des croisements.



Trouver le plus court graphe de découpe

NP-dur ! [Erickson et Har-Peled, 2004]

Théorème ([CdV, 2010], étendant [Erickson et Whittlesey, 2006])

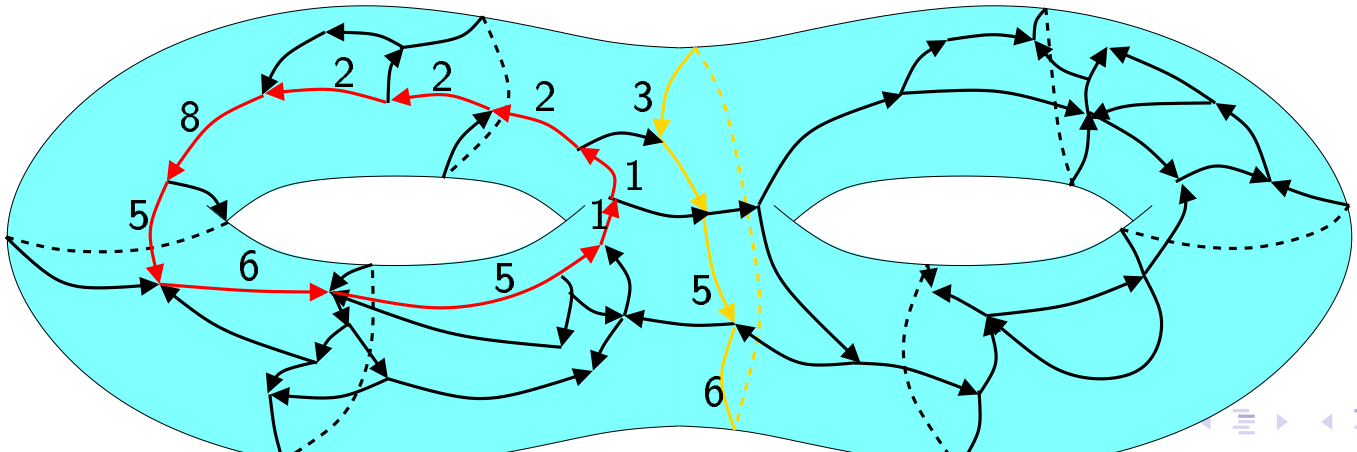
Soit P un ensemble de points sur une surface à métrique des croisements $\mathscr{S}$. On peut calculer le plus court graphe de découpe dont l'ensemble des sommets est exactement P en temps $O(n \log n + gn + |P|)$.

3. Plus courtes courbes non triviales, extensions

Aperçu : plus courtes courbes non triviales

g : genre, k : nombre d'arêtes de la plus courte courbe non triviale.

	non orienté	orienté
pondéré	**$O(n^2 \log n)$** [Erickson–Har-Peled'04] $O(g^{3/2} n^{3/2} \log n)$ non-sép $g^{O(g)} n^{3/2}$ non-contr } [Cabello–Mohar'07] $g^{O(g)} n \log n$ [Kutz'06] $O(g^3 n \log n)$ [Cabello–Chambers'07] **$O(g^2 n \log n)$** [Cabello–Chambers–Erickson'12] **$g^{O(g)} n \log\log n$** [Italiano et coll.'11]	$O(n^2 \log n)$ [Cab–CdV–Laz'10] **$O(g^{1/2} n^{3/2} \log n)$** [Cab–CdV–Laz'10] $2^{O(g)} n \log n$ non-sép [Erickson–Nayyeri'11] **$O(g^2 n \log n)$** non-sép $g^{O(g)} n \log n$ non-contr } [Erickson'11] **$O(g^3 n \log n)$** non-contr [Fox'13]
non pondéré	$O(n^3)$ [Thomassen'90] **$O(n^2)$** [Cab–CdV–Laz'10] **$O(gnk)$** [Cab–CdV–Laz'10] $O(gn)$ pour 2-approx [Erickson–Har-Peled'04] **$O(gn/\varepsilon)$** pour $(1+\varepsilon)$-approx [Cab–CdV–Laz'10]	**$O(n^2)$** [Cab–CdV–Laz'10] **$O(gnk)$** [Cab–CdV–Laz'∞]



Résultat

Edge-width (ew) : Longueur d'un plus court chemin fermé non trivial.

Théorème [Cabello, CdV, Lazarus, 2010]

Soit G non pondéré plongé sur $\mathscr{S}$. Soit $k \geq 1$. On peut en temps $O(gnk)$:

- calculer $\mathrm{ew}(G)$, si $\mathrm{ew}(G) \leq k$;
- ou détecter que $\mathrm{ew}(G) > k$.

Corollaire

On peut calculer $\mathrm{ew}(G)$ en temps $O(gn\,\mathrm{ew}(G))$.

Démonstration

Appliquer le théorème avec $k = 1, 2, 4, 8, \ldots, 2^p, \ldots$ jusqu'à ce que l'algorithme renvoie $\mathrm{ew}(G)$. □

Résultat

Edge-width (ew) : Longueur d'un plus court chemin fermé non trivial.

Théorème [Cabello, CdV, Lazarus, 2010]

Soit G non pondéré plongé sur $\mathscr{S}$. Soit $k \geq 1$. On peut en temps $O(gnk)$:

- calculer $\mathrm{ew}(G)$, si $\mathrm{ew}(G) \leq k$;
- ou détecter que $\mathrm{ew}(G) > k$.

Corollaire

On peut calculer $\mathrm{ew}(G)$ en temps $O(gn\,\mathrm{ew}(G))$.

Démonstration

Appliquer le théorème avec $k = 1, 2, 4, 8, \ldots, 2^p, \ldots$ jusqu'à ce que l'algorithme renvoie $\mathrm{ew}(G)$. □

Résultat

Edge-width (ew) : Longueur d'un plus court chemin fermé non trivial.

Théorème [Cabello, CdV, Lazarus, 2010]

Soit G non pondéré plongé sur $\mathscr{S}$. Soit $k \geq 1$. On peut en temps $O(gnk)$:

- calculer $\mathrm{ew}(G)$, si $\mathrm{ew}(G) \leq k$;
- ou détecter que $\mathrm{ew}(G) > k$.

Corollaire

On peut calculer $\mathrm{ew}(G)$ en temps $O(gn\,\mathrm{ew}(G))$.

Démonstration

Appliquer le théorème avec $k = 1, 2, 4, 8, \ldots, 2^p, \ldots$ jusqu'à ce que l'algorithme renvoie $\mathrm{ew}(G)$. $\square$

Motivations : algorithmes linéaires

Vérifier que $\mathrm{ew}(G)$ est plus petit qu'une certaine valeur (et si oui, la calculer) sert de routine de base dans plusieurs algorithmes linéaires pour :

- si p est fixé, déterminer si un graphe est dessinable dans le plan avec au plus p croisements [Kawarabayashi, Reed, 2007] ;
- tester l'isomorphisme entre deux graphes de genre $\leq g$ fixé [Kawarabayashi, Mohar, 2009] ;
- calculer certains cycles induits passant par p sommets donnés de G, si G est plongé sur $\mathscr{S}$ fixée [Kobayashi, Kawarabayashi, 2009].

Ces algorithmes utilisaient une 2-approximation du calcul de $\mathrm{ew}(G)$.

Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

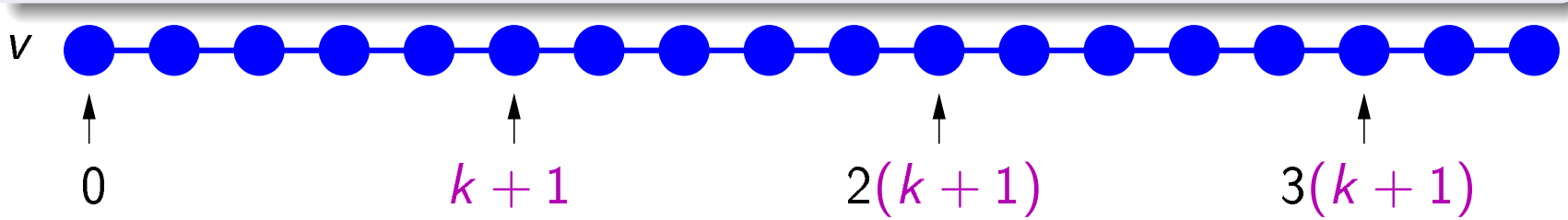
- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme*. Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

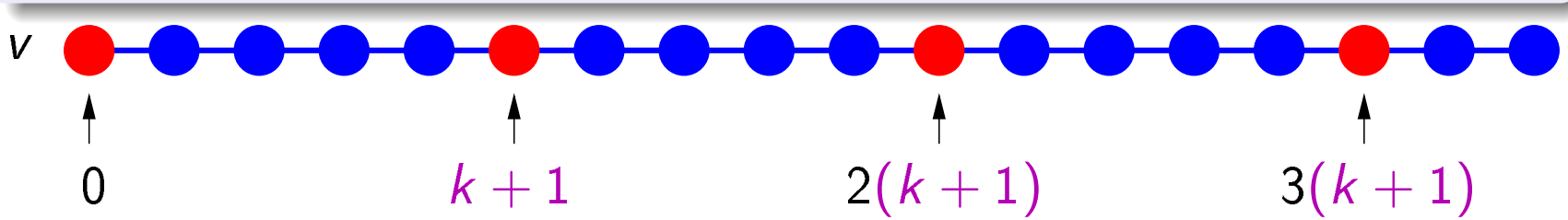


Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

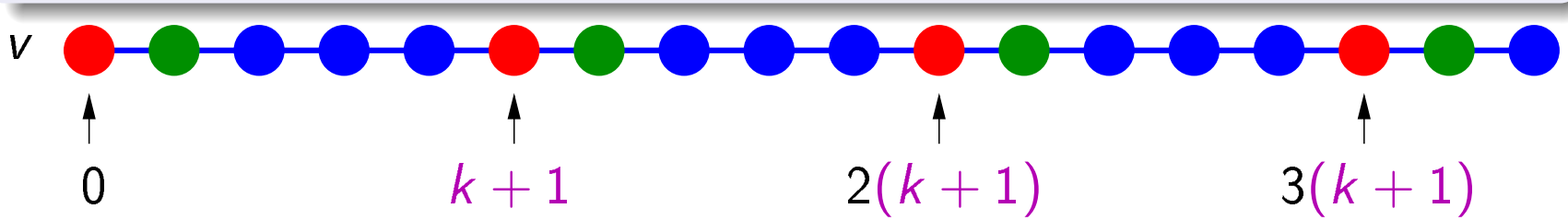


Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

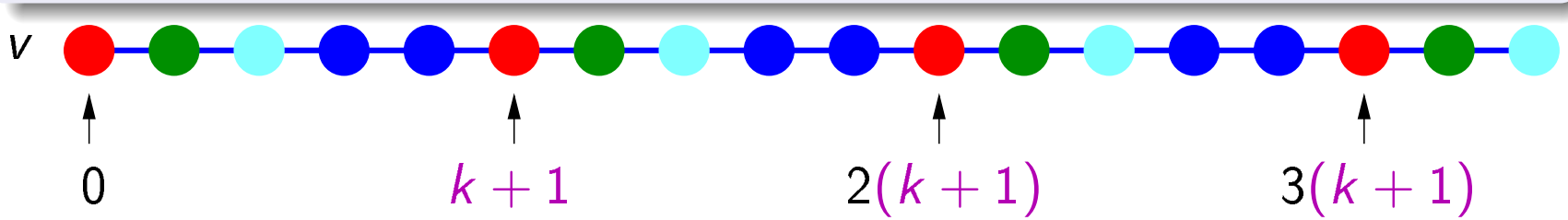


Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

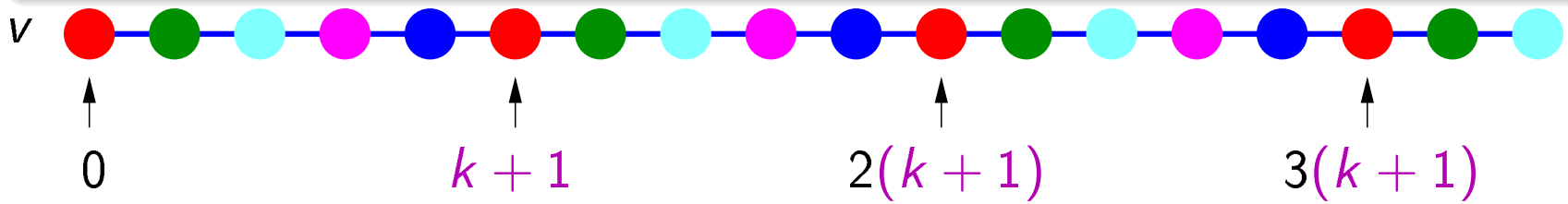


Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

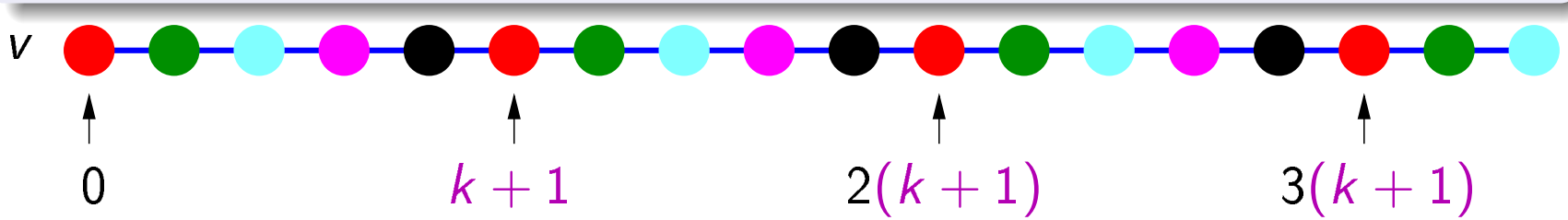


Aperçu de l'algorithme : Paralléliser !

- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme.* Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.

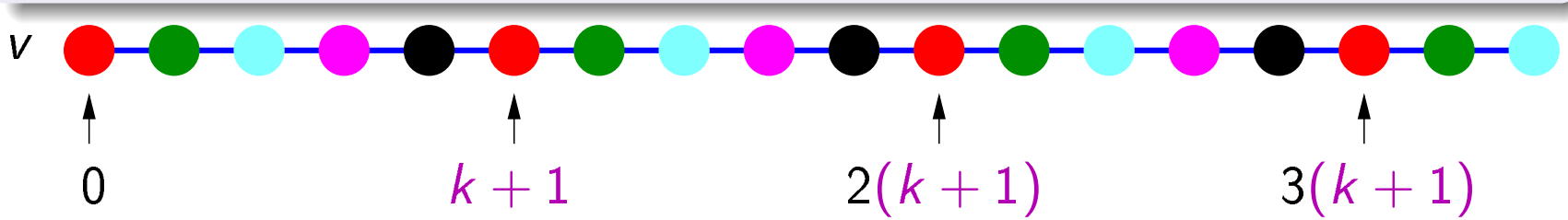


Aperçu de l'algorithme : Paralléliser !

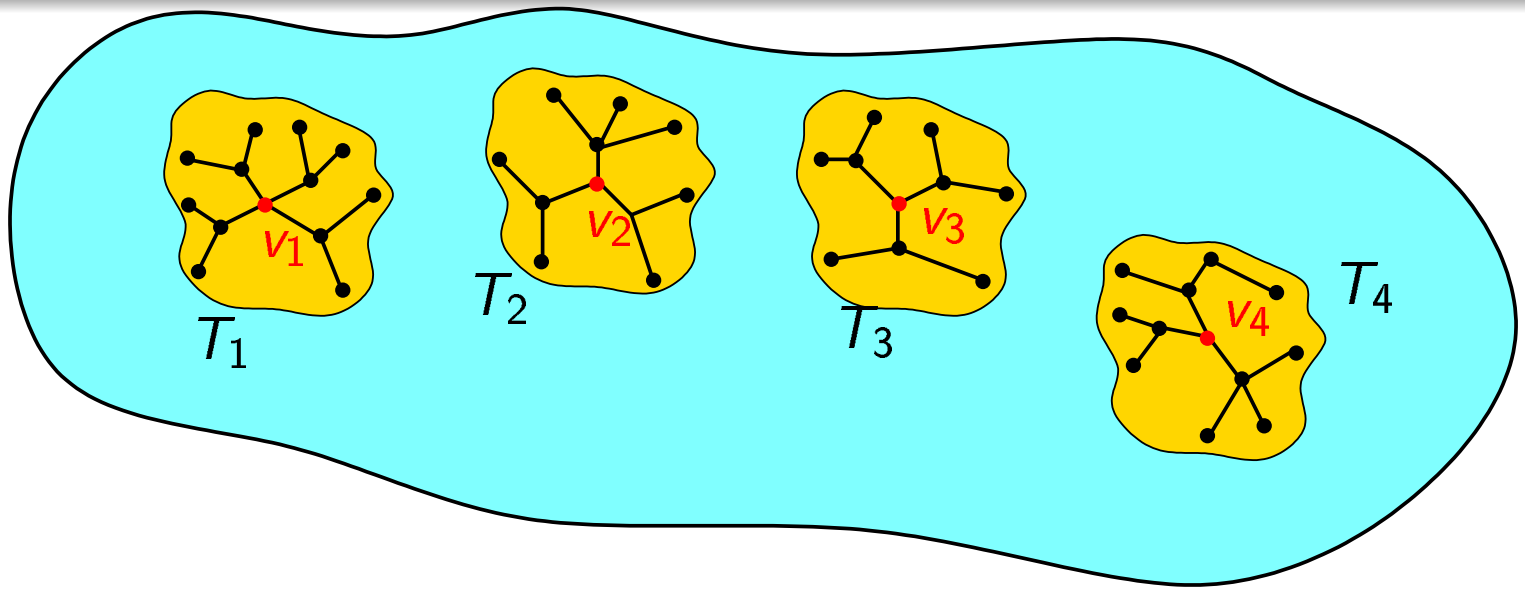
- Soit $v \in V$ quelconque et soit K l'ensemble des sommets du plus court graphe de découpe basé en v : K est inclus dans l'union de $4g$ plus courts chemins issus de v.
- Tout cycle non contractile passe par un sommet de K.

Algorithme

- On regroupe K en $O(gk)$ paquets de sorte que deux sommets distincts dans un paquet soient à distance au moins $k+1$.
- *Lemme*. Étant donné un paquet P, on peut, en temps $O(n)$, trouver un plus court lacet non contractile basé en un sommet dans P, s'il y en a un de longueur au plus k.



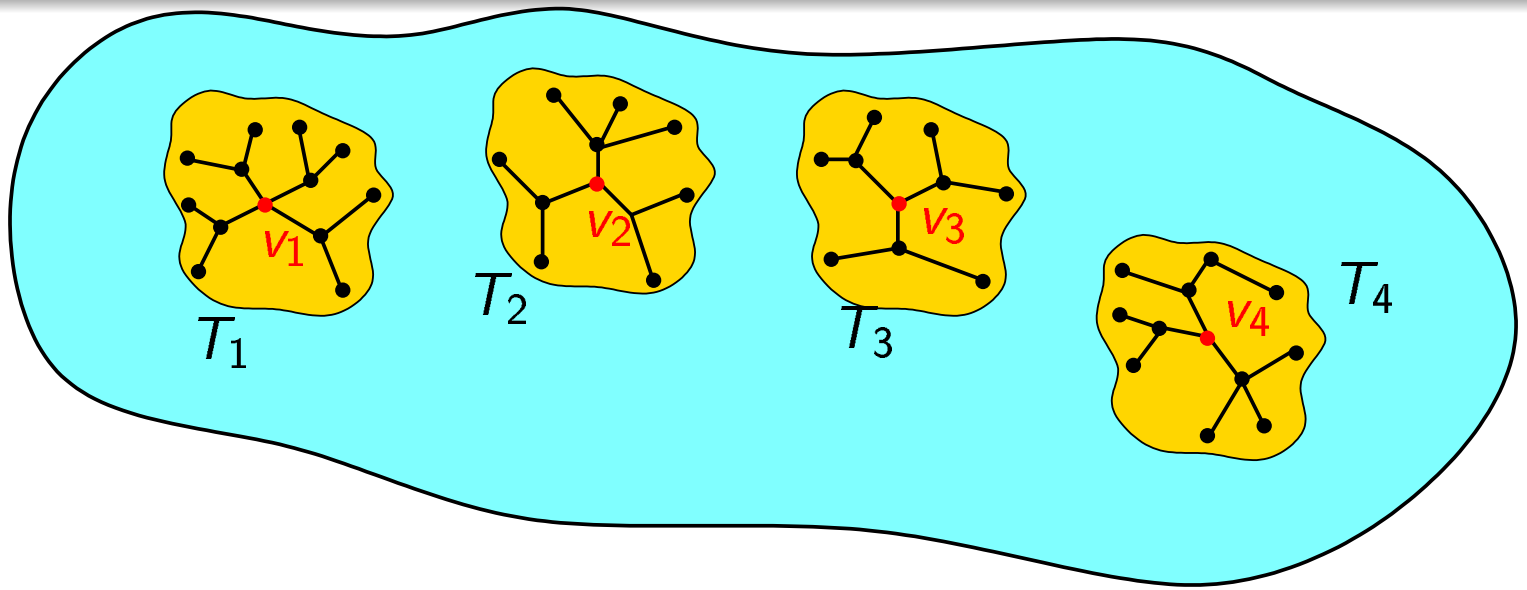
Parallélisation



Démonstration du lemme

- Pour $v_i \in P$, calculer un arbre de plus courts chemins T_i de profondeur $\lfloor k/2 \rfloor$ issu de v_i. Les T_i sont disjoints.
- Étendre l'union des T_i en un arbre couvrant T de G.
- S'il existe un plus court lacet non contractile à sommets dans T_i, il croise exactement une arête de $C = (E(G) \setminus T)^*$.
- Un lacet croisant exactement une arête c de C est contractile si et seulement si c sépare C en deux composantes connexes, l'une étant un arbre.

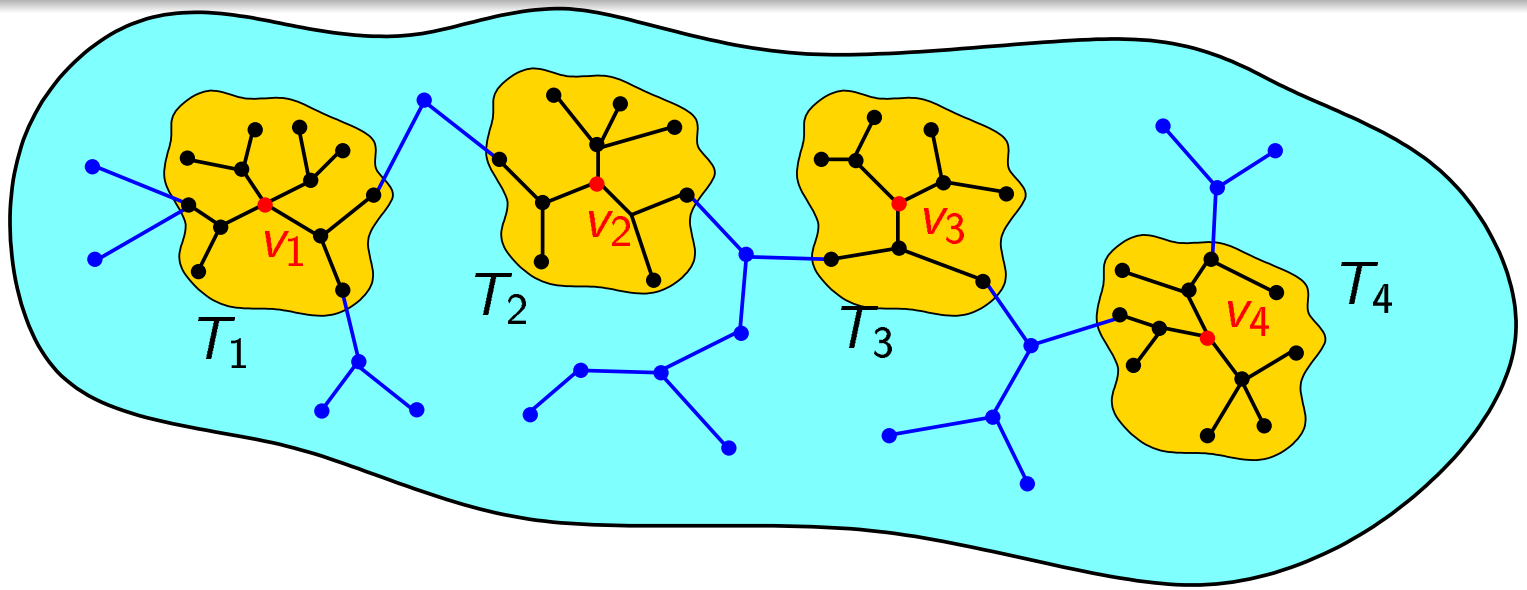
Parallélisation



Démonstration du lemme

- Pour $v_i \in P$, calculer un arbre de plus courts chemins T_i de profondeur $\lfloor k/2 \rfloor$ issu de v_i. Les T_i sont disjoints.
- Étendre l'union des T_i en un arbre couvrant T de G.
- S'il existe un plus court lacet non contractile à sommets dans T_i, il croise exactement une arête de $C = (E(G) \setminus T)^*$.
- Un lacet croisant exactement une arête c de C est contractile si et seulement si c sépare C en deux composantes connexes, l'une étant un arbre.

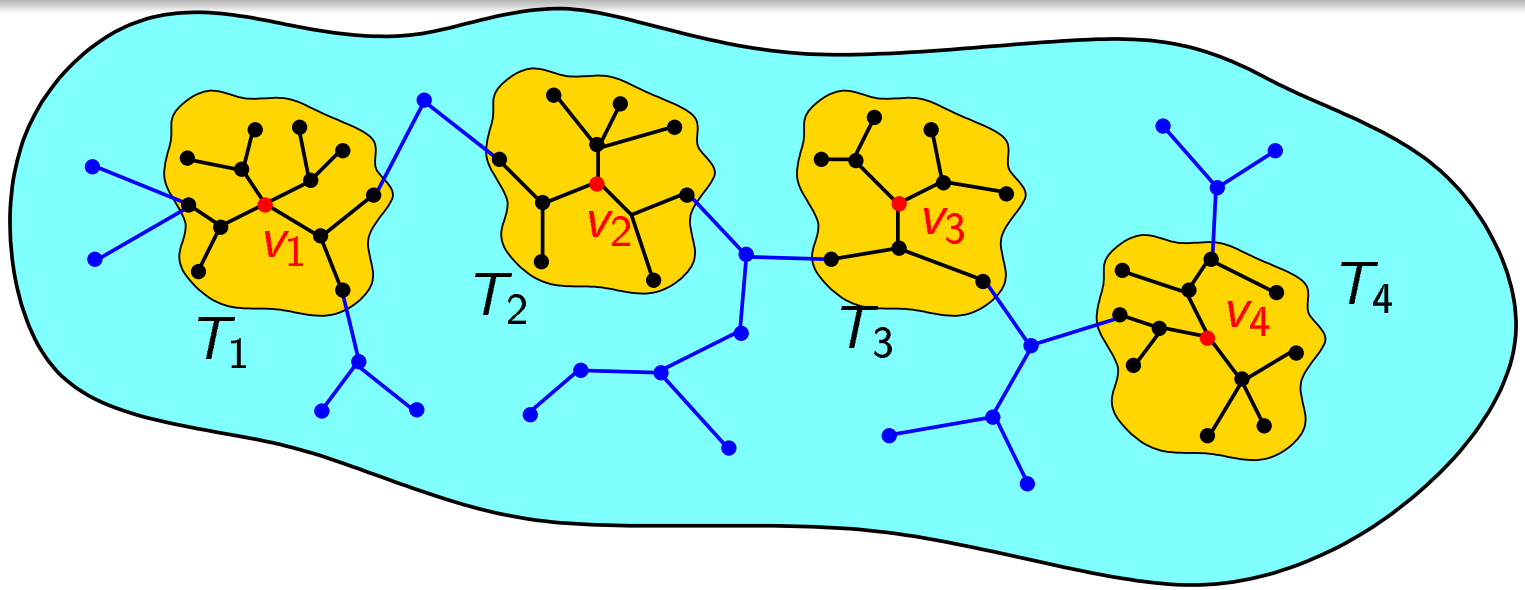
Parallélisation



Démonstration du lemme

- Pour $v_i \in P$, calculer un arbre de plus courts chemins T_i de profondeur $\lfloor k/2 \rfloor$ issu de v_i. Les T_i sont disjoints.
- Étendre l'union des T_i en un arbre couvrant T de G.
- S'il existe un plus court lacet non contractile à sommets dans T_i, il croise exactement une arête de $C = (E(G) \setminus T)^*$.
- Un lacet croisant exactement une arête c de C est contractile si et seulement si c sépare C en deux composantes connexes, l'une étant un arbre.

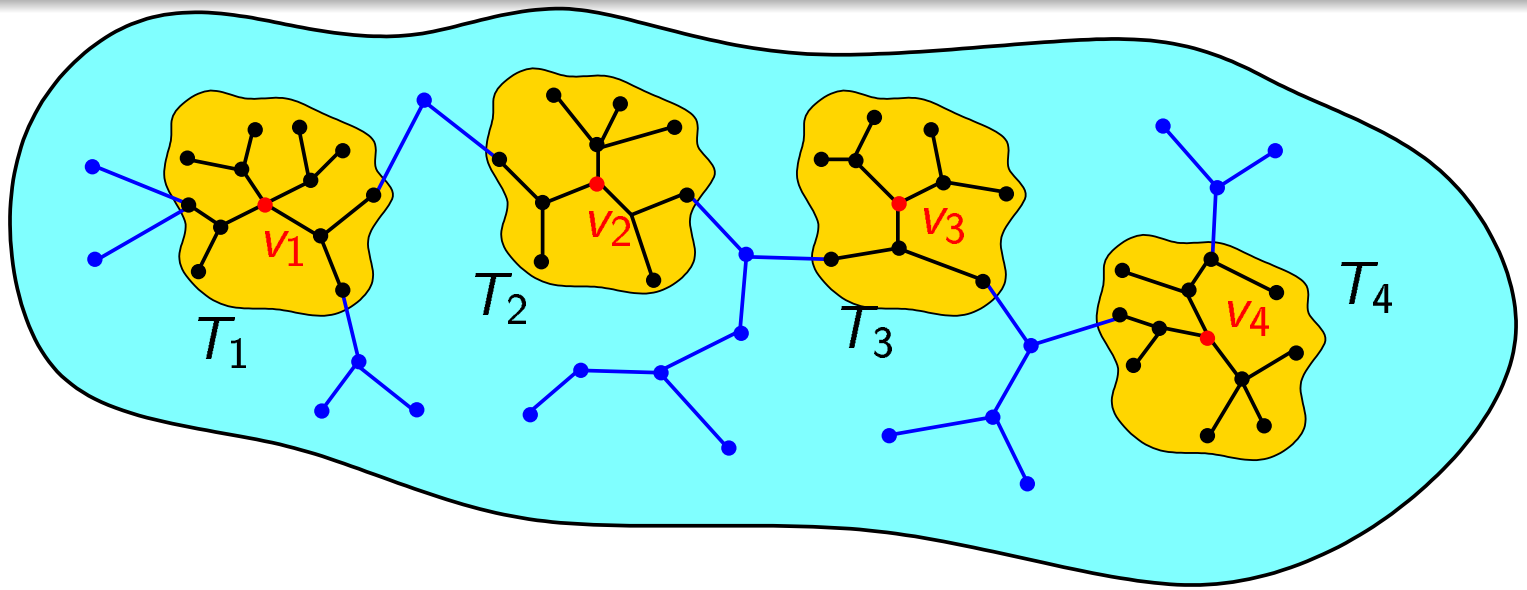
Parallélisation



Démonstration du lemme

- Pour $v_i \in P$, calculer un arbre de plus courts chemins T_i de profondeur $\lfloor k/2 \rfloor$ issu de v_i. Les T_i sont disjoints.
- Étendre l'union des T_i en un arbre couvrant T de G.
- S'il existe un plus court lacet non contractile à sommets dans T_i, il croise exactement une arête de $C = (E(G) \setminus T)^*$.
- Un lacet croisant exactement une arête c de C est contractile si et seulement si c sépare C en deux composantes connexes, l'une étant un arbre.

Parallélisation



Démonstration du lemme

- Pour $v_i \in P$, calculer un arbre de plus courts chemins T_i de profondeur $\lfloor k/2 \rfloor$ issu de v_i. Les T_i sont disjoints.
- Étendre l'union des T_i en un arbre couvrant T de G.
- S'il existe un plus court lacet non contractile à sommets dans T_i, il croise exactement une arête de $C = (E(G) \setminus T)^*$.
- Un lacet croisant exactement une arête c de C est contractile si et seulement si c sépare C en deux composantes connexes, l'une étant un arbre.

4. Bornes, plus courtes courbes non triviales

Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.

De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.

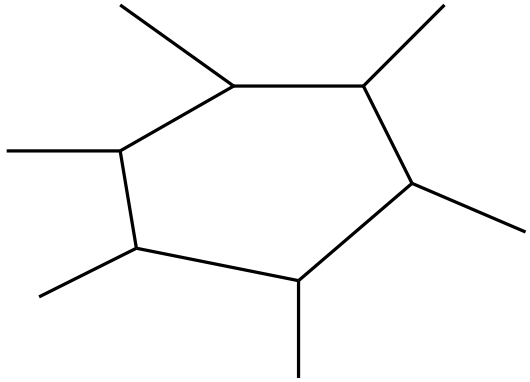
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.

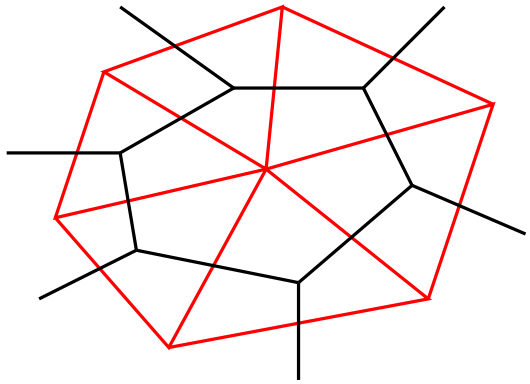
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.

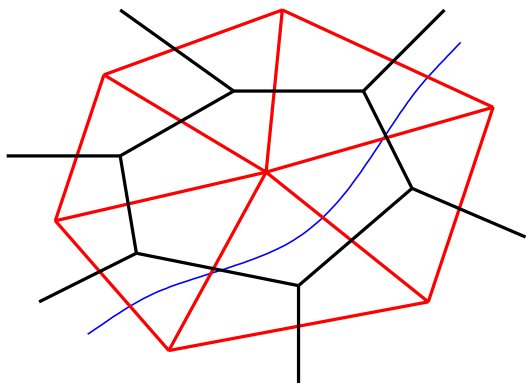
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.

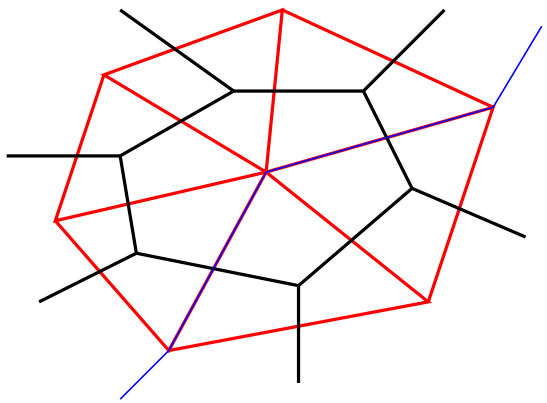
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.

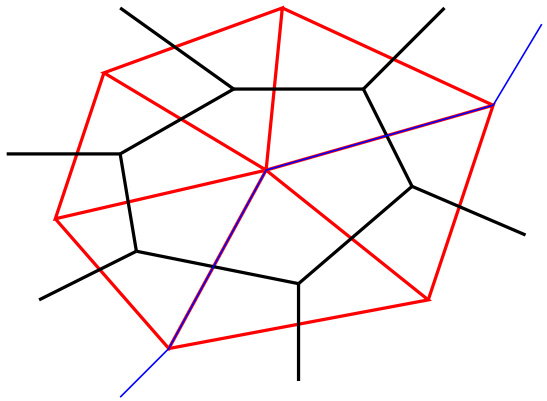
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

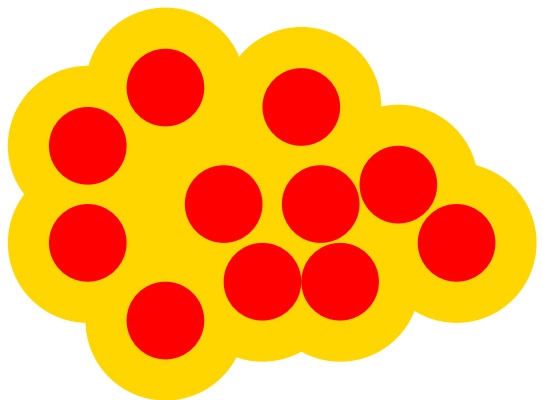
De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.



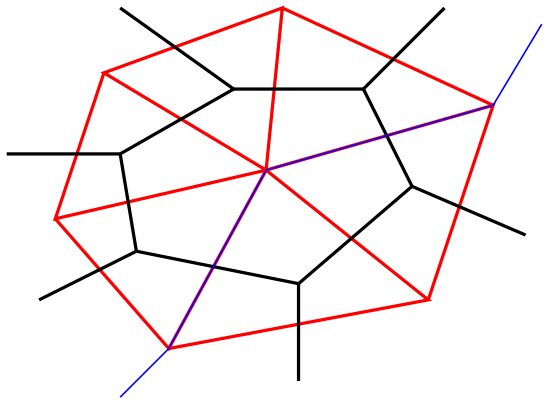
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

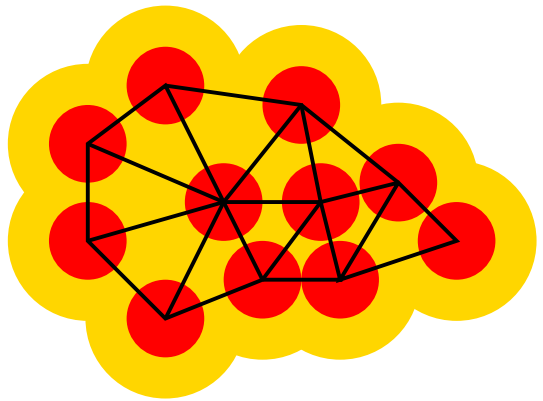
De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.



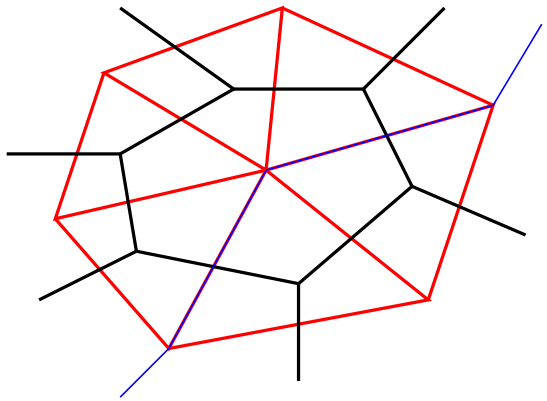
Dictionnaire

[CdV, Hubard, de Mesmay, 2014 (?)]

- M : surface à métrique des croisements trivalente, non pondérée à n sommets
- R : surface riemannienne d'aire a

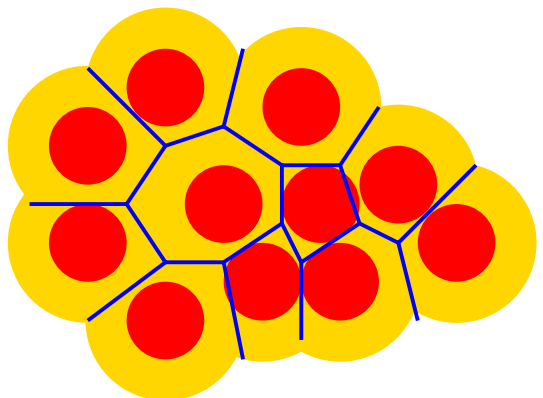
De M à R : triangles équilatéraux + lissage ($a = \sqrt{3}n/4$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\ell)$ sur R.



De R à M : ε-sample + triangulation de Delaunay ($n = \Theta(a/\varepsilon^2)$)

Courbe non triviale de longueur ℓ sur M $\Leftrightarrow$ courbe non triviale de longueur $\Theta(\varepsilon\ell)$ sur R.



Géométrie systolique

Résultats de géométrie systolique [Gromov, 1983], [Buser–Sarnak, 1994]

Une plus courte courbe non triviale sur une surface riemannienne R d'aire a est de longueur $O(\sqrt{a/g}\log g)$. Cette borne est optimale.

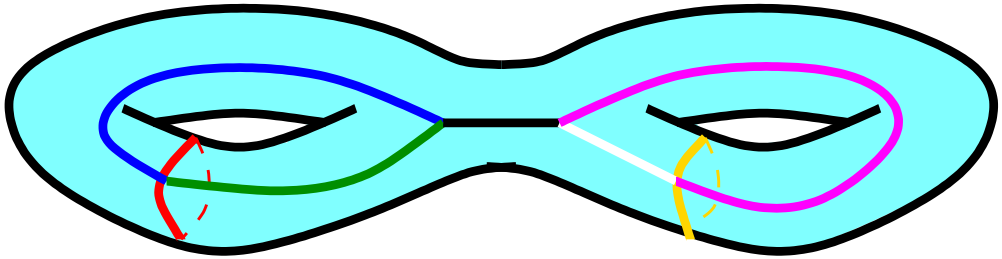
Corollaire immédiat

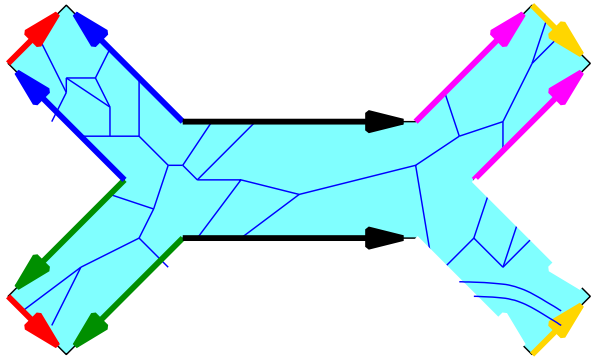
Une plus courte courbe non triviale sur une surface à métrique des croisements M non pondérée à n sommets est de longueur $O(\sqrt{n/g}\log g)$. Cette borne est optimale.

Améliore [Hutchinson, 1988], [Przytycka et Przytycki, 1990–1997], ...

5. Bornes, plus court graphe de découpe

Motifs de graphe de découpe





Décomposition canonique d'une surface de genre g

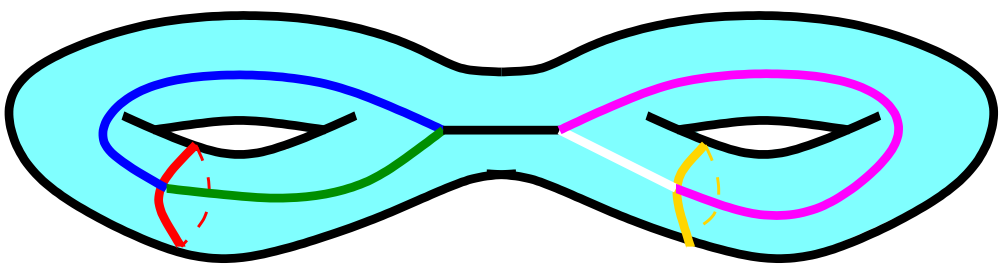
Soit C une carte combinatoire d'un graphe de découpe d'une surface de genre g.
On veut décomposer une surface à métrique des croisements $\mathscr{S}$ non pondérée de genre g et de complexité n en plongeant C sur $\mathscr{S}$. Quelle longueur faut-il ?

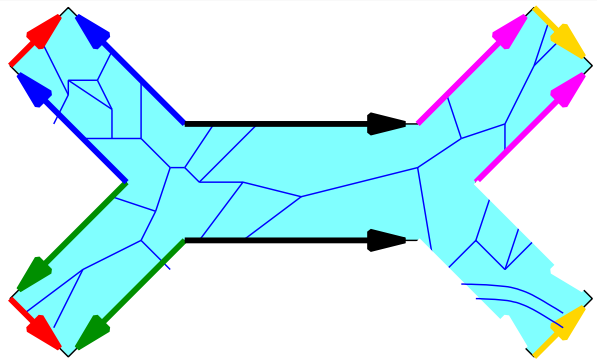
- Pas plus de $O(gn)$ [Lazarus, Pocchiola, Vegter, Verroust, 2001].
- On montre « presque » que pour tout choix de C, il faut parfois $\Omega(n^{7/6})$ [CdV, Hubard, de Mesmay, 2014 (?)].

Idée de la démonstration

Raffinement de [Guth, Parlier, Young, 2011]

- Pour tout g, soit C_g une carte combinatoire d'un graphe de découpe d'une surface de genre g. On montre qu'une surface à métrique des croisements trivalente, non pondérée, aléatoire à n sommets, si elle a genre g, n'a pas de plongement de C_g de longueur $\leq n^{7/6-\varepsilon}$.
- Il y a $\sim n^{n/2}$ surfaces à métrique des croisements trivalentes, non pondérées. Elles sont de genre entre 0 et $(n+2)/4$.
- Soit $g \geq 0$; le nombre de surfaces à métrique des croisements trivalentes à n sommets contenant C_g de longueur $\leq L$ est $f_{n,L}(g) \leq 2^{O(n)}(L/g)^{12g}$.





Idée de la démonstration

Raffinement de [Guth, Parlier, Young, 2011]

- Pour tout g, soit C_g une carte combinatoire d'un graphe de découpe d'une surface de genre g. On montre qu'une surface à métrique des croisements trivalente, non pondérée, aléatoire à n sommets, si elle a genre g, n'a pas de plongement de C_g de longueur $\leq n^{7/6-\varepsilon}$.
- Il y a $\sim n^{n/2}$ surfaces à métrique des croisements trivalentes, non pondérées. Elles sont de genre entre 0 et $(n+2)/4$.
- Soit $g \geq 0$; le nombre de surfaces à métrique des croisements trivalentes à n sommets contenant C_g de longueur $\leq L$ est $f_{n,L}(g) \leq 2^{O(n)}(L/g)^{12g}$.
- Si $L = n^{7/6-\varepsilon}$, alors

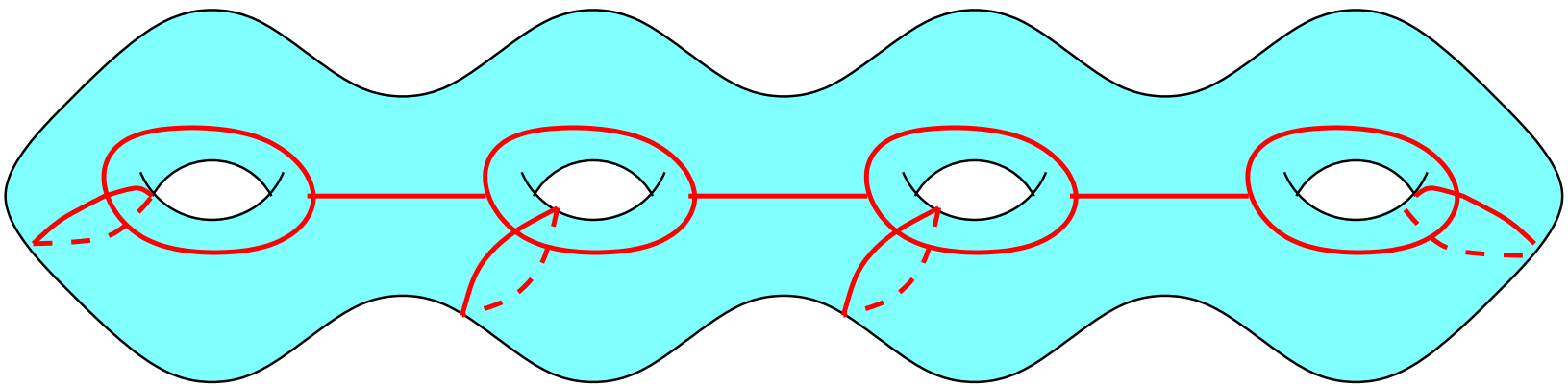
$$\sum_{g=0}^{(n+2)/4} f_{n,L}(g) \ll n^{n/2}.$$

Perspectives

Perspectives : courbes sur les surfaces

Complexité du calcul de décompositions les plus courtes

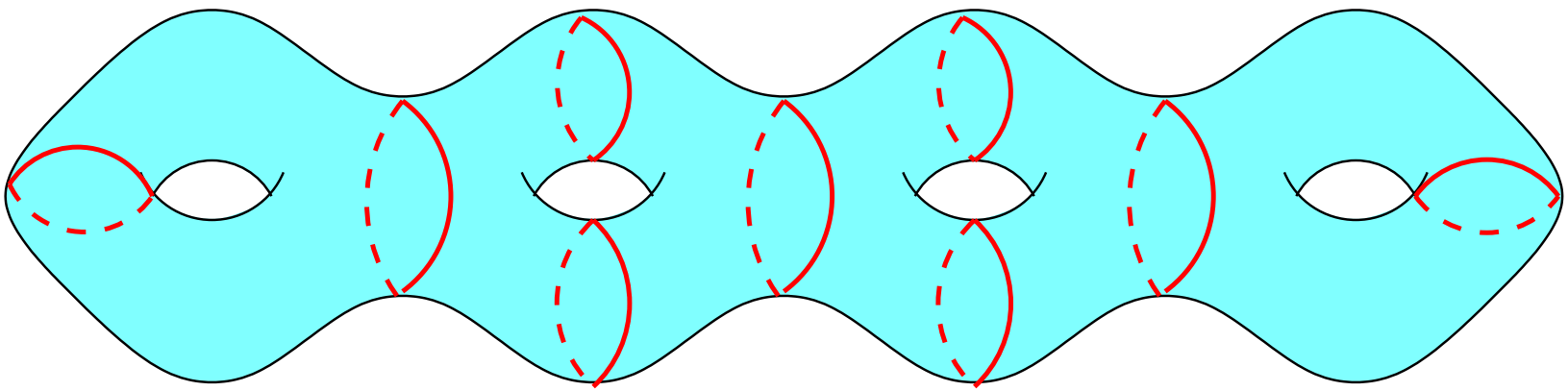
- graphe de découpe sans fixer les sommets :
  - C'est NP-dur, est-ce abordable à g fixé (FPT) ? W[1]-dur ?
  - Meilleure approximation que [Erickson et Har-Peled, 2004] ?
- décompositions en pantalons ?
- découpage en une surface de genre zéro ?



Perspectives : courbes sur les surfaces

Complexité du calcul de décompositions les plus courtes

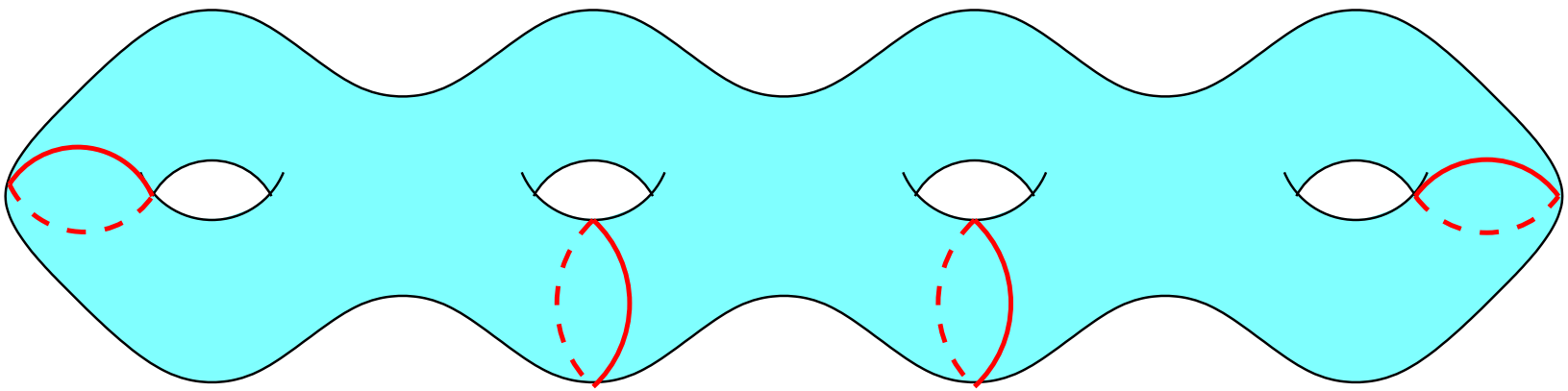
- graphe de découpe sans fixer les sommets :
  - C'est NP-dur, est-ce abordable à g fixé (FPT) ? W[1]-dur ?
  - Meilleure approximation que [Erickson et Har-Peled, 2004] ?
- décompositions en pantalons ?
- découpage en une surface de genre zéro ?



Perspectives : courbes sur les surfaces

Complexité du calcul de décompositions les plus courtes

- graphe de découpe sans fixer les sommets :
  - C'est NP-dur, est-ce abordable à g fixé (FPT) ? W[1]-dur ?
  - Meilleure approximation que [Erickson et Har-Peled, 2004] ?
- décompositions en pantalons ?
- découpage en une surface de genre zéro ?



Pour en savoir plus : HDR disponible sur `http://www.di.ens.fr/~colin/`