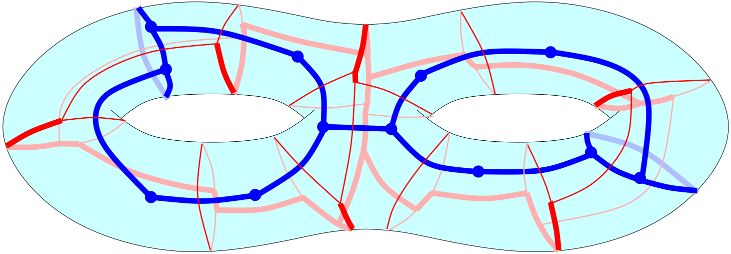
Graphs on surfaces: topological algorithms

Éric Colin de Verdière

École normale supérieure (Paris) and CNRS

ÉPIT 2016



More Results. . .

Shortest non-trivial closed curves

g: genus, k: size of output (number of edges of shortest non-trivial closed curve)

	non-directed	directed
weighted	**$O(n^2 \log n)$** [Erickson–Har-Peled'04] $O(g^{3/2} n^{3/2} \log n)$ non-sep, $g^{O(g)} n^{3/2}$ 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 al.'11]	$O(n^2 \log n)$ [Cabello–CdV–Lazarus'10] **$O(g^{1/2} n^{3/2} \log n)$** [Cab–CdV–Laz] $2^{O(g)} n \log n$ non-sep [Erickson–Nayyeri'11] **$O(g^2 n \log n)$** non-sep, $g^{O(g)} n \log n$ non-contr } [Erickson'11] **$O(g^3 n \log n)$** non-contr [Fox'13]
unweighted	$O(n^3)$ [Thomassen'90] **$O(n^2)$** [Cabello–CdV–Lazarus'10] **$O(gnk)$** [Cabello–CdV–Lazarus'10] $O(gn)$ for 2-approx [Erickson–Har-Peled'04] **$O(gn/\varepsilon)$** for $(1+\varepsilon)$-approx [Cabello–CdV–Lazarus'10]	**$O(n^2)$** [Cabello–CdV–Lazarus'10] **$O(gnk)$** [Cabello–CdV–Lazarus'16]

Other problems solved

Curves and decompositions

- shortest curves: shortest splitting cycle [Chambers et al., 2006], shortest essential cycle [Erickson, Worah, 2010], some non-separating cycle as short as possible in its homotopy class [Cabello et al., 2008];
- decompositions: canonical system of loops [Lazarus et al., 2001];
- bounds on their length [CdV, Hubard, de Mesmay, 2015].

Decision problems on deformations

Deciding homotopy / isotopy for paths / closed curves / graphs [Lazarus, Rivaud, 2012; Erickson, Whittlesey, 2013; CdV, de Mesmay, 2013; . . .].

Crossings

- drawing a graph in the plane with $\leq k$ crossings [Kawarabayashi, Reed, 2007];
- making curves minimally crossing [Matoušek et al., 2013].

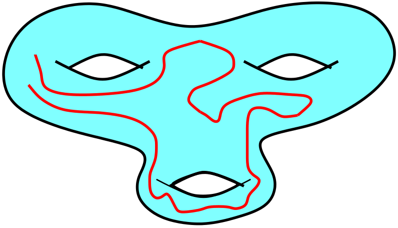
Flows, cuts, cycle bases, homology bases

- min cut and max flow [Chambers et al., 2012], . . . ;
- cycle and homology bases [Borradaile et al., 2016].

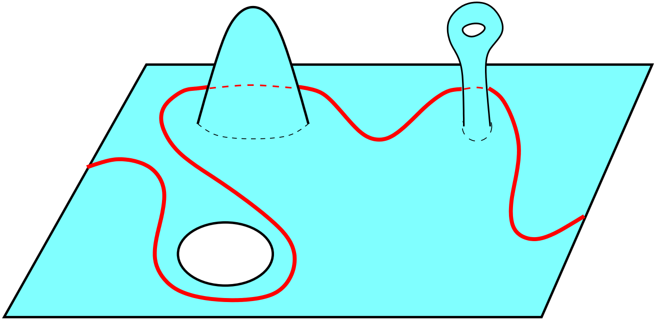
MANY other algorithms for graphs embeddable on a fixed surface.

Path tightening

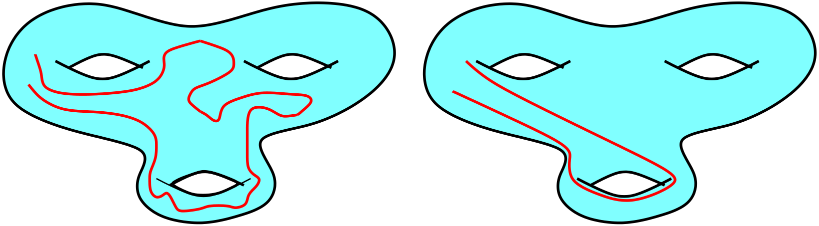
Problem



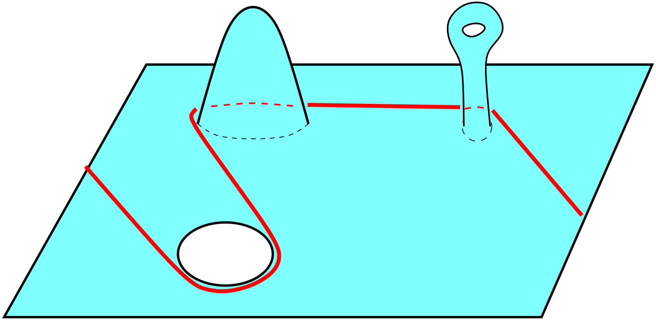
Local optimization doesn't work!



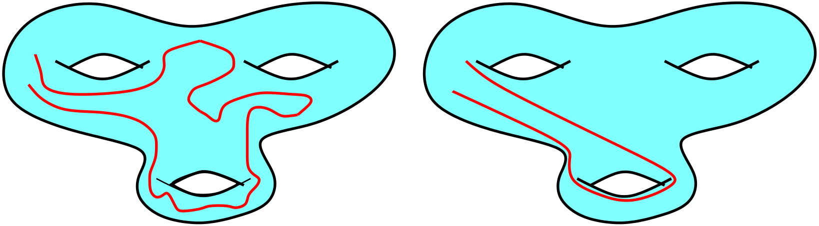
Problem



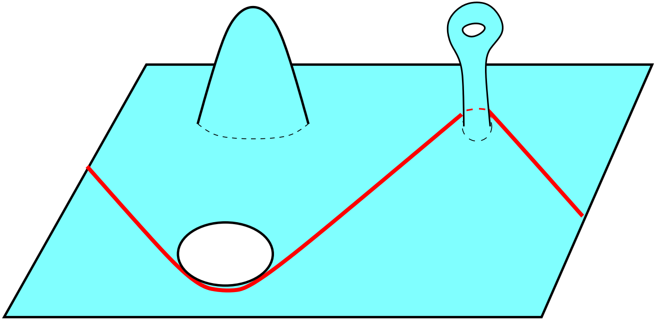
Local optimization doesn't work!



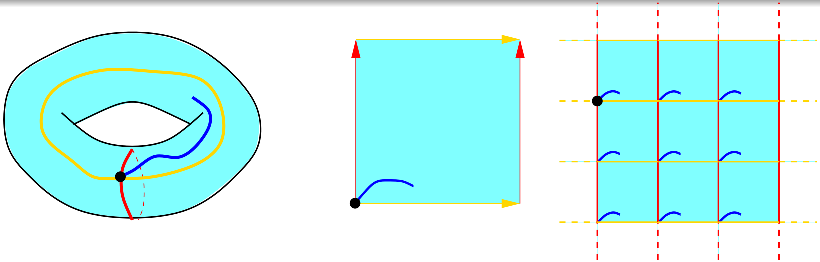
Problem



Local optimization doesn't work!

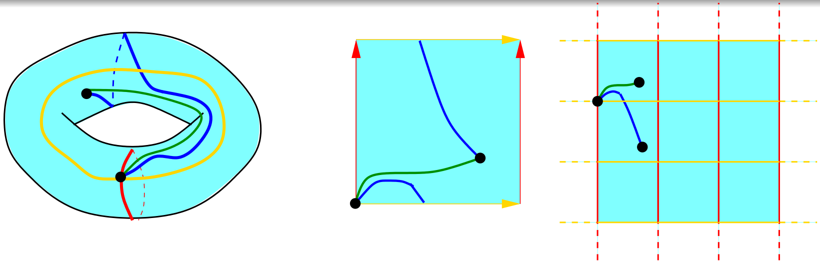


Universal cover $\tilde{\mathscr{S}}$ of the torus ($g = 1$ handle)



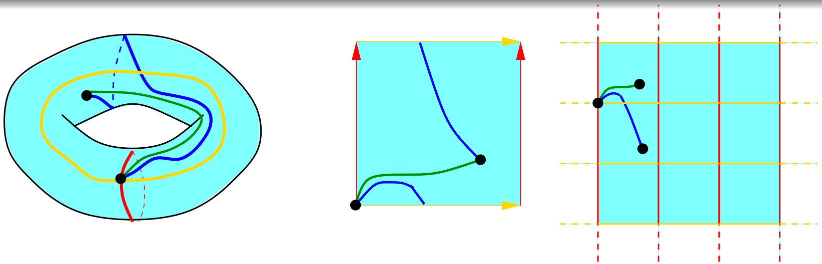
- Every path on $\mathscr{S}$ can be "lifted" to $\tilde{\mathscr{S}}$.
- Two paths are homotopic iff they admit lifts with the same endpoints.

Universal cover $\tilde{\mathscr{S}}$ of the torus ($g = 1$ handle)



- Every path on $\mathscr{S}$ can be "lifted" to $\tilde{\mathscr{S}}$.
- Two paths are homotopic iff they admit lifts with the same endpoints.

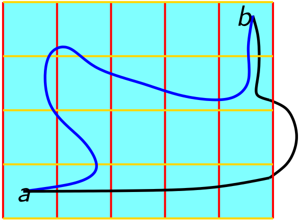
Universal cover $\tilde{\mathscr{S}}$ of the torus ($g = 1$ handle)

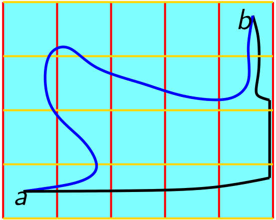


- Every path on $\mathscr{S}$ can be "lifted" to $\tilde{\mathscr{S}}$.
- Two paths are homotopic iff they admit lifts with the same endpoints.

It "suffices" to compute shortest paths in $\tilde{\mathscr{S}}$.

Properties of a shortest cut graph with one vertex





Lemma

If the initial cycles form a shortest one-vertex cut graph, then the “horizontals” and the “verticals” are shortest paths.

Corollary

The shortest path connecting a to b in $\tilde{\mathscr{S}}$ remains in the rectangle containing a and b.

Algorithm ($g = 1$ handle)

- Compute a shortest one-vertex cut graph;

$\rightarrow$ $O(n^2 \log n)$

$\rightarrow$ Actually, $O(n \log n)$.

- count the algebraic number of crossings p with the "horizontal" cycle and q with the "vertical" cycle;

$\rightarrow$ $p, q = O(k)$ where k=complexity of the input path.

- glue $p \times q$ copies of the square;

$\rightarrow$ $O(pqn) = O(k^2 n)$.

$\rightarrow$ Actually, $O(p + q)$ copies suffice: $O(kn)$.

- compute a shortest path between the corresponding points of the grid.

$\rightarrow$ $O(k^2 n \log(k^2 n))$ (Dijkstra).

$\rightarrow$ Actually, $O(kn)$.

Total: $O(n^2 \log n + k^2 n \log kn)$.

Actually: $O(n \log n + kn)$ [CdV, Erickson, 2006]+[CdV, Jouhet, ∞].

Algorithm ($g = 1$ handle)

- Compute a shortest one-vertex cut graph;

$\rightarrow O(n^2 \log n)$

$\rightarrow$ Actually, $O(n \log n)$.

- count the algebraic number of crossings p with the "horizontal" cycle and q with the "vertical" cycle;

$\rightarrow p, q = O(k)$ where k=complexity of the input path.

- glue $p \times q$ copies of the square;

$\rightarrow O(pqn) = O(k^2 n)$.

$\rightarrow$ Actually, $O(p + q)$ copies suffice: $O(kn)$.

- compute a shortest path between the corresponding points of the grid.

$\rightarrow O(k^2 n \log(k^2 n))$ (Dijkstra).

$\rightarrow$ Actually, $O(kn)$.

Total: $O(n^2 \log n + k^2 n \log kn)$.

Actually: $O(n \log n + kn)$ [CdV, Erickson, 2006]+[CdV, Jouhet, ∞].

Algorithm ($g = 1$ handle)

- Compute a shortest one-vertex cut graph;

$\to O(n^2 \log n)$

$\to$ Actually, $O(n \log n)$.

- count the algebraic number of crossings p with the "horizontal" cycle and q with the "vertical" cycle;

$\to p, q = O(k)$ where k=complexity of the input path.

- glue $p \times q$ copies of the square;

$\to O(pqn) = O(k^2 n)$.

$\to$ Actually, $O(p + q)$ copies suffice: $O(kn)$.

- compute a shortest path between the corresponding points of the grid.

$\to O(k^2 n \log(k^2 n))$ (Dijkstra).

$\to$ Actually, $O(kn)$.

Total: $O(n^2 \log n + k^2 n \log kn)$.

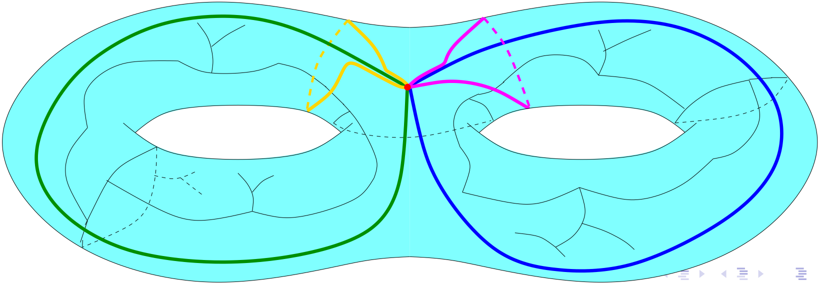
Actually: $O(n \log n + kn)$ [CdV, Erickson, 2006]+[CdV, Jouhet, ∞].

Okay, but what about $g \geq 2$ handles?

- The lifts of a shortest cut graph are not shortest paths.
- What is a good replacement for the shortest cut graph?
- What does the universal cover look like?

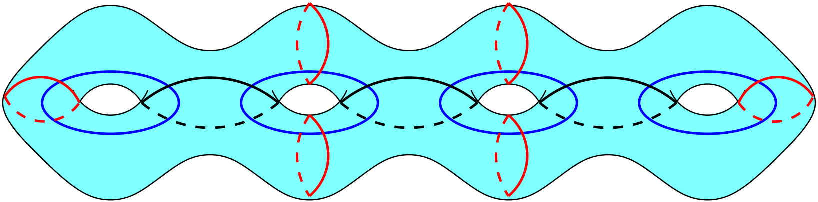
Okay, but what about $g \geq 2$ handles?

- The lifts of a shortest cut graph are not shortest paths.
- What is a good replacement for the shortest cut graph?
- What does the universal cover look like?



Okay, but what about $g \geq 2$ handles?

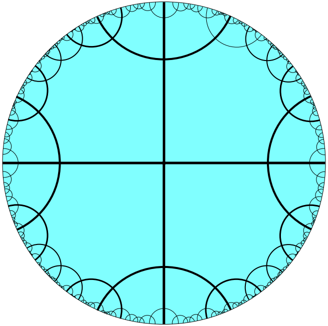
- The lifts of a shortest cut graph are not shortest paths.
- What is a good replacement for the shortest cut graph?
- What does the universal cover look like?

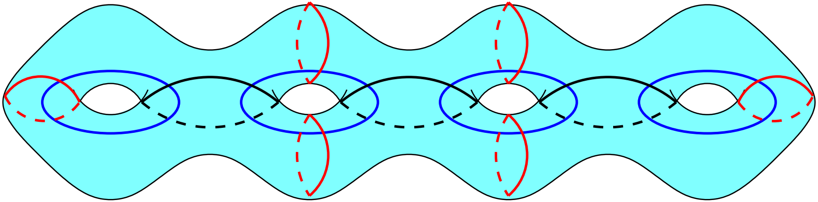


octagonal decomposition
each path lifts to a shortest path

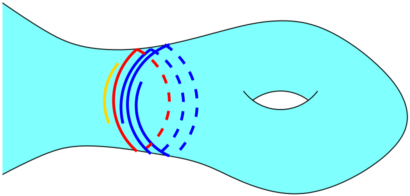
Okay, but what about $g \geq 2$ handles?

- The lifts of a shortest cut graph are not shortest paths.
- What is a good replacement for the shortest cut graph?
- What does the universal cover look like?





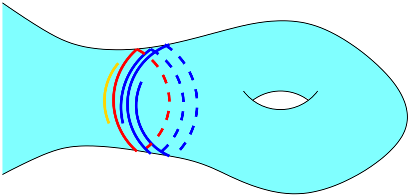
octagonal decomposition
each path lifts to a shortest path

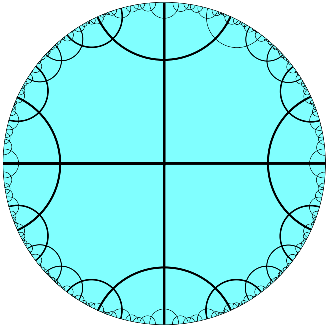


Lemma

- Let γ be a tight cycle,
- let p be a path "wrapping around" γ.
- Then p is tight.

Technical Lemma and Convexity





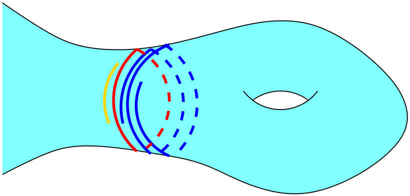
Lemma

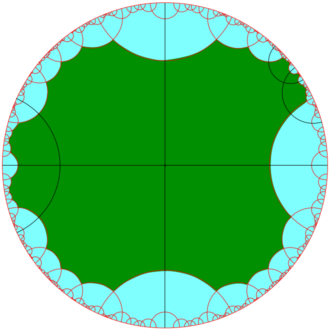
- Let γ be a tight cycle,
- let p be a path "wrapping around" γ.
- Then p is tight.

Convexity

- line: lift of a cycle
- half-plane: delimited by a line
- convex: intersection of half-planes
- the shortest point with endpoints in a convex set C stays in C.

Technical Lemma and Convexity





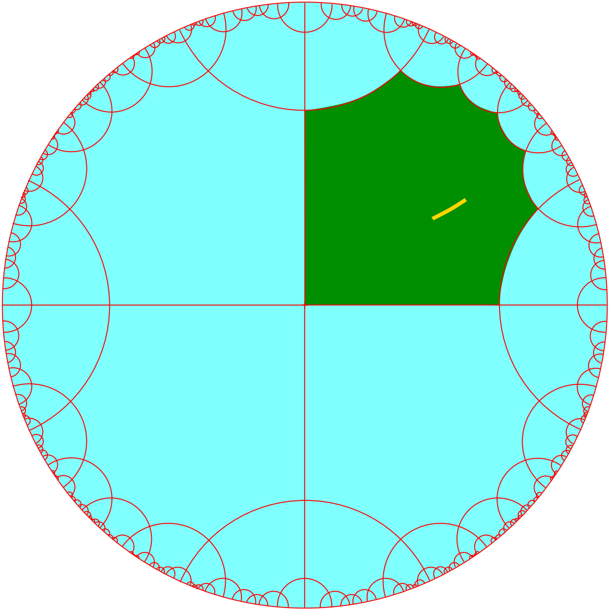
Lemma

- Let γ be a tight cycle,
- let p be a path "wrapping around" γ.
- Then p is tight.

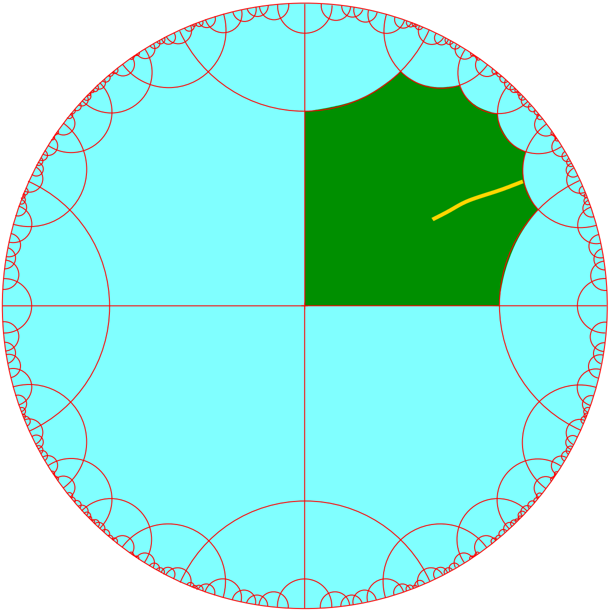
Convexity

- line: lift of a cycle
- half-plane: delimited by a line
- convex: intersection of half-planes
- the shortest point with endpoints in a convex set C stays in C.

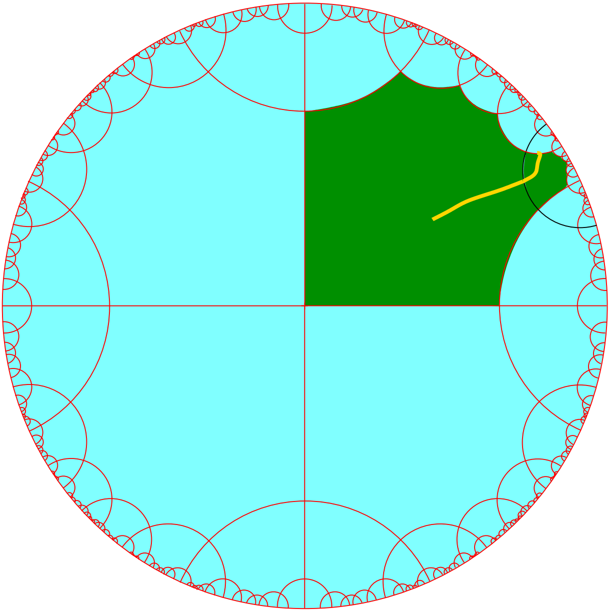
Building the convex region of $\tilde{\mathscr{S}}$



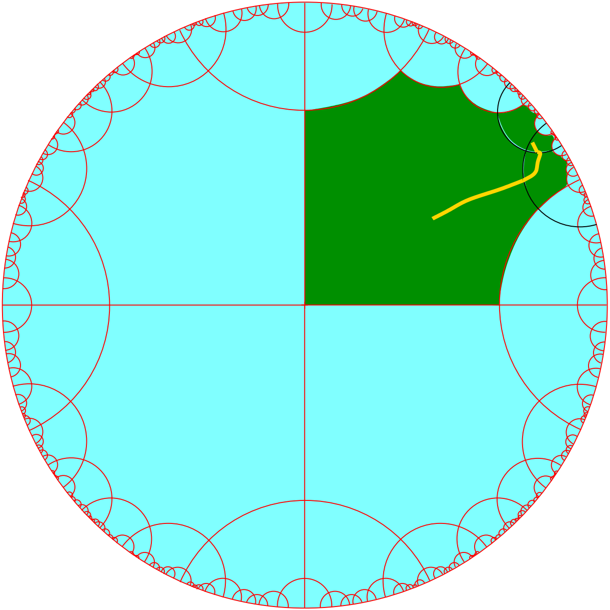
Building the convex region of $\tilde{\mathscr{S}}$



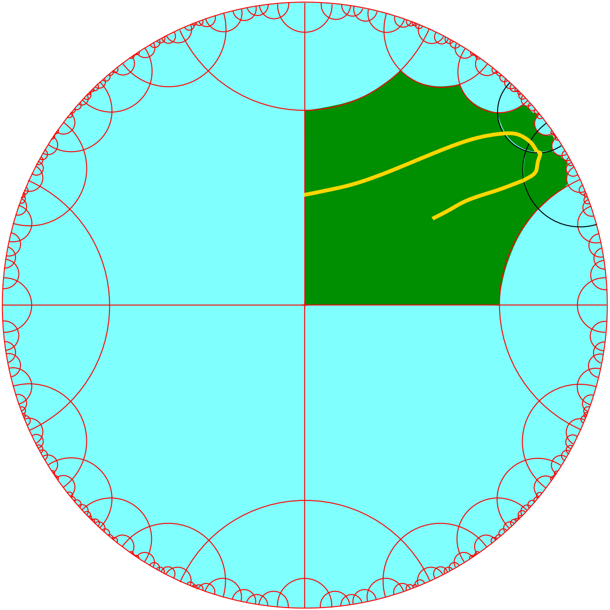
Building the convex region of $\tilde{\mathscr{S}}$



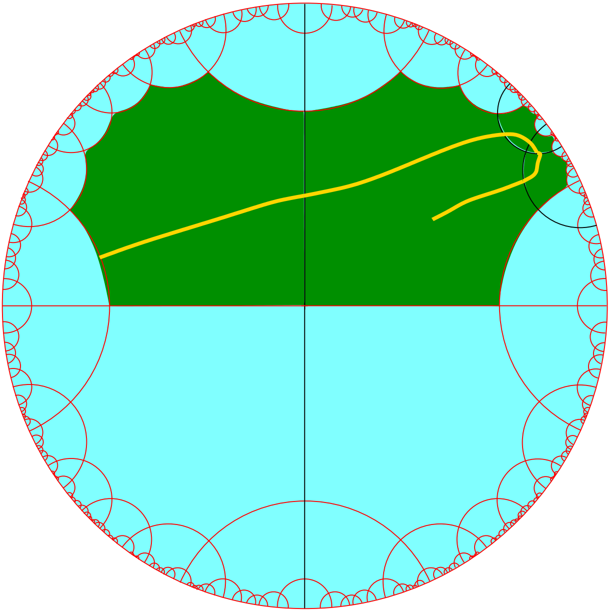
Building the convex region of $\tilde{\mathscr{S}}$



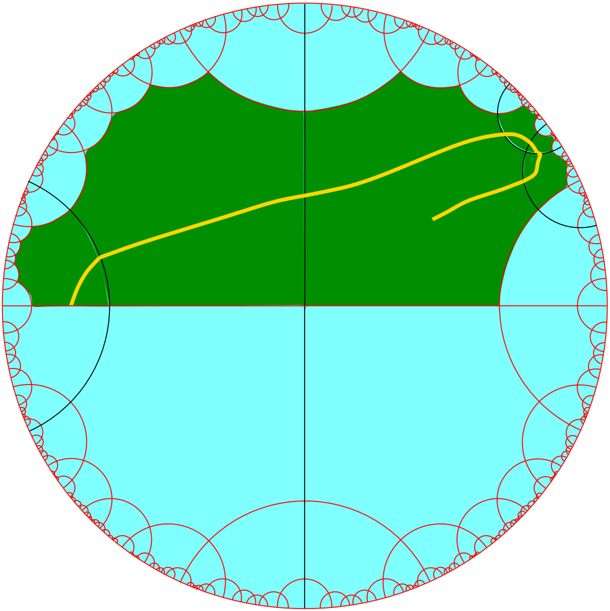
Building the convex region of $\tilde{\mathscr{S}}$



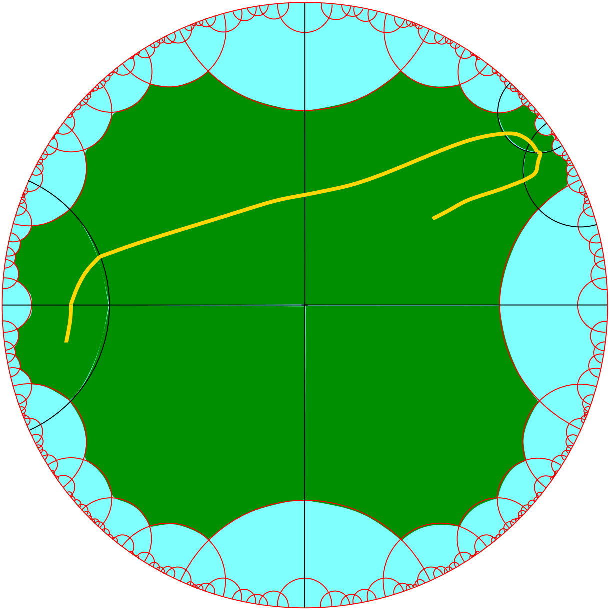
Building the convex region of $\tilde{\mathscr{S}}$



Building the convex region of $\tilde{\mathscr{S}}$



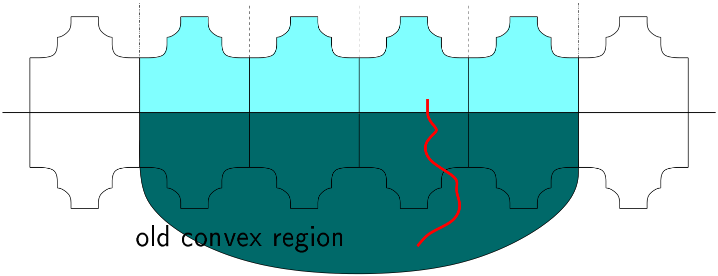
Building the convex region of $\tilde{\mathscr{S}}$



Building the convex region of $\tilde{\mathscr{S}}$, details

Incremental construction

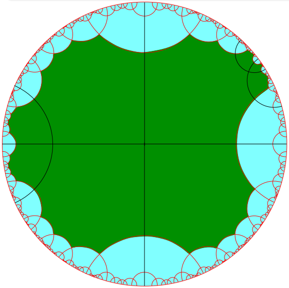
- Start with a copy of the octagon containing the source of $\tilde{p}$.
- When $\tilde{p}$ crosses a new line, augment the convex region.

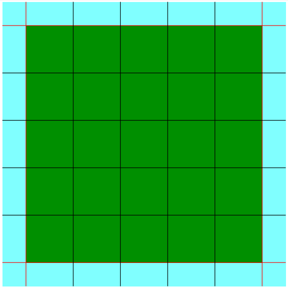


Complexity

Size of the convex region: hyperbolicity

If $\tilde{c}$ crosses m “lines”, the convex region contains $O(m)$ octagons.





Proof

- Area=O(perimeter): Indeed, iteratively remove an octagon farthest from the “center”. That octagon has ≥ 5 sides on the boundary, so each step decreases the perimeter;
- perimeter is $O(m)$, because:
  - at most $2m$ flat vertices on the boundary of the convex region
  - at most 6 corner vertices between two consecutive flat vertices.

Building the octagonal decomposition: technical lemma

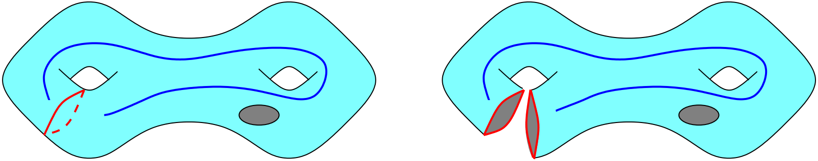
Arc: path with endpoints on the boundary of the surface

Lemma

On a surface, let

- α be a tight simple curve that is either an arc or a non-contractible cycle,
- β be a simple path or cycle disjoint from α.

Then β is tight on S iff it is tight on $S \backslash\backslash \alpha$.



Building the octagonal decomposition

skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

Building the octagonal decomposition ▸ skip

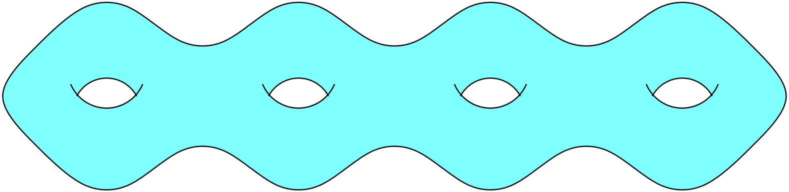
On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

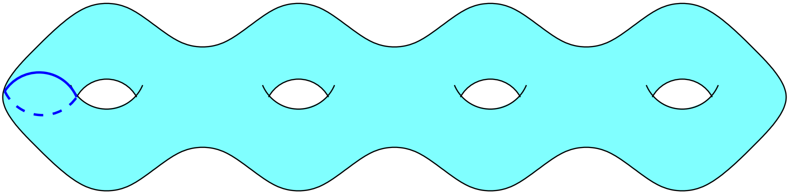
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

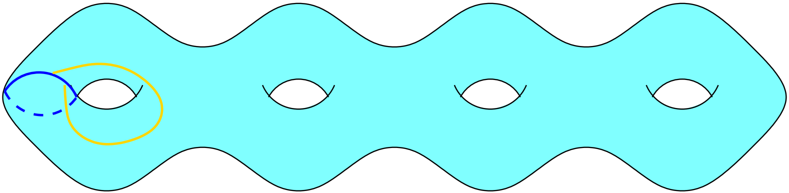
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

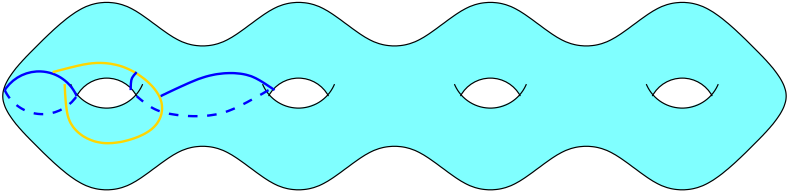


Building the octagonal decomposition

skip

On a surface with complexity n, genus g, and b boundaries:

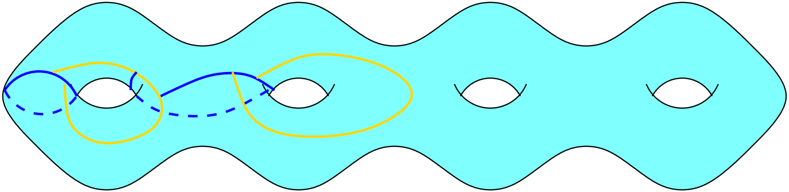
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

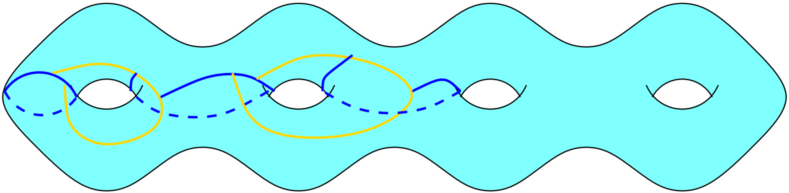
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

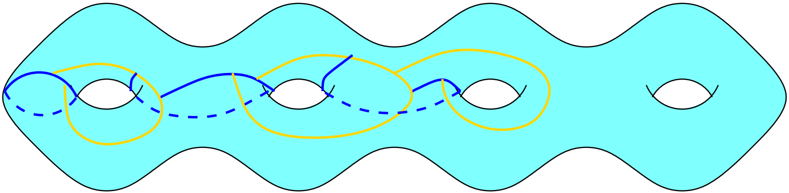
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

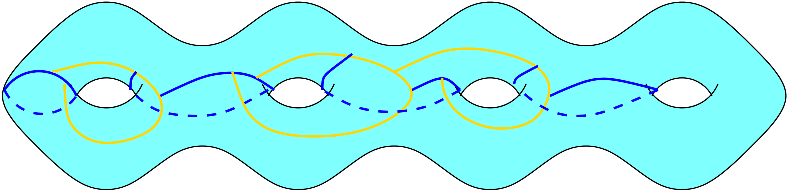
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

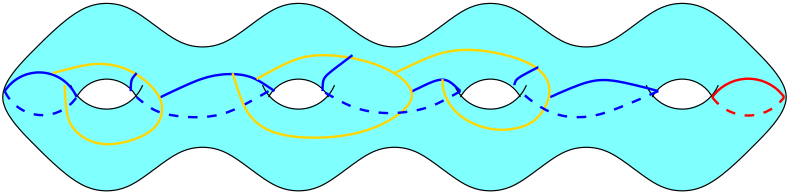
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

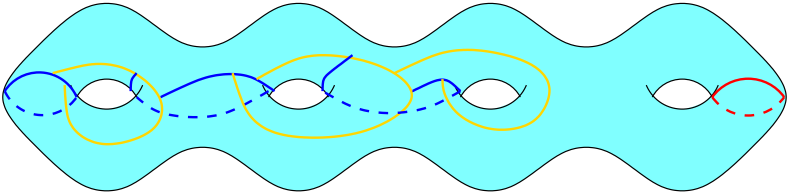
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

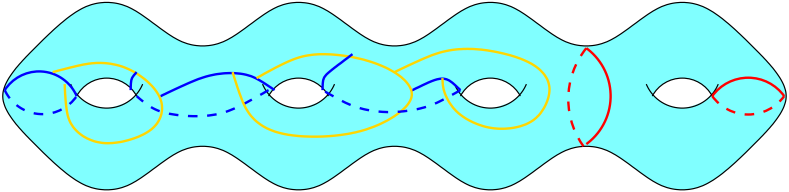
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

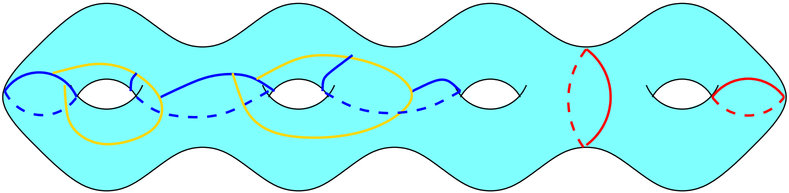


Building the octagonal decomposition

▸ skip

On a surface with complexity n, genus g, and b boundaries:

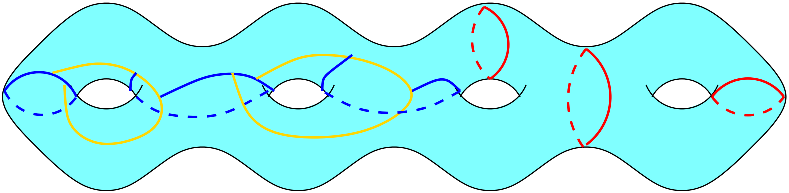
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

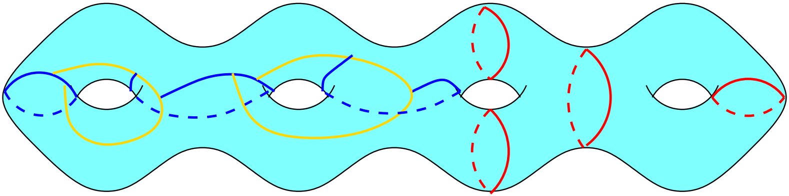
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]

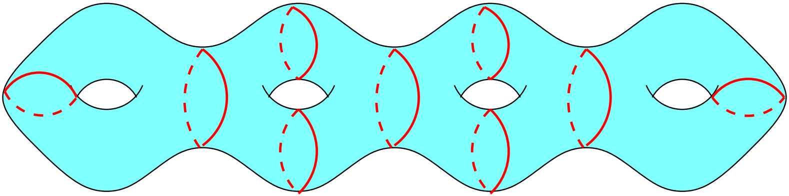


Building the octagonal decomposition

skip

On a surface with complexity n, genus g, and b boundaries:

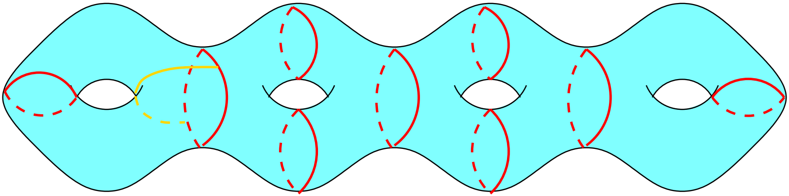
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

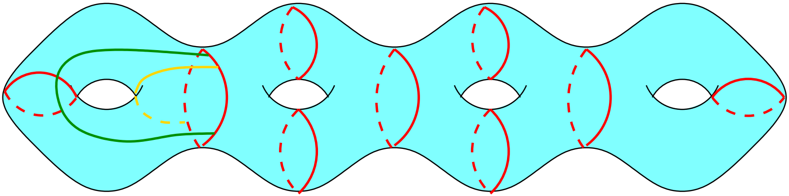
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

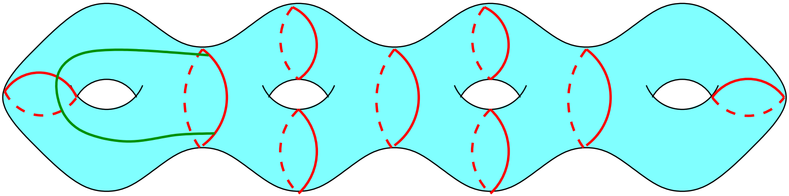
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

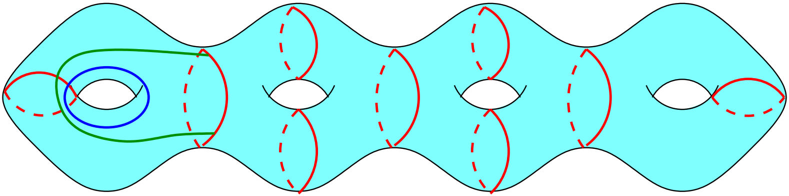
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

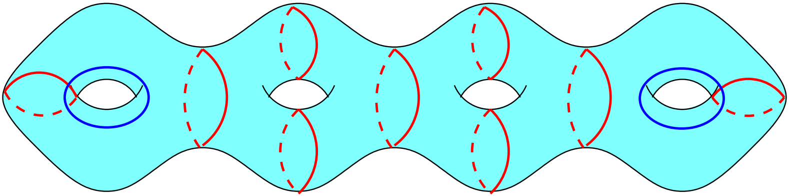
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

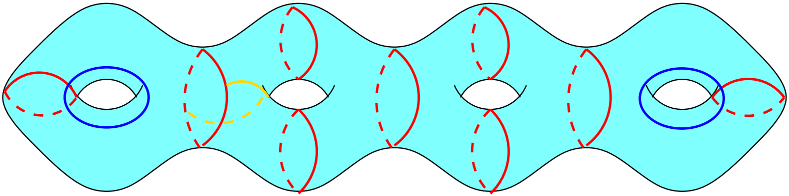
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

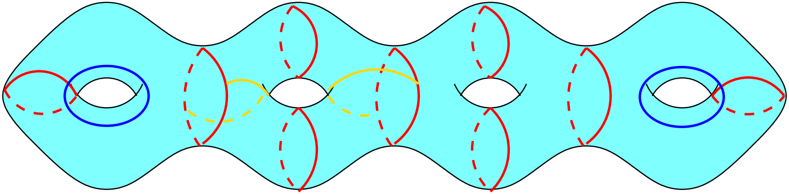
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

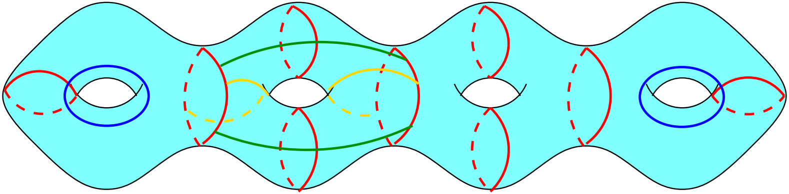
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

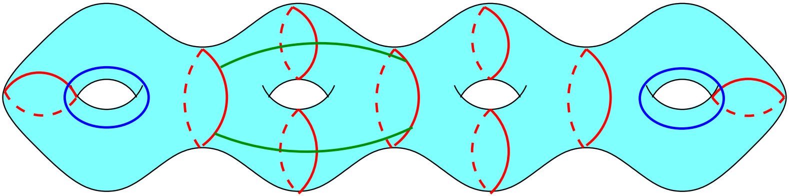
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

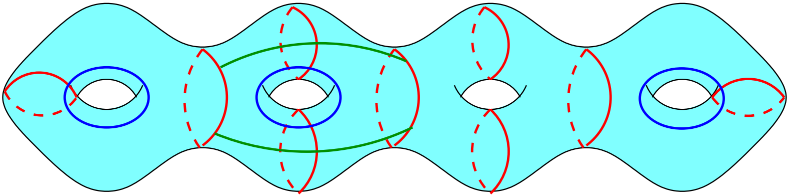
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

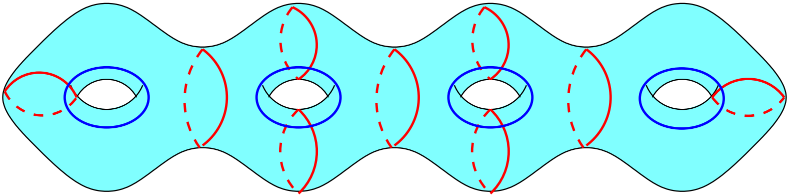
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

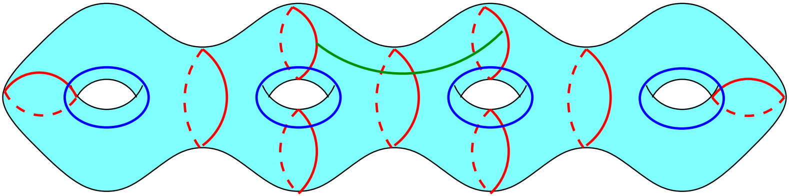
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

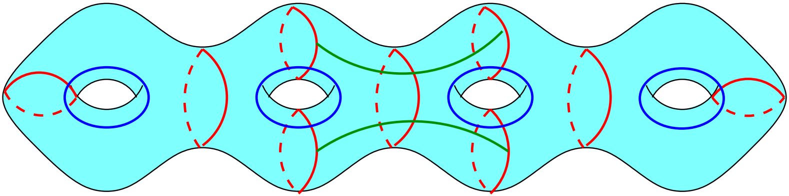
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

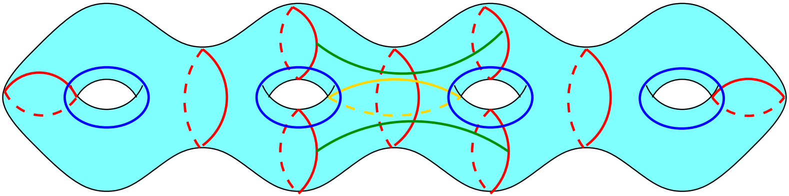
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

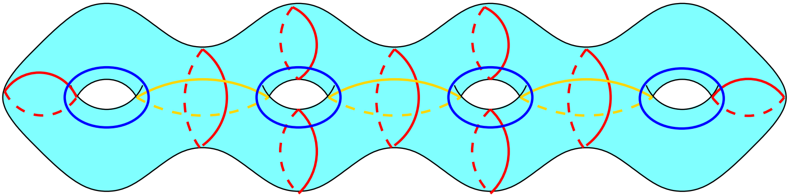
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

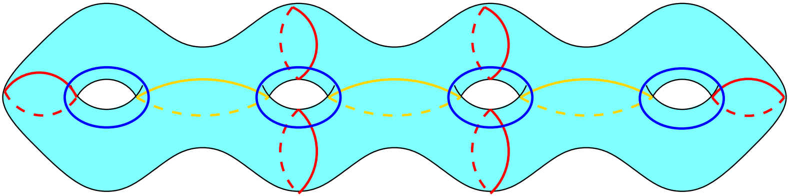
1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003],[Frederickson, 1987]



Building the octagonal decomposition ▸ skip

On a surface with complexity n, genus g, and b boundaries:

1. computation of a shortest non-separating cycle: $O(n^2 \log n)$
2. computation of a shortest arc between two points of the boundary (or between two distinct boundaries): $O(n \log n)$ [Dijkstra]
3. if $b \geq 1$, computation of a shortest non-separating arc: $O(n \log n)$ [Erickson, Har-Peled, 2002]
4. if $g = 0$ and $b \in \{2, 3\}$, computation of a shortest cycle homotopic to a given boundary: $O(n \log n)$ [CdV, Lazarus, 2003], [Frederickson, 1987]

Complexity [CdV, Erickson, 2006]

Preprocessing step: Computing the octagonal decomposition

$O(gn \log n)$ (improvement using [Cabello et al., 2008]).

Details: cross-metric surface

- Graph G cellularly embedded on $\mathscr{S}$
- Path p in G with k edges
- Octagonal decomposition in the cross-metric surface (in general position w.r.t. G^*)
- Admitted: Each cycle of the octagonal decomposition enters $O(1)$ times every face of G^*.

Tightening algorithm

$O(gnk)$, where k is the complexity of the input path, because

- p crosses the octagonal decomposition $O(gk)$ times,
- each octagon has complexity $O(n)$.

Minimum cut algorithm

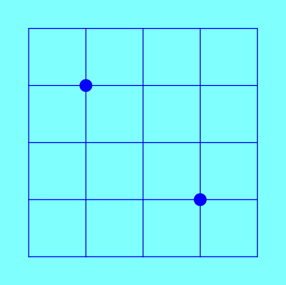
Problem

The minimum cut problem

Given

- $G = (V, E)$: a weighted, undirected graph;
- s, t: two vertices of G,

compute $W \subset V$ containing s but not t that minimizes the sum of the weights of the edges between W and $V \setminus W$.



Theorem [Chambers, Erickson, Nayyeri, 2009]

If G is embedded on a surface of genus g, this problem can be solved in $O(g^{O(g)} n \log n)$ time.

Best result known before

Algorithms for sparse graphs in $O(n^2 \log n)$ [Sleator, Tarjan, 1983] and $O(n^{3/2} \log n \log C)$ [Goldberg, Rao, 1998].

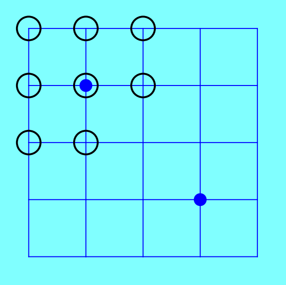
Problem

The minimum cut problem

Given

- $G = (V, E)$: a weighted, undirected graph;
- s, t: two vertices of G,

compute $W \subset V$ containing s but not t that minimizes the sum of the weights of the edges between W and $V \setminus W$.



Theorem [Chambers, Erickson, Nayyeri, 2009]

If G is embedded on a surface of genus g, this problem can be solved in $O(g^{O(g)} n \log n)$ time.

Best result known before

Algorithms for sparse graphs in $O(n^2 \log n)$ [Sleator, Tarjan, 1983] and $O(n^{3/2} \log n \log C)$ [Goldberg, Rao, 1998].

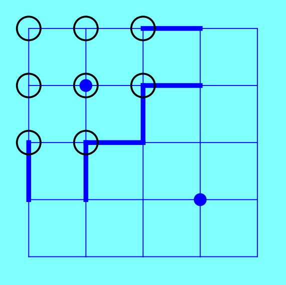
Problem

The minimum cut problem

Given

- $G = (V, E)$: a weighted, undirected graph;
- s, t: two vertices of G,

compute $W \subset V$ containing s but not t that minimizes the sum of the weights of the edges between W and $V \setminus W$.



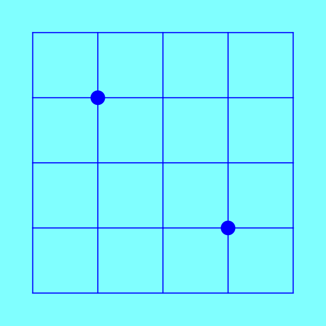
Theorem [Chambers, Erickson, Nayyeri, 2009]

If G is embedded on a surface of genus g, this problem can be solved in $O(g^{O(g)} n \log n)$ time.

Best result known before

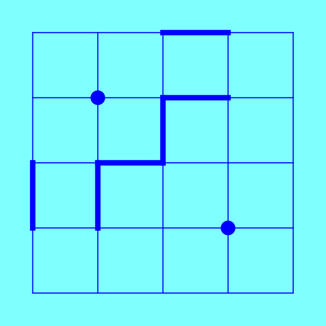
Algorithms for sparse graphs in $O(n^2 \log n)$ [Sleator, Tarjan, 1983] and $O(n^{3/2} \log n \log C)$ [Goldberg, Rao, 1998].

Preliminaries: case of the plane (=of the sphere)



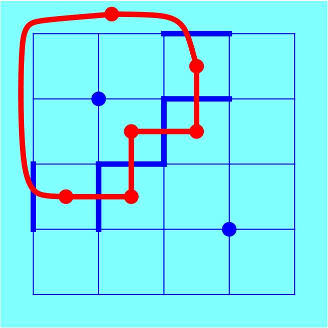
- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

Preliminaries: case of the plane (=of the sphere)



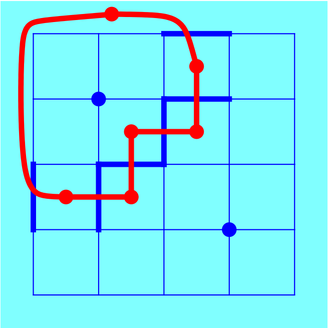
- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

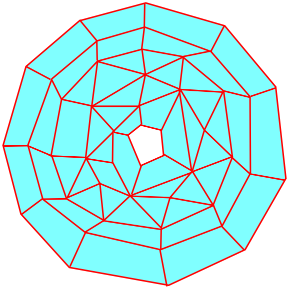
Preliminaries: case of the plane (=of the sphere)



- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

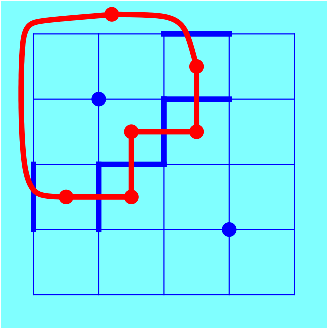
Preliminaries: case of the plane (=of the sphere)

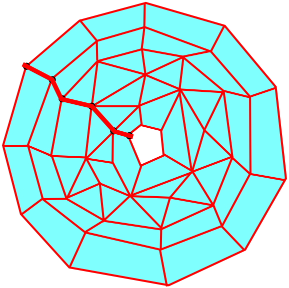




- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

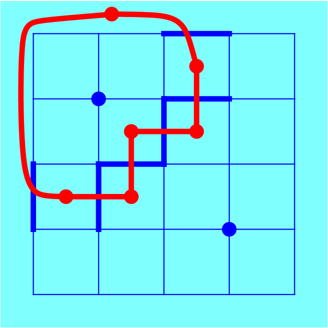
Preliminaries: case of the plane (=of the sphere)

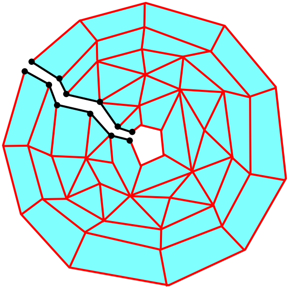




- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

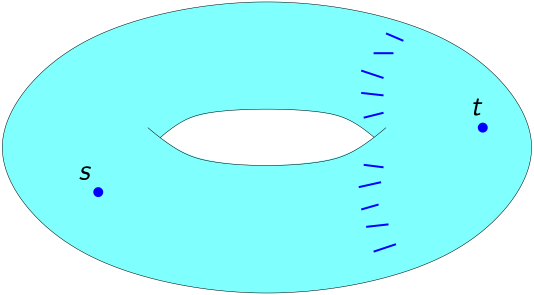
Preliminaries: case of the plane (=of the sphere)





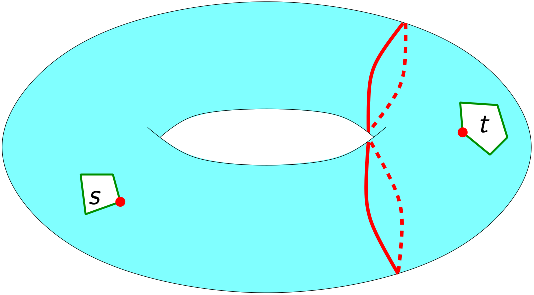
- Minimum cut in $G \leftrightarrow$ cycle γ separating s from t, of minimum length in G^*,
- or equivalently in the cross-metric surface defined by G.
- γ can be computed in $O(n \log n)$ time [Reif 1983; Frederickson 1987; Henzinger et al. 1997].

Case of surfaces



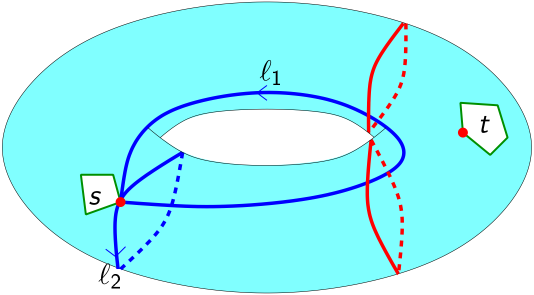
- Cut in $G \to$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

Case of surfaces



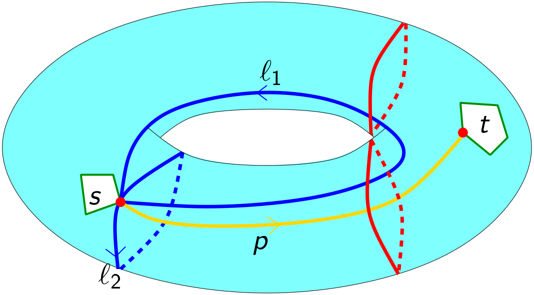
- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

Case of surfaces



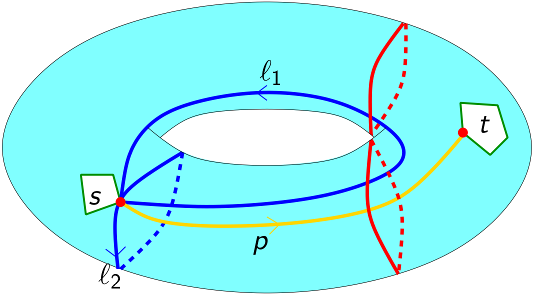
- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

Case of surfaces



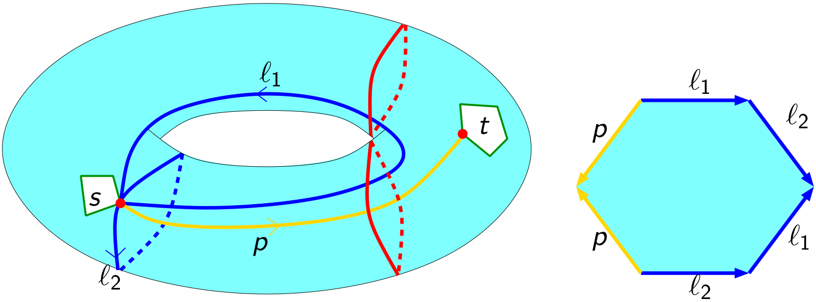
- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

Case of surfaces



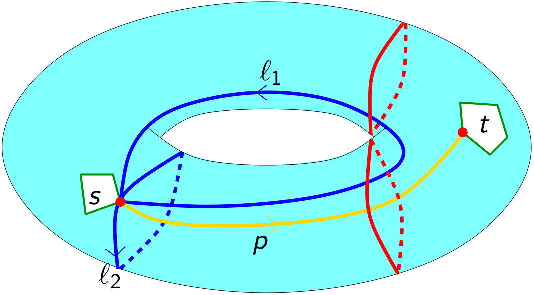
- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

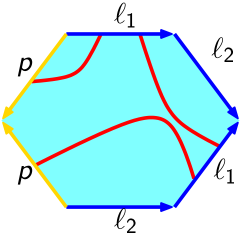
Case of surfaces



- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

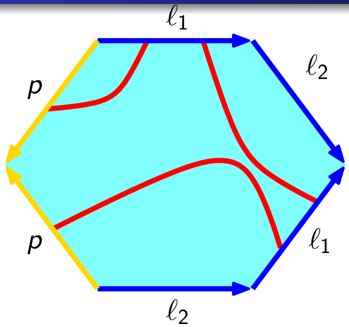
Case of surfaces





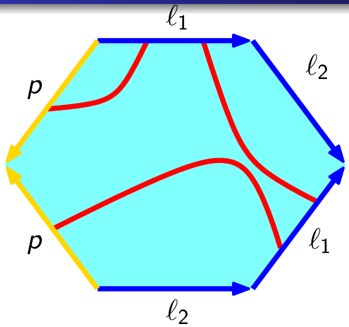
- Cut in G $\rightarrow$ family Γ of disjoint cycles in the cross-metric surface defined by G.
- Compute a cut graph based at s, obtaining loops $\ell_1, \ldots, \ell_{2g}$.
- Let p be a shortest path from s to t.
- Γ is a cut iff it crosses every ℓ_i an even number of times and p an odd number of times.

Algorithm



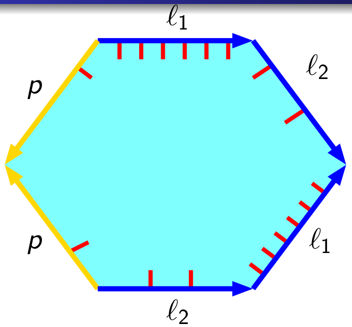
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm



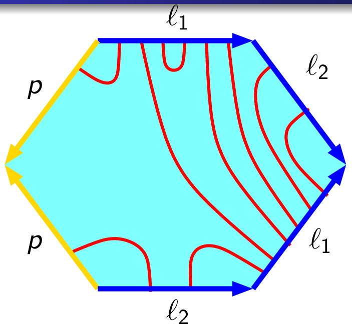
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm



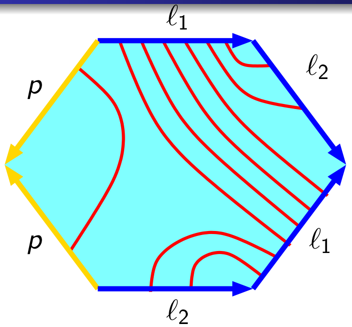
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm



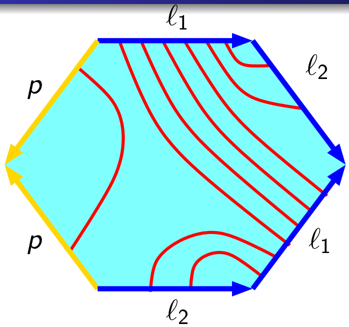
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm



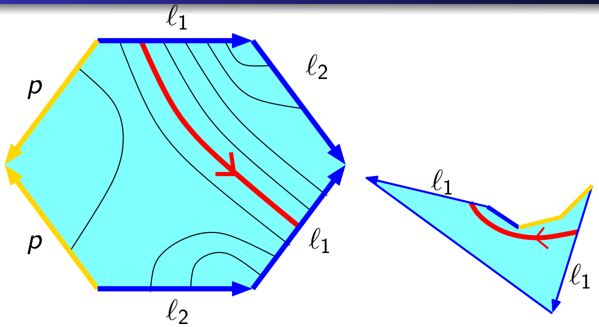
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm

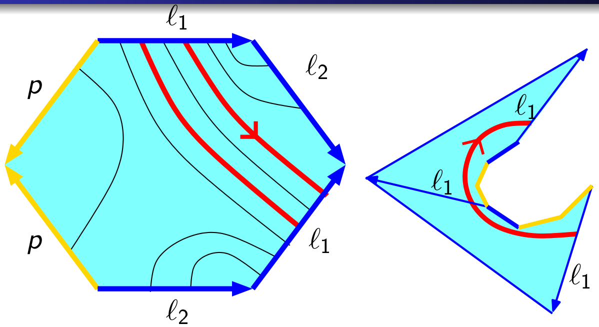


- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm

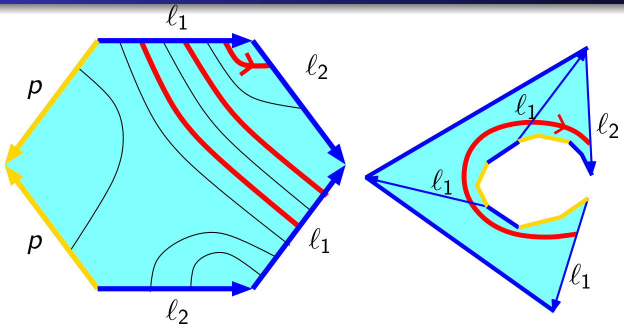


- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.



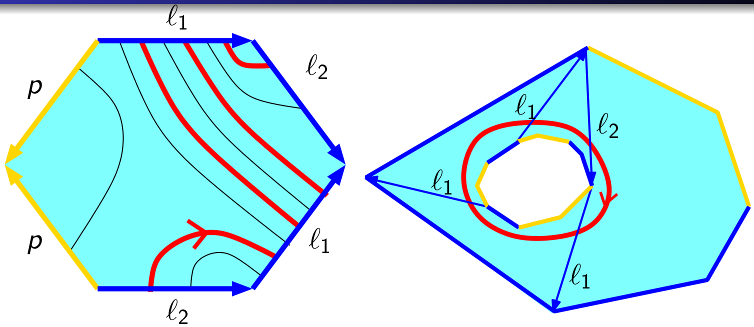
- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm



- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Algorithm

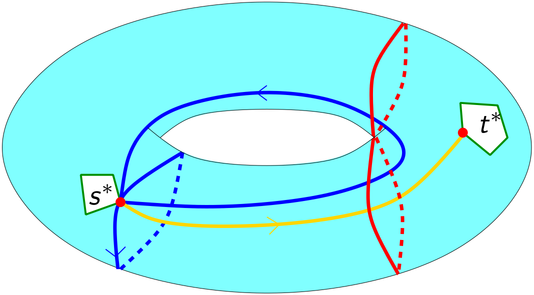


- *Lemma*. If Γ corresponds to a minimum cut, then it crosses $O(g)$ times p and each ℓ_i.
- Enumerate all possible patterns that form disjoint cycles. There are $g^{O(g)}$ possibilities.
- For each pattern, and each cycle appearing in this pattern, compute the shortest cycle that crosses p and the ℓ_i in the same order. This boils down to the planar case! Complexity $O(g^2 n \log n)$.
- Total running time: $O(g^{O(g)} n \log n)$.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

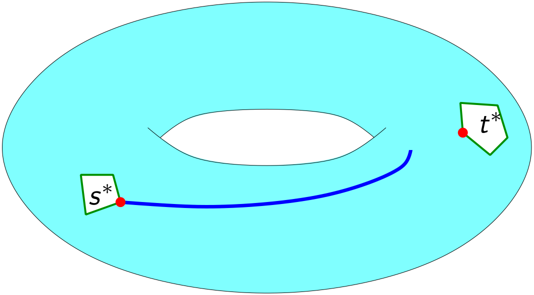


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

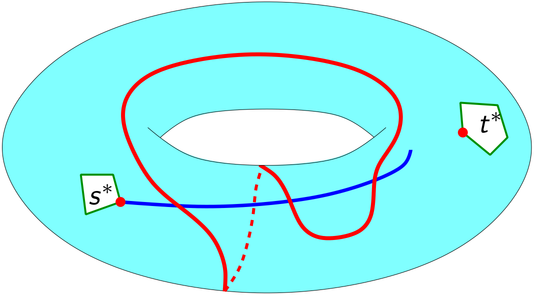


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

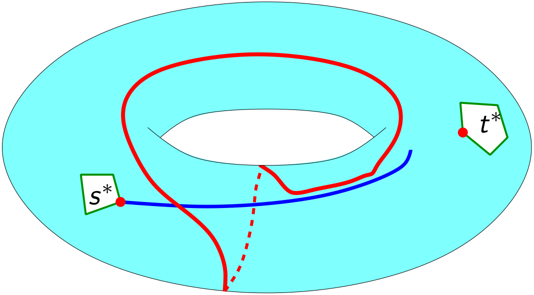


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

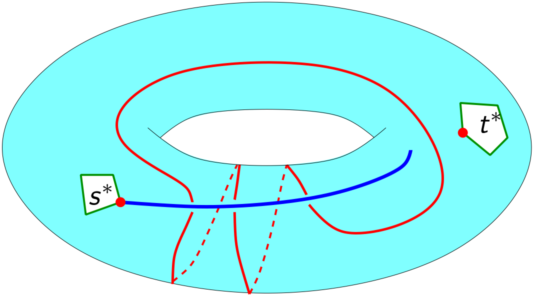


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

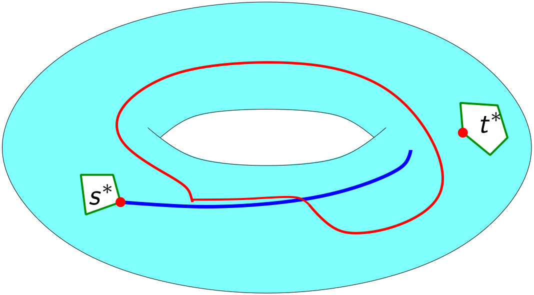


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

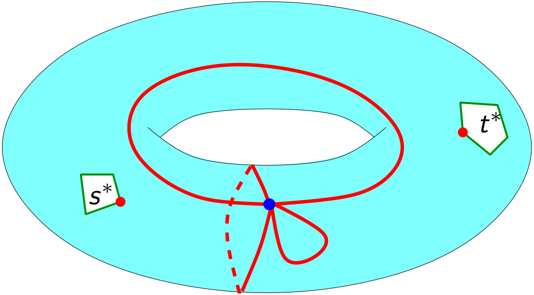


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

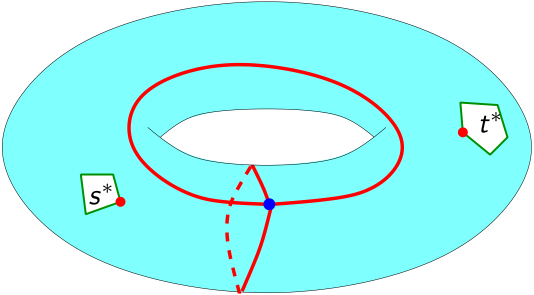


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

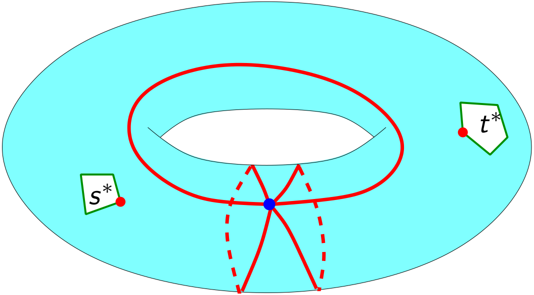


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.

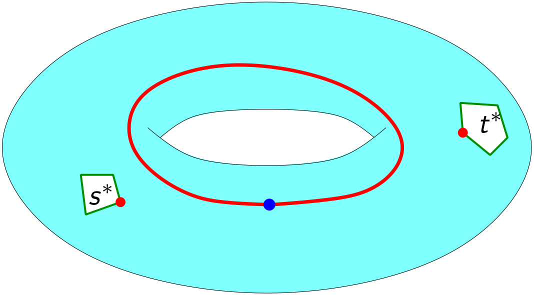


Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Proof of the lemma

Lemma

Γ has $O(g)$ crossings with p and with each of the ℓ_i.



Let us shrink the shortest path to a single vertex v:
Γ becomes a family of non-crossing loops based at v, such that no face of this graph is a disk bounded by one or two loops.

Thanks!

Table of contents