Approximating the Crossing Number of Graphs Embeddable in Any Orientable Surface

Petr Hliněný

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Rep.

http://www.fi.muni.cz/~hlineny

joint work with **Markus Chimani**

Faculty of Computer Science, TU Dortmund, Germany

1 History of Crossing Number

A WW II story for start

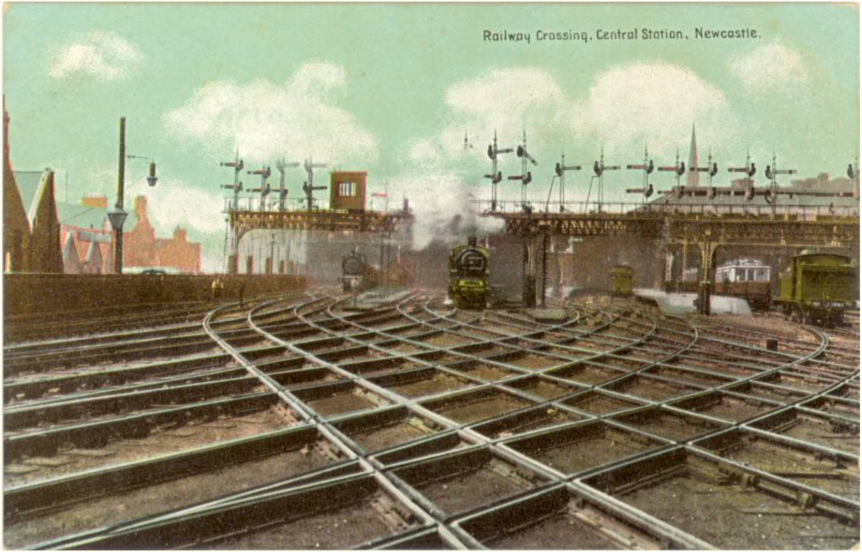
"*There were some kilns where the bricks were made and some open storage yards where the bricks were stored. All the kilns were connected by rail with all the storage yards. The bricks were carried on small wheeled trucks to the storage yards…the work was not difficult; the trouble was only at the crossings. The trucks generally jumped the rails there, and the bricks fell out of them; in short this caused a lot of trouble and loss of time…the idea occurred to me that this loss of time could have been minimized if the number of crossings of the rails had been minimized.*

But what is the minimum number of crossings?

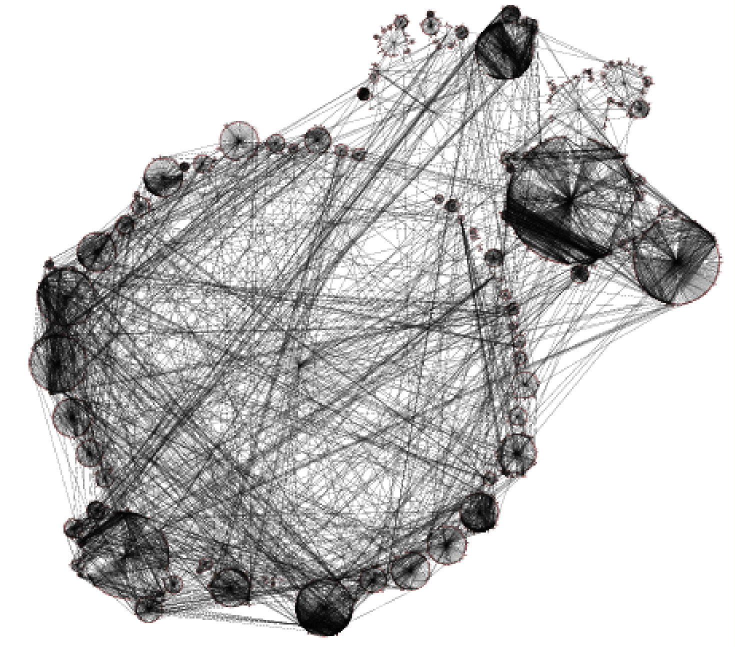
…This problem has become a notoriously difficult unsolved problem."

Pál Turán, *A note of welcome.*
Journal of Graph Theory (1977)

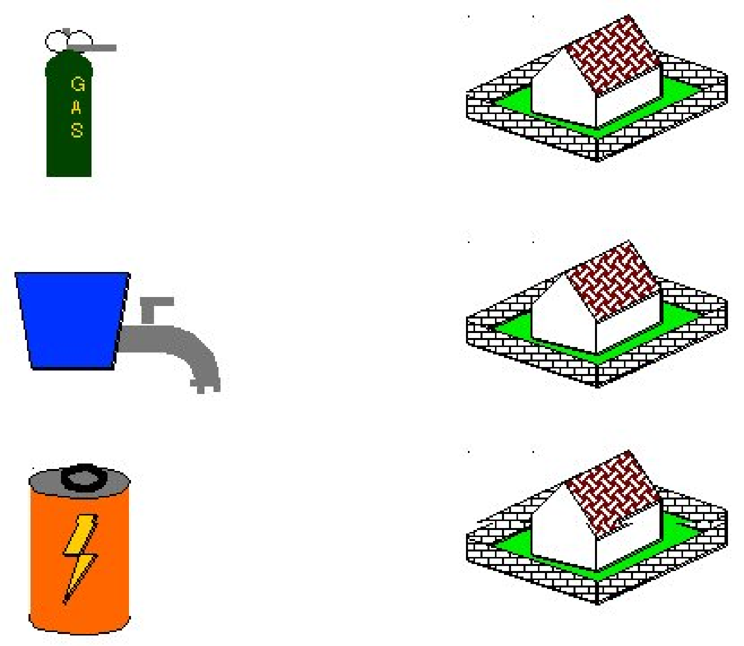
Crossings. . .



and even more crossings.



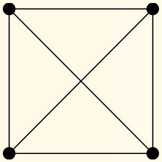
Can you avoid all the crossings?

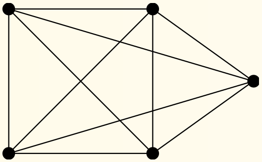


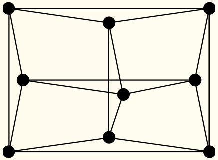
The definition

Definition. *Drawing of a graph* G:

- The vertices of G are distinct points,
  and every edge $e = uv \in E(G)$ is a simple curve joining u to v.
- No edge passes through another vertex,
  and no three edges intersect in a common point.



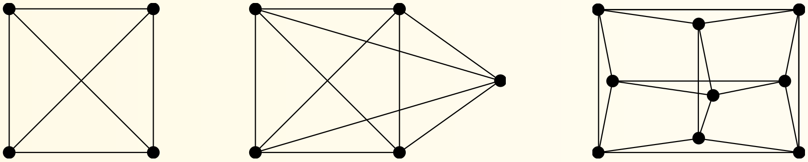




The definition

Definition. *Drawing of a graph* G:

- The vertices of G are distinct points,
  and every edge $e = uv \in E(G)$ is a simple curve joining u to v.
- No edge passes through another vertex,
  and no three edges intersect in a common point.



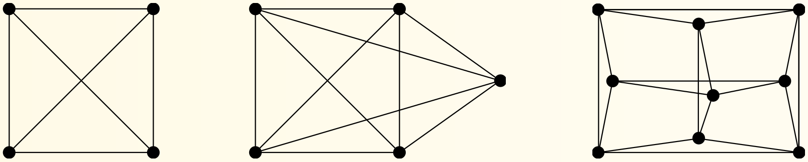
Definition. **Crossing number** *cr*(G)
is the smallest number of edge crossings in a drawing of G.

The definition

Definition. *Drawing of a graph* G:

– The vertices of G are distinct points,
and every edge $e = uv \in E(G)$ is a simple curve joining u to v.

– No edge passes through another vertex,
and no three edges intersect in a common point.



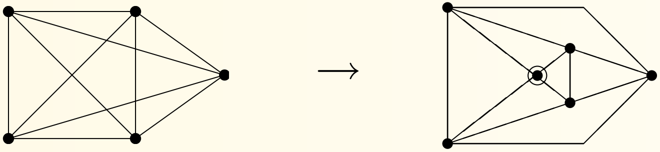
Definition. **Crossing number** *cr*(G)
is the smallest number of edge crossings in a drawing of G.

Warning. There are slight variations of the definition of crossing number, some giving different numbers! (Like counting *odd-crossing pairs* of edges. [Pelsmajer, Schaeffer, Štefankovič, 2005]. . .)

2 How to Compute the Crossing Number

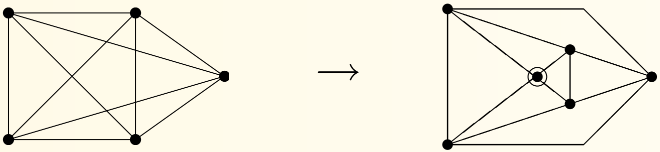
Observation. The problem $\textsc{CrossingNumber}(\leq k)$ is in NP:

Guess a suit. drawing of G, then replace crossings with new vertices, and test planarity.



2 How to Compute the Crossing Number

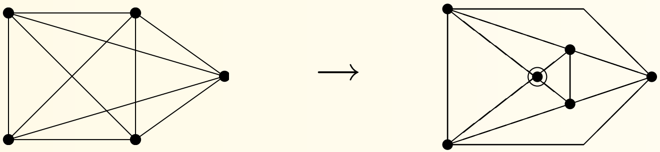
Observation. The problem $\textsc{CrossingNumber}(\leq k)$ is in NP:
Guess a suit. drawing of G, then replace crossings with new vertices, and test planarity.



Theorem 1. [**Grohe**, 2001] $\textsc{CrossingNumber}(\leq k)$ *is in FPT with parameter k, i.e. solvable in time $O(f(k) \cdot n^2)$.*
[**Kawarabayashi and Reed**, 2007] $\ldots$ *in time $O(f'(k) \cdot n)$.*

2 How to Compute the Crossing Number

Observation. The problem CrossingNumber($\leq k$) is in NP:
Guess a suit. drawing of G, then replace crossings with new vertices, and test planarity.



Theorem 1. [**Grohe**, 2001] CrossingNumber($\leq k$) *is in FPT with parameter k, i.e. solvable in time $O(f(k) \cdot n^2)$.*
[**Kawarabayashi and Reed**, 2007] ... *in time $O(f'(k) \cdot n)$.*

Practical algorithm. [**Chimani, Mutzel, and Bomze**, 2008]
A branch & bound approach that can compute exactly the crossing numbers of "real-world" graphs on up to ~ 100 vertices.

But, what else?

Bad news

Theorem 2. [**Garey and Johnson**, 1983] CrossingNumber *is* NP*-hard.*

Bad news

Theorem 2. [**Garey and Johnson**, 1983] CrossingNumber *is* NP*-hard.*

Theorem 3. [**PH**, 2004] CrossingNumber *is* NP*-complete even on simple* 3*-connected cubic graphs, and hence also in the minor–monotone variant.*

and things are getting much worse now. . .

Bad news

Theorem 2. [**Garey and Johnson**, 1983] CrossingNumber *is NP-hard.*

Theorem 3. [**PH**, 2004] CrossingNumber *is NP-complete even on simple 3-connected cubic graphs, and hence also in the minor–monotone variant.*

and things are getting much worse now. . .

Theorem 4. [**Cabello and Mohar**, 2010]
CrossingNumber *is NP-complete even on almost-planar (near-planar) graphs, i.e. graphs that result from a planar graph by adding one edge!*

(The maximum degree is unbounded in this case, though.)

Bad news

Theorem 2. [**Garey and Johnson**, 1983] CROSSINGNUMBER *is* NP*-hard.*

Theorem 3. [**PH**, 2004] CROSSINGNUMBER *is* NP*-complete even on simple* 3*-connected cubic graphs, and hence also in the minor–monotone variant.*

and things are getting much worse now...

Theorem 4. [**Cabello and Mohar**, 2010]
CROSSINGNUMBER *is* NP*-complete even on almost-planar (near-planar) graphs, i.e. graphs that result from a planar graph by adding one edge!*

(The maximum degree is unbounded in this case, though.)

So, what can be computed efficiently?

- Perhaps, we could compute the crossing number of bounded-degree almost-planar graphs? (True for cubic almost-pl. by [Riskin, 1996].)

Bad news

Theorem 2. [**Garey and Johnson**, 1983] CrossingNumber *is* NP*-hard.*

Theorem 3. [**PH**, 2004] CrossingNumber *is* NP*-complete even on simple* 3*-connected cubic graphs, and hence also in the minor–monotone variant.*

and things are getting much worse now...

Theorem 4. [**Cabello and Mohar**, 2010]
CrossingNumber *is* NP*-complete even on almost-planar (near-planar) graphs, i.e. graphs that result from a planar graph by adding one edge!*

(The maximum degree is unbounded in this case, though.)

So, what can be computed efficiently?

- Perhaps, we could compute the crossing number of bounded-degree almost-planar graphs? (True for cubic almost-pl. by [Riskin, 1996].)
- Or, we may resort to approximations...

Approximating the cossing number

Theorem 5. [**Even, Guha and Schieber**, 2002]
CrossingNumber *can be approximated in polynomial time:* $cr(G)+|V(G)|$ *up to a factor of* $\log^3|V(G)|$ *for graphs* G *of bounded degree.*

This result relates to VLSI design problems. . .

Then a series of constant-factor approximations (in case of bounded degrees):

Theorem 6. [**PH and Salazar**, 2006] CROSSINGNUMBER *can be approximated in linear time up to a factor of* $\Delta(G)$ *for almost-planar graphs* G*.*

[**Cabello and Mohar**, 2008] ... *factor of* $\lfloor \Delta(G)/2 \rfloor$*.*

Then a series of constant-factor approximations (in case of bounded degrees):

Theorem 6. [**PH and Salazar**, 2006] CrossingNumber *can be approximated in linear time up to a factor of* $\Delta(G)$ *for almost-planar graphs* G*.*

[**Cabello and Mohar**, 2008] ... *factor of* $\lfloor \Delta(G)/2 \rfloor$*.*

Theorem 7. [**Gitler, PH, Leaños and Salazar**, 2007]
CrossingNumber *can be approximated in polynomial time up to a factor of* $\frac{9}{2}\Delta(G)^2$ *for projective graphs* G*.*

Theorem 8. [**PH and Salazar**, 2007] CrossingNumber *can be approximated in polynomial time up to a factor of* $6\Delta(G)^2$ *for toroidal graphs* G*.*

(The latter two results assume "sufficiently dense" embeddability of G in the specified surface, and use a subroutine for computing the *edge-width*.)

Then a series of constant-factor approximations (in case of bounded degrees):

Theorem 6. [**PH and Salazar**, 2006] CrossingNumber *can be approximated in linear time up to a factor of* $\Delta(G)$ *for almost-planar graphs* G*.*

[**Cabello and Mohar**, 2008] ... *factor of* $\lfloor \Delta(G)/2 \rfloor$*.*

Theorem 7. [**Gitler, PH, Leaños and Salazar**, 2007]
CrossingNumber *can be approximated in polynomial time up to a factor of* $\frac{9}{2}\Delta(G)^2$ *for projective graphs* G*.*

Theorem 8. [**PH and Salazar**, 2007] CrossingNumber *can be approximated in polynomial time up to a factor of* $6\Delta(G)^2$ *for toroidal graphs* G*.*

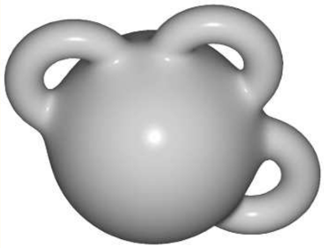
(The latter two results assume "sufficiently dense" embeddability of G in the specified surface, and use a subroutine for computing the *edge-width*.)

Theorem 9. [**Chimani, PH and Mutzel**, 2008]
CrossingNumber *can be approximated in polynomial time up to a factor of* $d(x) \cdot \lfloor \Delta(G)/2 \rfloor$ *for apex graphs* G *(*x *is the apex vertex).*

3 New Result(s)

Definition. An *orientable surface of genus g* results from a sphere by adding g "handles".

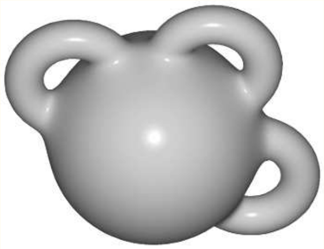
Sphere, torus, double-torus, triple-torus (in the picture), . . .



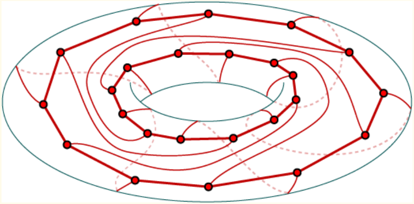
3 New Result(s)

Definition. An *orientable surface of genus g* results from a sphere by adding g "handles".

Sphere, torus, double-torus, triple-torus (in the picture), ...



Definition. An *embedding* of a graph in a surface is a drawing without crossings.



Main result

Informally: Graphs of bounded degrees and "densely" embeddable in any fixed orientable surface have polynomial constant-factor approximation algorithm for CrossingNumber.

Main result

Informally: Graphs of bounded degrees and "densely" embeddable in any fixed orientable surface have polynomial constant-factor approximation algorithm for CrossingNumber.

Definition. *Dual graph* of an embedded graph G – "faces $\rightarrow$ vertices" G^*.

Definition. *Edge-width* of an embedded graph G – the shortest length of a noncontractible cycle in G. (Embedding "density" measure. . .)

Main result

Informally: Graphs of bounded degrees and "densely" embeddable in any fixed orientable surface have polynomial constant-factor approximation algorithm for CrossingNumber.

Definition. *Dual graph* of an embedded graph G – "faces $\rightarrow$ vertices" G^*.

Definition. *Edge-width* of an embedded graph G – the shortest length of a noncontractible cycle in G. (Embedding "density" measure...)

Theorem 10. *Let G be a multigraph embeddable in an orientable surface of genus $g \geq 1$ with nonseparating dual edge-width at least $2^{g+2}\Delta(G)$.*

The next Algorithm 11 computes a drawing of G in the plane with at most $3 \cdot 2^{3g+2} \cdot \Delta(G)^2 \cdot cr(G)$ crossings. Its running time is $O(n \log n)$.

Main result

Informally: Graphs of bounded degrees and “densely” embeddable in any fixed orientable surface have polynomial constant-factor approximation algorithm for CrossingNumber.

Definition. *Dual graph* of an embedded graph G – “faces $\rightarrow$ vertices” G^*.

Definition. *Edge-width* of an embedded graph G – the shortest length of a noncontractible cycle in G. (Embedding “density” measure. . .)

Theorem 10. *Let G be a multigraph embeddable in an orientable surface of genus $g \geq 1$ with nonseparating dual edge-width at least $2^{g+2}\Delta(G)$.*

The next Algorithm 11 computes a drawing of G in the plane with at most $3 \cdot 2^{3g+2} \cdot \Delta(G)^2 \cdot cr(G)$ crossings. Its running time is $O(n \log n)$.

Hence this is a constant factor approximation algorithm for CrossingNumber *$cr(G)$ in the case of bounded degrees by Δ and bounded genus g.*

This widely extends our previous Theorems 7 and 8.

Related mathematical aspects

Some deep new math considerations are needed to prove the lower bound on $cr(G)$, i.e. to relate unknown $cr(G)$ to the number of crossings produced by our algorithm...

- Deep considerations of "embedding density" of graphs in surfaces, and new density estimates related to "surface cutting".

Related mathematical aspects

Some deep new math considerations are needed to prove the lower bound on $cr(G)$, i.e. to relate unknown $cr(G)$ to the number of crossings produced by our algorithm...

- Deep considerations of "embedding density" of graphs in surfaces, and new density estimates related to "surface cutting".
- New useful "embedding density" measure defined – the *stretch of* G.

Related mathematical aspects

Some deep new math considerations are needed to prove the lower bound on $cr(G)$, i.e. to relate unknown $cr(G)$ to the number of crossings produced by our algorithm...

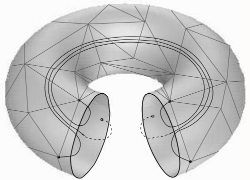
- Deep considerations of "embedding density" of graphs in surfaces, and new density estimates related to "surface cutting".
- New useful "embedding density" measure defined – the *stretch of G*.
- A new technical concept of *bipolarity* of a subembedding appears very helpful in the proofs.

4 Sketch of the Proof

4 Sketch of the Proof

The easy side – Algorithmic upper bound

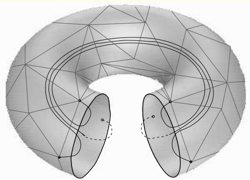
- Basic idea: iteratively "*cut and open*" a handle, and redraw the affected edges through the rest of the graph.



4 Sketch of the Proof

The easy side – Algorithmic upper bound

- Basic idea: iteratively "*cut and open*" a handle, and redraw the affected edges through the rest of the graph.



- Similar to prev. upper bounds on the crossing num. of surface-embedded graphs, e.g. [Böröczky, Pach, Tóth, 2006] and [Djidjev and Vrt'o, 2006].

  Yet, our upper bound is stronger and thus allows for an approximat. alg.

Algorithm 11. DRAWING A SURFACE-EMBEDDABLE GRAPH IN THE PLANE

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

Algorithm 11. Drawing a surface-embeddable graph in the plane

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

II) For $i = 1, 2, \ldots, g$; we use [Kutz, 2006] to compute, in the dual graph G_i^*, a nonseparating dual cycle γ_i of length $c_i = ew^*(G_i)$.

Algorithm 11. Drawing a surface-embeddable graph in the plane

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

II) For $i = 1, 2, \ldots, g$; we use [Kutz, 2006] to compute, in the dual graph G_i^*, a nonseparating dual cycle γ_i of length $c_i =$ *ew*$^*(G_i)$.

We construct an embedding $\boldsymbol{G_{i+1} = G_i/\gamma_i}$ by cutting G_i along γ_i.
(G_{i+1} is a spanning subgraph of G_i, and G_{i+1} has genus $g - i$.)

Algorithm 11. DRAWING A SURFACE-EMBEDDABLE GRAPH IN THE PLANE

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

II) For $i = 1, 2, \ldots, g$; we use [Kutz, 2006] to compute, in the dual graph G_i^*, a nonseparating dual cycle γ_i of length $c_i = ew^*(G_i)$.
We construct an embedding $\boldsymbol{G_{i+1} = G_i/\gamma_i}$ by cutting G_i along γ_i.
(G_{i+1} is a spanning subgraph of G_i, and G_{i+1} has genus $g - i$.)

III) Now, G_{g+1} is a planar embedding (spanning G!).
For any "missing" edge $\boldsymbol{e = v_1 v_2} \in F = E(G) \setminus E(G_{g+1})$ we compute, using breadth-first search, a shortest dual path $\pi(v_1, v_2)$ between the "cut-face" incident to v_1 and the "cut-face" incident to v_2 in G_{g+1}^*.

Algorithm 11. DRAWING A SURFACE-EMBEDDABLE GRAPH IN THE PLANE

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

II) For $i = 1, 2, \ldots, g$; we use [Kutz, 2006] to compute, in the dual graph G_i^*, a nonseparating dual cycle γ_i of length $c_i = ew^*(G_i)$.
We construct an embedding $\boldsymbol{G_{i+1} = G_i/\gamma_i}$ by cutting G_i along γ_i.
(G_{i+1} is a spanning subgraph of G_i, and G_{i+1} has genus $g - i$.)

III) Now, G_{g+1} is a planar embedding (spanning G!).
For any "missing" edge $\boldsymbol{e = v_1 v_2} \in F = E(G) \setminus E(G_{g+1})$ we compute, using breadth-first search, a shortest dual path $\pi(v_1, v_2)$ between the "cut-face" incident to v_1 and the "cut-face" incident to v_2 in G_{g+1}^*.
This can be done such that no two distinct paths $\pi(v_1, v_2)$, $\pi(v_1', v_2')$ intersect more than once.

Algorithm 11. Drawing a surface-embeddable graph in the plane

Given is a nonpl. graph G embeddable in the orientable surface $\mathcal{S}_g$ of genus g.

I) We construct an *embedding* G_1 of G in $\mathcal{S}_g$ using [Mohar, 1999].

II) For $i = 1, 2, \ldots, g$; we use [Kutz, 2006] to compute, in the dual graph G_i^*, a nonseparating dual cycle γ_i of length $c_i = ew^*(G_i)$.
We construct an embedding $\boldsymbol{G_{i+1} = G_i/\gamma_i}$ by cutting G_i along γ_i.
(G_{i+1} is a spanning subgraph of G_i, and G_{i+1} has genus $g - i$.)

III) Now, G_{g+1} is a planar embedding (spanning G!).
For any "missing" edge $\boldsymbol{e = v_1 v_2} \in F = E(G) \setminus E(G_{g+1})$ we compute, using breadth-first search, a shortest dual path $\pi(v_1, v_2)$ between the "cut-face" incident to v_1 and the "cut-face" incident to v_2 in G_{g+1}^*.
This can be done such that no two distinct paths $\pi(v_1, v_2)$, $\pi(v_1', v_2')$ intersect more than once.

IV) Within G_{g+1}, we draw every edge $e = v_1 v_2 \in F$ "along" the dual path $\pi = \pi(v_1, v_2)$, crossing the *len*(π) edges of G_{g+1} that are dual to $E(\pi)$.
We output the resulting drawing $\tilde{G}$ isomorphic to input G.

The difficult side – Proving a lower bound

Recall; "Algorithm 11 computes $R \leq 3 \cdot 2^{3g+2} \cdot \Delta(G)^2 \cdot \mathit{cr}(G)$ crossings". Since we have so far no idea what $\mathit{cr}(G)$ should be, we have to lower-estimate $\mathit{cr}(G)$ based on the run and the results of Algorithm 11.

The difficult side – Proving a lower bound

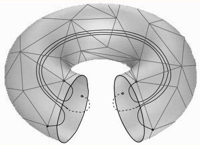
Recall; "Algorithm 11 computes $R \leq 3 \cdot 2^{3g+2} \cdot \Delta(G)^2 \cdot cr(G)$ crossings". Since we have so far no idea what $cr(G)$ should be, we have to lower-estimate $cr(G)$ based on the run and the results of Algorithm 11.

- Easily,

$$R \leq 3 \cdot (2^{g+1} - 2 - g) \cdot \max\{len(\gamma_i) \cdot \ell_i : i = 1, 2, \ldots, g\}$$

  where γ_i is the dual "cut-cycle" at step i,

  and ℓ_i is the dual distance of the two "cut-faces" in G_{i+1}.



The difficult side – Proving a lower bound

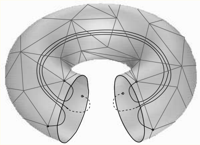
Recall; "Algorithm 11 computes $R \leq 3 \cdot 2^{3g+2} \cdot \Delta(G)^2 \cdot cr(G)$ crossings". Since we have so far no idea what $cr(G)$ should be, we have to lower-estimate $cr(G)$ based on the run and the results of Algorithm 11.

- Easily,

$$R \leq 3 \cdot (2^{g+1} - 2 - g) \cdot \max\{len(\gamma_i) \cdot \ell_i : i = 1, 2, \ldots, g\}$$

  where γ_i is the dual "cut-cycle" at step i,

  and ℓ_i is the dual distance of the two "cut-faces" in G_{i+1}.



- The difficult part is now to prove the lower bound

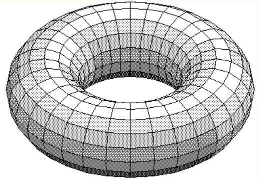
$$2^{-2g-1} \cdot \Delta(G)^{-2} \cdot \max\{len(\gamma_i) \cdot \ell_i : i = 1, 2, \ldots, g\} \leq cr(G)\,. \qquad (1)$$

5 "Mathematical" Lower Bound

For a rigorous presentation of the proof, the bound (1) is made independent of the algorithm:

Theorem 12. *Let G be a graph embedded in the orientable surface of genus $g \geq 1$ with nonseparating dual edge-width $c = ew^*(G) \geq 2^{g+2}\Delta(G)$, and let γ be any nonseparating dual cycle in G of length c. If the shortest γ-switching ear in G^* has length ℓ, then the crossing number of G satisfies*

$$cr(G) \geq 2^{-2g-1} \cdot \Delta(G)^{-2} \cdot c\ell \,. \qquad (2)$$

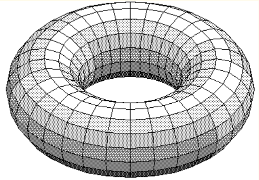


5 "Mathematical" Lower Bound

For a rigorous presentation of the proof, the bound (1) is made independent of the algorithm:

Theorem 12. *Let G be a graph embedded in the orientable surface of genus $g \geq 1$ with nonseparating dual edge-width $c = ew^*(G) \geq 2^{g+2}\Delta(G)$, and let γ be any nonseparating dual cycle in G of length c. If the shortest γ-switching ear in G^* has length ℓ, then the crossing number of G satisfies*

$$cr(G) \geq 2^{-2g-1} \cdot \Delta(G)^{-2} \cdot c\ell \,. \qquad (2)$$

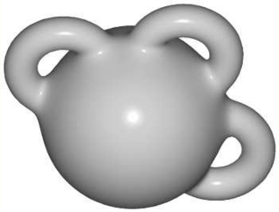


Base case. True for the torus, by [PH and Salazar, 2007] (cf. Theorem 8).

The core idea is to find an $\Omega(c) \times \Omega(\ell)$ *toroidal grid* as a minor in G...

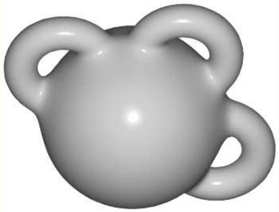
Induction on g: higher surfaces

- Our toroidal grid minor is "hiding somewhere" in G, and we want to find it. So we cut the handles of $\mathcal{S}_g$ down to a torus where we discover it.



Induction on g: higher surfaces

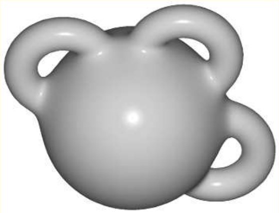
- Our toroidal grid minor is "hiding somewhere" in G, and we want to find it. So we cut the handles of $\mathcal{S}_g$ down to a torus where we discover it.



- Really?

Induction on g: higher surfaces

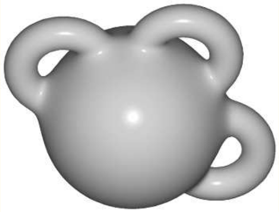
- Our toroidal grid minor is "hiding somewhere" in G, and we want to find it. So we cut the handles of $\mathcal{S}_g$ down to a torus where we discover it.



- Really? Although sounding easy, this is much complicated by the fact we must not cut through our desired toroidal grid!

Induction on g: higher surfaces

- Our toroidal grid minor is "hiding somewhere" in G, and we want to find it. So we cut the handles of $\mathcal{S}_g$ down to a torus where we discover it.

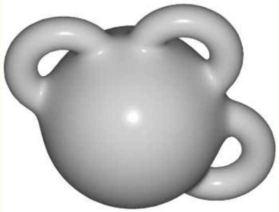


- Really? Although sounding easy, this is much complicated by the fact we must not cut through our desired toroidal grid!

- Here we use: *stretch*$(G) =$ min *len*$(\alpha)\cdot$*len*(β) over all "*one-leaping*" pairs of dual cycles in G.

Induction on g: higher surfaces

- Our toroidal grid minor is "hiding somewhere" in G, and we want to find it. So we cut the handles of $\mathcal{S}_g$ down to a torus where we discover it.



- Really? Although sounding easy, this is much complicated by the fact we must not cut through our desired toroidal grid!

- Here we use: *stretch*$(G) =$ min *len*$(\alpha)\cdot$*len*(β) over all "*one-leaping*" pairs of dual cycles in G.

  First phase – cut some handles to raise the stretch up to $\Omega(c\cdot\ell)$. (difficult!)

  Second phase – cut the rest down to a torus (which might destroy a particular toroidal grid, but cannot significantly lower the stretch).

6 Final Remarks

- **Approximation factor.** While the dependency on Δ is mild (and seems unavoidable for structural reasons – dual edge-width vs. face-width), what could be done to reduce the exponential dep. on genus g?

6 Final Remarks

- **Approximation factor.** While the dependency on Δ is mild (and seems unavoidable for structural reasons – dual edge-width vs. face-width), what could be done to reduce the exponential dep. on genus g?

  The exponential dep. on g pops up suddenly at many places and it is unavoidable on a local scale, but still, there might be a completely different approach reducing this to a $poly(g)$ factor...

6 Final Remarks

- **Approximation factor.** While the dependency on Δ is mild (and seems unavoidable for structural reasons – dual edge-width vs. face-width), what could be done to reduce the exponential dep. on genus g?

  The exponential dep. on g pops up suddenly at many places and it is unavoidable on a local scale, but still, there might be a completely different approach reducing this to a $poly(g)$ factor...

- **Nonorientable surfaces.** We believe the same approach will work, but there are many more complications.

6 Final Remarks

- **Approximation factor.** While the dependency on Δ is mild (and seems unavoidable for structural reasons – dual edge-width vs. face-width), what could be done to reduce the exponential dep. on genus g?

  The exponential dep. on g pops up suddenly at many places and it is unavoidable on a local scale, but still, there might be a completely different approach reducing this to a $poly(g)$ factor...

- **Nonorientable surfaces.** We believe the same approach will work, but there are many more complications.

  Particularly, a "cheapest" cut though an embedding can now have three forms: cutting a *handle*, an *antihandle*, or a *crosscap*.

6 Final Remarks

- **Approximation factor.** While the dependency on Δ is mild (and seems unavoidable for structural reasons – dual edge-width vs. face-width), what could be done to reduce the exponential dep. on genus g?

  The exponential dep. on g pops up suddenly at many places and it is unavoidable on a local scale, but still, there might be a completely different approach reducing this to a $poly(g)$ factor...

- **Nonorientable surfaces.** We believe the same approach will work, but there are many more complications.

  Particularly, a "cheapest" cut though an embedding can now have three forms: cutting a *handle*, an *antihandle*, or a *crosscap*.

- **Density requirement.** Our lower bound in Theorem 12 requires sufficient nonseparating dual edge-width to hold true, but the cases of non-densely embeddable graphs could, perhaps, be independently solved using "multiple-edge insertion" analogous to Theorem 9 (apex gr. approx).