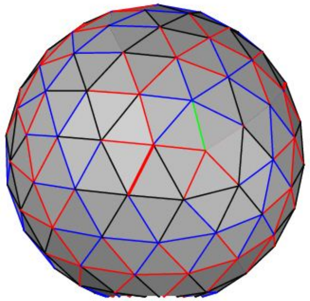
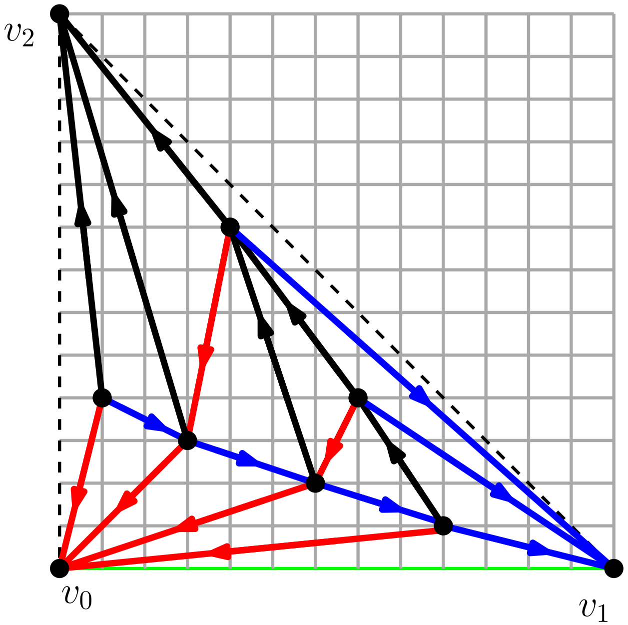
Schnyder woods for higher genus surfaces: from graph encoding to graph drawing





JCB 2014, Labri

Luca Castelli Aleardi

(joint works with O. Devillers, E. Fusy, A. Kostrygin, T. Lewiner)

ÉCOLE POLYTECHNIQUE

LIRMM Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

EGOS Embedded Graphs and their Oriented Structures ANR Project 2012-2015

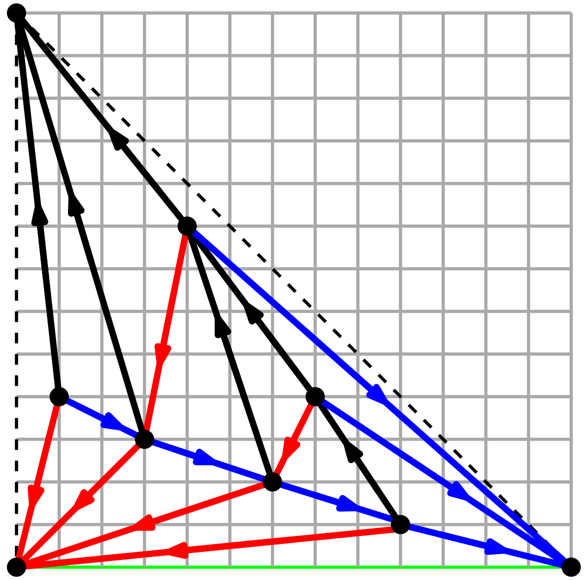
Some facts about planar graphs

("As I have known them")

Some facts about planar graphs

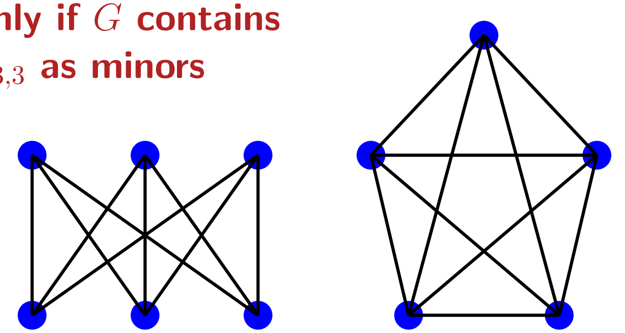
Thm (Schnyder, Trotter, Felsner)

G **planar if and only if** $dim(G) \leq 3$



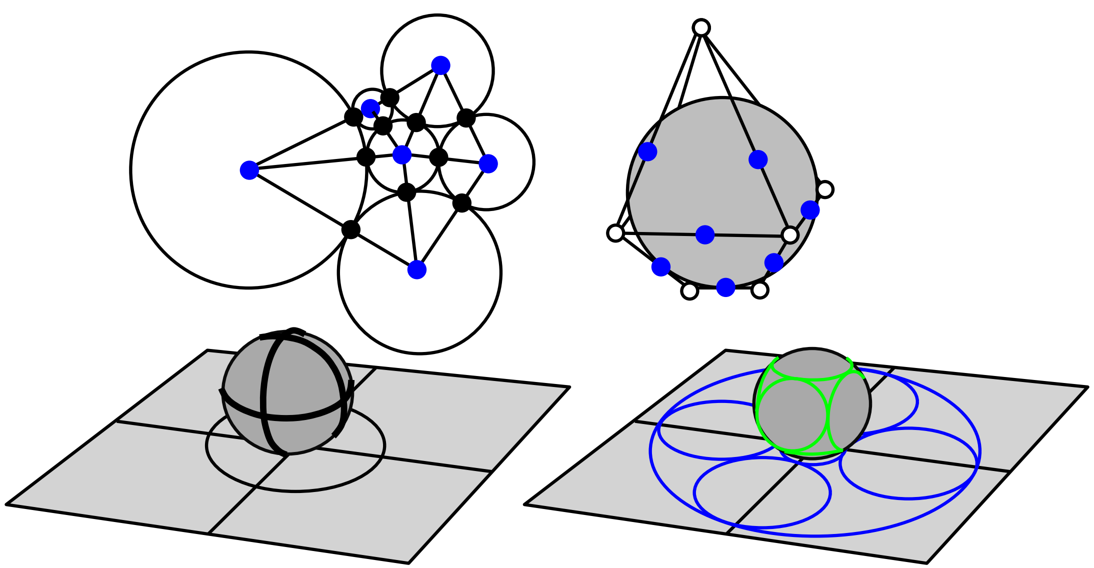
Thm (Kuratowski, excluded minors)

G **planar if and only if** G **contains neither** K_5 **nor** $K_{3,3}$ **as minors**



Thm (Koebe-Andreev-Thurston)

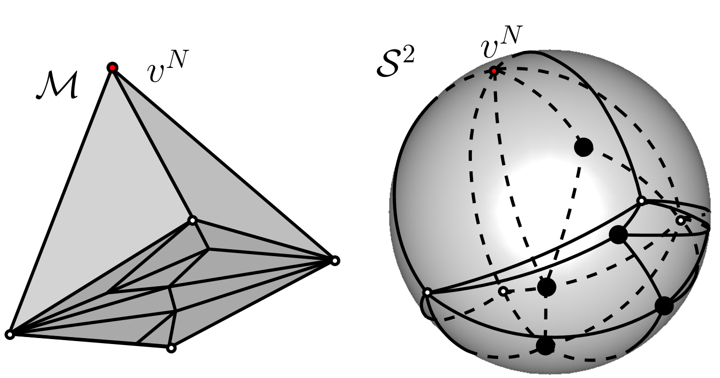
Every planar graph with n **vertices is isomorphic to the intersection graph of** n **disks in the plane.**



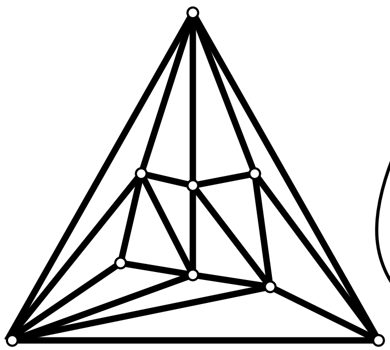
Thm (Y. Colin de Verdière)

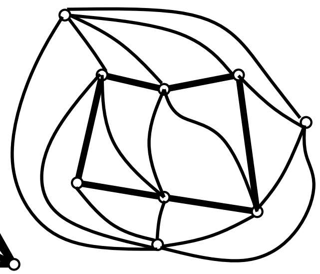
G **planar if and only if** $\mu(G) \leq 3$
($\mu(G) =$ **multiplicity of** λ_2 **of a generalized laplacian)**

$$L_G = \begin{bmatrix} 4 & -1 & \dots & \dots & 0 \\ -1 & 5 & \dots & & \\ \dots & & \dots & & \\ \dots & & & & \dots \\ 0 & \dots & & & 3 \end{bmatrix} \qquad L_G[i,k] = \begin{cases} deg(v_i) \\ -A_G[i,j] \end{cases}$$



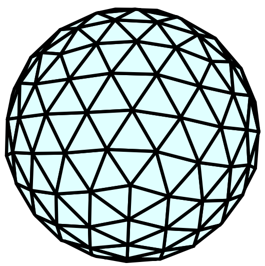
Planar triangulations

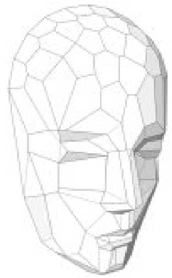




$$n - e + f = 2$$

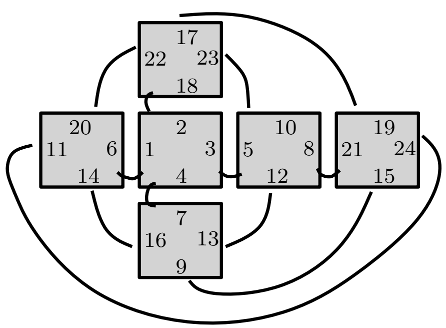
$$e = 3n - 6$$

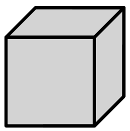


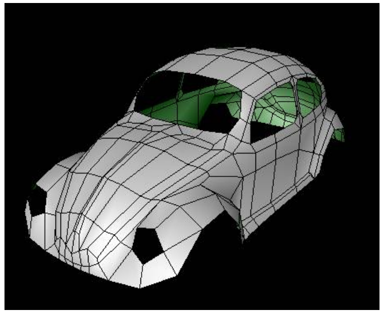


$\phi = (1,2,3,4)(17,23,18,22)(5,10,8,12)(21,19,24,15)\ldots$

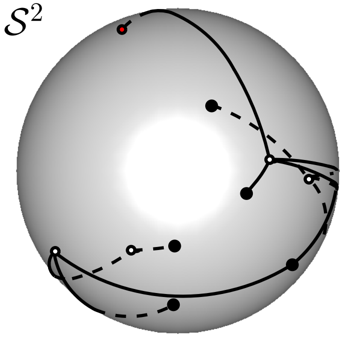
$\alpha = (2,18)(4,7)(12,13)(9,15)(14,16)(10,23)\ldots$







$\mathcal{S}^2$



Schnyder woods and canonical orderings: overview of applications

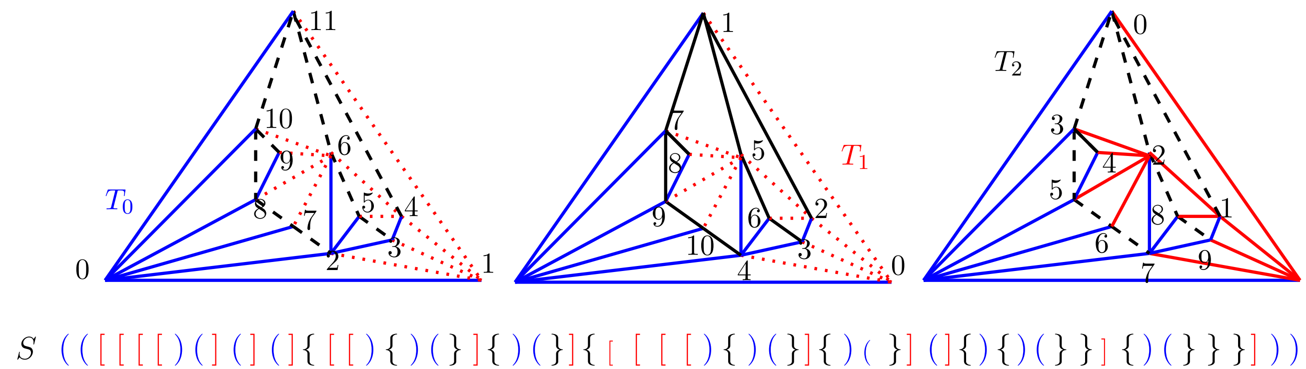
(**graph drawing**, **graph encoding**, succinct representations, compact data structures, exhaustive graph enumeration, bijective counting, greedy drawings, spanners, contact representations, planarity testing, untangling of planar graphs, Steinitz representations of polyhedra, ...)

Some (classical) applications

(Chuang, Garg, He, Kao, Lu, Icalp'98)

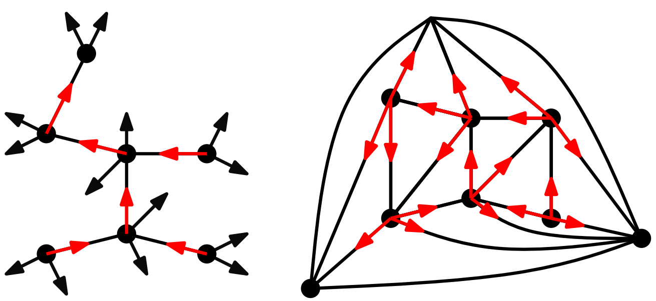
(He, Kao, Lu, 1999)

Graph encoding



(Poulalhon-Schaeffer, Icalp 03)

bijective counting, random generation

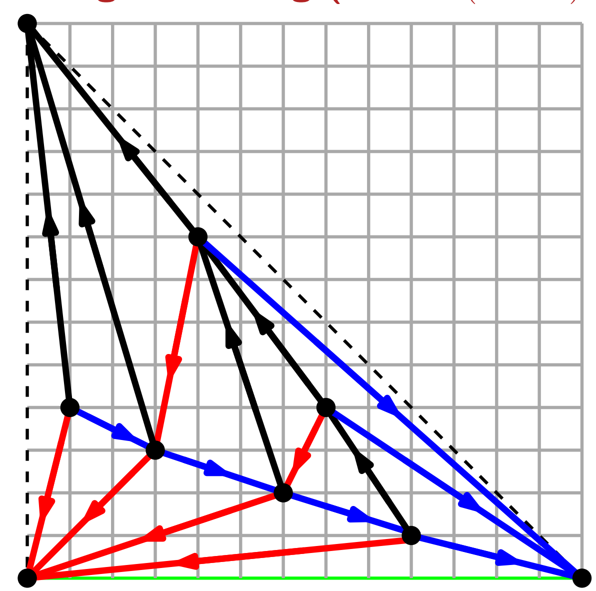


$$c_n = \frac{2(4n+1)!}{(3n+2)!(n+1)!}$$

$\Rightarrow$ optimal encoding ≈ 3.24 bits/vertex

Thm (Schnyder '90)

planar straight-line grid drawing (on a $O(n \times n)$ grid)



More ("recent") applications

Schnyder woods, TD-Delaunay graphs, orthogonal surfaces and Half-Θ_6-graphs

[Bonichon et al., WG'10, Icalp '10, ...]

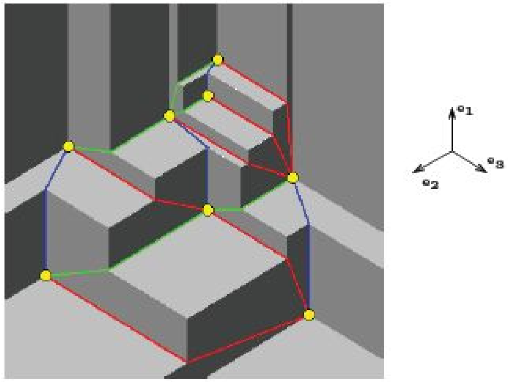
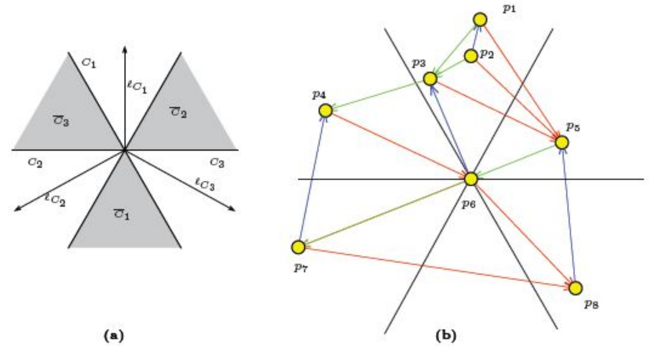


Figure 2: A coplanar orthogonal surface with its geodesic embedding.



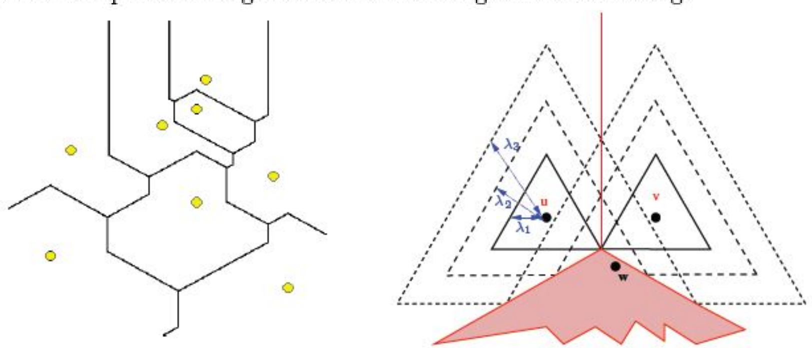
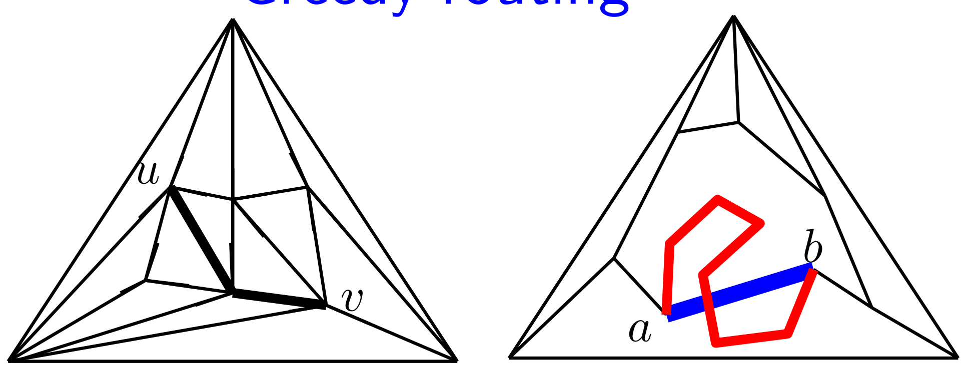


Figure 3: (a) TD-Voronoi diagram. (b) $\lambda_1 < \lambda_2 < \lambda_3$ stand for three triangular distances. Set $\{u, v\}$ is an ambiguous point set, however $\{u, v, w\}$ is non-ambiguous.

Greedy routing



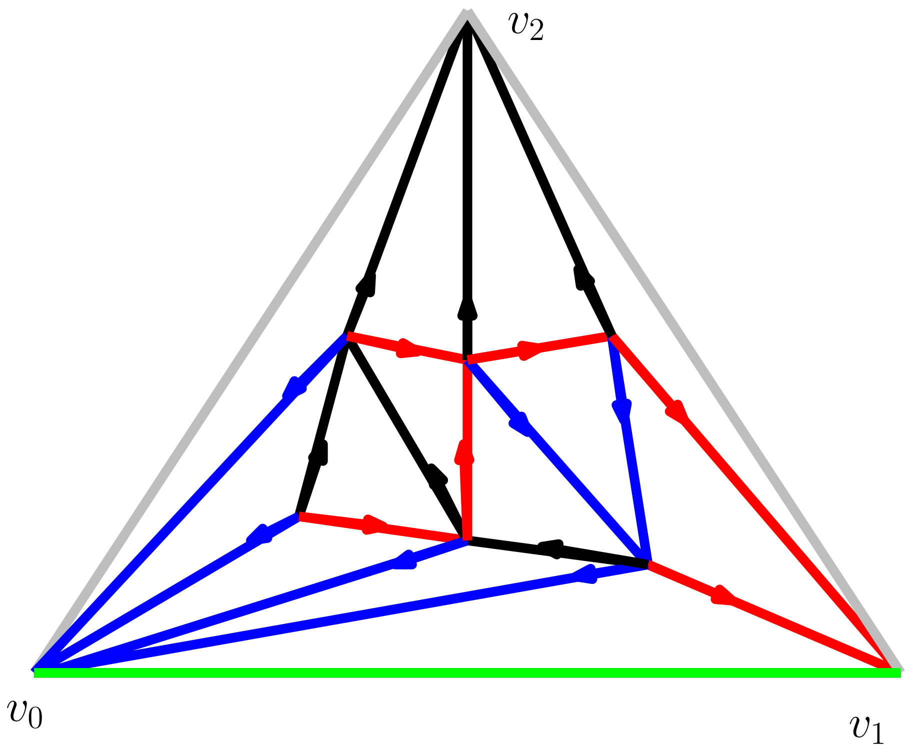
Every planar triangulation admits a **greedy drawing** (Dhandapani, Soda08)

(conjectured by Papadimitriou and Ratajczak for 3-connected planar graphs)

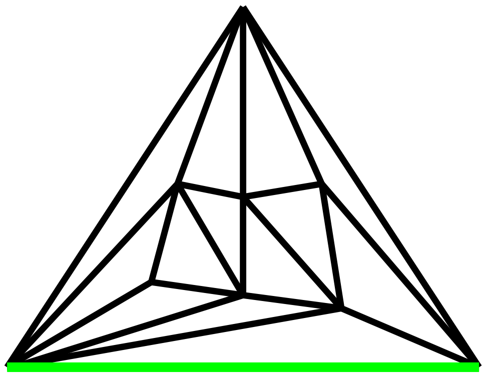
Schnyder woods

(the definition)

Schnyder woods: (planar) definition



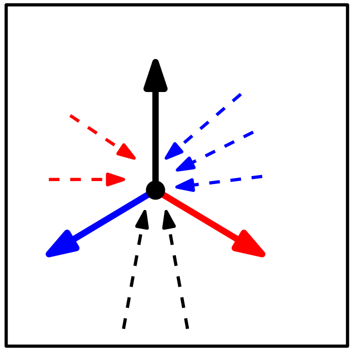
[Schnyder '90]



rooted triangulation on n nodes

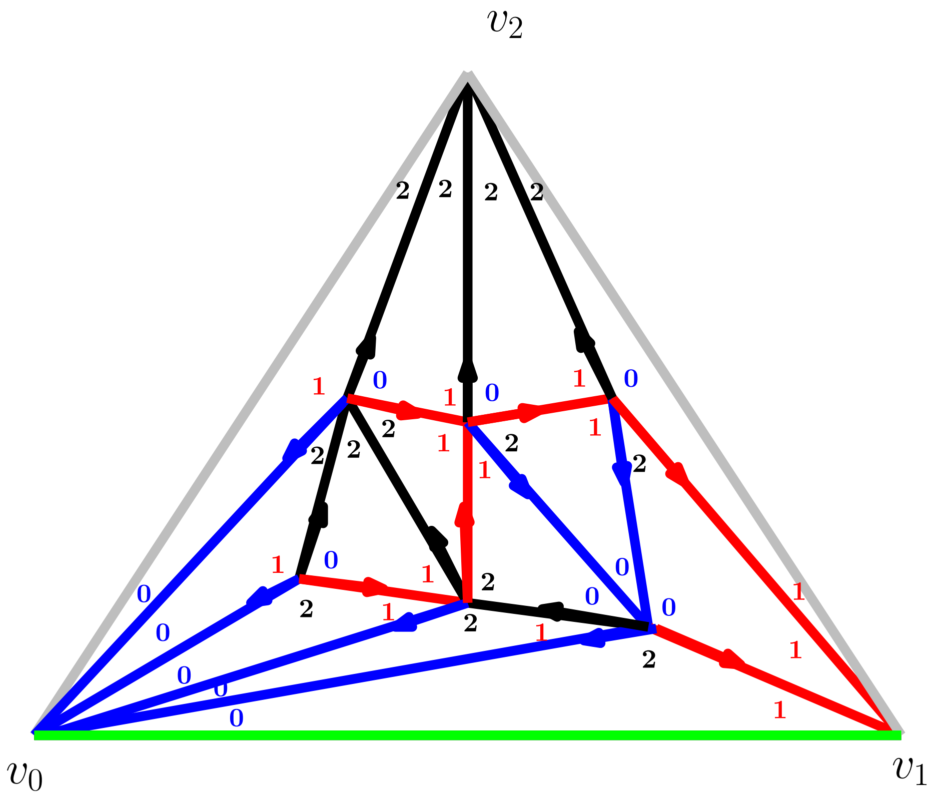
A Schnyder wood of a (rooted) planar triangulation is partition of all inner edges into three sets T_0, T_1 and T_2 such that

i) edge are colored and oriented in such a way that each inner nodes has exaclty one outgoing edge of each color

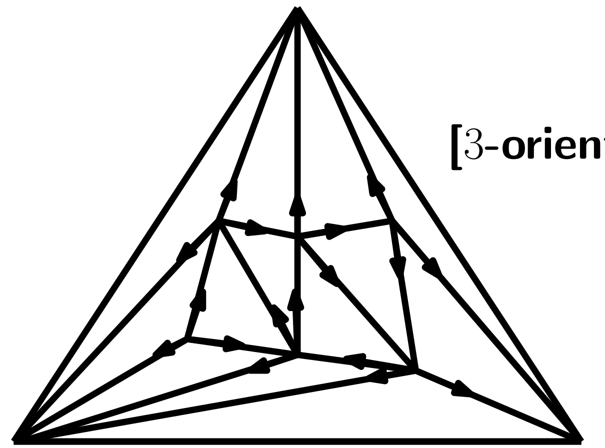


ii) colors and orientations around each inner node must respect the local Schnyder condition

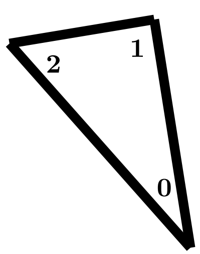
Schnyder woods: equivalent formulation



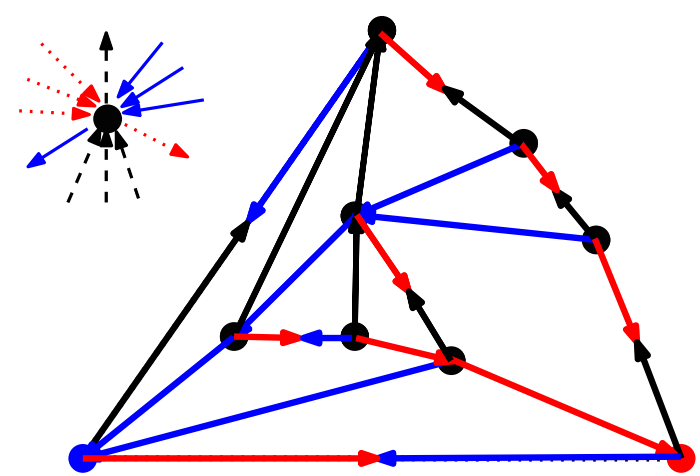
[normal labeling]



[3-orientation]

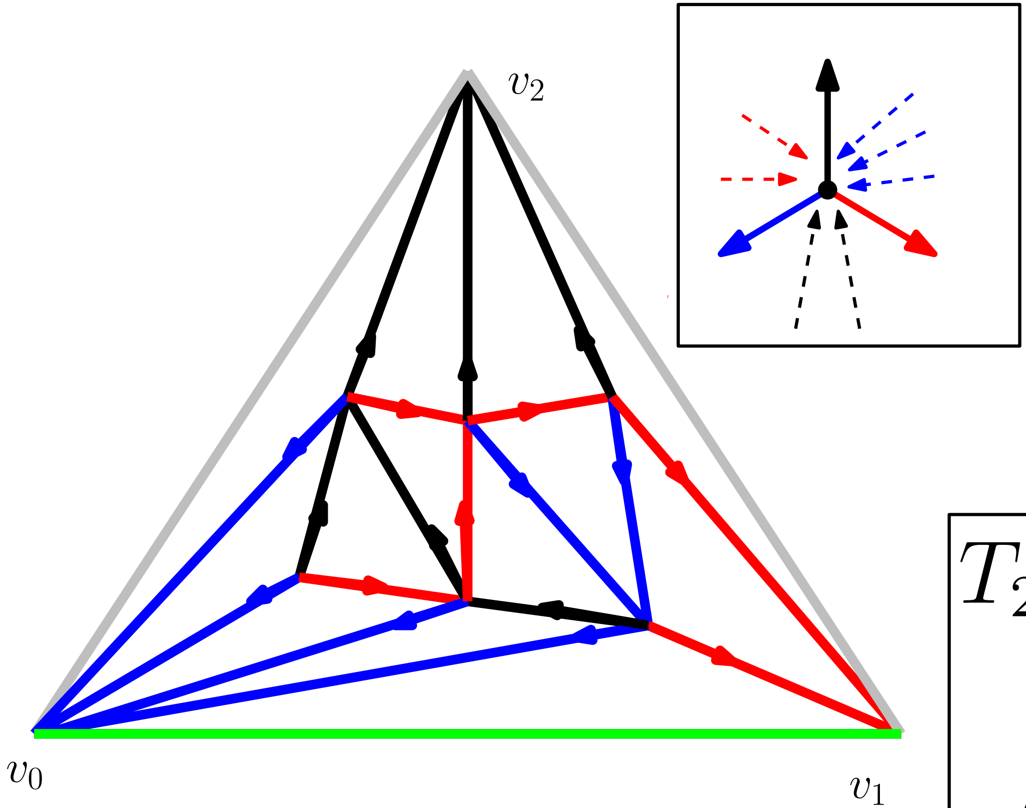


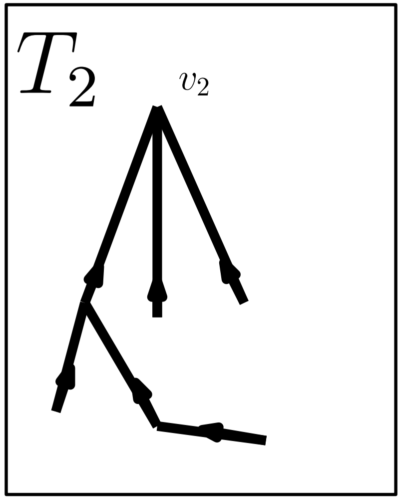
3-connected graphs [Felsner]

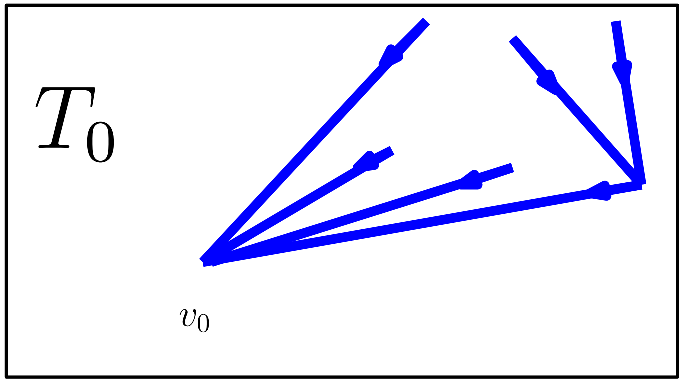


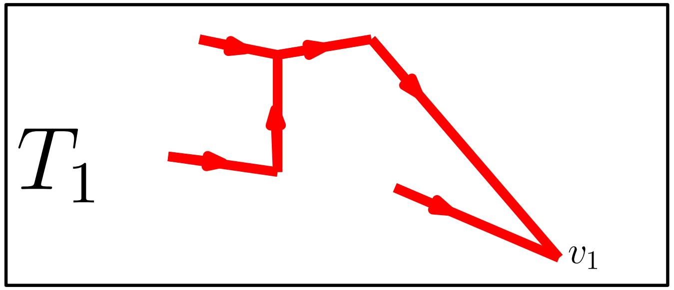
Schnyder woods: spanning property

[Schnyder '90]









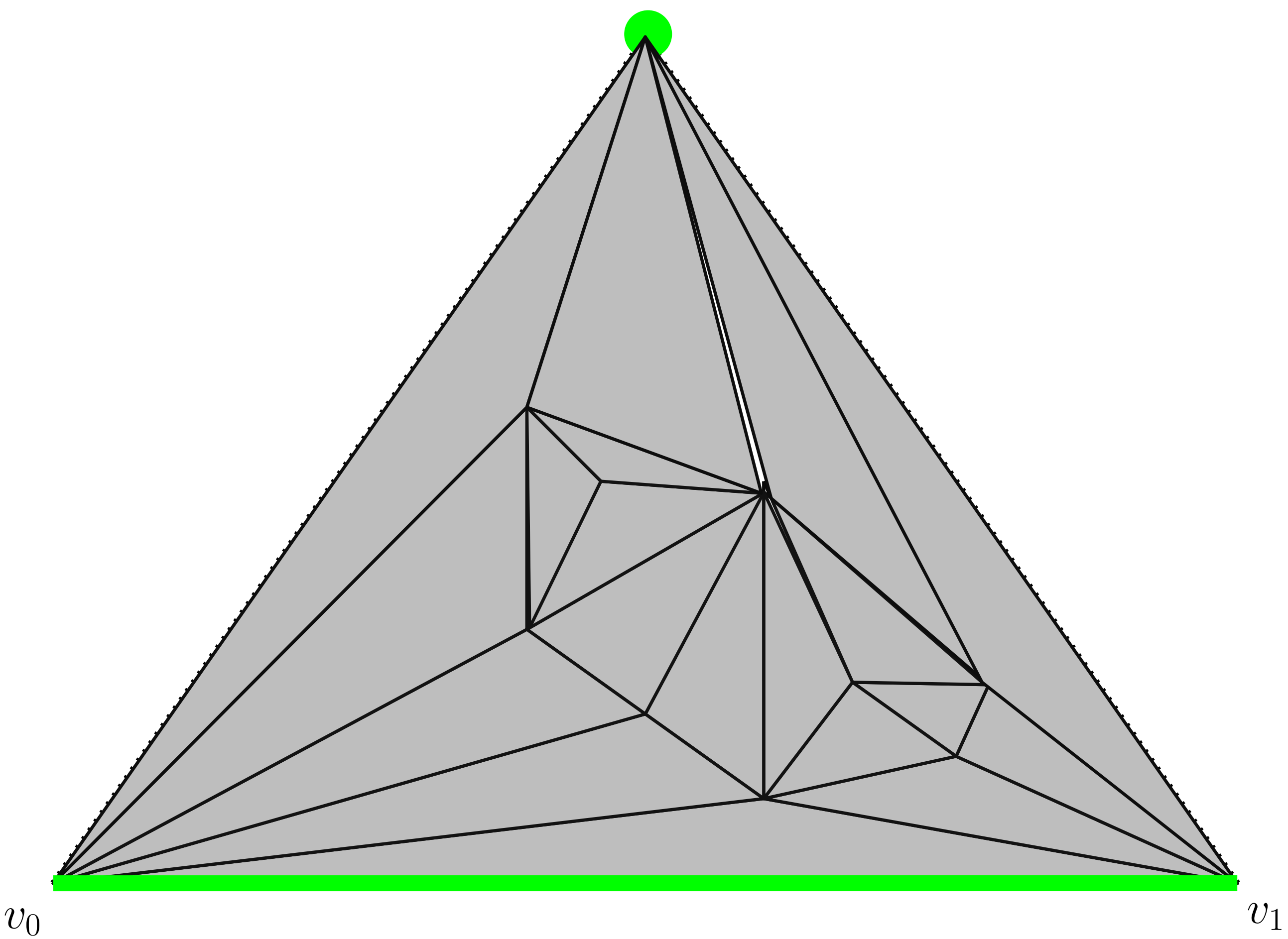
Theorem
The three sets T_0, T_1, T_2 are spanning trees of the inner vertices of $\mathcal{T}$ (each rooted at vertex v_i)

Schnyder woods: existence (algorithm I)

[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.

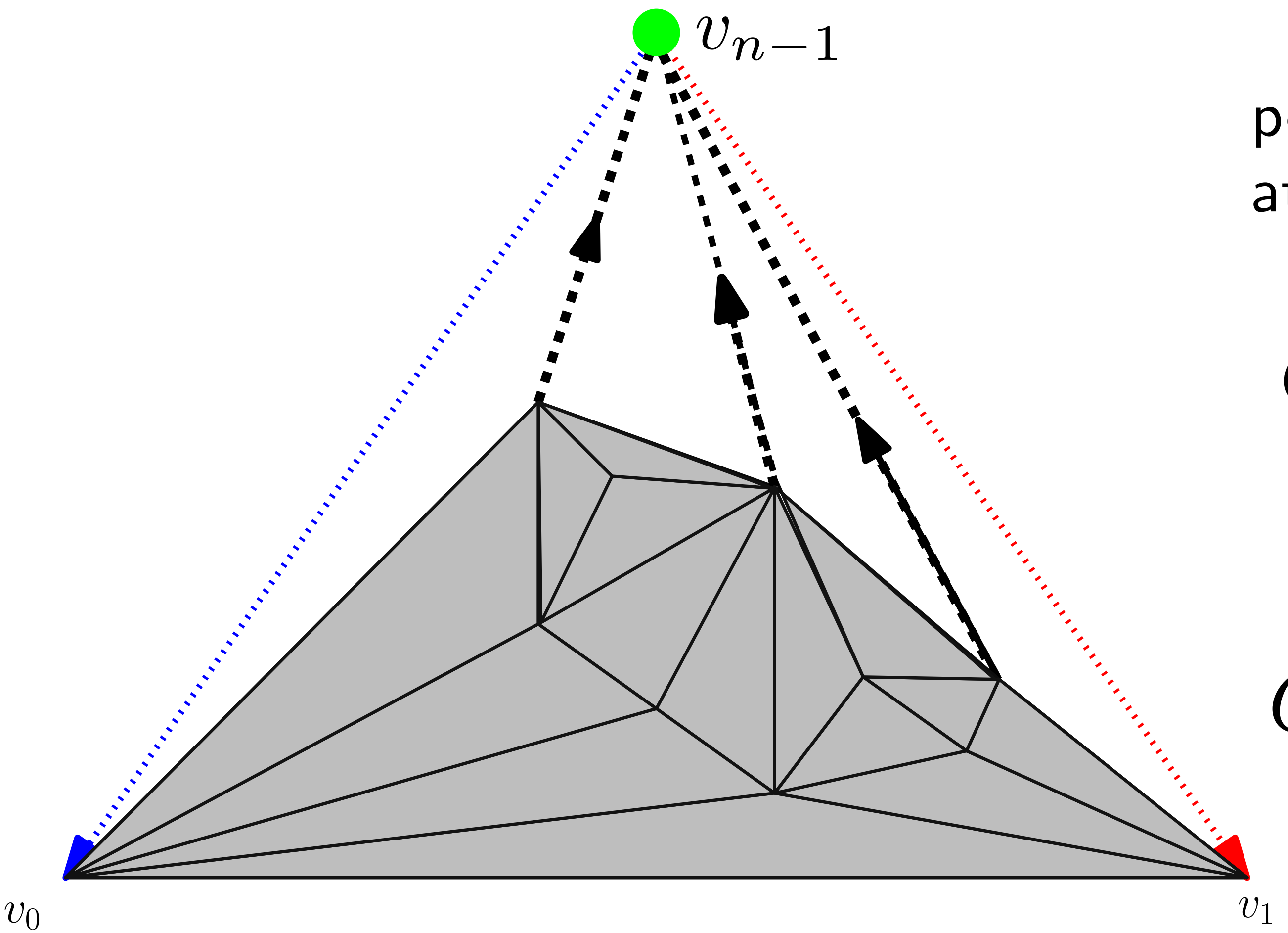


Schnyder woods: existence (algorithm I)

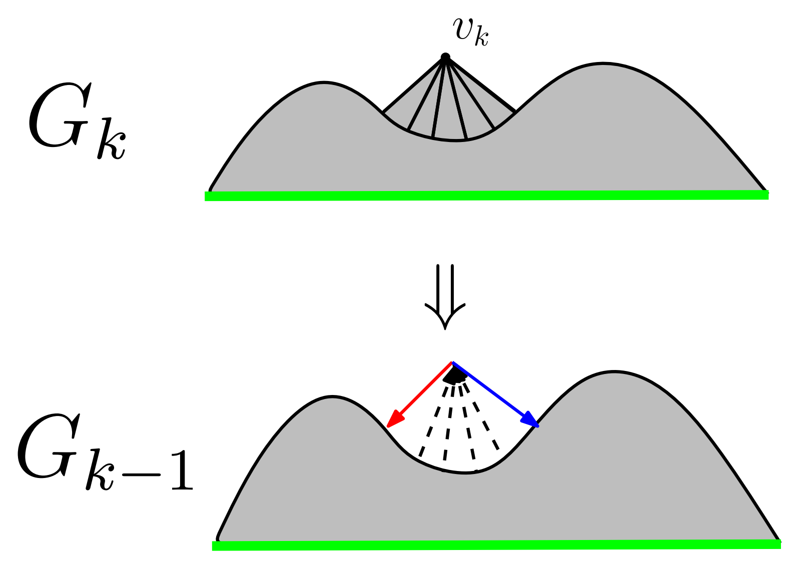
[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.



perform a vertex conquest at each step

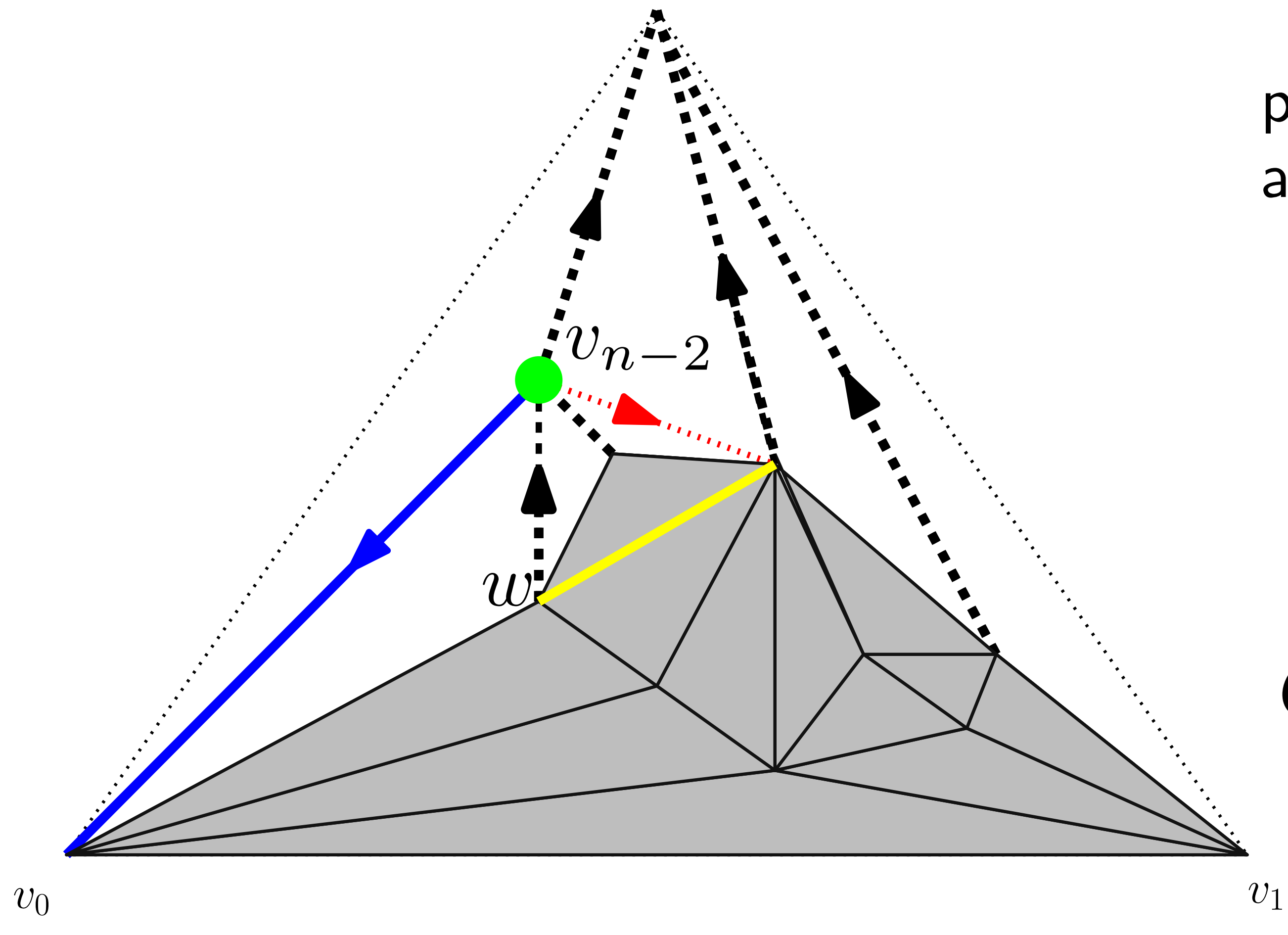


Schnyder woods: existence (algorithm I)

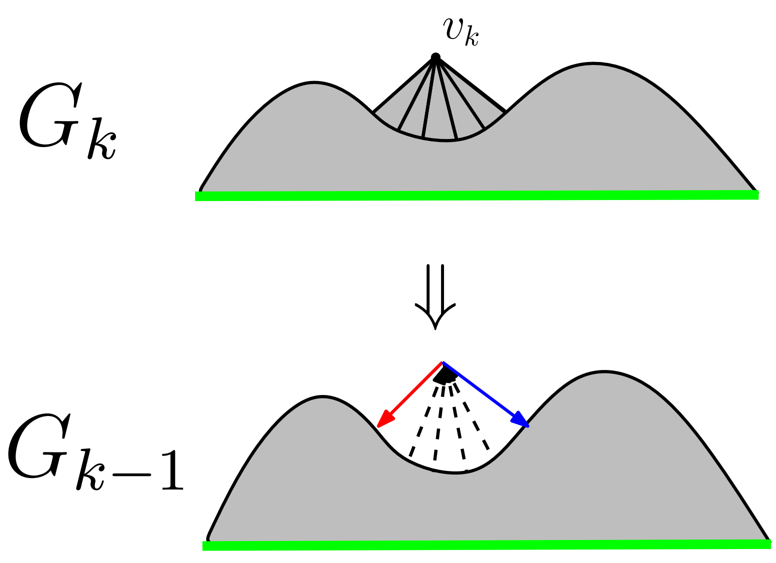
[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.



perform a vertex conquest at each step

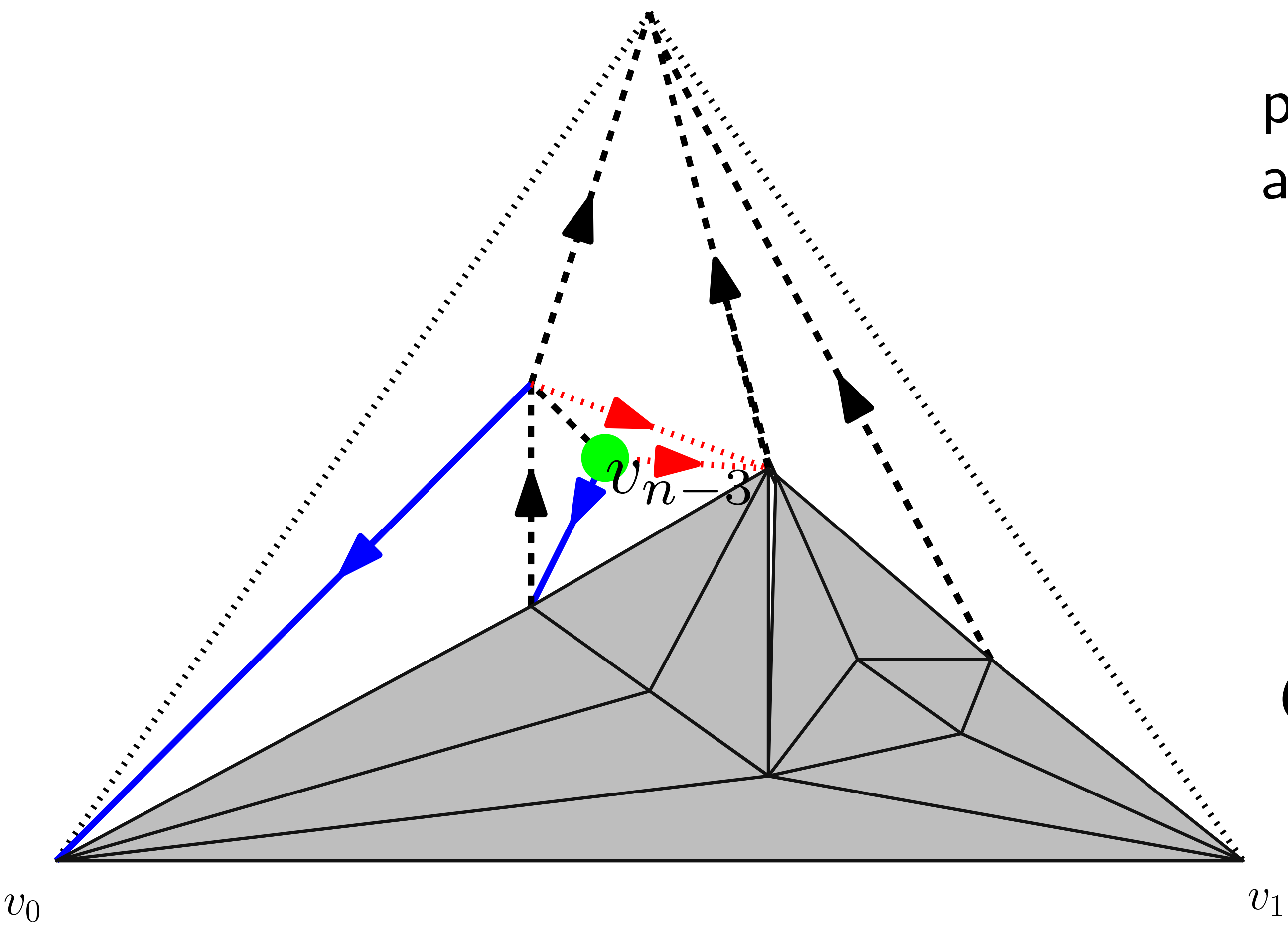


Schnyder woods: existence (algorithm I)

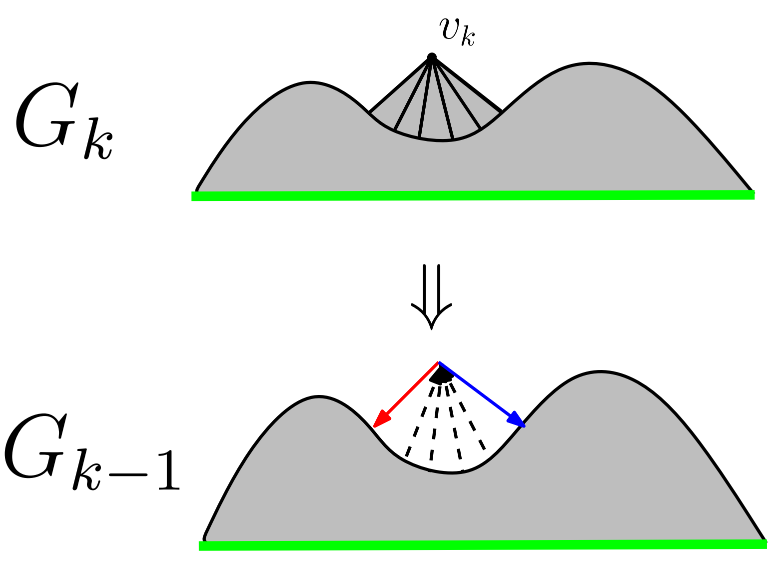
[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.



perform a vertex conquest at each step

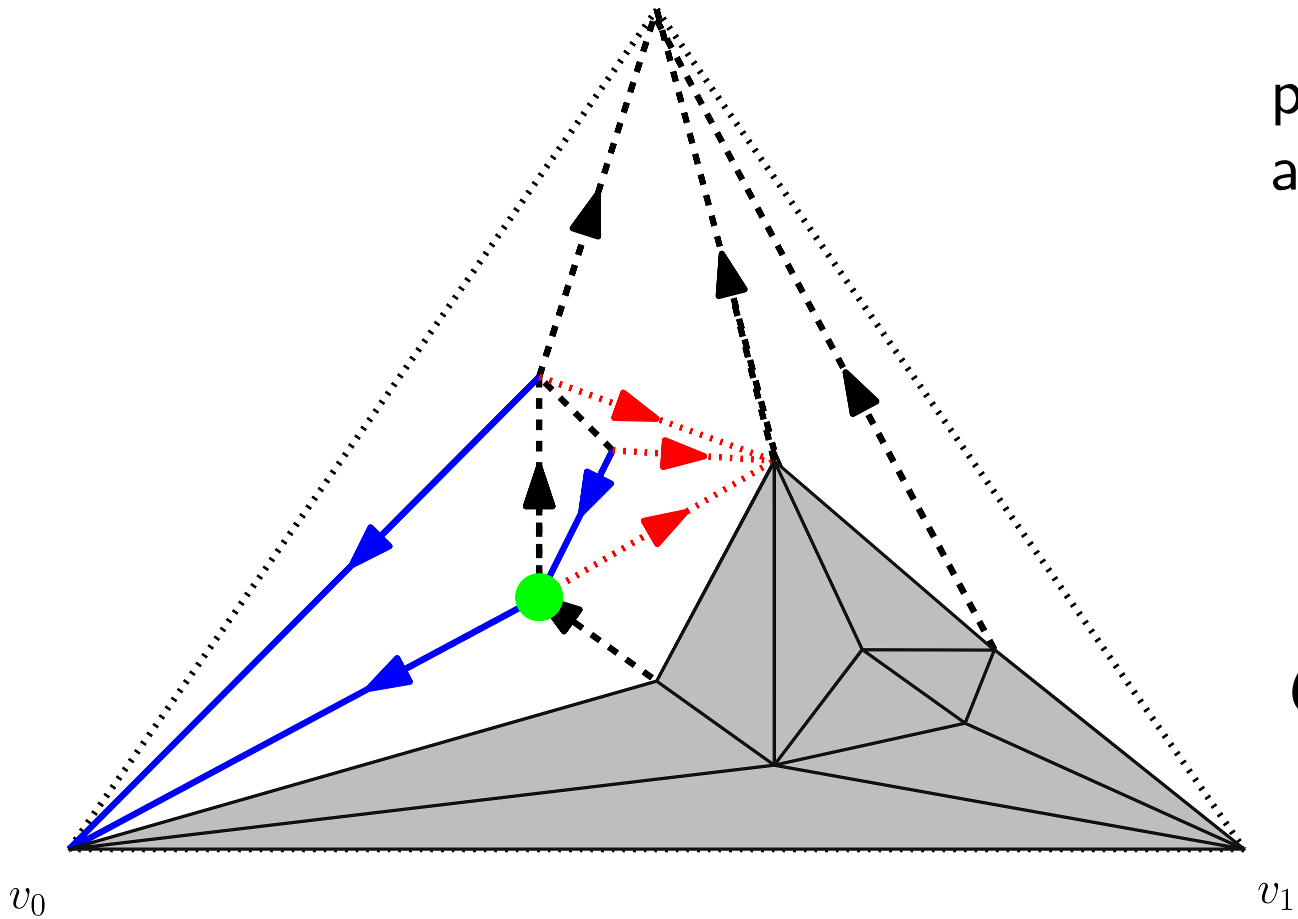


Schnyder woods: existence (algorithm I)

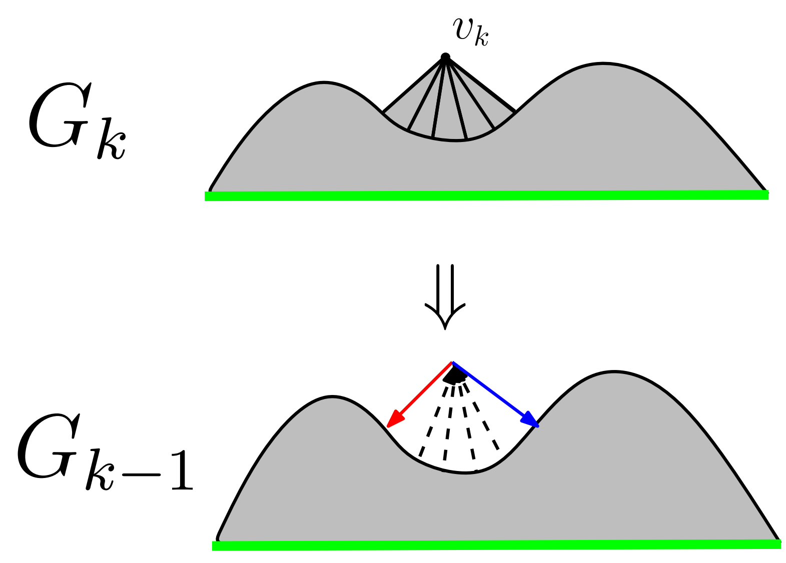
[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.



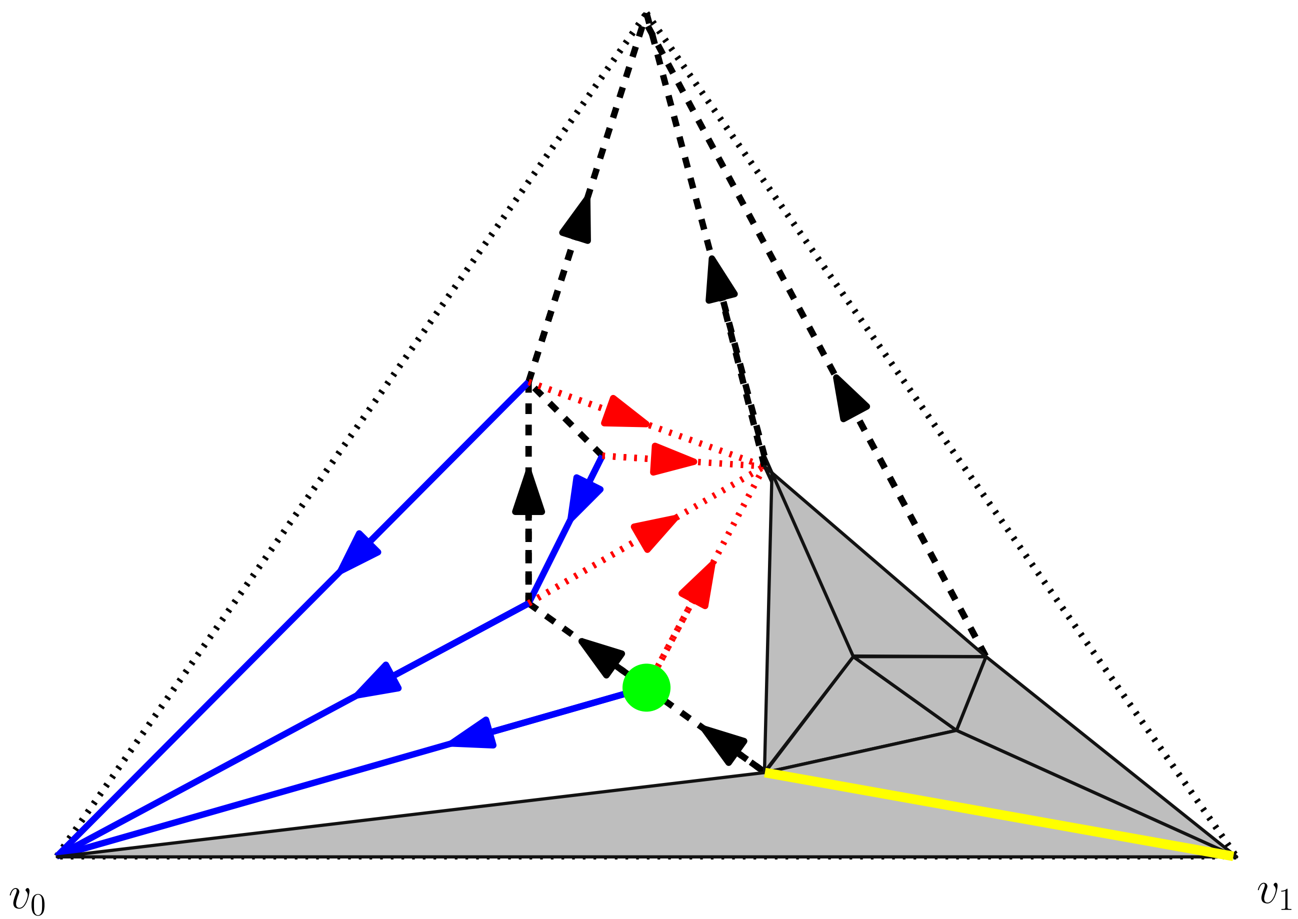
perform a vertex conquest at each step



Schnyder woods: existence (algorithm I)

[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face



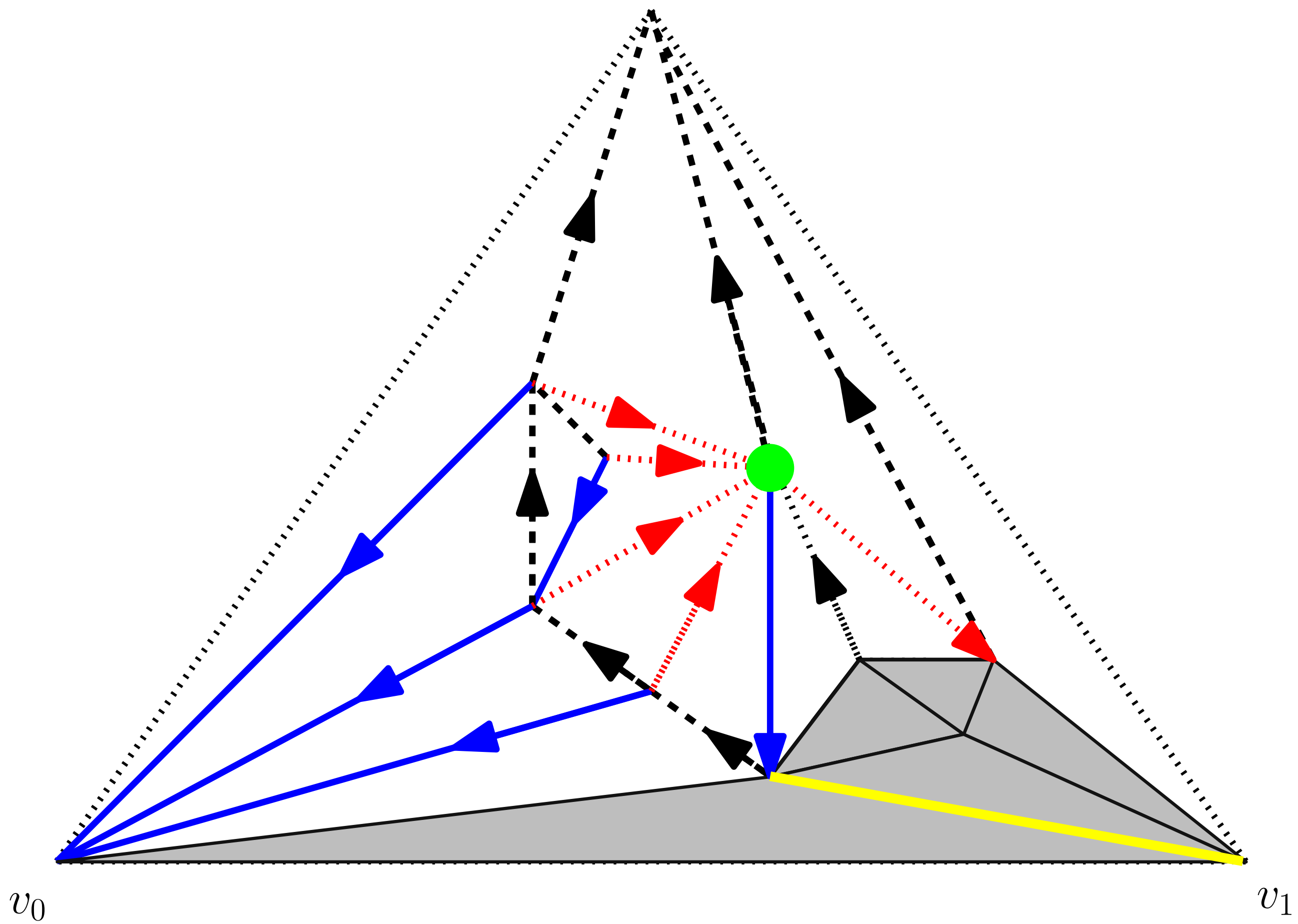
Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.

Schnyder woods: existence (algorithm I)

[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

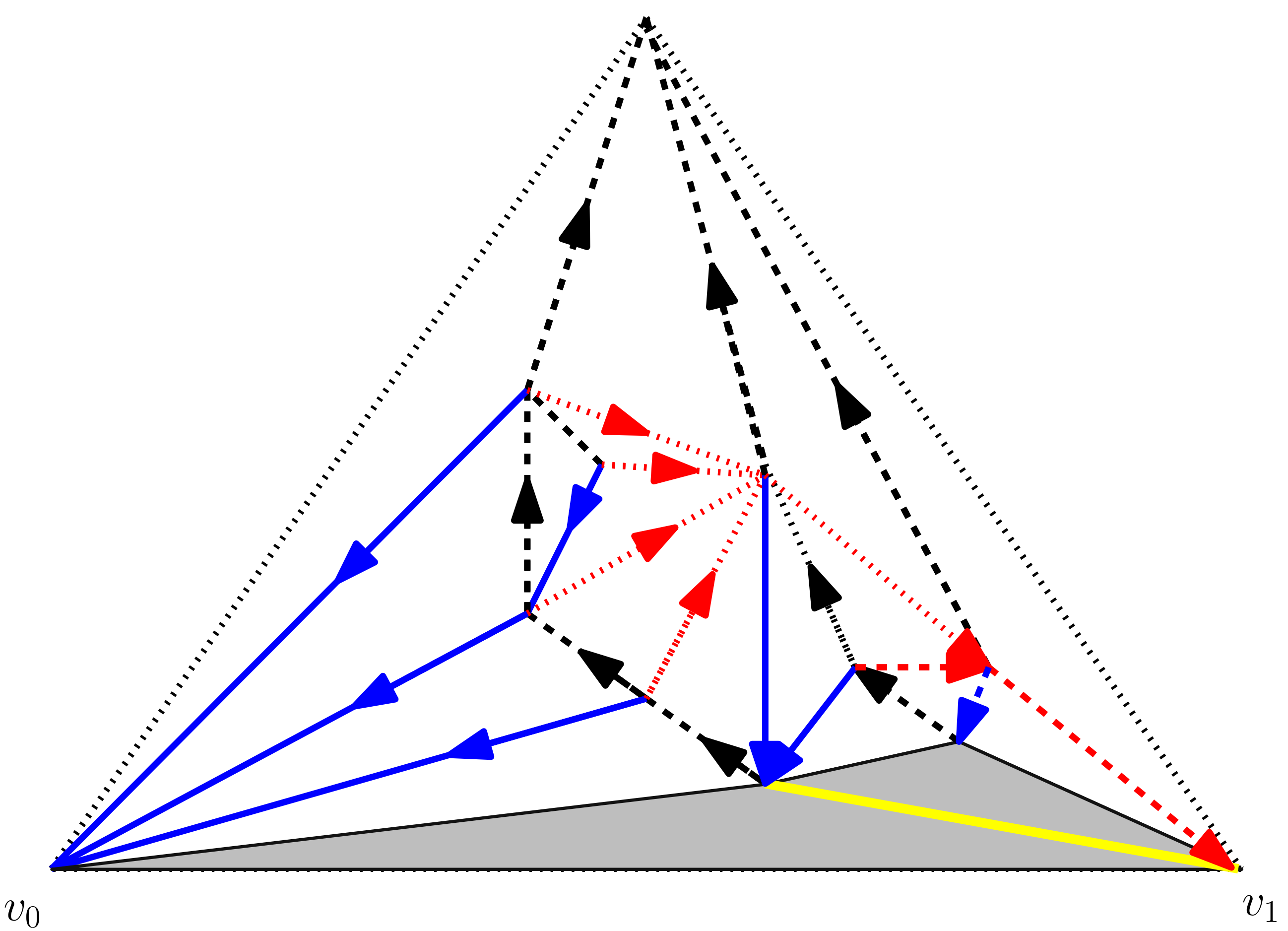
Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.



Schnyder woods: existence (algorithm I)

[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face



Theorem

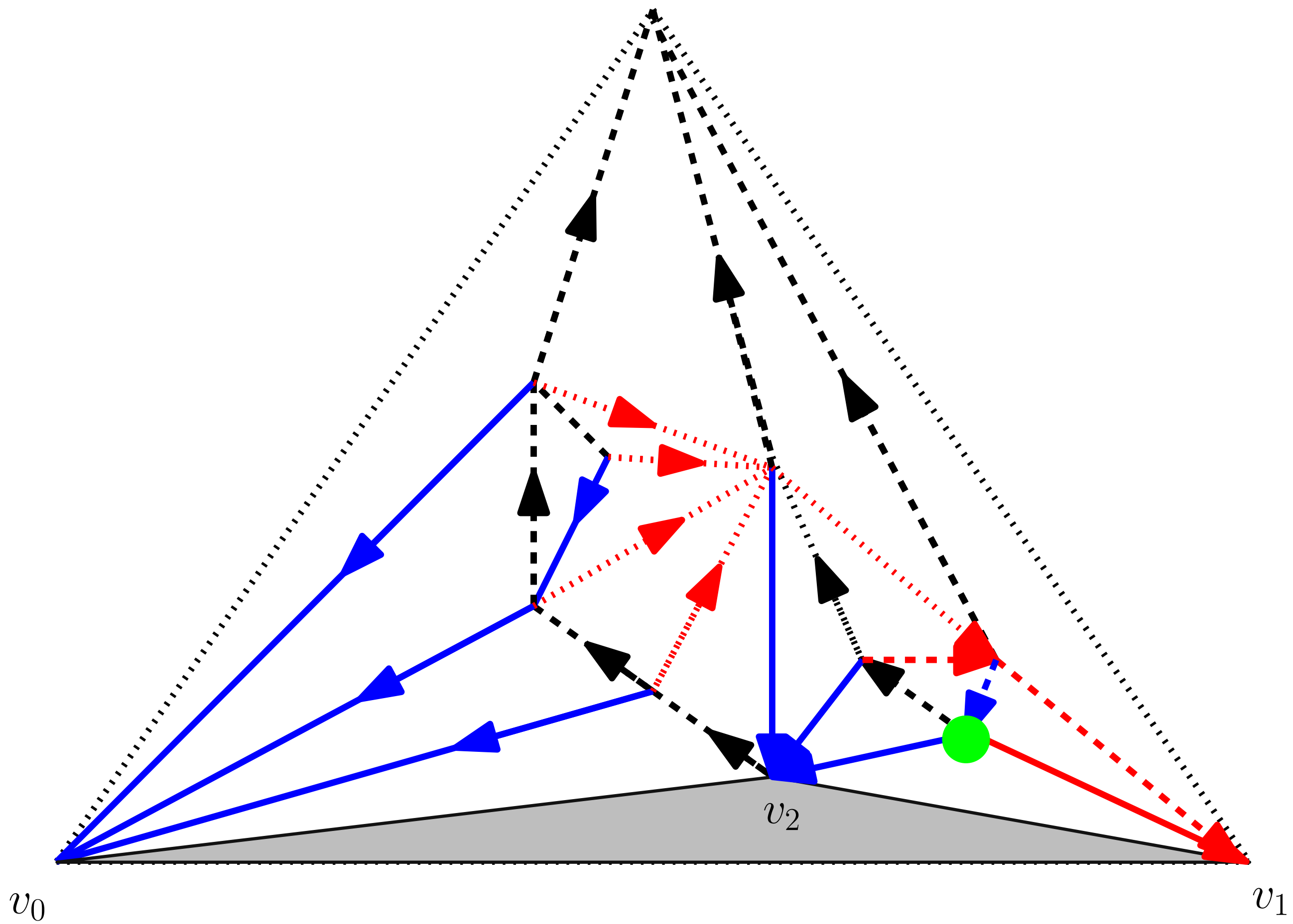
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.

Schnyder woods: existence (algorithm I)

[incremental vertex shelling, Brehm's thesis]

The traversal starts from the root face

Theorem
Every planar triangulation admits a Schnyder wood, which can be computed in linear time.

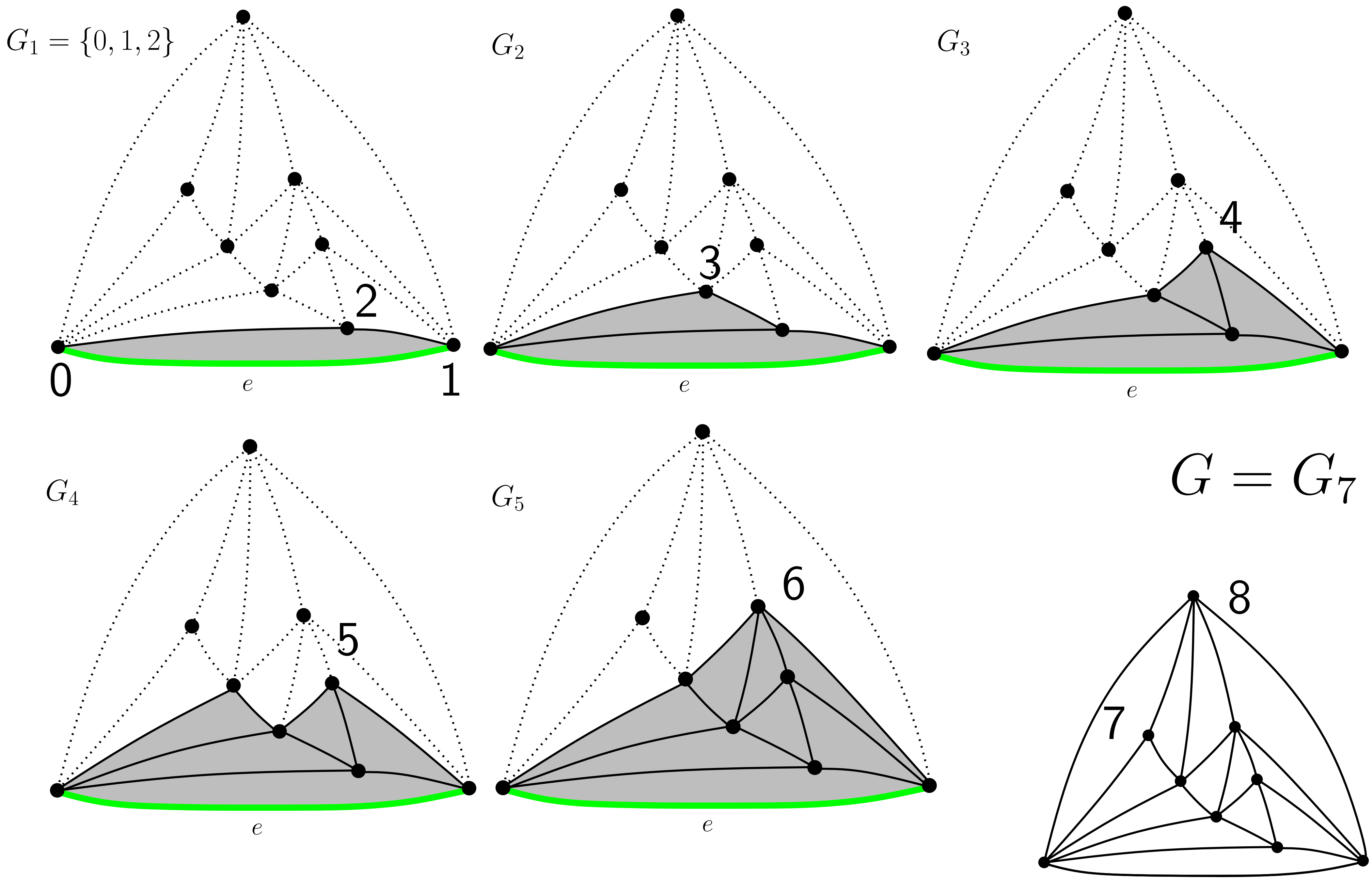


Canonical orderings

(the definition)

Canonical orderings: definition

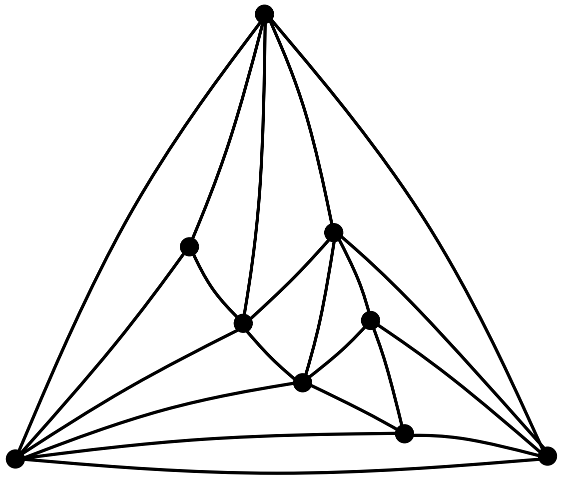
[de Fraysseix Pach Pollack]



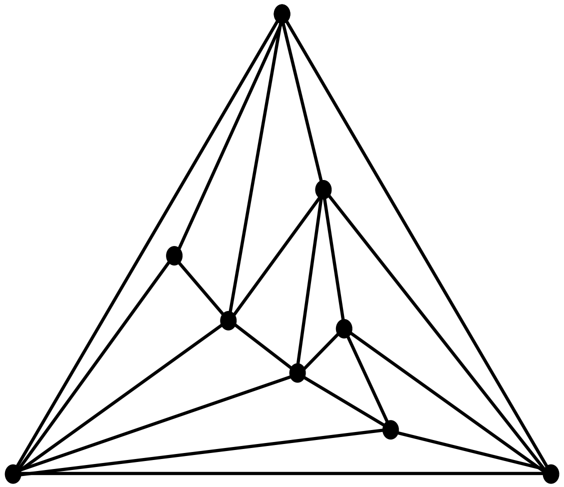
Planar straight-line drawings

(of planar graphs)

Planar straight-line drawings



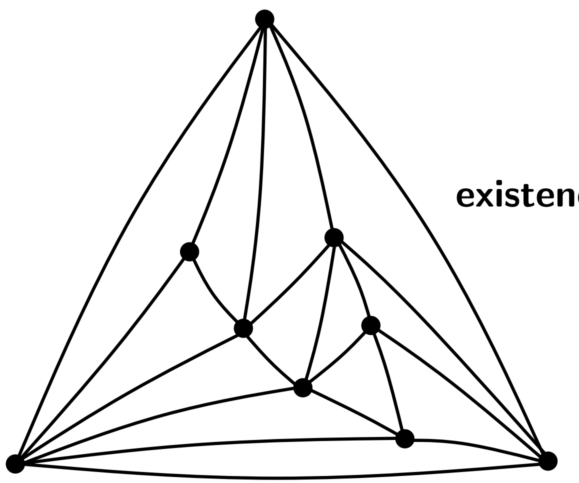
$\Rightarrow$



[Wagner'36]

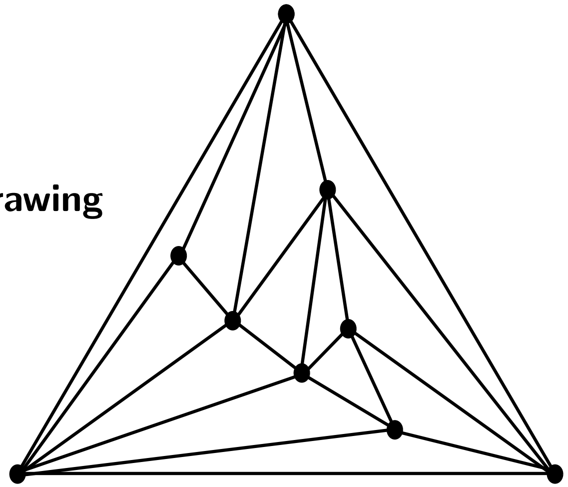
[Fary'48]

Planar straight-line drawings



existence of straight-line drawing

$\Rightarrow$

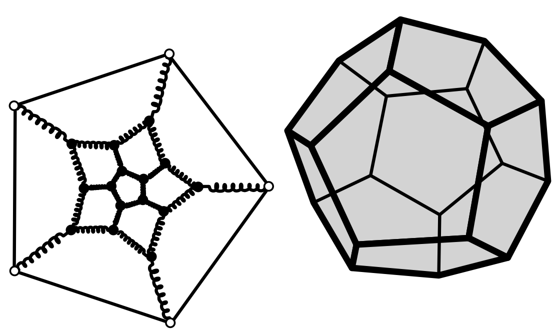


[Wagner'36]

[Fary'48]

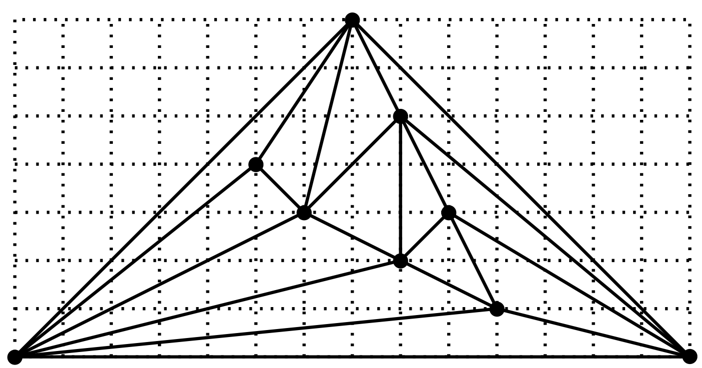
[Stein'51]

Classical algorithms:



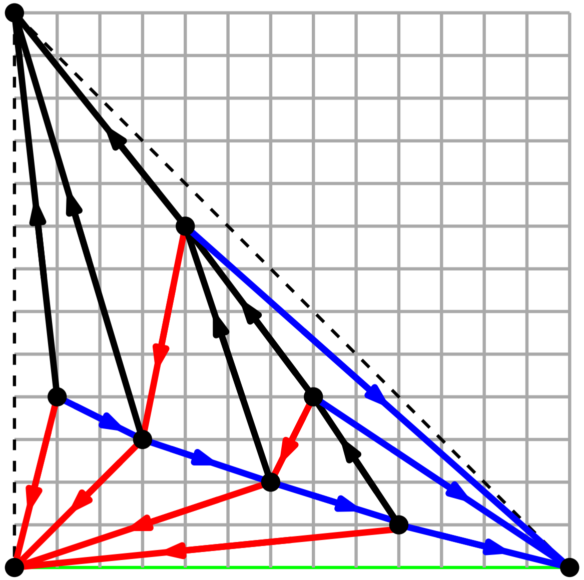
[Tutte'63]

spring-embedding



[De Fraysseix, Pach, Pollack 89]

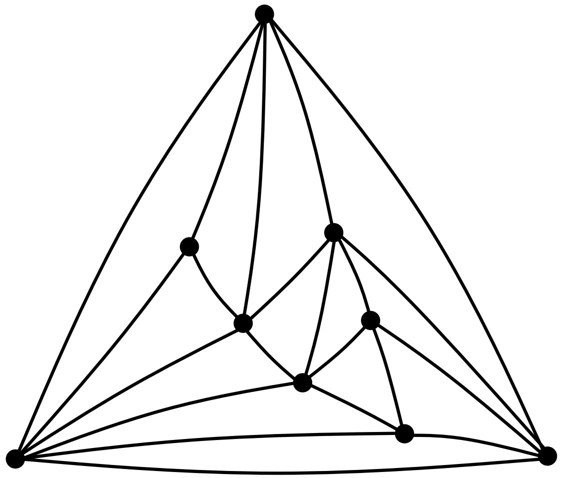
incremental (**Shift-algorithm**)



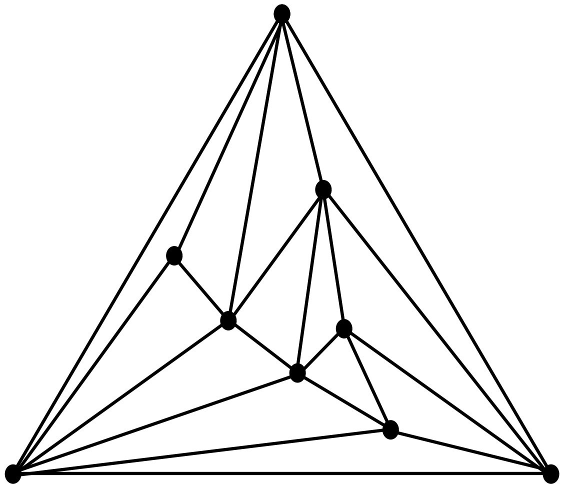
[Schnyder'90]

face-counting principle

Planar straight-line drawings



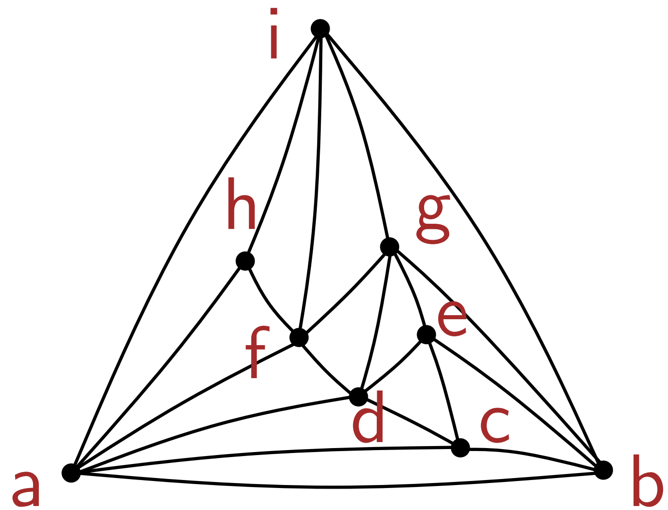
$\Rightarrow$



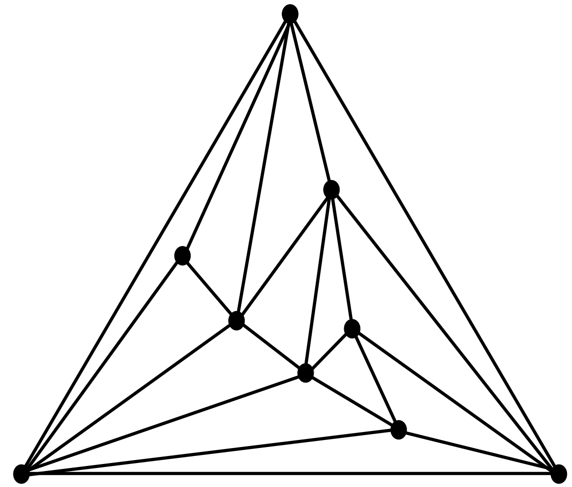
[Wagner'36]

[Fary'48]

Planar straight-line grid drawings



⇒



[Wagner'36]

[Fary'48]

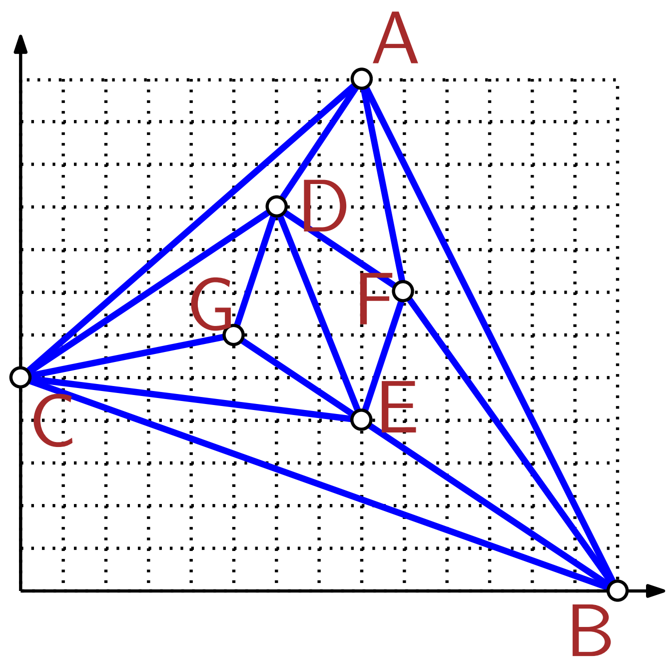
Input of the problem

set of triangle faces

(a, b, c) (d, e, g) (i, g, b)
(a, c, d) (e, b, g) (i, b, a)
(d, c, e) (a, f, h)
(c, b, e) (a, h, i)
(a, d, f) (i, h, f)
(f, d, g) (i, f, g)

Output

geometric coordinates of vertices

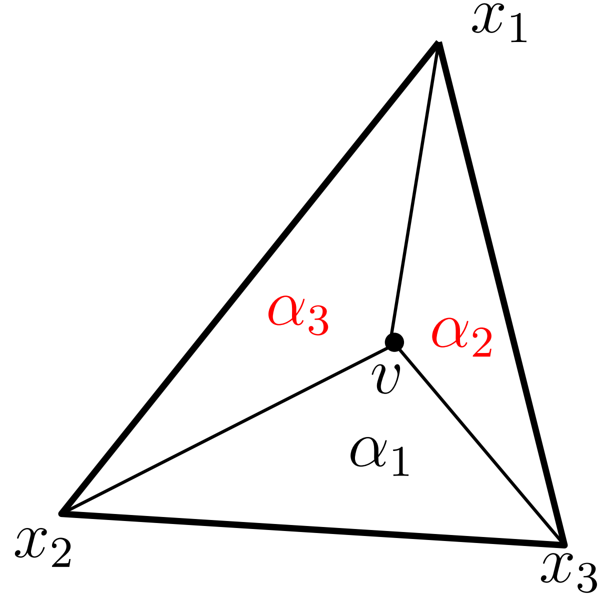


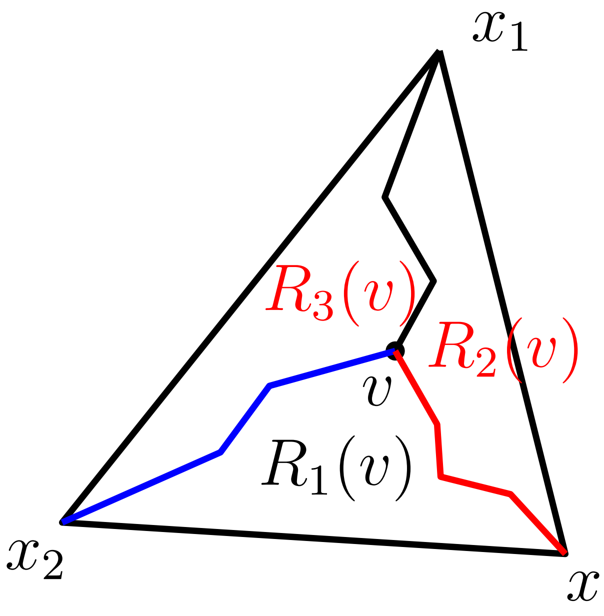
Face counting algorithm

(Schnyder algorithm, 1990)

Face counting algorithm

Geometric interpretation





$v = \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3$

where α_i is the normalized area

$v = \frac{|R_1(v)|}{|T|} x_1 + \frac{|R_2(v)|}{|T|} x_2 + \frac{|R_3(v)|}{|T|} x_3$

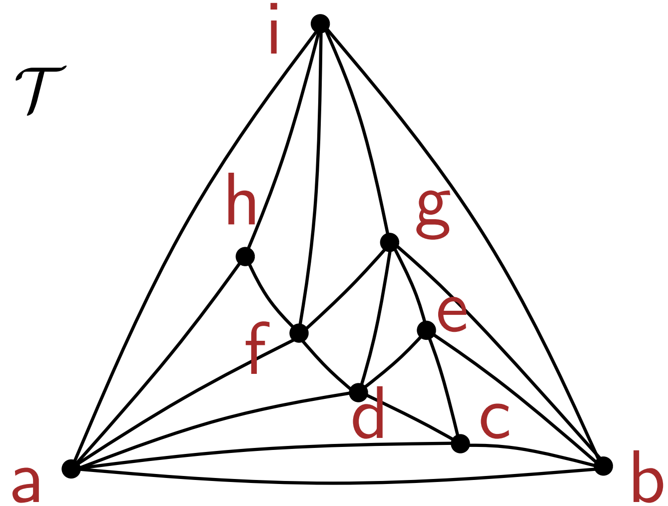
where $|R_i(v)|$ is the number of triangles

Theorem (Schnyder, Soda '90)

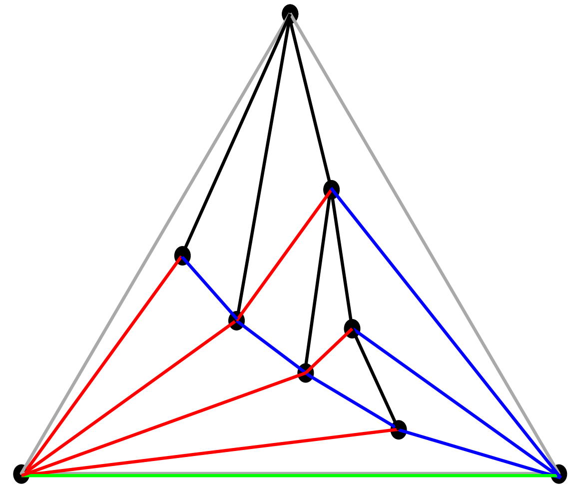
For a triangulation $\mathcal{T}$ having n vertices, we can draw it on a grid of size $(2n-5) \times (2n-5)$, by setting $x_1 = (2n-5, 0)$, $x_2 = (0, 0)$ and $x_3 = (0, 2n-5)$.

Face counting algorithm

Input: $\mathcal{T}$



$\Rightarrow$



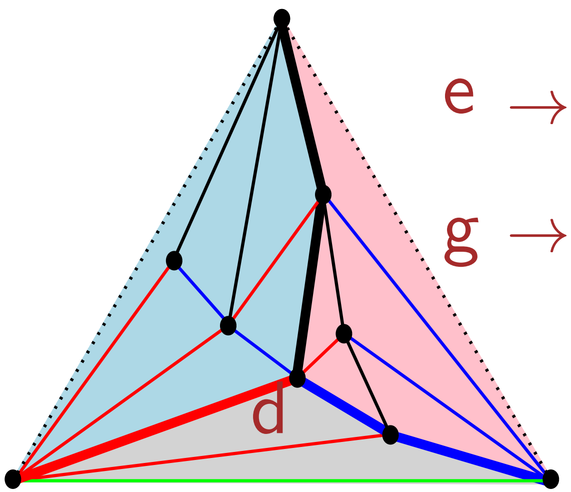
$\mathcal{T}$ endowed with a Schnyder wood

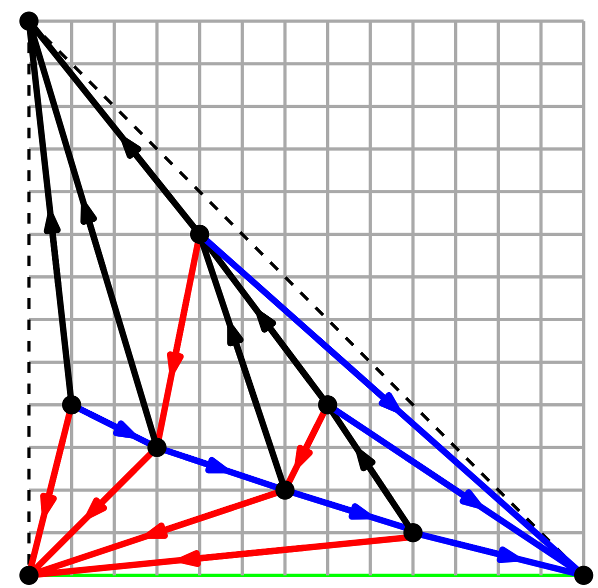
a $\to (0,0)$ b $\to (0,1)$ i $\to (1,0)$

c $\to (\frac{9}{13}, \frac{1}{13})$ **d** $\to (\mathbf{\frac{5}{13}}, \mathbf{\frac{6}{13}})$

e $\to (\frac{7}{13}, \frac{4}{13})$ f $\to (\frac{3}{13}, \frac{3}{13})$

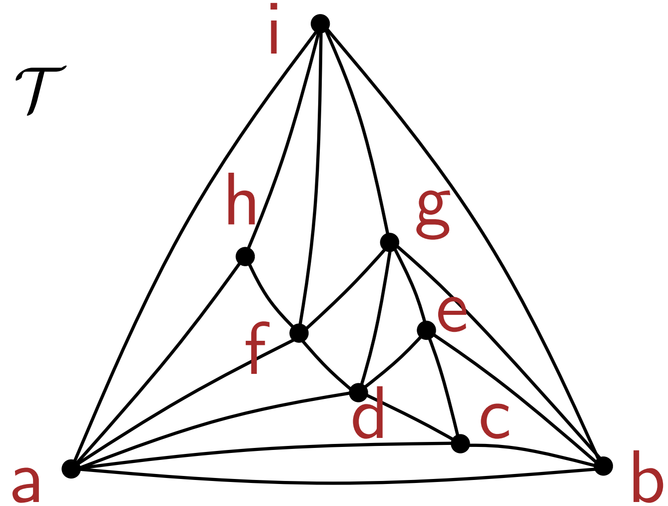
g $\to (\frac{4}{13}, \frac{8}{13})$ h $\to (\frac{1}{13}, \frac{4}{13})$



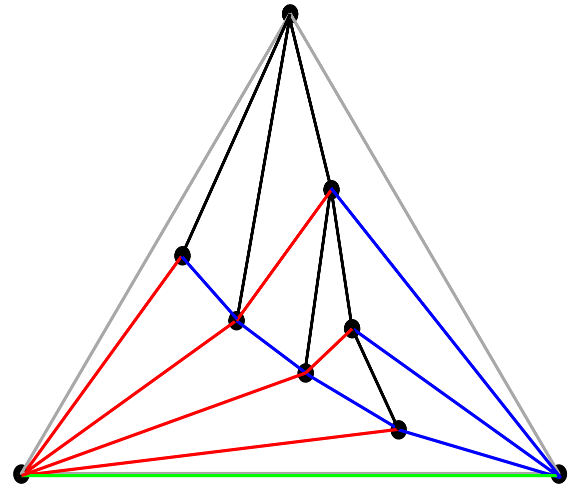


Face counting algorithm: proof (sketch)

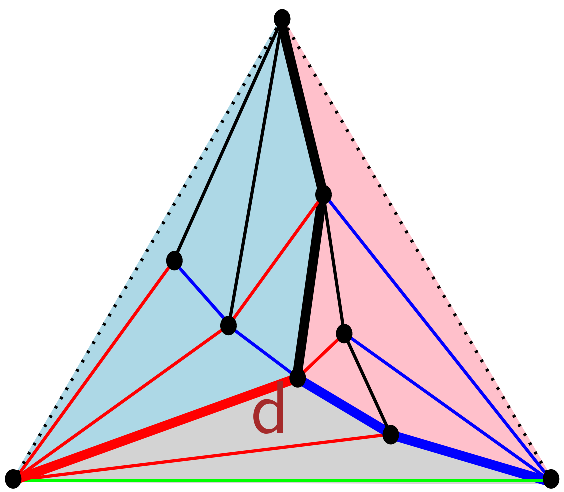
Input: $\mathcal{T}$



$\Rightarrow$



$\mathcal{T}$ endowed with a Schnyder wood



a $\rightarrow (13, 0, 0)$

b $\rightarrow (0, 13, 0)$

c $\rightarrow (9, 3, 1)$

d $\rightarrow (5, 6, 2)$

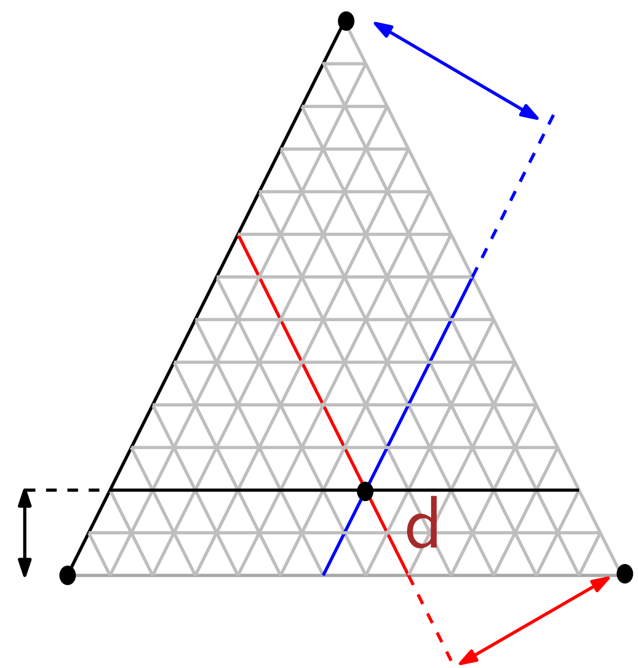
e $\rightarrow (2, 7, 4)$

f $\rightarrow (7, 3, 3)$

g $\rightarrow (1, 4, 8)$

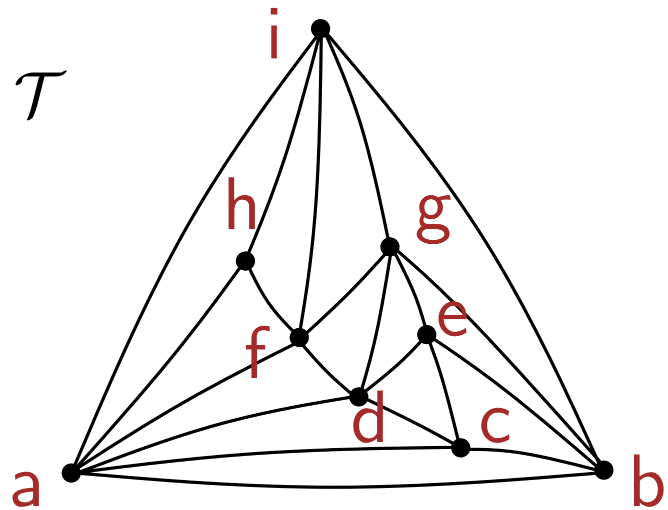
h $\rightarrow (8, 1, 4)$

i $\rightarrow (0, 0, 13)$

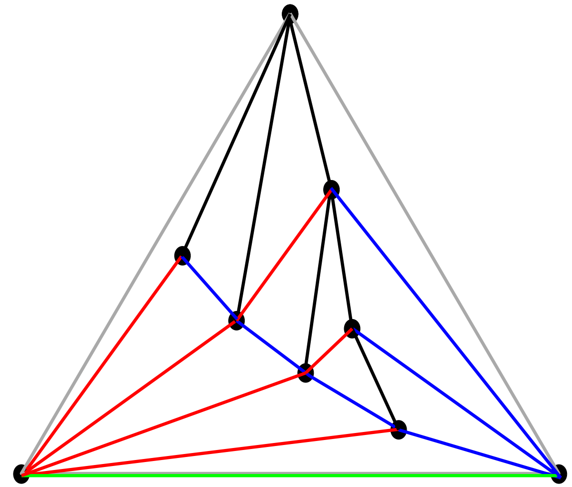


Face counting algorithm: proof (sketch)

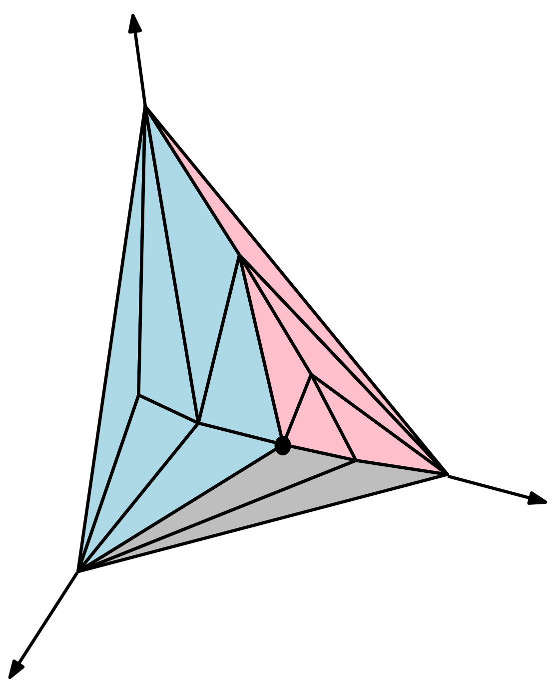
Input: $\mathcal{T}$



$\Rightarrow$



$\mathcal{T}$ endowed with a Schnyder wood



a $\rightarrow (13, 0, 0)$

b $\rightarrow (0, 13, 0)$

c $\rightarrow (9, 3, 1)$

d $\rightarrow (5, 6, 2)$

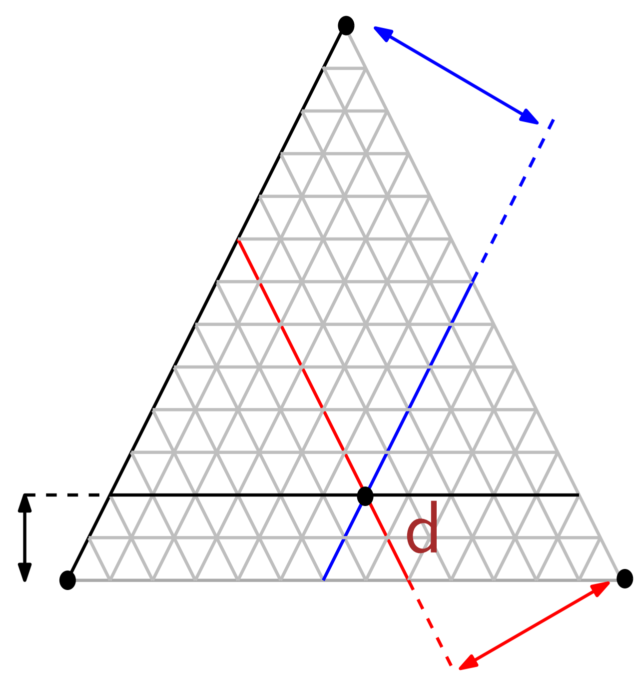
e $\rightarrow (2, 7, 4)$

f $\rightarrow (7, 3, 3)$

g $\rightarrow (1, 4, 8)$

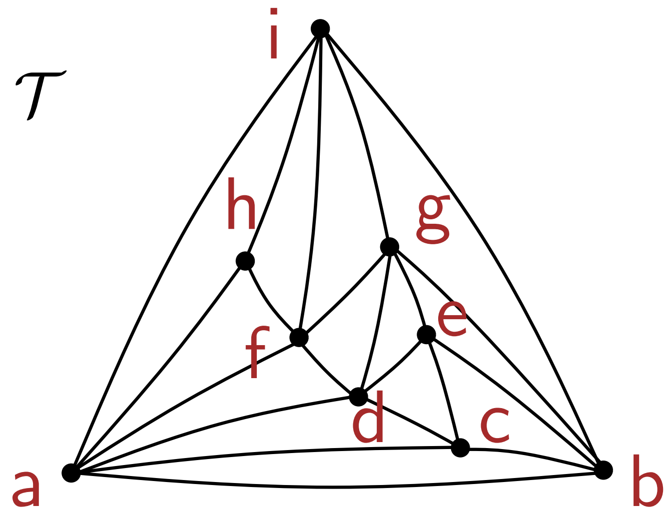
h $\rightarrow (8, 1, 4)$

i $\rightarrow (0, 0, 13)$

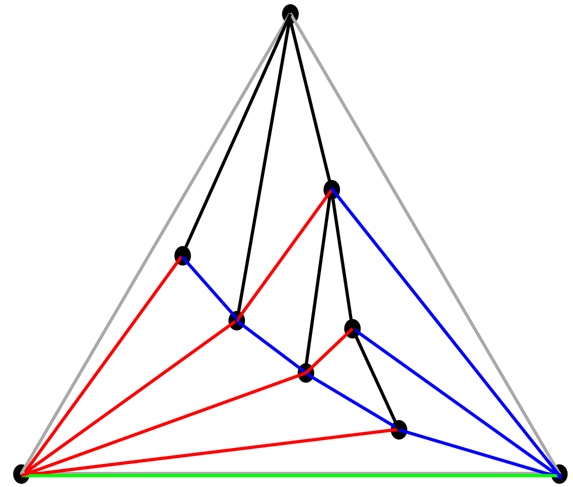


Face counting algorithm: proof (sketch)

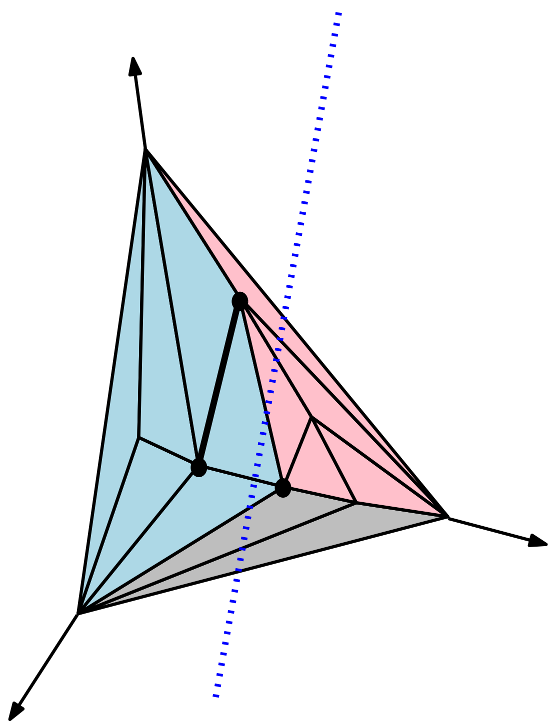
Input: $\mathcal{T}$



$\Rightarrow$



$\mathcal{T}$ endowed with a Schnyder wood



a $\rightarrow (13, 0, 0)$

b $\rightarrow (0, 13, 0)$

c $\rightarrow (9, 3, 1)$

d $\rightarrow (5, 6, 2)$

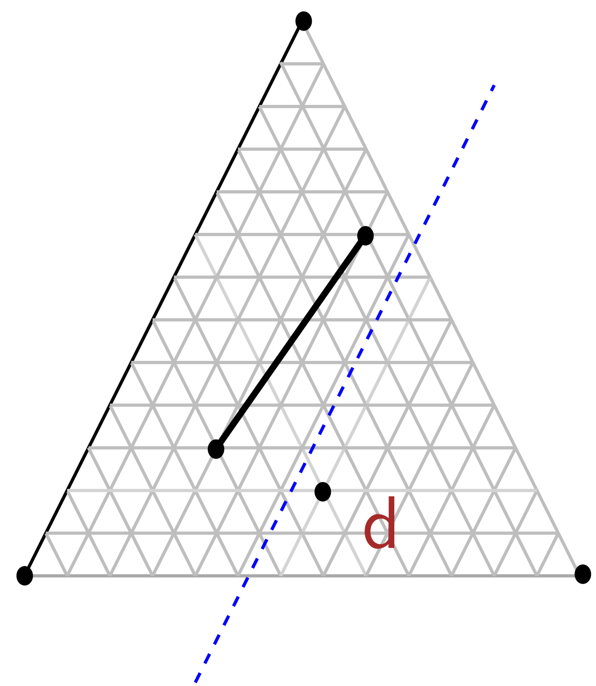
e $\rightarrow (2, 7, 4)$

f $\rightarrow (7, 3, 3)$

g $\rightarrow (1, 4, 8)$

h $\rightarrow (8, 1, 4)$

i $\rightarrow (0, 0, 13)$

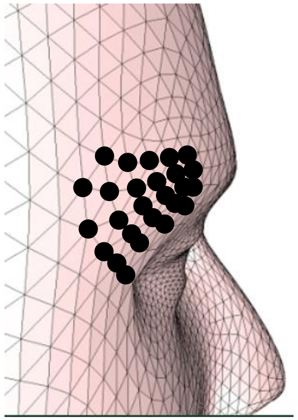


Graph encoding

(practical) motivation

Geometric v.s combinatorial information

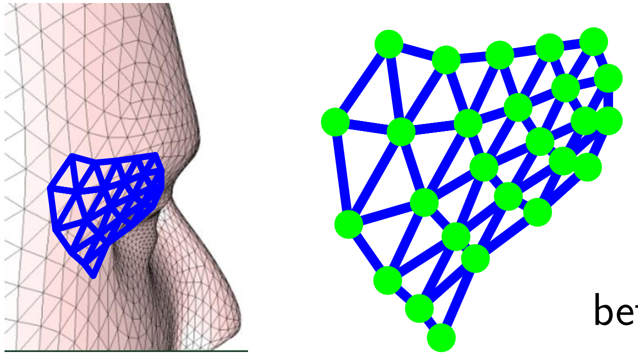
Geometry



vertex coordinates

between 30 et 96 bits/vertex

"Connectivity": the underlying triangulation



adjacency relations between triangles, vertices

vertex	1 reference to a triangle
triangle	3 references to vertices 3 references to triangles

$13n \log n$ or $416n$ bits

David statue (Stanford's Digital Michelangelo Project, 2000)

2 billions polygons

32 Giga bytes (without compression)

No existing algorithm nor data structure for dealing with the entire model

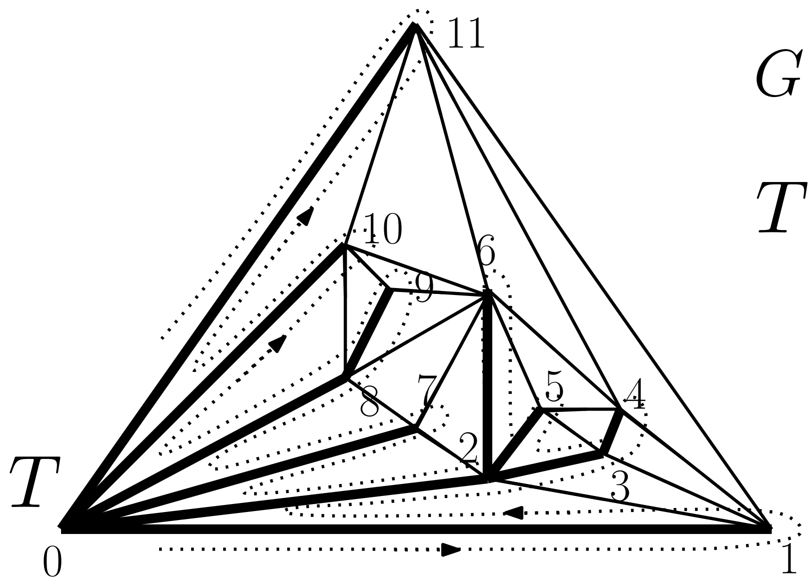
$$\#\{\text{triangulations}\} = \frac{2(4n+1)!}{(3n+2)!(n+1)!} \approx \frac{16}{27}\sqrt{\frac{3}{2\pi}}n^{-5/2}\left(\frac{256}{27}\right)^n$$

$$\Rightarrow \quad \text{entropy} = \log_2 \frac{256}{27} \approx 3.24 \text{ bpv.}$$

A simple encoding scheme

Turan encoding of planar map (1984)

$12n$ bits encoding scheme



$G = (V, E) \qquad |V| = n \qquad |E| = e$

$T :=$ (any) vertex spanning tree of G

T ( ) ((( )) ( ) ( )) ( ) (( )) ( ) ( )

parenthesis word of size $2n$

$G \setminus T$ [[[[ ]]] [[[[ ]] [ ]] [[[[[[ ]] [ ]]] [[ ]]] [ ]]]]

parenthesis word of size $2n$

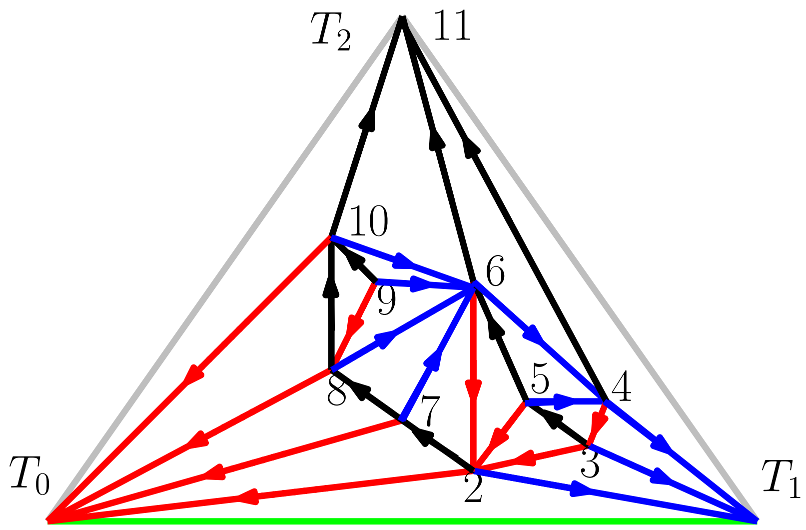
$length(S) = 2e$ symbols

$S(G)$ ([[[ ) ( ] ( ] [[[ )]]) ( ]] [ ) . . .

$(2 \log_2 4)e = 4e = 12n$ bits

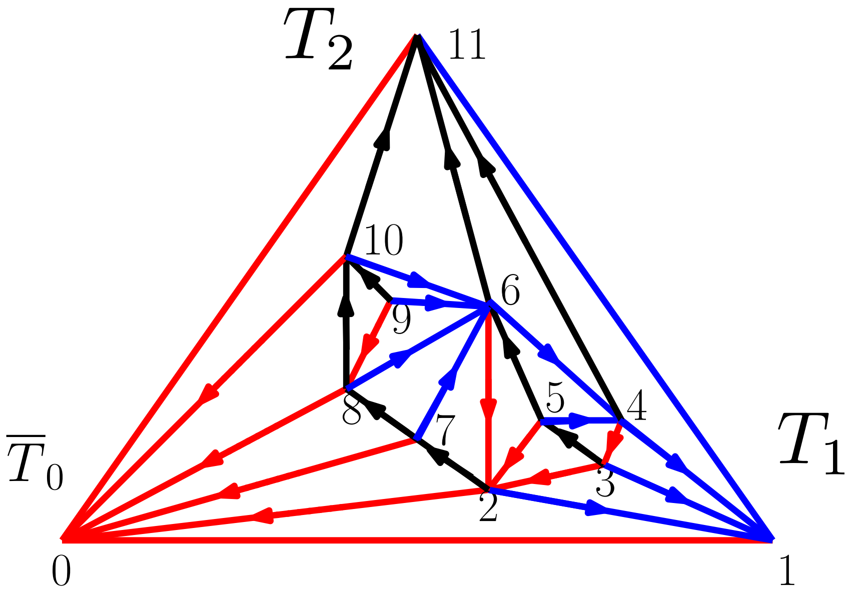
A more efficient encoding

Canonical orderings - Schnyder woods (He, Kao, Lu '99)



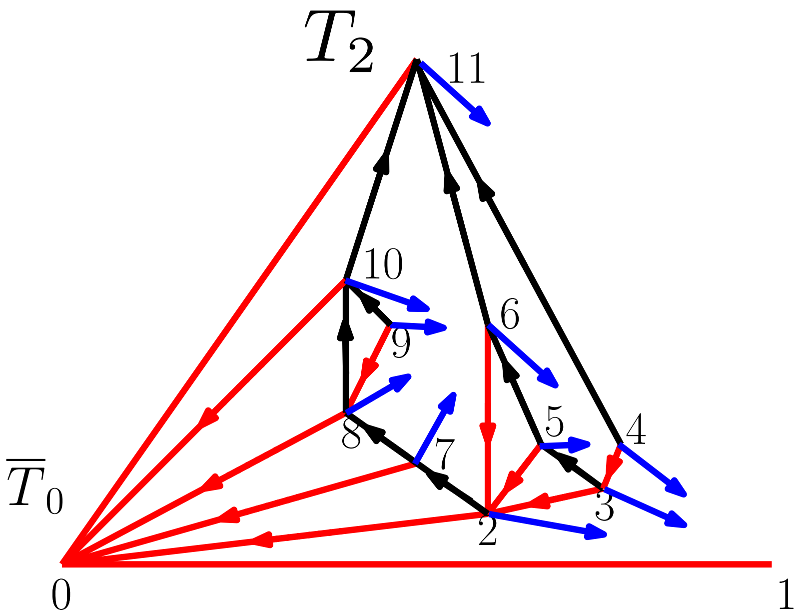
A more efficient encoding

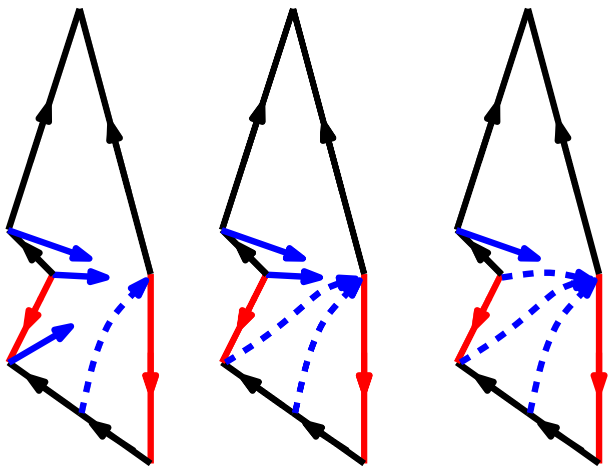
Canonical orderings - Schnyder woods (He, Kao, Lu '99)



A more efficient encoding

Canonical orderings - Schnyder woods (He, Kao, Lu '99)

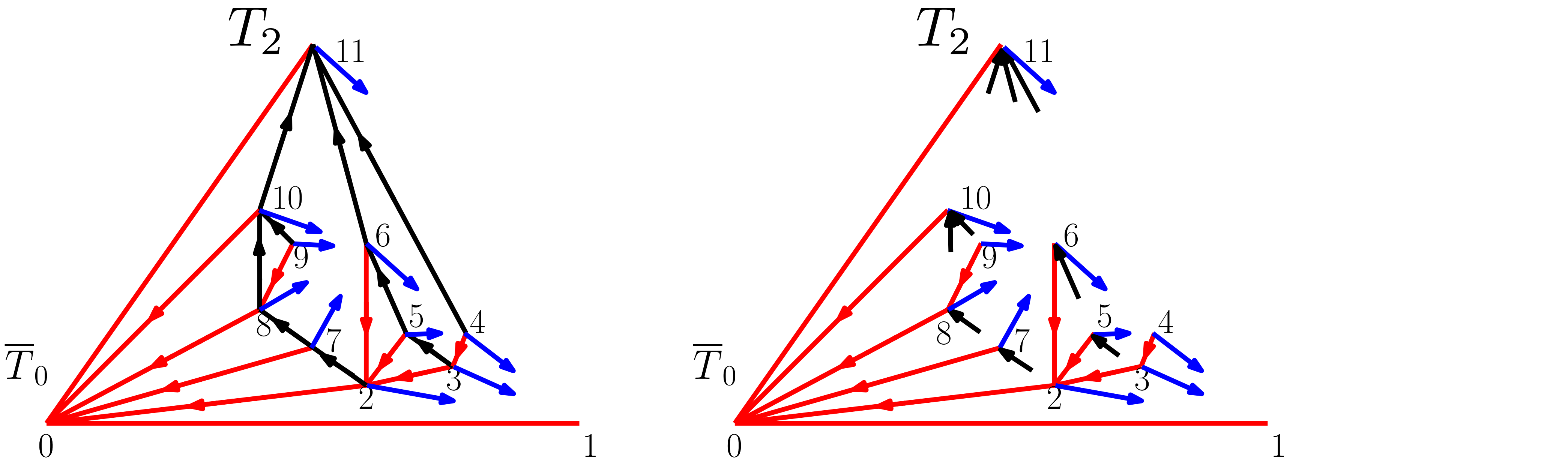




T_1 is redundant: reconstruct from T_0, T_2

A more efficient encoding

Canonical orderings - Schnyder woods (He, Kao, Lu '99)



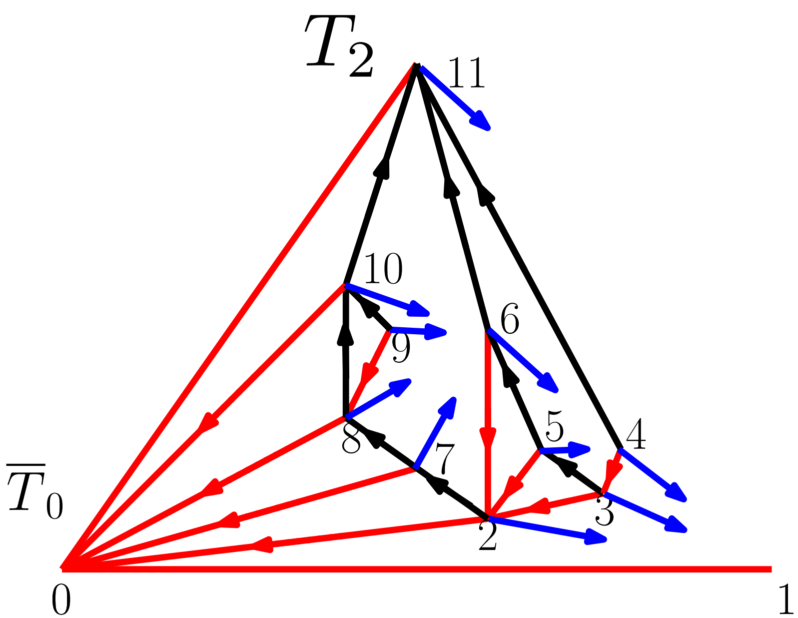
T_1 is redundant: reconstruct from T_0, T_2

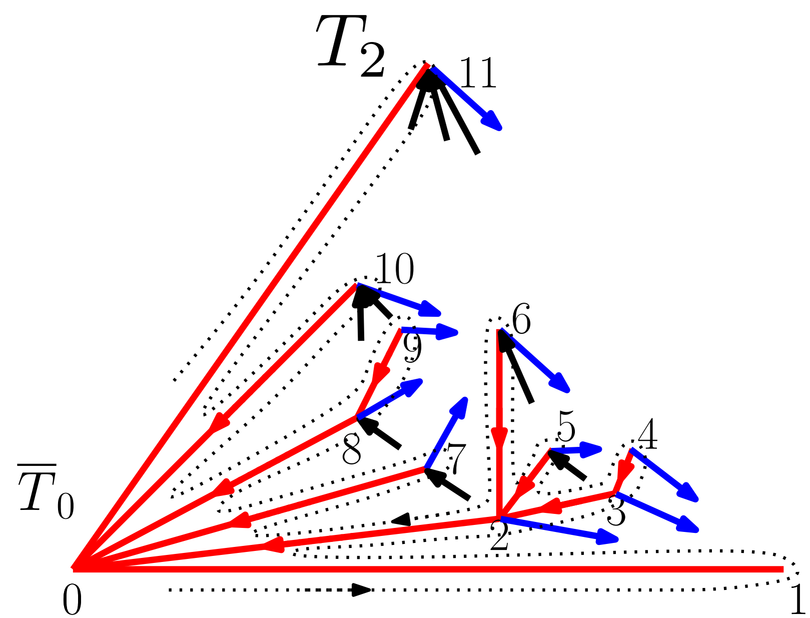
T_2 can be reconstructed from T_0 and the number of ingoing edges (for each node)

A more efficient encoding

Canonical orderings - Schnyder woods (He, Kao, Lu '99)

$4n$ bits (for triangulations)





$\overline{T}_0$ ( ) ((( )) ( ) ( )) ( ) (( )) ( ) ( )

$2(n-1)$ symbols$= 2(n-1)$ bits

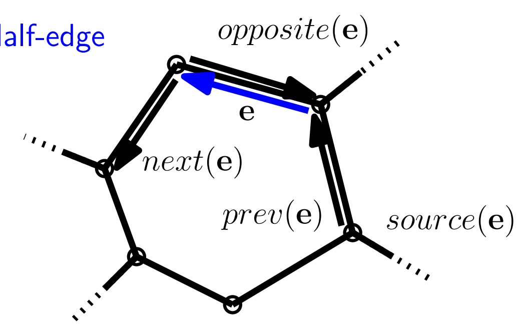
$\overline{T}_2$ 00000101010100110111

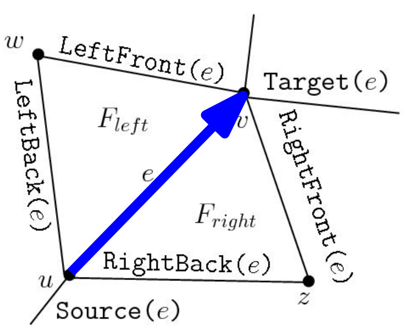
$(n-1)+(n-3) = 2n-4$ bits

Compact (practical) mesh data structures

	Data Structure	size	navigation time	vertex access	dynamic
(non compact) data structures	Half-edge/Winged-edge/Quad-edge	$18n + n$	$O(1)$	$O(1)$	*yes*
	Triangle based DS / Corner Table	$12n + n$	$O(1)$	$O(1)$	*yes*
compact data structures	Directed edge (Campagna et al. '99)	$12n + n$	$O(1)$	$O(1)$	*yes*
	2D Catalogs (Castelli Aleardi et al., '06)	$7.67n$	$O(1)$	$O(1)$	*yes*
	Star vertices (Kallmann et al. '02)	$7n$	$O(d)$	$O(1)$	*no*
	TRIPOD (Snoeyink, Speckmann, '99)	$6n$	$O(1)$	$O(d)$	*no*
	SOT (Gurung et al. 2010)	$6n$	$O(1)$	$O(d)$	*no*
	SQUAD (Gurung et al. 2011) ε between 0.09 and 0.3	$(4 + \varepsilon)n$	$O(1)$	$O(d)$	*no*
	ESQ (Castelli Aleardi, Devillers, Rossignac'12)	$4.8n$	$O(1)$	$O(d)$	*yes*
	Castelli Aleardi and Devillers (2011)	$4n$ (or $6n$)	$O(1)$	$O(d)$ (or $O(1)$)	*no*
	LR (Gurung et al. 2011) δ about 0.8 and 0.3	$(2 + \delta)n$	$O(1)$	$O(d)$	*no*

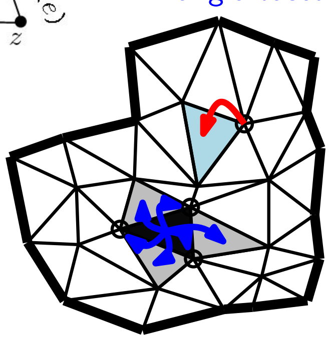
Half-edge

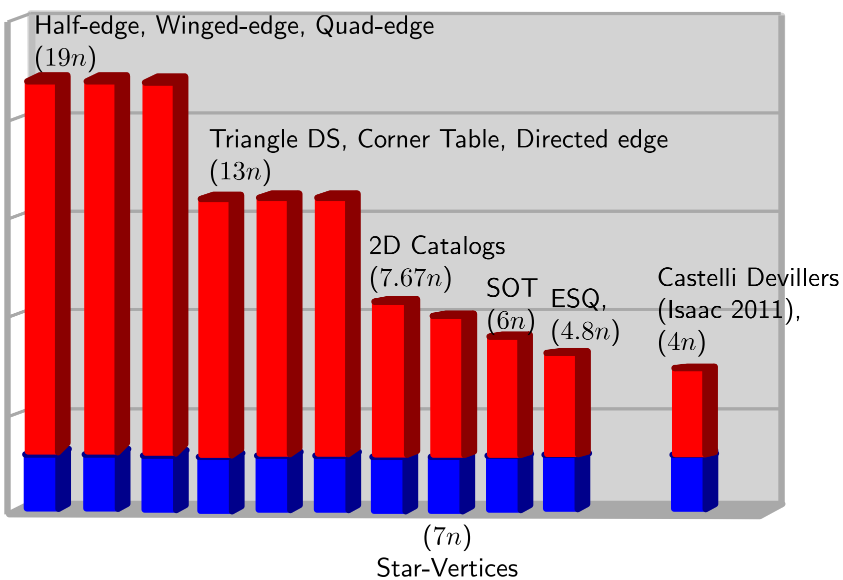




Winged-edge

Triangle-based

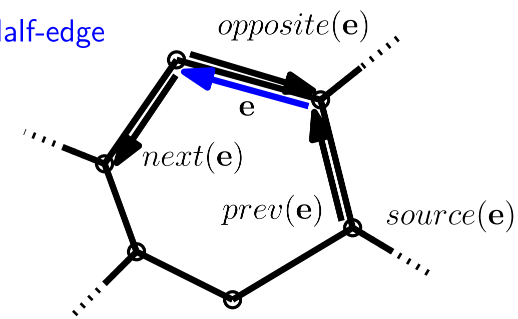


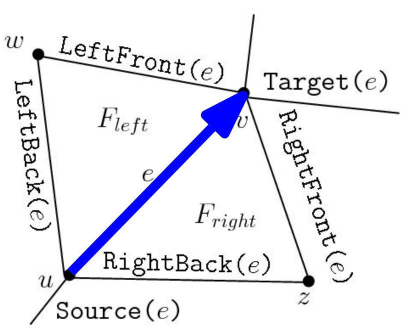


Compact (practical) mesh data structures

	Data Structure	size	navigation time	vertex access	dynamic
(non compact) data structures	Half-edge/Winged-edge/Quad-edge	$18n + n$	$O(1)$	$O(1)$	*yes*
	Triangle based DS / Corner Table	$12n + n$	$O(1)$	$O(1)$	*yes*
compact data structures	Directed edge (Campagna et al. '99)	$12n + n$	$O(1)$	$O(1)$	*yes*
	2D Catalogs (Castelli Aleardi et al., '06)	$7.67n$	$O(1)$	$O(1)$	*yes*
	Star vertices (Kallmann et al. '02)	$7n$	$O(d)$	$O(1)$	*no*
	TRIPOD (Snoeyink, Speckmann, '99)	$6n$	$O(1)$	$O(d)$	*no*
	SOT (Gurung et al. 2010)	$6n$	$O(1)$	$O(d)$	*no*
	SQUAD (Gurung et al. 2011) (ε between 0.09 and 0.3)	$(4 + \varepsilon)n$	$O(1)$	$O(d)$	*no*
	ESQ (Castelli Aleardi, Devillers, Rossignac'12)	$4.8n$	$O(1)$	$O(d)$	*yes*
	Castelli Aleardi and Devillers (2011)	$4n$ (or $6n$)	$O(1)$	$O(d)$ (or $O(1)$)	*no*
	LR (Gurung et al. 2011) (δ about 0.8 and 0.3)	$(2+\delta)n$	$O(1)$	$O(d)$	*no*

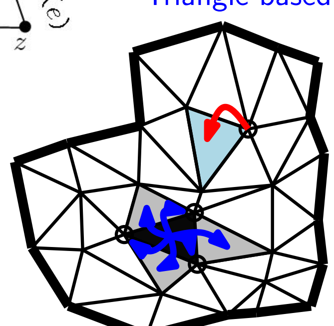
Half-edge

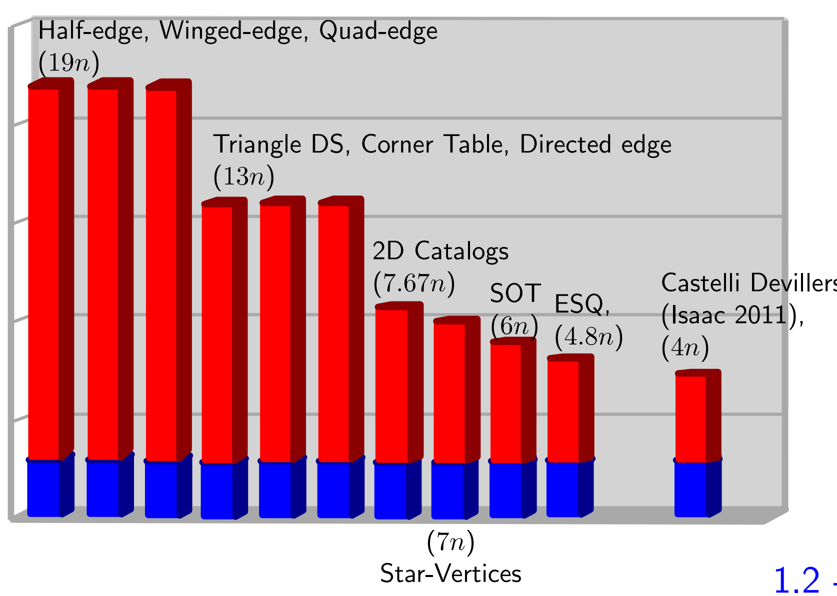


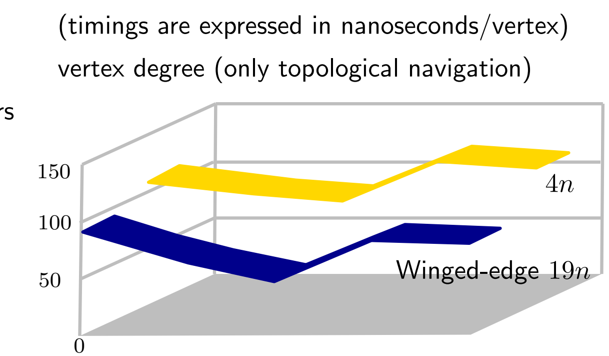


Winged-edge

Triangle-based



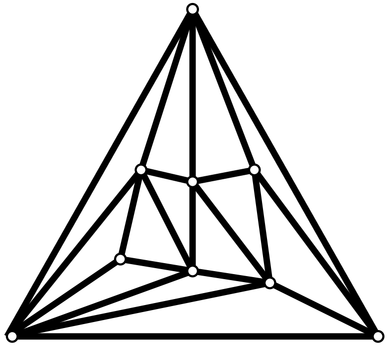




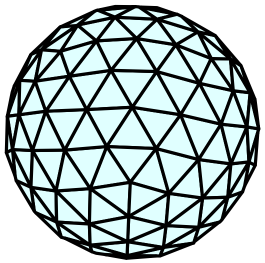
1.2 - 1.9 times slower than Winged-edge
(experimental evaluation)

Graphs on surfaces

Graphs on surfaces

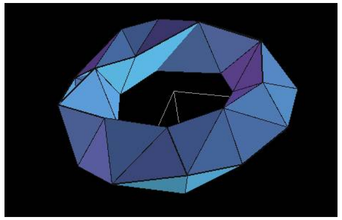


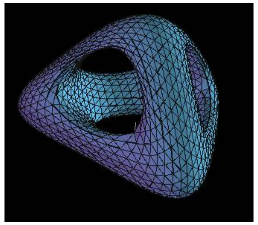
$$e = 3n - 6$$



$$n - e + f = 2 - 2g$$

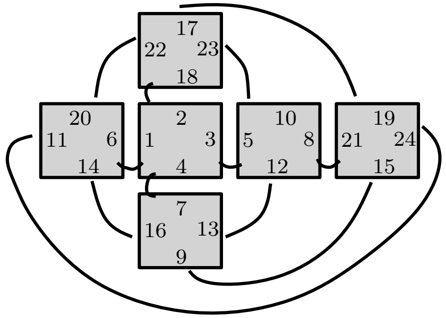
$g = 1 \qquad e = 3n$

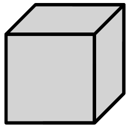


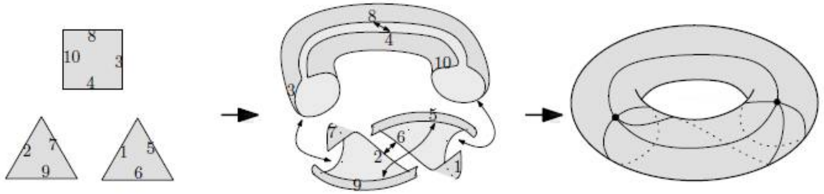


$\phi = (1,2,3,4)(17,23,18,22)(5,10,8,12)(21,19,24,15)\ldots$

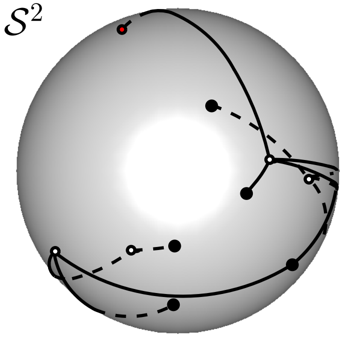
$\alpha = (2,18)(4,7)(12,13)(9,15)(14,16)(10,23)\ldots$

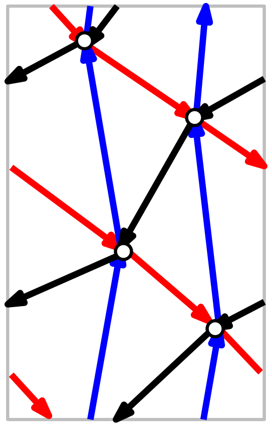


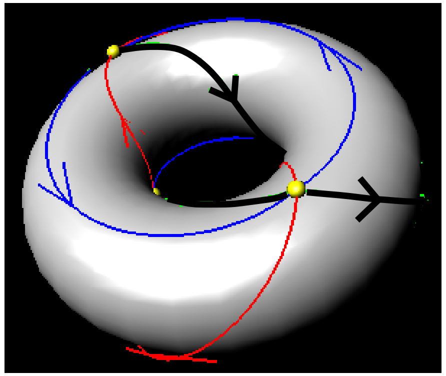




$\mathcal{S}^2$

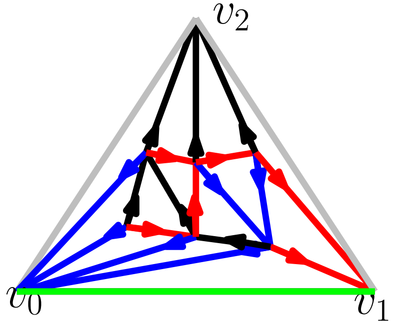


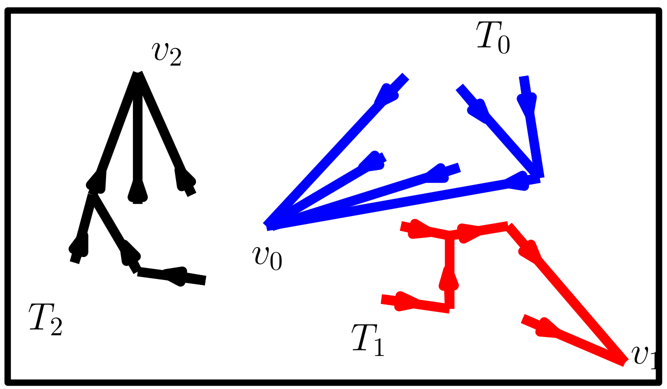


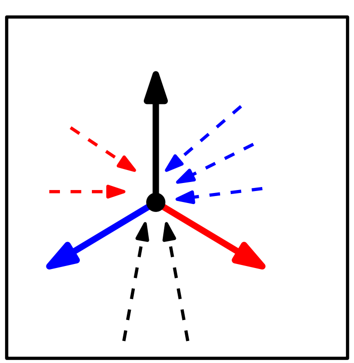


what can we to extend to higher genus?

$$e = 3n - 6$$

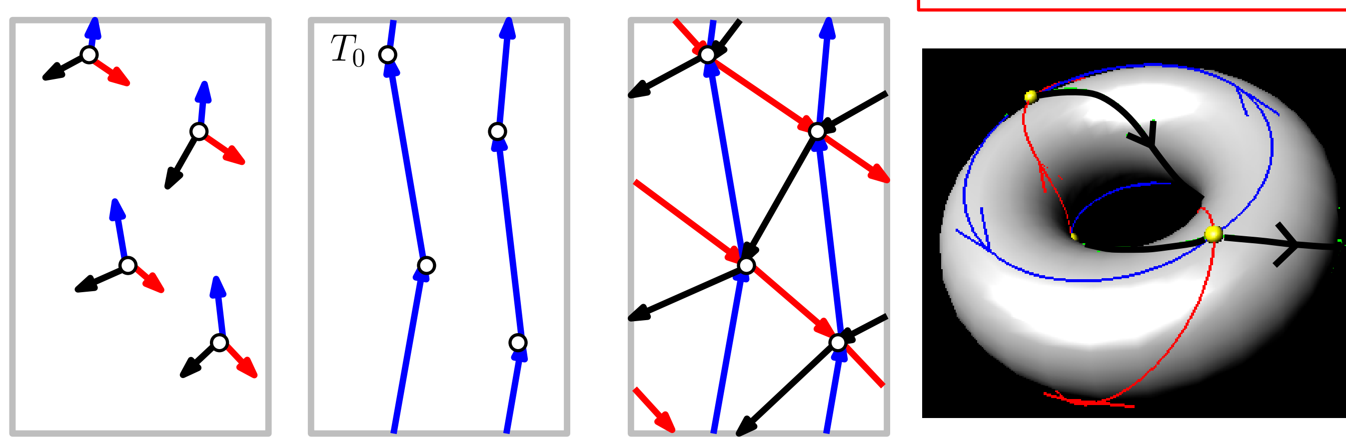






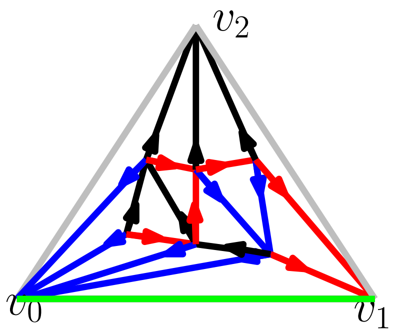
[Goncalves Lévêque, DCG'14]

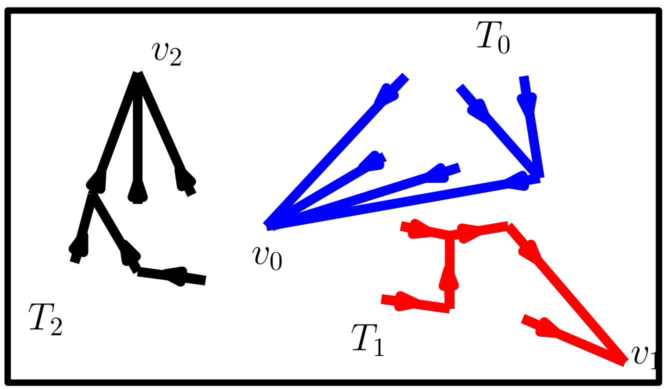
$g = 1$ $\quad e = 3n$



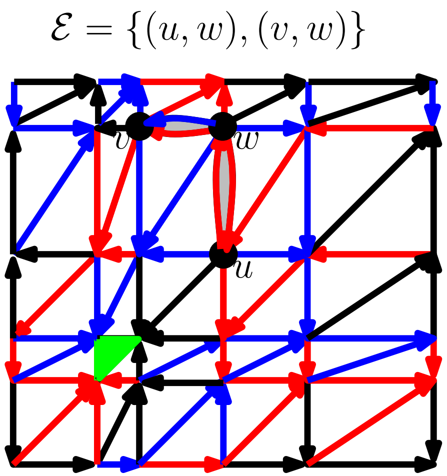
what can we to extend to higher genus?

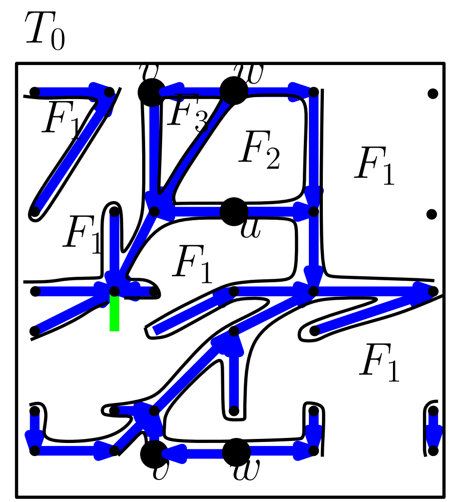
$$e = 3n - 6$$

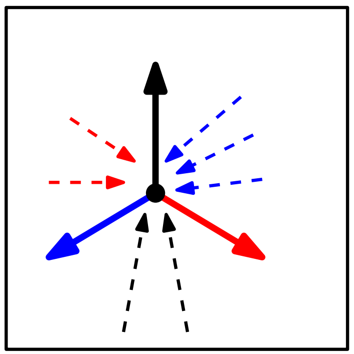




[Castelli-Aleardi Fusy Lewiner, SoCG'08]

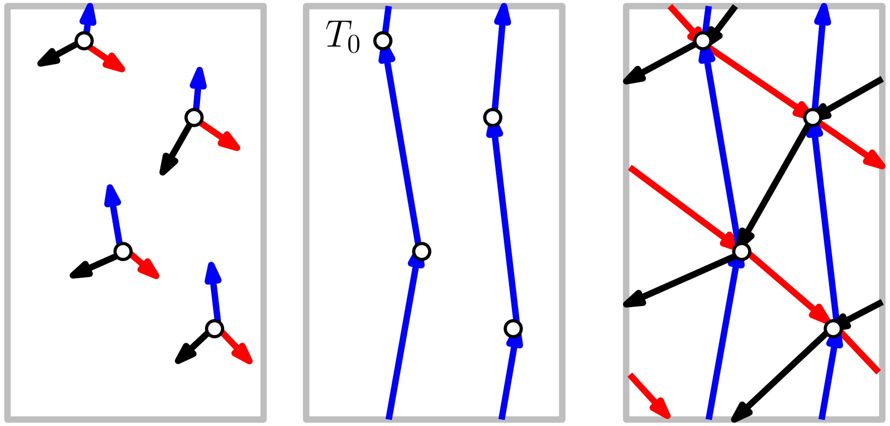


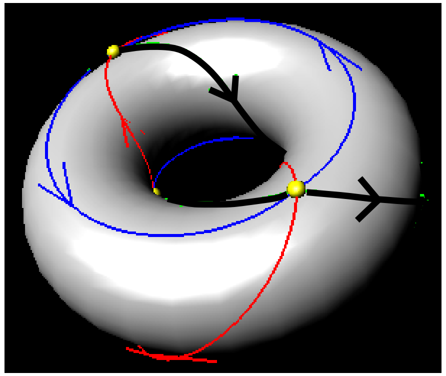




[Goncalves Lévêque, DCG'14]

$g = 1 \qquad e = 3n$





Schnyder woods and higher genus surfaces

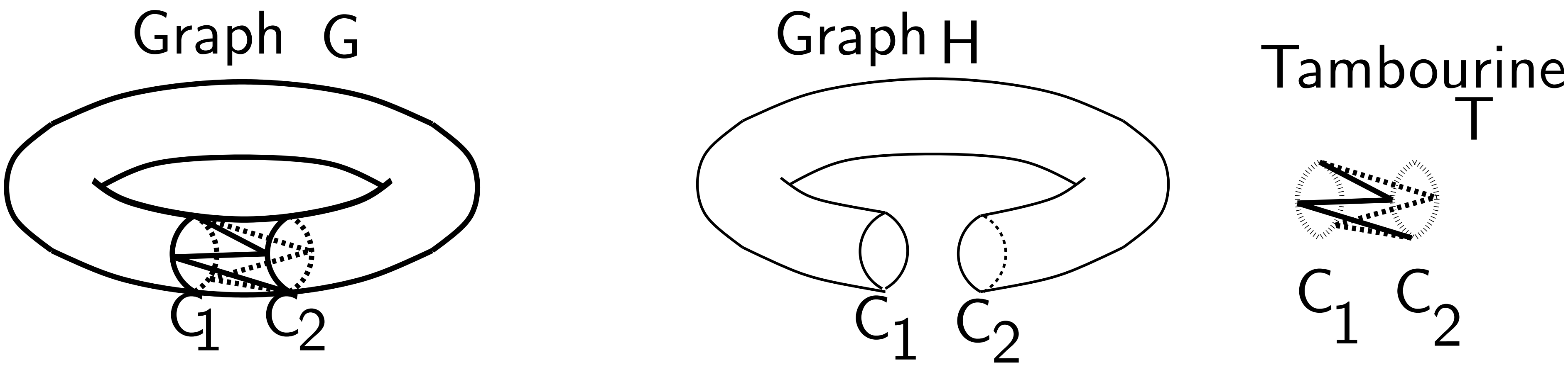
(several possible generalizations)

(pioneeristic) toroidal tree decomposition

[Bonichon Gavoille Labourel, 2005]

the "tambourine" solution

Compute a pair of adjacent non contractible cycles



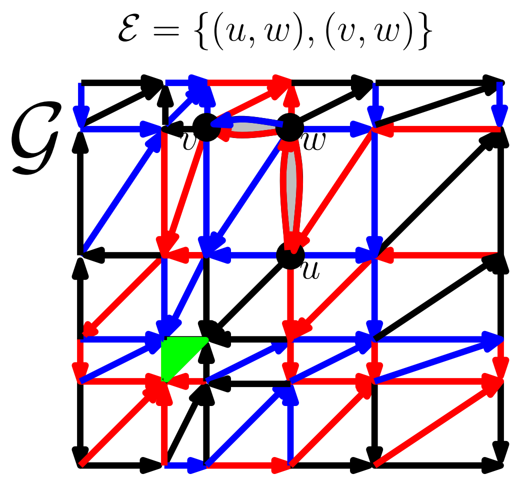
Result:

Inconvenients:

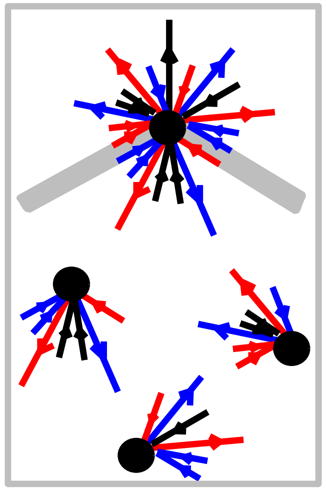
- valid only for toroidal triangulations (genus 1)
- potentially large number of vertices (on C_1 and C_2) not satisfying the local condition
- shortest non trivial cycles are "hard" to compute

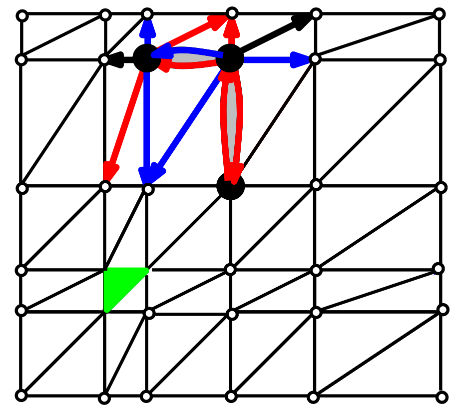
Definition I: genus g Schnyder woods

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



local condition for multiple vertices





Def: partition of all "inner" edges into four sets

T_0, T_1, T_2 and $\mathcal{E}$

such that

almost all vertices have outgoing degree 3

all edges in T_0, T_1 and T_2 have one color/orientation

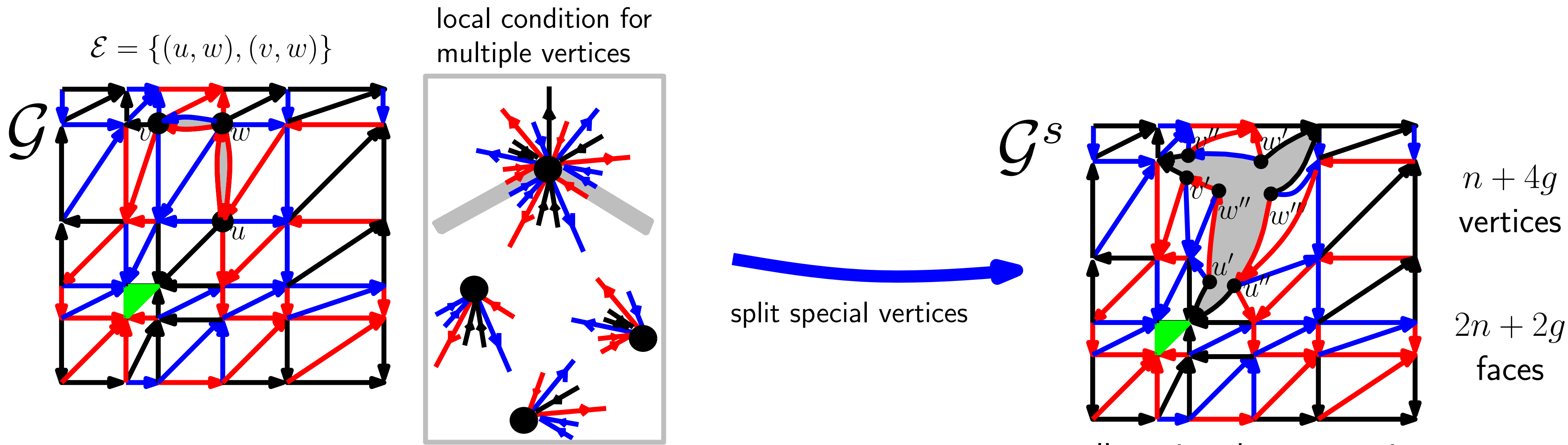
at most $4g$ special vertices (outdegree > 3)

the set $\mathcal{E}$ contains at most $2g$ edges (multiple edges)

new local conditions around special vertices

Definition I: genus g Schnyder woods

[Castelli-Aleardi Fusy Lewiner, SoCG'08]



all vertices have outgoing degree at most 3

Def: partition of all "inner" edges into four sets

T_0, T_1, T_2 and $\mathcal{E}$

such that

almost all vertices have outgoing degree 3

all edges in T_0, T_1 and T_2 have one color/orientation

at most $4g$ special vertices (outdegree > 3)

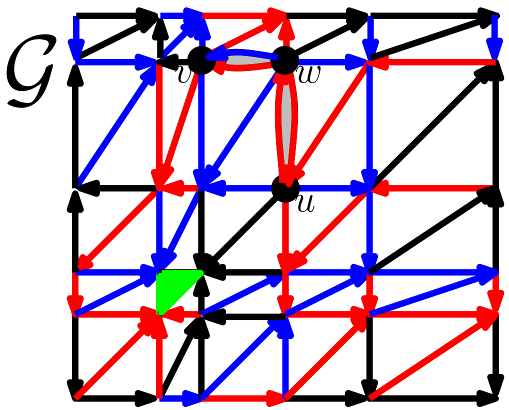
the set $\mathcal{E}$ contains at most $2g$ edges (multiple edges)

new local conditions around special vertices

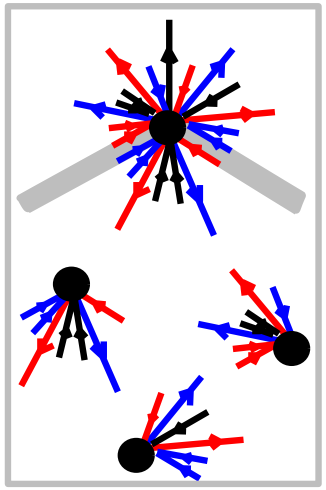
Genus g Schnyder woods: spanning property

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

$\mathcal{E} = \{(u,w),(v,w)\}$

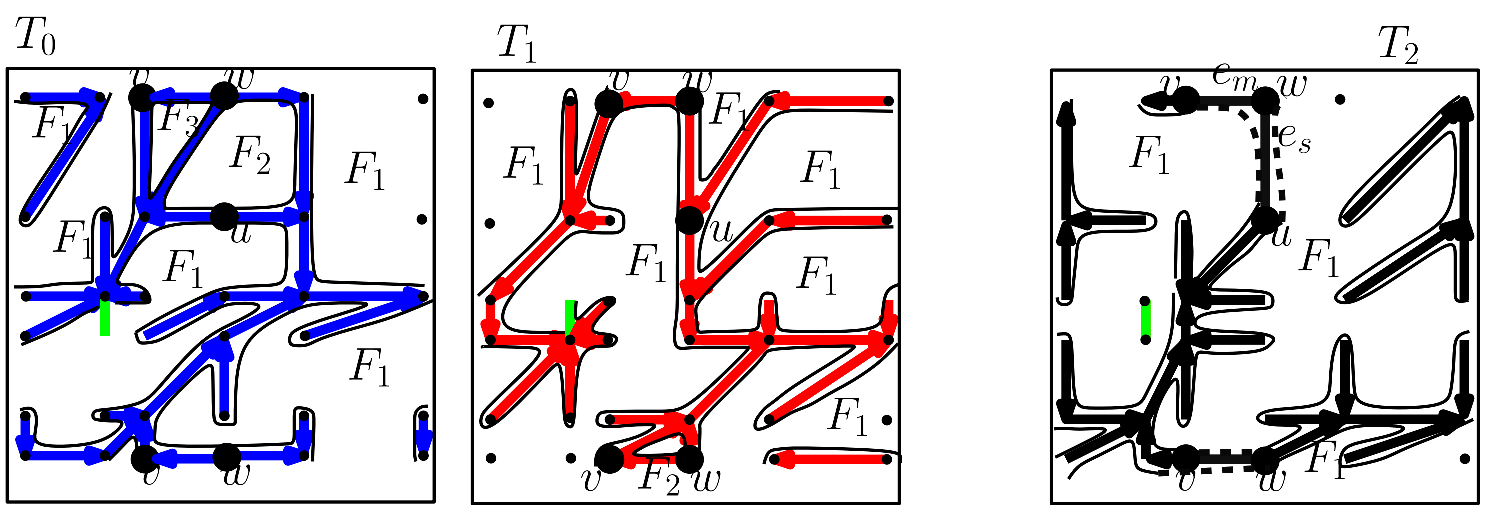


local condition for multiple vertices



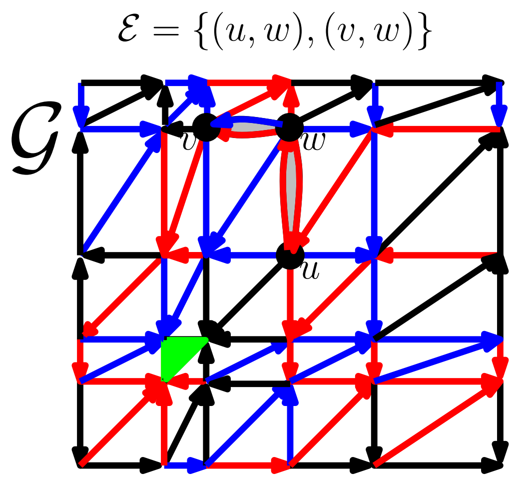
Theorem
The three sets of edges T_0 and T_1 (red and blue edges), as well as the set $T_2 \cup \mathcal{E}$ (black edges and special edges) are maps of genus g satisfying:

- T_0, T_1 are maps with at most $1 + 2g$ faces;
- $T_2 \cup \mathcal{E}$ is a 1 face map (a g-tree)

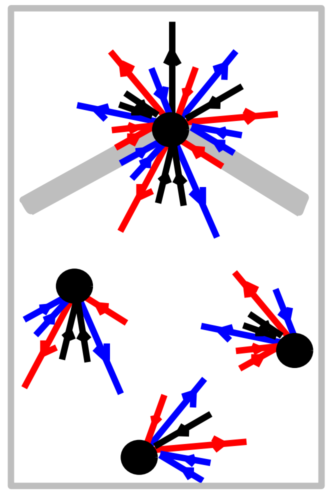


Genus g Schnyder woods: application

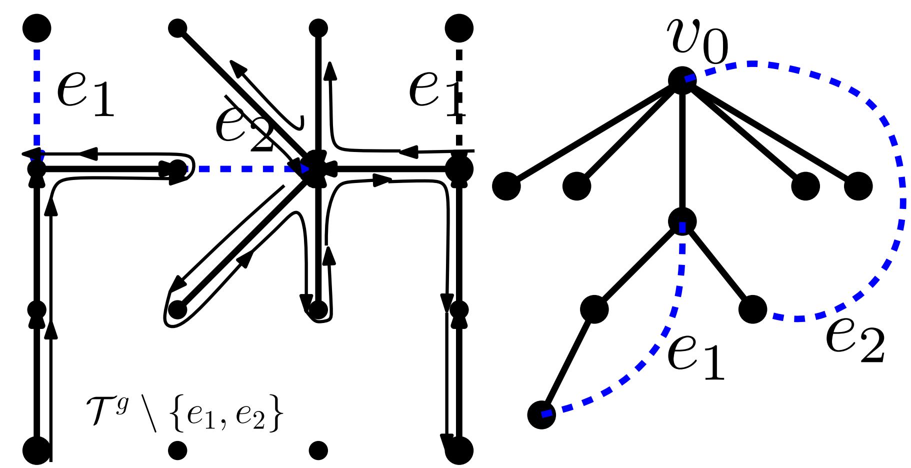
[Castelli-Aleardi Fusy Lewiner, SoCG'08]



local condition for multiple vertices



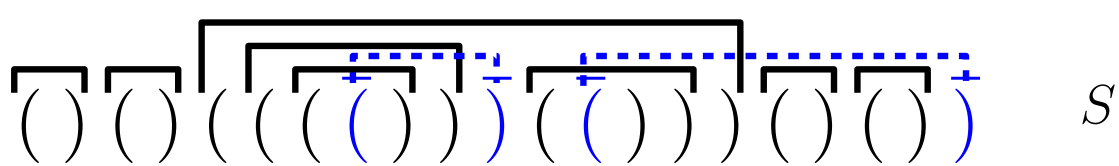
Corollary
A triangulation of genu g having n vertices can be encoded with $4n + O(g \log n)$ bits



Encode map $T_2 \cup \mathcal{E}$: a tree plus $2g$ edges: $2n + O(g \log n)$ bits

Mark special vertices: $O(g \log n)$ bits

Store outgoing edges incident to special edges: $O(g \log n)$ bits

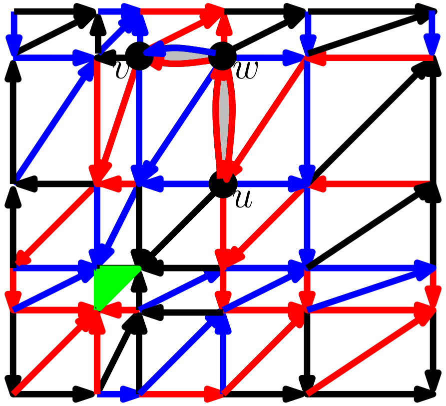


For each node in $T_2 \cup \mathcal{E}$ store the number of ingoing edges of color 0: $2n + O(g \log n)$ bits

Genus g Schnyder woods: existence

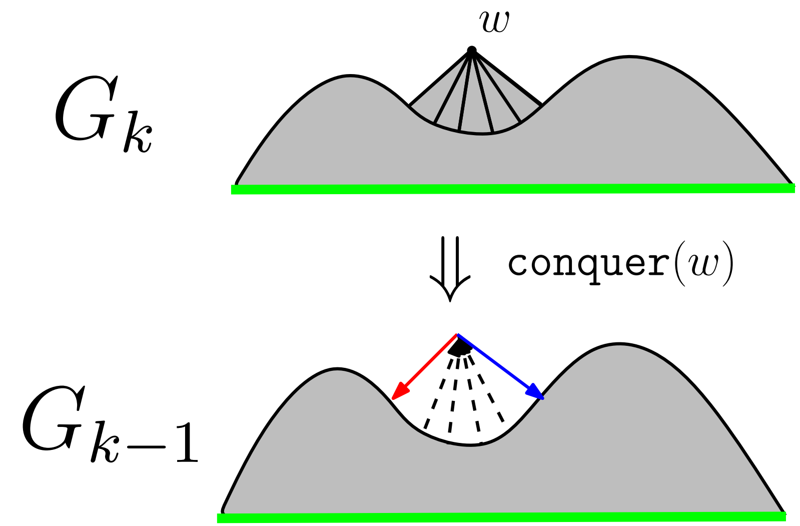
[Castelli-Aleardi Fusy Lewiner, SoCG'08]

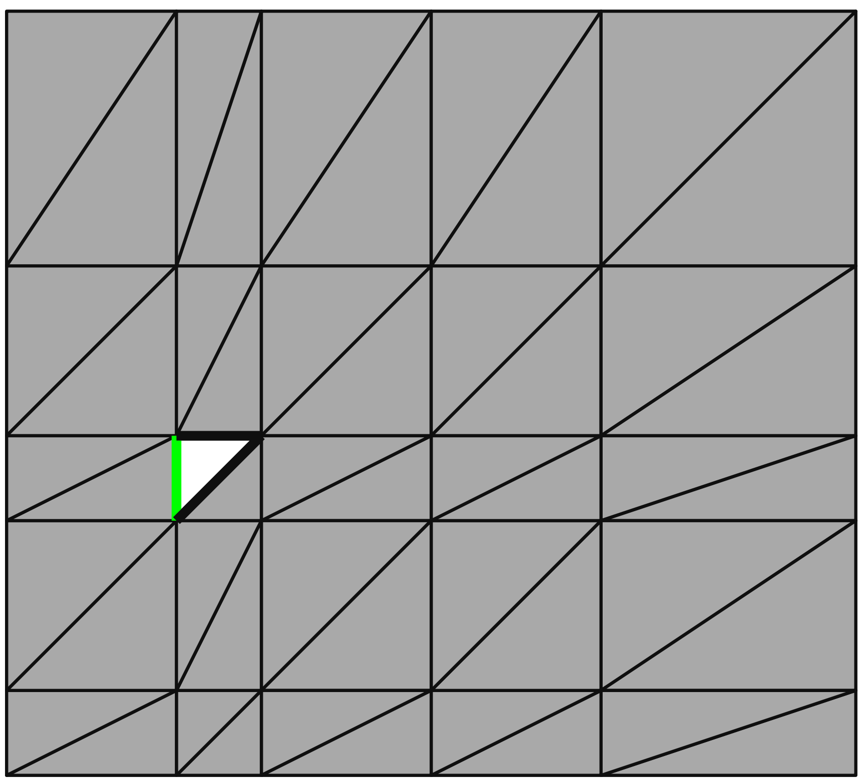
$\mathcal{E} = \{(u, w), (v, w)\}$



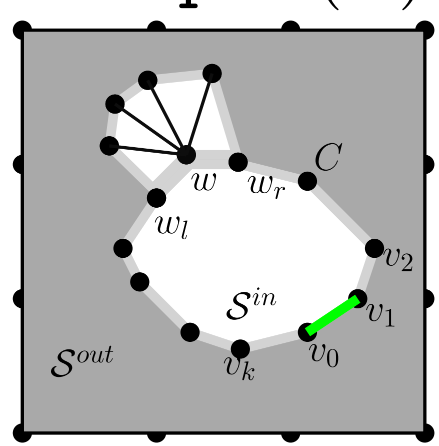
Incremental algorithm

Perform a vertex conquest (as far as you can)





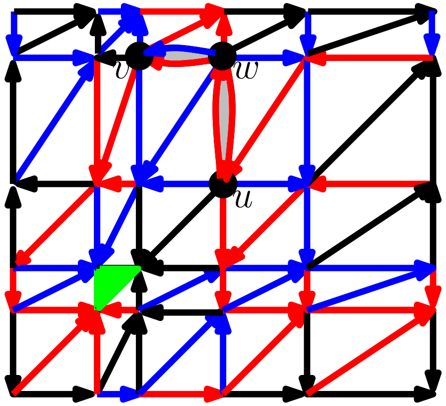
conquer(w)



Genus g Schnyder woods: existence

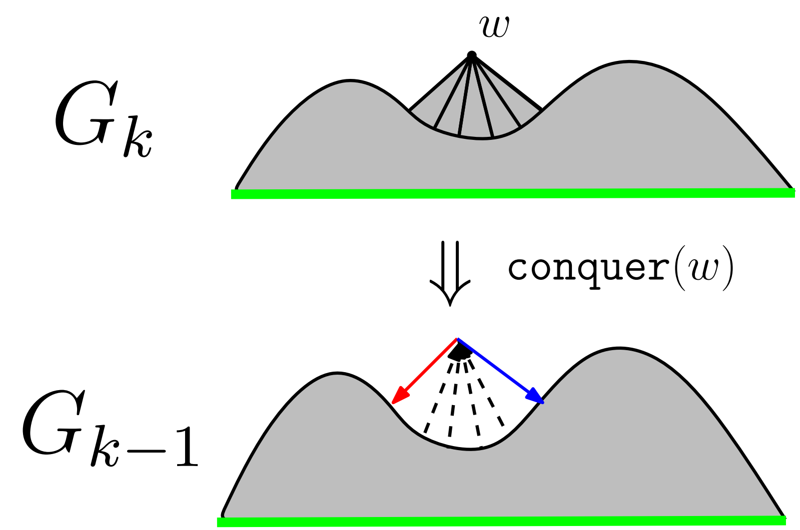
[Castelli-Aleardi Fusy Lewiner, SoCG'08]

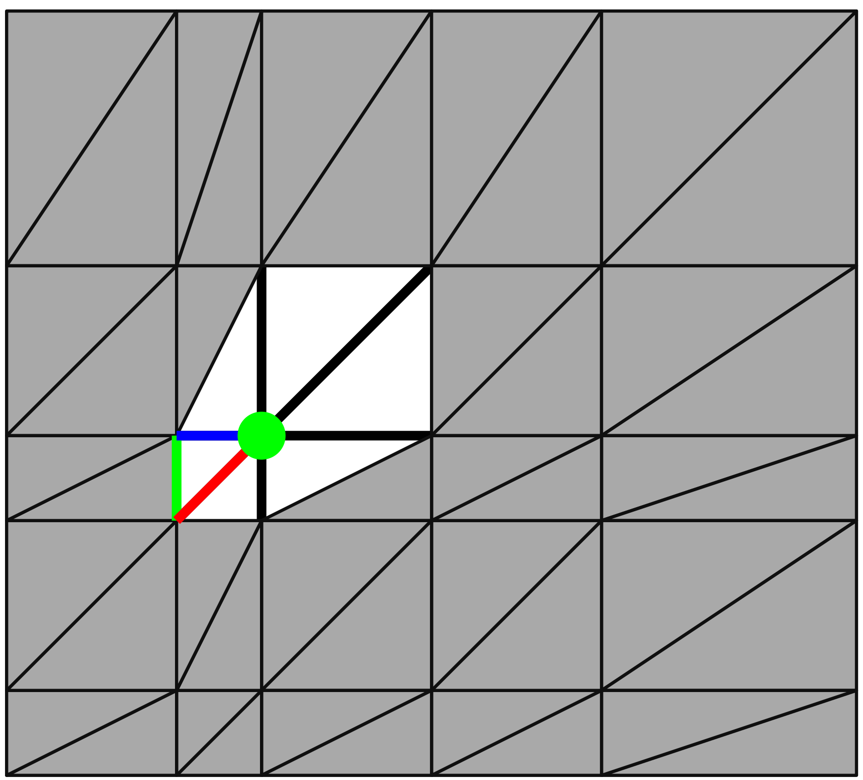
$\mathcal{E} = \{(u,w),(v,w)\}$



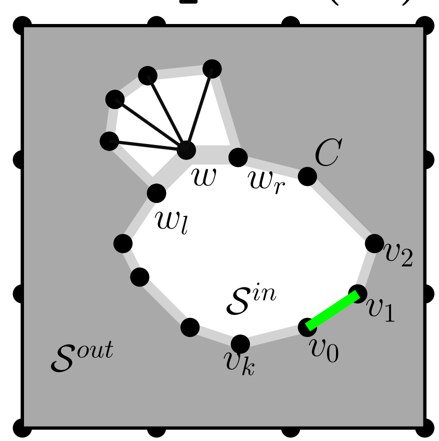
Incremental algorithm

Perform a vertex conquest (as far as you can)





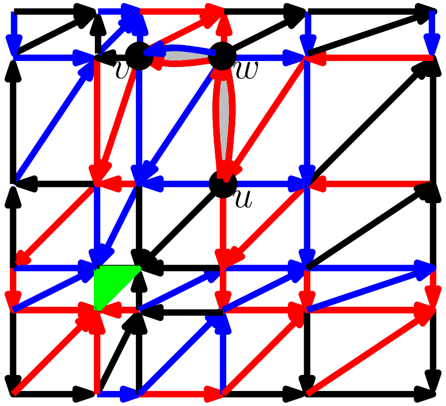
conquer(w)



Genus g Schnyder woods: existence

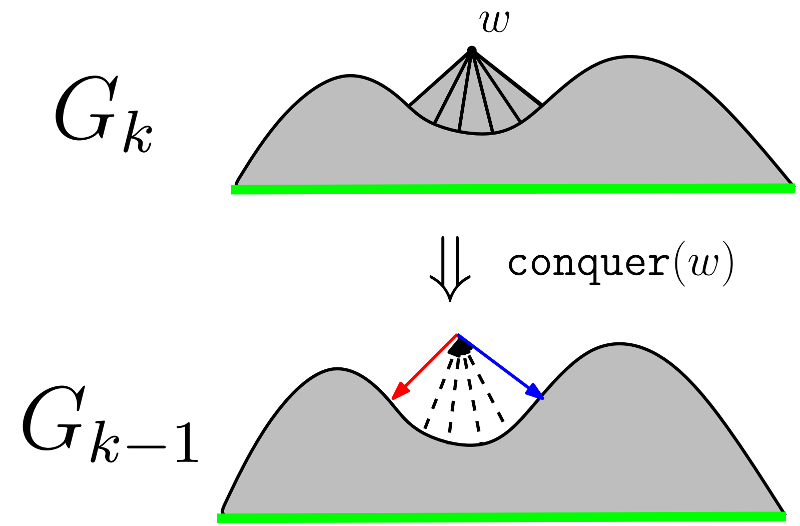
[Castelli-Aleardi Fusy Lewiner, SoCG'08]

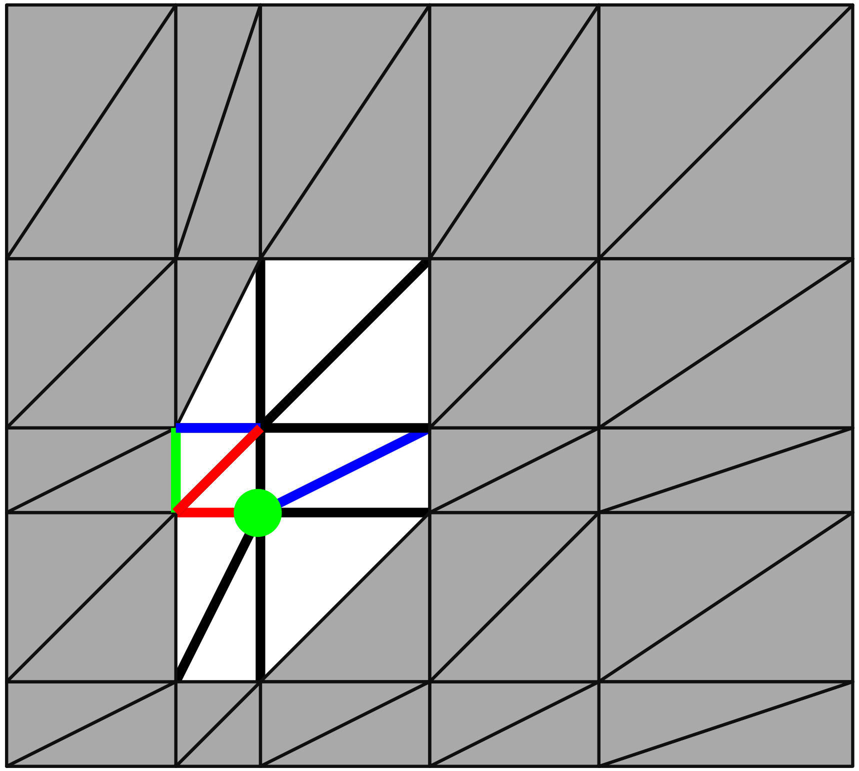
$\mathcal{E} = \{(u,w),(v,w)\}$



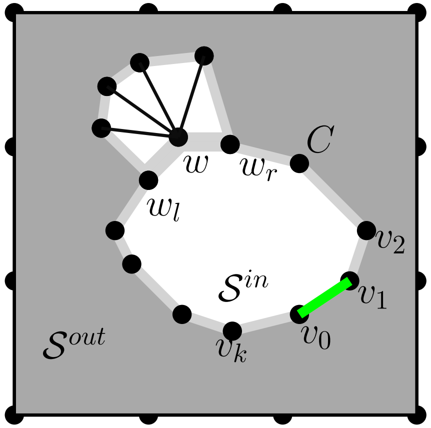
Incremental algorithm

Perform a vertex conquest (as far as you can)





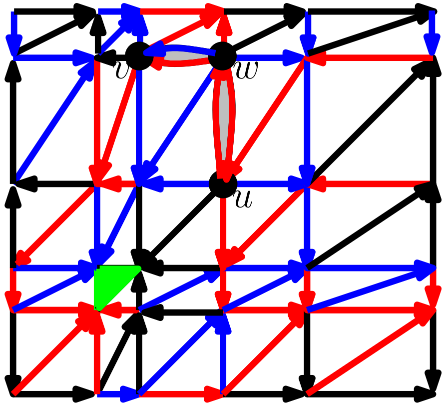
conquer(w)



Genus g Schnyder woods: existence

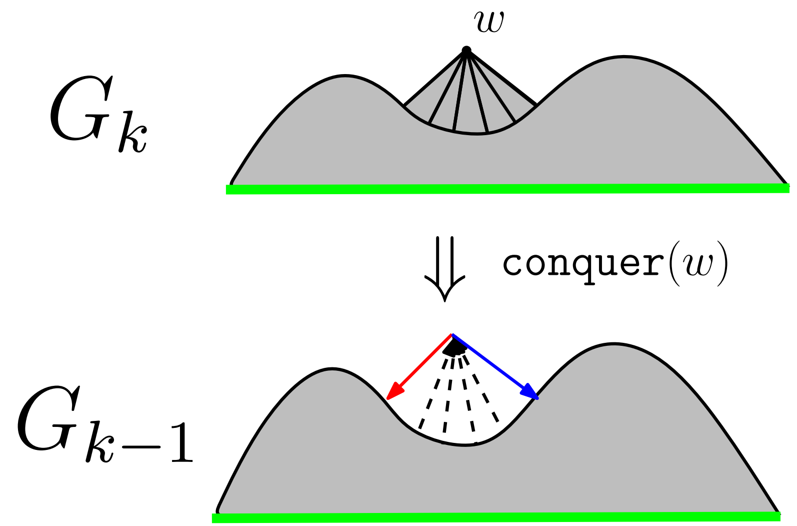
[Castelli-Aleardi Fusy Lewiner, SoCG'08]

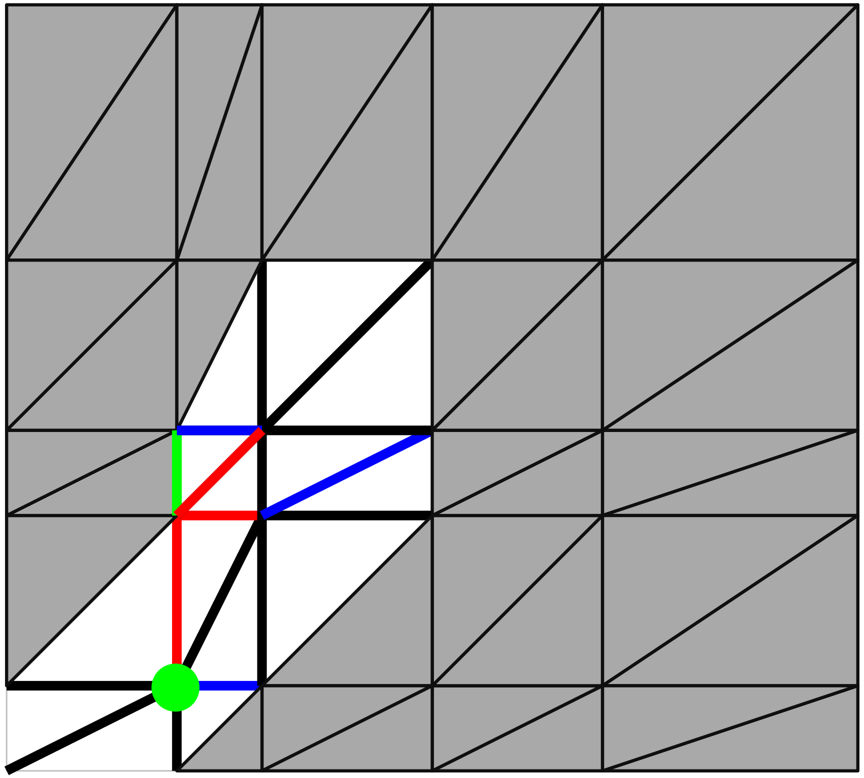
$\mathcal{E} = \{(u,w),(v,w)\}$



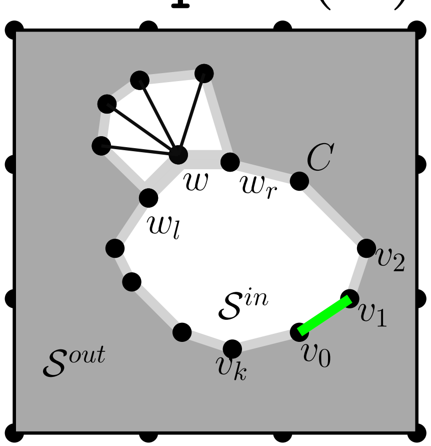
Incremental algorithm

Perform a vertex conquest (as far as you can)





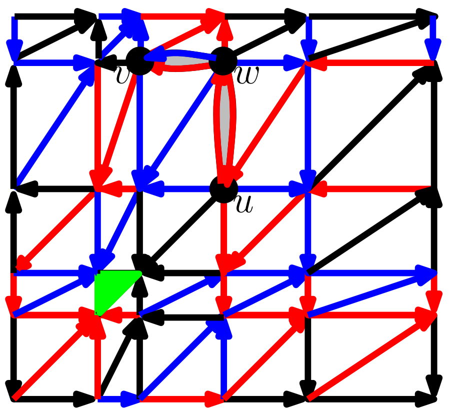
conquer(w)



Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

$\mathcal{E} = \{(u,w),(v,w)\}$

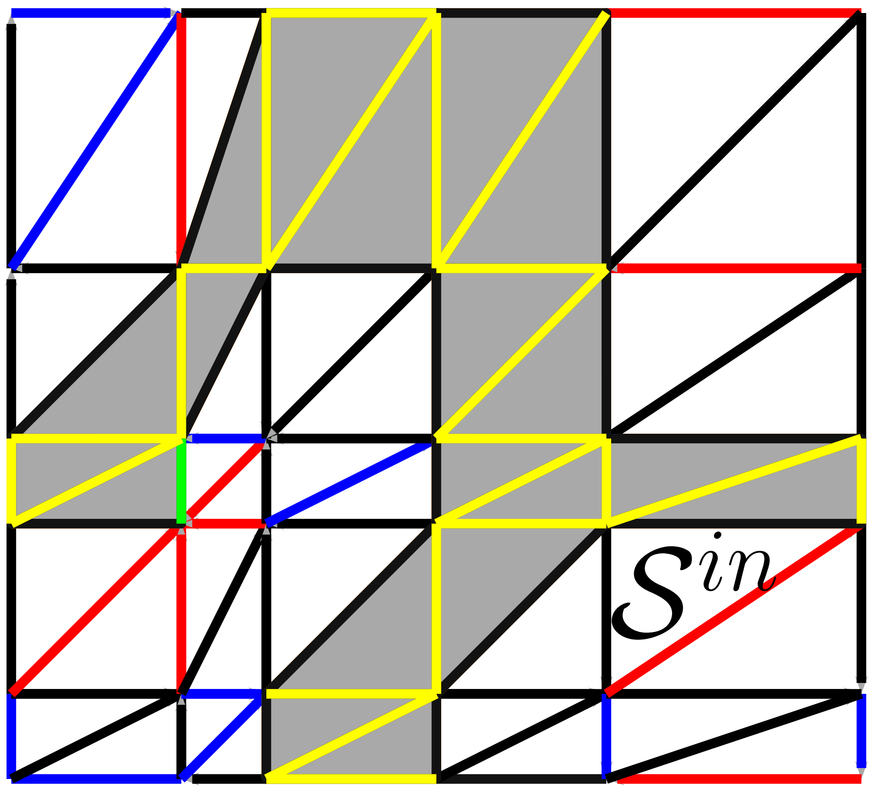


Incremental algorithm

Perform a vertex conquest (as far as you can)

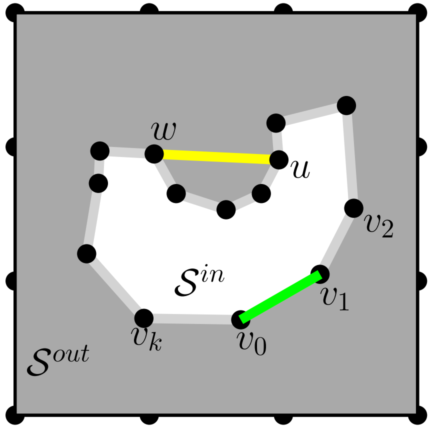
when you get stuck

$\mathcal{S}^{in}$ is a topological disk



No more free vertices

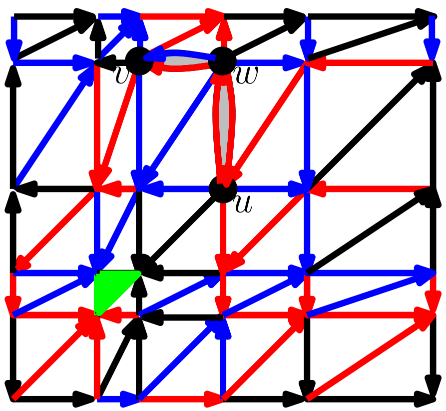
chordal edge (u,w)



Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

$\mathcal{E} = \{(u,w),(v,w)\}$



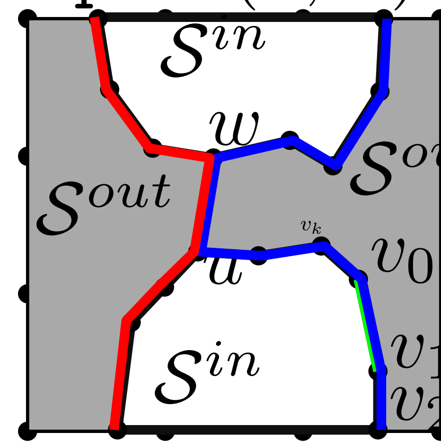
Incremental algorithm

Perform a vertex conquest (as far as you can)

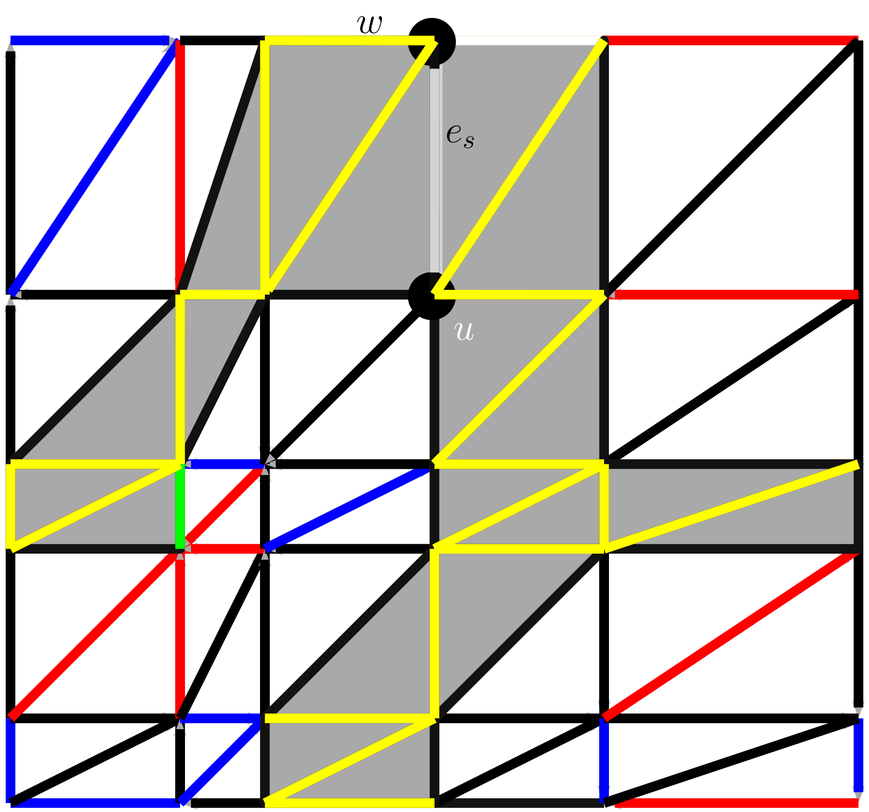
when you get stuck

perform edge split

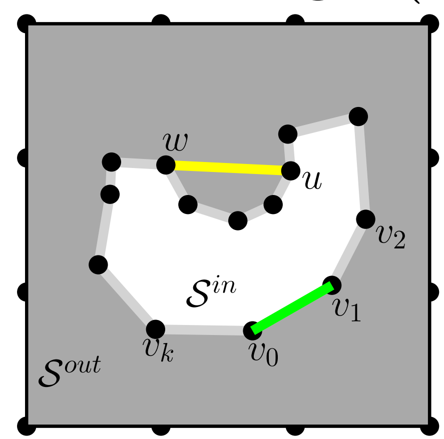
$\texttt{split}(u, w)$



$\mathcal{S}^{in}$ is a topological disk



chordal edge (u, w)

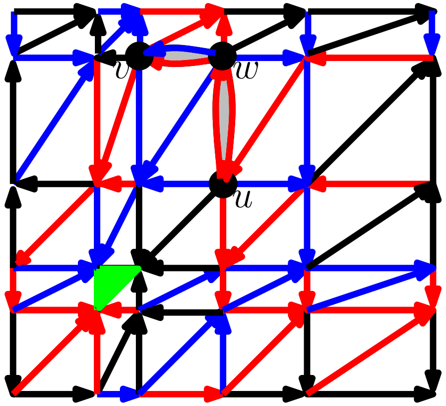


Now there are free vertices

Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

$\mathcal{E} = \{(u,w),(v,w)\}$



Incremental algorithm

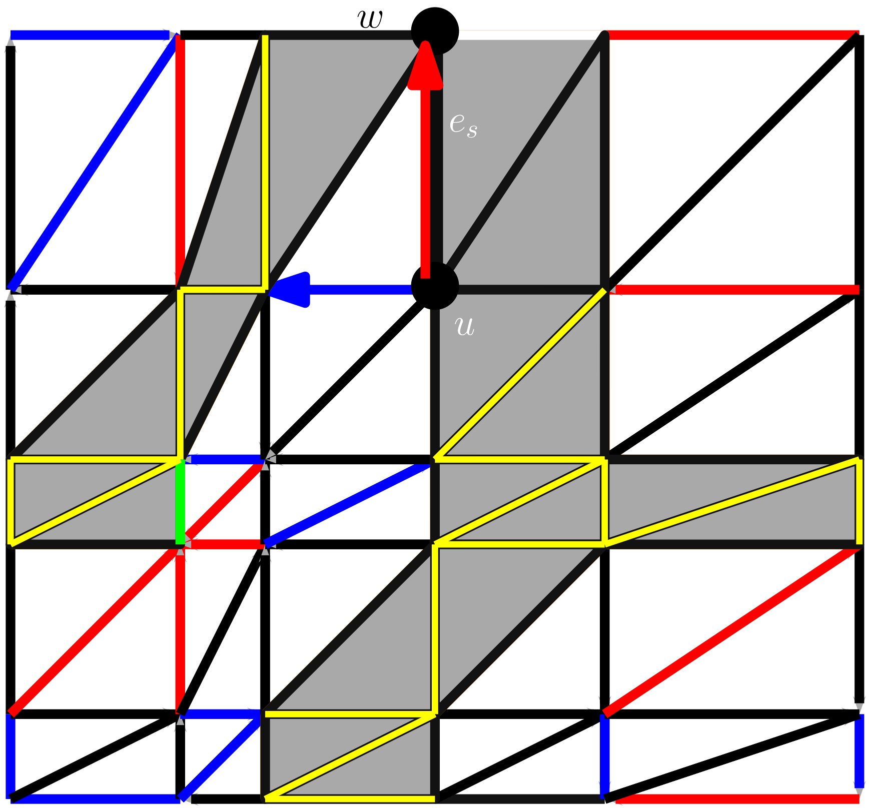
Perform a vertex conquest (as far as you can)

when you get stuck

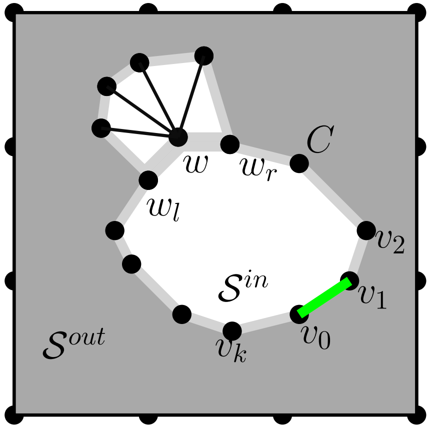
perform edge split

Perform a vertex conquest (as far as you can)

$\mathcal{S}^{in}$ is a topological disk



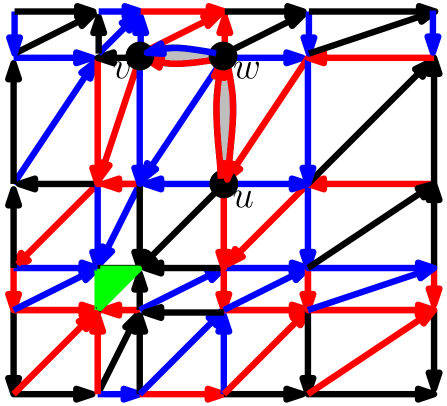
conquer(w)



Genus g Schnyder woods: existence

[Castelli-Aleardi Fusy Lewiner, SoCG'08]

$\mathcal{E} = \{(u,w),(v,w)\}$

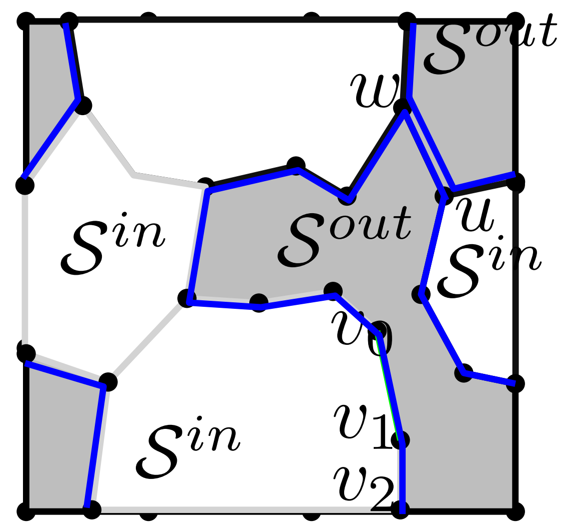


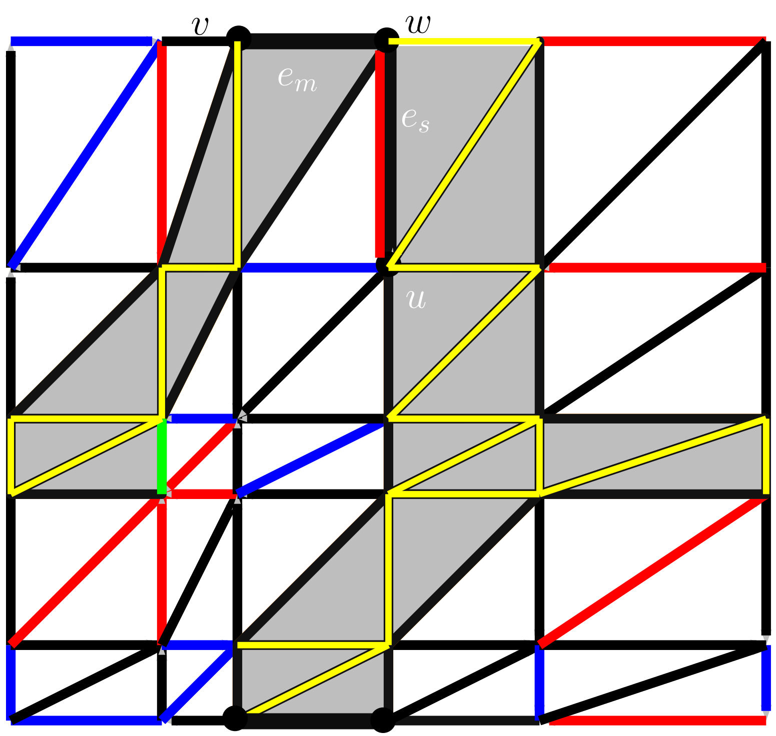
Incremental algorithm

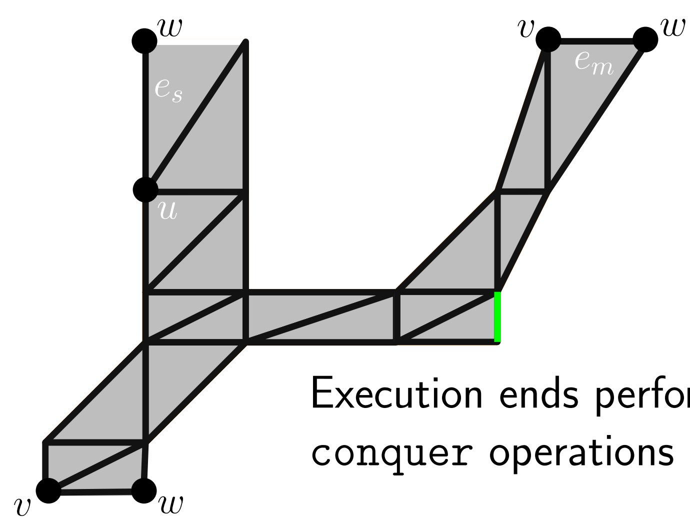
Perform a vertex conquest (as far as you can)
when you get stuck
    perform edge split
Perform a vertex conquest (as far as you can)
    perform edge split

$\mathcal{S}^{in}$ is a topological disk

$\texttt{merge}(u,w)$







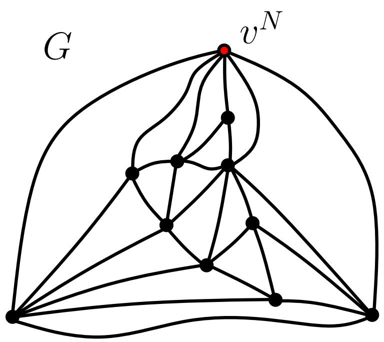
Execution ends performing a sequence of `conquer` operations

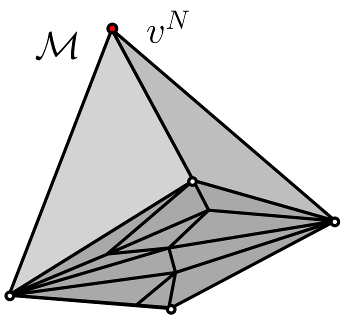
Periodic straight-line drawings

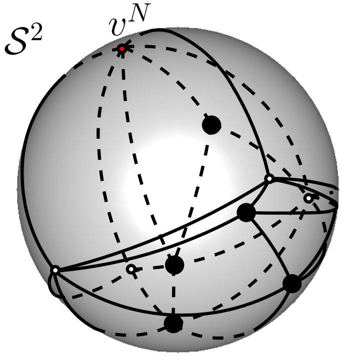
(of higher genus graphs)

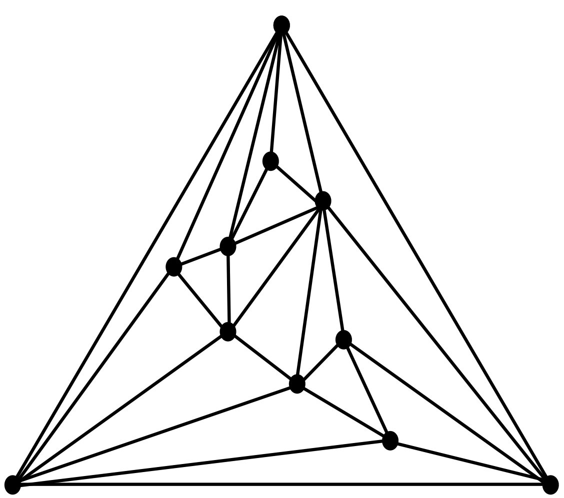
Drawing higher genus graphs

$g = 0$

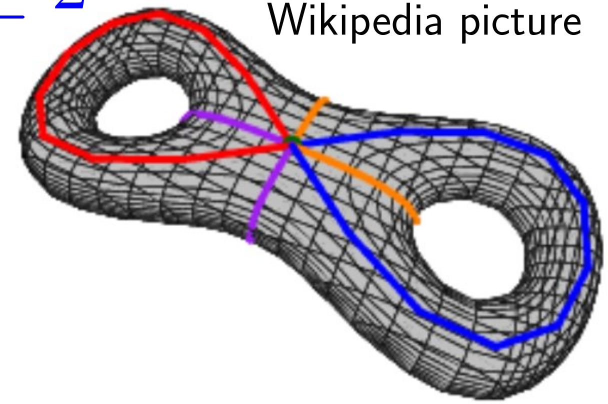






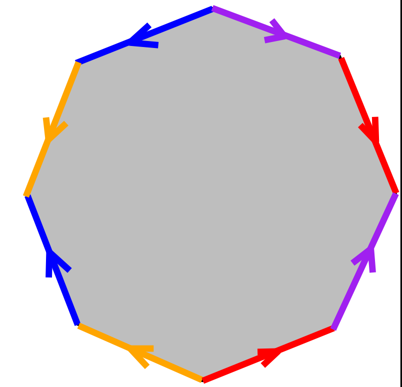


$g \geq 2$



Wikipedia picture

Polygonal scheme

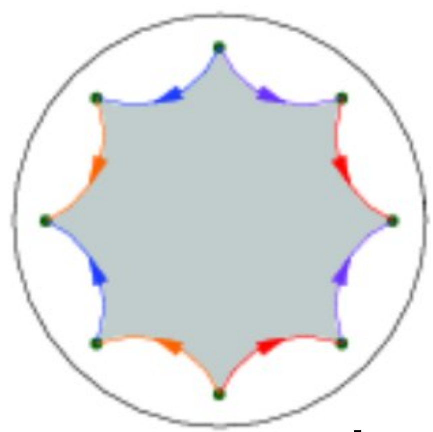


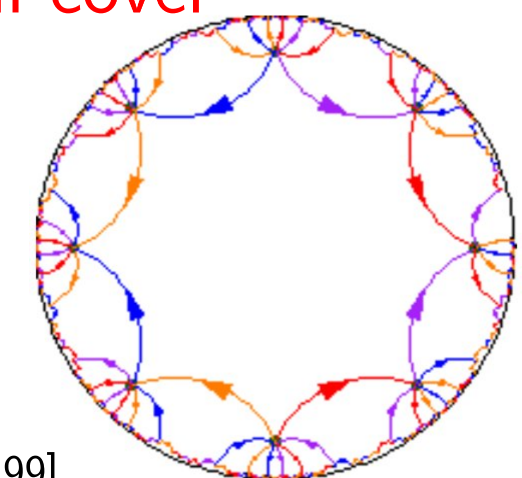
drawing in polynomial area

[Duncan, Goodrich, Kobourov, GD'09]

[Chambers, Eppstein, Goodrich, Löffler, GD'10]

Universal cover





[Mohar'99]

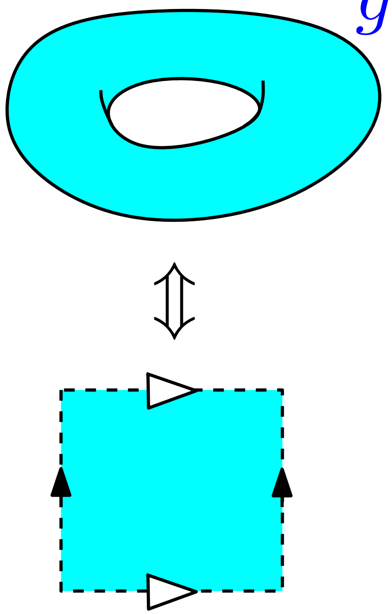
periodic drawing

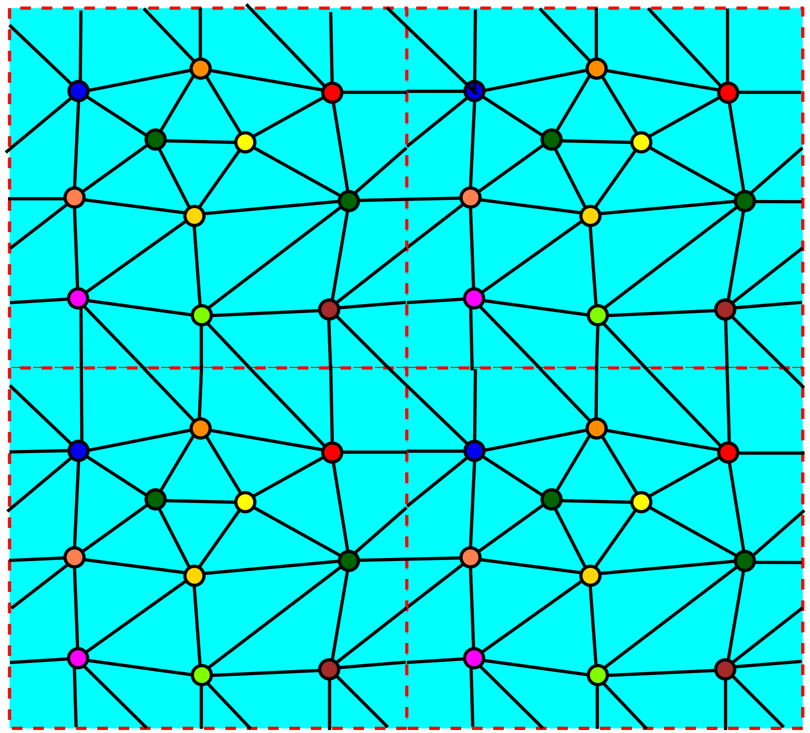
out of circle packing

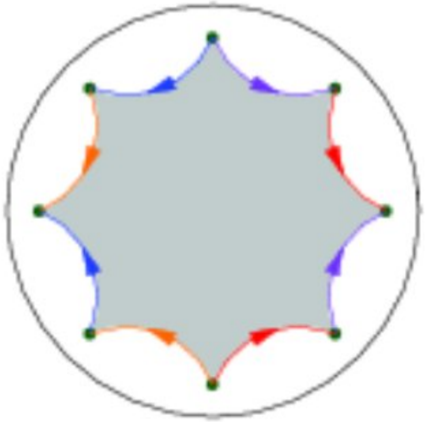
Drawing toroidal graphs

On the torus

$g = 1$



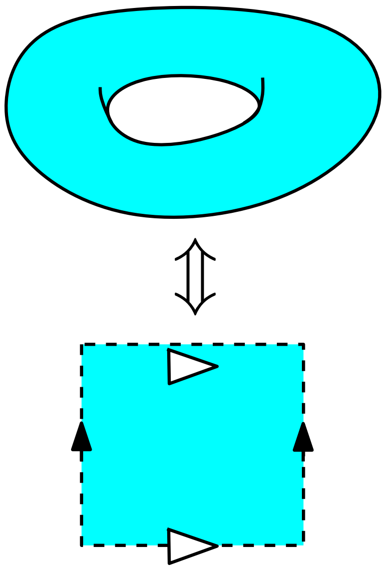


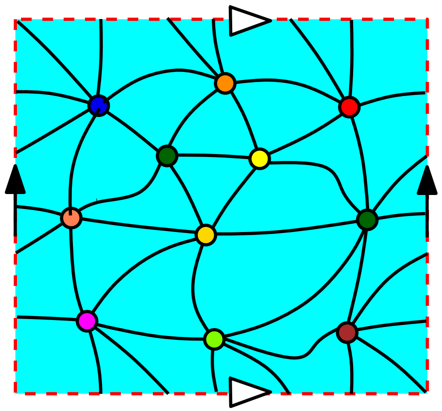


(Palais de la Découverte, Fête de la Science, October 2013)

Periodic straight-line drawings

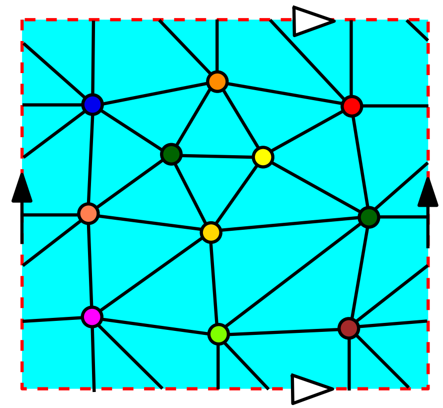
On the torus





drawing on the flat torus

$\Rightarrow$



straight-line drawing

x-periodic and

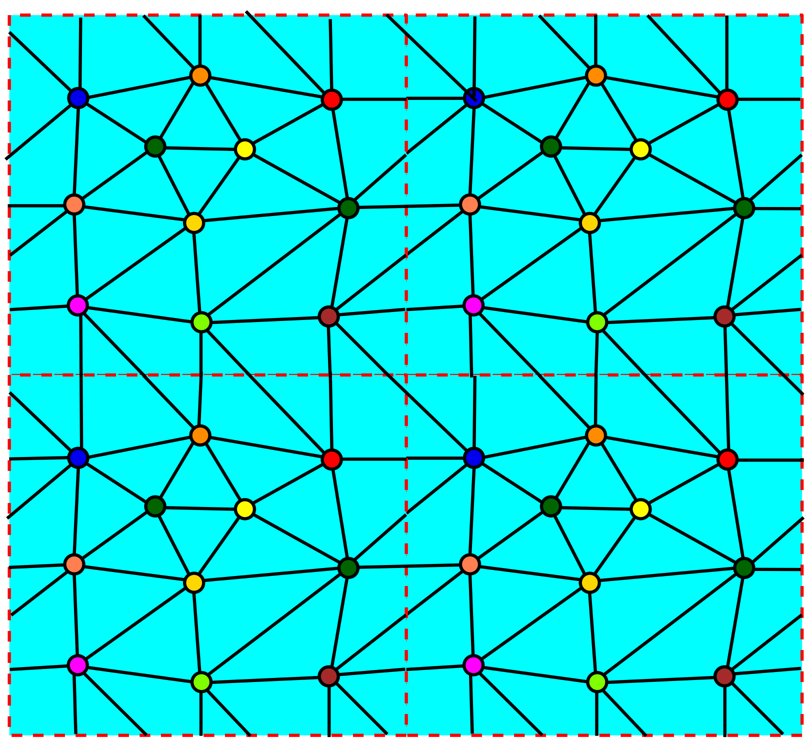
y-periodic drawing

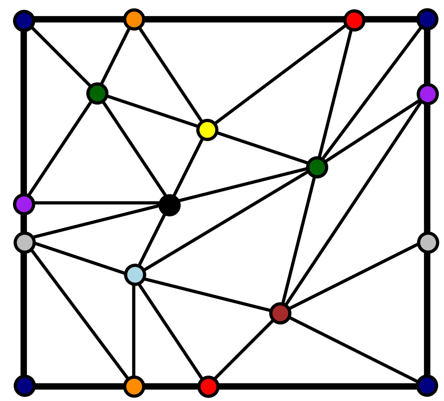
[Castelli Devillers Fusy, GD'12]

$O(n \times n^{\frac{3}{2}})$ **grid**

[Goncalves Lévêque, DCG]

$O(n^2 \times n^2)$ **grid**





straight-line frame

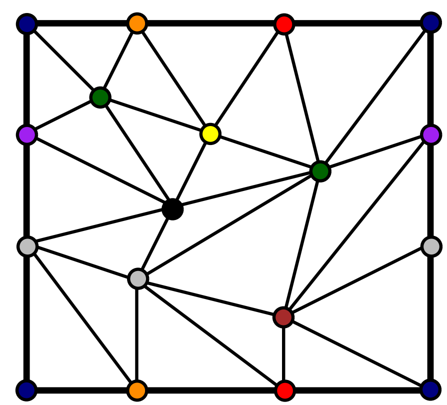
not x-periodic

not y-periodic

[Chambers et al., GD'10]

[Duncan et al., GD'09]

$O(n \times n^2)$ **grid**



straight-line frame

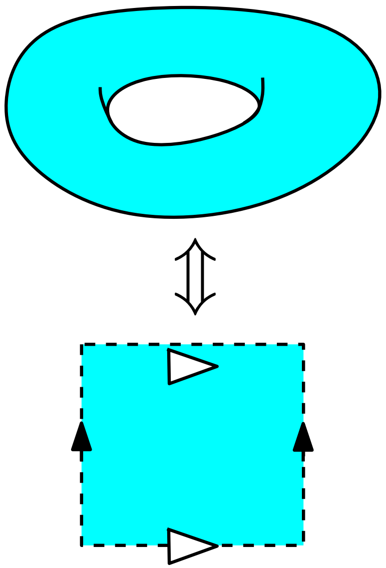
x-periodic and

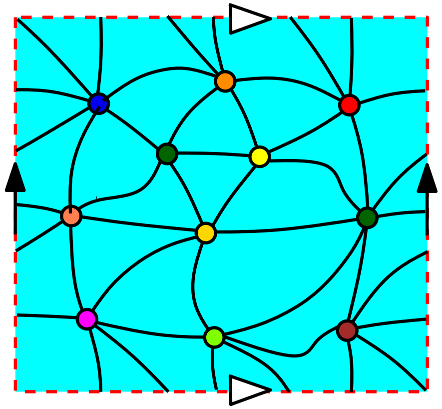
y-periodic drawing

[Castelli Fusy Kostrygin, Latin'14]

Periodic straight-line drawings

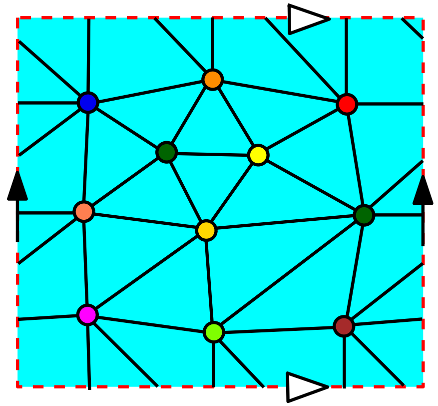
On the torus





drawing on the flat torus

⇒



straight-line drawing

x-periodic and

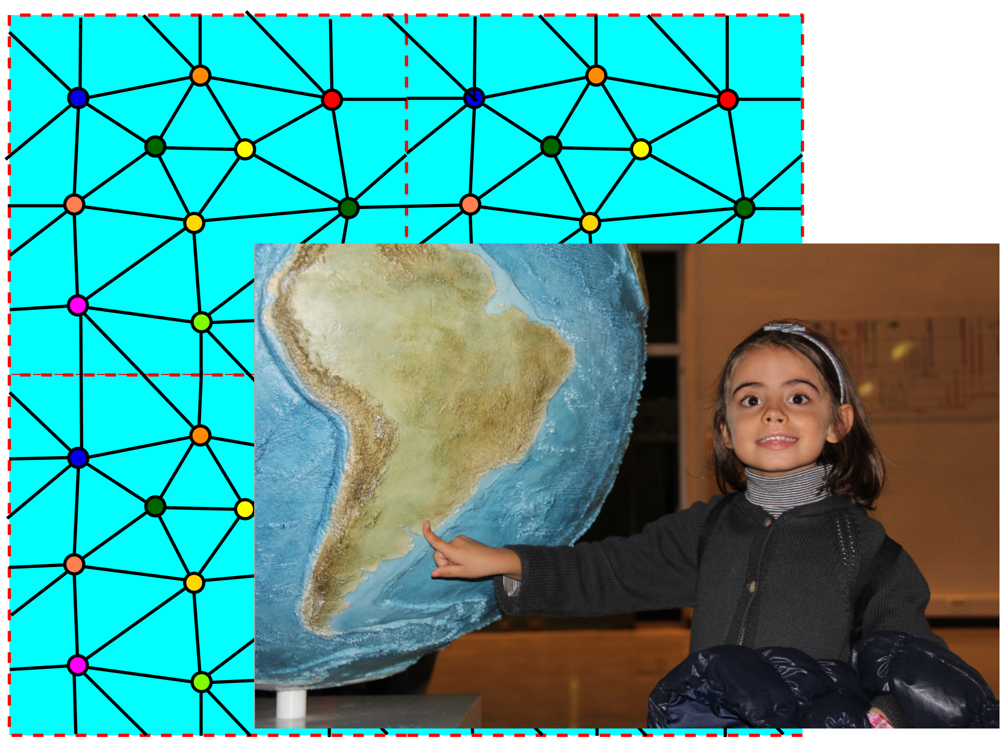
y-periodic drawing

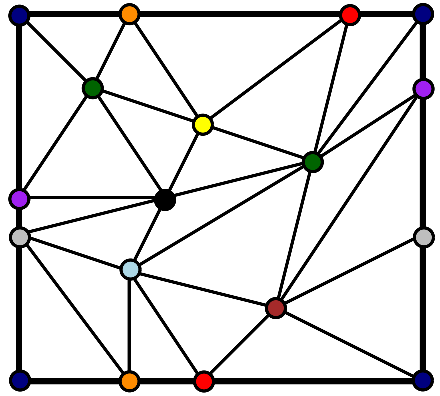
[Castelli Devillers Fusy, GD'12]

$O(n \times n^{\frac{3}{2}})$ **grid**

[Goncalves Lévêque, DCG]

$O(n^2 \times n^2)$ **grid**





straight-line frame

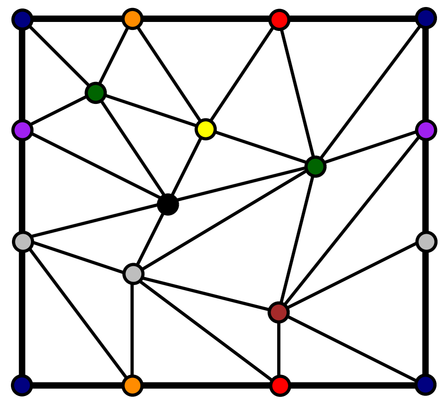
not x-periodic

not y-periodic

[Chambers et al., GD'10]

[Duncan et al., GD'09]

$O(n \times n^2)$ **grid**



straight-line frame

x-periodic and

y-periodic drawing

[Castelli Fusy Kostrygin, Latin'14]

$O(n^4 \times n^4)$ **grid**

A shift-algorithm for the torus

1. Recall algorithm of

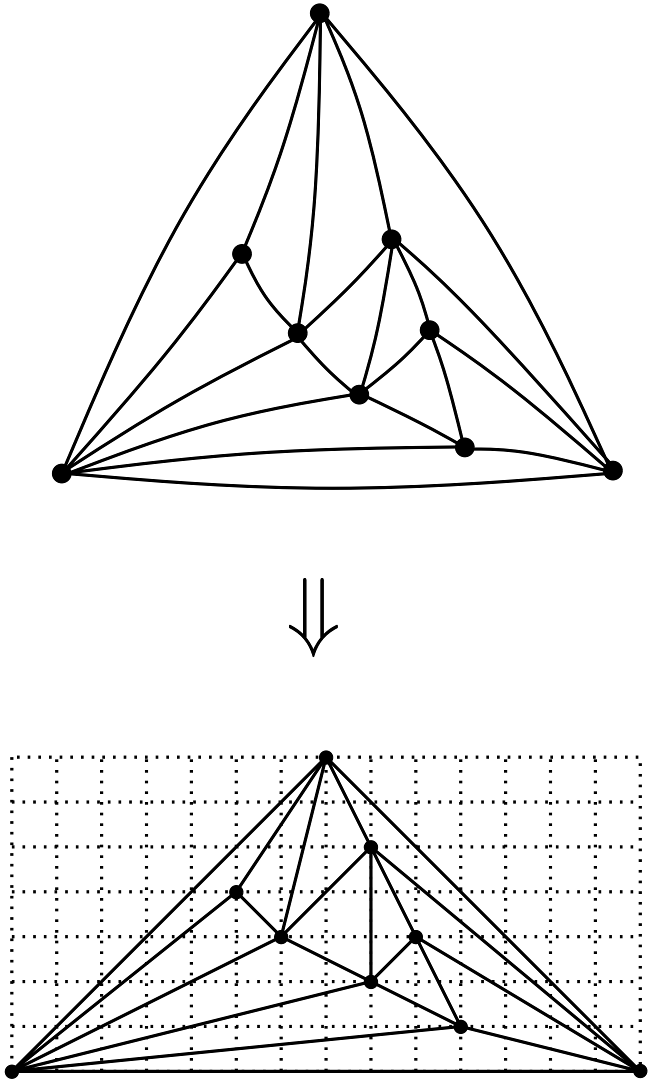
2. Extend to the cylinder

3. Get toroidal drawings

[De Fraysseix et al'89]

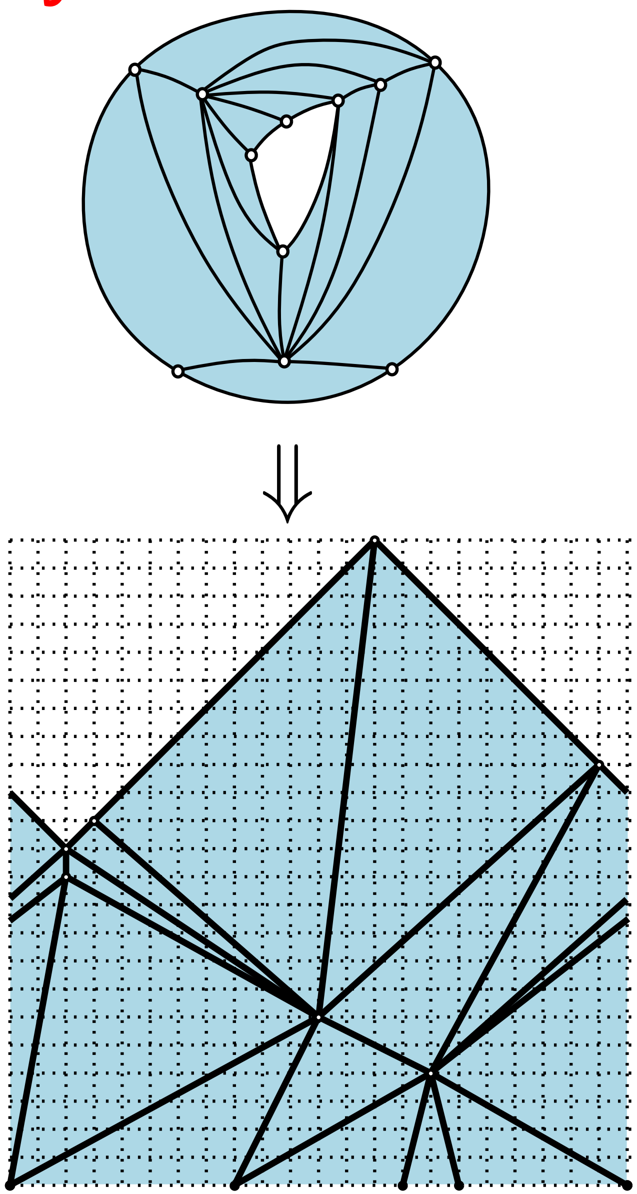
[Castelli Aleardi Fusy Devillers 2012]

Plane



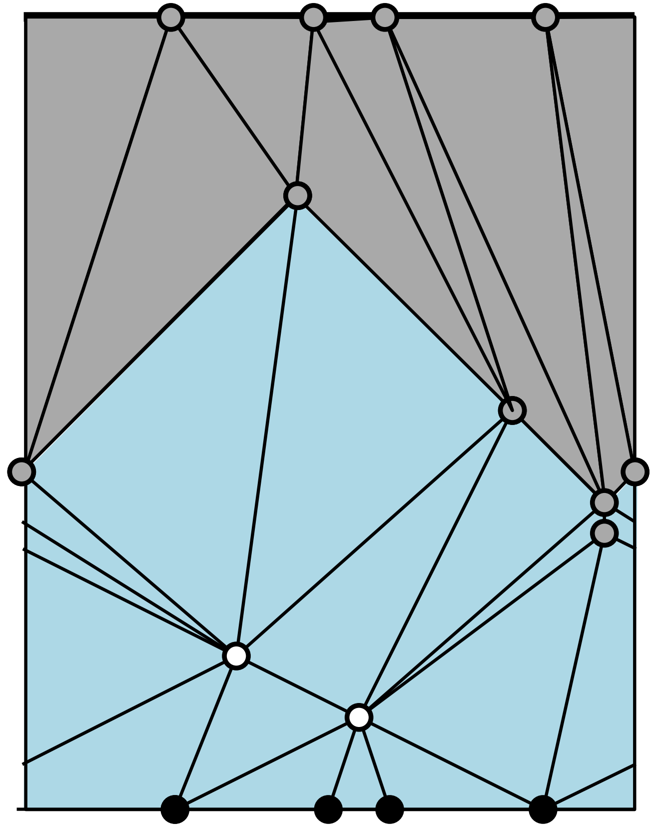
Grid $2n-4 \times n-2$

Cylinder



Grid $\leq 2n \times n(2d+1)$

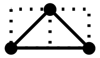
Torus

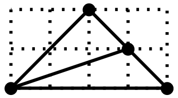


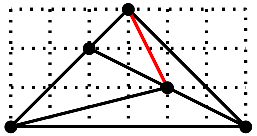
Grid$\leq 2n \times (1+n(2c+1))$

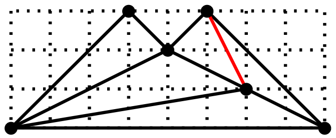
Incremental drawing algorithm

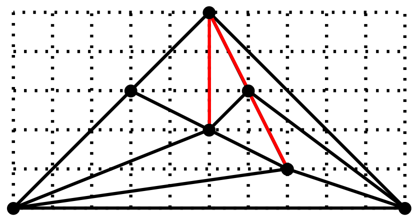
[de Fraysseix, Pollack, Pach'89]

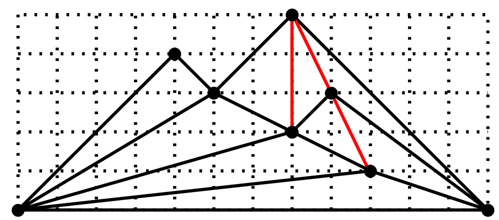
1. 

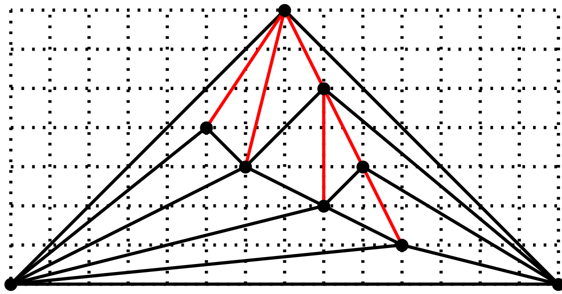
2. 

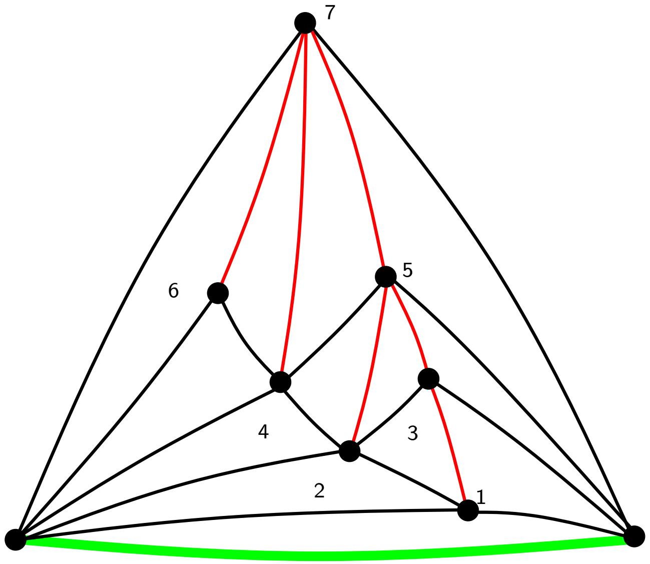
3. 

4. 

5. 

6. 

7. 

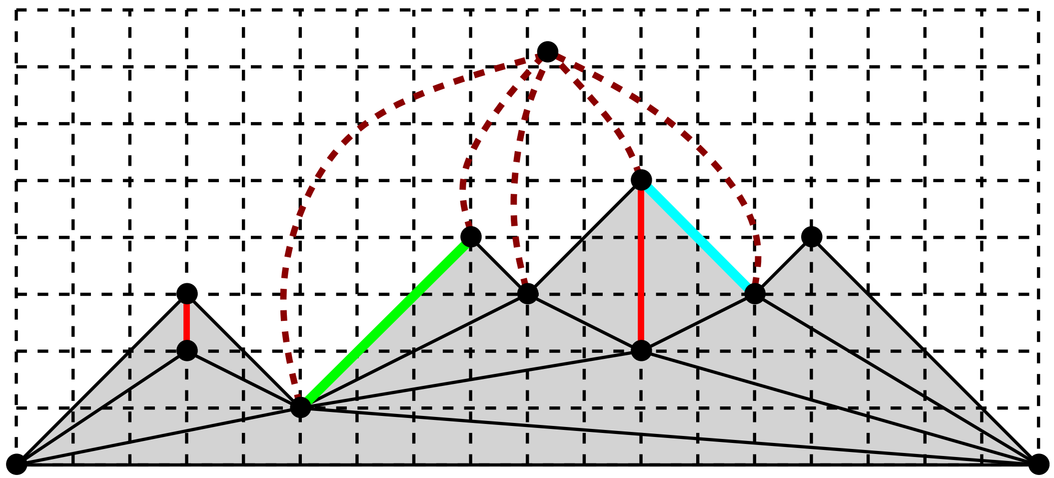


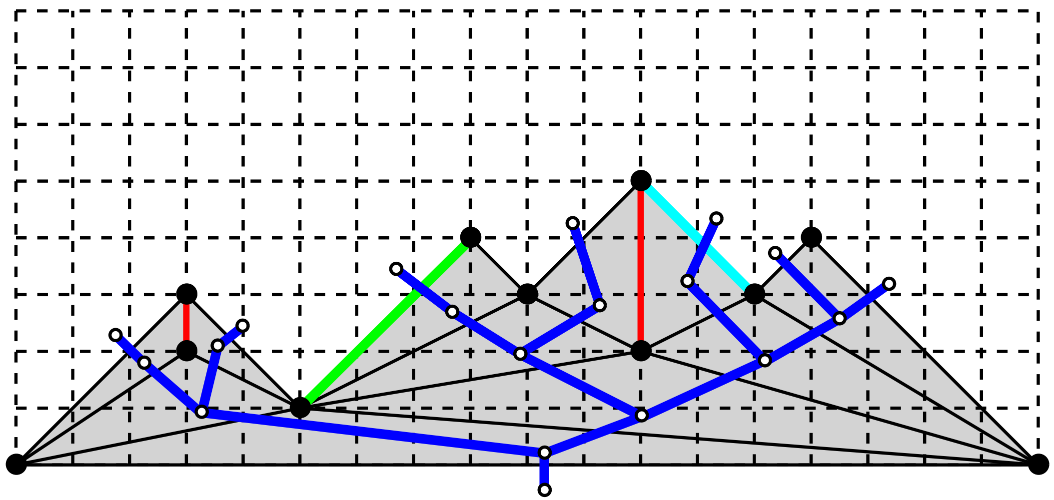
Grid size of G_k: $2k \times k$

Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

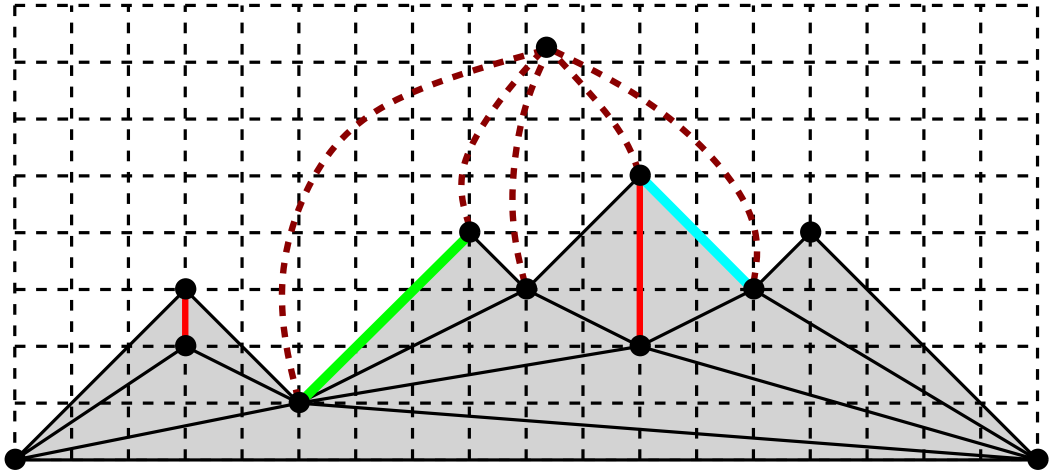


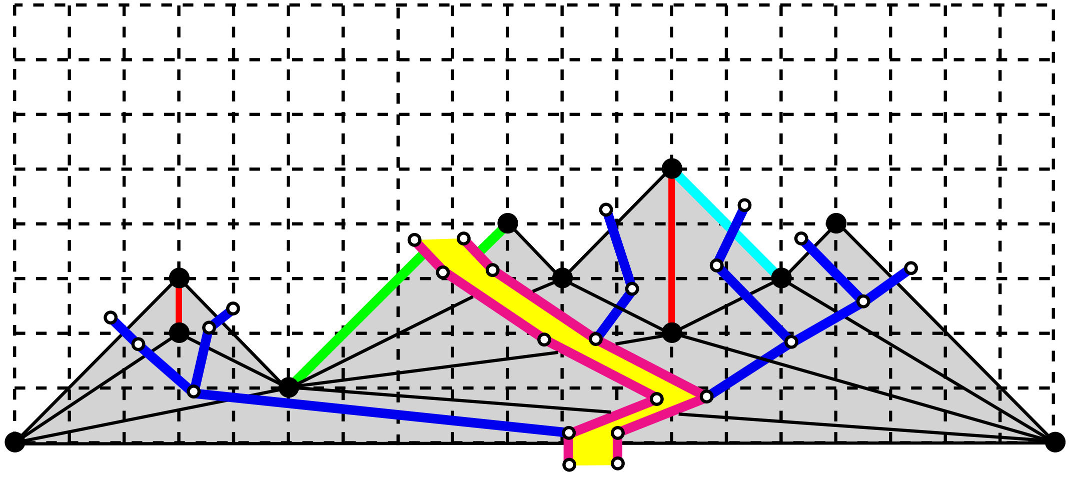


Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

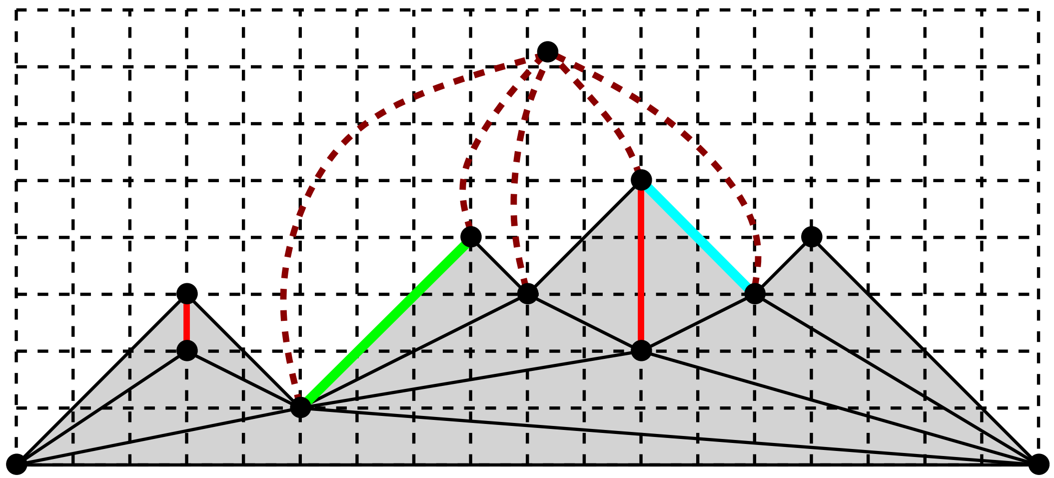


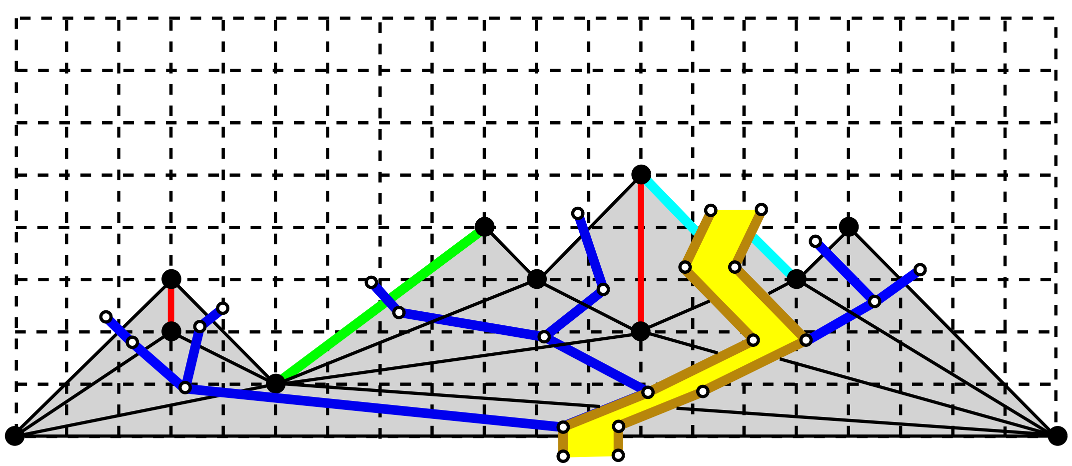


Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

G_{k-1}

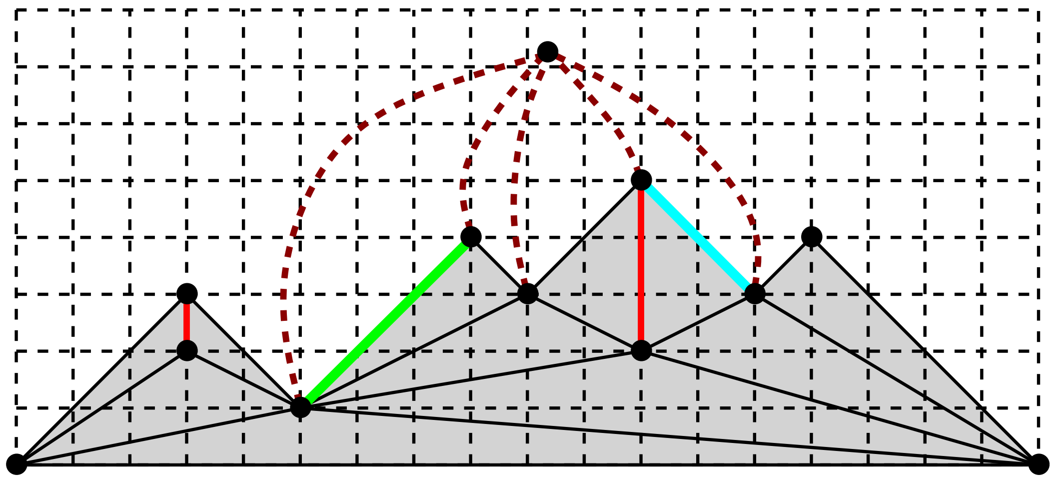




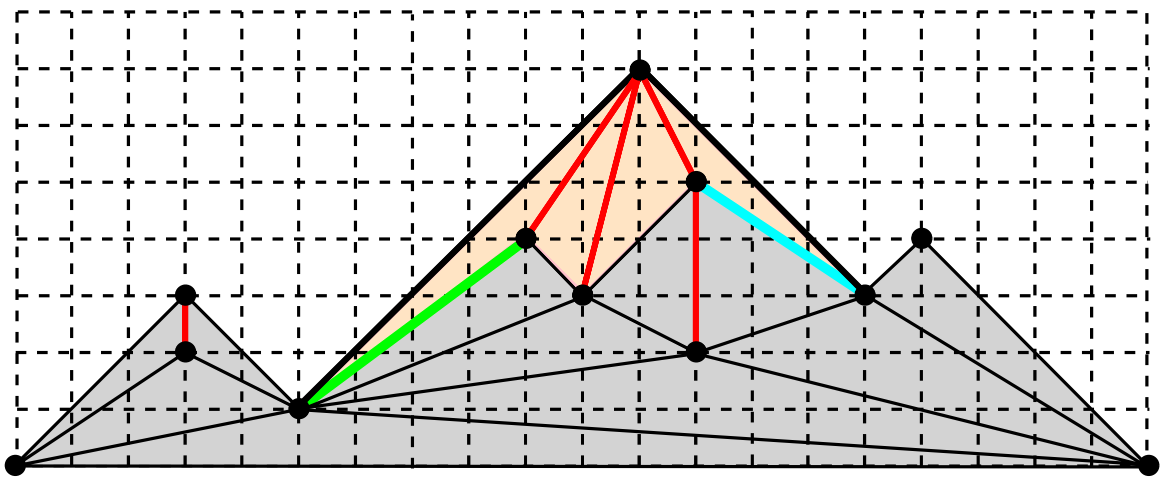
Reformulation of the shift-step

At each step: insert two vertical strips of width 1 using the dual tree

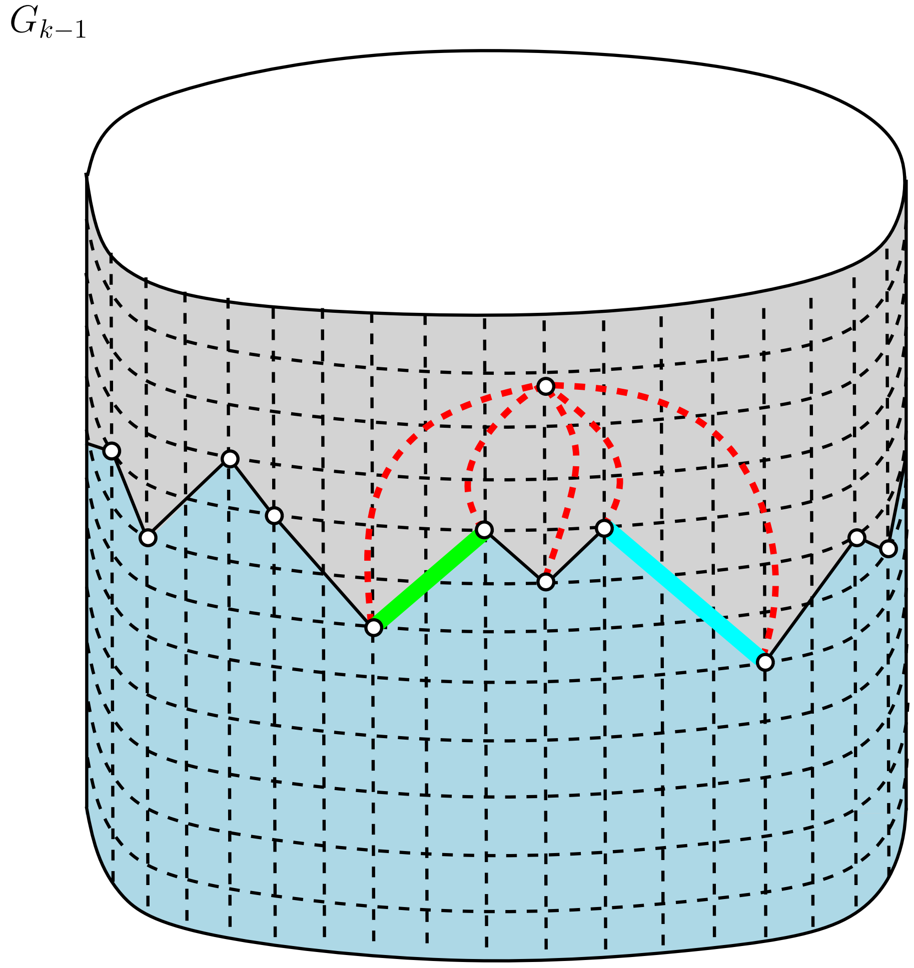
G_{k-1}



G_k

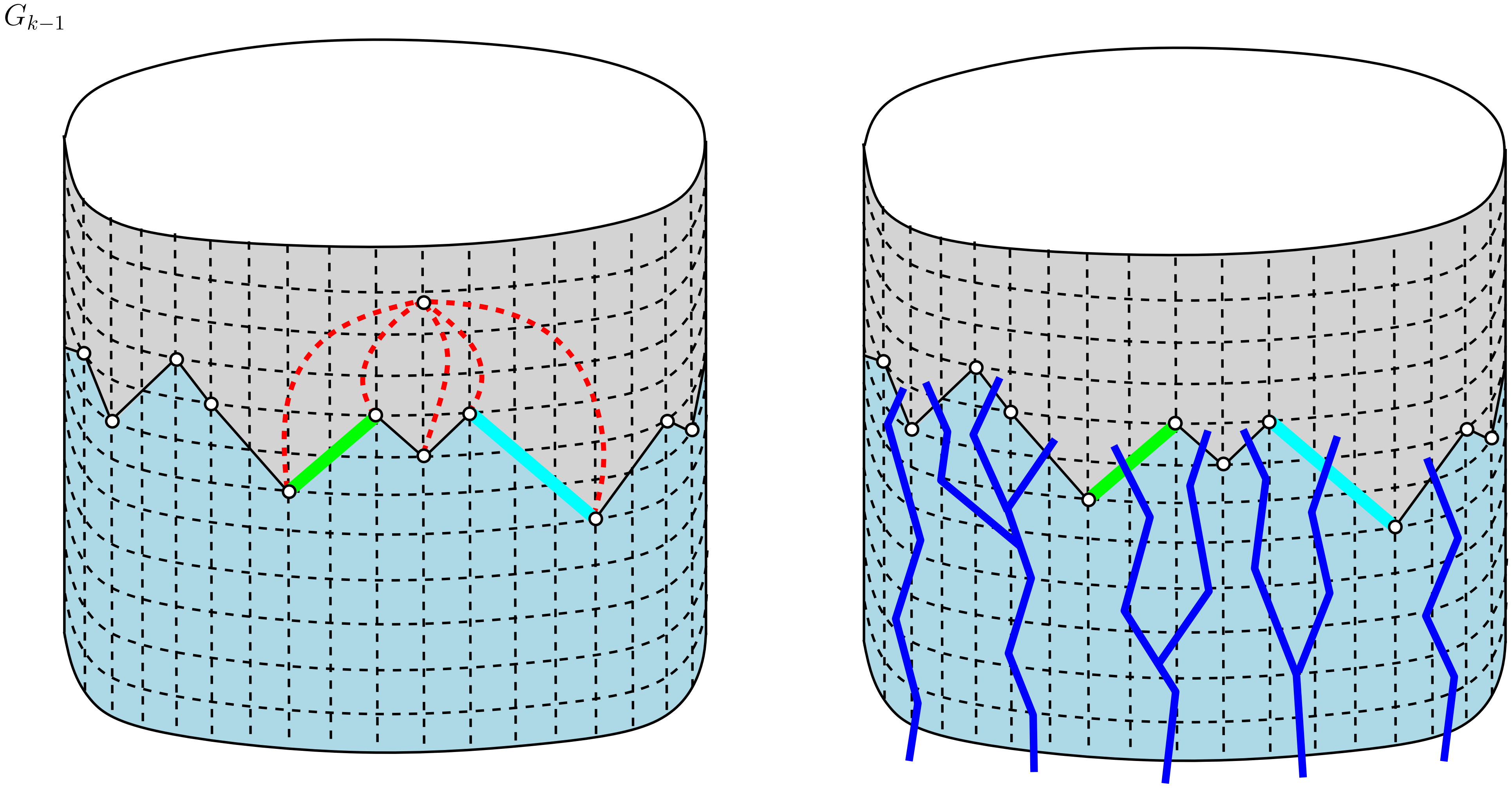


Extension to the cylinder: drawing algorithm



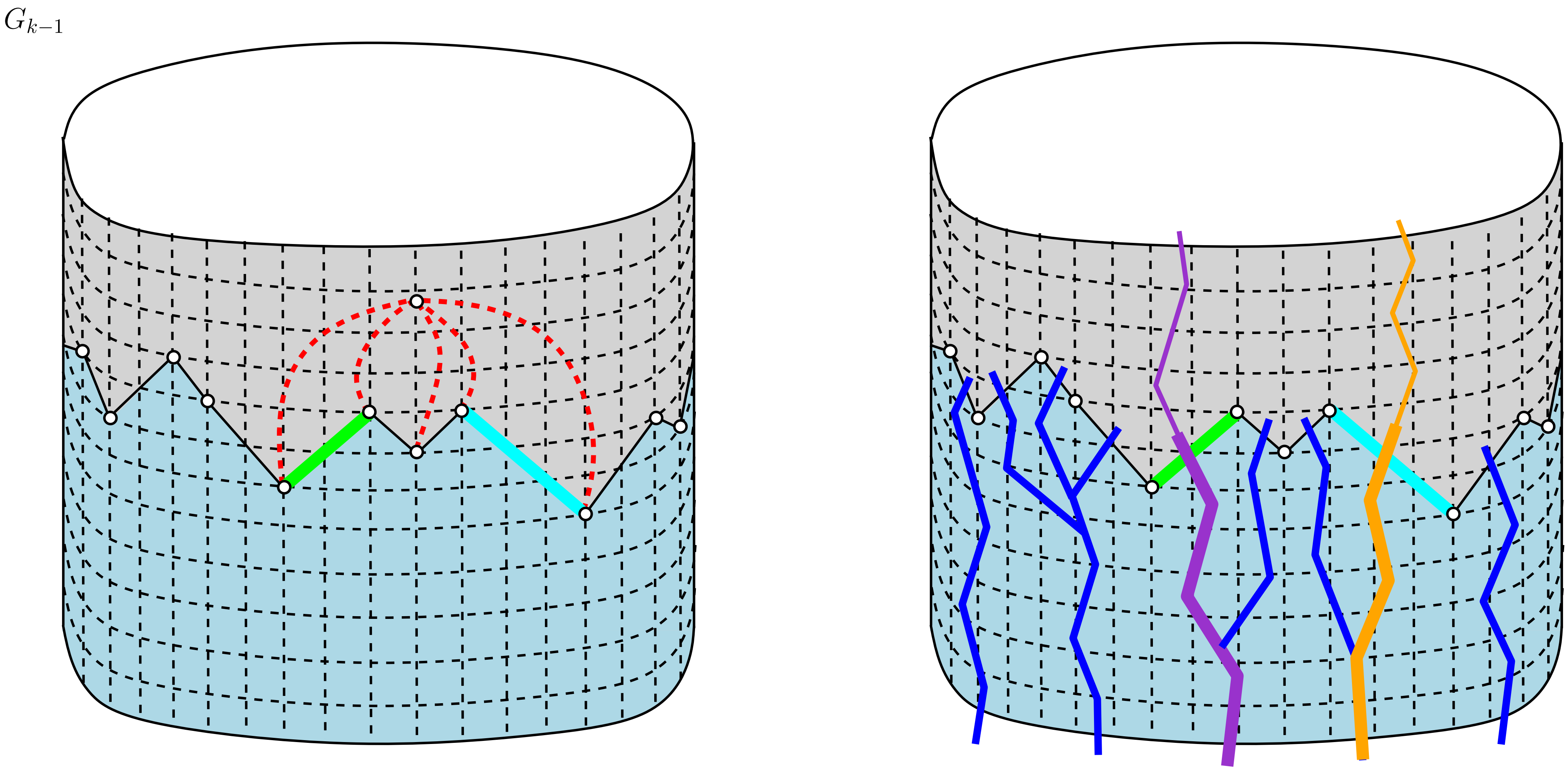
At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

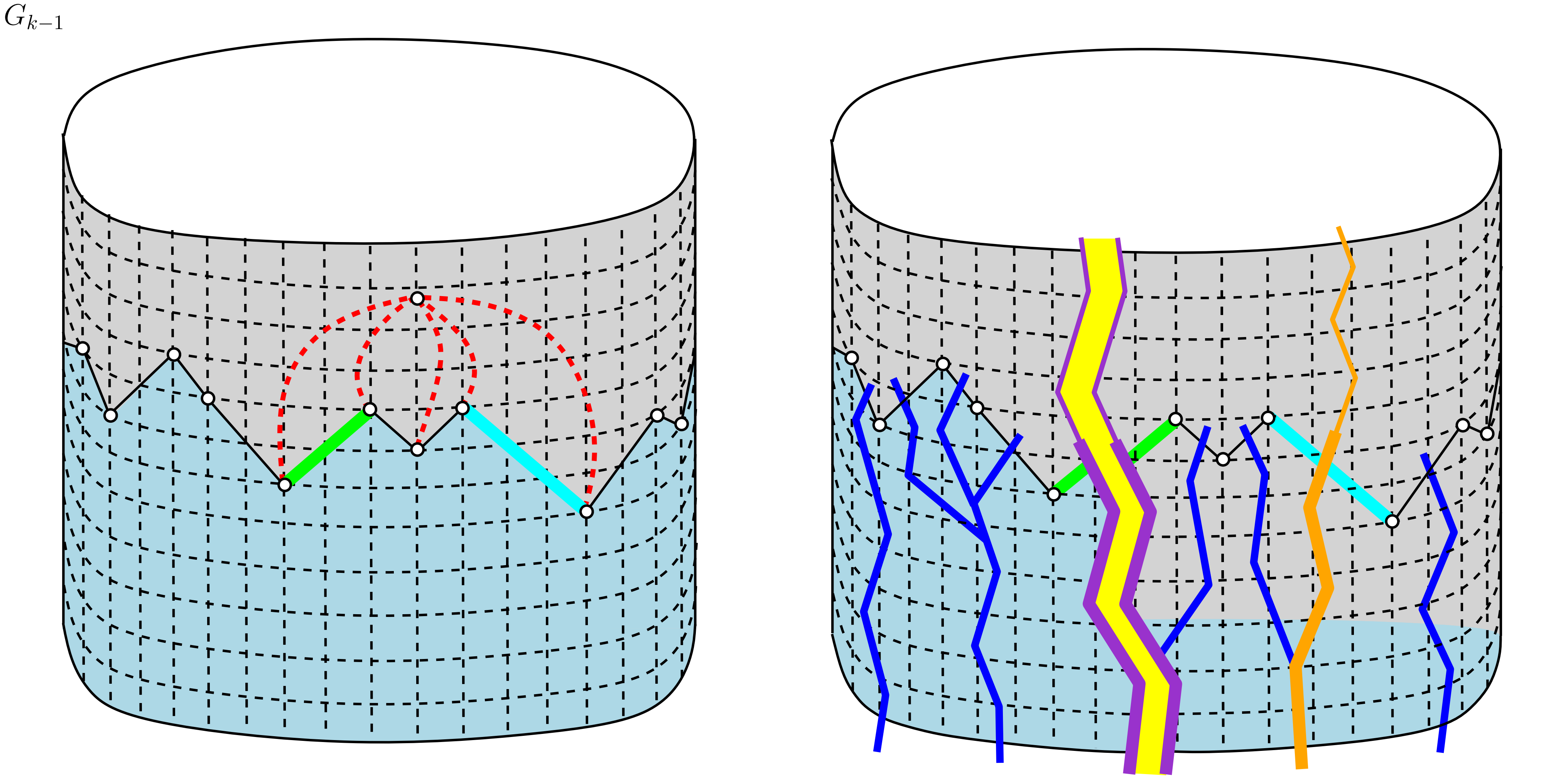
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

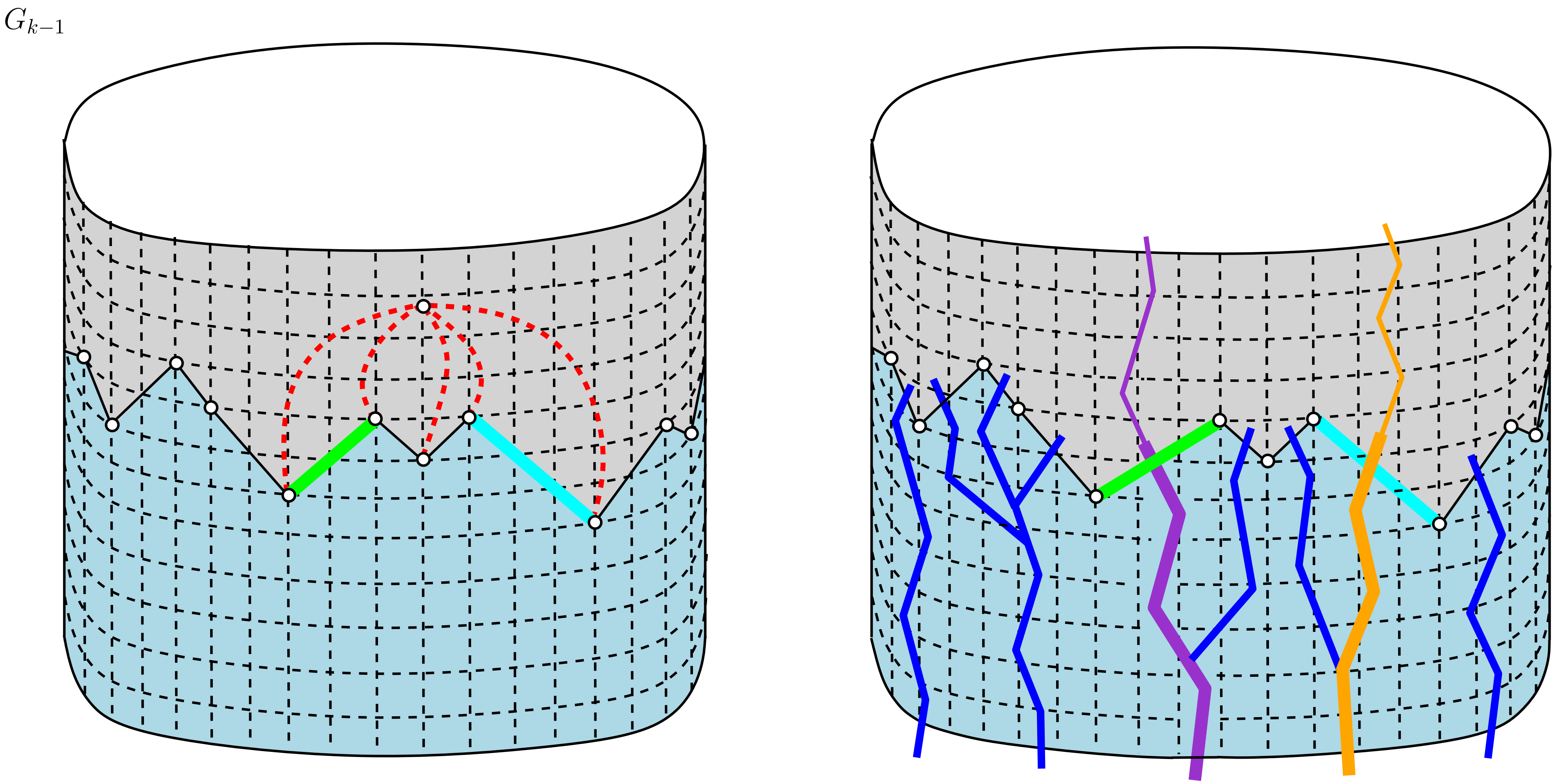
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

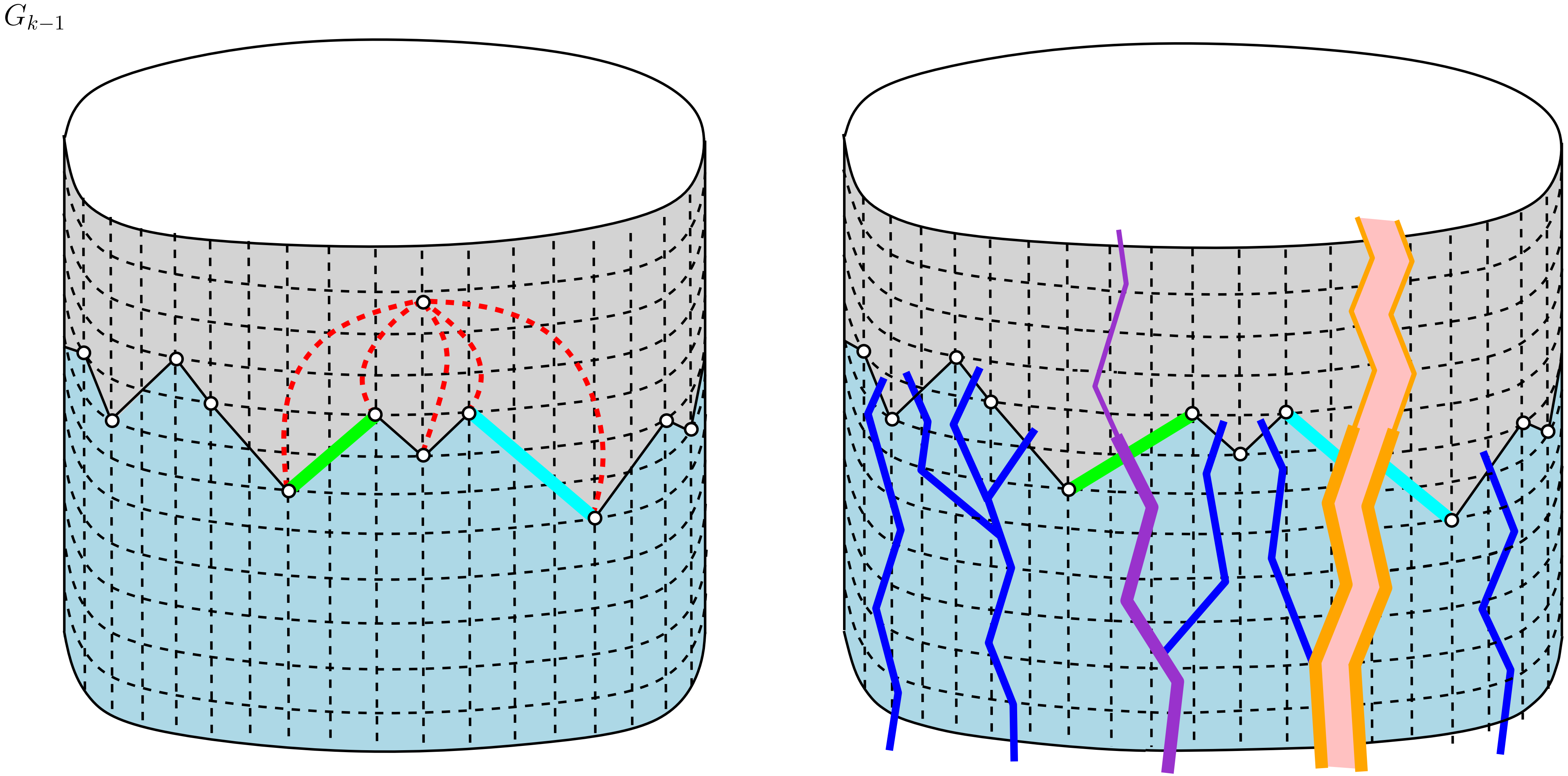
- insert the next vertex as in the planar case

Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

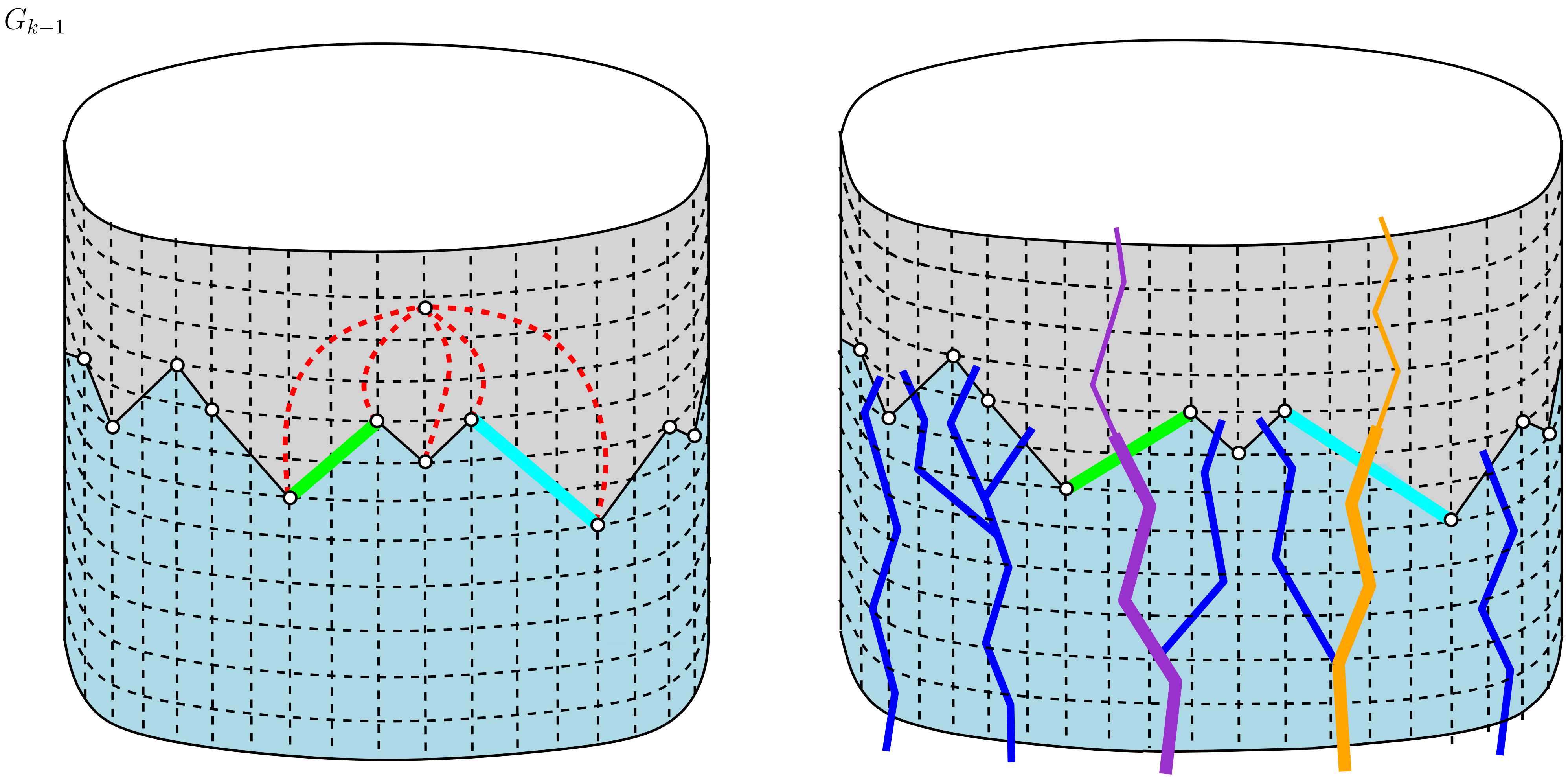
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1

- insert the next vertex as in the planar case

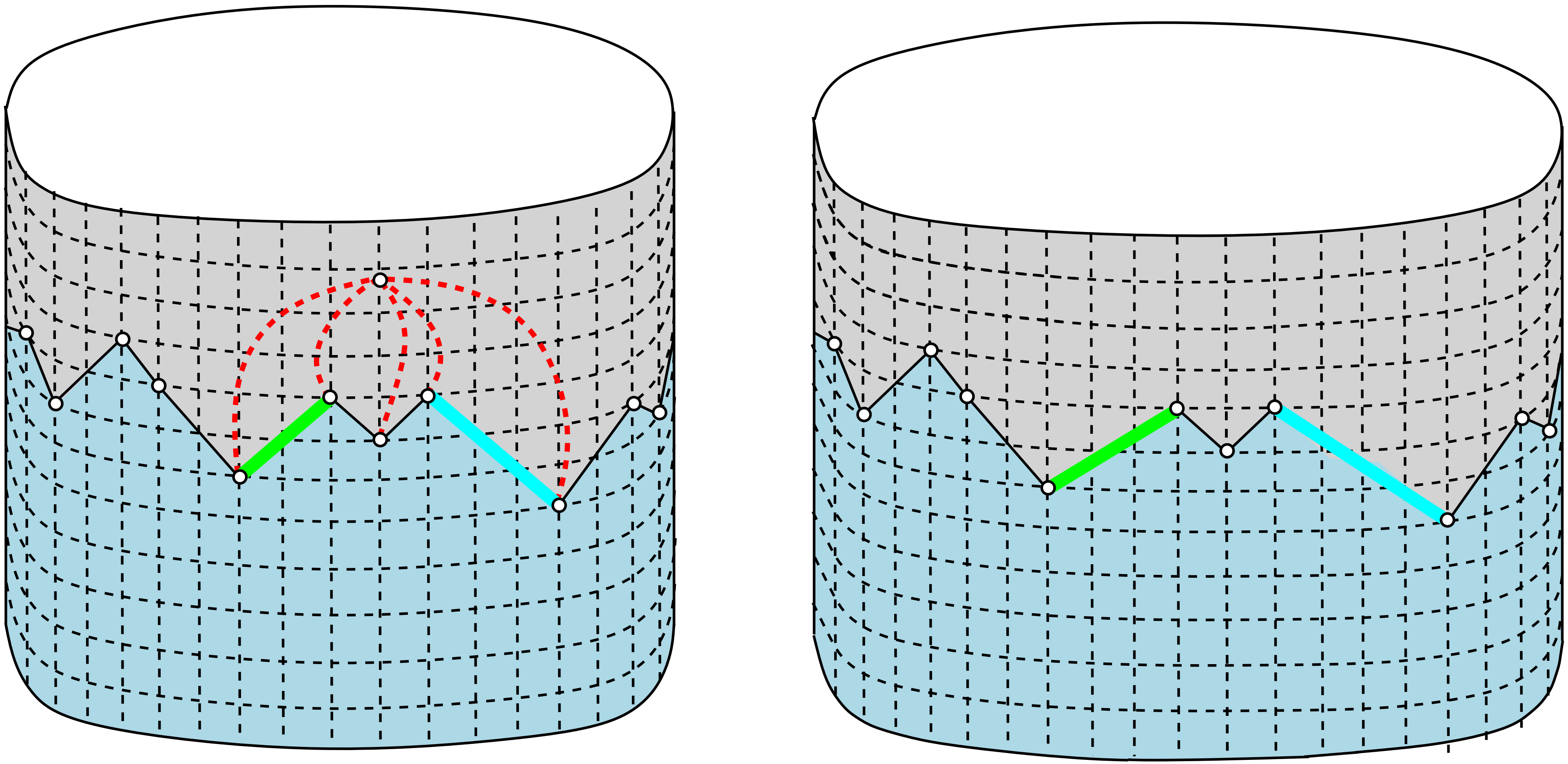
Extension to the cylinder: drawing algorithm



At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

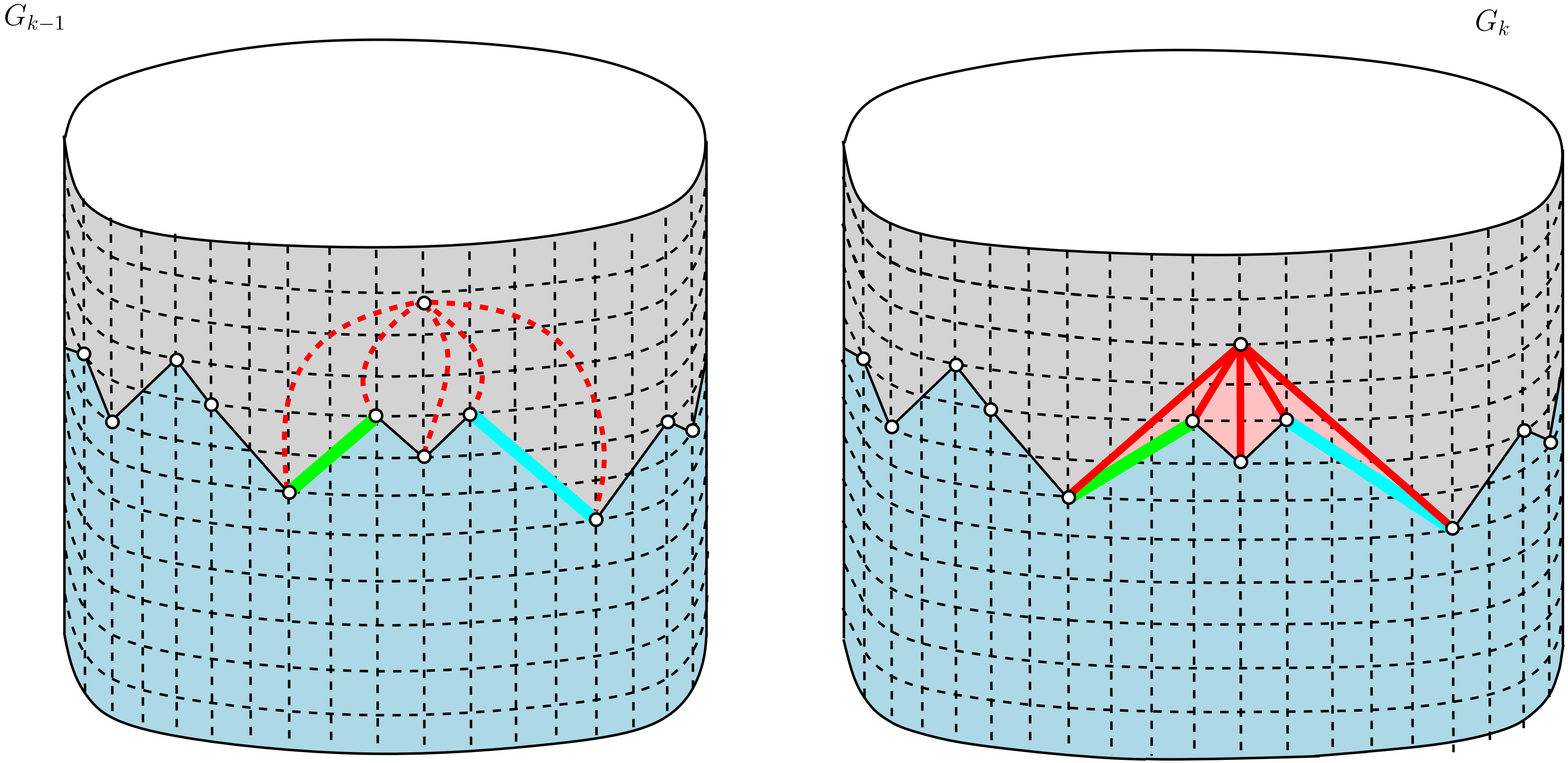
Extension to the cylinder: drawing algorithm

G_{k-1}



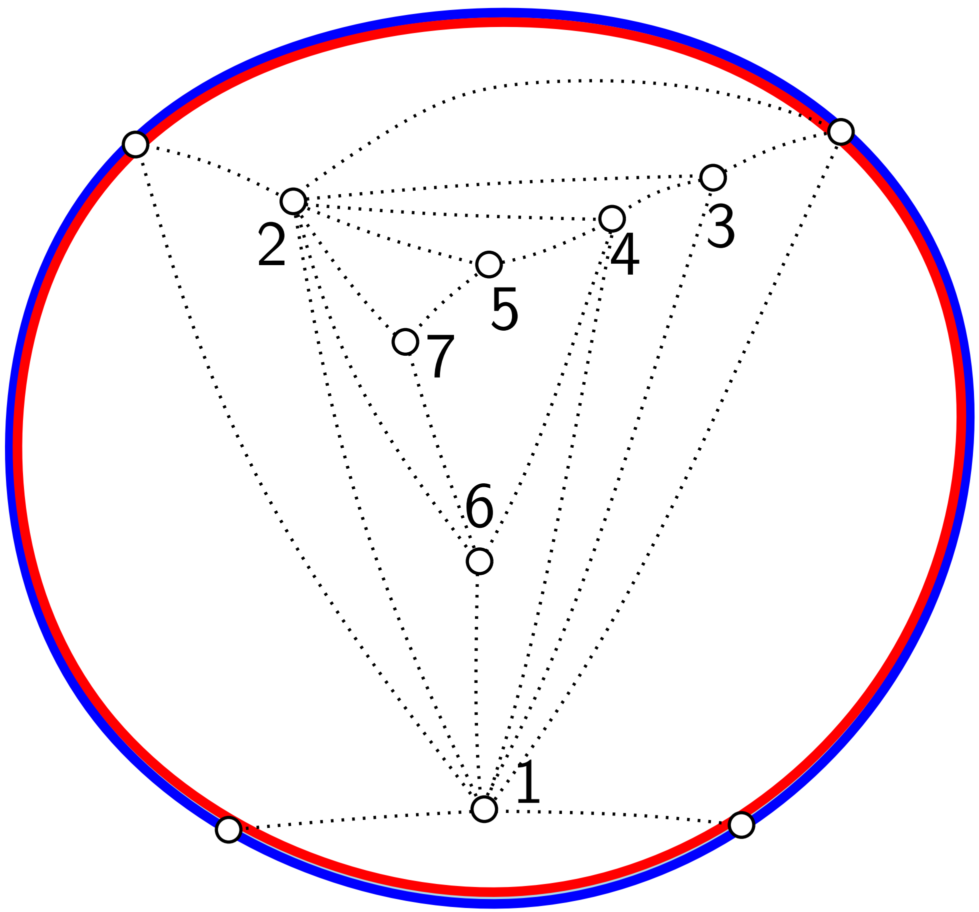
At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

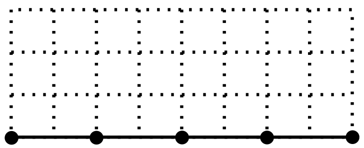
Extension to the cylinder: drawing algorithm



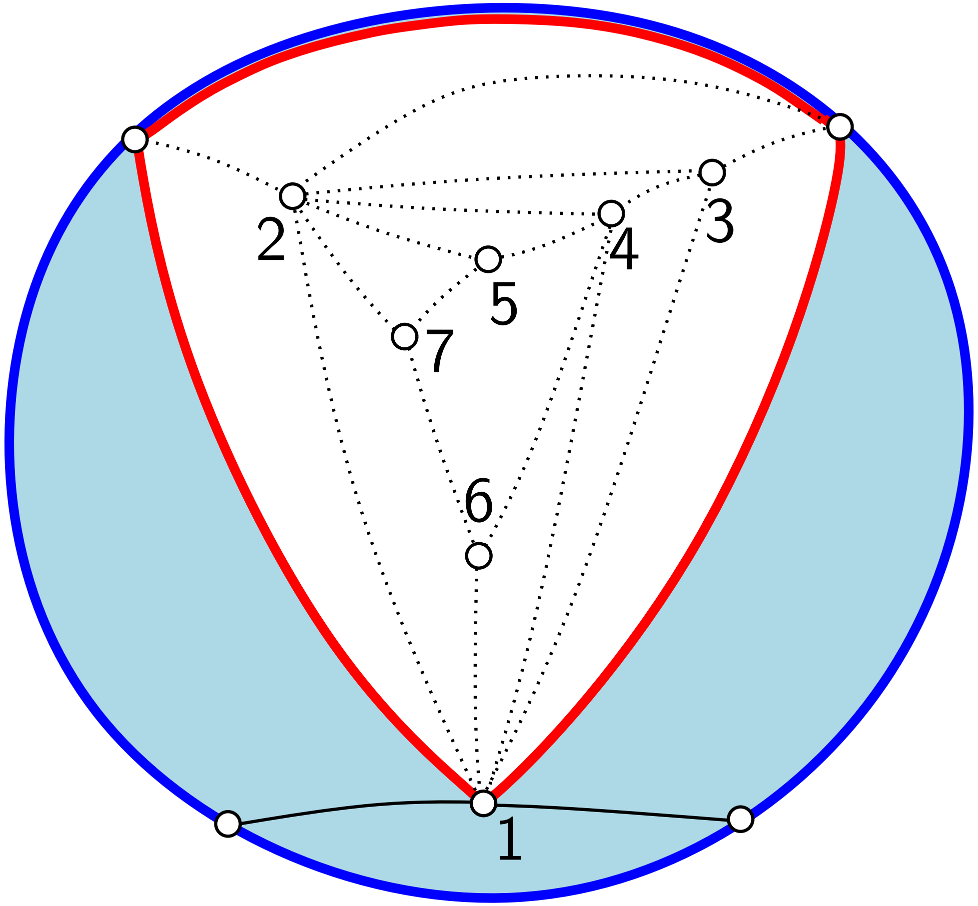
At each step: - insert two vertical strips of width 1
- insert the next vertex as in the planar case

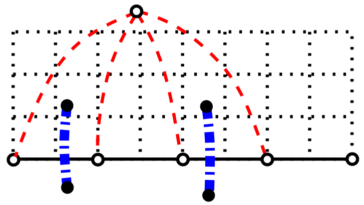
Extension to the cylinder: drawing algorithm

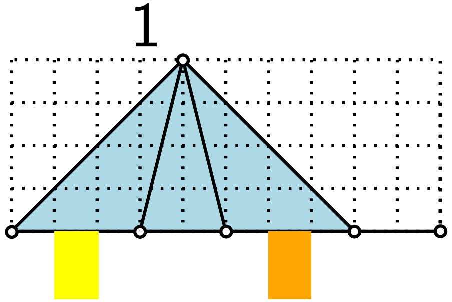




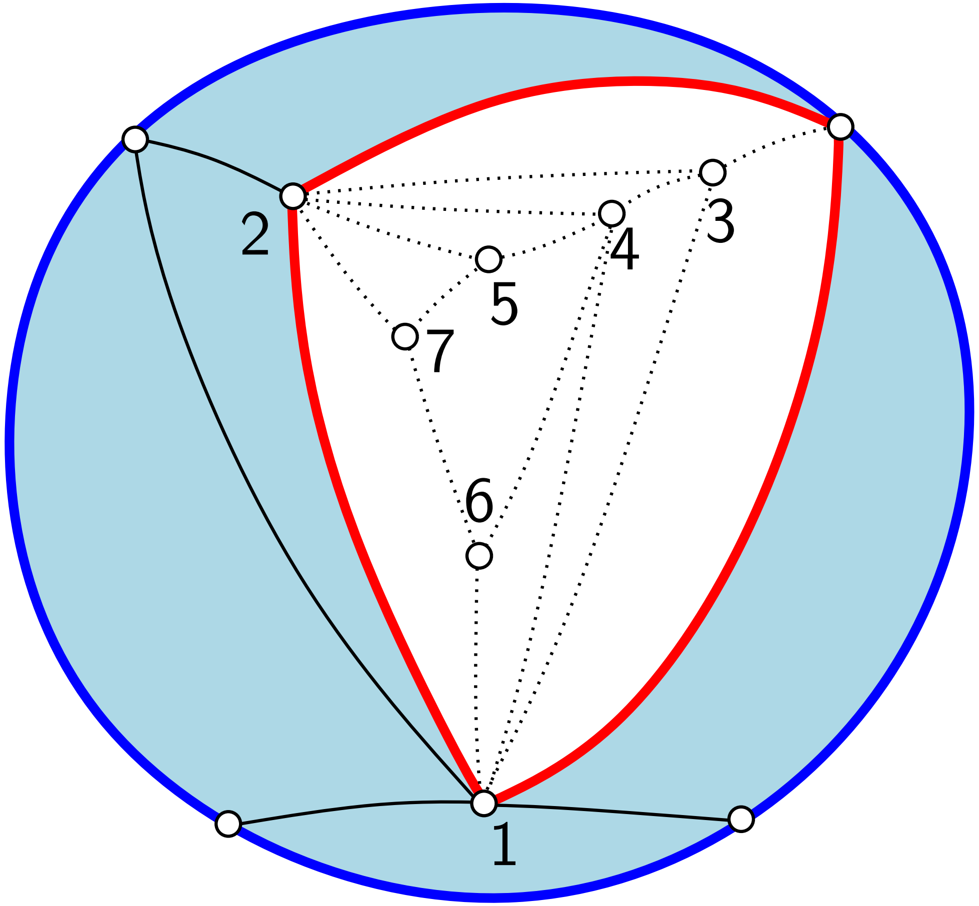
Extension to the cylinder: drawing algorithm

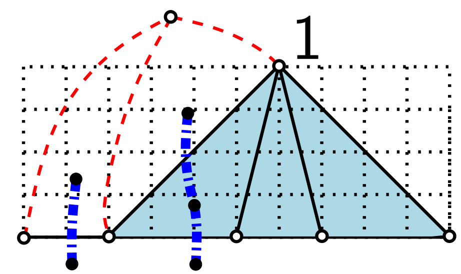


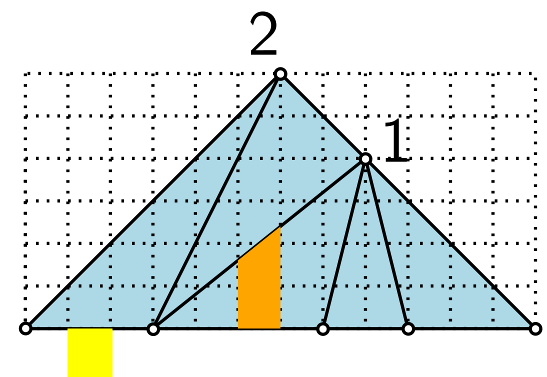




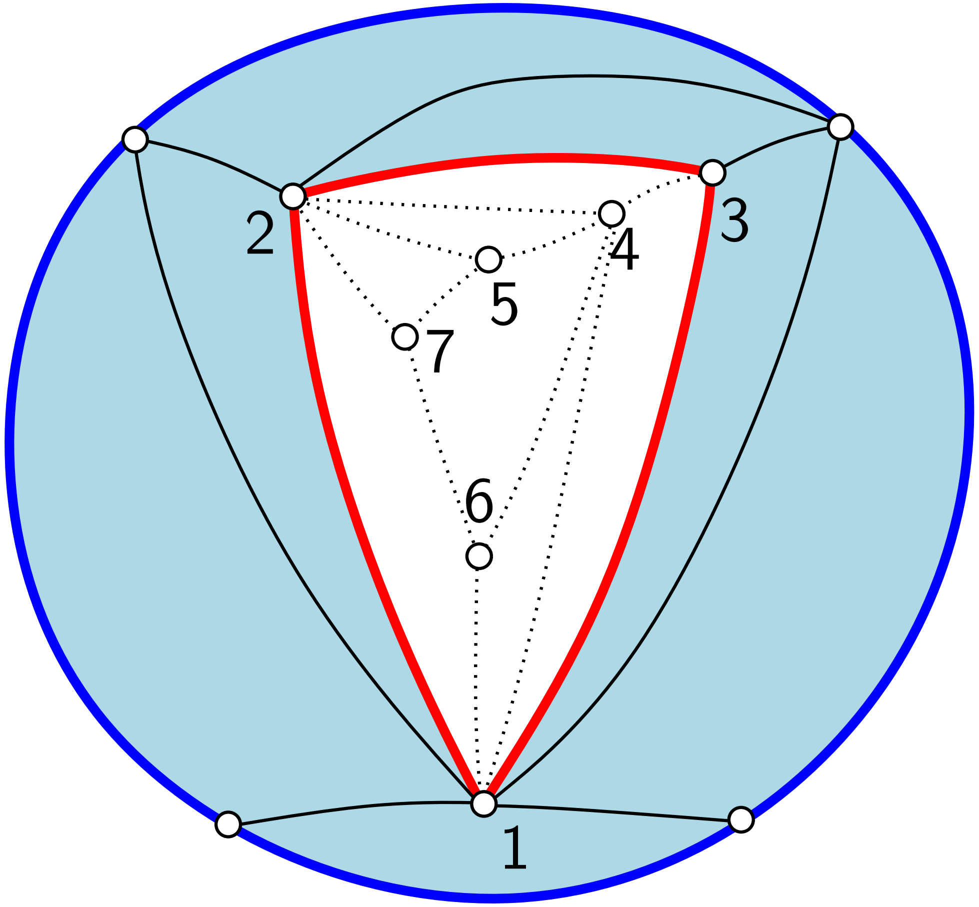
Extension to the cylinder: drawing algorithm

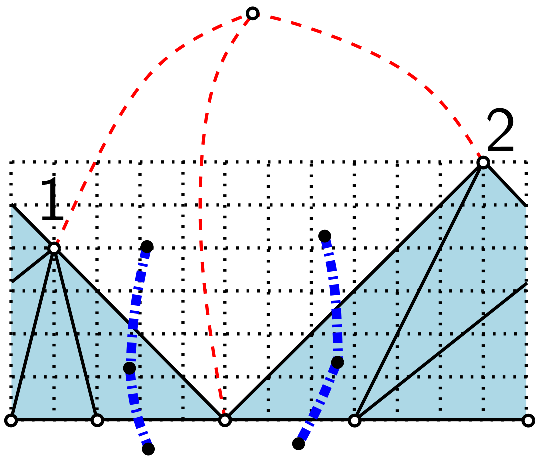


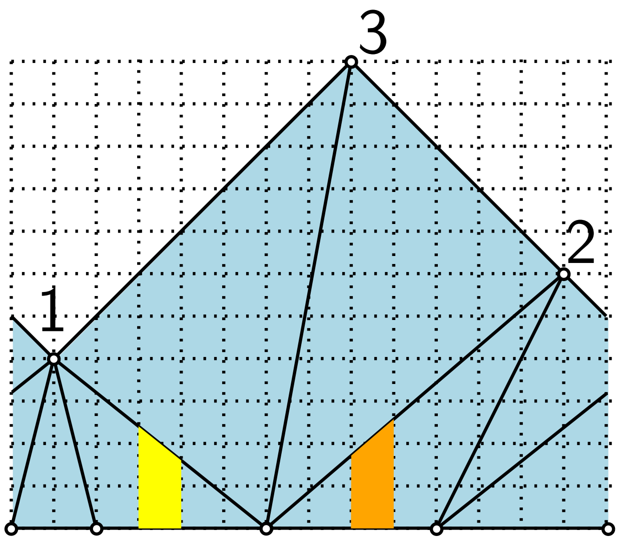




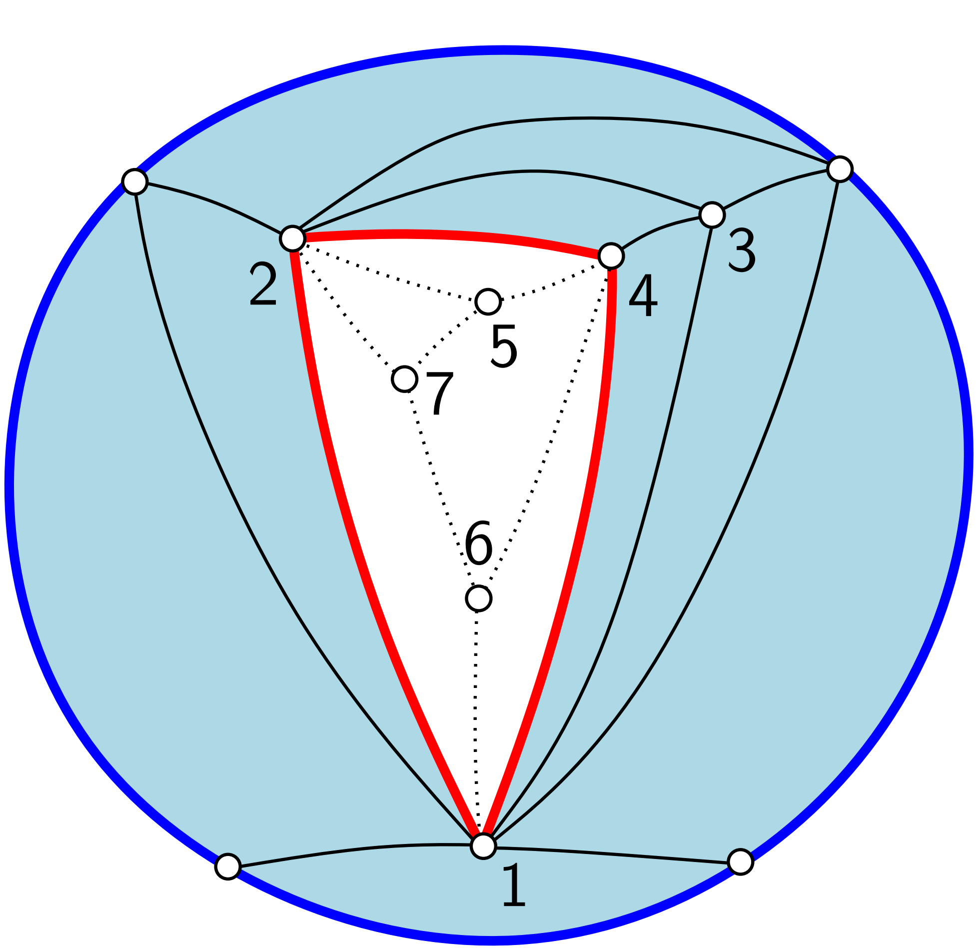
Extension to the cylinder: drawing algorithm

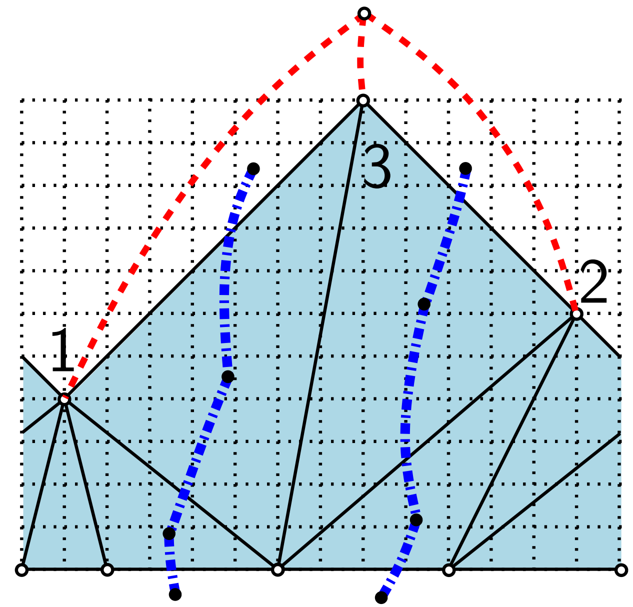


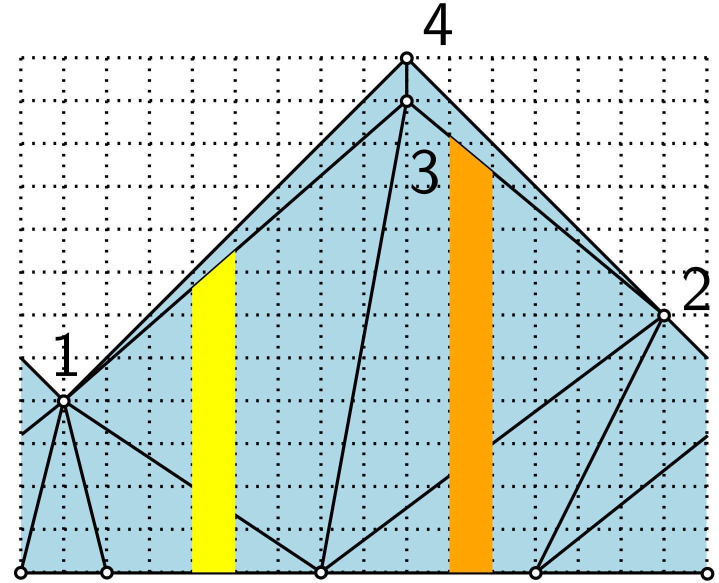




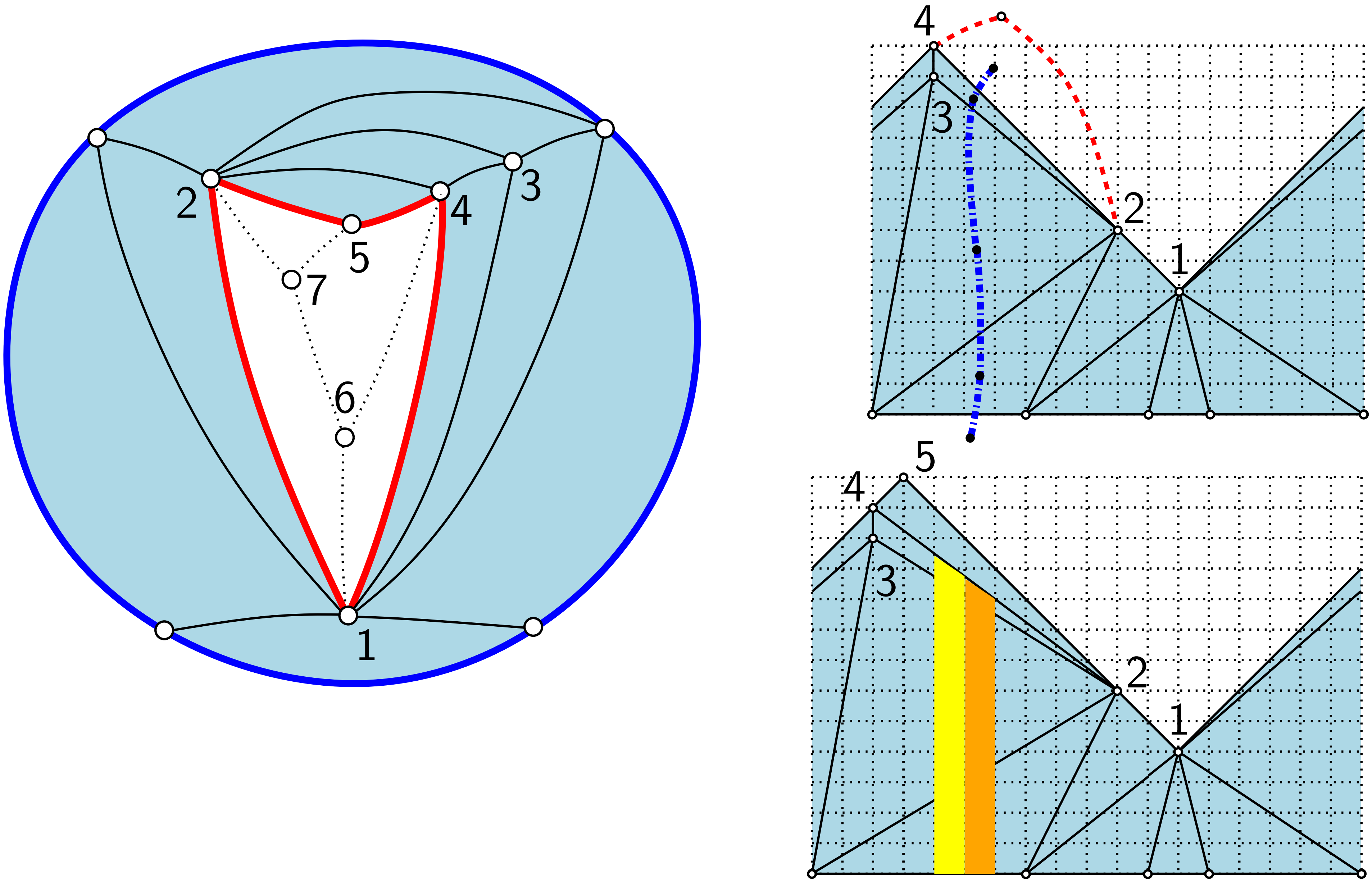
Extension to the cylinder: drawing algorithm



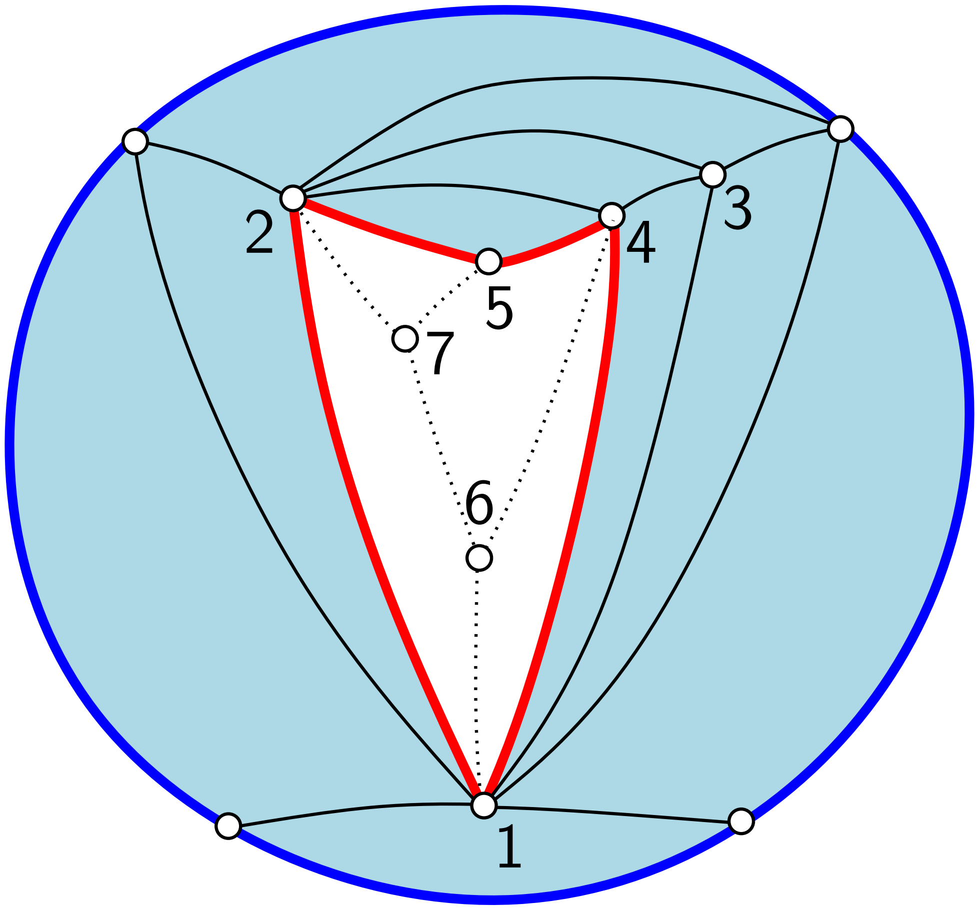


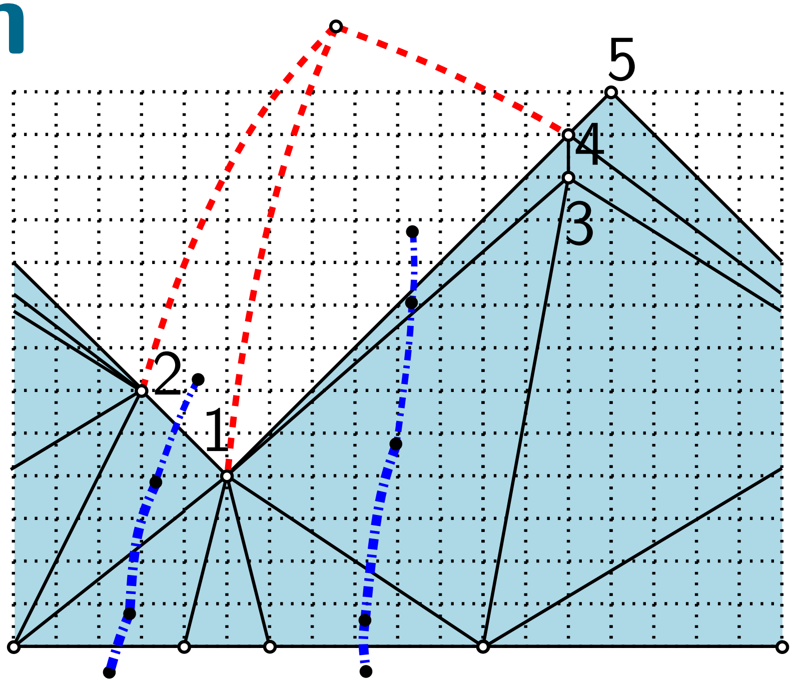


Extension to the cylinder: drawing algorithm

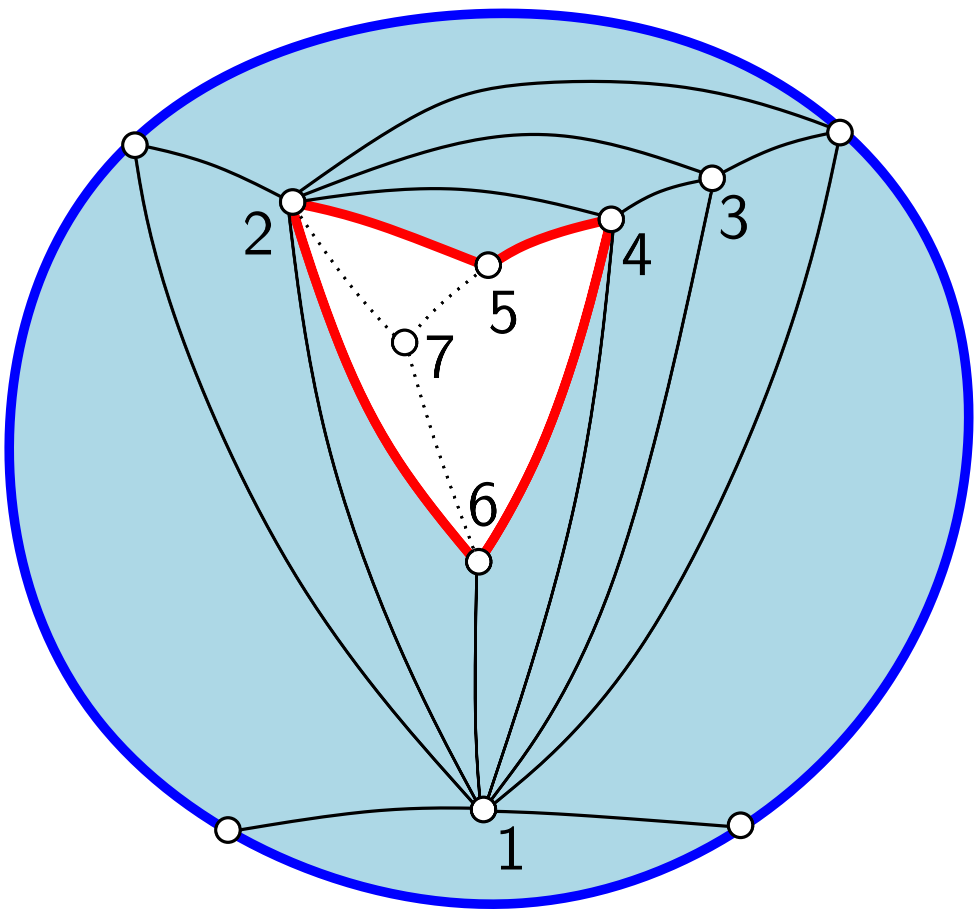


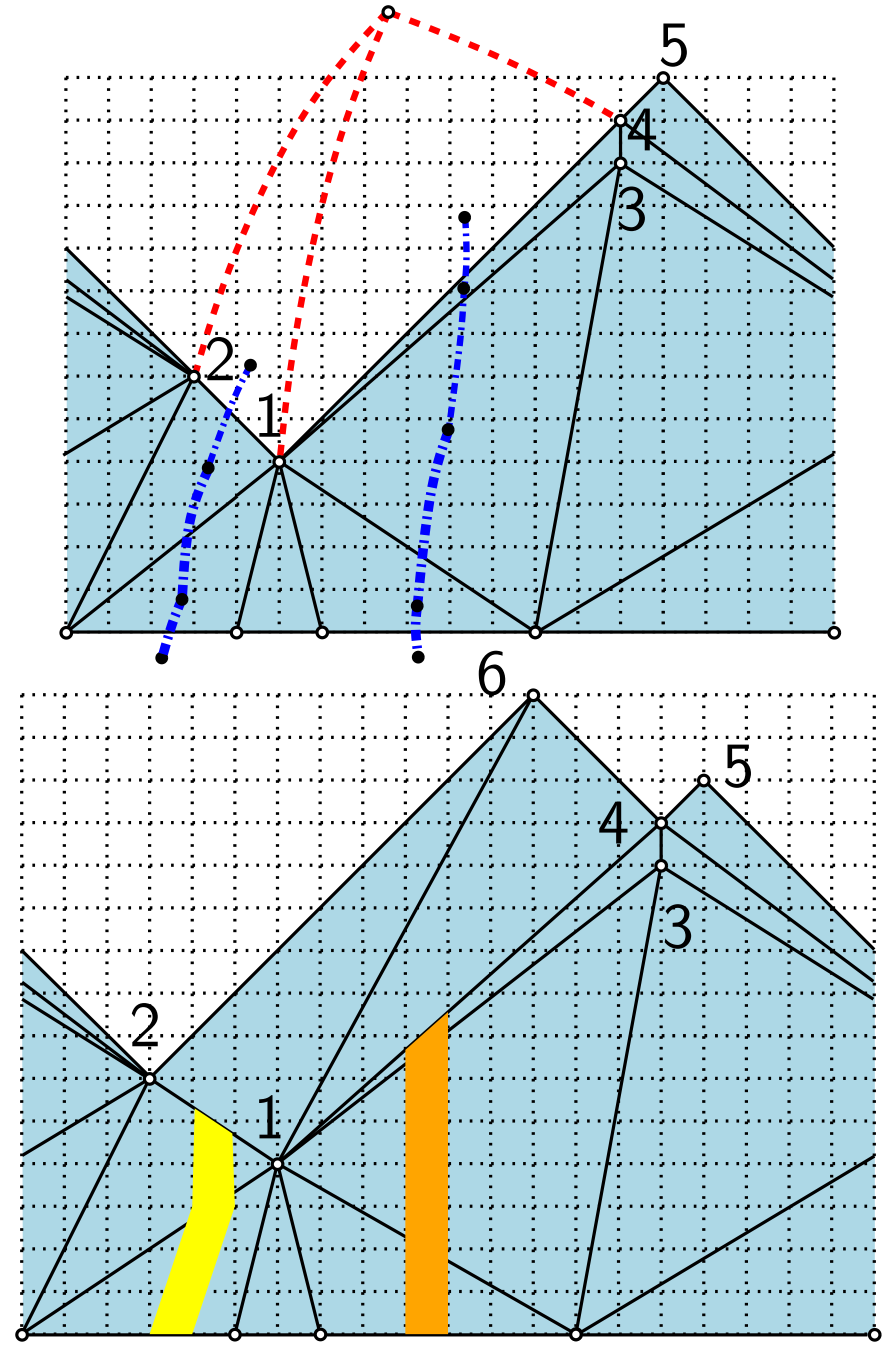
Extension to the cylinder: drawing algorithm



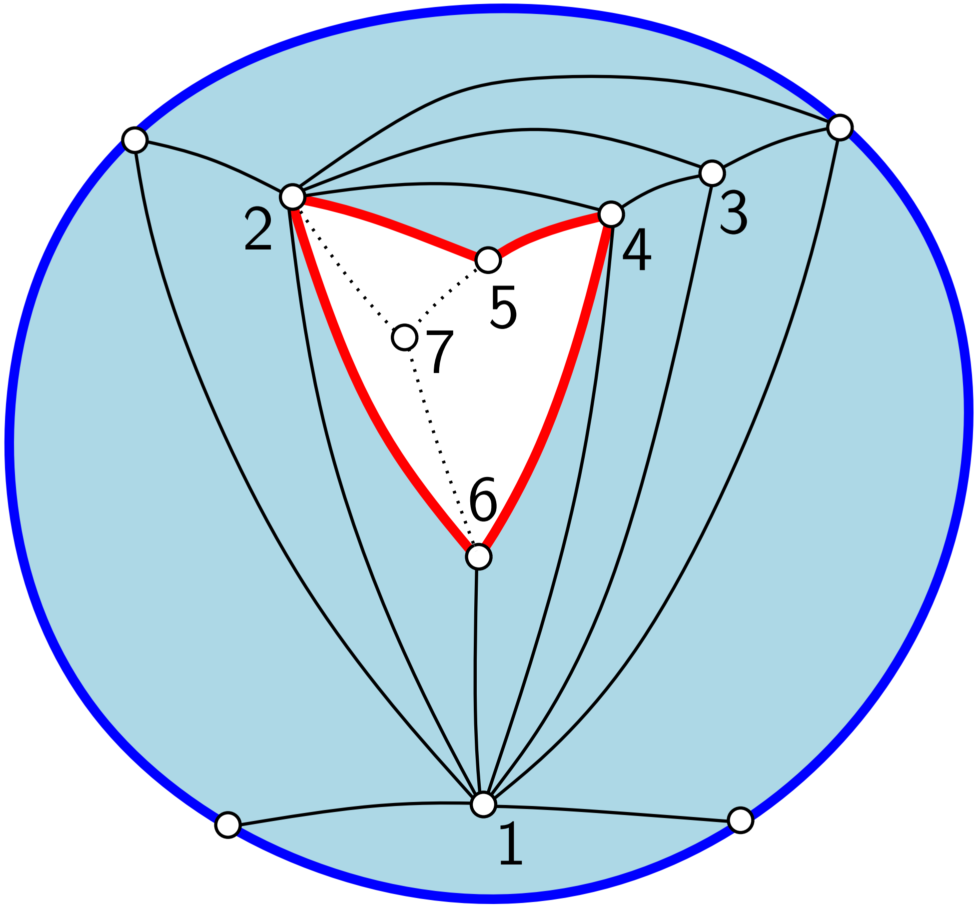


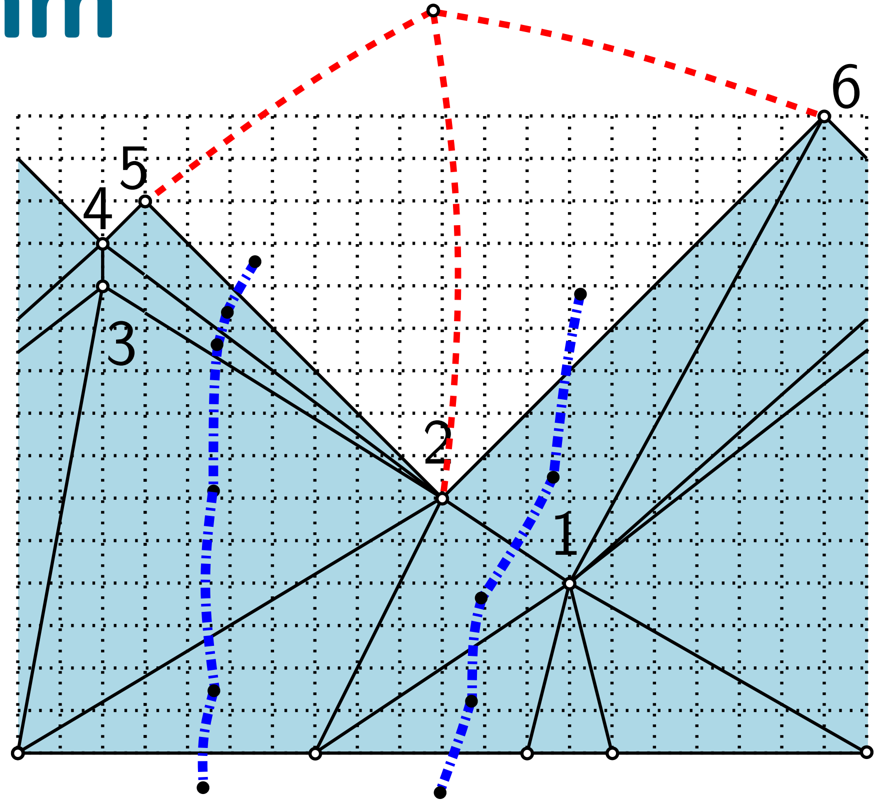
Extension to the cylinder: drawing algorithm



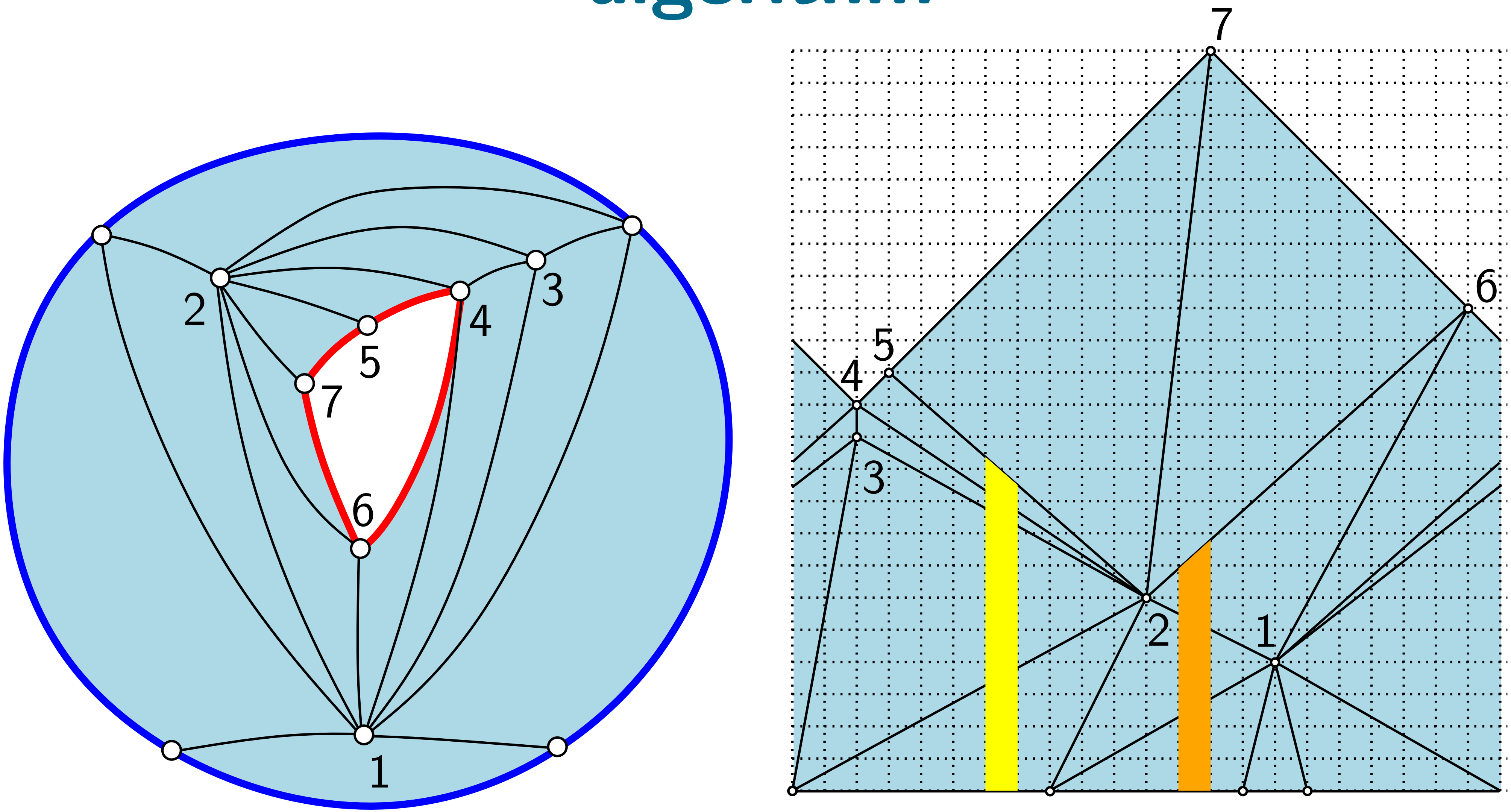


Extension to the cylinder: drawing algorithm





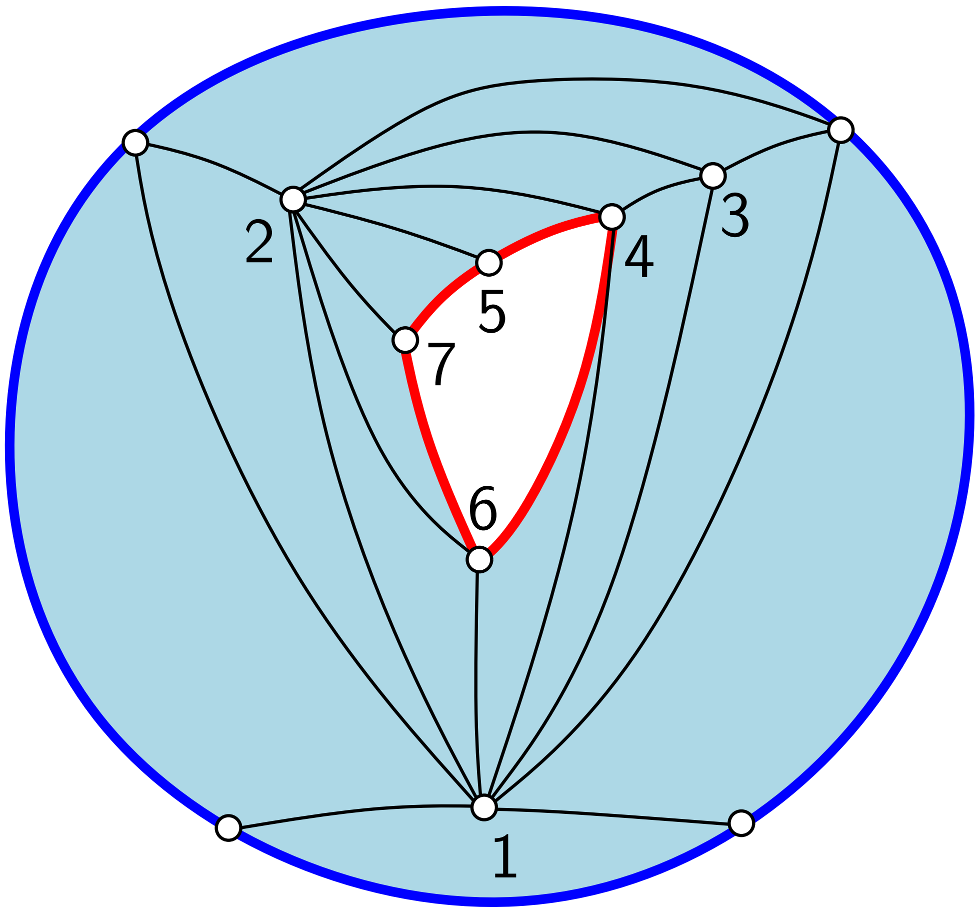
Extension to the cylinder: drawing algorithm

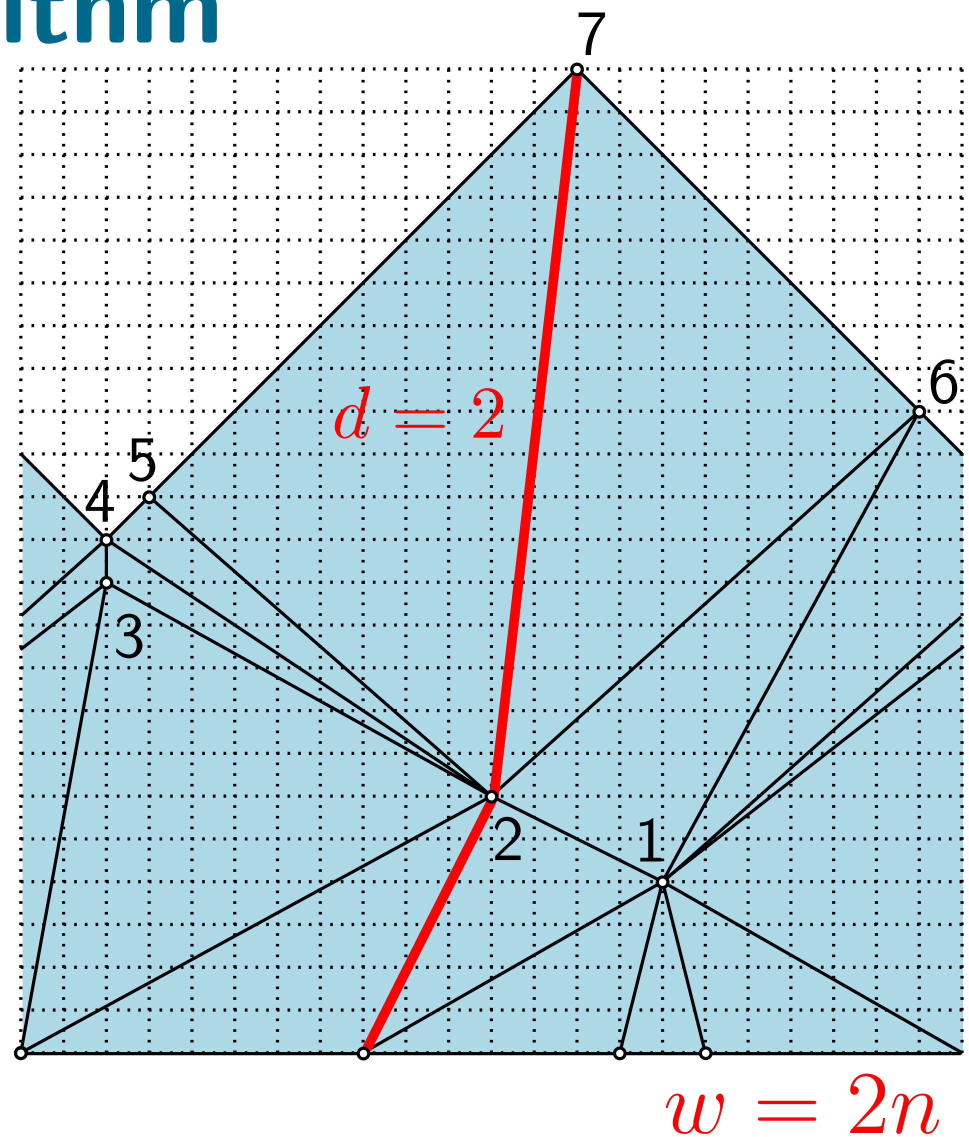


Width $= 2n$ Height $\leq n(n-3)/2$

Can also deal with chordal edges incident to outermost cycle

Extension to the cylinder: drawing algorithm





Each edge has vertical extension at most w

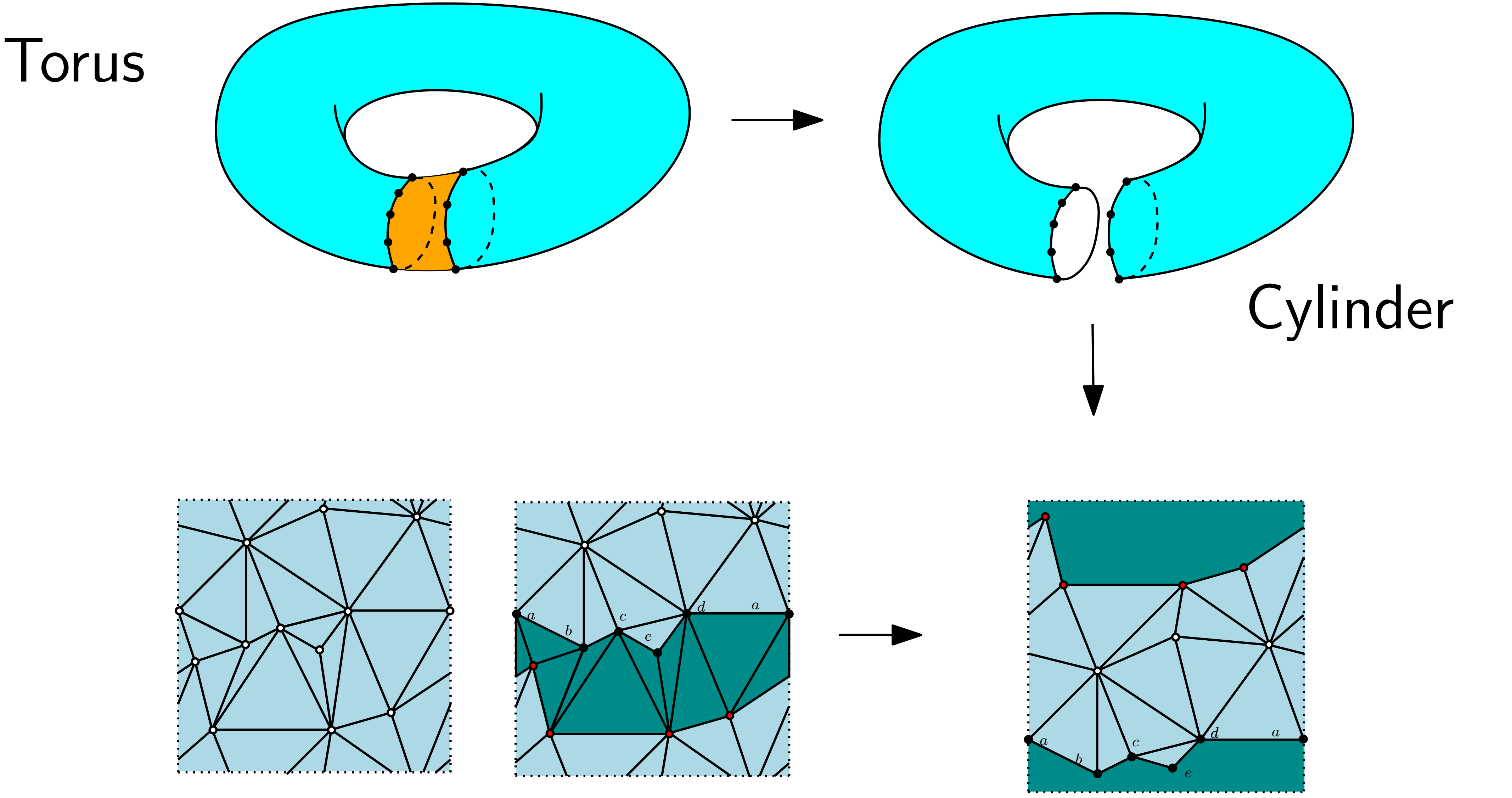
$$\Rightarrow h \leq n(2d+1)$$

with d the graph-distance between the two boundaries

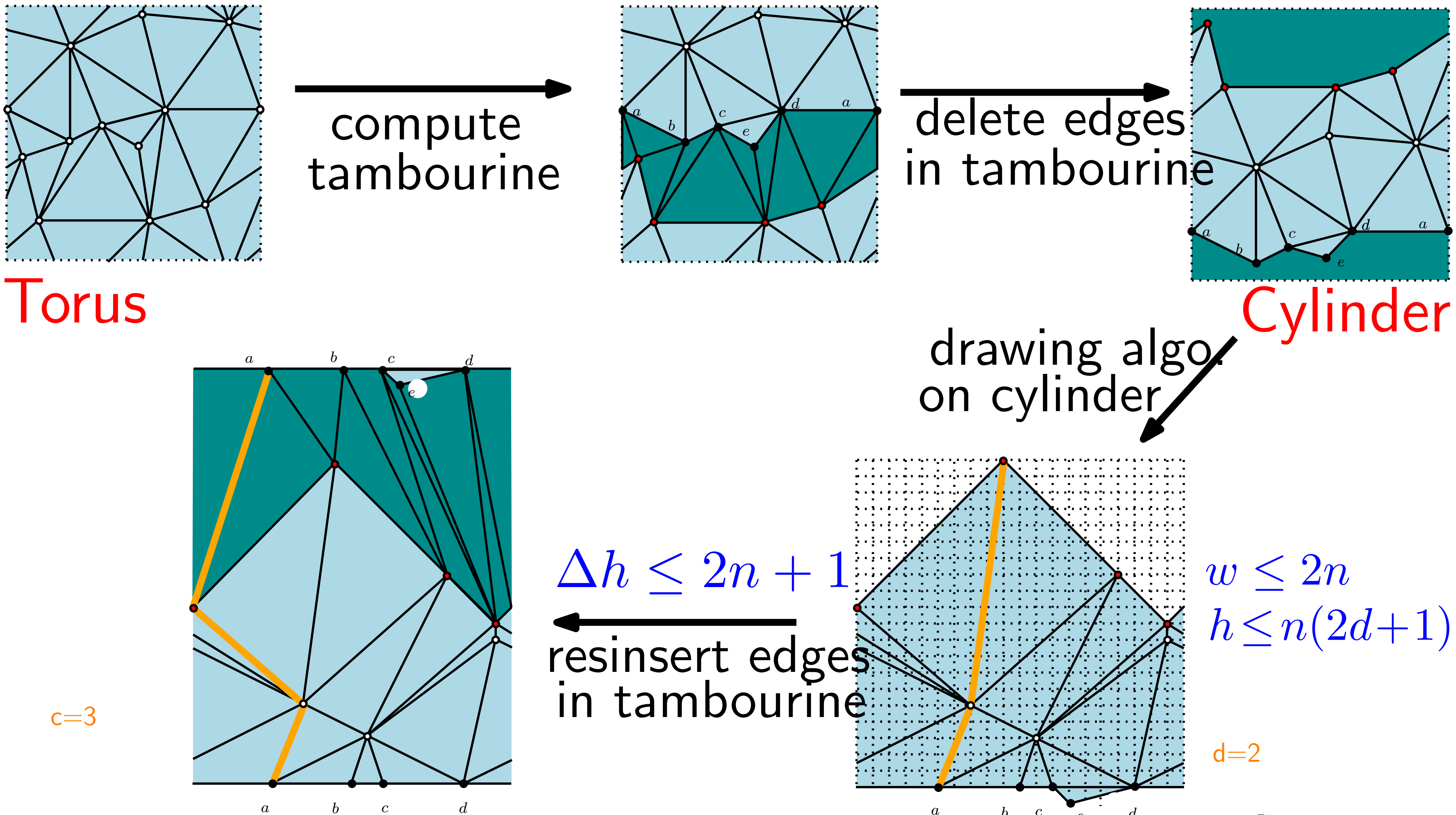
Getting toroidal drawings

Every toroidal triangulation admits a "tambourine"

[**Bonichon, Gavoille, Labourel'06**]



Getting toroidal drawings



Let $c =$length shortest non-contractible cycle, $c \leq \sqrt{2n}$ **[Hutchinson, Albert'78]**

Can choose tambourine so that $d < c \Rightarrow h = O(n^{3/2})$

Schnyder woods for toroidal graphs

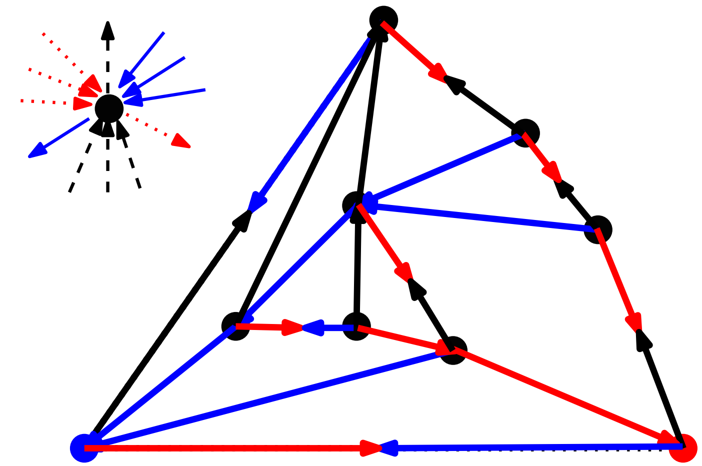
Toroidal Schnyder woods: definition

[Goncalves Lévêque, DCG'14]

$g = 1 \qquad e = 3n$

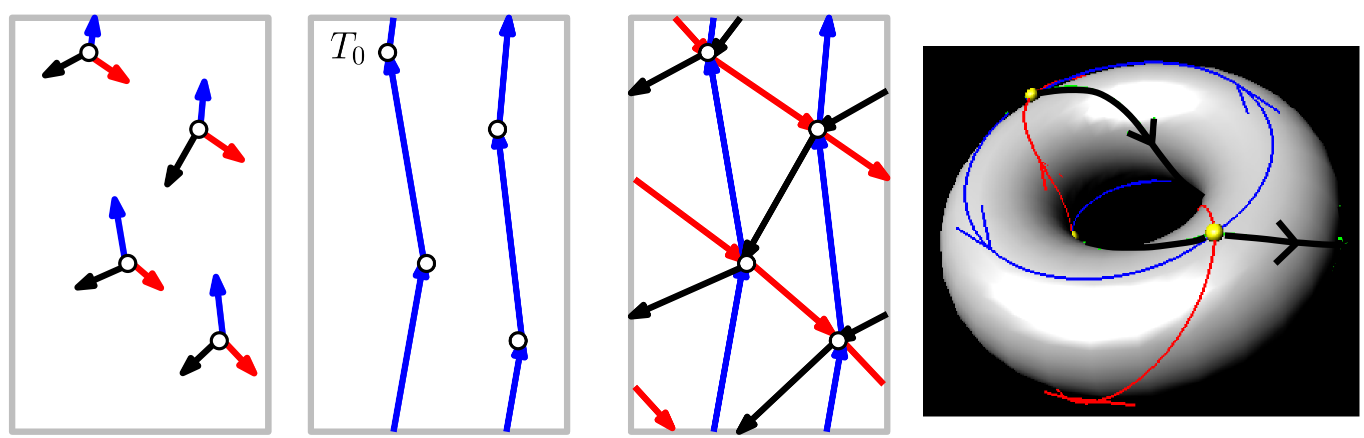
Planar Schnyder woods [Felsner 2001]

- Schnyder local rule (for half-edges)
- no monochromatic cycles



Toroidal Schnyder woods [Goncalves Lévêque, DCG'14]

- Schnyder local rule (for half-edges)
- every monochromatic cycle intersects at least one monochromatic cycle of each color

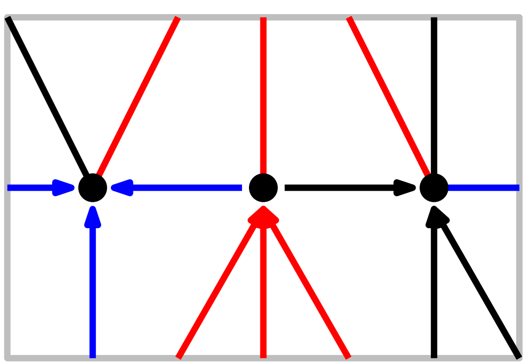


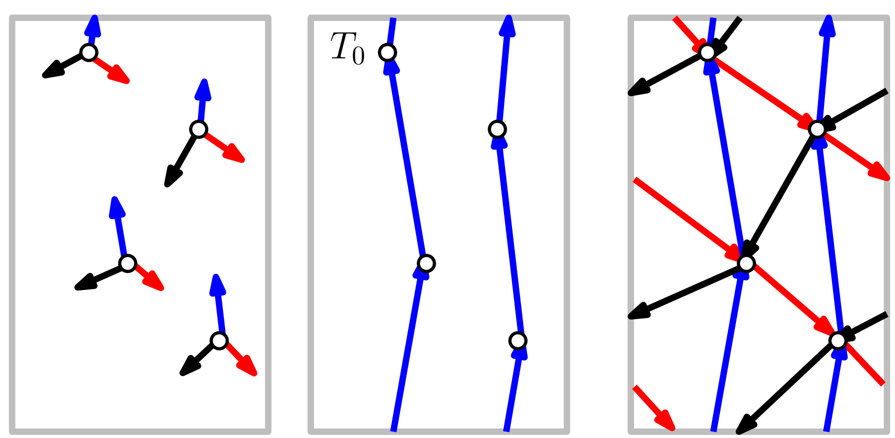
Toroidal Schnyder woods: existence

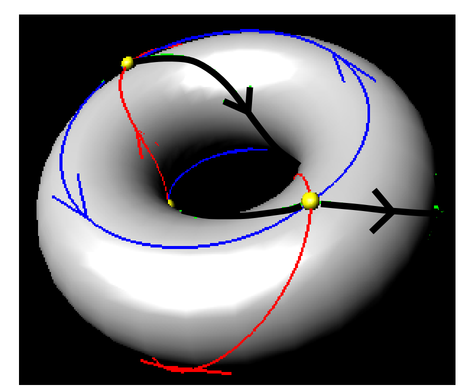
[Goncalves Lévêque, DCG'14]

$g = 1 \quad e = 3n$

no pair of intersecting monochromatic cycles



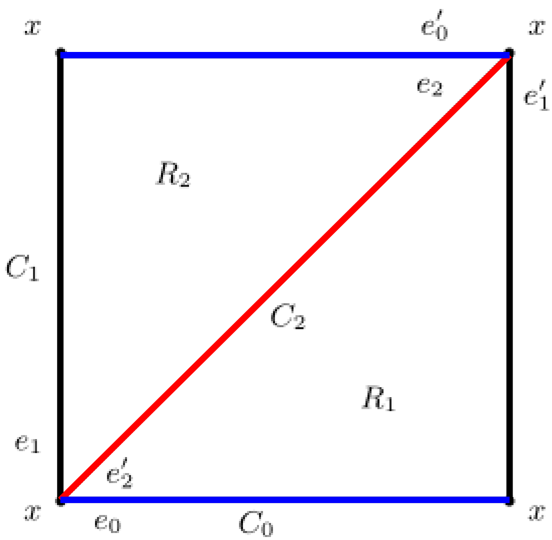


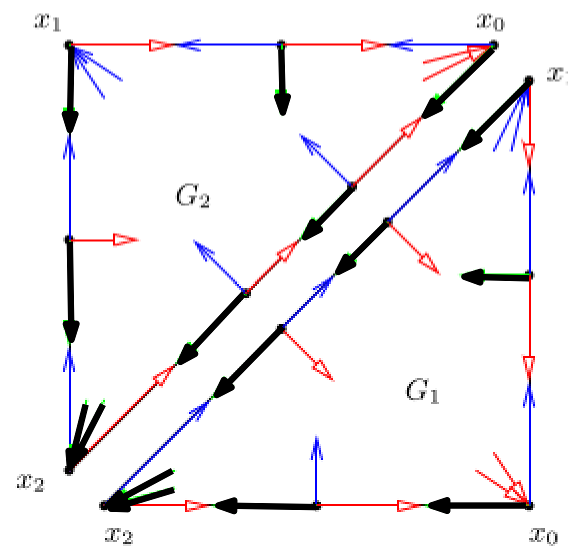


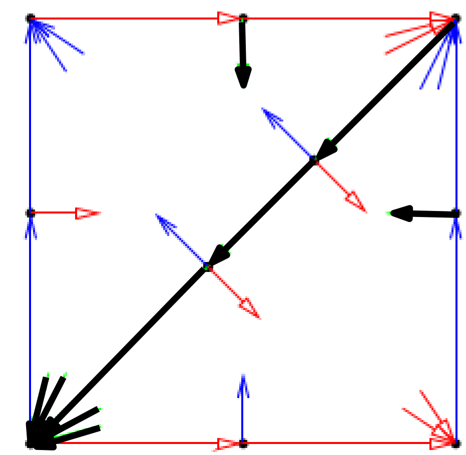
Toroidal Schnyder woods: existence

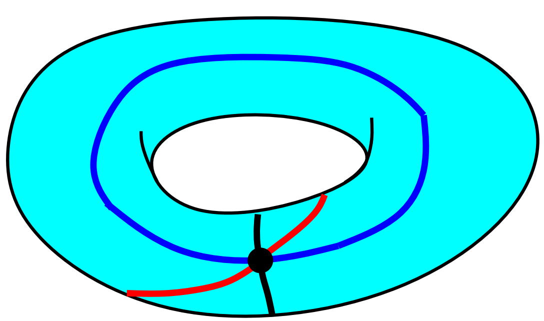
Thm[Fijavz] (planar simple triangulations)

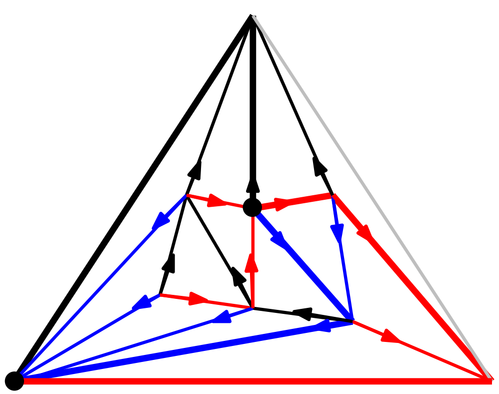
A simple toroidal triangulation contains three non-contractible and non-homotopic cycles that all intersect on one vertex and that are pairwise dis- joint otherwise.











Toroidal Schnyder woods: drawing

(planar simple triangulations)

Thm[Goncalves Lévêque]

A simple toroidal triangulation admits a straight-line periodic drawing on a grid of size $O(n^2 \times n^2)$