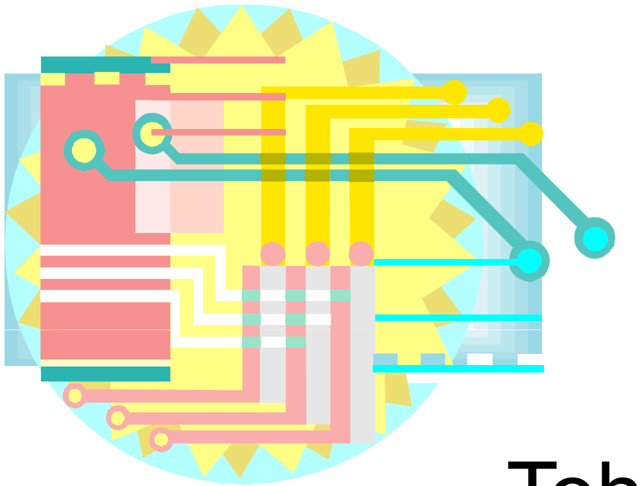
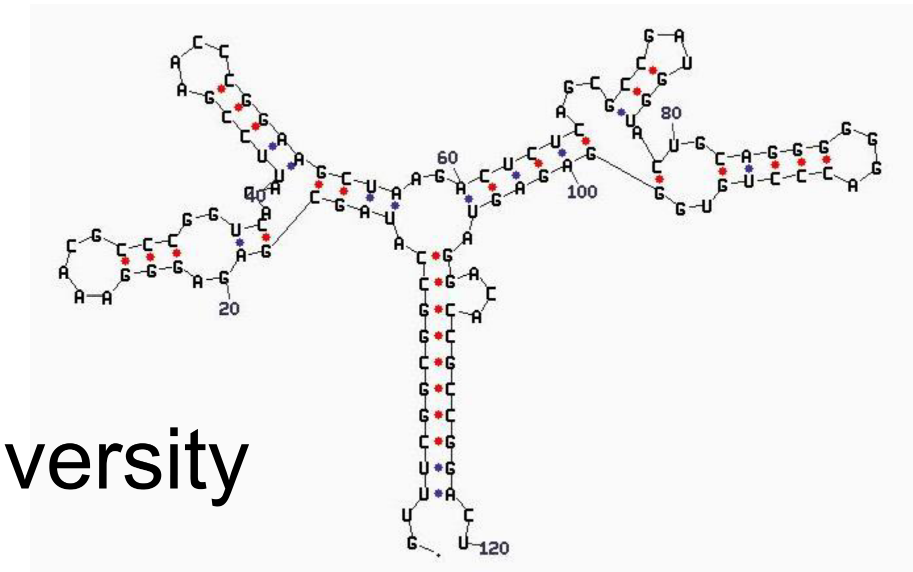
Drawing Plane Graphs

Takao Nishizeki

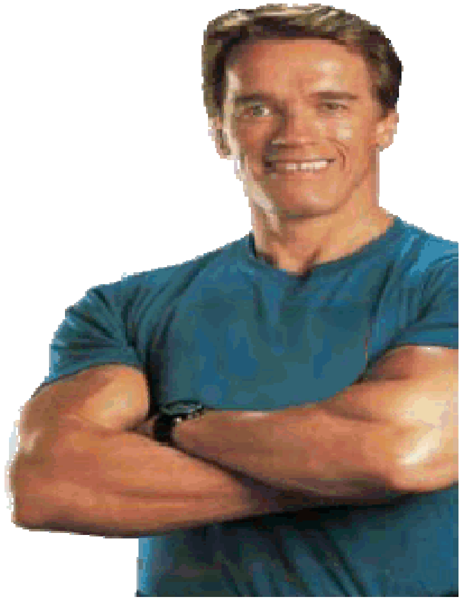




Tohoku University

White House photo by Eric Draper

US President



California Governor

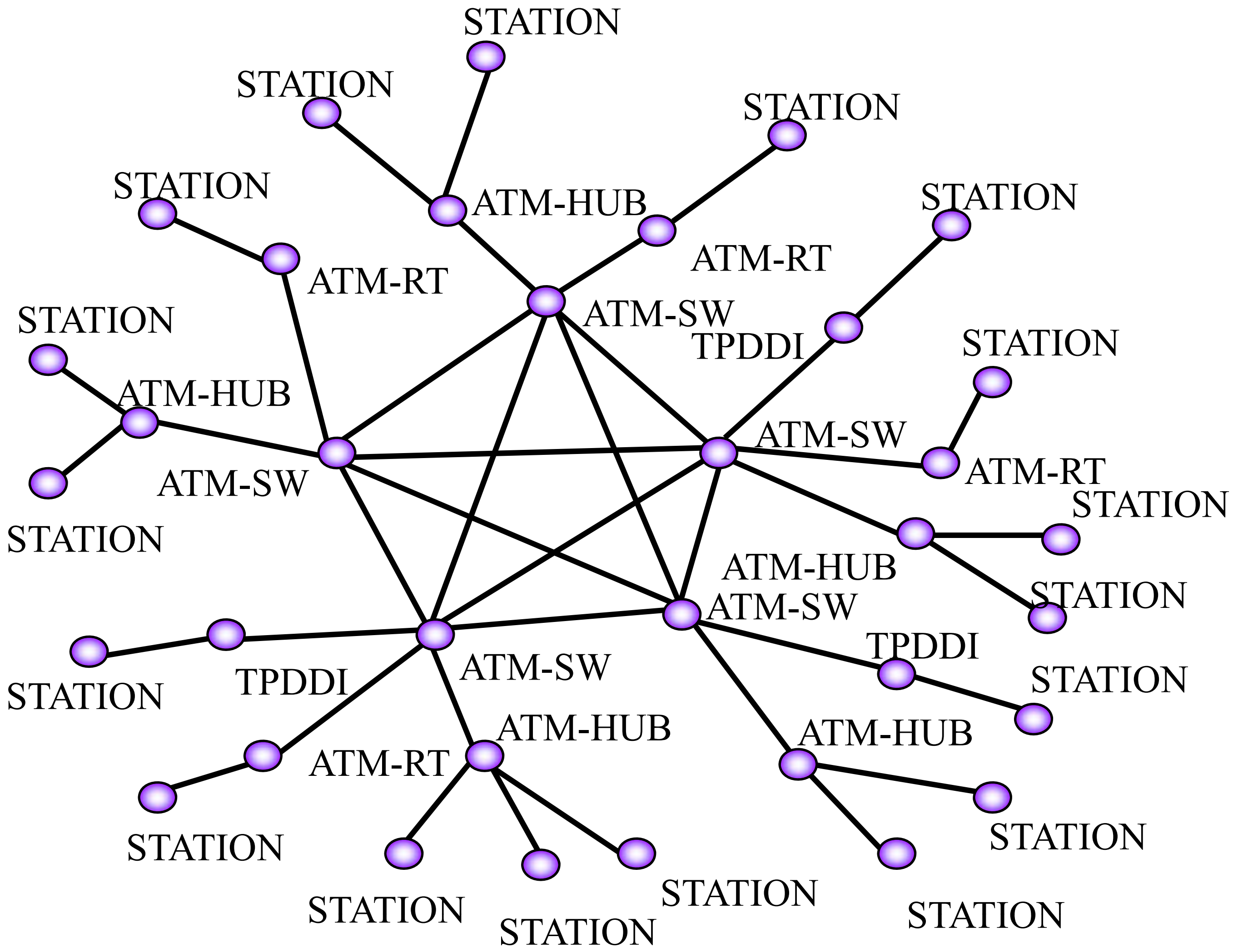
White House photo by Eric Draper

US President

California Governor

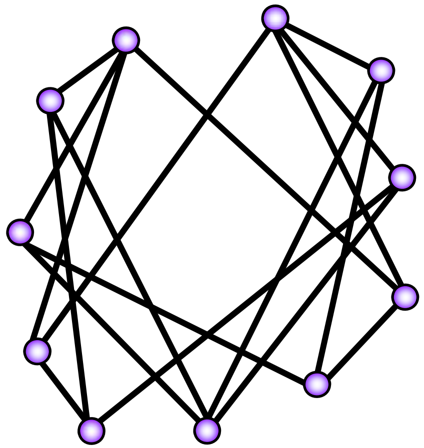
What is the common feature?

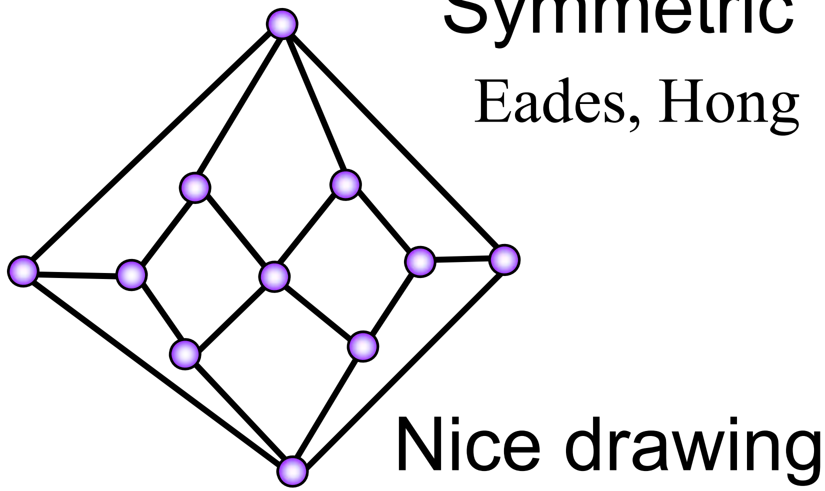
Graphs and Graph Drawings



A diagram of a computer network

Objectives of Graph Drawings



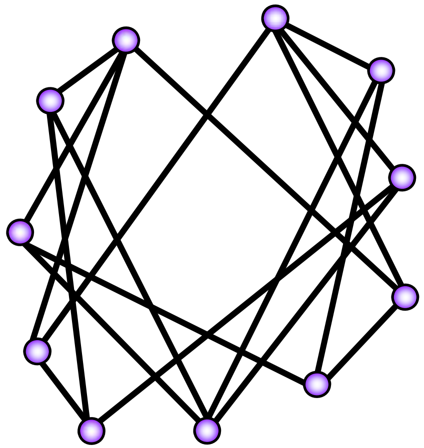


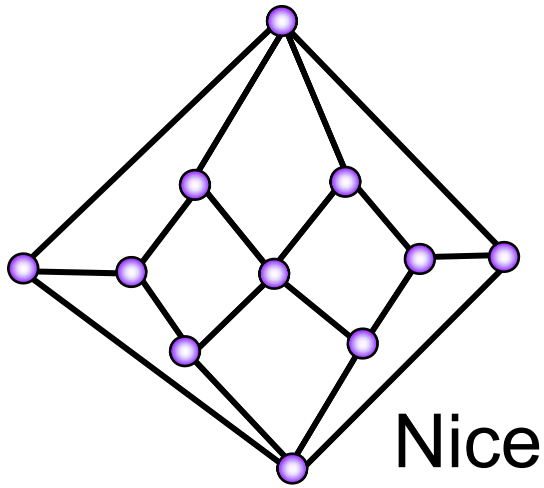
structure of the graph is difficult to understand

structure of the graph is easy to understand

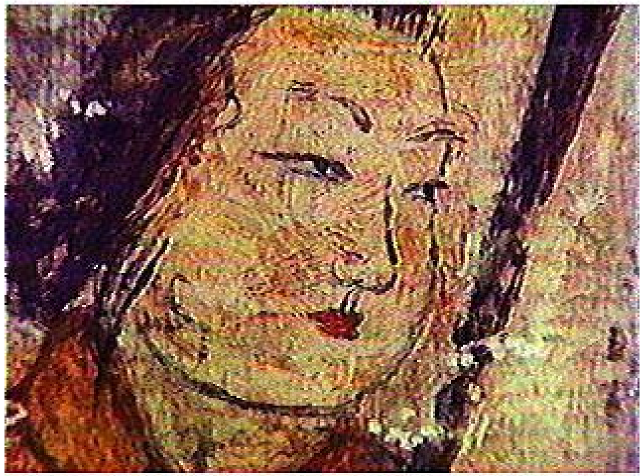
- To obtain a nice representation of a graph so that the structure of the graph is easily understandable.

Objectives of Graph Drawings





Nice drawing

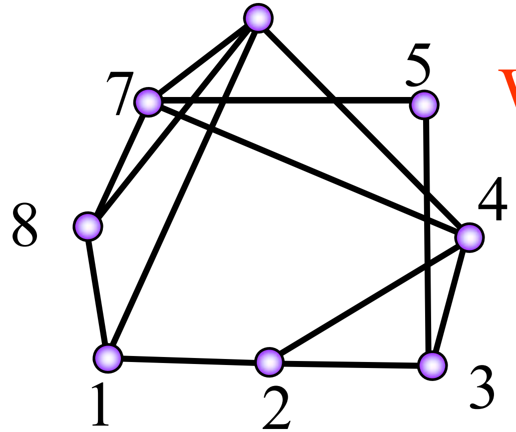


Ancient beauty

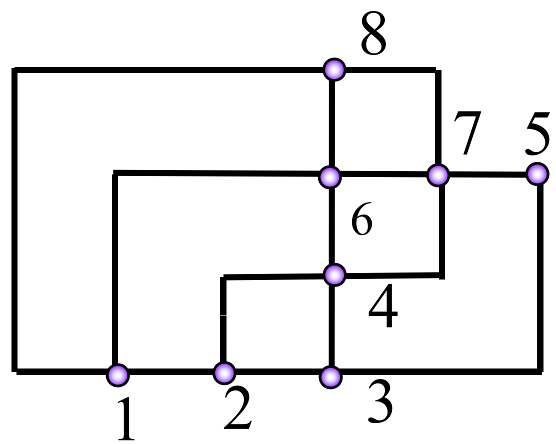
Modern beauty

Objectives of Graph Drawings

Diagram of an electronic circuit



Wire crossings



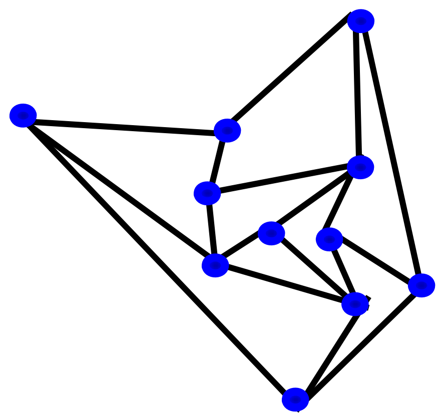
not suitable for single layered PCB

suitable for single layered PCB

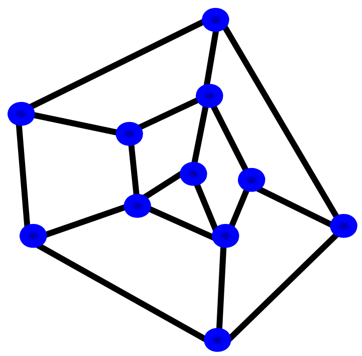
- The drawing should satisfy some criterion arising from the application point of view.

Drawings of Plane Graphs

- Straight line drawing

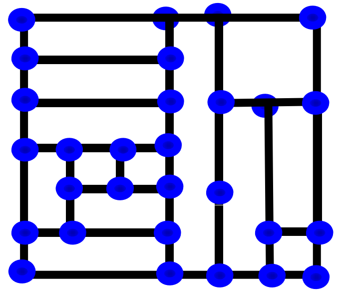


- Convex drawing

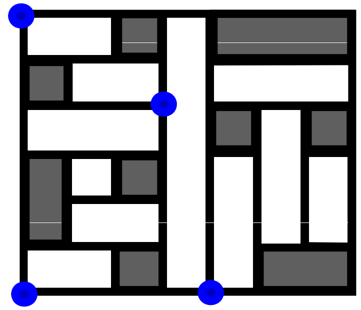


Drawings of Plane Graphs

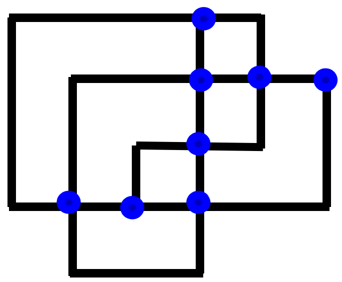
- Rectangular drawing



- Box-rectangular drawing



- Orthogonal drawing



Book

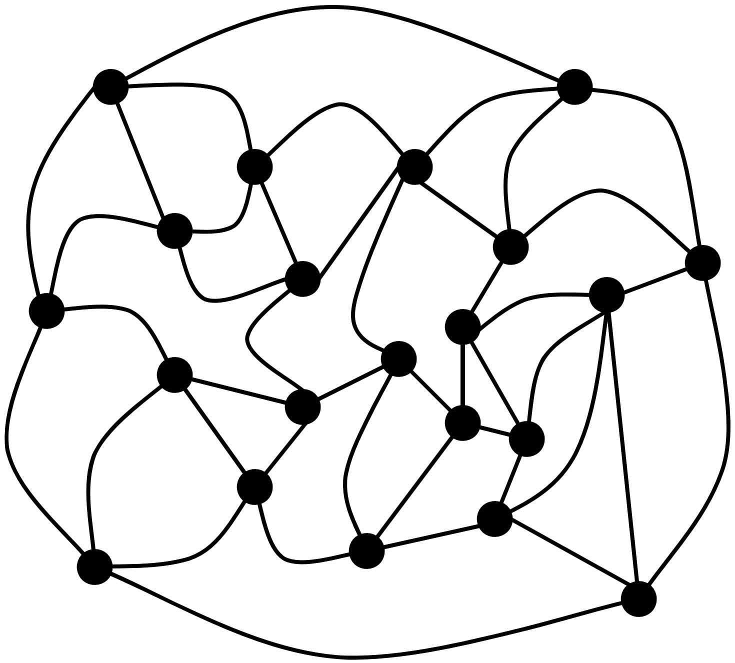
Planar Graph Drawing

by

Takao Nishizeki
Md. Saidur Rahman

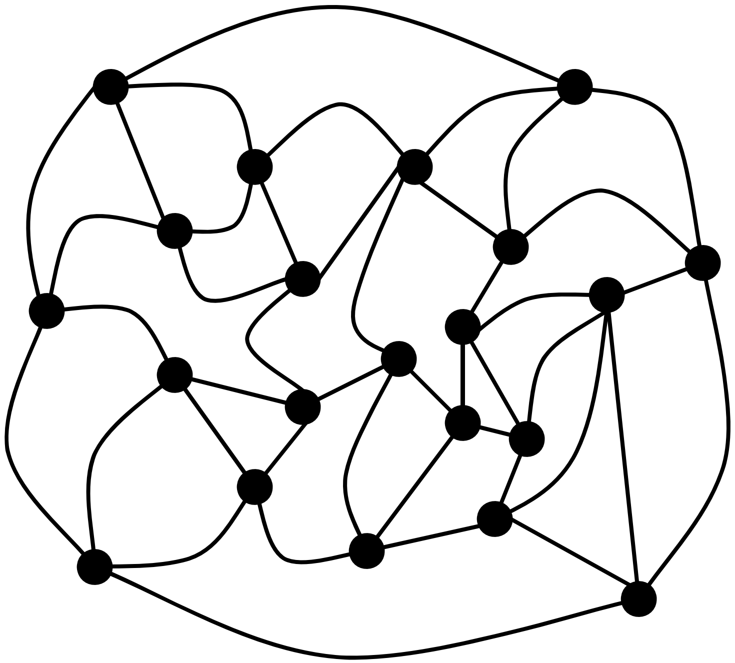
http://www.nishizeki.ecei.tohoku.ac.jp/nszk/saidur/gdbook.html

Straight Line Drawing

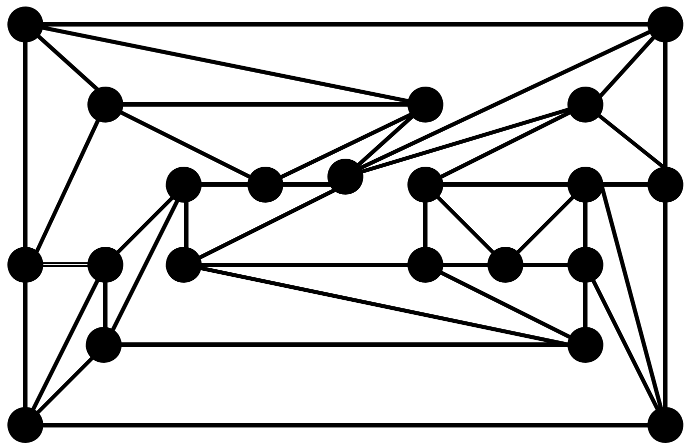


Plane graph

Straight Line Drawing

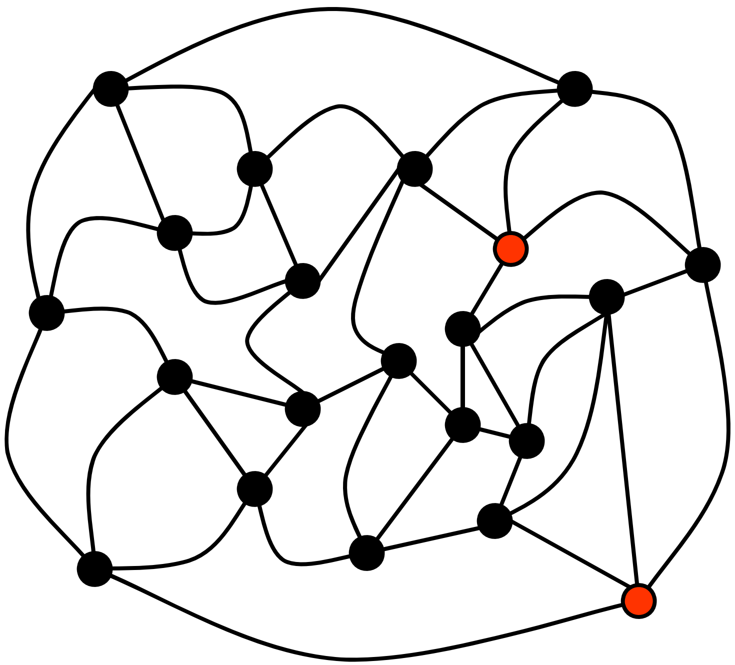


Plane graph

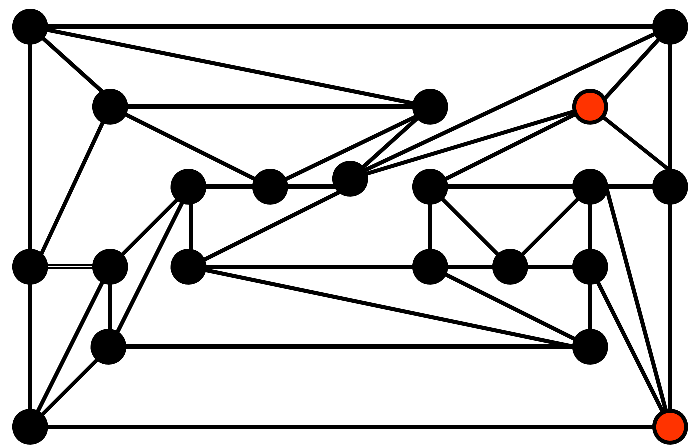


Straight line drawing

Straight Line Drawing



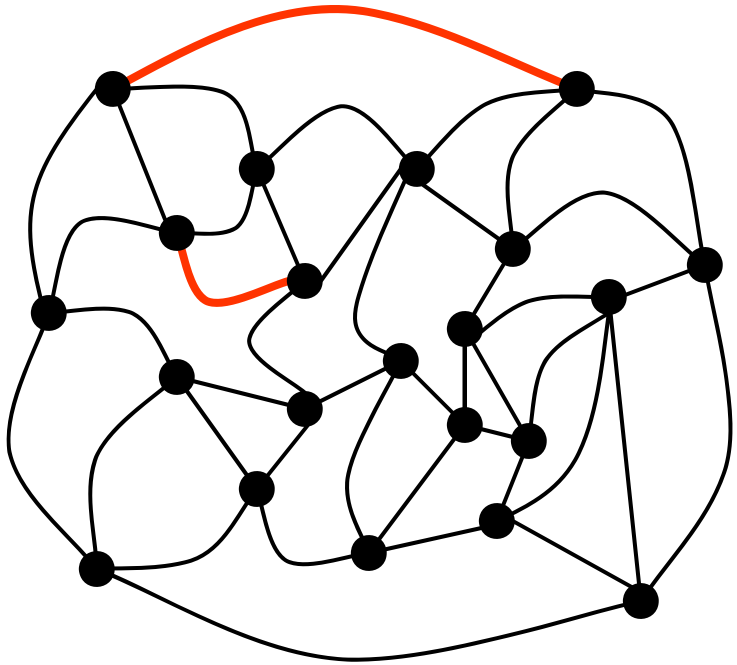
Plane graph



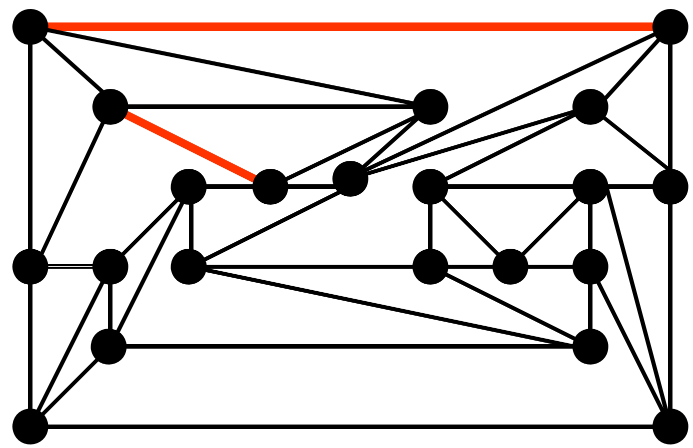
Straight line drawing

Each vertex is drawn as a point.

Straight Line Drawing



Plane graph



Straight line drawing

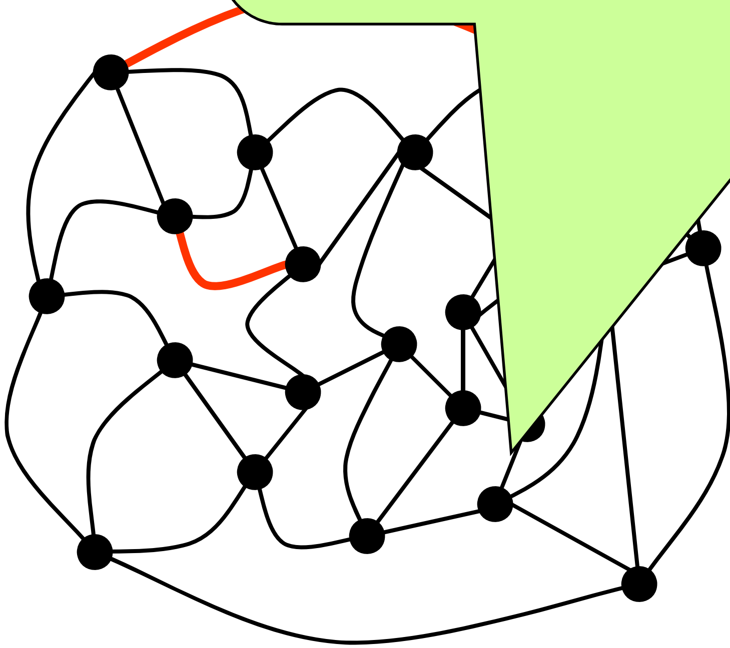
Each vertex is drawn as a point.

Each edge is drawn as a single straight line segment.

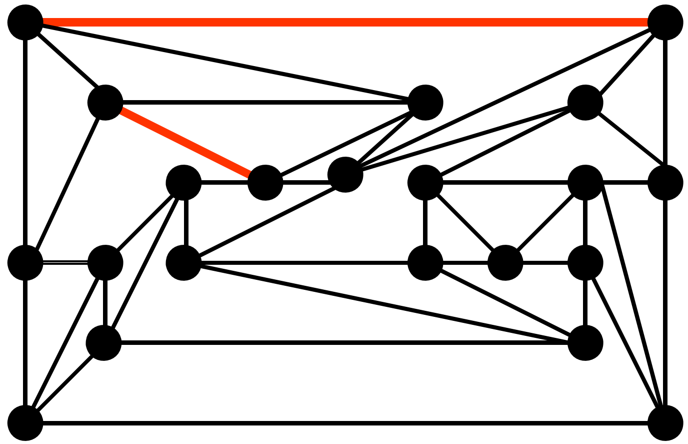
Every plane graph has a straight line drawing.

Wagner ’36   Fary ’48

Polynomial-time algorithm



Plane graph

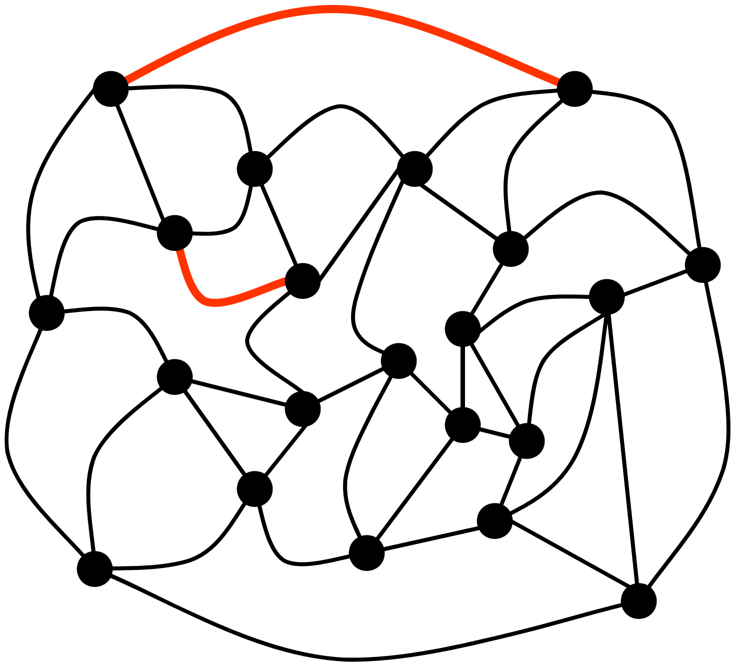


Straight line drawing

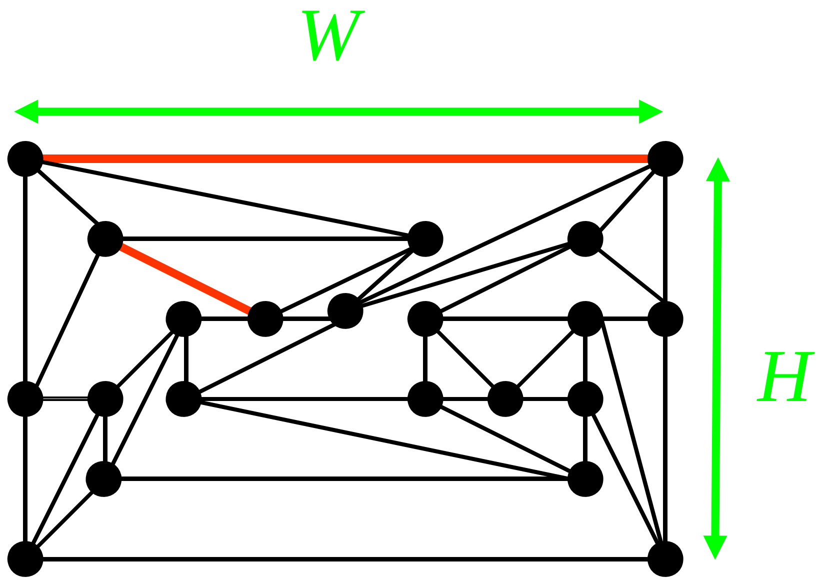
Each vertex is drawn as a point.

Each edge is drawn as a single straight line segment.

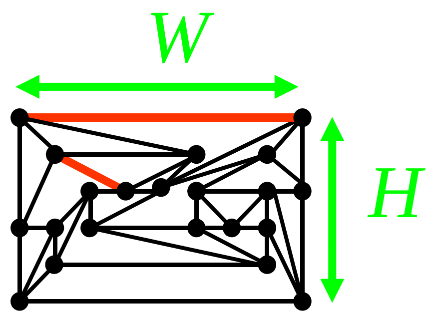
Straight Line Drawing



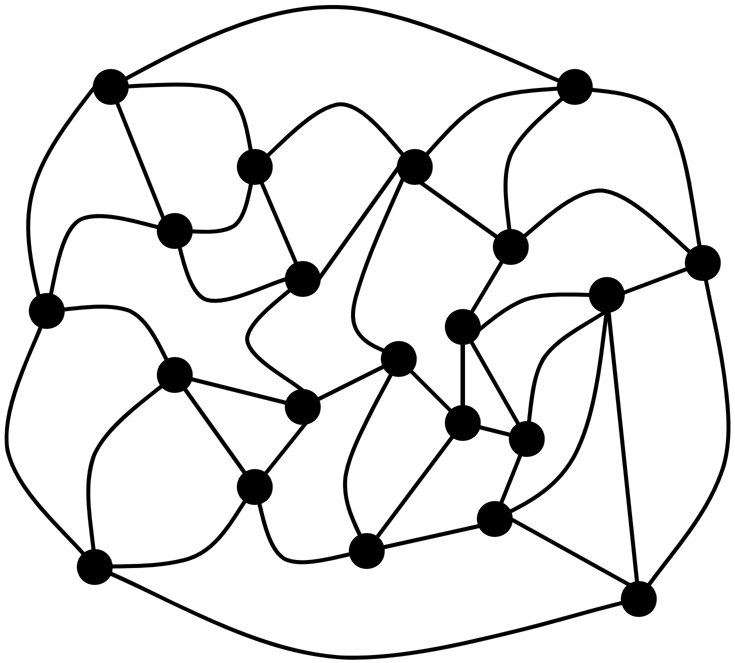
Plane graph



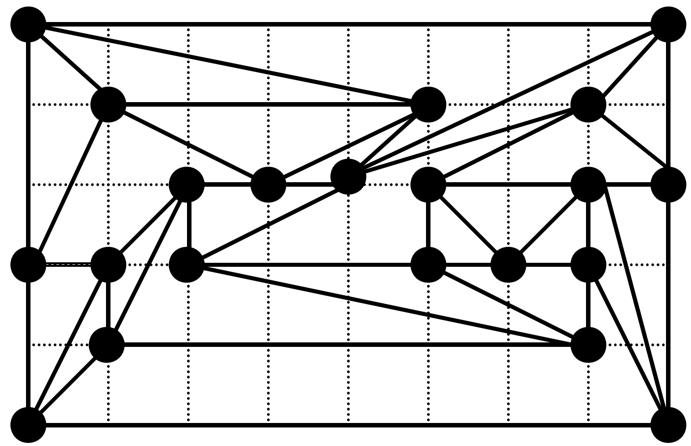
Straight line drawing



Straight Line Grid Drawing



Plane graph

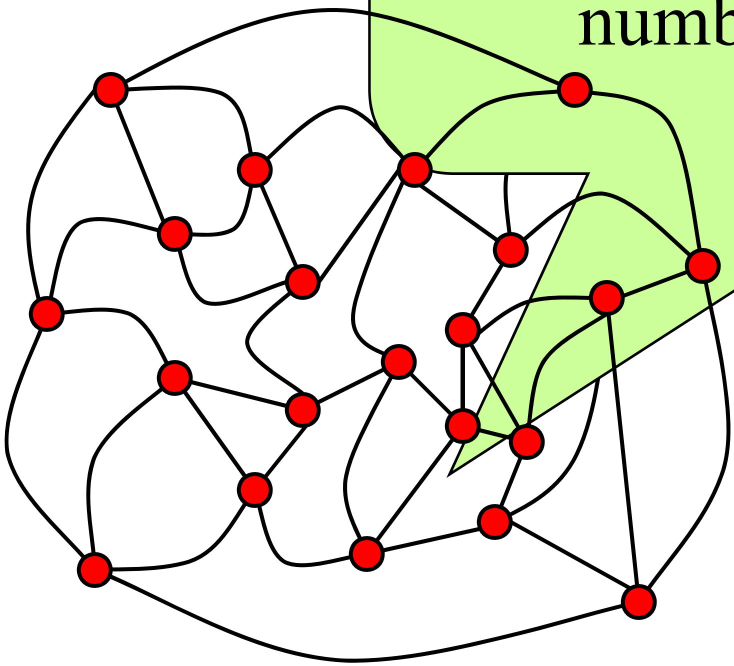


Straight line grid drawing.

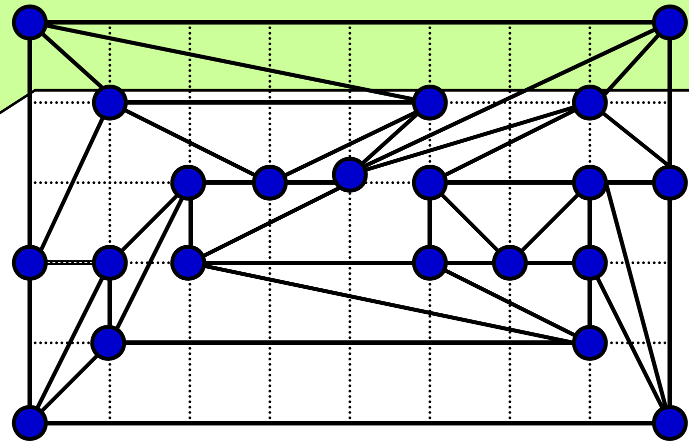
In a straight line grid drawing each vertex is drawn on a grid point.

Wagner '36 Fary '48

Grid-size is not polynomial of the number of vertices *n*

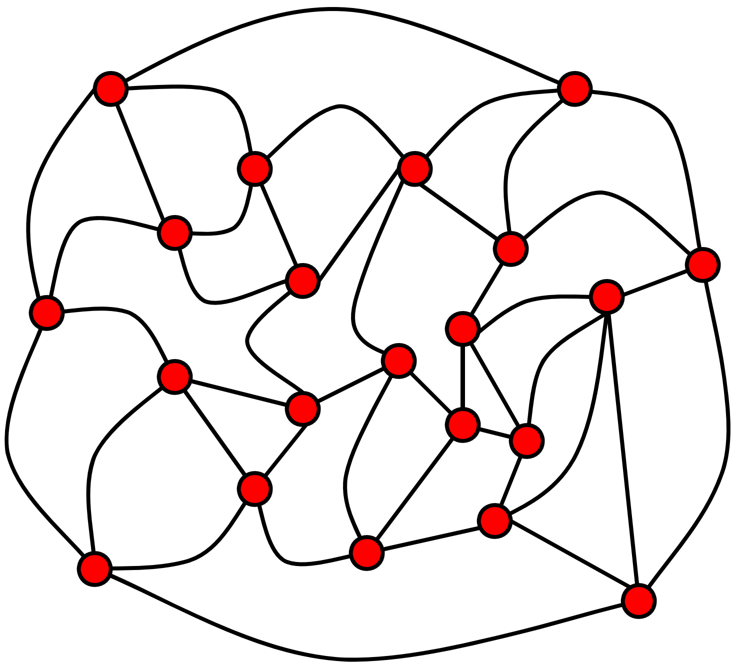


Plane graph



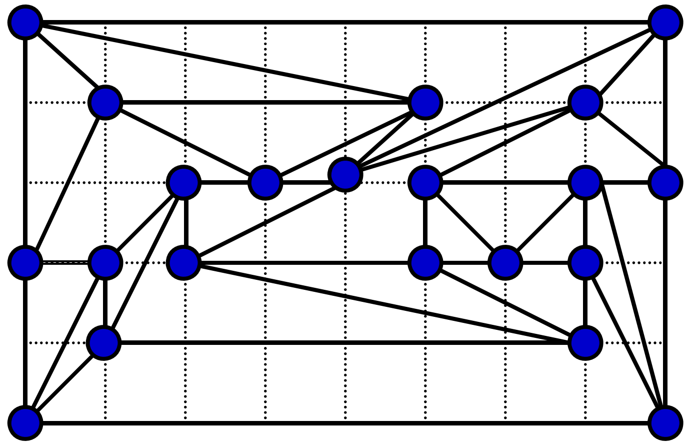
Straight line grid drawing.

Straight Line Grid Drawing



Plane graph

$W \le 2n$



$H \le n$

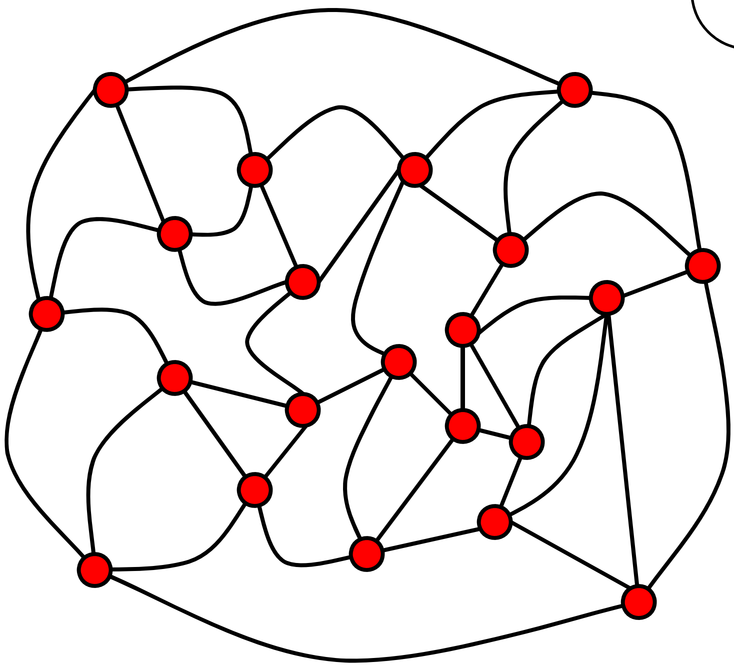
Straight line grid drawing.

de Fraysseix *et al.* '90

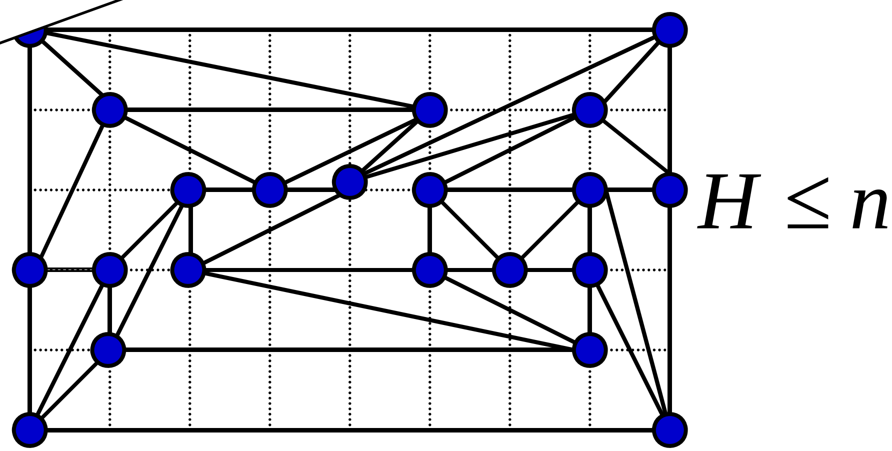
$$W \times H \le 2n^2$$

Chrobak and Payne ’95

Linear algorithm



Plane graph

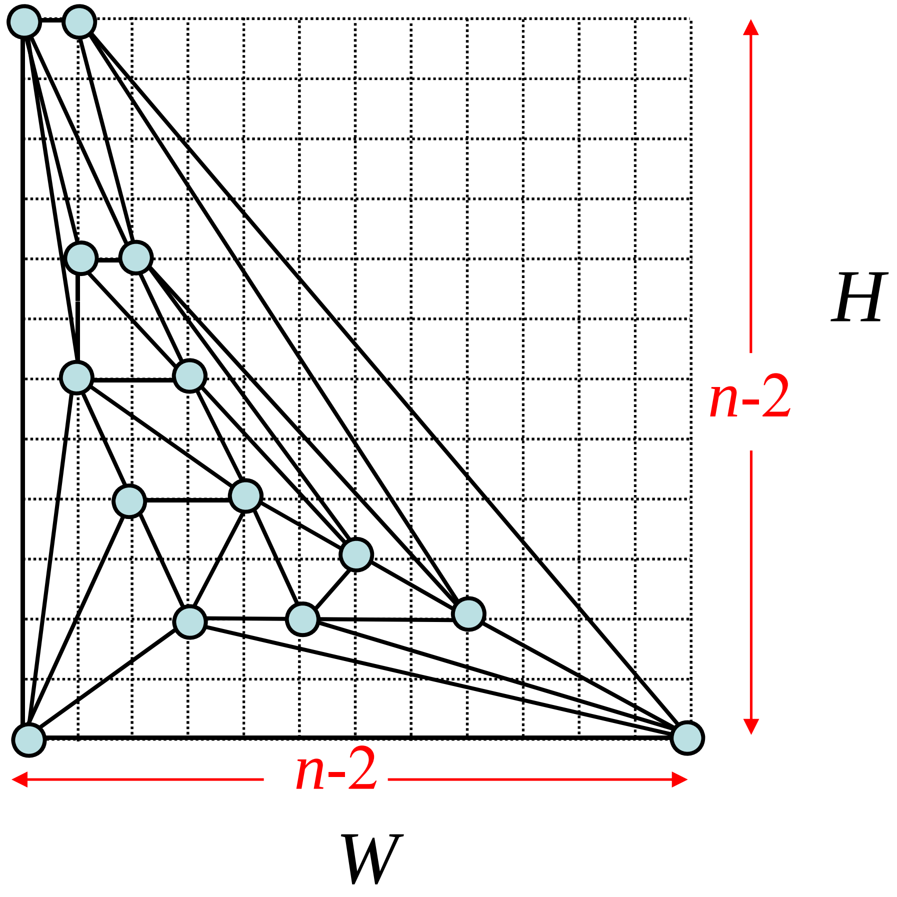


Straight line grid drawing.

de Fraysseix *et al.* ’90

$$W \times H \leq 2n^2$$

Schnyder '90

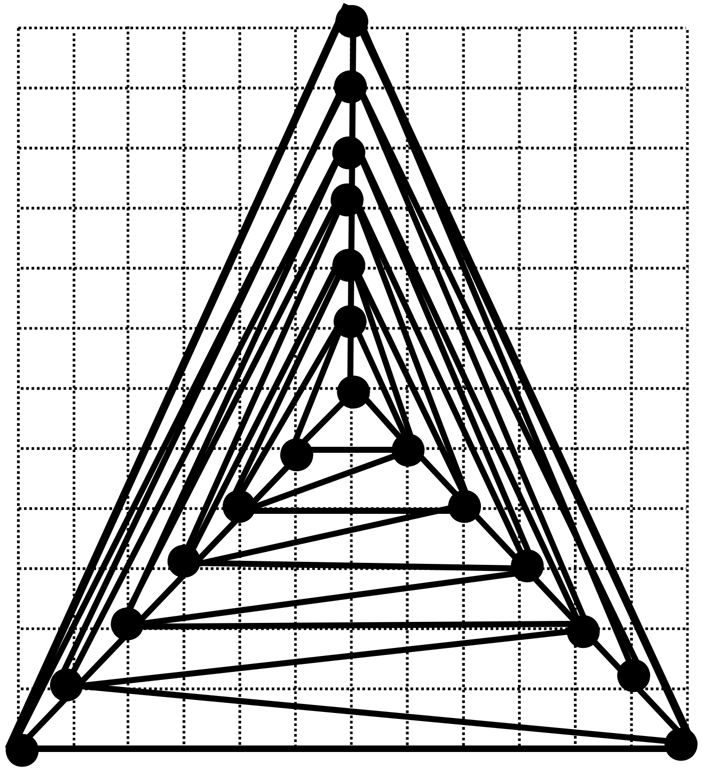


$$W \times H \leq n^2$$

Upper bound

What is the minimum size of a grid required for a straight line drawing?

Lower Bound



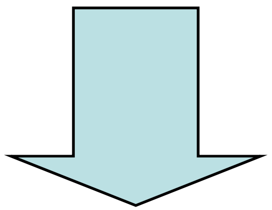
$$H \geq \frac{2}{3}n$$

$$W \geq \frac{2}{3}n$$

$$W \times H \geq \left(\frac{2}{3}n\right)^2$$

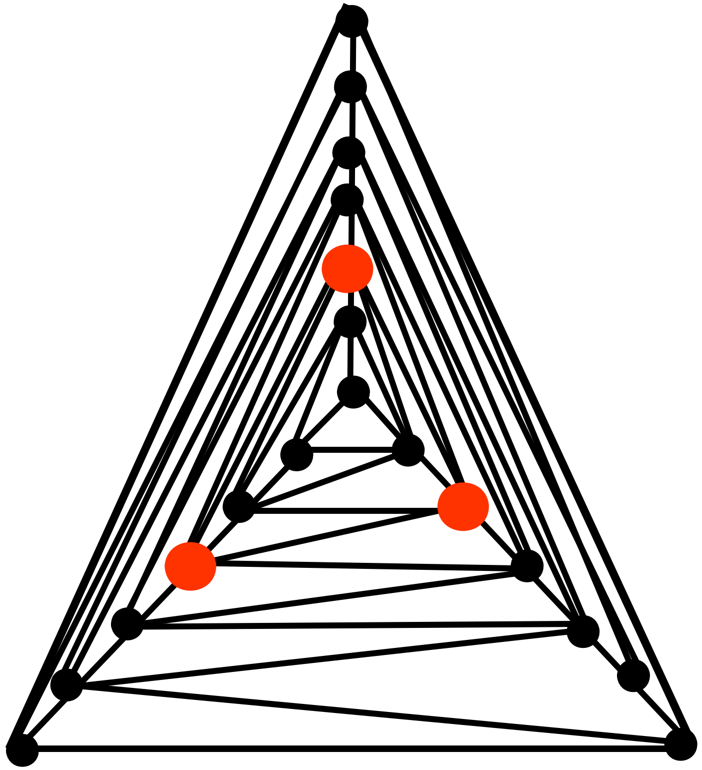
A restricted class of plane graphs may have more compact grid drawing.

Triangulated plane graph

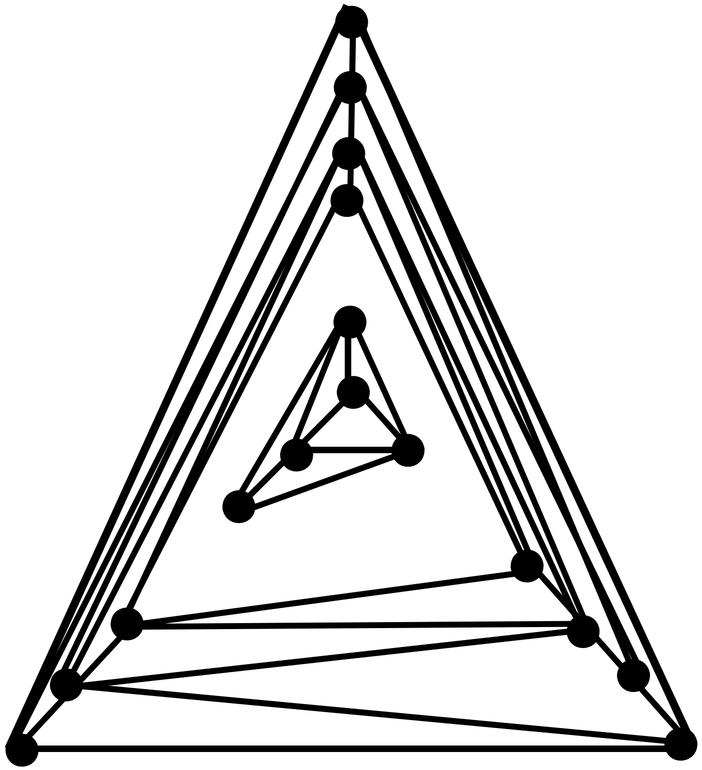


3-connected graph

4-connected ?



not 4-connected



disconnected

How much area is required for 4-connected plane graphs?

Straight line grid drawing

Miura *et al.* '01

Input: 4-connected plane graph G

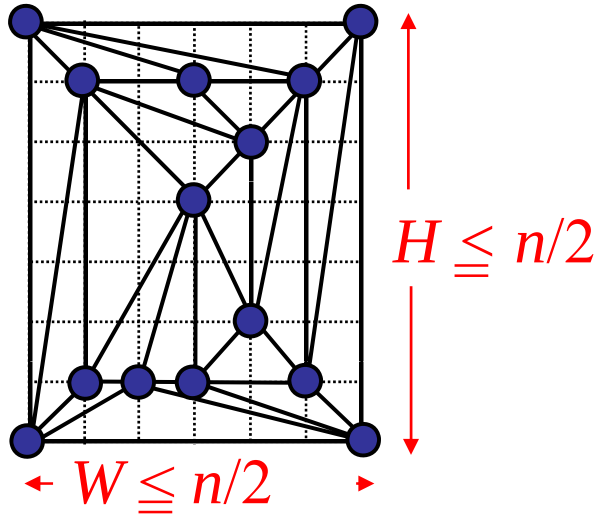
Output: a straight line grid drawing

Grid Size :

$$W, H \leq \frac{n}{2}$$

Area:

$$W \times H \leq \frac{n^2}{4}$$

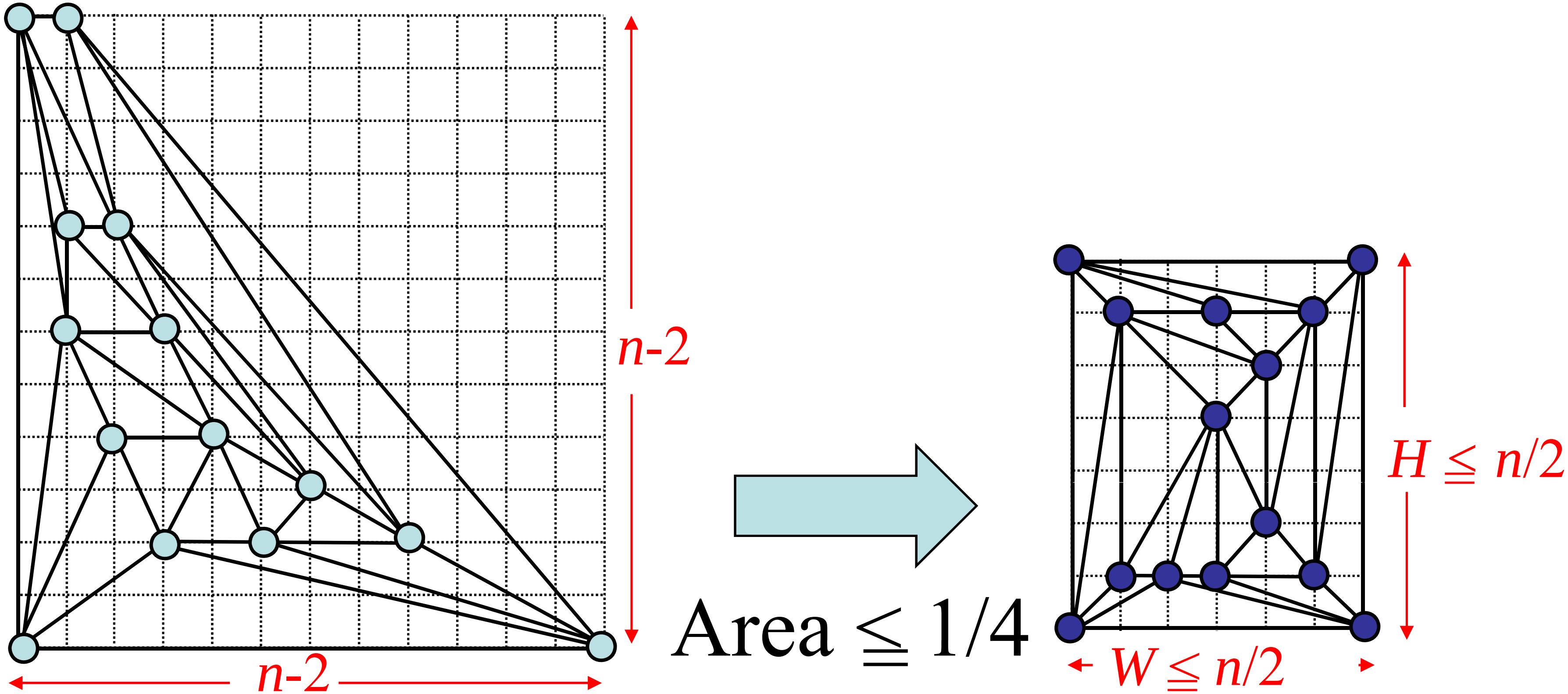


Schnyder ’90

plane graph *G*

Miura *et al.* ’01

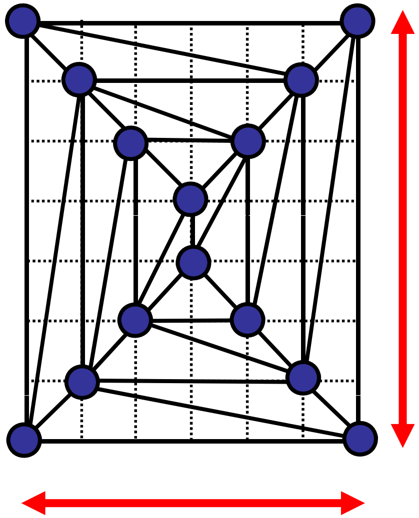
4-connected plane graph *G*



Area $\doteqdot n^2$

Area $\leqq n^2/4$

The algorithm of Miura *et al.* is
best possible



$$H \geq \frac{n}{2}$$

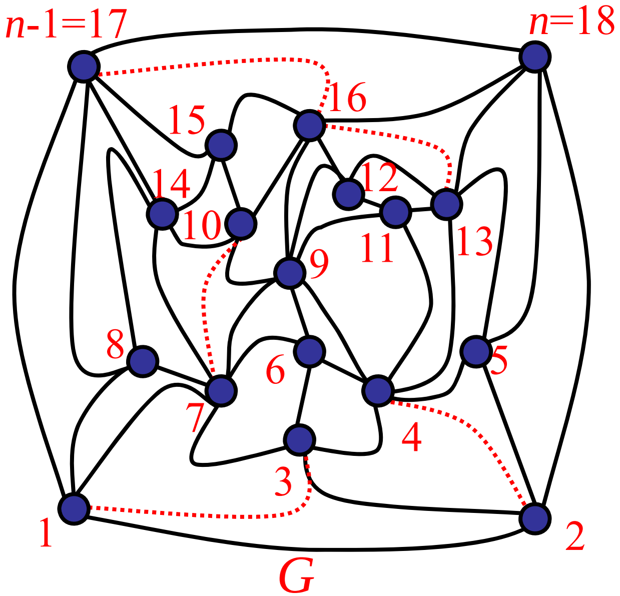
$$W \geq \frac{n}{2}$$

$$W \times H \geq \frac{n^2}{4}$$

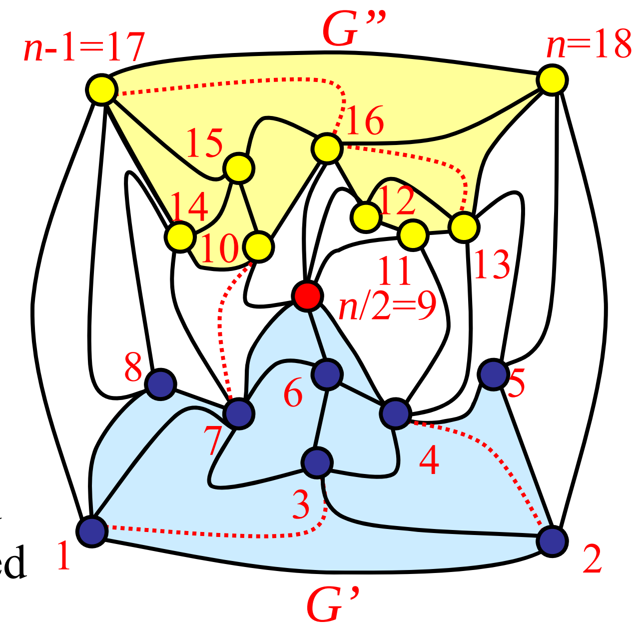
Main idea

Triangulate all inner faces

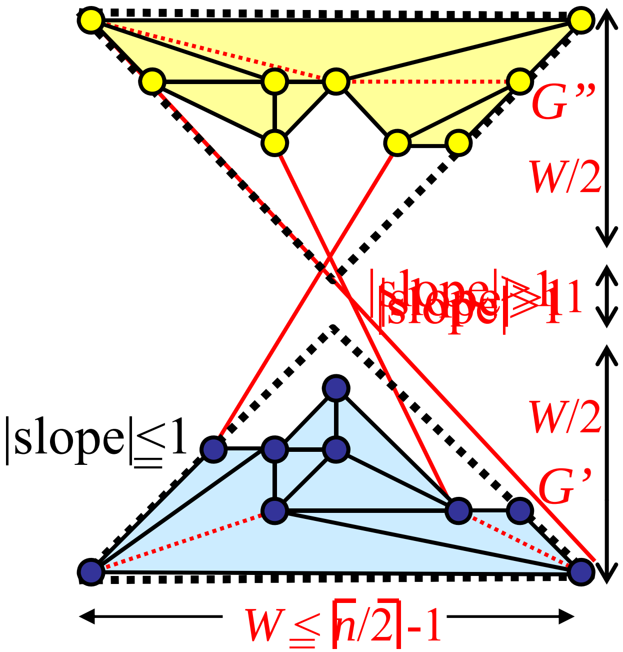
Step1: find a 4-canonical ordering



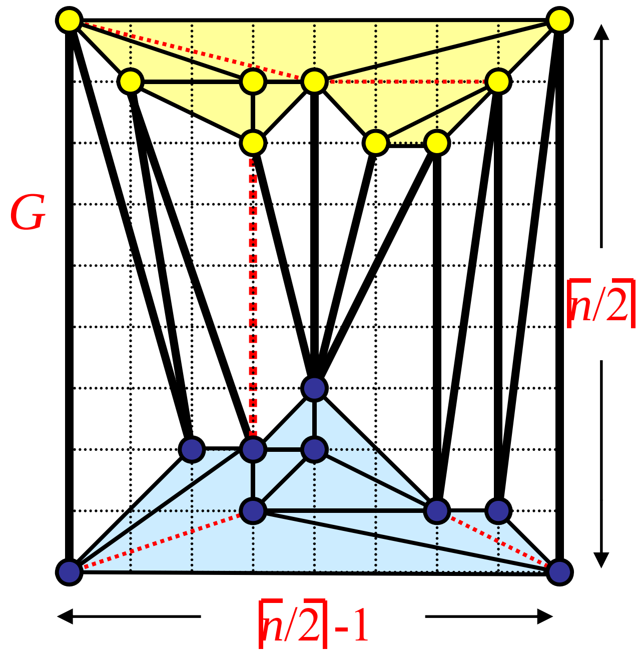
Step2: Divide G into two halves G' and G''



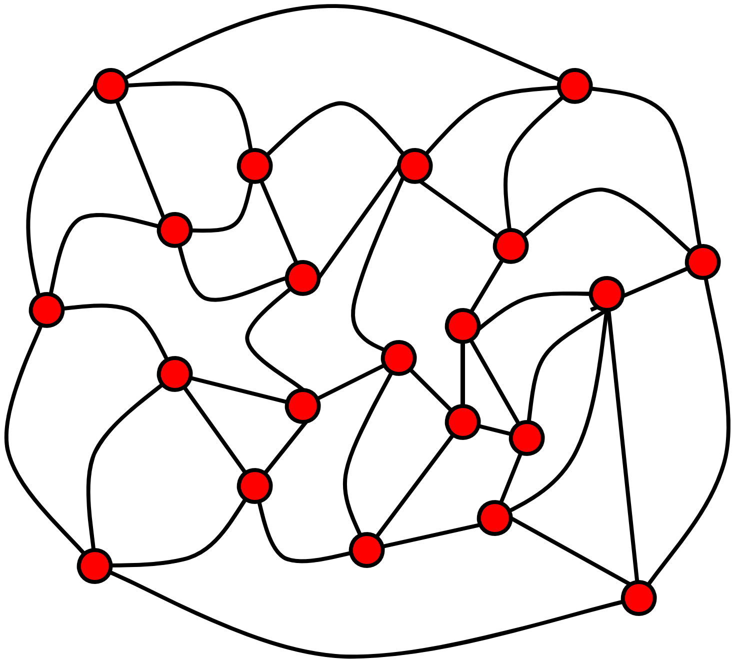
Step3 and 4 : Draw G' and G'' in isosceles right-angled triangles

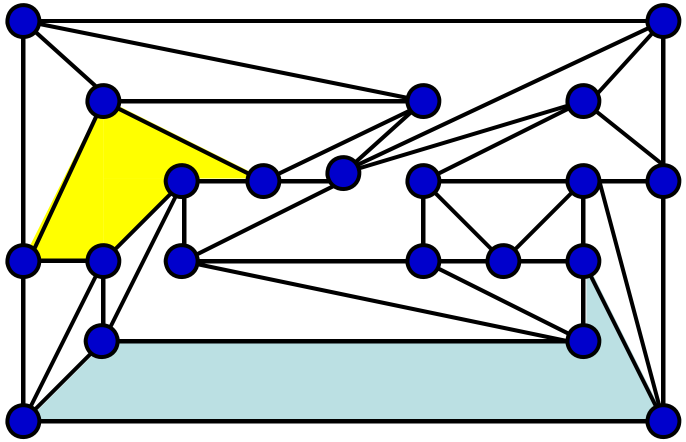


Step5: Combine the drawings of G' and G''

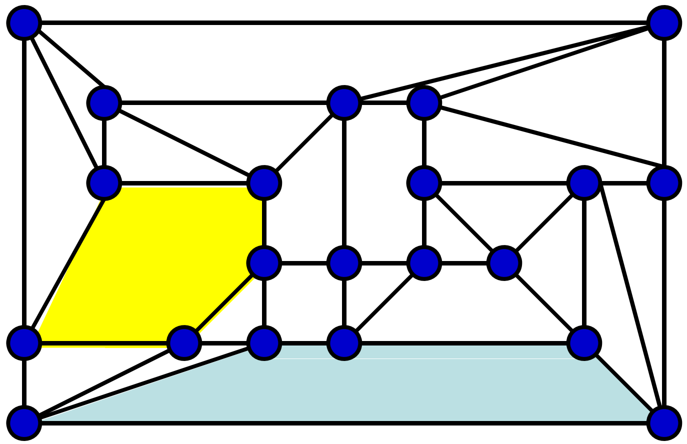


Draw a graph *G* on the plane "nicely"





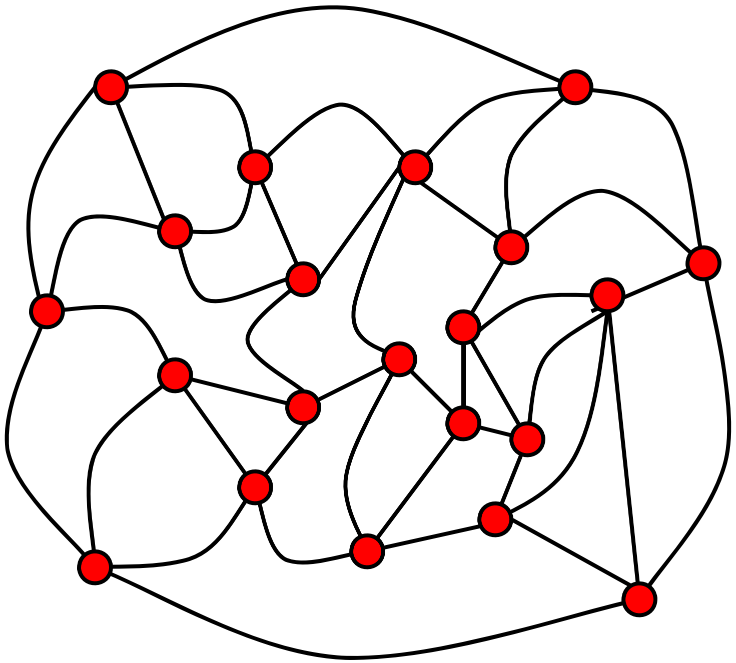
Straight line drawing

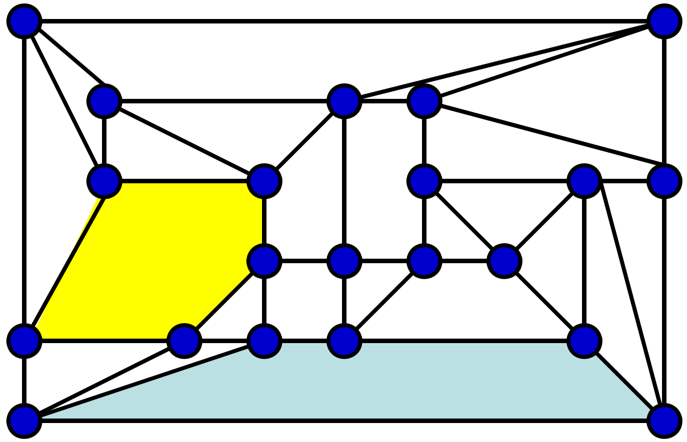


Convex drawing

A convex drawing is a straight line drawing where each face is drawn as a convex polygon.

Convex Drawing





Convex drawing

Tutte 1963

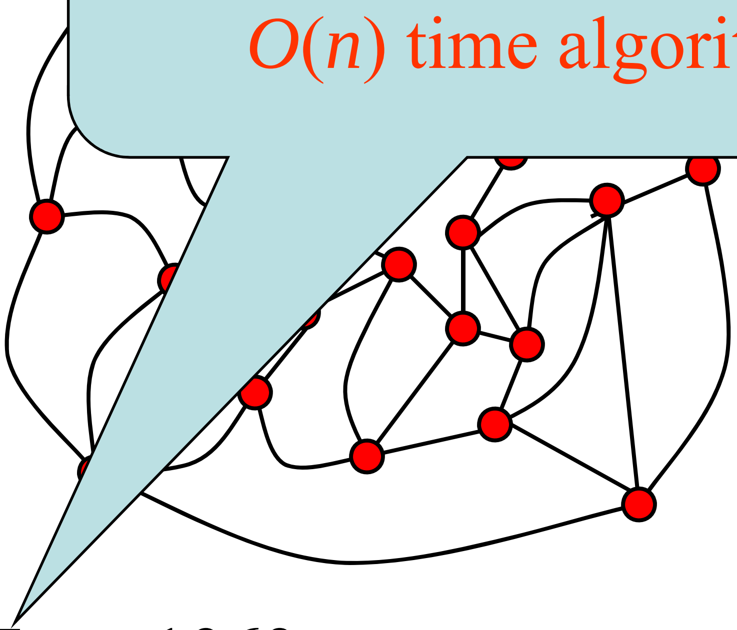
Every 3-connected planar graph has a convex drawing.

A necessary and sufficient condition for a plane graph to have a convex drawing. Thomassen ’80

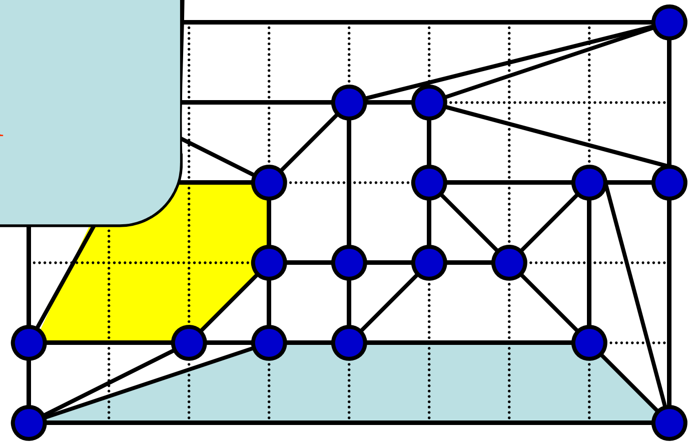
Convex Drawing

Chiba *et al.* ’84

$O(n)$ time algorithm



Tutte 1963



Convex drawing

Every 3-connected planar graph has a convex drawing

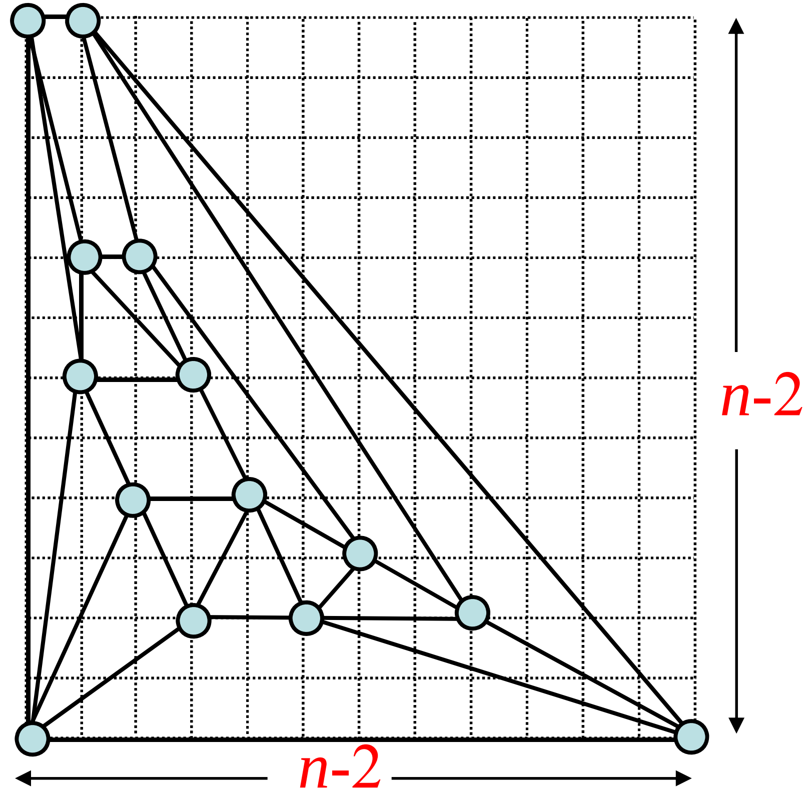
A necessary and sufficient condition for a plane graph to have a convex drawing. Thomassen ’80

Convex Grid Drawing

Chrobak and Kant '97

Input: 3-connected graph

Output: convex grid drawing



Grid Size

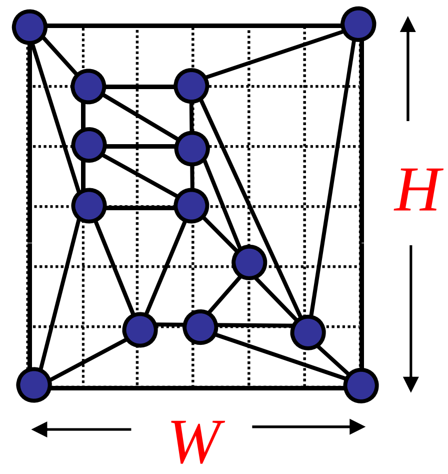
Area $W \times H \leq n^2$

Convex Grid Drawing

Miura *et al.* 2000

Input : 4-connected plane graph

Output: Convex grid drawing



Grid Size

Half-perimeter

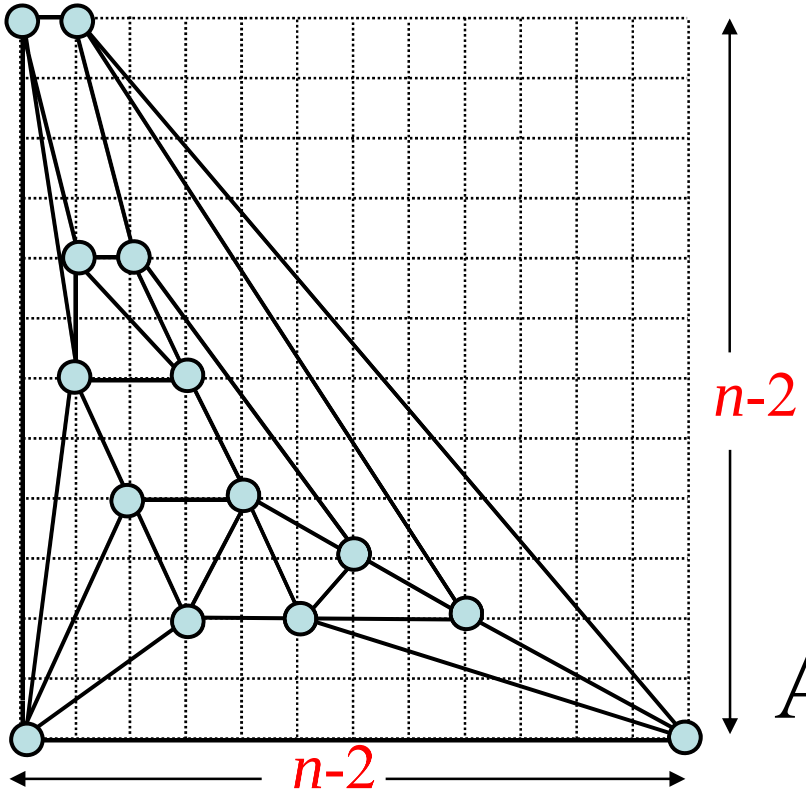
$$W + H \leq n - 1$$

Area

$$W \times H \leq \frac{n^2}{4}$$

Chrobak and Kant '97

3-connected graph

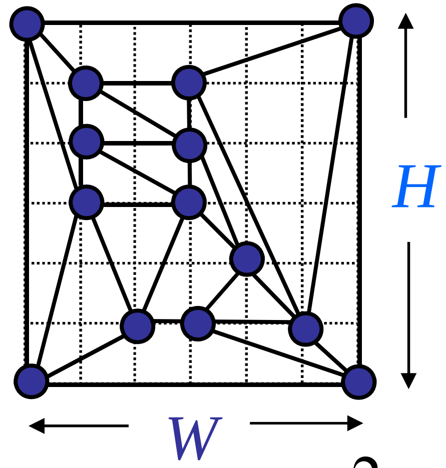


Area $\approx n^2$

Miura *et al.* 2000

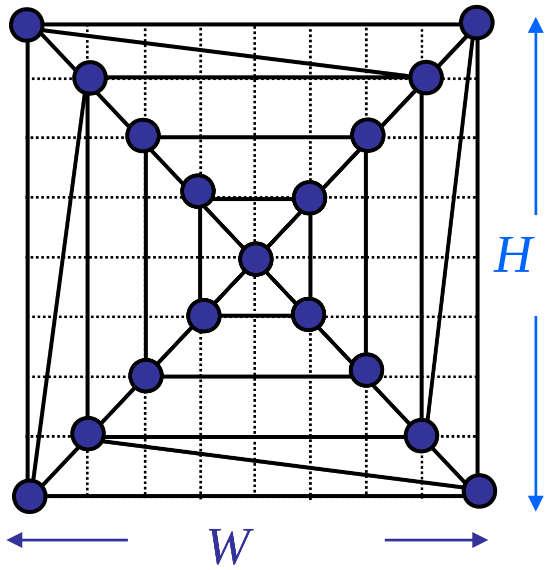
4-connected graph

Area $\leqq$ 1/4



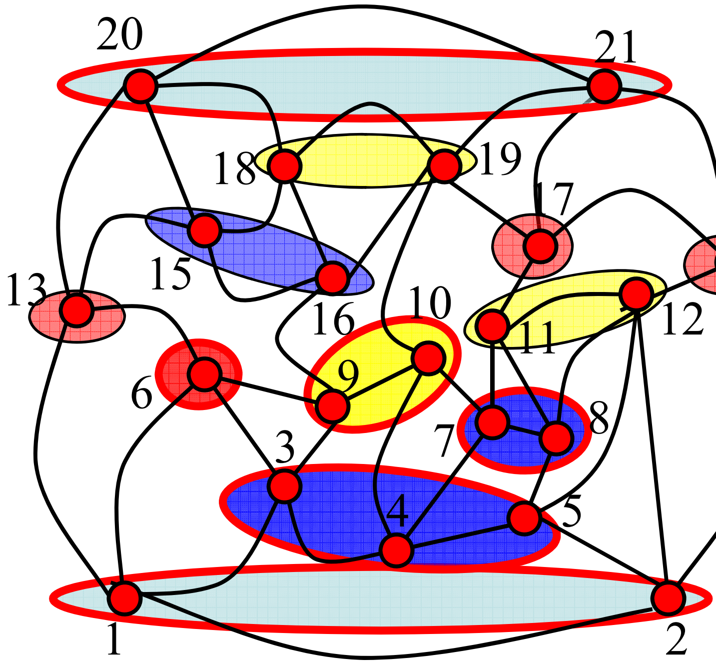
$$\text{Area} \leq \frac{n^2}{4}$$

The algorithm of Miura *et al.* is best possible

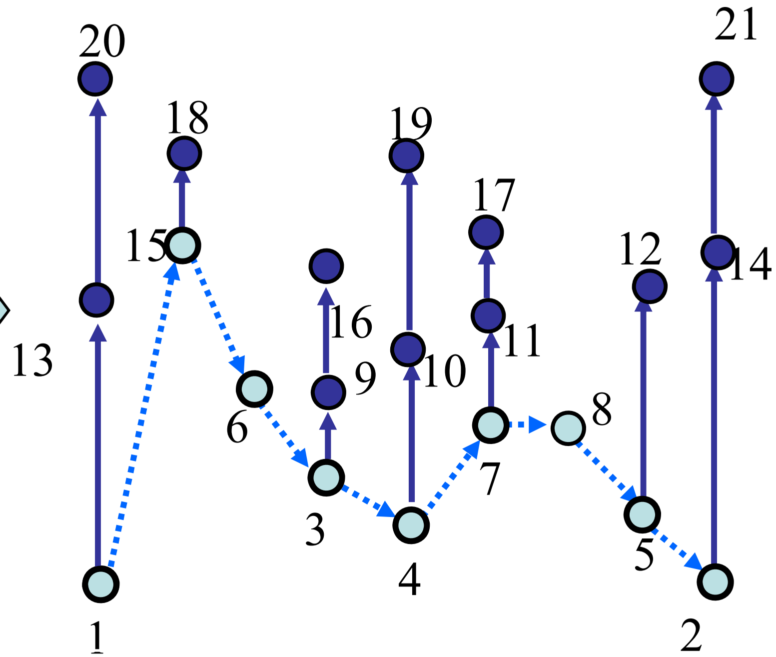


$$W \times H \geq \frac{n^2}{4}$$

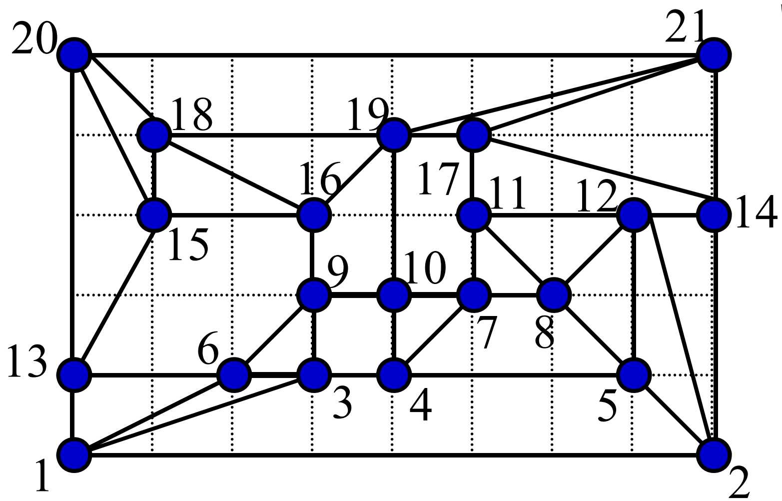
Main idea



1: 4-canonical decomposition
$O(n)$[NRN97]



2: Find paths

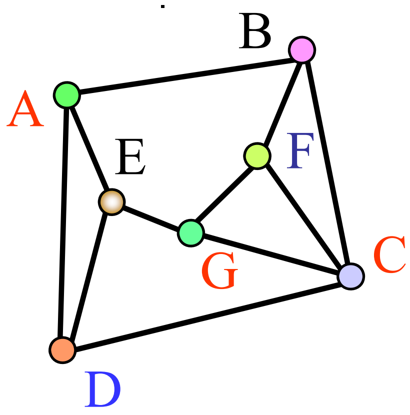


3: Decide x-coordinates

4: Decide y-coordinates

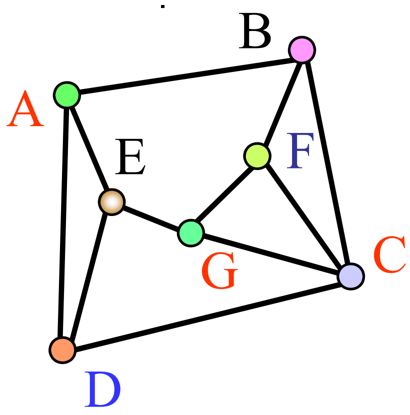
Time complexity: $O(n)$

VLSI Floorplanning

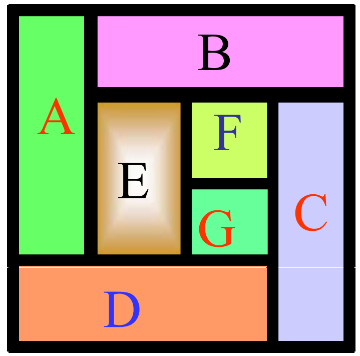


Interconnection graph

VLSI Floorplanning

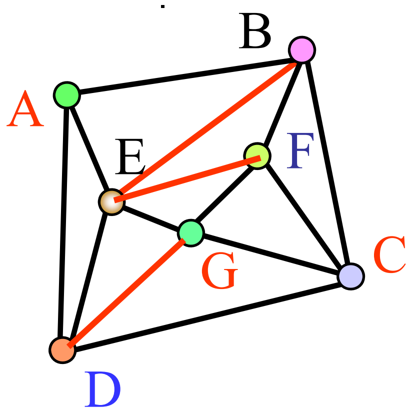


Interconnection graph

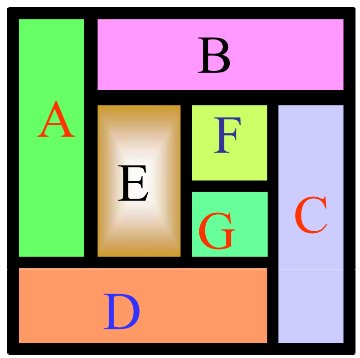


VLSI floorplan

VLSI Floorplanning

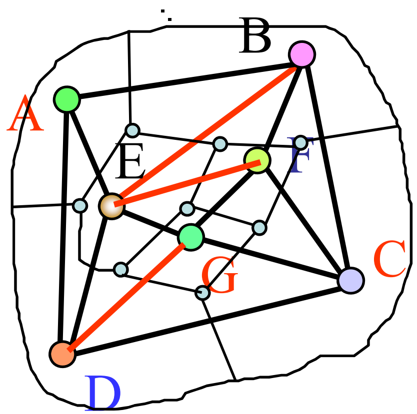


Interconnection graph

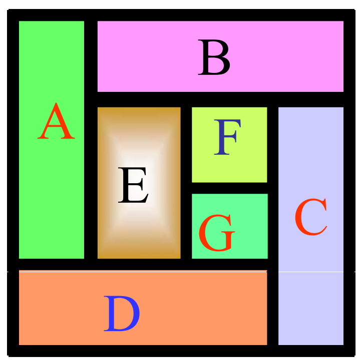


VLSI floorplan

VLSI Floorplanning

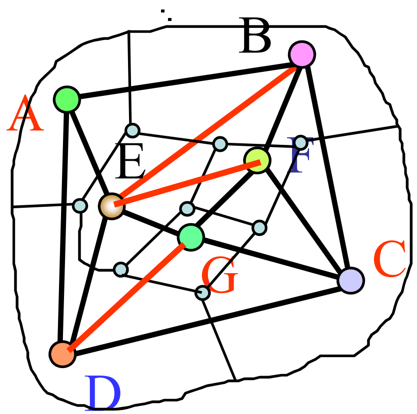


Interconnection graph

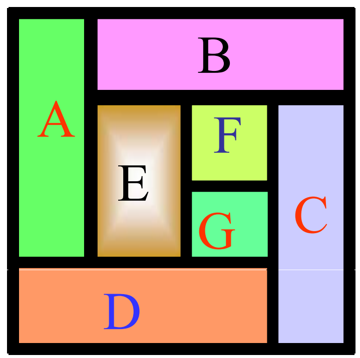


VLSI floorplan

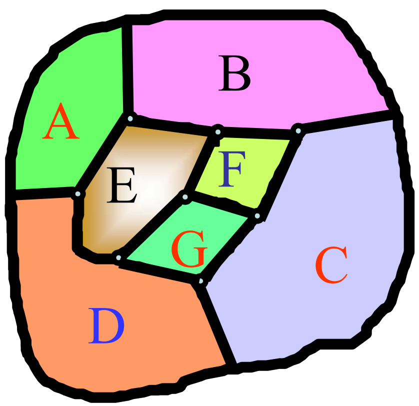
VLSI Floorplanning



Interconnection graph

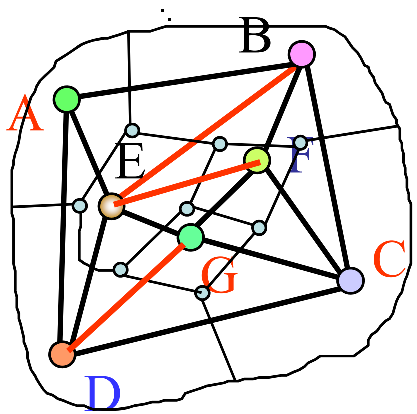


VLSI floorplan

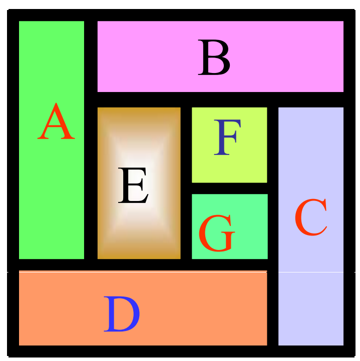


Dual-like graph

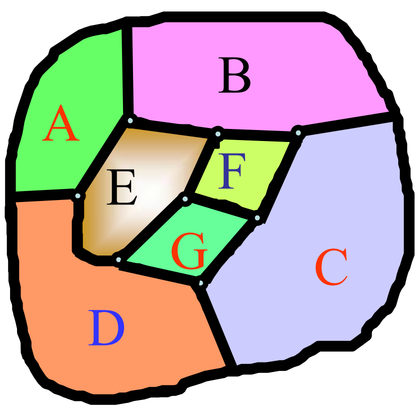
VLSI Floorplanning



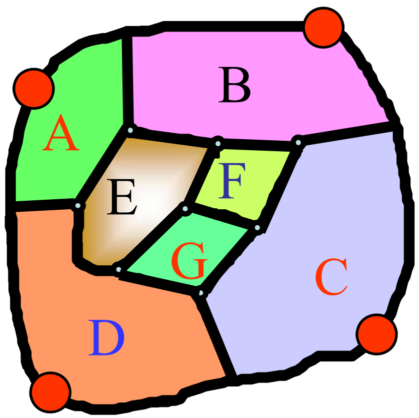
Interconnection graph



VLSI floorplan

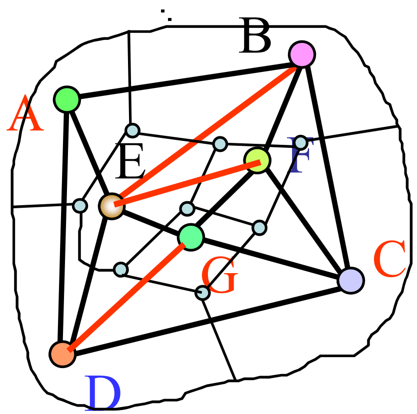


Dual-like graph

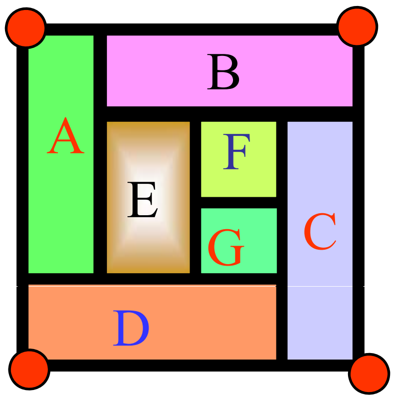


Add four corners

VLSI Floorplanning

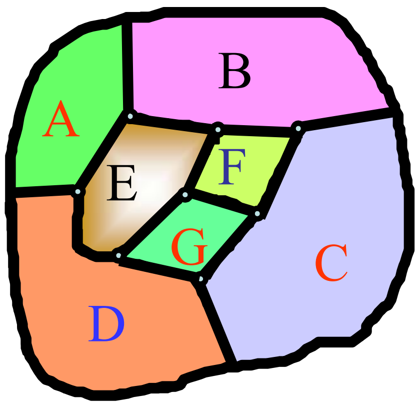


Interconnection graph

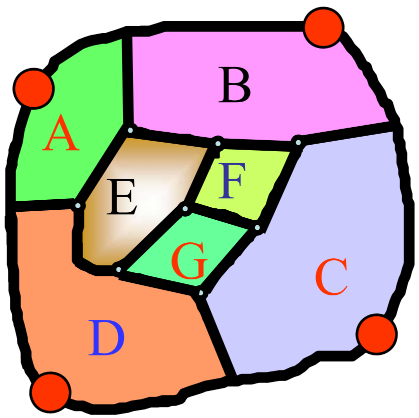


Rectangular drawing

VLSI floorplan

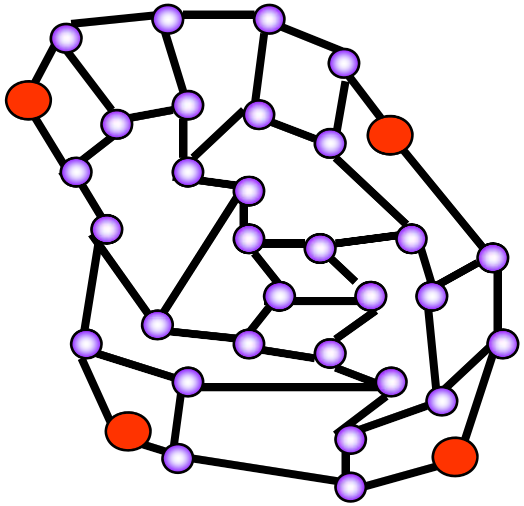


Dual-like graph



Add four corners

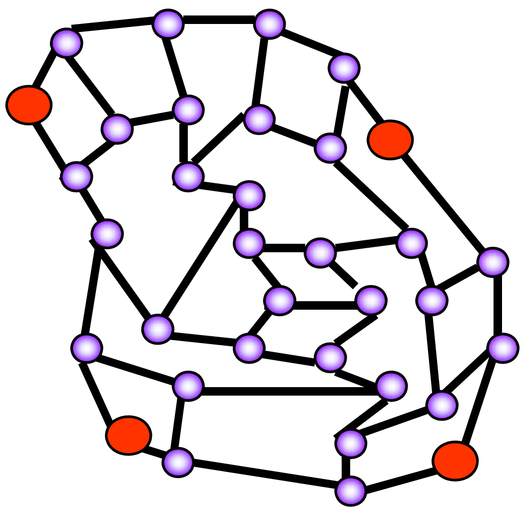
Rectangular Drawings



Plane graph G of $\Delta \leq 3$

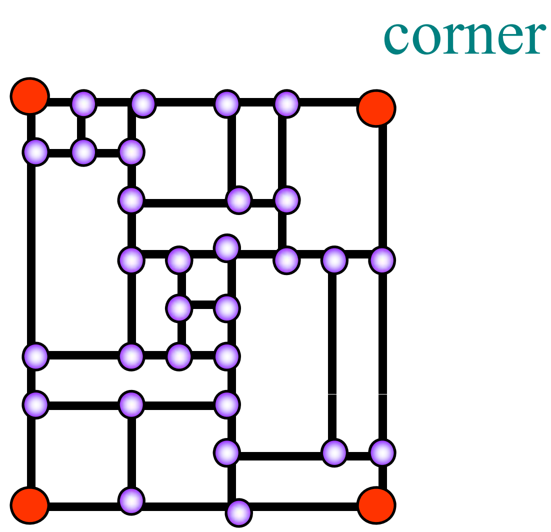
Input

Rectangular Drawings



Plane graph G of $\Delta \leq 3$

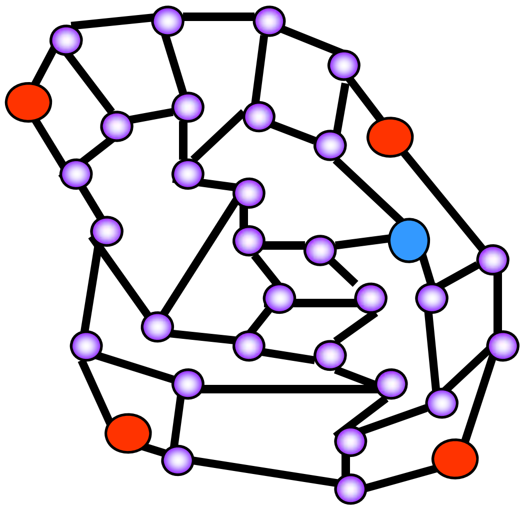
Input



Rectangular drawing of G

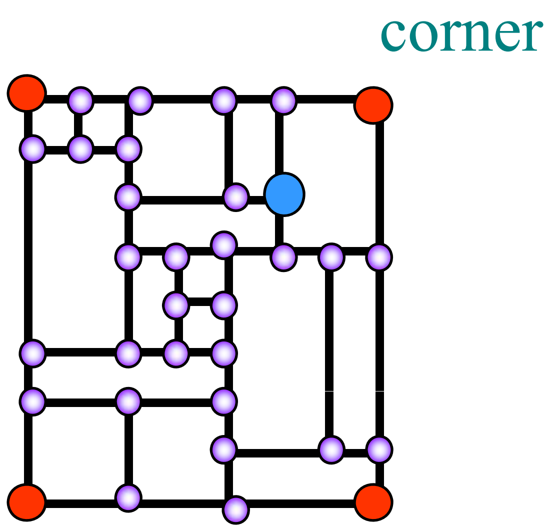
Output

Rectangular Drawings



Plane graph G of $\Delta \leq 3$

Input

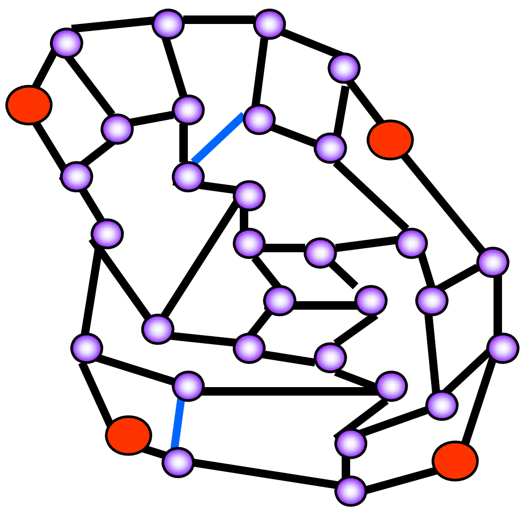


Rectangular drawing of G

Output

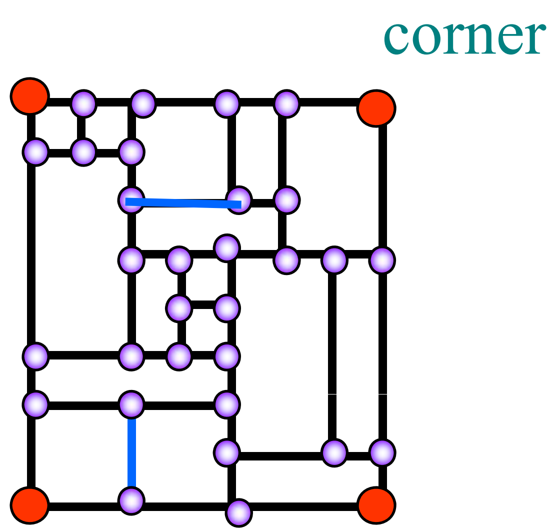
- Each vertex is drawn as a point.

Rectangular Drawings



Plane graph G of $\Delta \leq 3$

Input

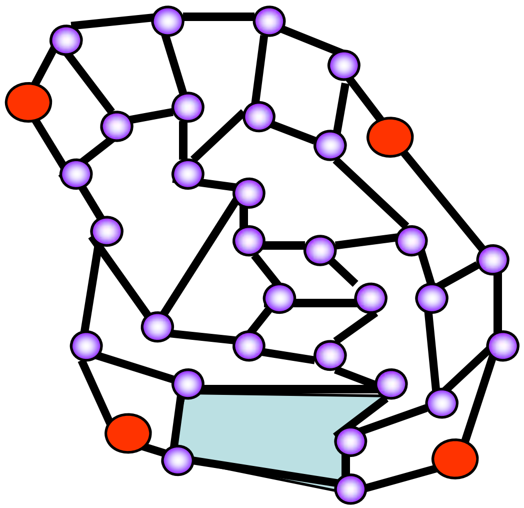


Rectangular drawing of G

Output

- Each vertex is drawn as a point.
- Each edge is drawn as a horizontal or a vertical line segment.

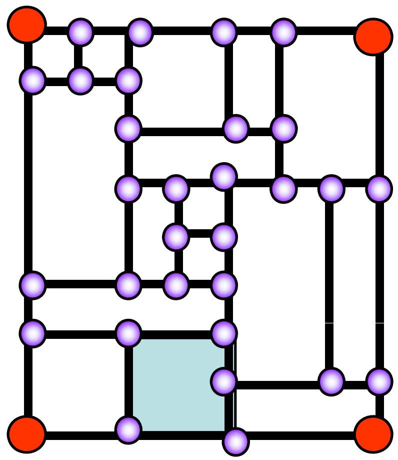
Rectangular Drawings



Plane graph G of $\Delta \leq 3$

Input

corner

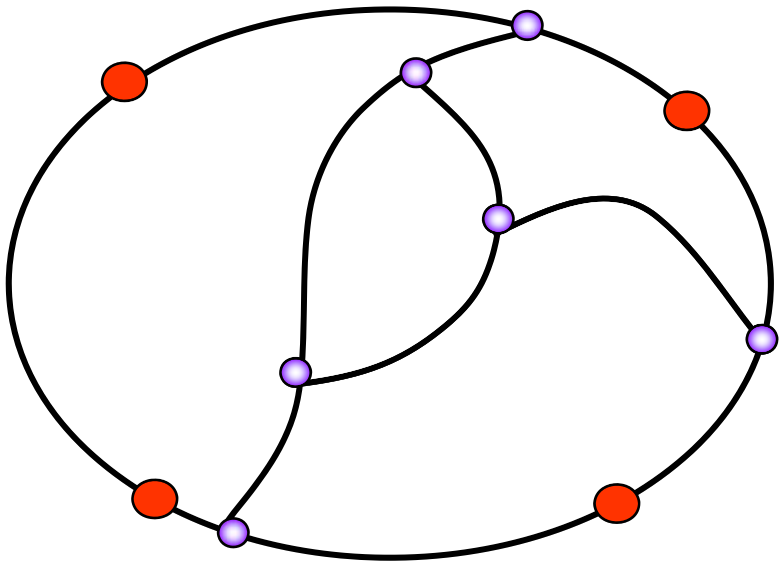


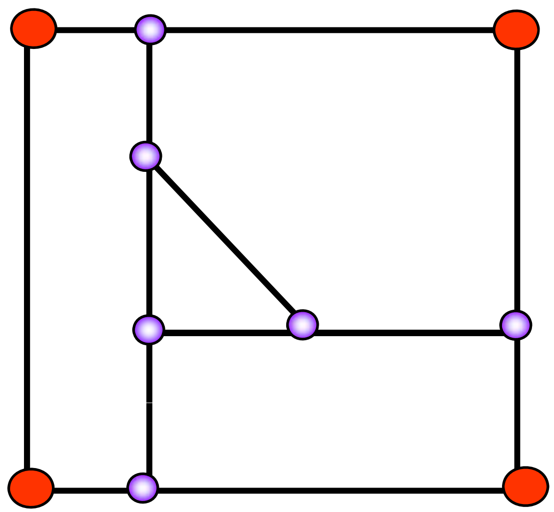
Rectangular drawing of G

Output

- Each vertex is drawn as a point.
- Each edge is drawn as a horizontal or a vertical line segment.
- Each face is drawn as a rectangle.

Not every plane graph has a rectangular drawing.





Thomassen '84,

Necessary and sufficient condition

Rahman, Nakano and Nishizeki '98

Linear-time algorithms

Rahman, Nakano and Nishizeki '02

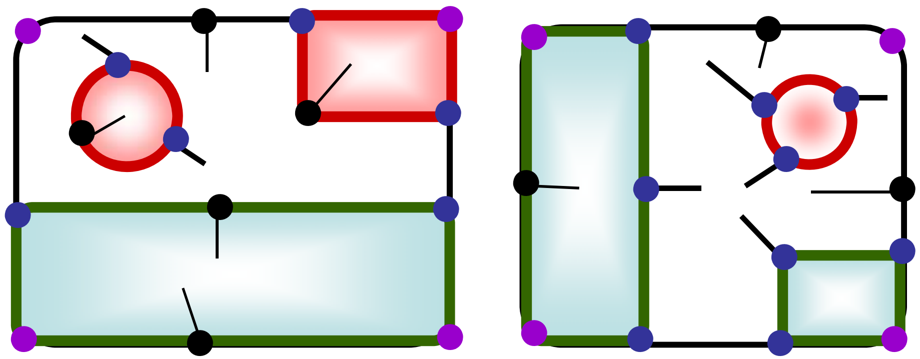
Linear algorithm for the case where corners are not designated in advance

A Necessary and Sufficient Condition by Thomassen ’84

G
- plane graph
- four vertices of degree 2 are designated as corners

G has a rectangular drawing if and only if

- every 2-legged cycle in *G* contains at least two designated vertices; and
- every 3-legged cycle in *G* contains at least one designated vertex.



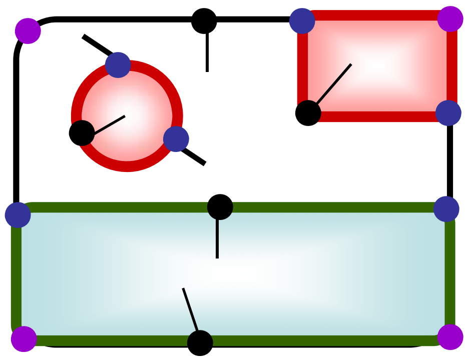
2-legged cycles 3-legged cycles

A Necessary and Sufficient Condition by Thomassen '84

G
- plane graph
- four vertices of degree 2 are designated as corners

G has a rectangular drawing if and only if

- every 2-legged cycle in *G* contains at least two designated vertices; and
- every 3-legged cycle in *G* contains at least one designated vertex.



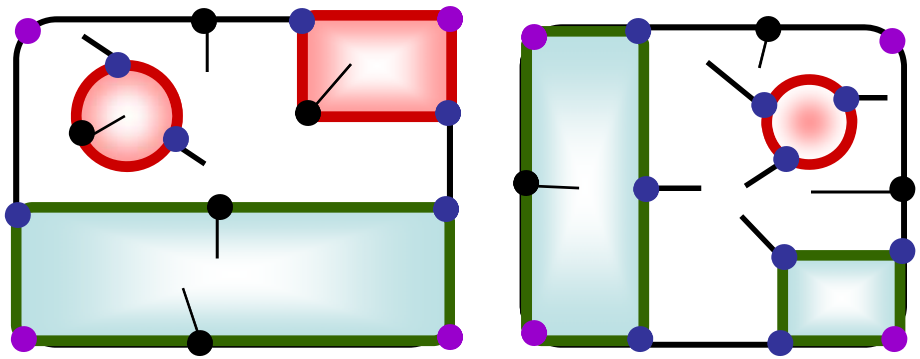
2-legged cycles

A Necessary and Sufficient Condition by Thomassen '84

G
- plane graph
- four vertices of degree 2 are designated as corners

G has a rectangular drawing if and only if

- every 2-legged cycle in *G* contains at least two designated vertices; and
- every 3-legged cycle in *G* contains at least one designated vertex.



2-legged cycles 3-legged cycles

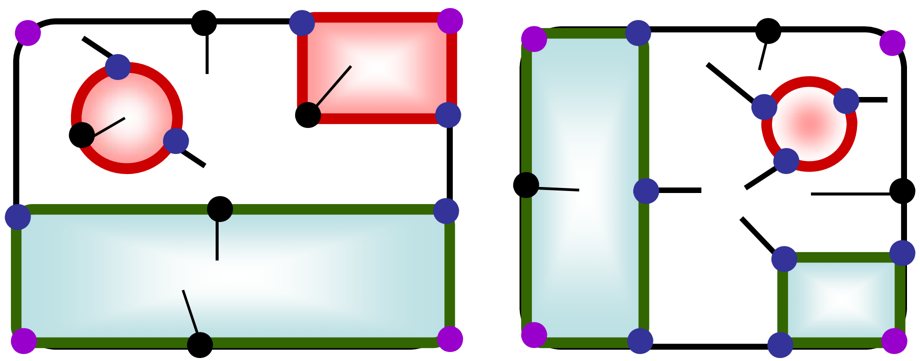
A Necessary and Sufficient Condition by Thomassen '84

G
- plane graph
- four vertices of degree 2 are designated as corners

G has a rectangular drawing if and only if

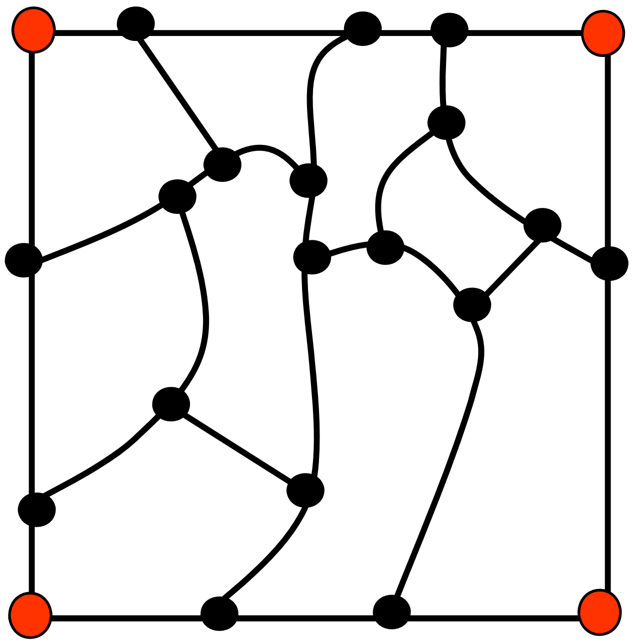
- every 2-legged cycle in *G* contains at least two designated vertices; and
- every 3-legged cycle in *G* contains at least one designated vertex.

Bad cycles

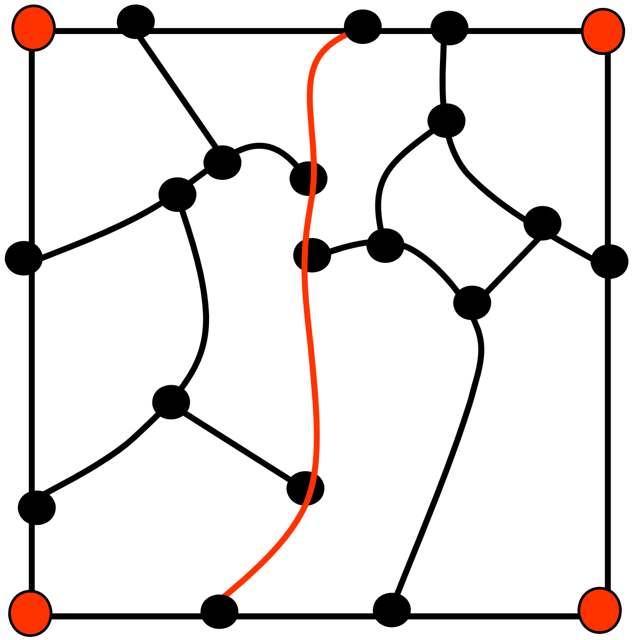


2-legged cycles   3-legged cycles

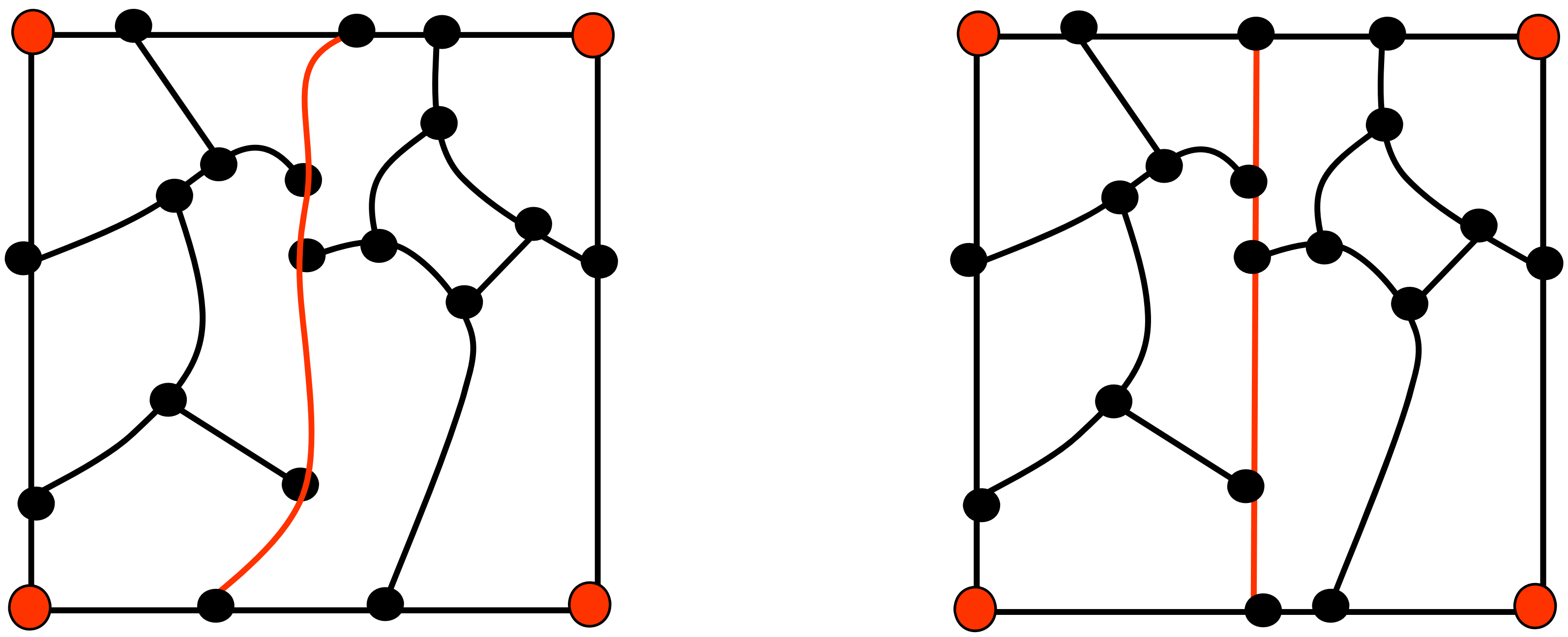
Outline of the Algorithm of Rahman *et al.*



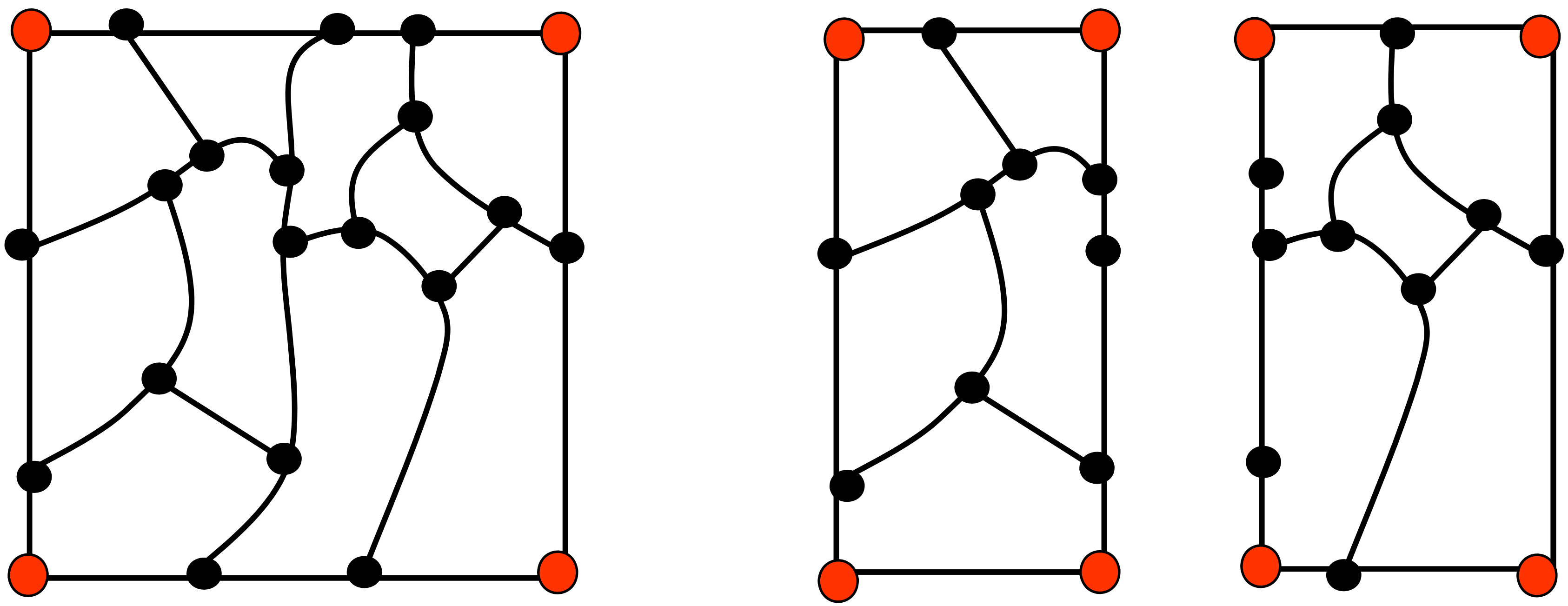
Outline of the Algorithm of Rahman *et al.*



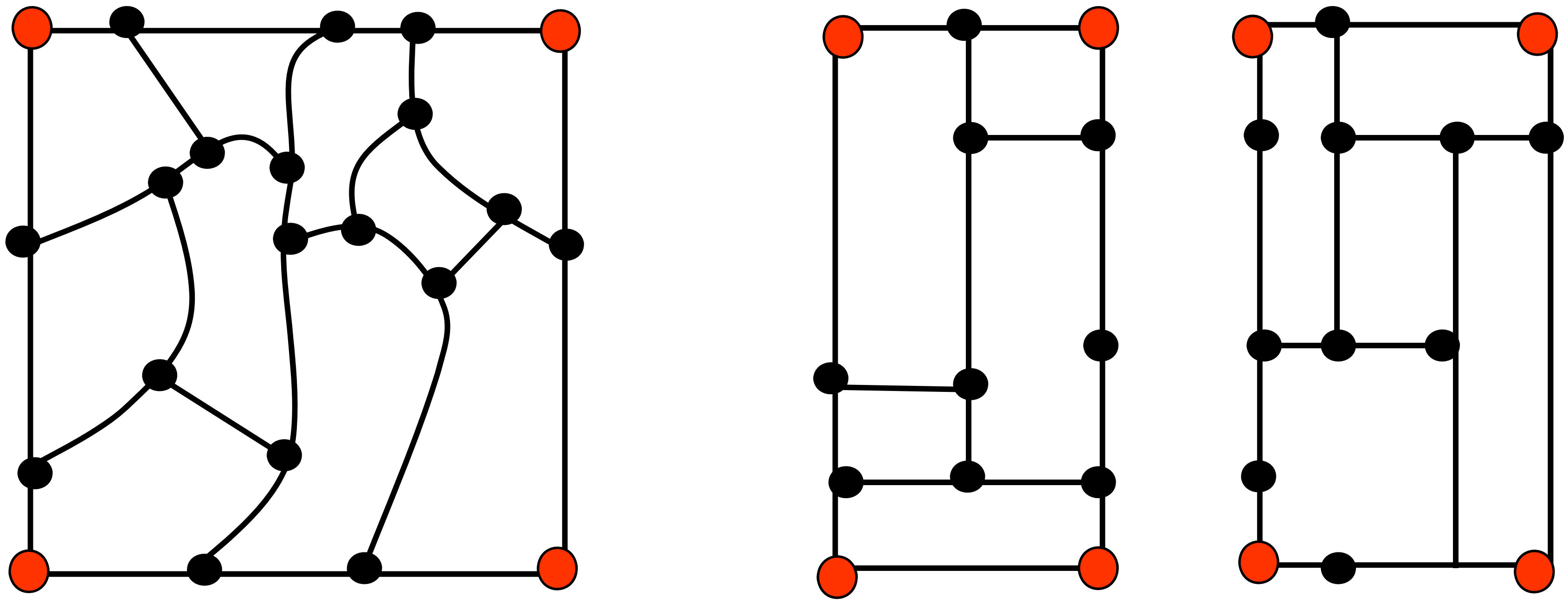
Outline of the Algorithm of Rahman *et al.*



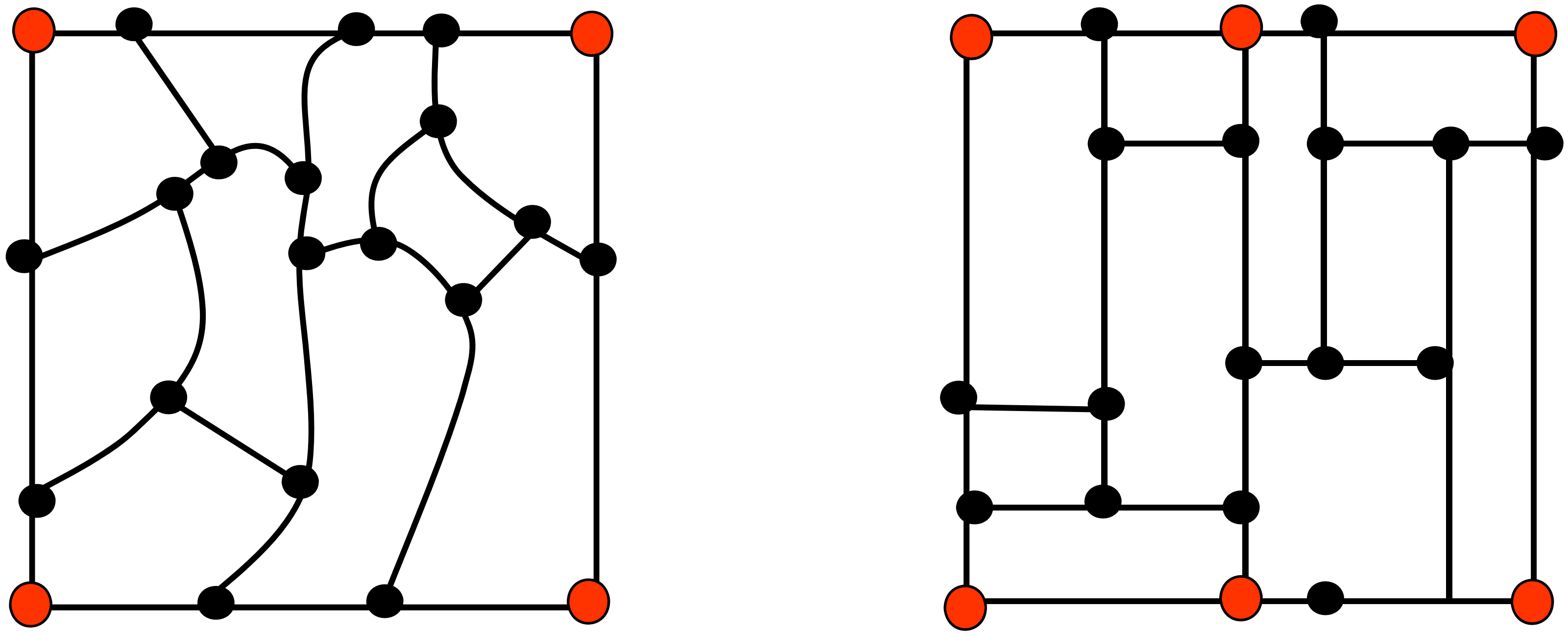
Outline of the Algorithm of Rahman *et al.*

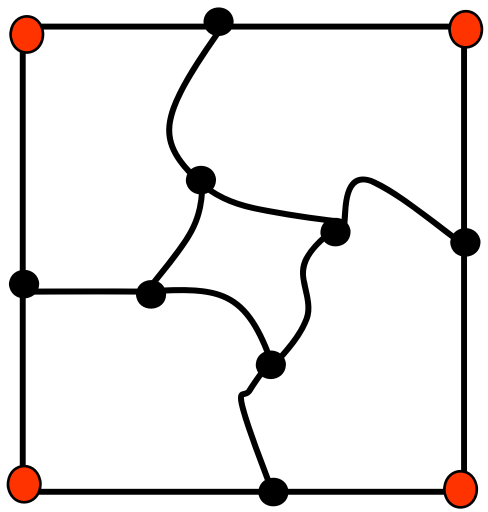


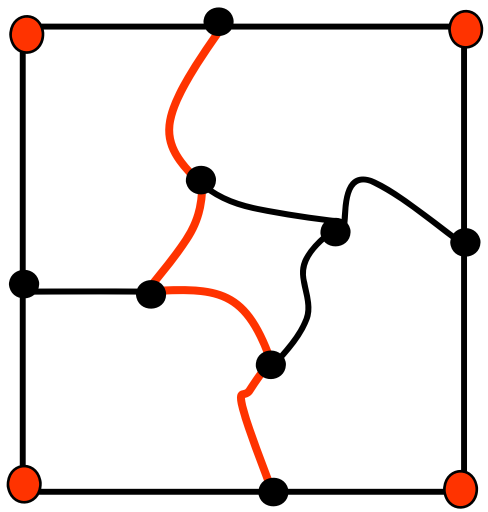
Outline of the Algorithm of Rahman *et al.*

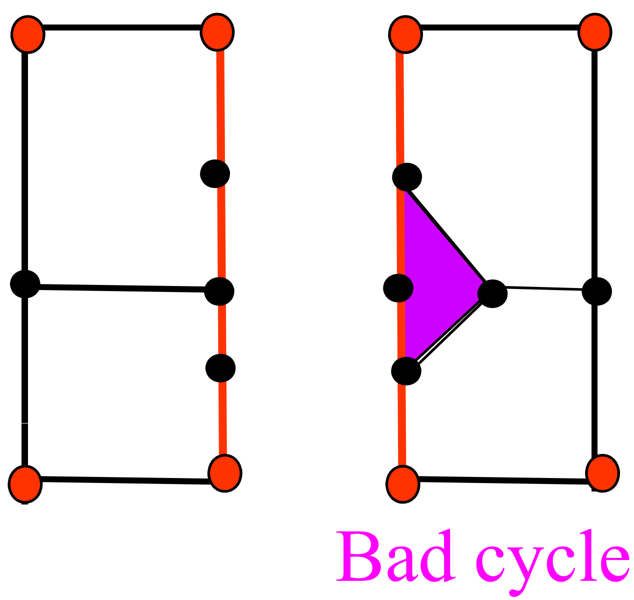


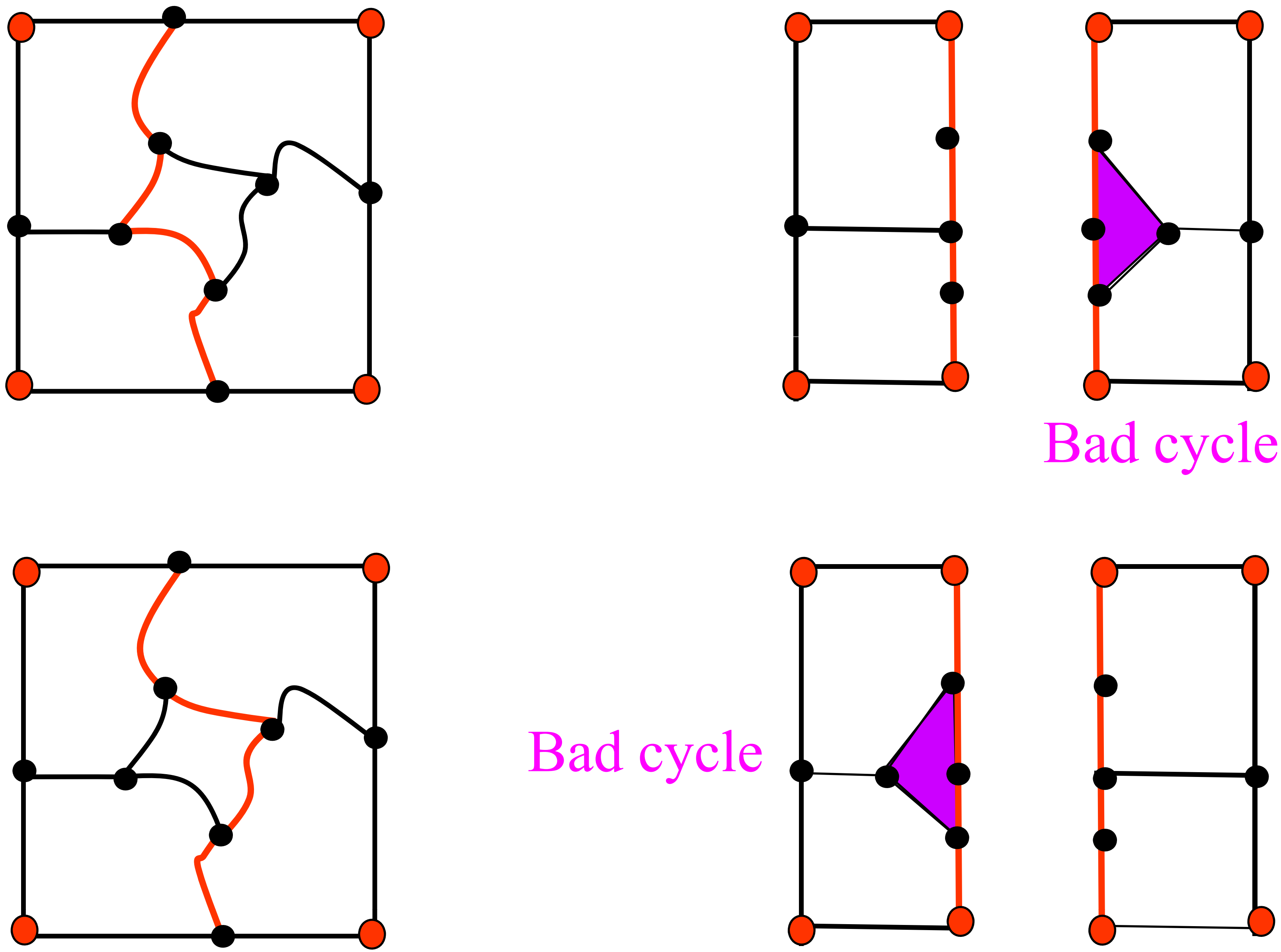
Outline of the Algorithm of Rahman *et al.*

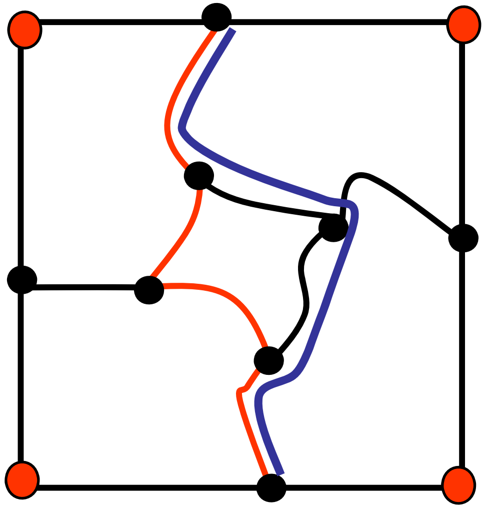




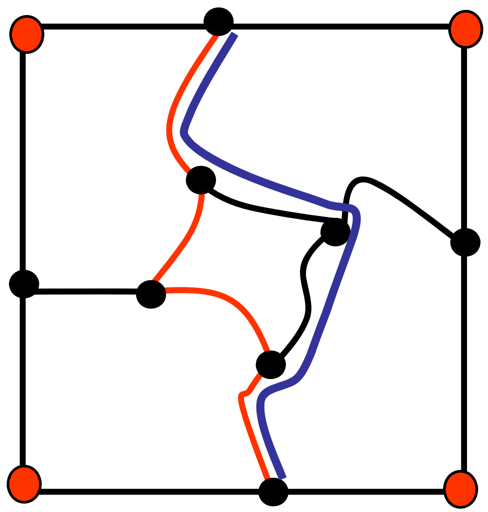



Bad cycle

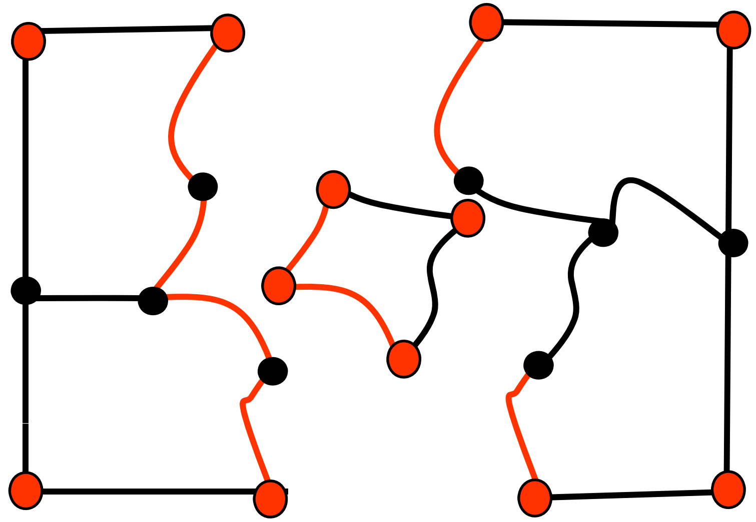

Bad cycle
Bad cycle



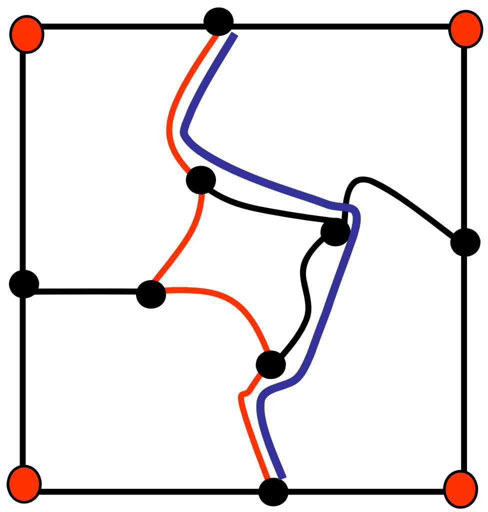
Partition-pair



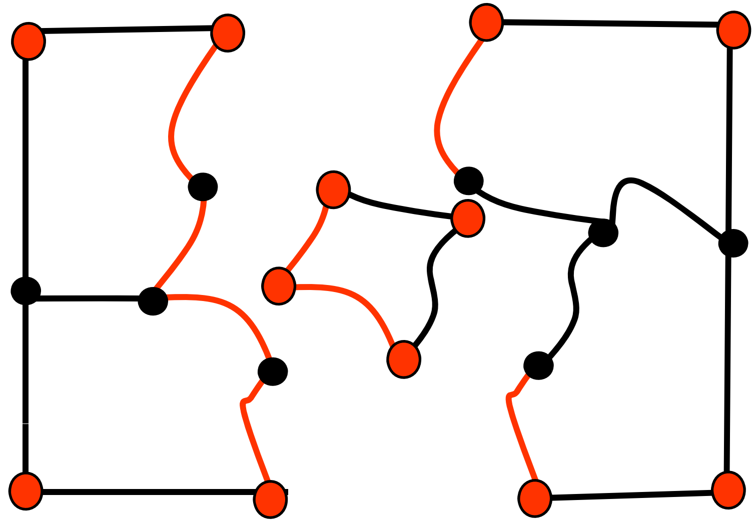
Partition-pair

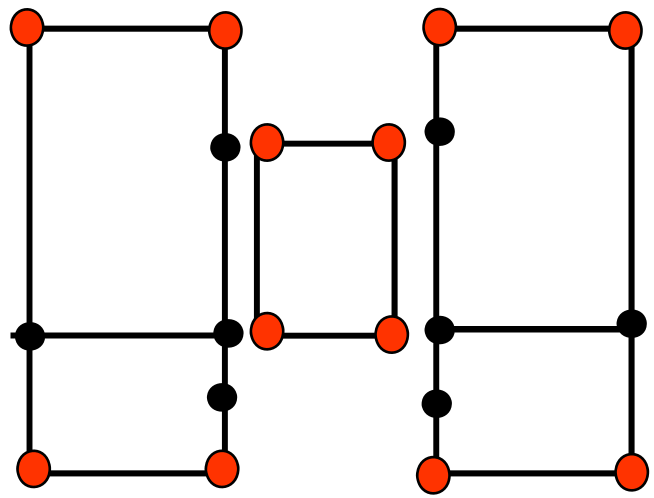


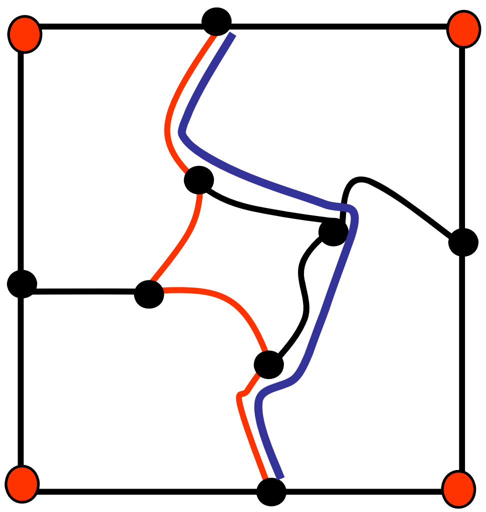
Splitting



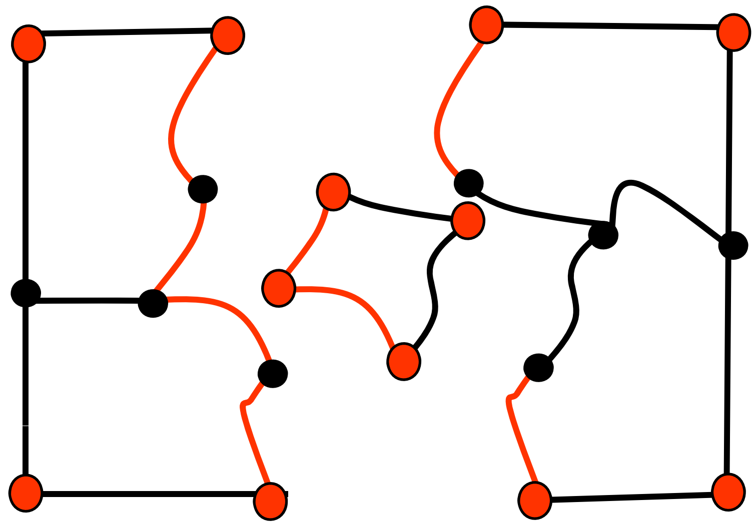
Partition-pair

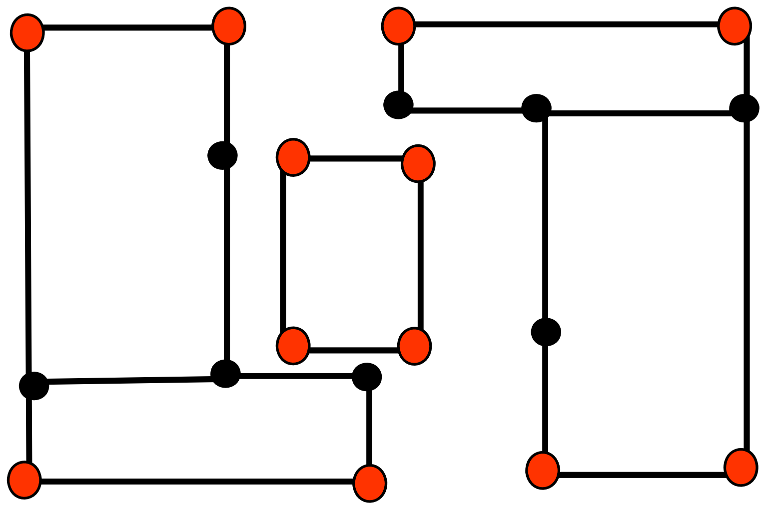


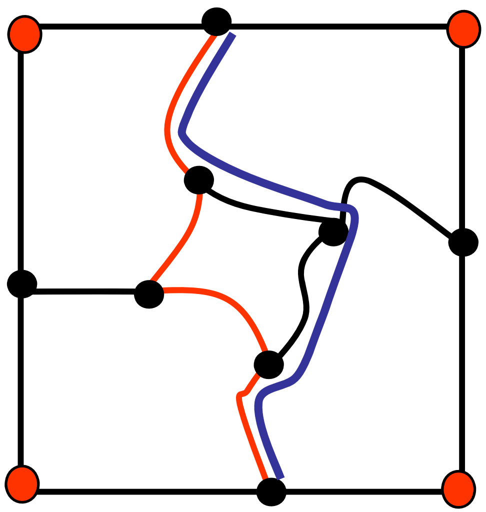


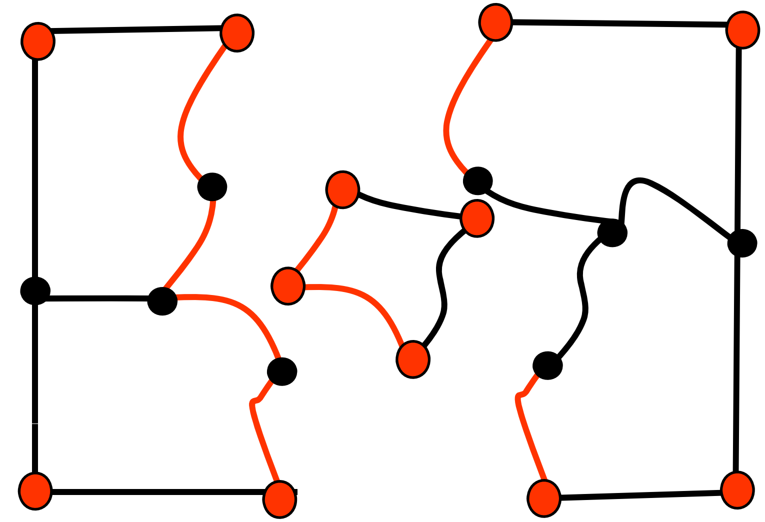


Partition-pair

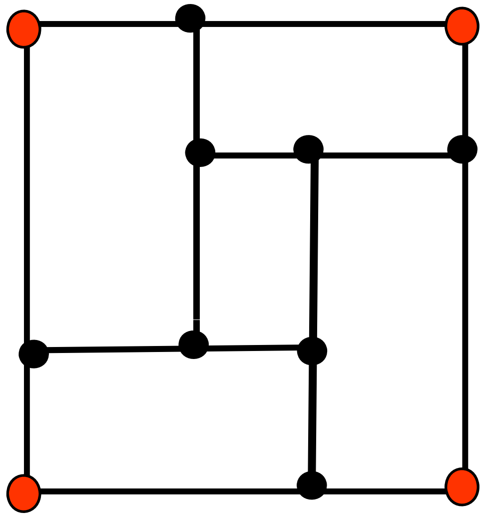


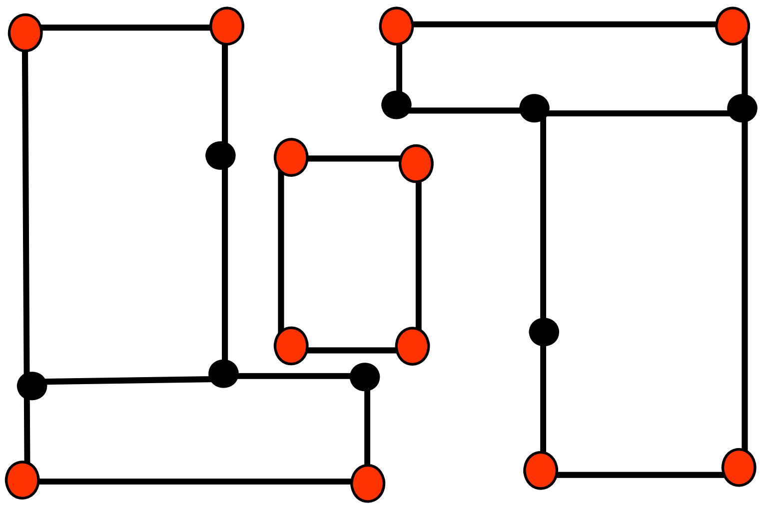




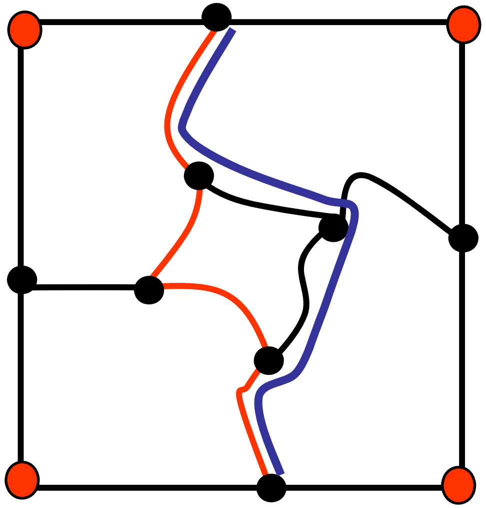


Partition-pair

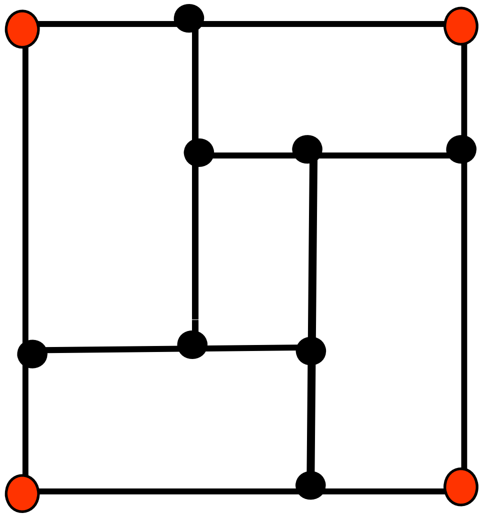




Rectangular drawing



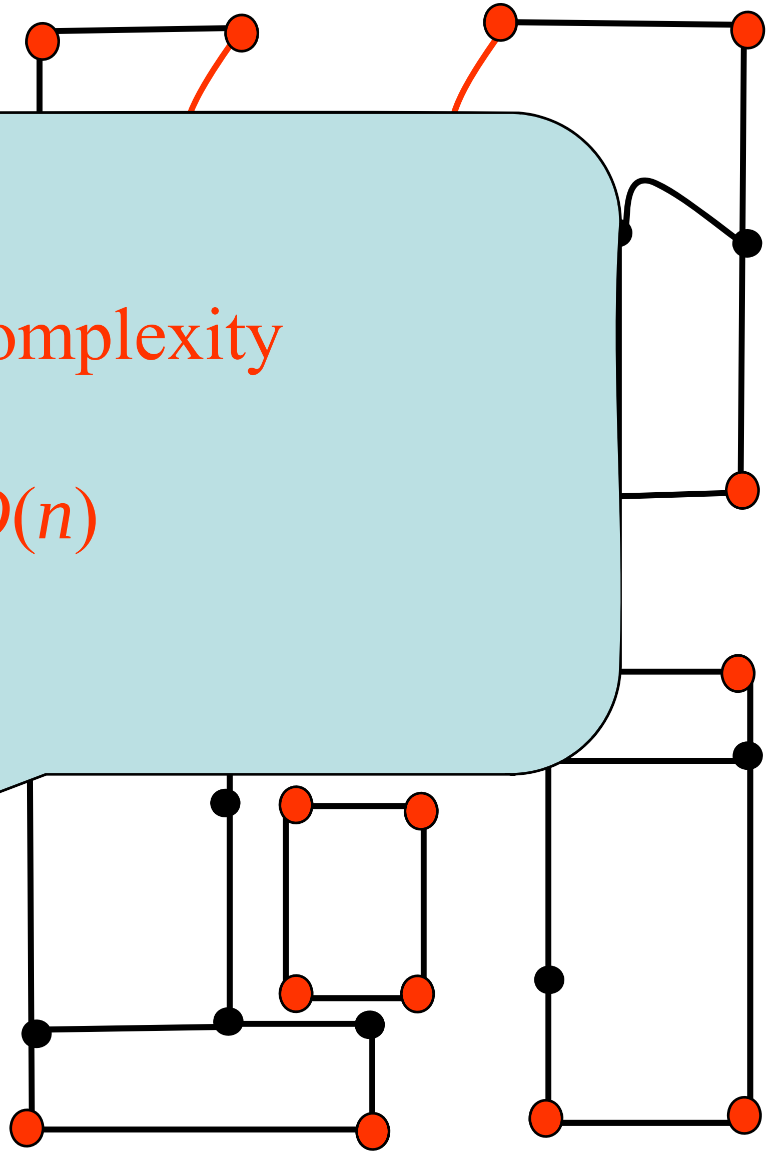
Partition-pair



Rectangular drawing

Time complexity

$O(n)$

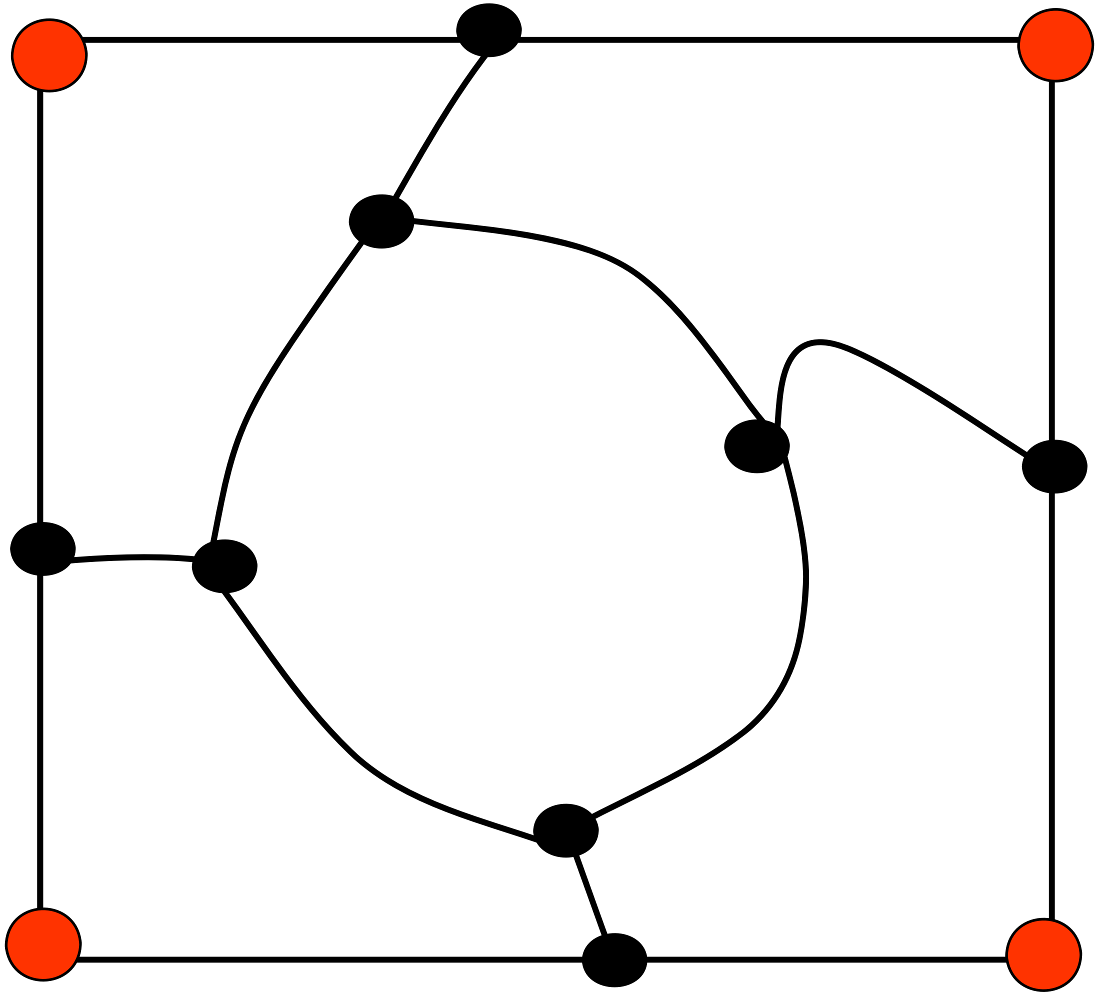


Miura, Haga, N. ’03, Working paper

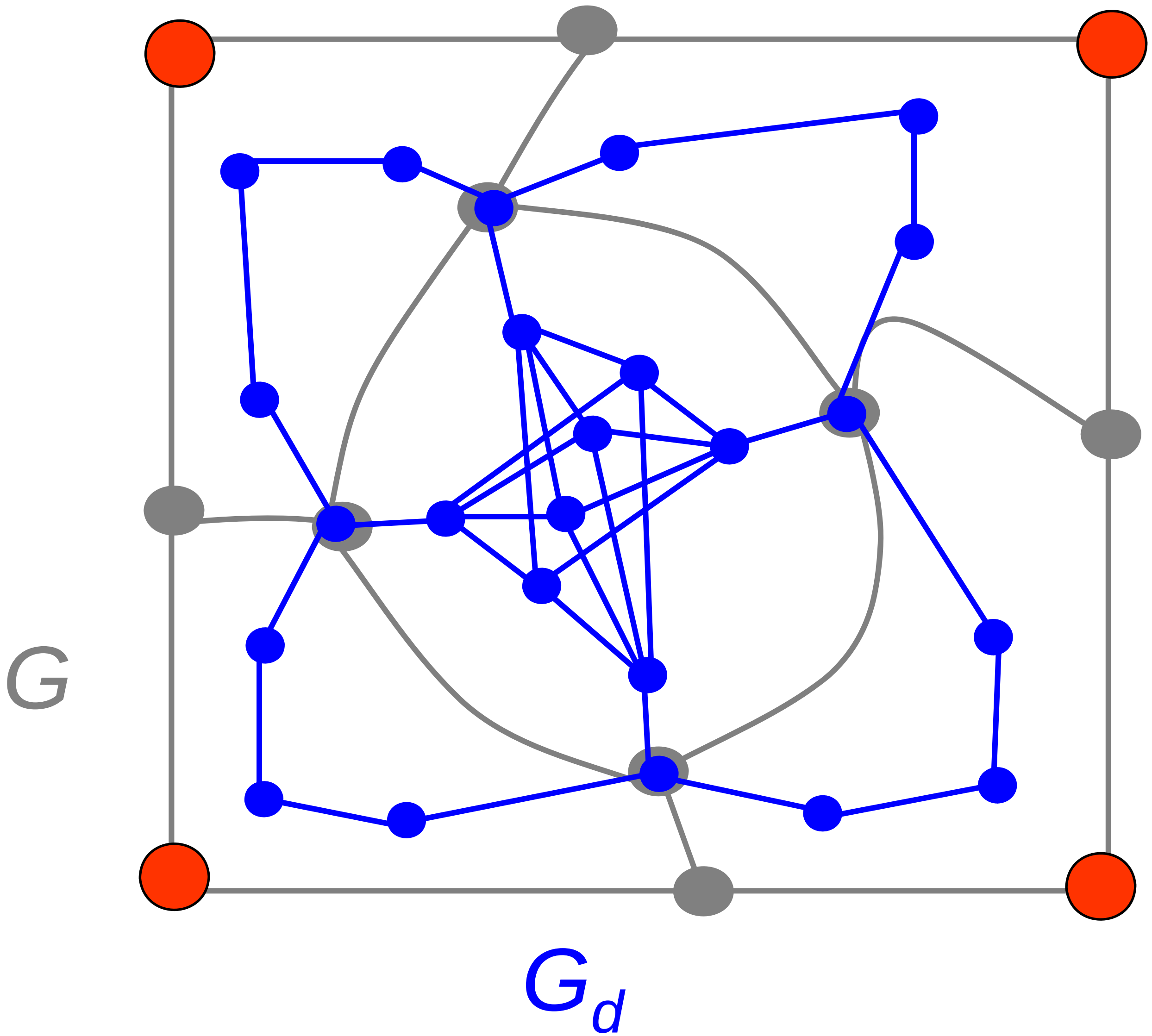
Rectangular drawing of plane graph G with $\Delta \leq 4$

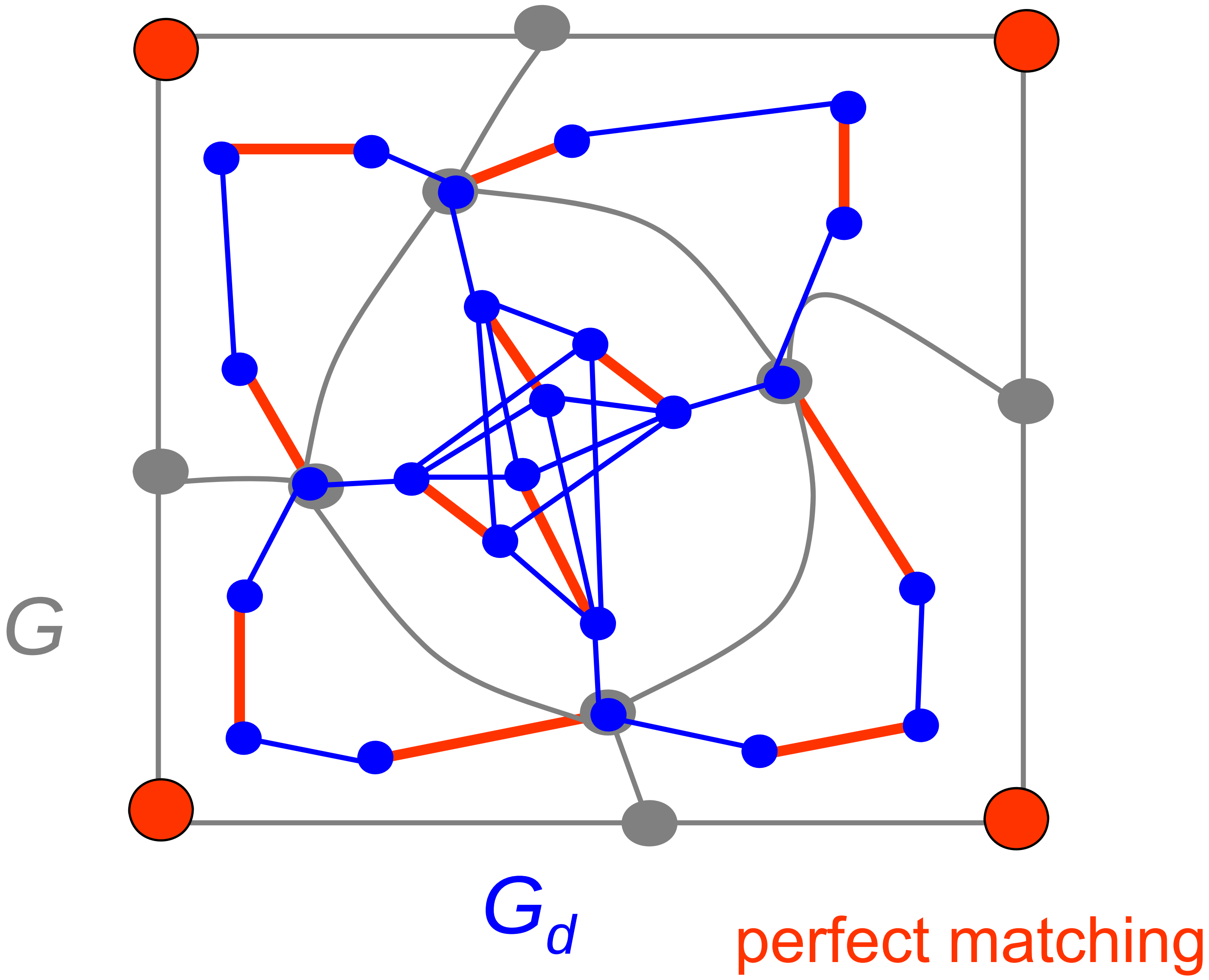
$\Updownarrow$

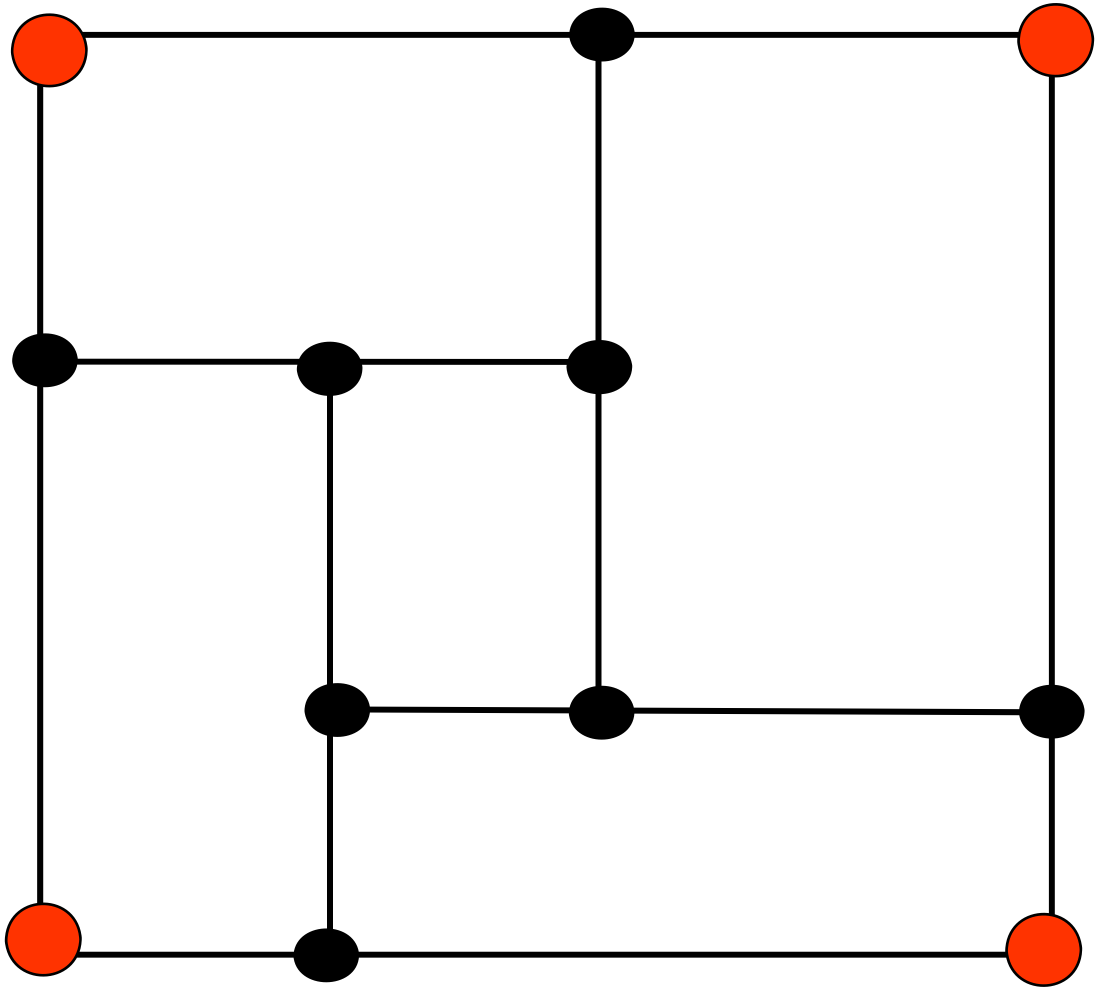
$\exists$ perfect matching in G_d



G


G
G_d

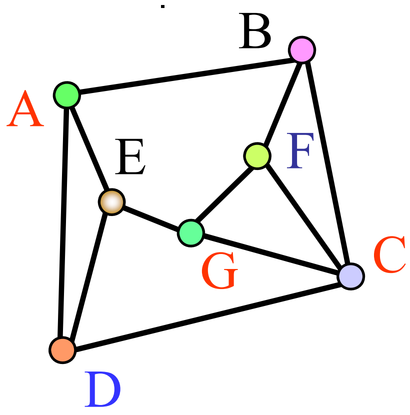

G
G_d
perfect matching



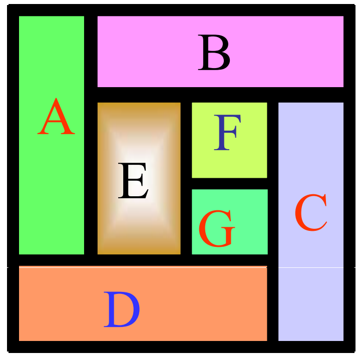
G

$O(n^{1.5})$

VLSI Floorplanning



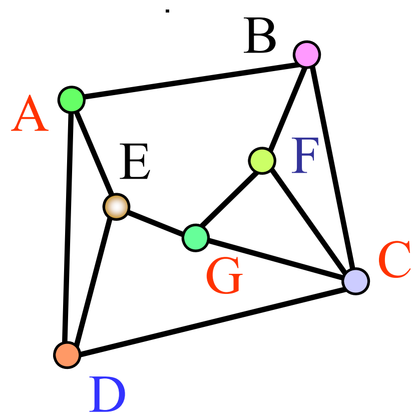
Interconnection graph



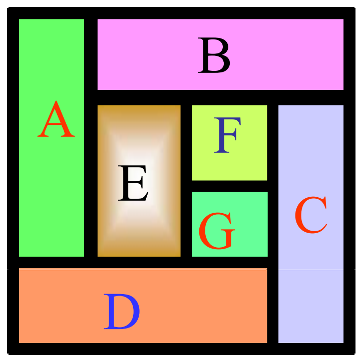
Rectangular drawing

VLSI floorplan

VLSI Floorplanning



Interconnection graph

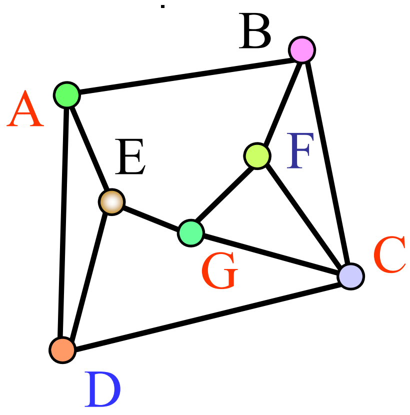


Rectangular drawing

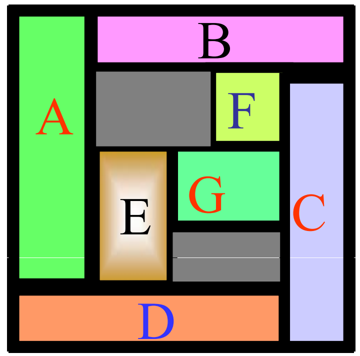
VLSI floorplan

Unwanted adjacency

Not desirable for MCM floorplanning and for some architectural floorplanning.



Interconnection graph

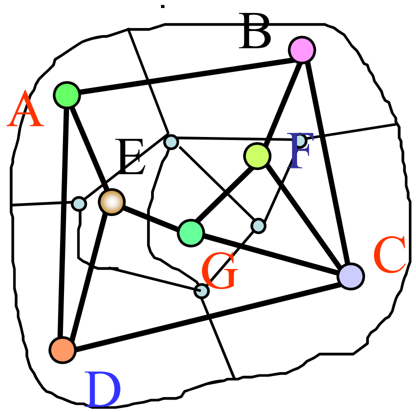


MCM Floorplanning

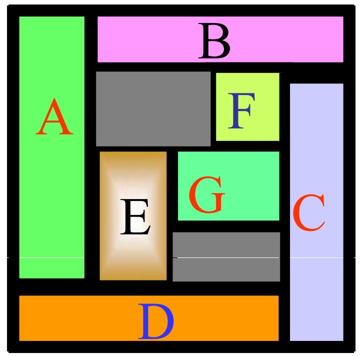
Sherwani

Architectural Floorplanning

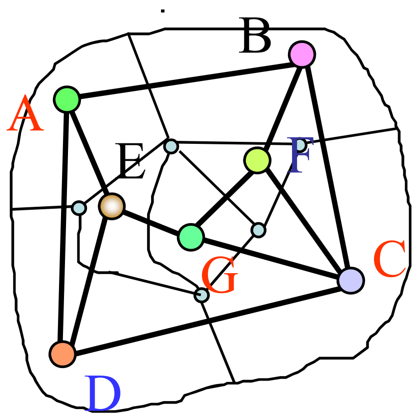
Munemoto, Katoh, Imamura



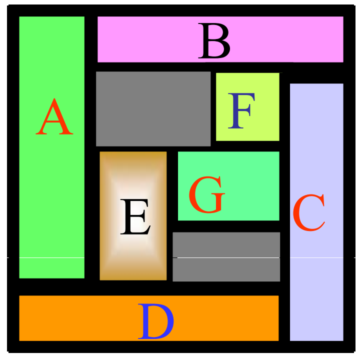
Interconnection graph



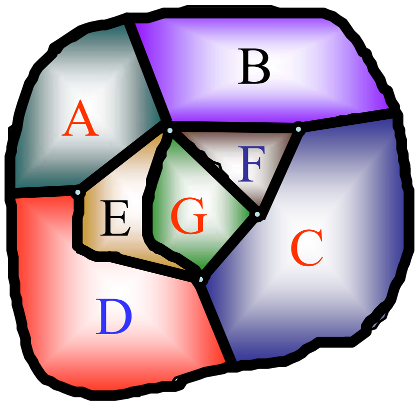
MCM Floorplanning
Architectural Floorplanning



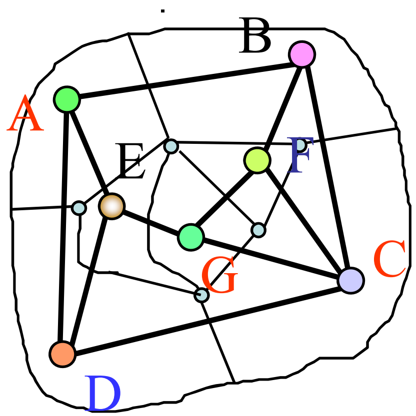
Interconnection graph



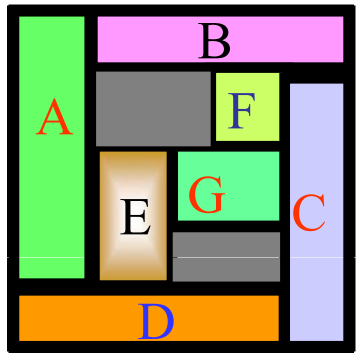
MCM Floorplanning
Architectural Floorplanning



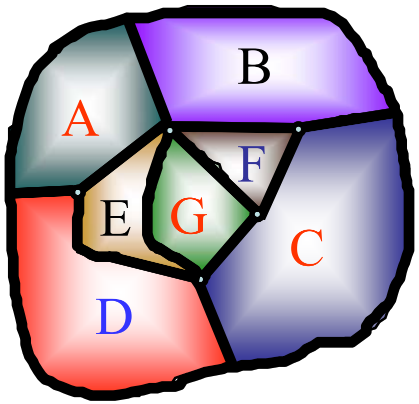
Dual-like graph



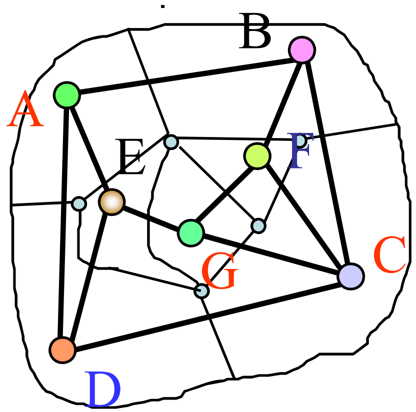
Interconnection graph



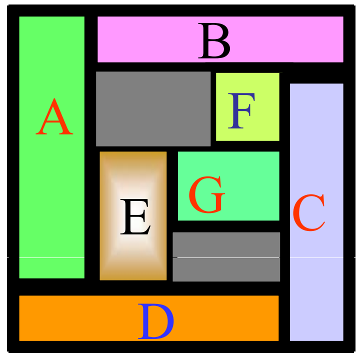
MCM Floorplanning
Architectural Floorplanning



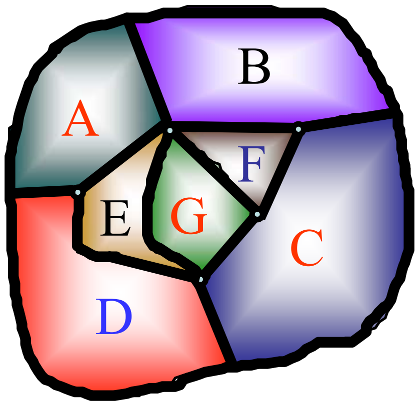
Dual-like graph



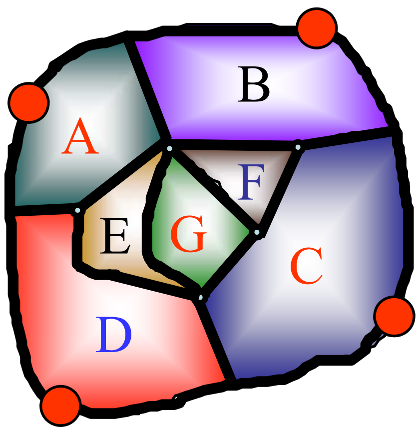
Interconnection graph



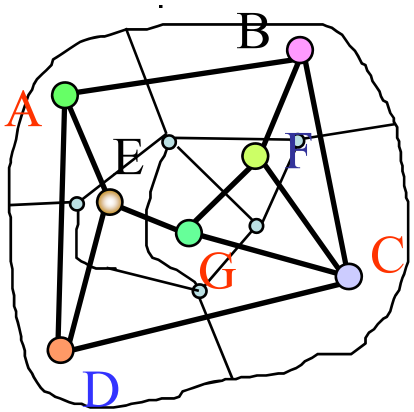
MCM Floorplanning
Architectural Floorplanning



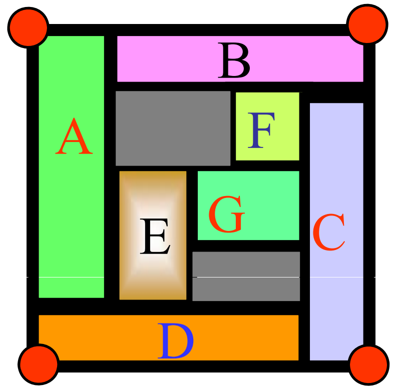
Dual-like graph



Box-Rectangular drawing

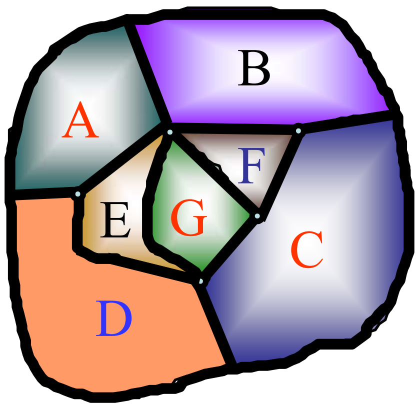


Interconnection graph

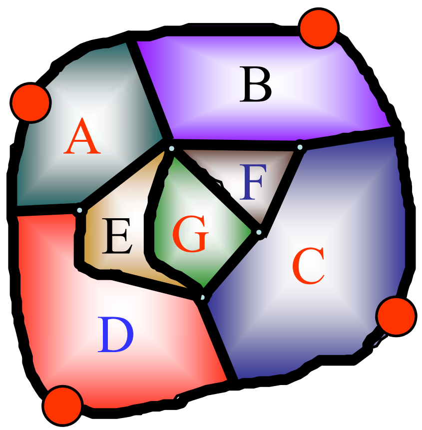


dead space

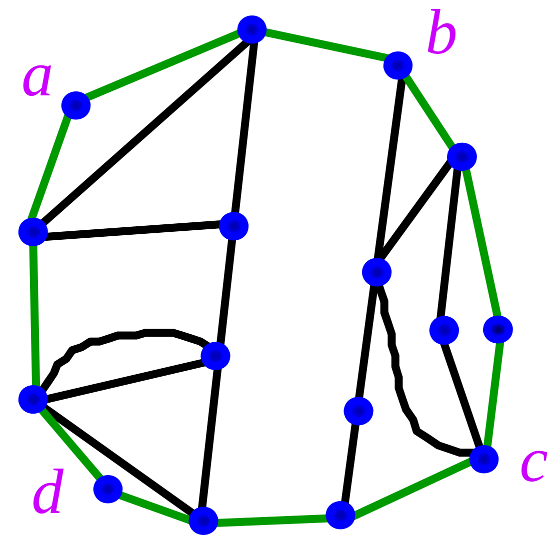
MCM Floorplanning
Architectural Floorplanning



Dual-like graph



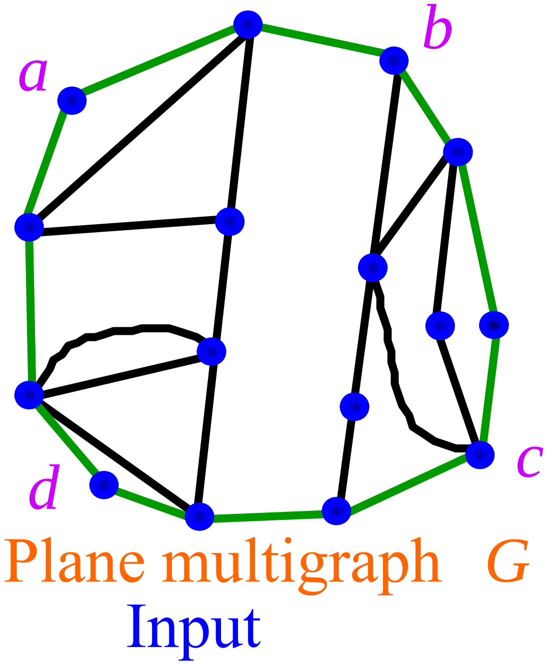
Box-Rectangular Drawing

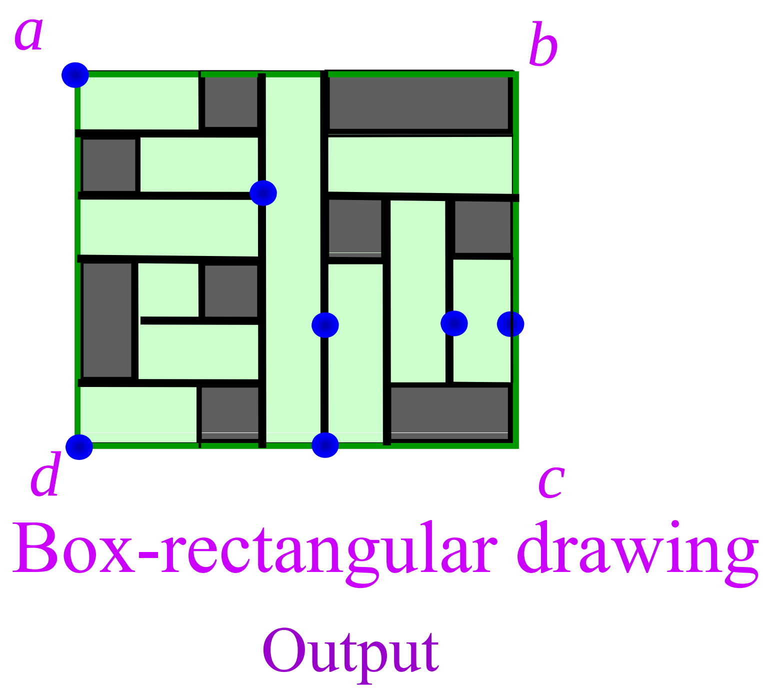


Plane multigraph G

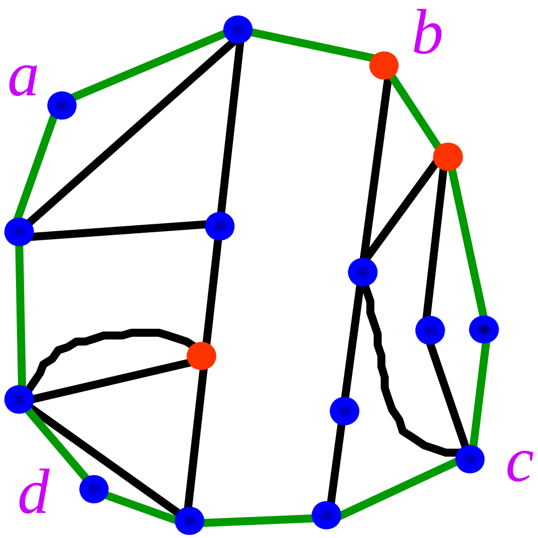
Input

Box-Rectangular Drawing



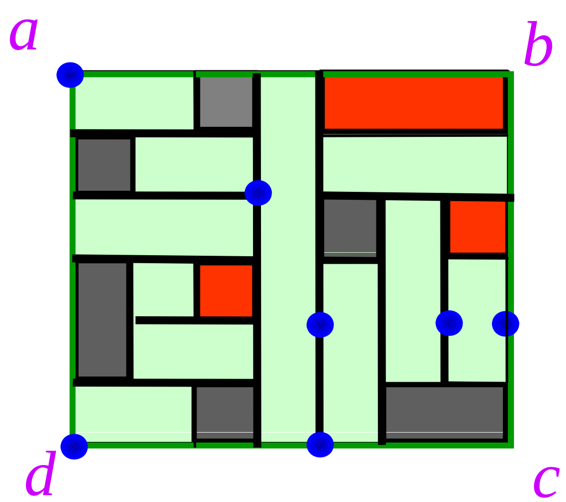


Box-Rectangular Drawing



Plane multigraph G

Input

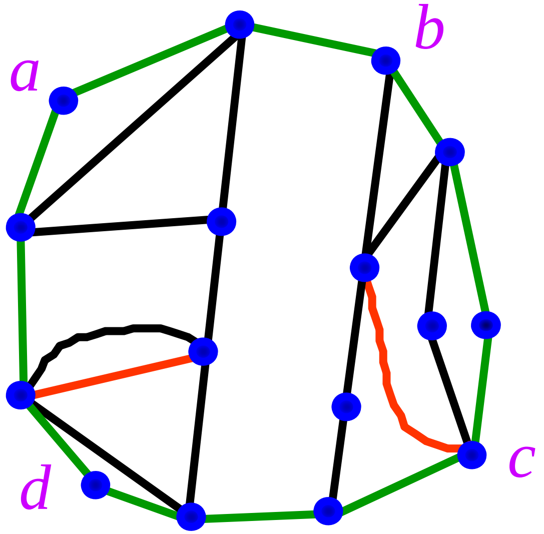


Box-rectangular drawing

Output

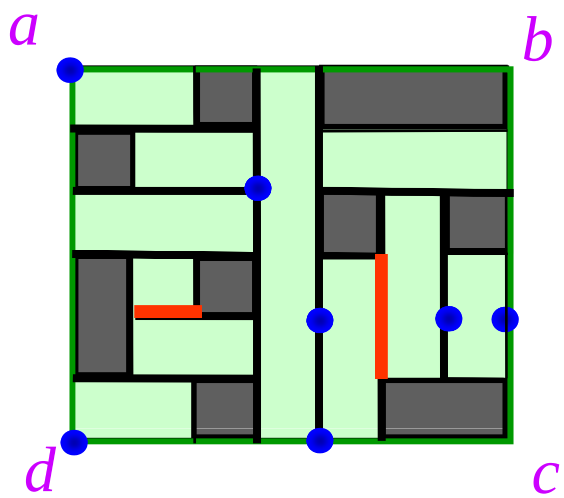
- Each vertex is drawn as a rectangle.

Box-Rectangular Drawing



Plane multigraph G

Input

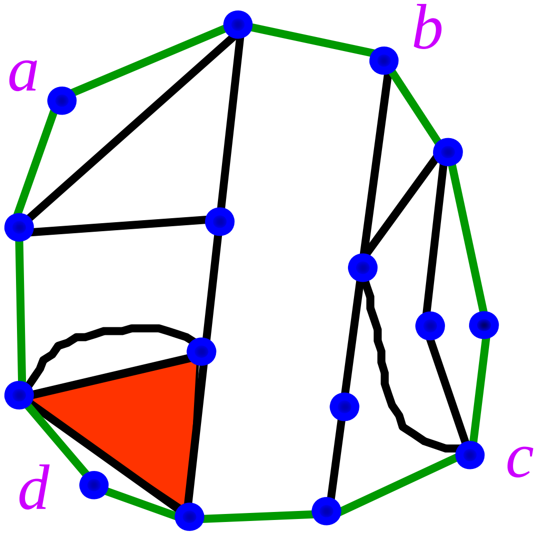


Box-rectangular drawing

Output

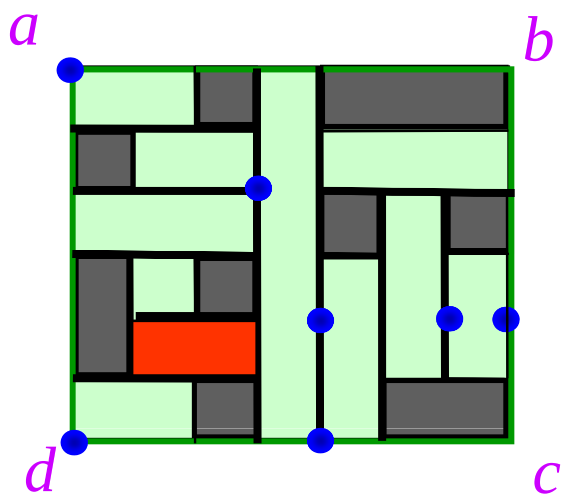
- Each vertex is drawn as a rectangle.
- Each edge is drawn as a horizontal or a vertical line segment.

Box-Rectangular Drawing



Plane multigraph *G*

Input



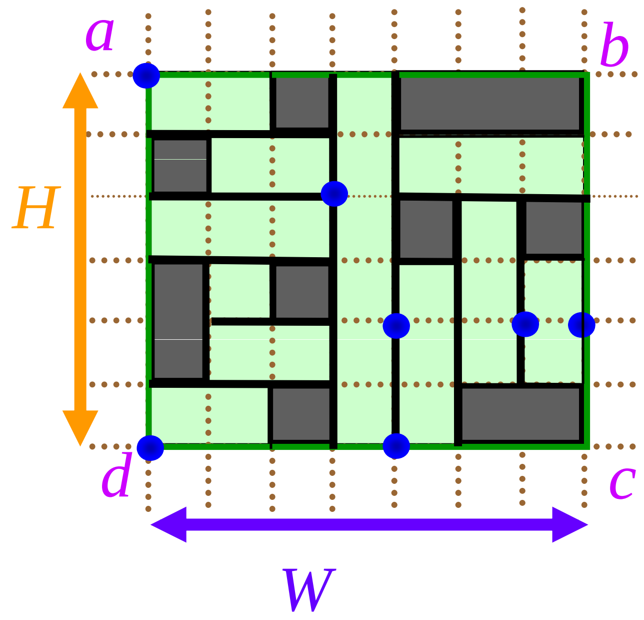
Box-rectangular drawing

Output

- Each vertex is drawn as a rectangle.
- Each edge is drawn as a horizontal or a vertical line segment.
- Each face is drawn as a rectangle.

Rahman *et al.* 2000

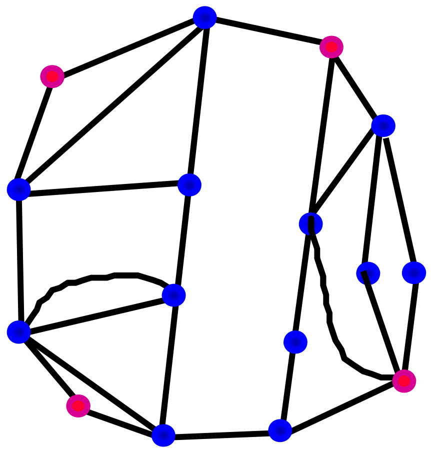
- A necessary and sufficient condition for a plane multigraph to have a box-rectangular drawing.
- A linear-time algorithm.
- $W + H \leq m + 2$, where *m* is the number of edges in G.



Algorithm of Rahman *et al.*

Main Idea: Reduction to a rectangular drawing problem

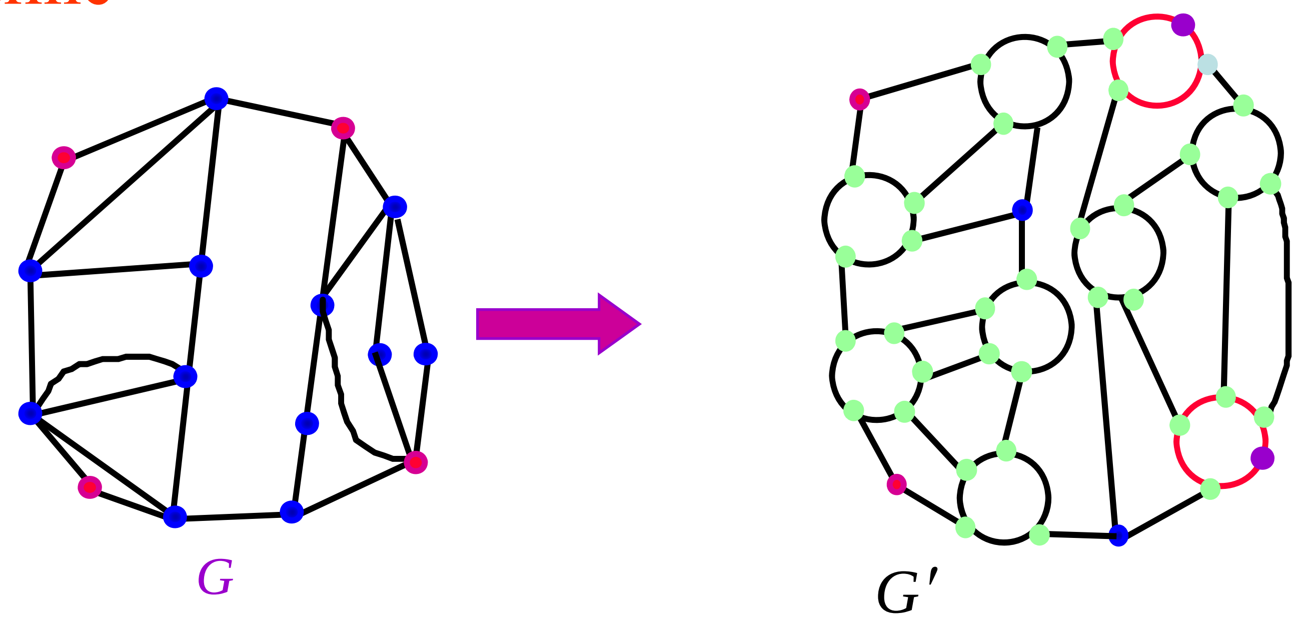
Outline



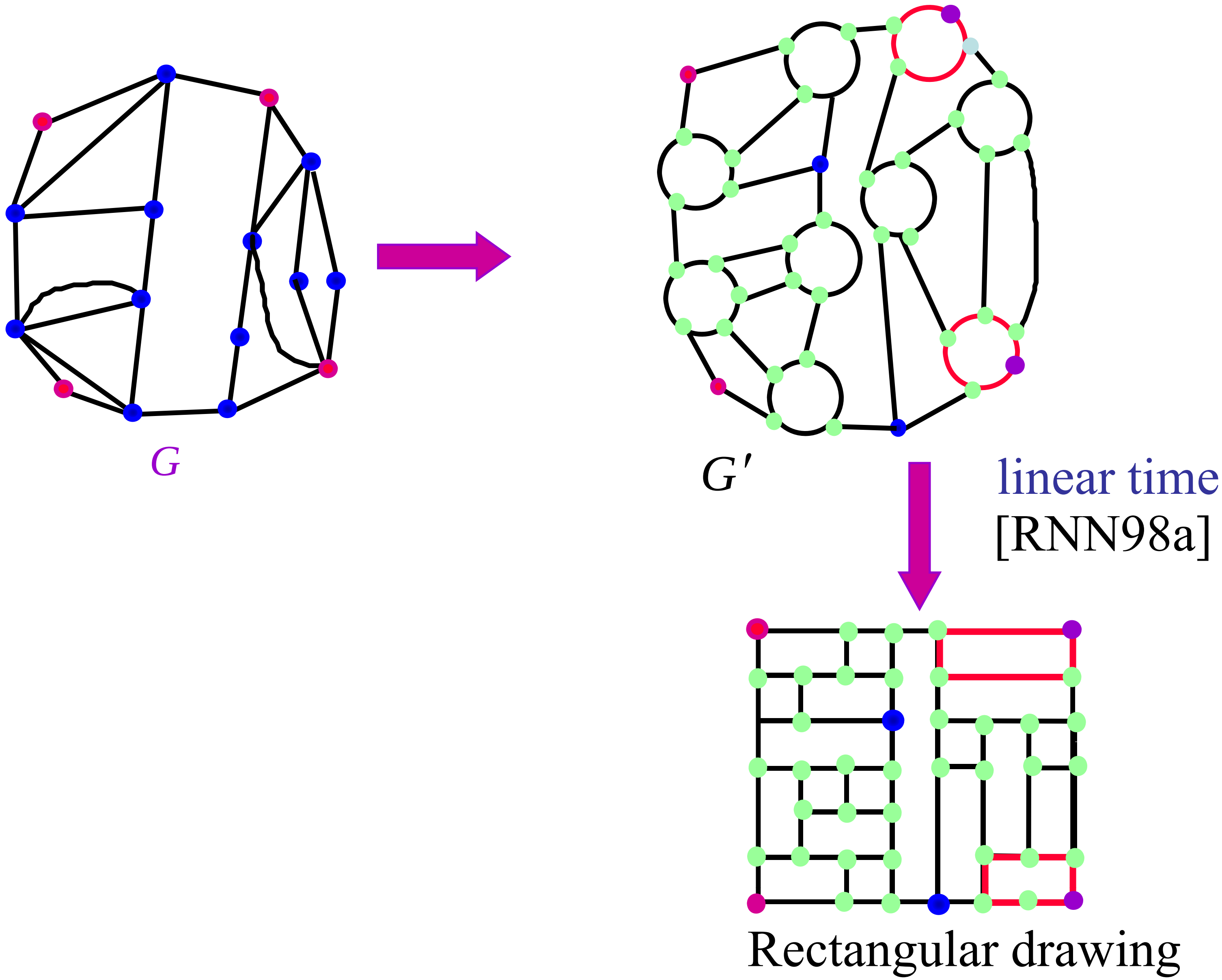
G

Outline

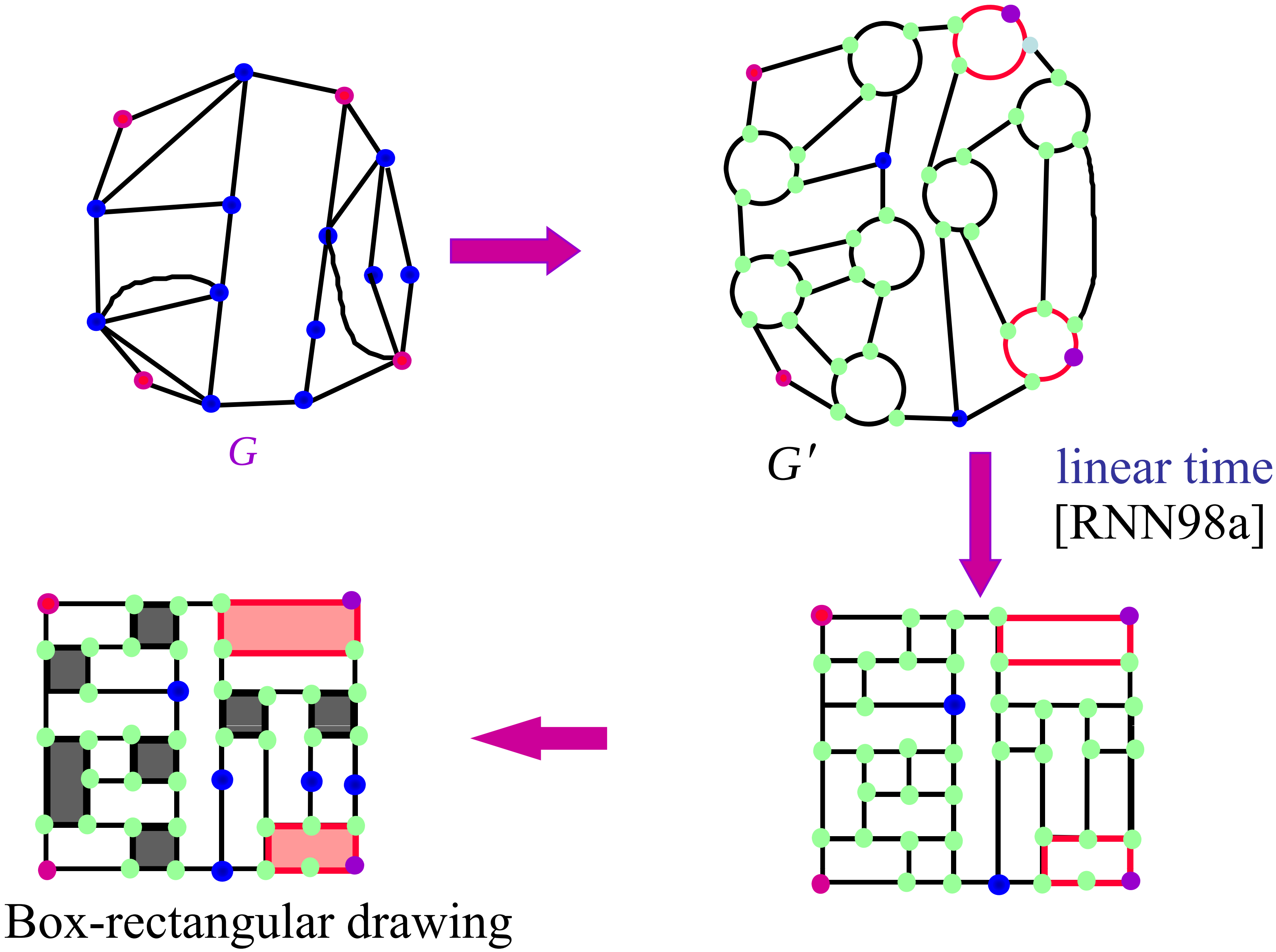
Replace each vertex of degree four or more by a cycle



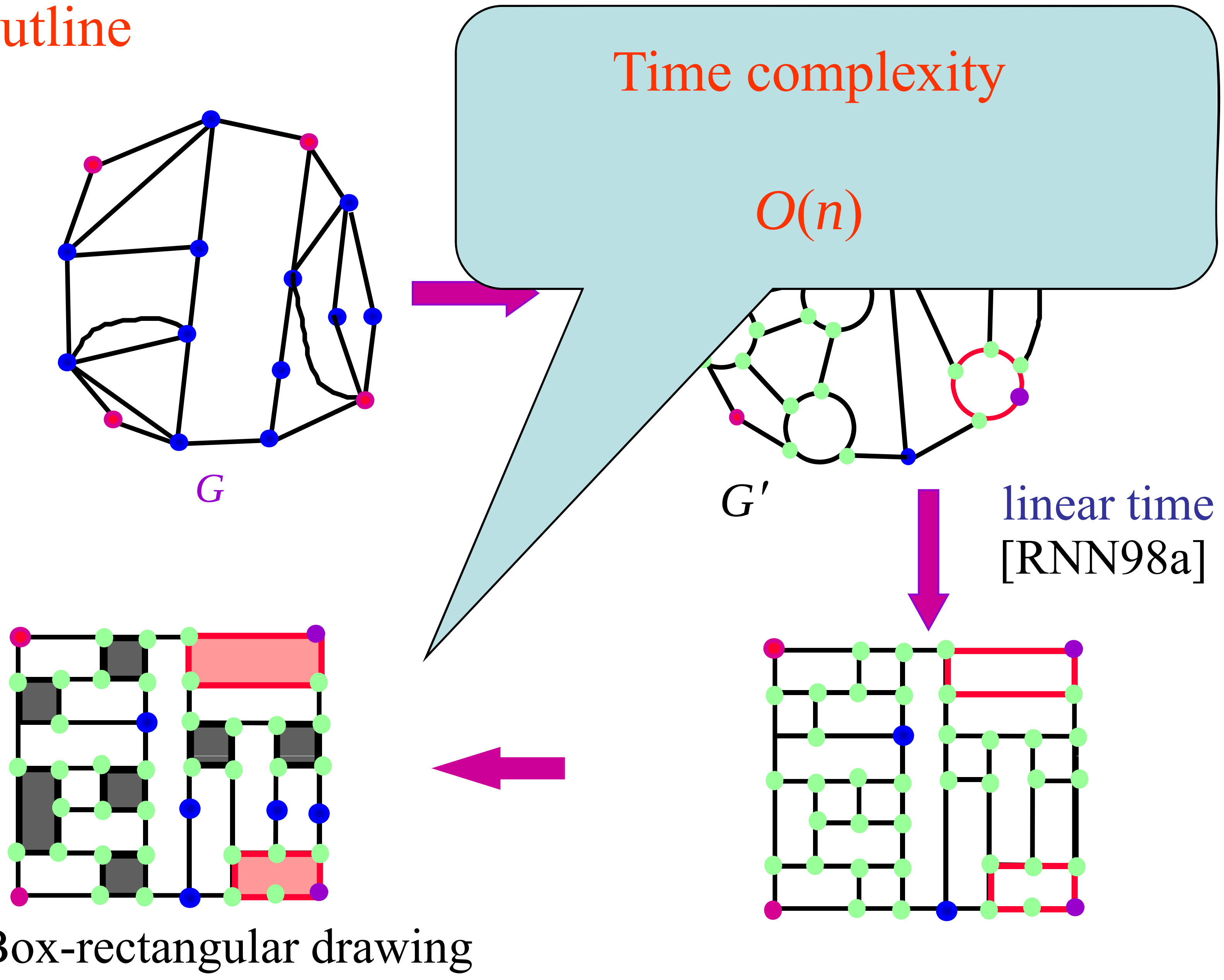
Outline



Outline

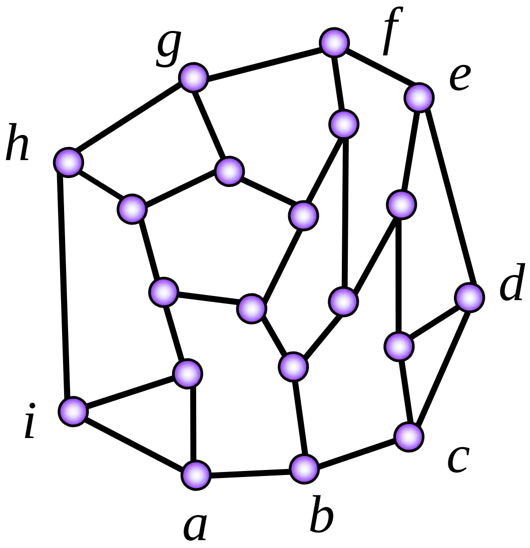


Outline



Time complexity
$O(n)$
G
G'
linear time
[RNN98a]
Box-rectangular drawing

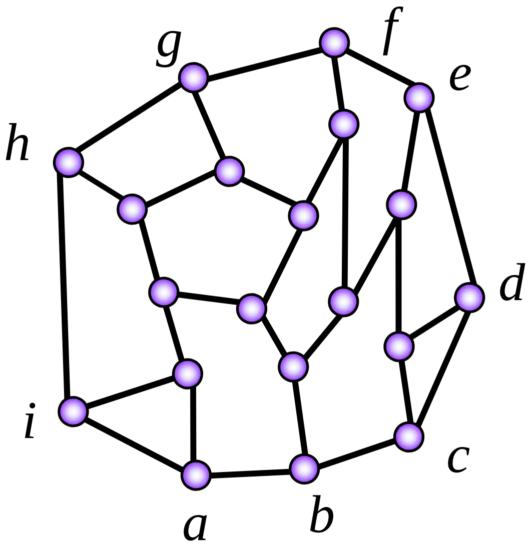
Orthogonal Drawings



plane graph G

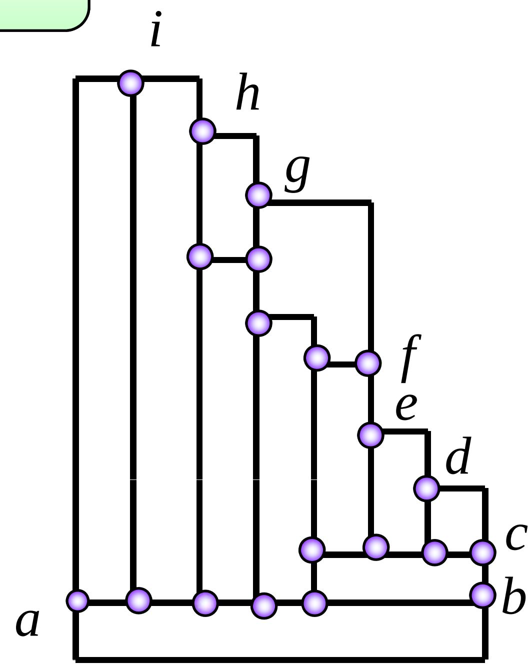
Input

Orthogonal Drawings



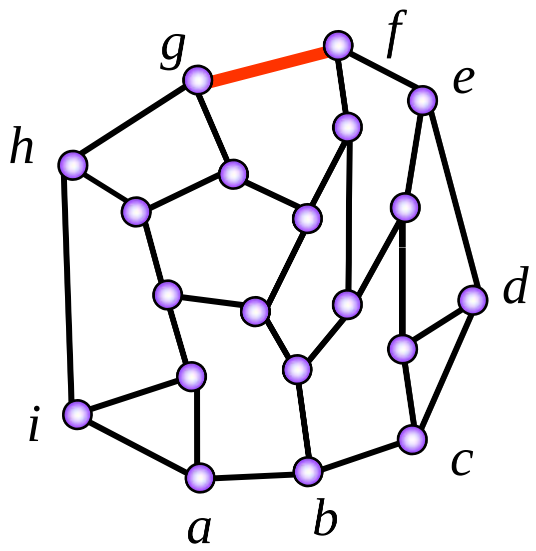
plane graph *G*

Input



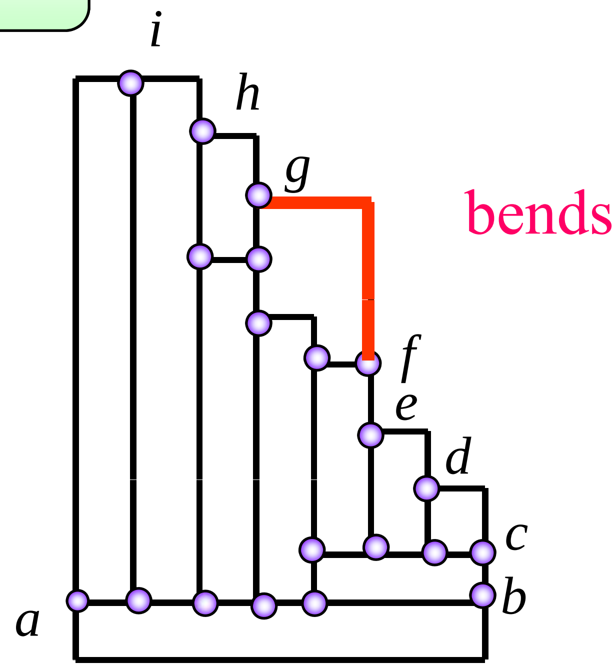
Output

Orthogonal Drawings



plane graph G

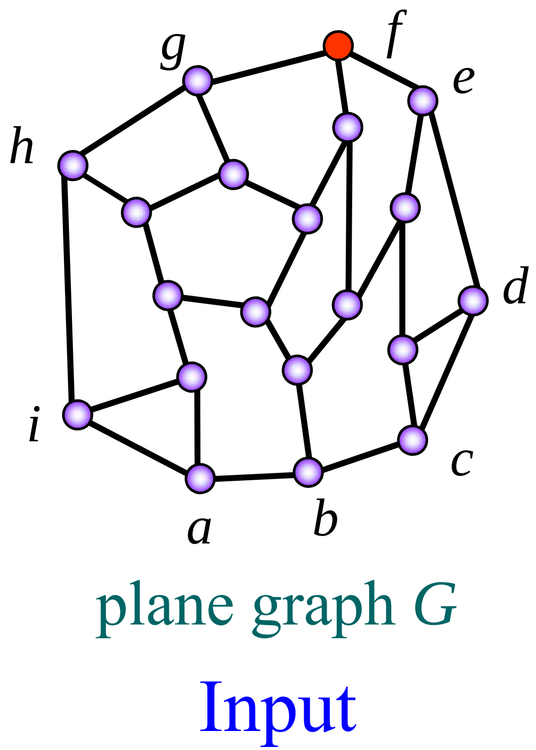
Input

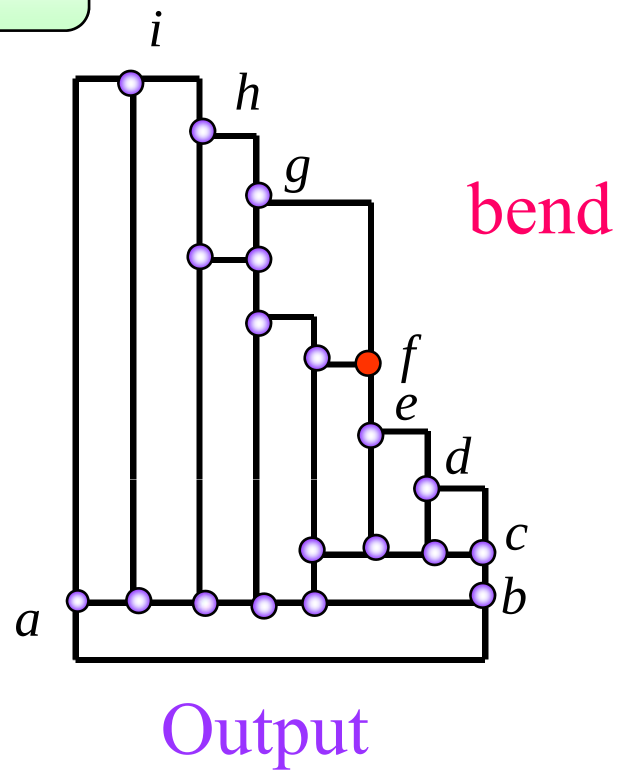


Output

- Each edge is drawn as an alternating sequence of horizontal and vertical line segments.

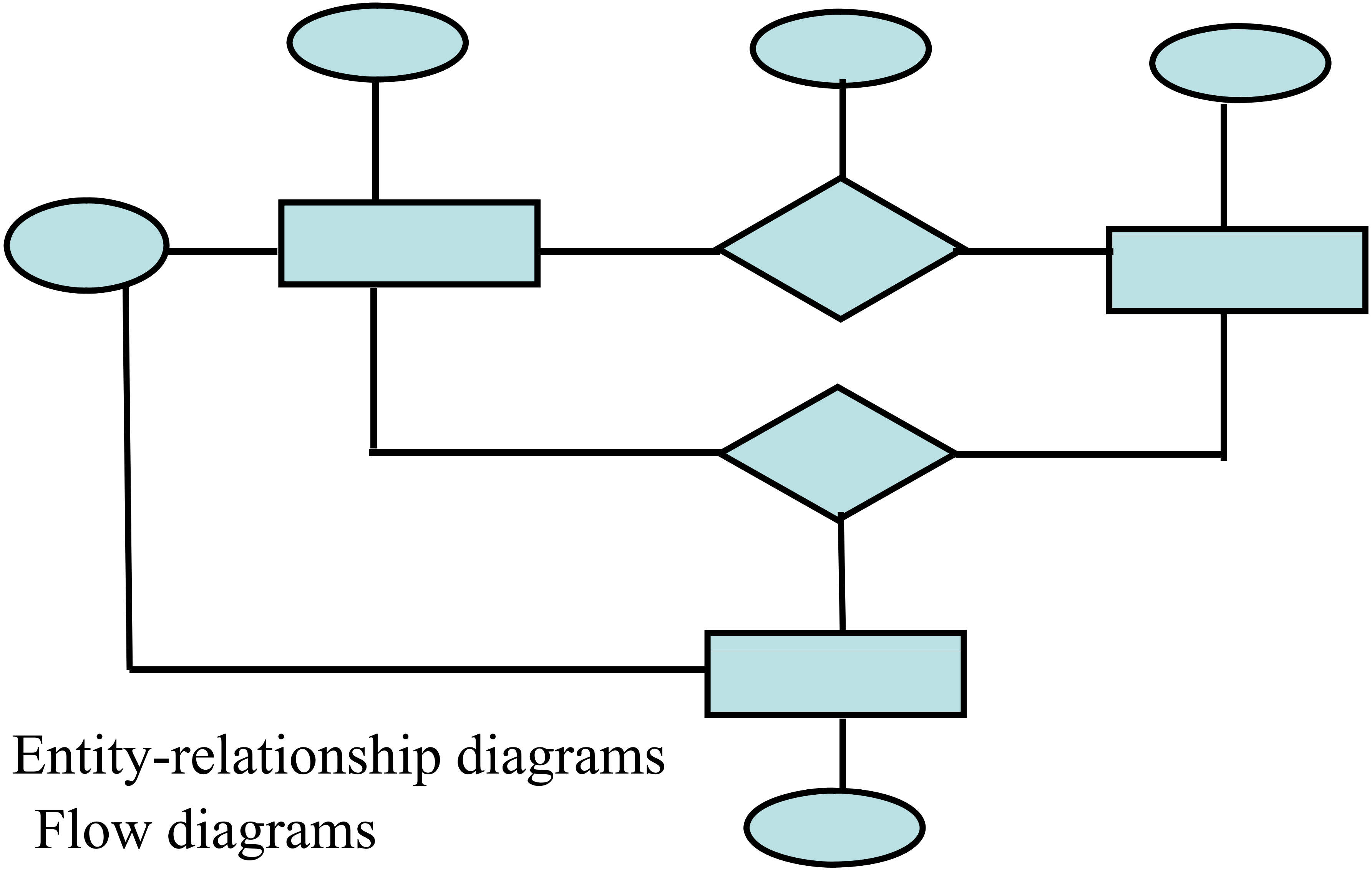
Orthogonal Drawings





- Each edge is drawn as an alternating sequence of horizontal and vertical line segments.
- Each vertex is drawn as a point.

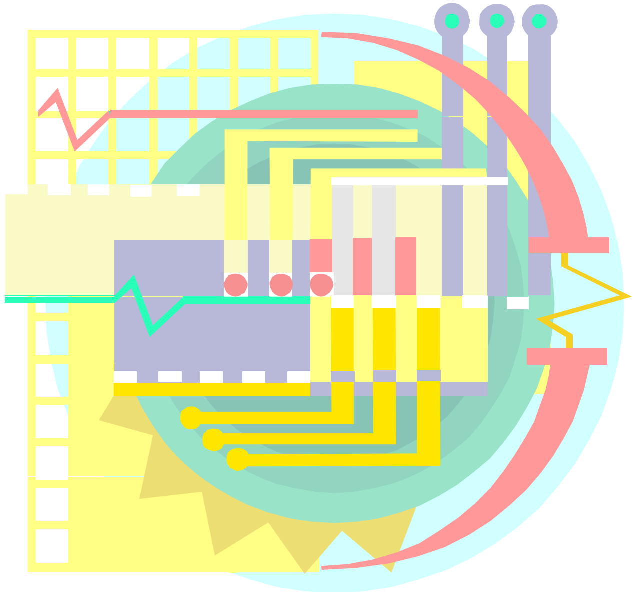
Applications

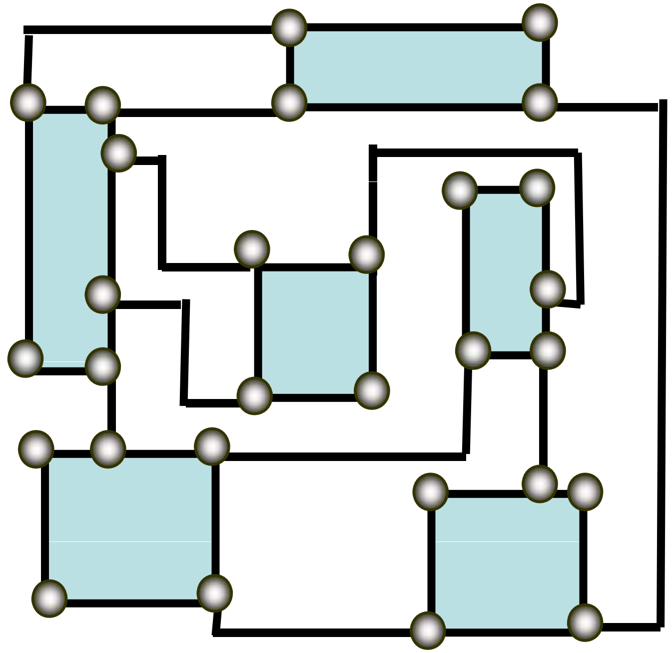


Entity-relationship diagrams

Flow diagrams

Applications

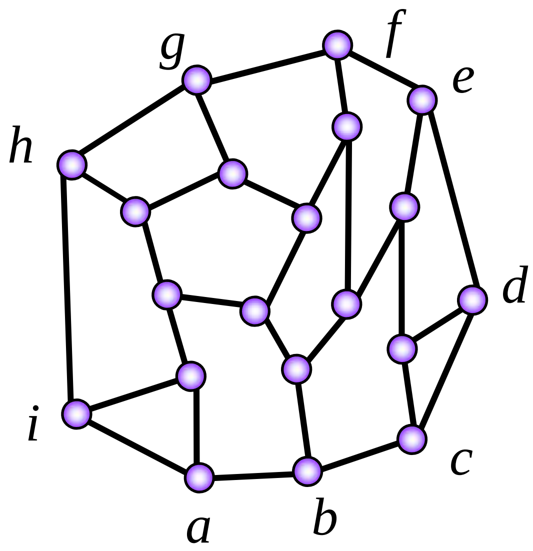




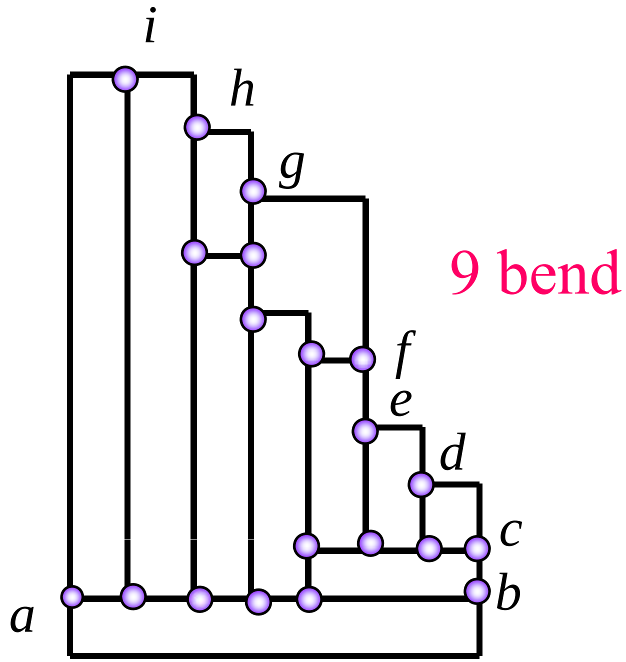
Circuit schematics

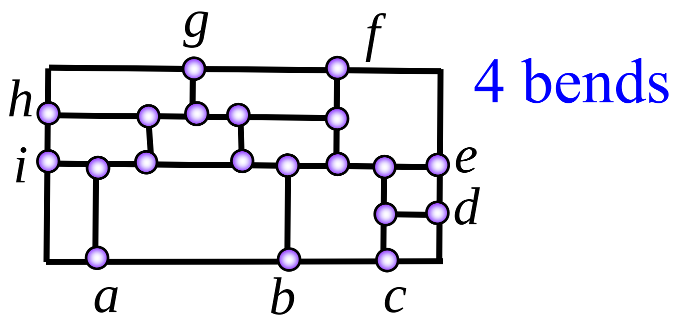
Minimization of bends reduces the number of “vias” or “throughholes,” and hence reduces VLSI fabrication costs.

Objective



plane graph *G*





To minimize the number of bends in an orthogonal drawing.

minimum number of bends.

Bend-Minimum Orthogonal Drawing

Garg and Tamassia '96

$O(n^{7/4}\log^{1/2} n)$ time for plane graph of $\Delta \leq 4$

Idea reduction to a minimum cost flow problem

Rahman, Nakano and Nishizeki '99

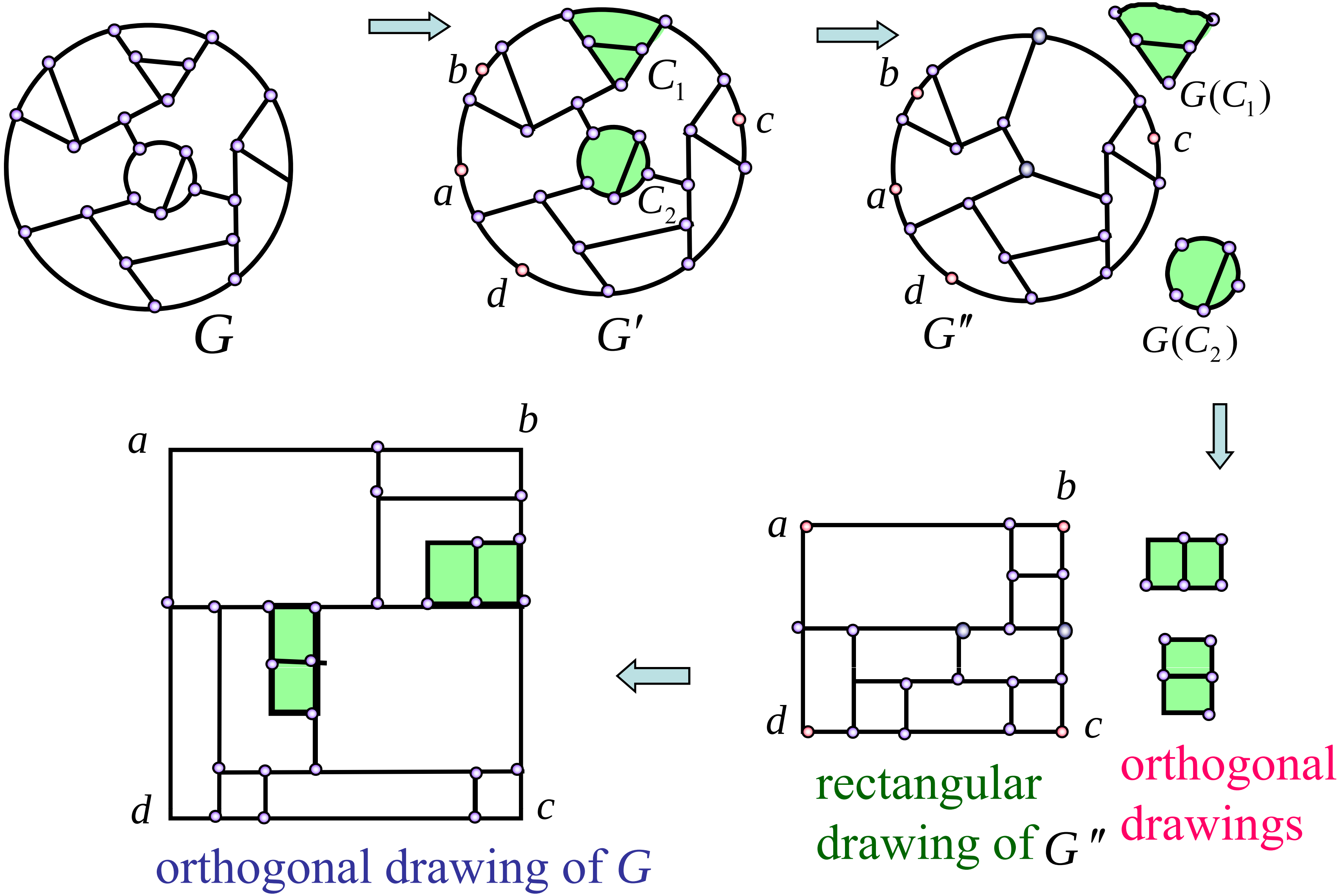
Linear for 3-connected cubic plane graph

Idea reduction to a rectangular drawing problem.

Rahman and Nishizeki '02

Linear for plane graph of $\Delta \leq 3$

Outline of the algorithm of [RNN99]



Open Problems and Future Research Direction

- More area-efficient straight-line or convex drawing algorithms.

  $$W \times H \le \left(\frac{2}{3}n\right)^2$$ for every planar graph?

- Linear algorithm for rectangular drawings of plane graphs of $\Delta \le 4$.

- Linear algorithm for bend-minimal orthogonal drawings of plane graphs of $\Delta \le 4$.

- Drawing of plane graphs with constraints like

Practical Applications ⇐
- Prescribed face areas
- Prescribed length of some edges
- Prescribed positions of some vertices

Book

Planar Graph Drawing

by

Takao Nishizeki
Md. Saidur Rahman

http://www.nishizeki.ecei.tohoku.ac.jp/nszk/saidur/gdbook.html

Book

Planar Graph Drawing

by

http://www.nishizeki.ecei.tohoku.ac.jp/nszk/saidur/gdbook.html

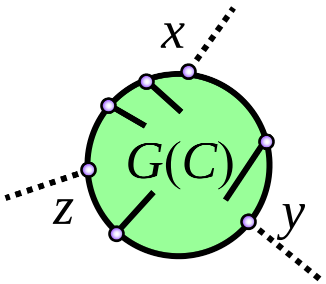
ISAAC in Sendai, Tohoku

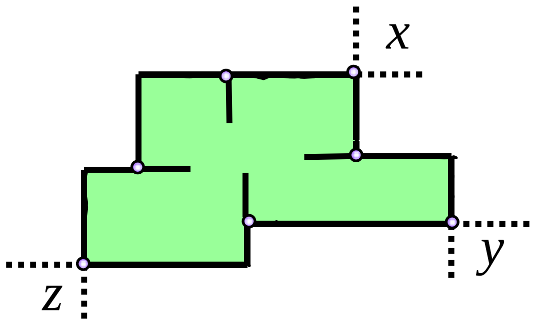
Probably 2007


Sendai
Kyoto
Tokyo

Thank You

Properties of a drawing of $G(C)$

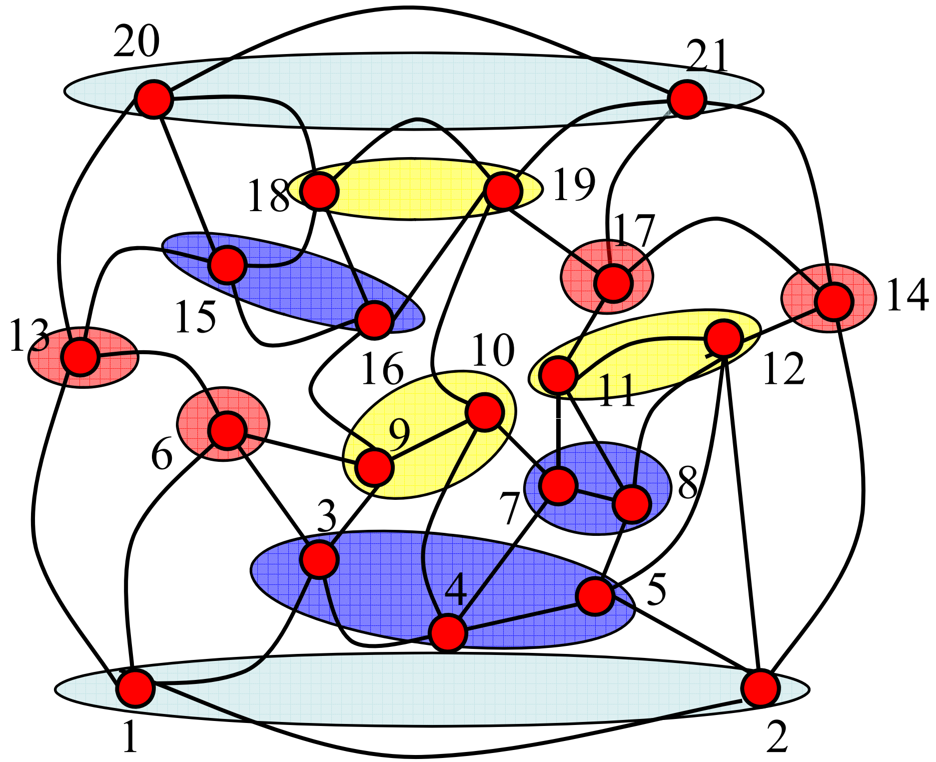




orthogonal drawing of $G(C)$

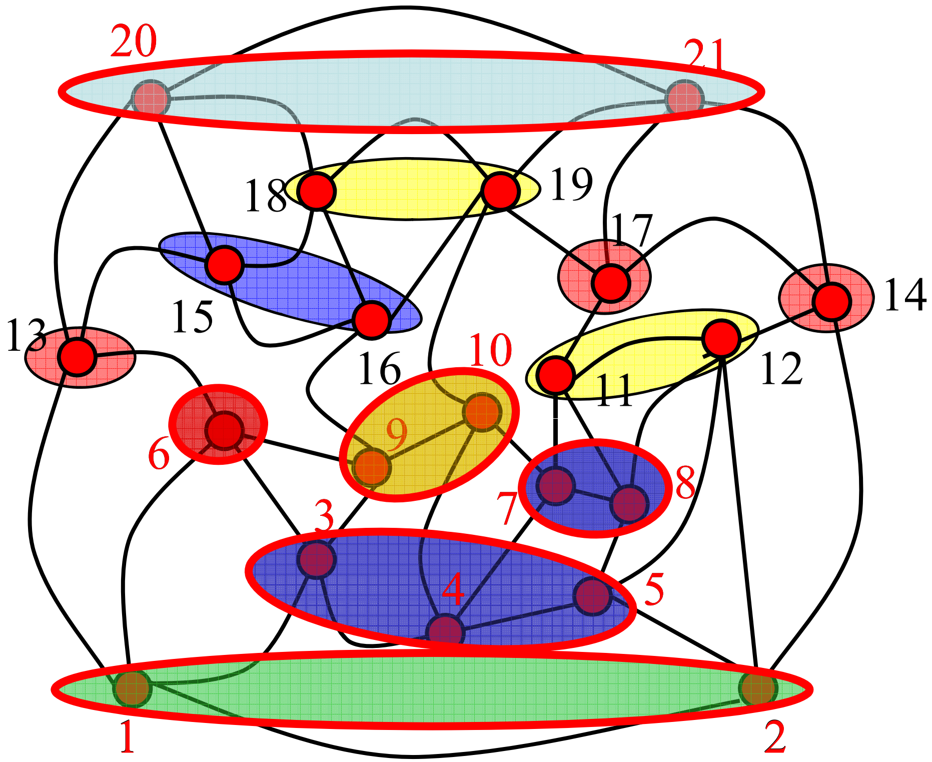
- minimum number of bends
- the six open halflines are free

Main idea

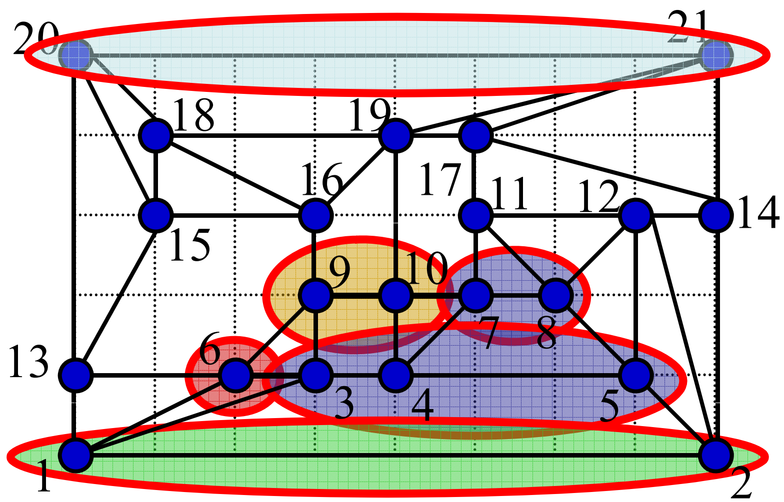


4-canonical decomposition

Main idea

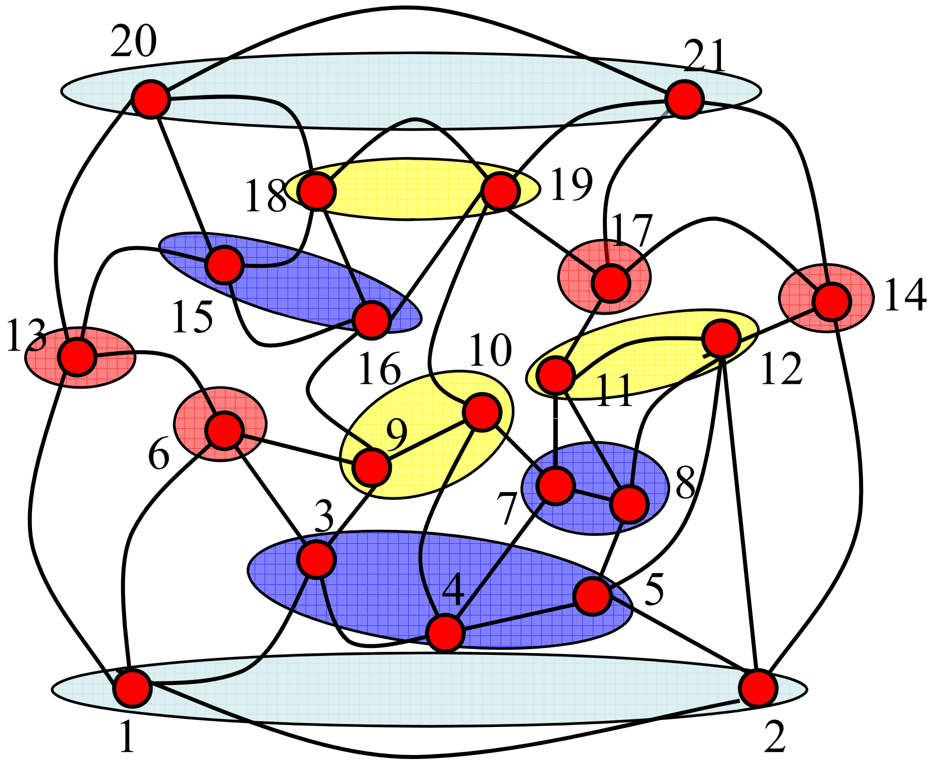


4-canonical decomposition

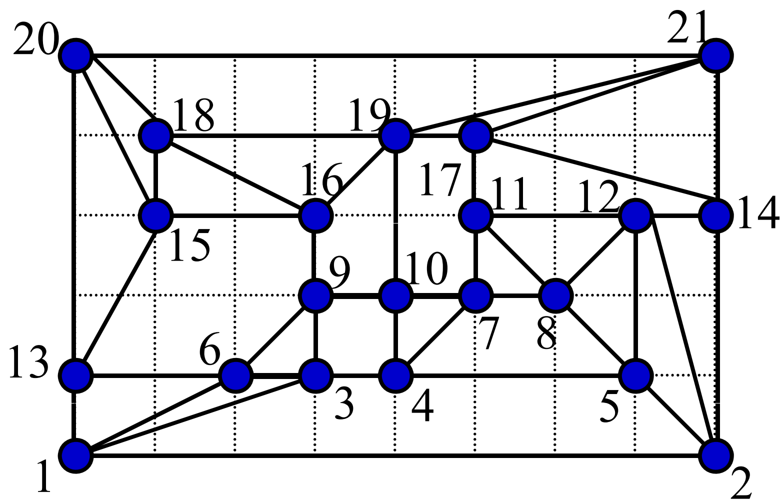


Add a group of vertices one by one.

Main idea

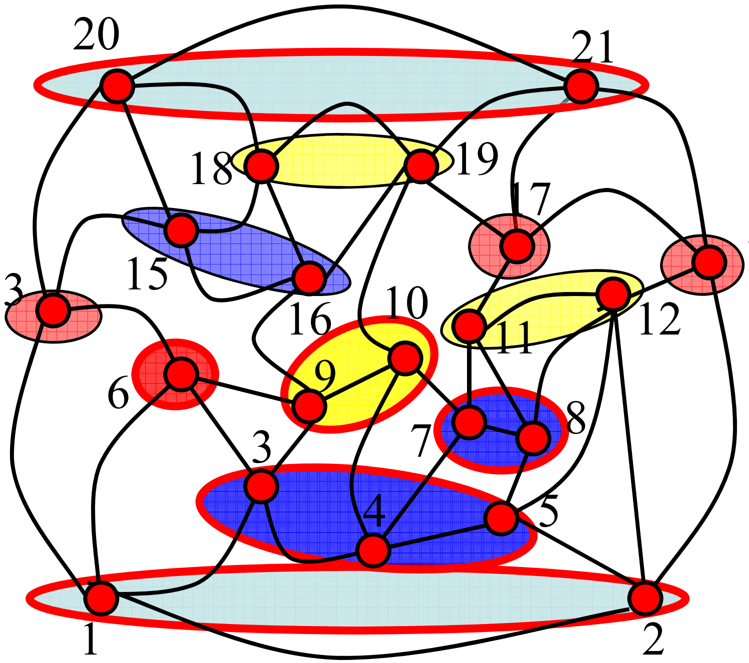


4-canonical decomposition



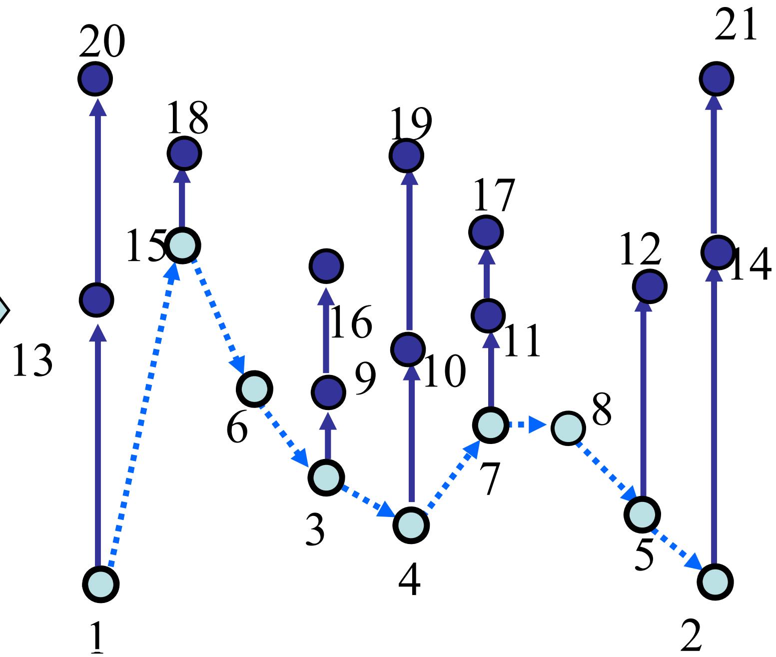
Add a group of vertices one by one.

Main idea



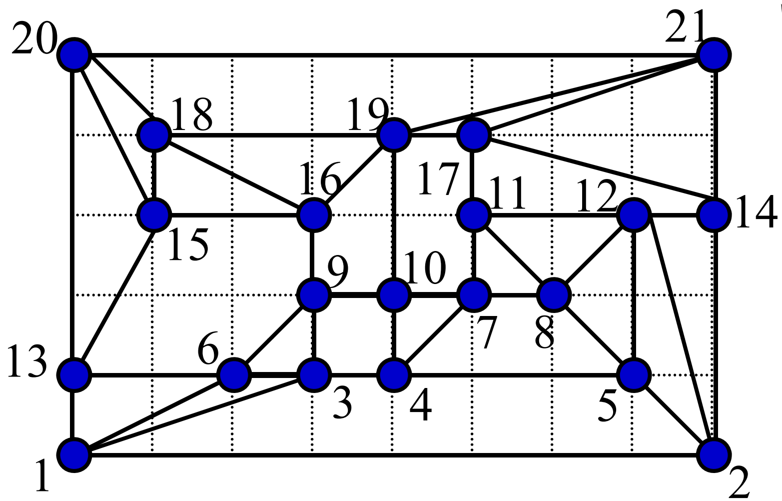
1:4-canonical decomposition

$O(n)$[NRN97]



2: Find paths

3: Decide x-coordinates

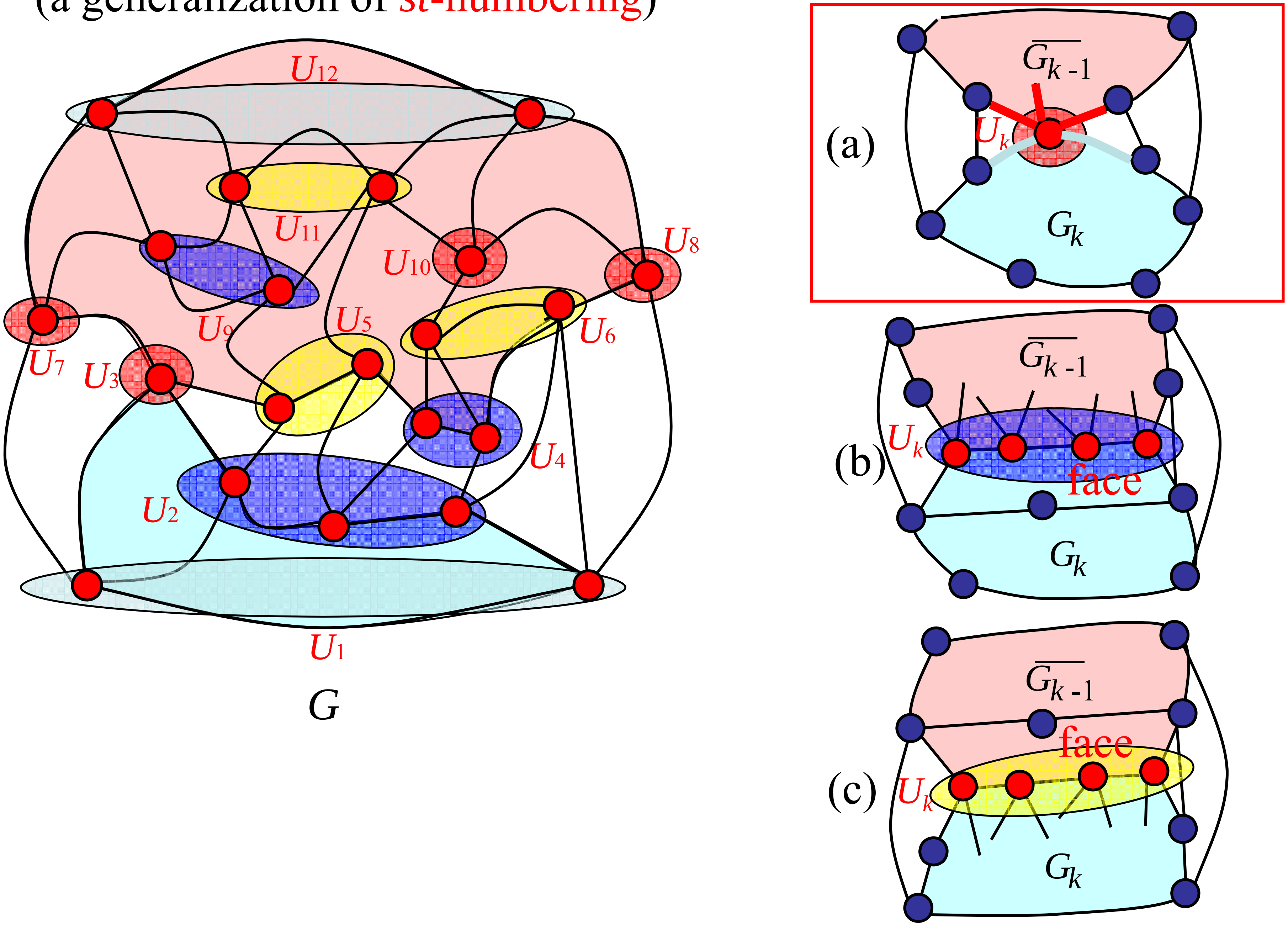


4: Decide y-coordinates

Time complexity: $O(n)$

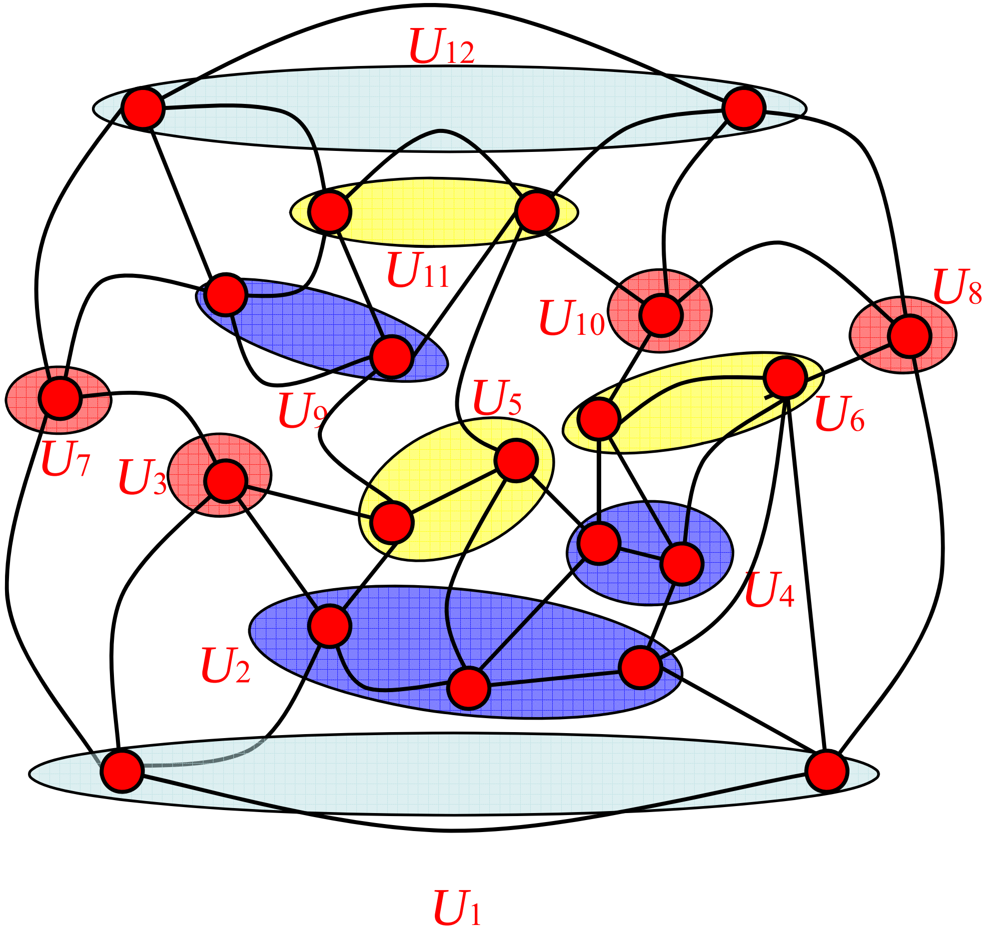
4-canonical decomposition[NRN97]

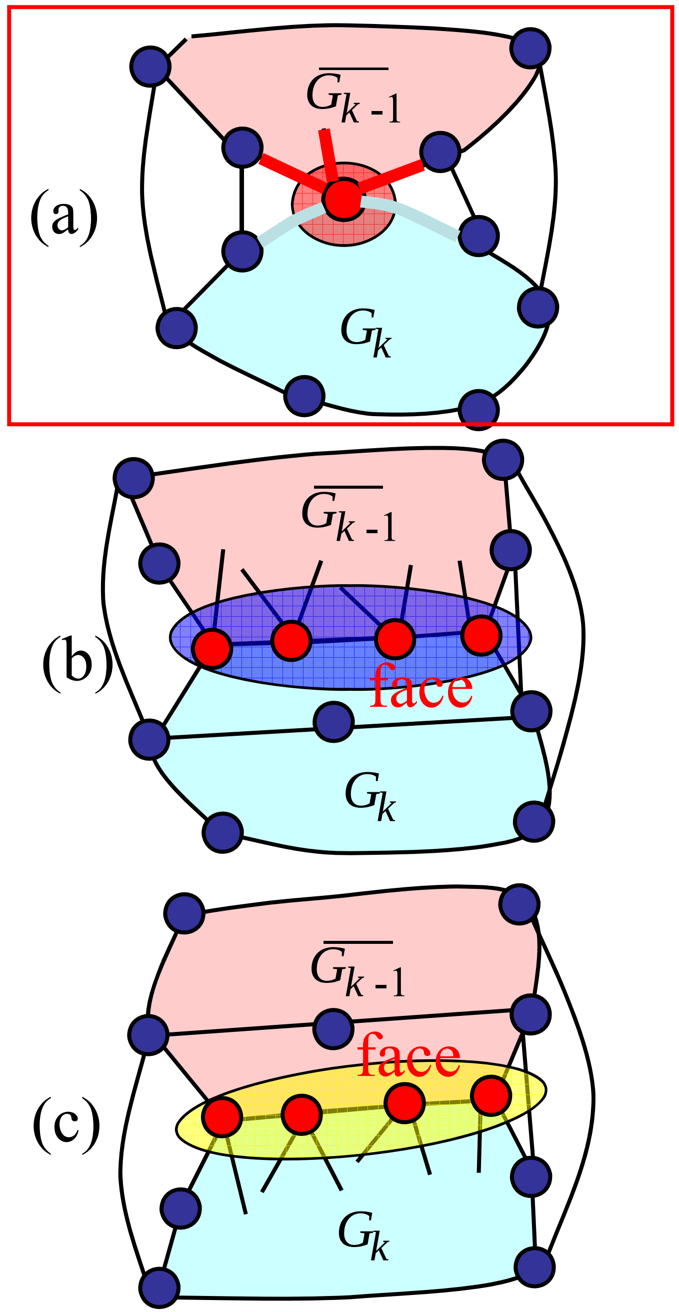
(a generalization of *st*-numbering)



4-canonical decomposition[NRN97]

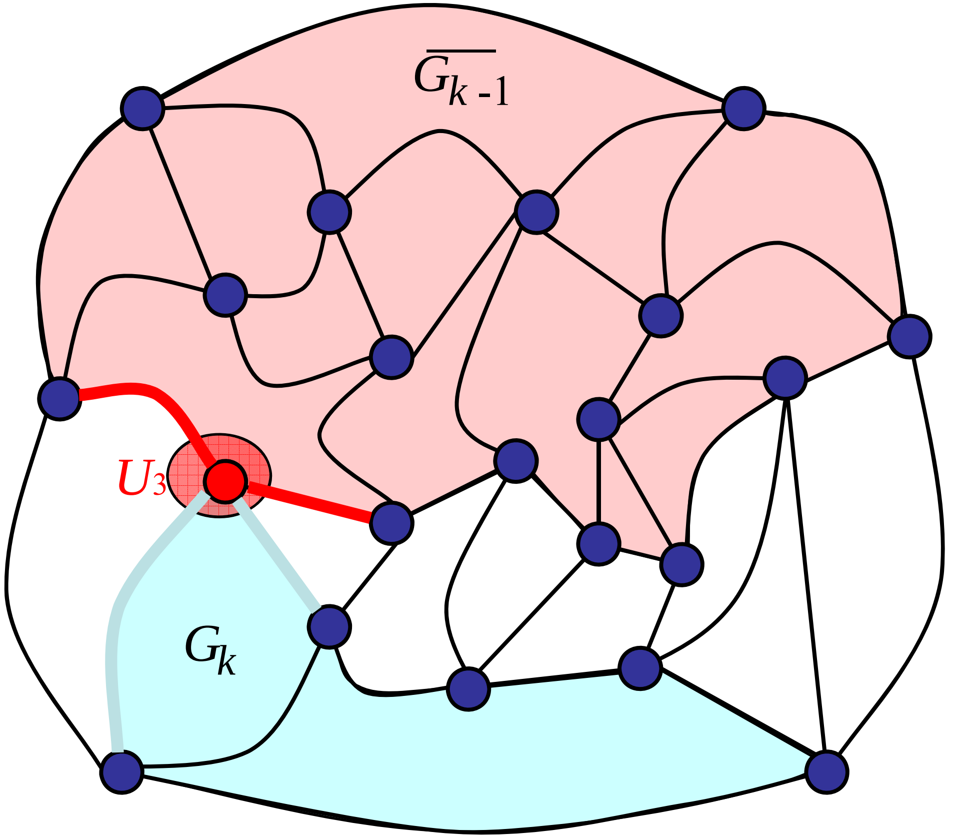
(a generalization of *st*-numbering)

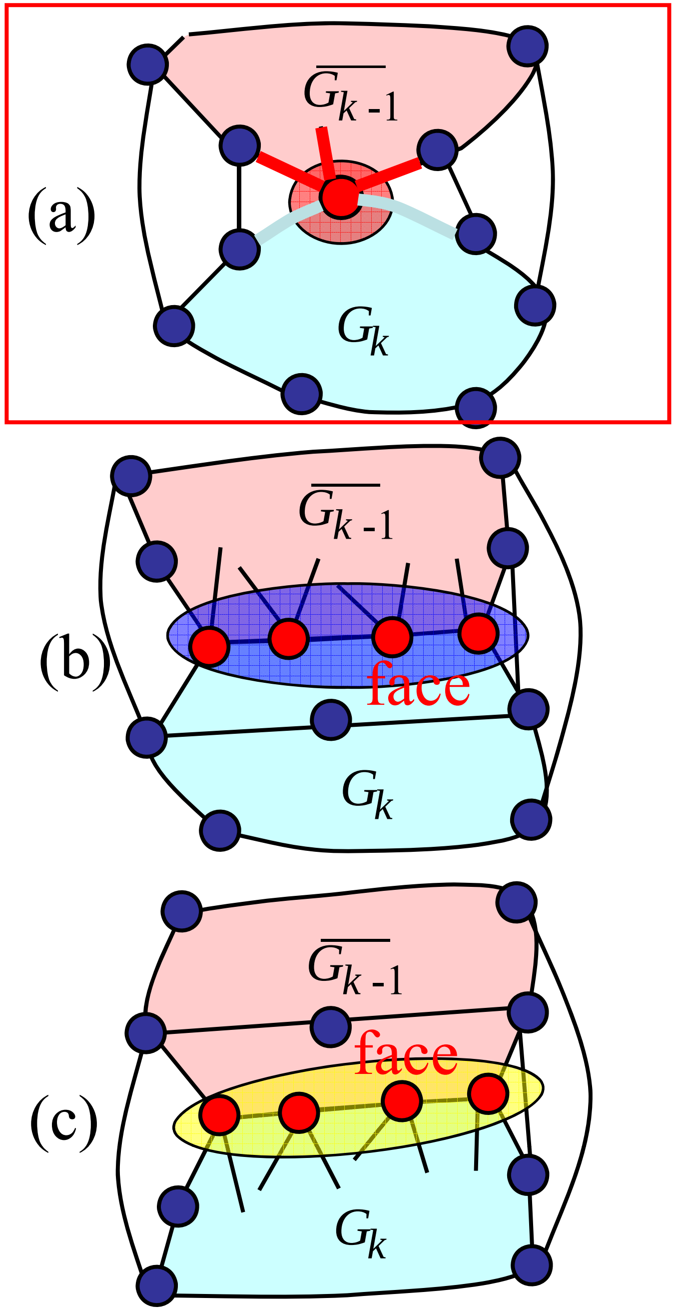




4-canonical decomposition[NRN97]

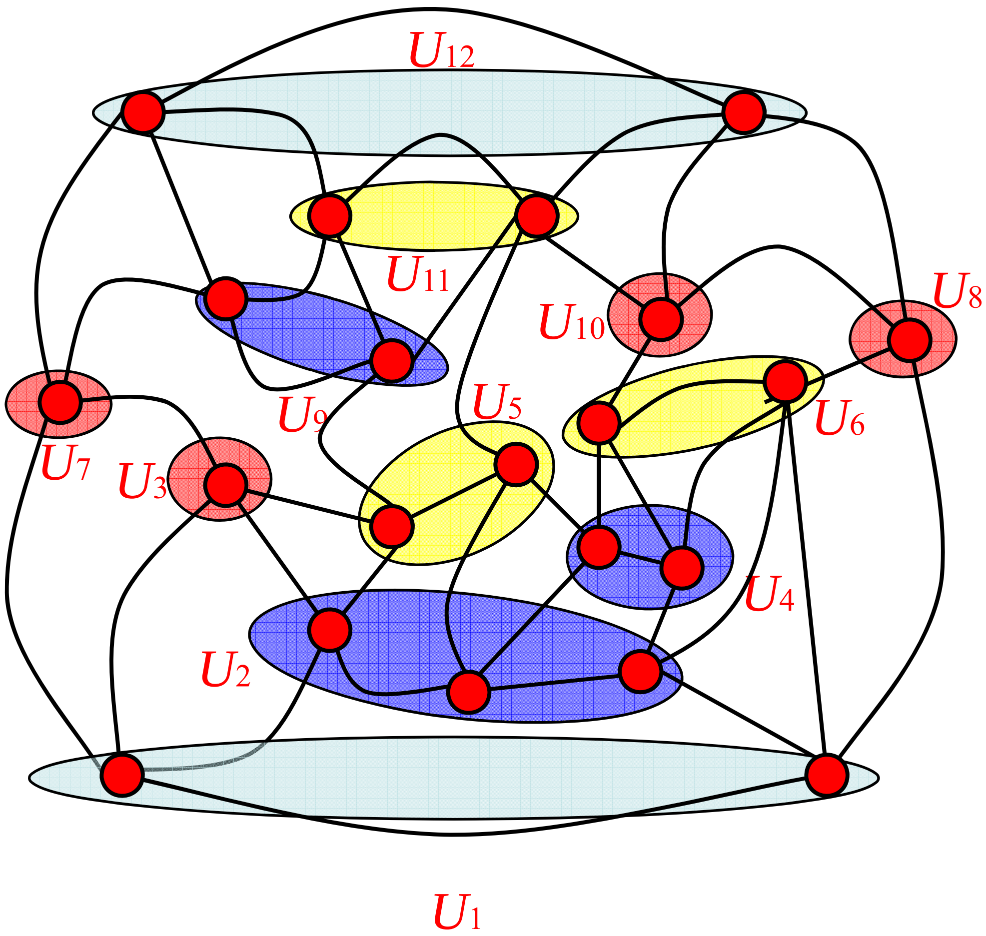
(a generalization of *st*-numbering)

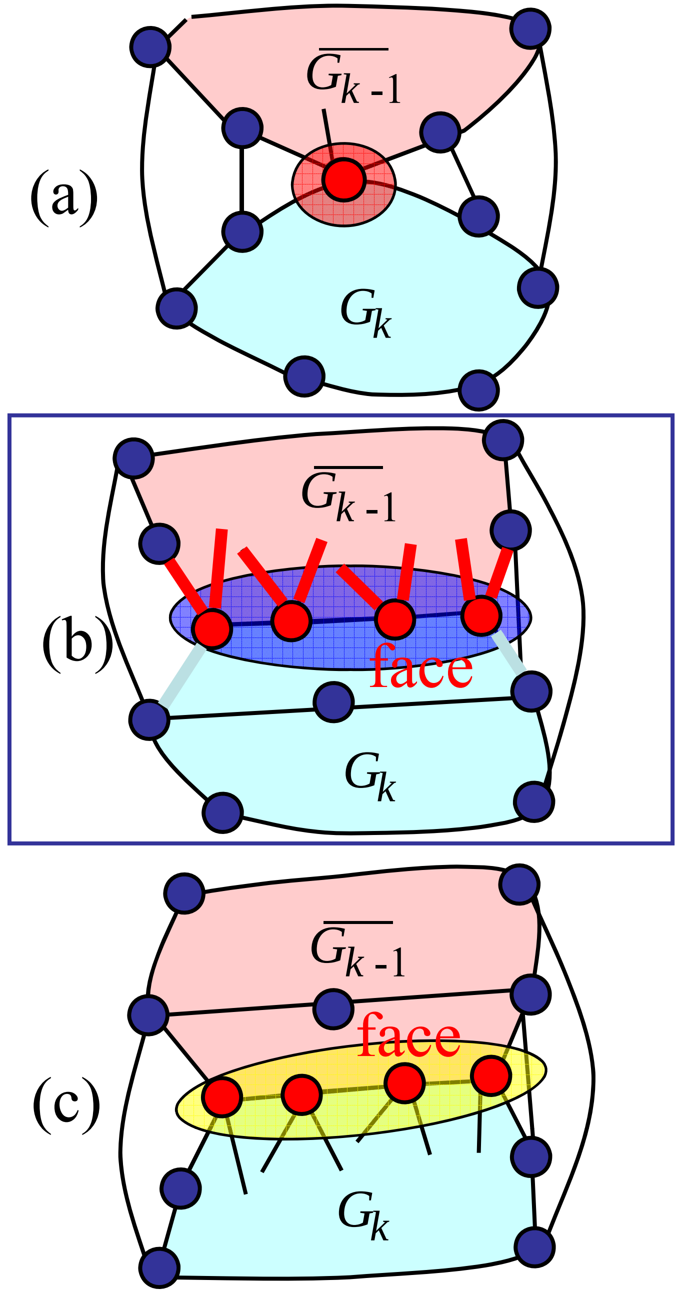




4-canonical decomposition[NRN97]

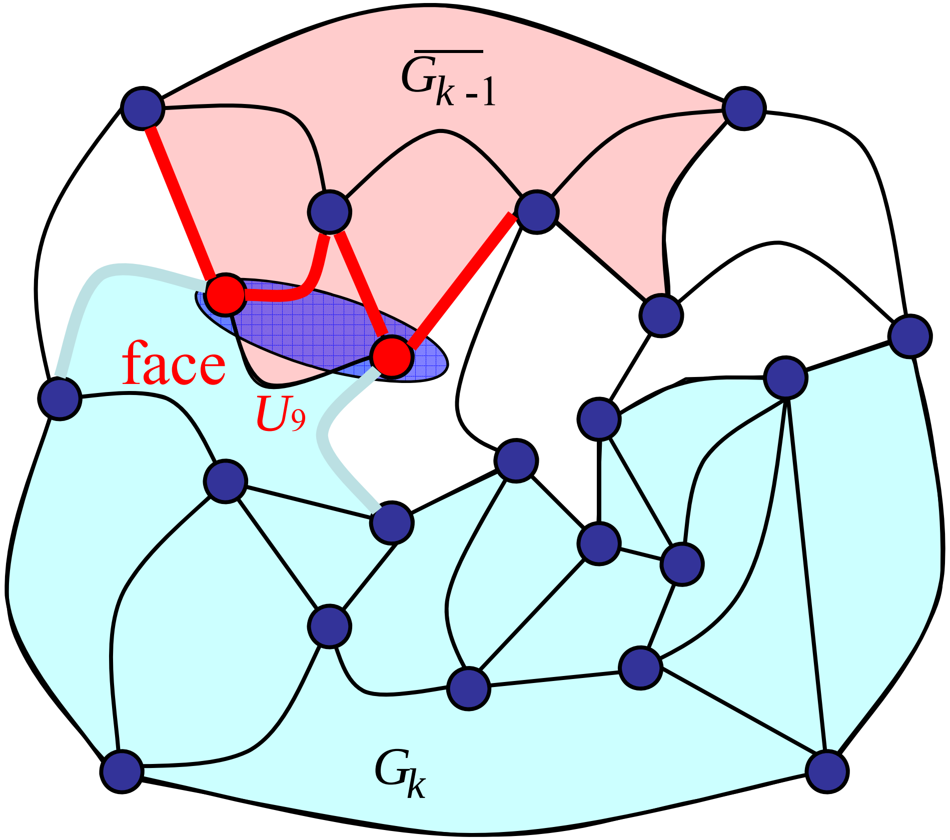
(a generalization of *st*-numbering)

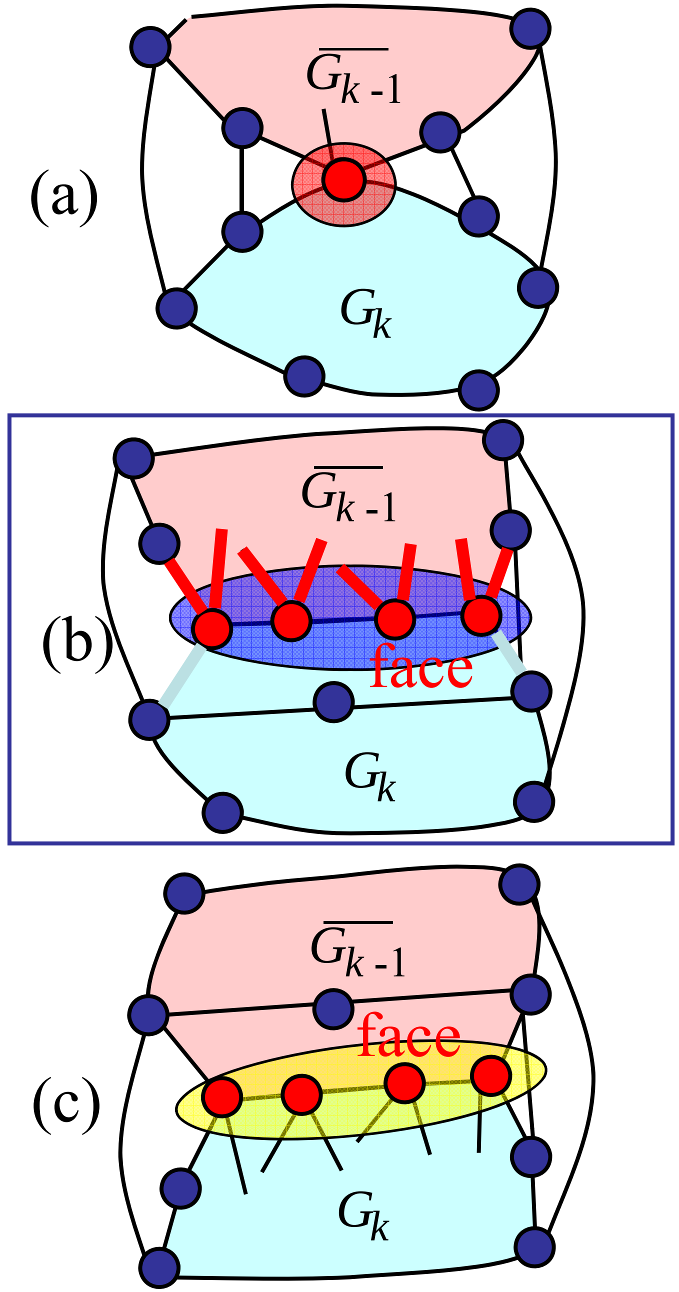




4-canonical decomposition[NRN97]

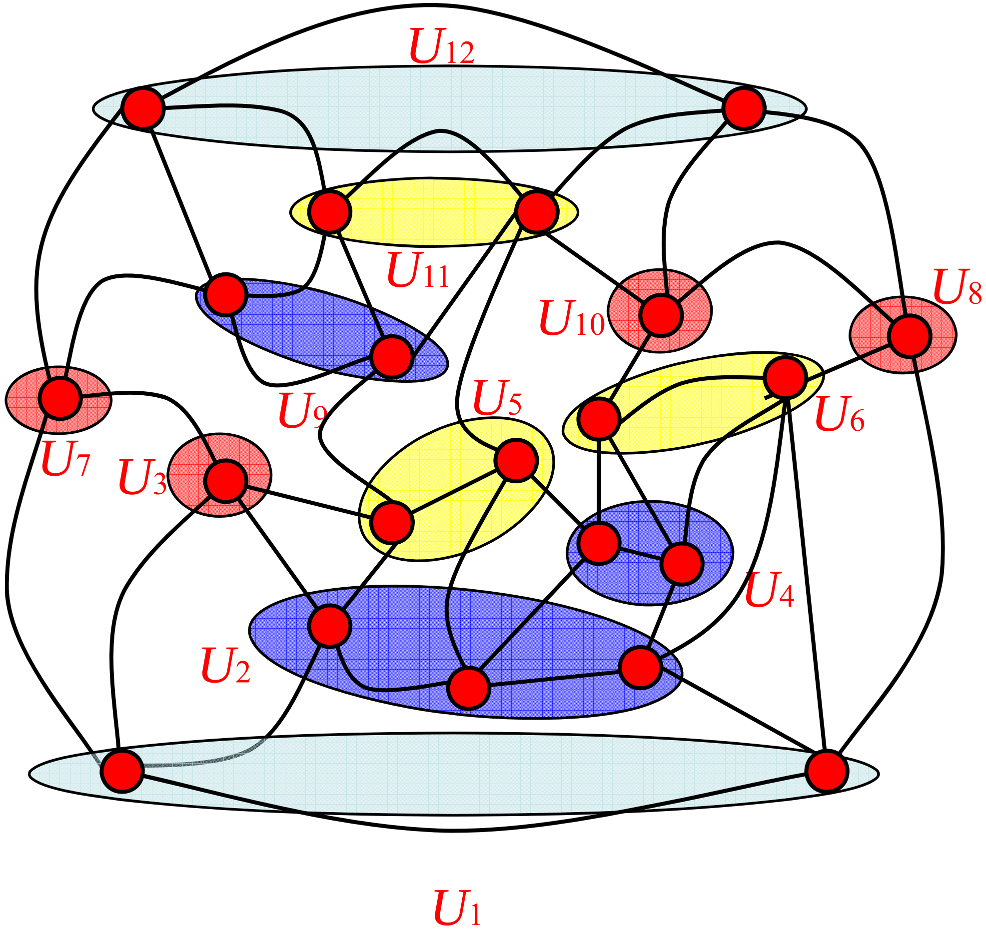
(a generalization of *st*-numbering)

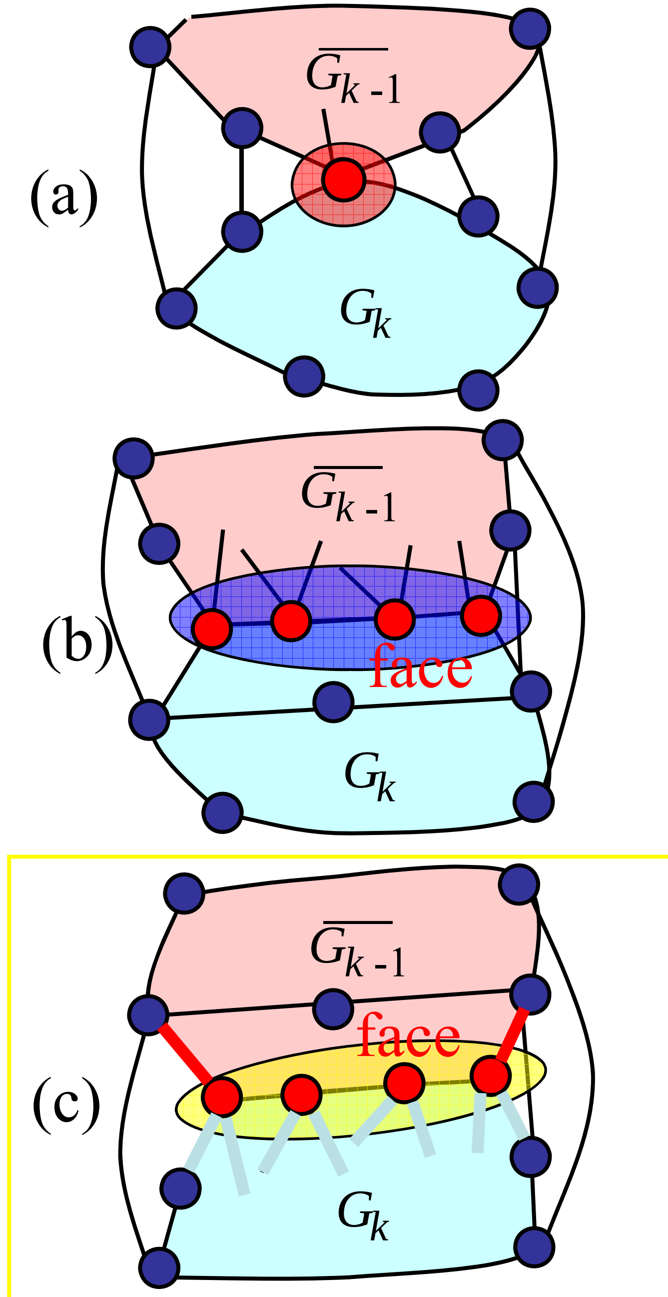




4-canonical decomposition[NRN97]

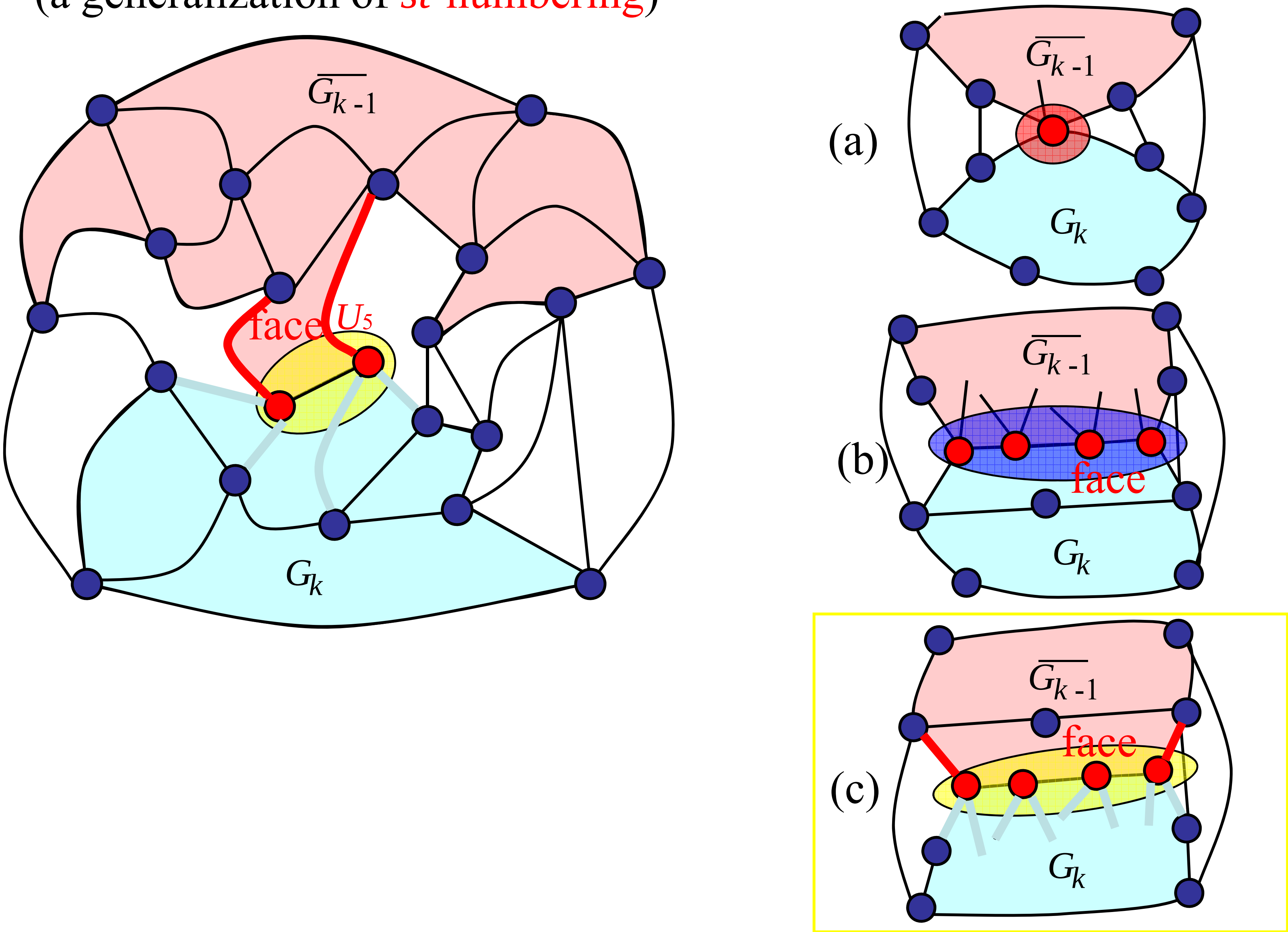
(a generalization of *st*-numbering)





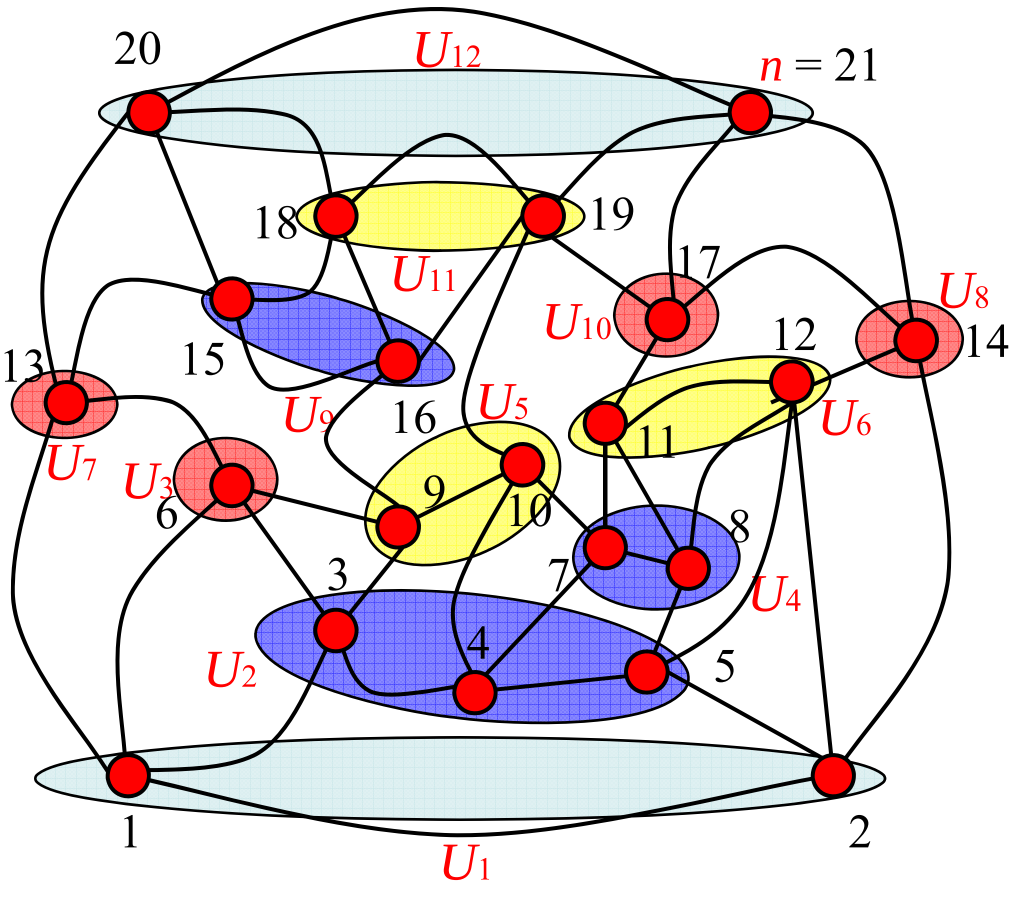
4-canonical decomposition[NRN97]

(a generalization of *st*-numbering)

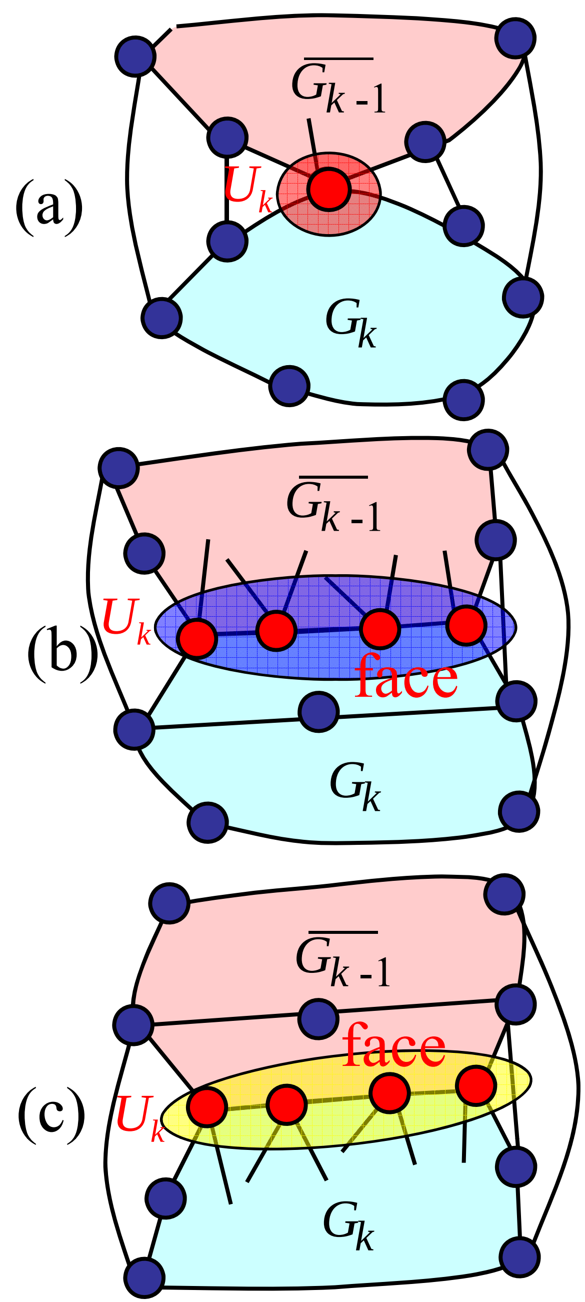


4-canonical decomposition[NRN97] $O(n)$

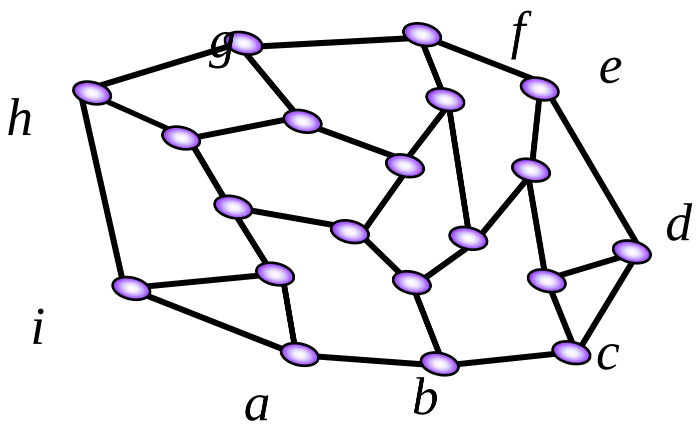
(a generalization of *st*-numbering)



G

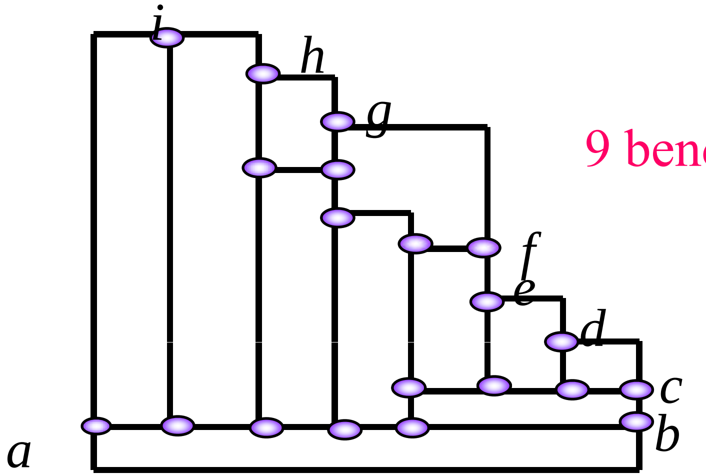


Orthogonal Drawings



plane graph *G*

Input



9 bends

Output

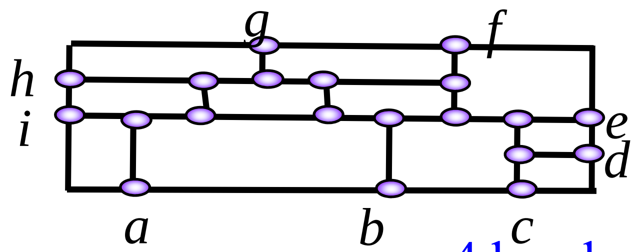
- Each edge is drawn as an alternating sequence of horizontal and vertical line segments.
- Each vertex is drawn as a point.

Applications

Circuit schematics, Data-flow diagrams, Entity-relationship diagrams [T87, BK97].

Objective

To minimize the number of bends in an orthogonal drawing.



4 bends

an orthogonal drawing with the minimum number of bends.

Known Result

Garg and Tamassia [GT96]

$O(n^{7/4} \log^{1/2} n)$ time algorithm for finding an orthogonal drawing of a plane graph of $\Delta \leq 4$ with the minimum number of bends.

Idea reduction to a minimum cost flow problem

Rahman, Nakano and Nishizeki [RNN99]

A linear time algorithm to find an orthogonal drawing of a 3-connected cubic plane graph with the minimum number of bends.

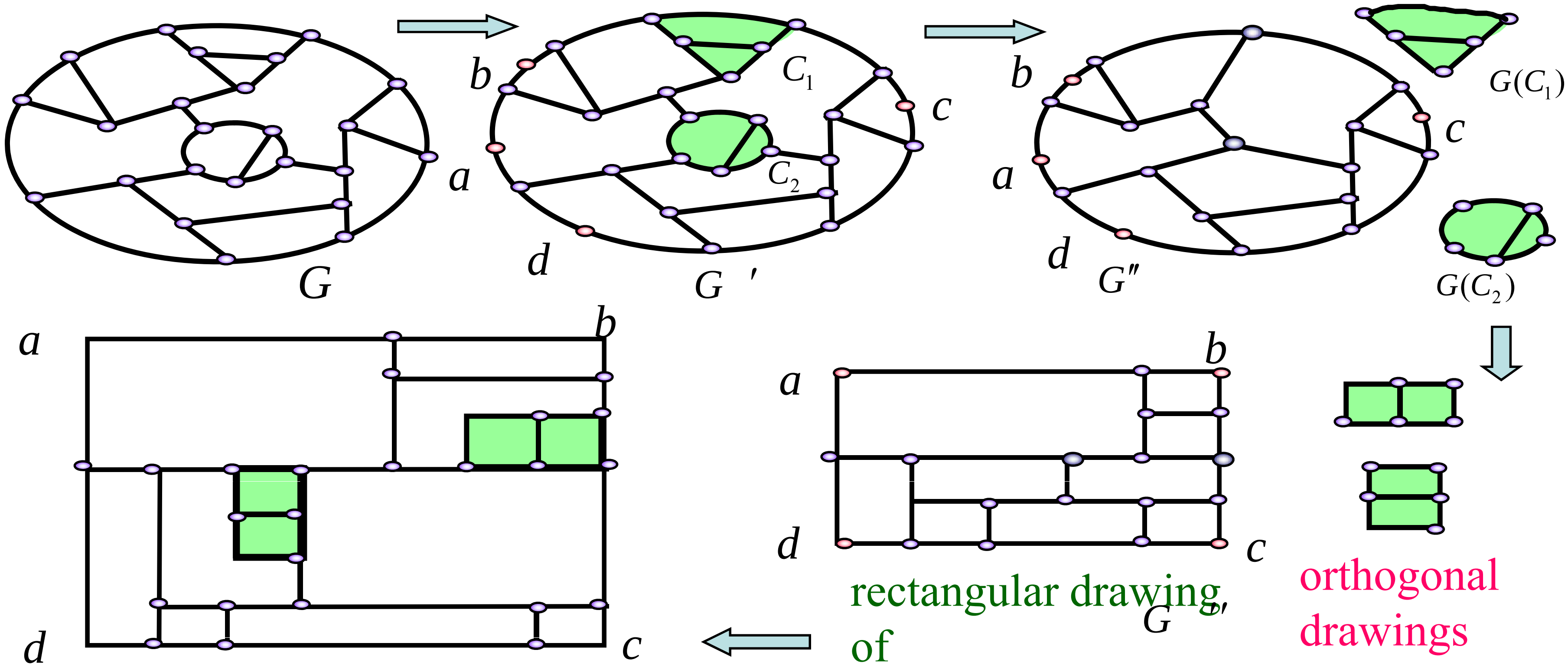
Idea reduction to a rectangular drawing problem.

Rahman and Nishizeki [RN02]

A linear time algorithm to find an orthogonal drawing of a plane graph of $\Delta \leq 3$ with the minimum number of bends.

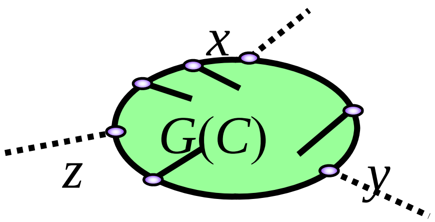
Idea reduction to a no-bend drawing problem.

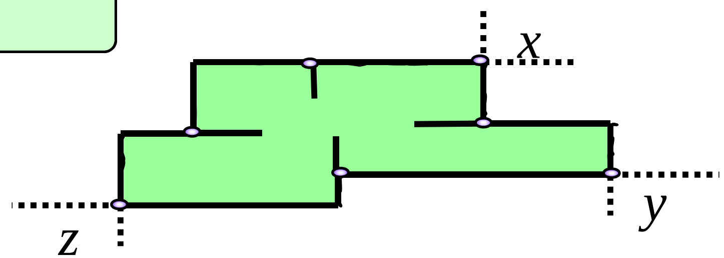
Outline of the algorithm of [RNN99]



orthogonal drawing of G

Properties of a drawing of $G(C)$

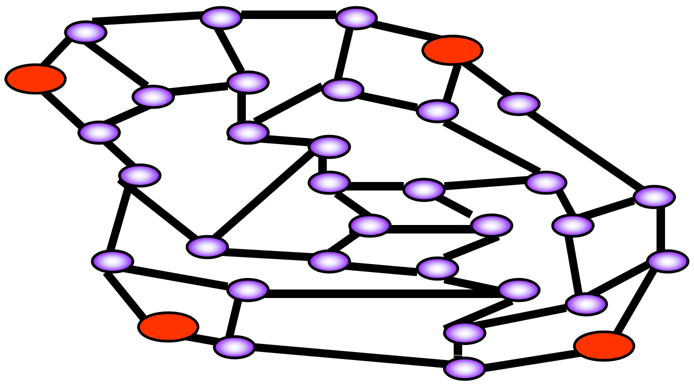




orthogonal drawing of $G(C)$

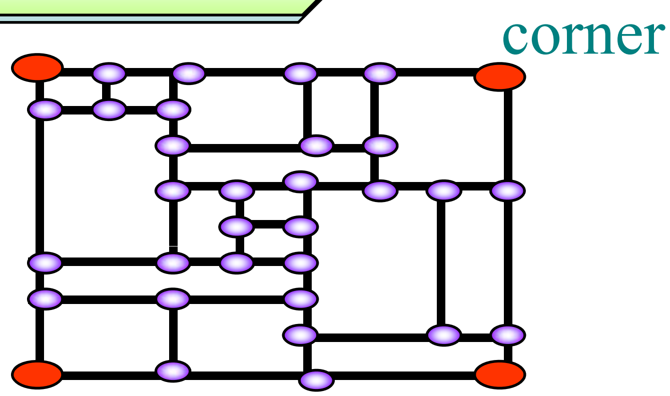
- minimum number of bends
- the six open halflines are free

Rectangular Drawings



Plane graph G of $\Delta \leq 3$

Input



Rectangular drawing of G

Output

- Each vertex is drawn as a point.
- Each edge is drawn as a horizontal or a vertical line segment.
- Each face is drawn as a rectangle.

Not every plane graph has a rectangular drawing.

Thomassen [84],

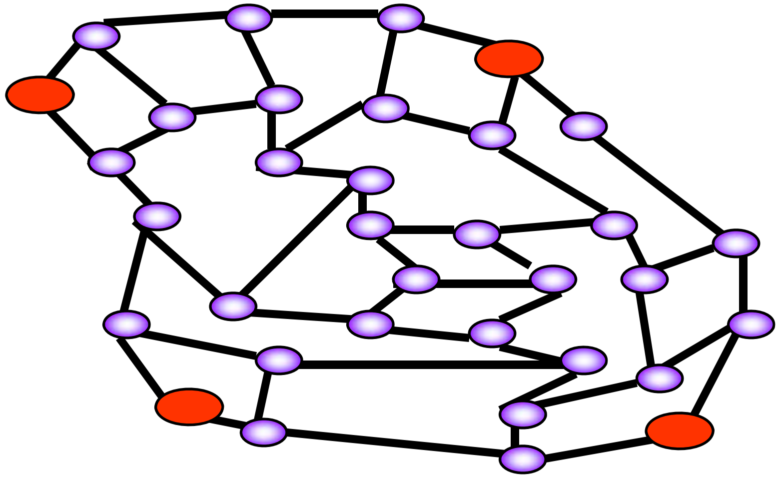
Rahman, Nakano and Nishizeki [02]

A necessary and sufficient condition.

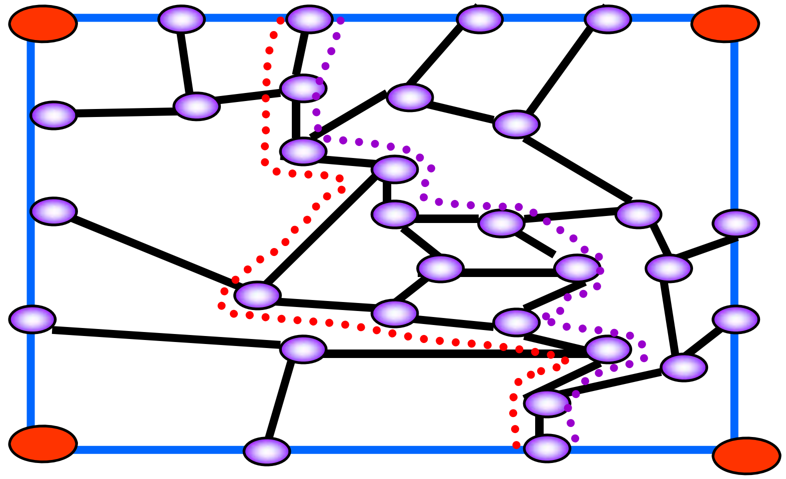
Rahman, Nakano and Nishizeki [98, 02]

A linear time algorithm

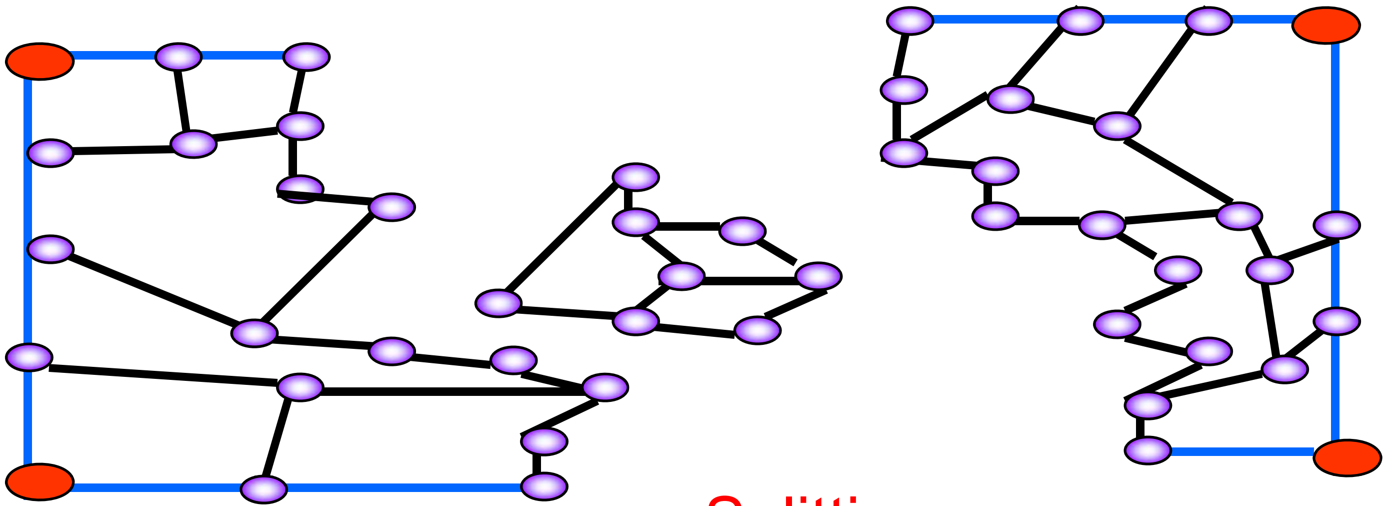
Rectangular Drawing Algorithm of [RNN98]



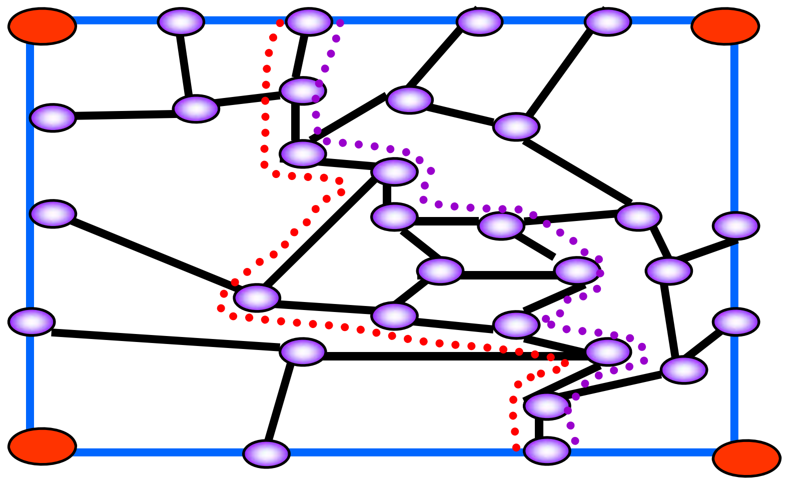
Input



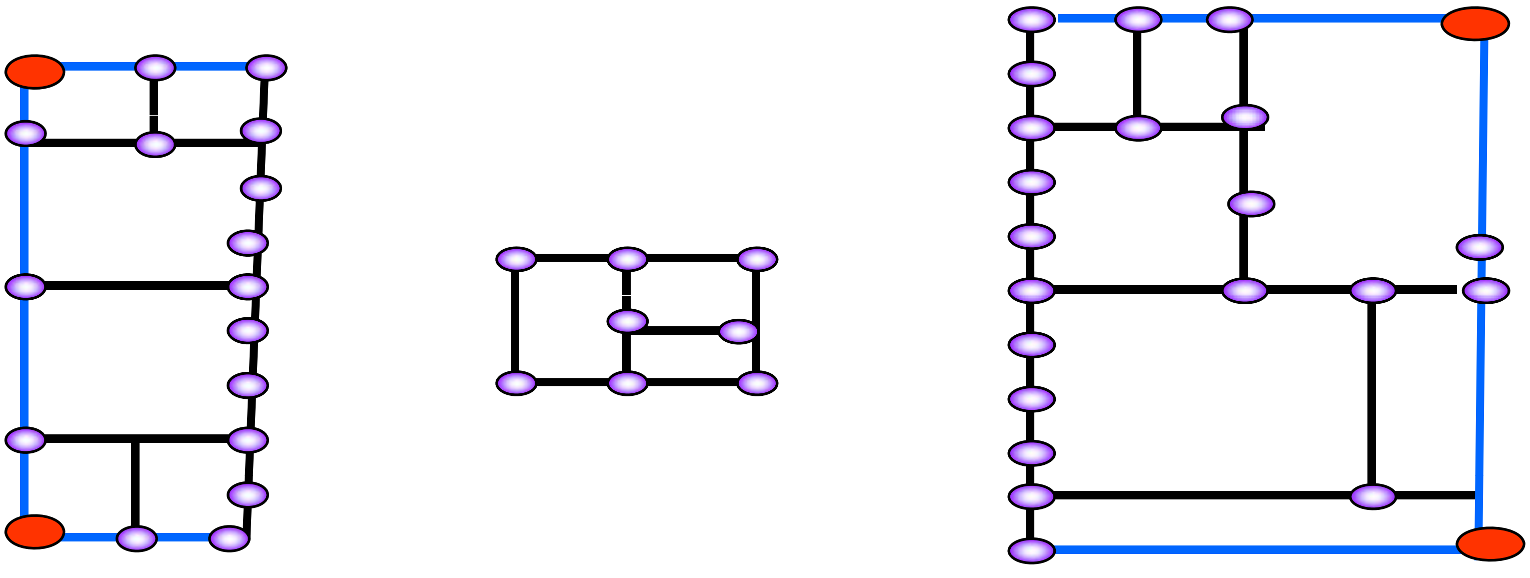
Partition-pair



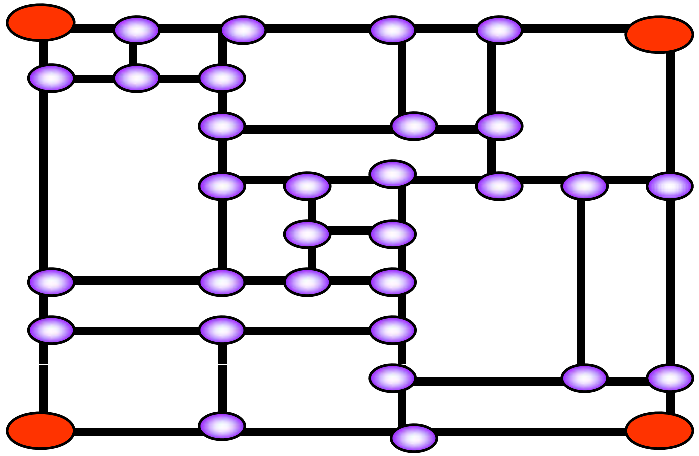
Splitting



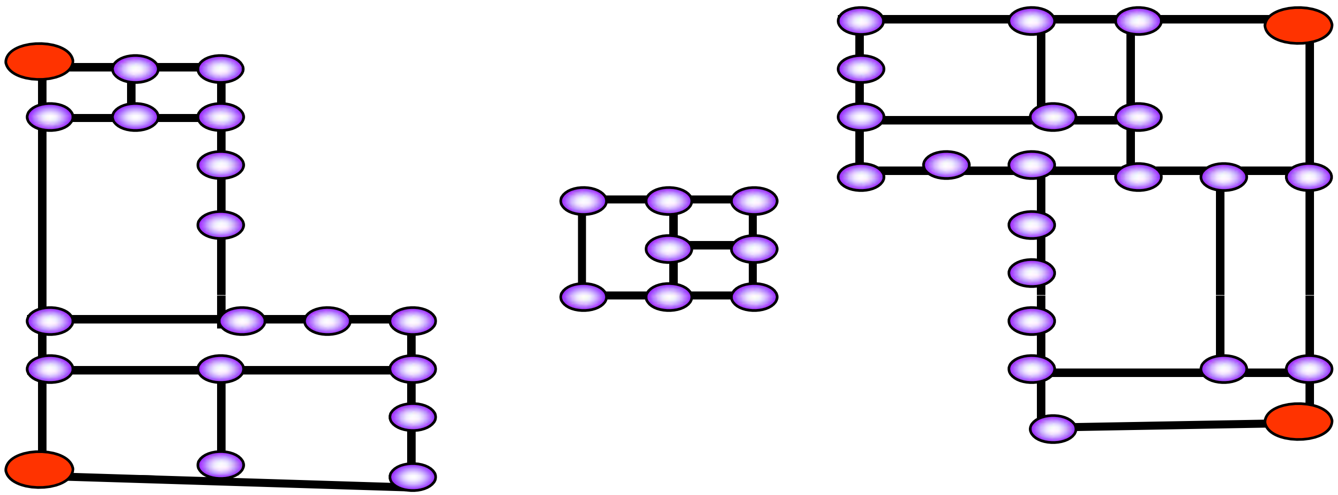
Partition-pair



rectangular drawing of each subgraph

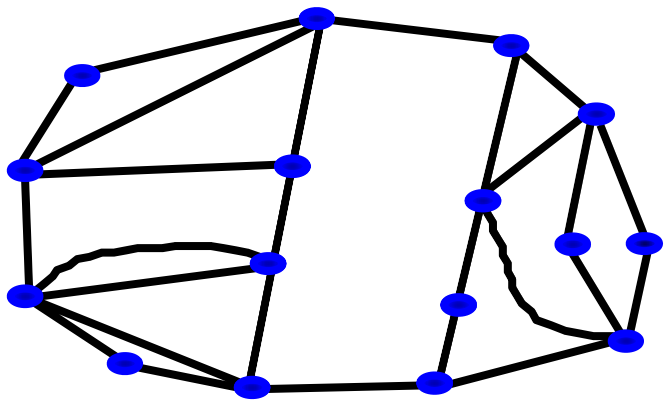


Rectangular Drawings



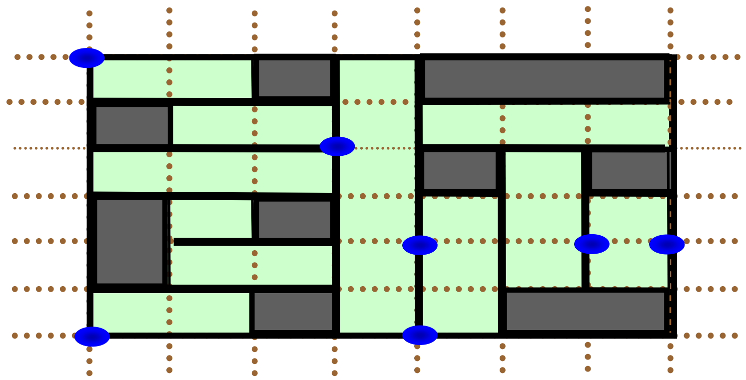
Modification of drawings
patching

Box-Rectangular Drawing



Plane multigraph

Input



Output

- Each vertex is drawn as a (possibly degenerated) rectangle on an integer grid.
- Each edge is drawn as a horizontal or a vertical line segment along grid line without bend.
- Each face is drawn as a rectangle.

Rahman, Nakano and Nishizeki [RNN00]

- A necessary and sufficient condition.
- A linear-time algorithm.

Reduce the problem to a rectangular drawing problem.

Conclusions

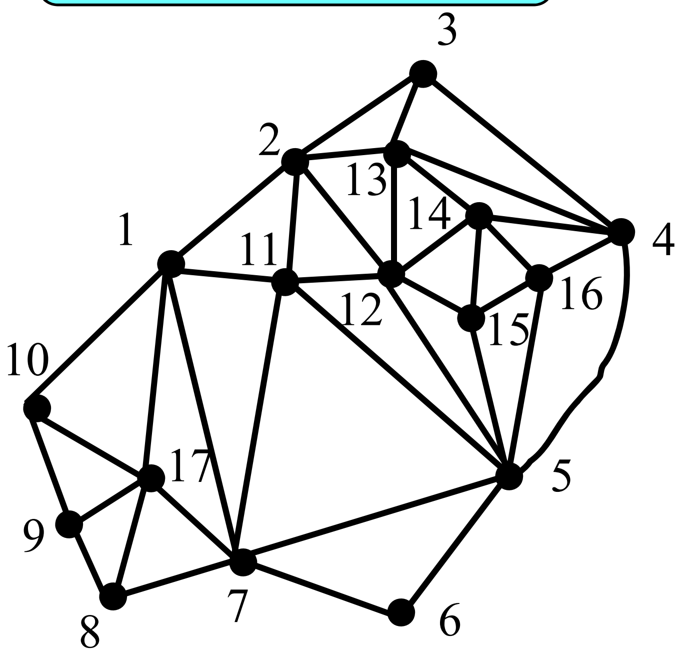
We surveyed the recent algorithmic results on various drawings of plane graphs.

Open Problems

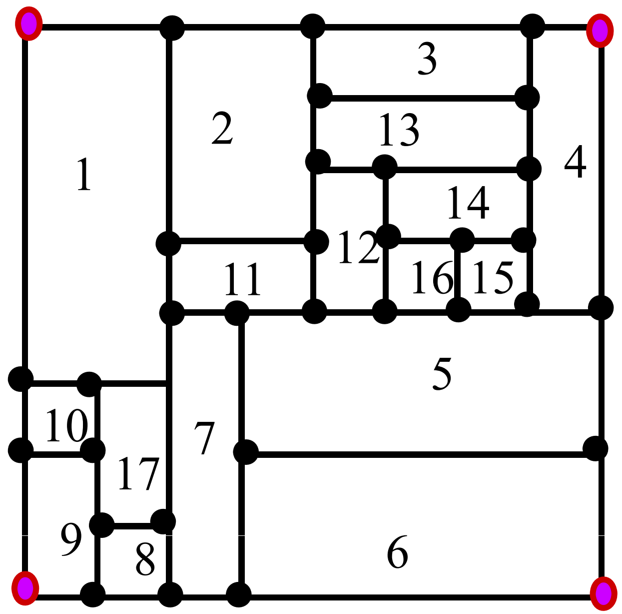
- Rectangular drawings of plane graphs of $\Delta \leq 4$.
- Efficient algorithm for bend-minimal orthogonal drawings of plane graphs of $\Delta \leq 4$
- Parallel algorithms for drawing of plane graphs.

Application

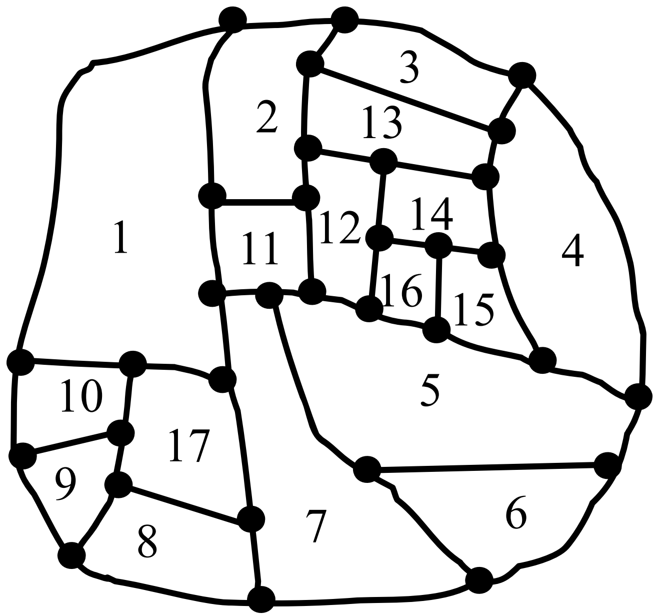
VLSI floorplanning



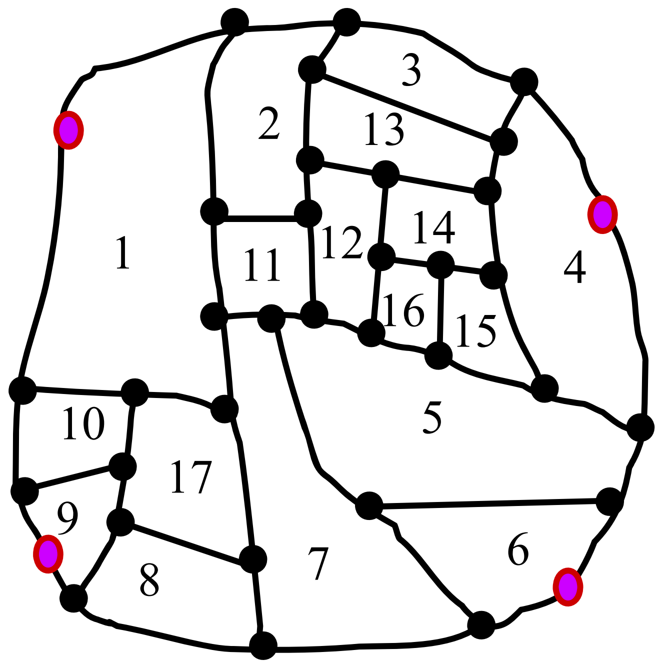
Interconnection Graph



Floor plan



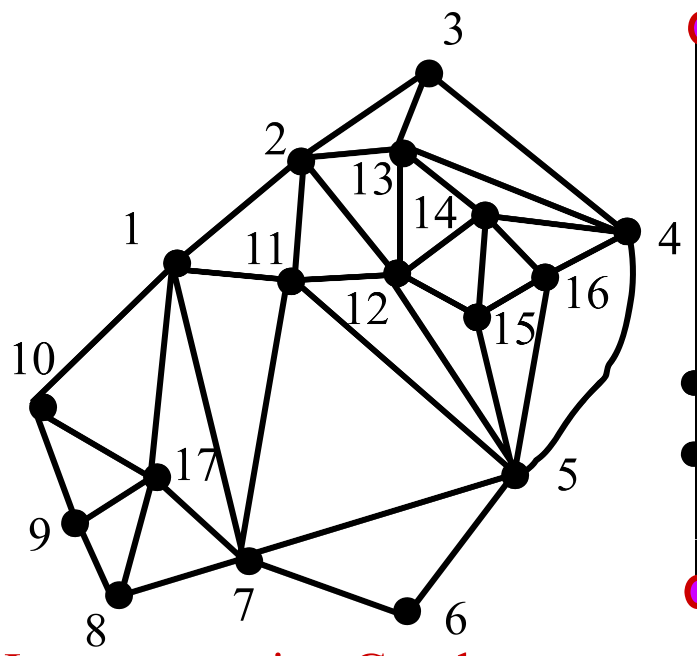
Dual-like graph



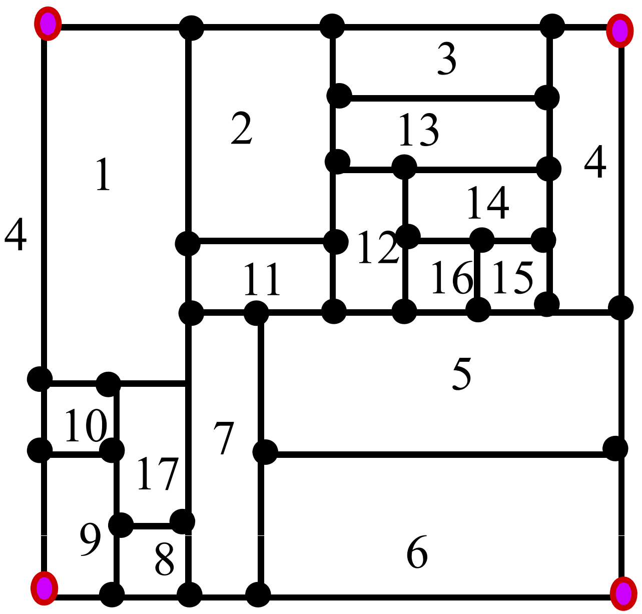
Insertion of four corners

Application

VLSI floorplanning



Interconnection Graph

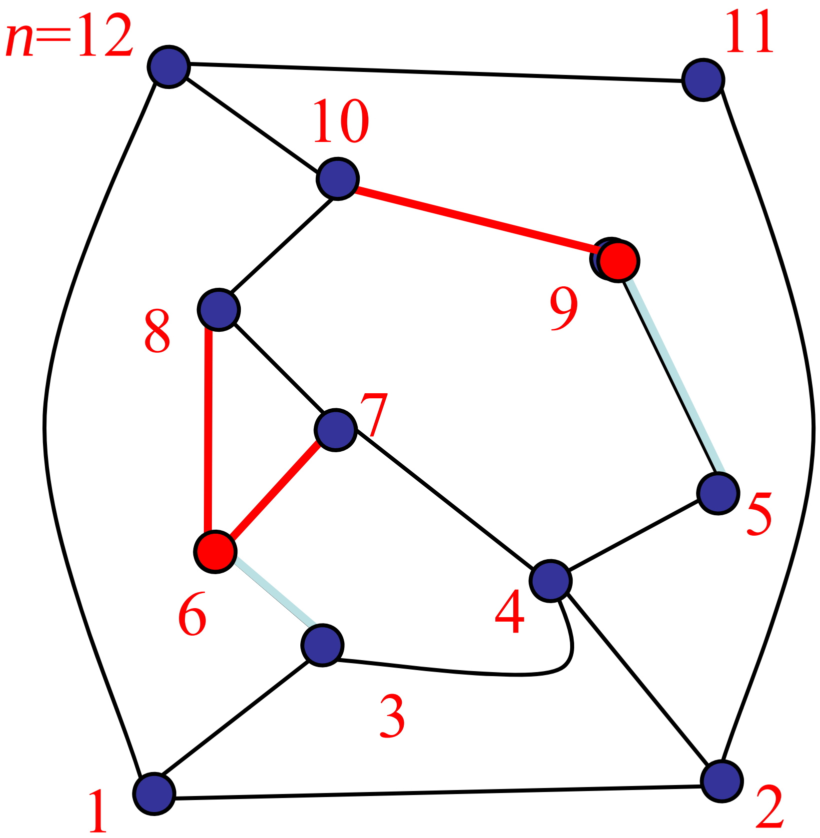


Floor plan

st-numbering[E79](2-connected graph)

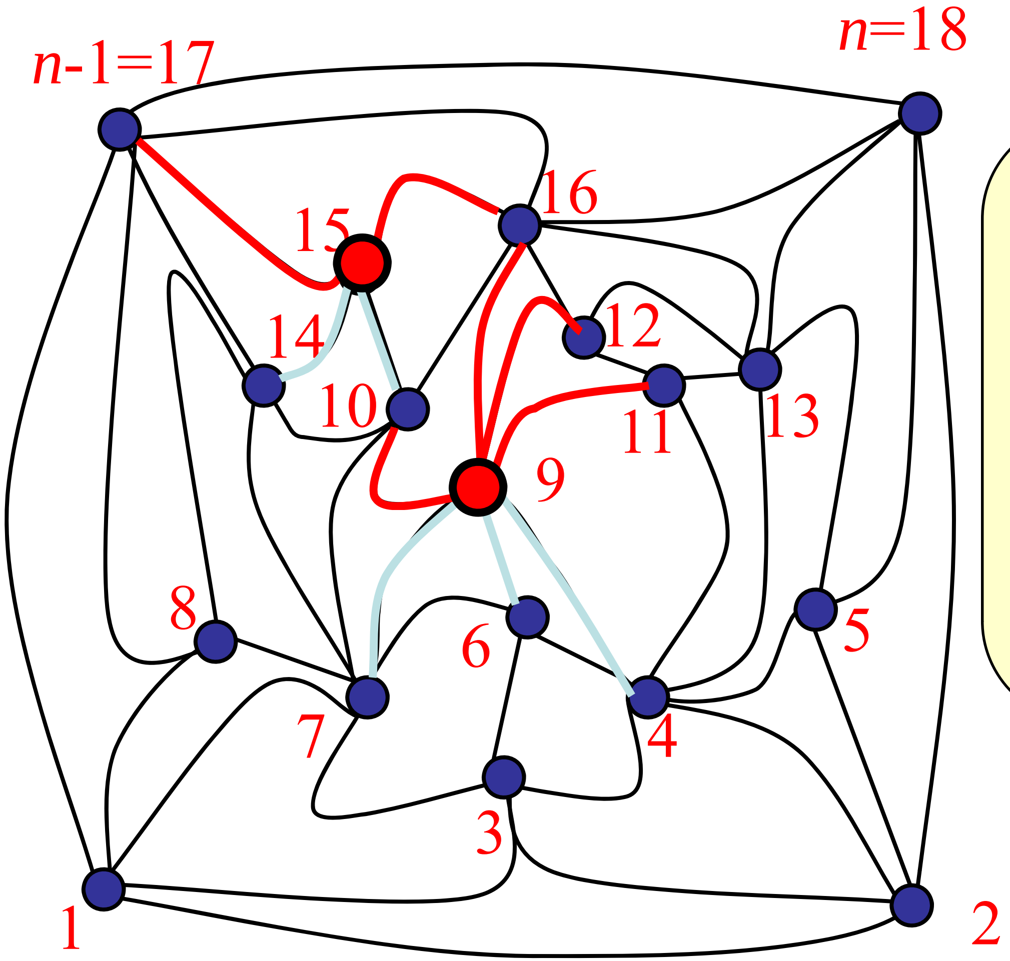
(1) $(1,n)$ is an edge of G

(2) Each vertex k, $2 \leqq k \leqq n$-1,
has at least one neighbor and at least one upper neighbor



Step 1: 4-canonical ordering [KH97]
(4-connected graph)

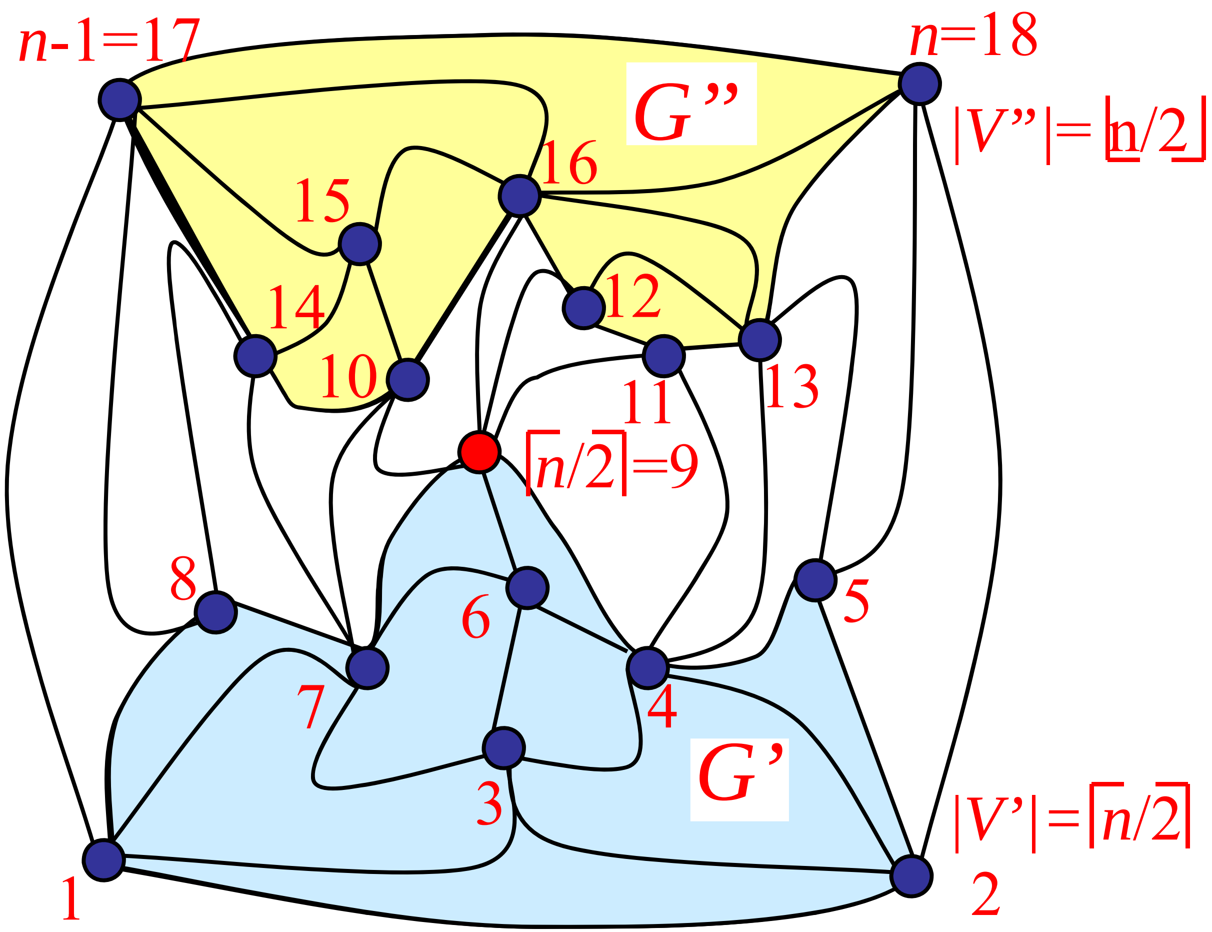
(a generalization of *st*-numbering)



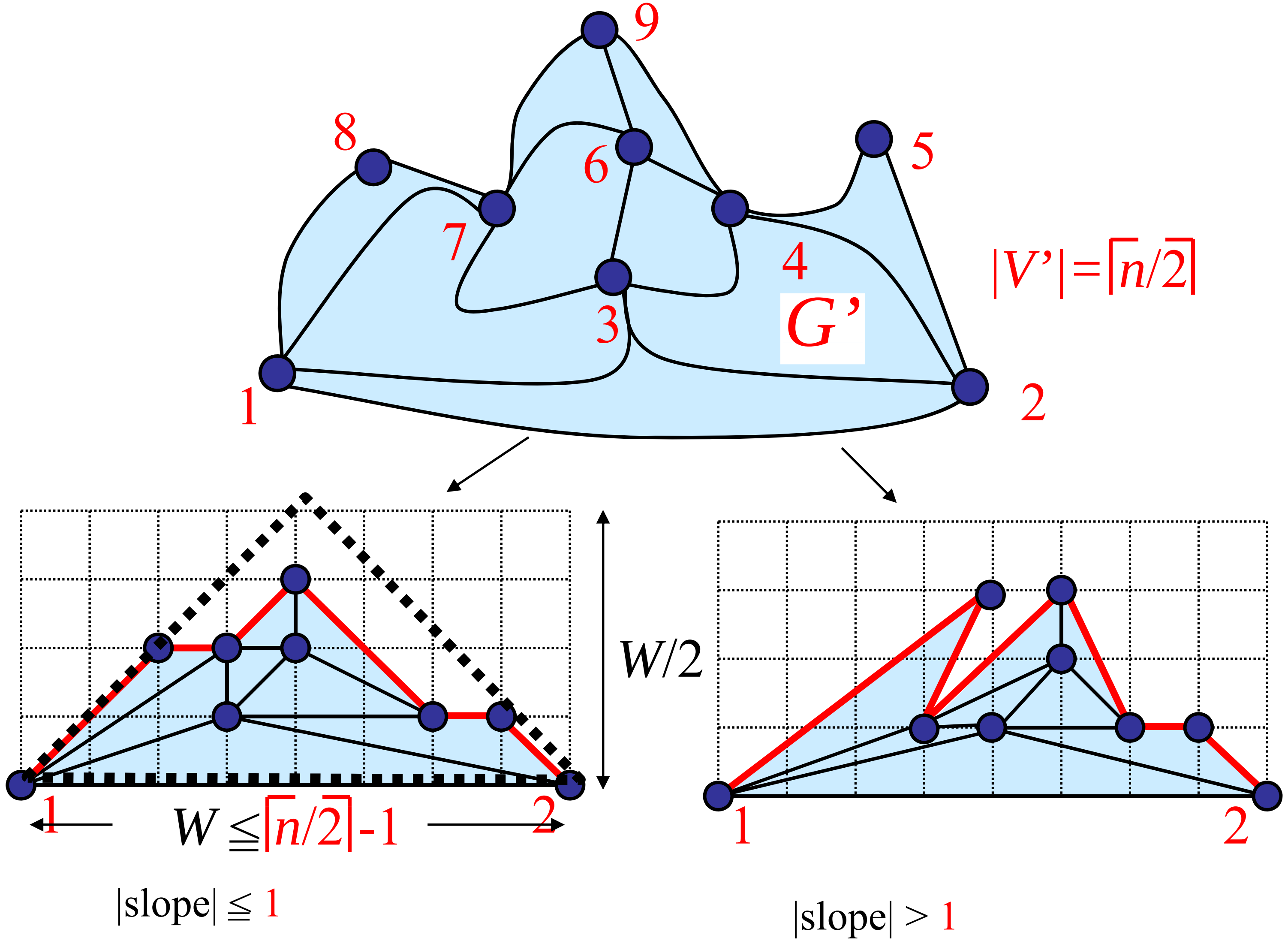
(1) (1,2) and (*n*,*n*-1) are edges on the outer face

(2) Each vertex k, $3 \leqq k \leqq n$-2, has at least two lower neighbors and has at least two upper neighbors

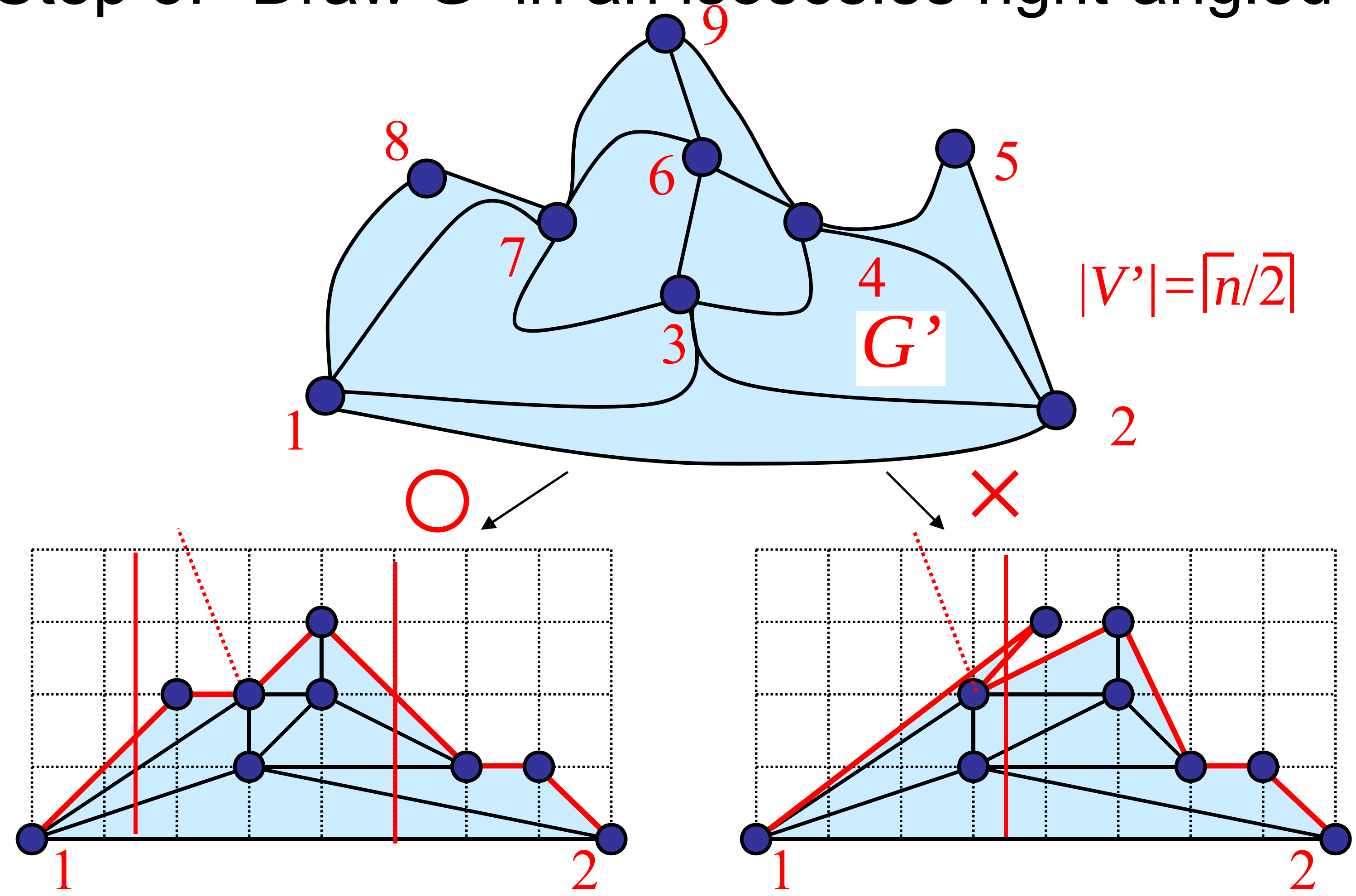
Step 2: Divide *G* into *G'* and *G''*



Step 3: Draw G' in an isosceles right-angled triangle



Step 3: Draw *G'* in an isosceles right-angled triangle



|外周上の辺の傾き| ≦ 1
x-単調

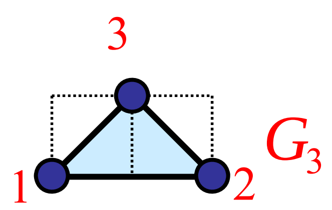
|外周上の辺の傾き| > 1
x-単調ではない

1 : 外周上の辺の傾きは高々1

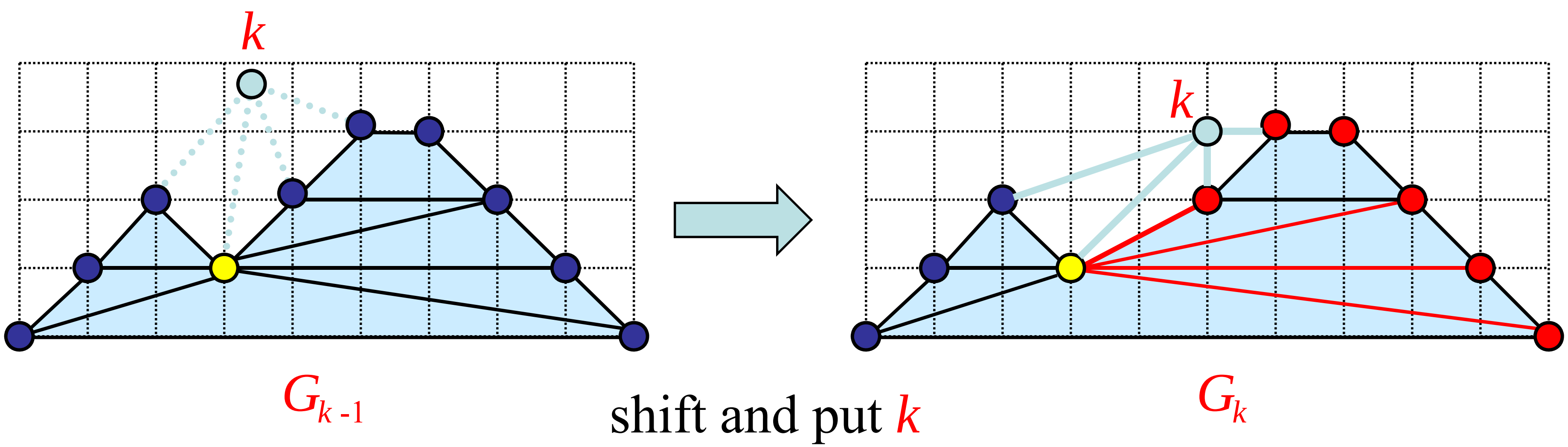
2 : 外周上で点1から2へ時計回りに進む道の描画はx-単調

各点の座標の決め方

Initialization:

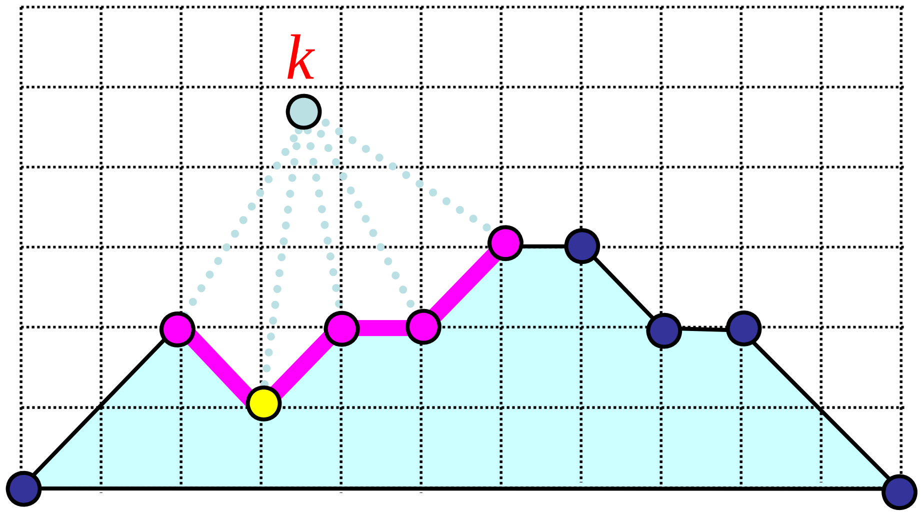


各点 $k,\ 4 \leqq \mathrm{k} \leqq \lceil n/2 \rceil$

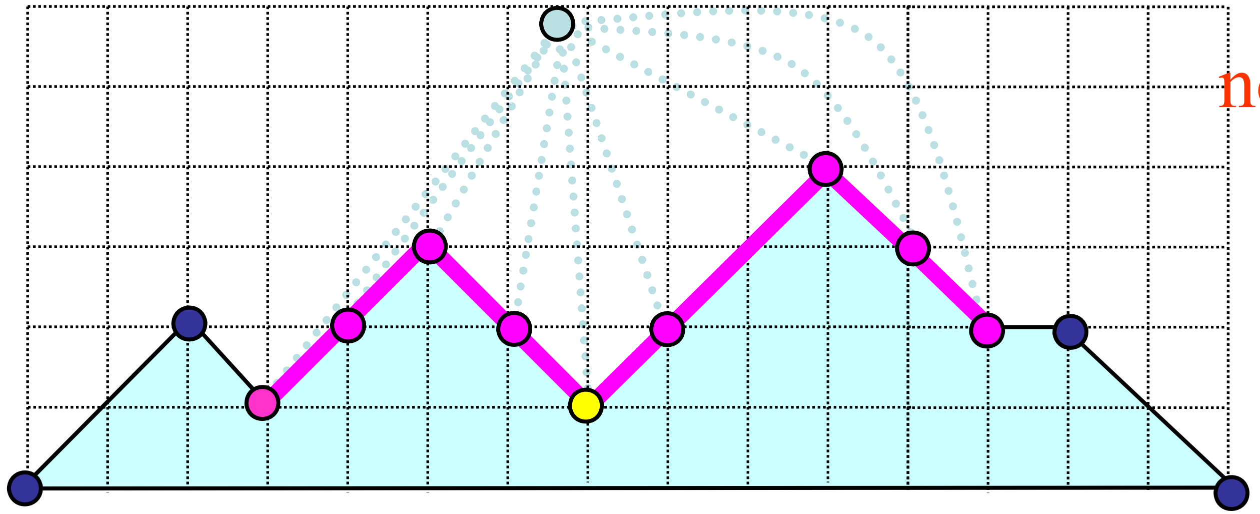


1: 外周上の辺の傾きは高々1
2: 外周上で点1から2へ時計回りに進む道の描画はx-単調

The drawing of the path passing through all *k*'s neighbors is "weakly convex"



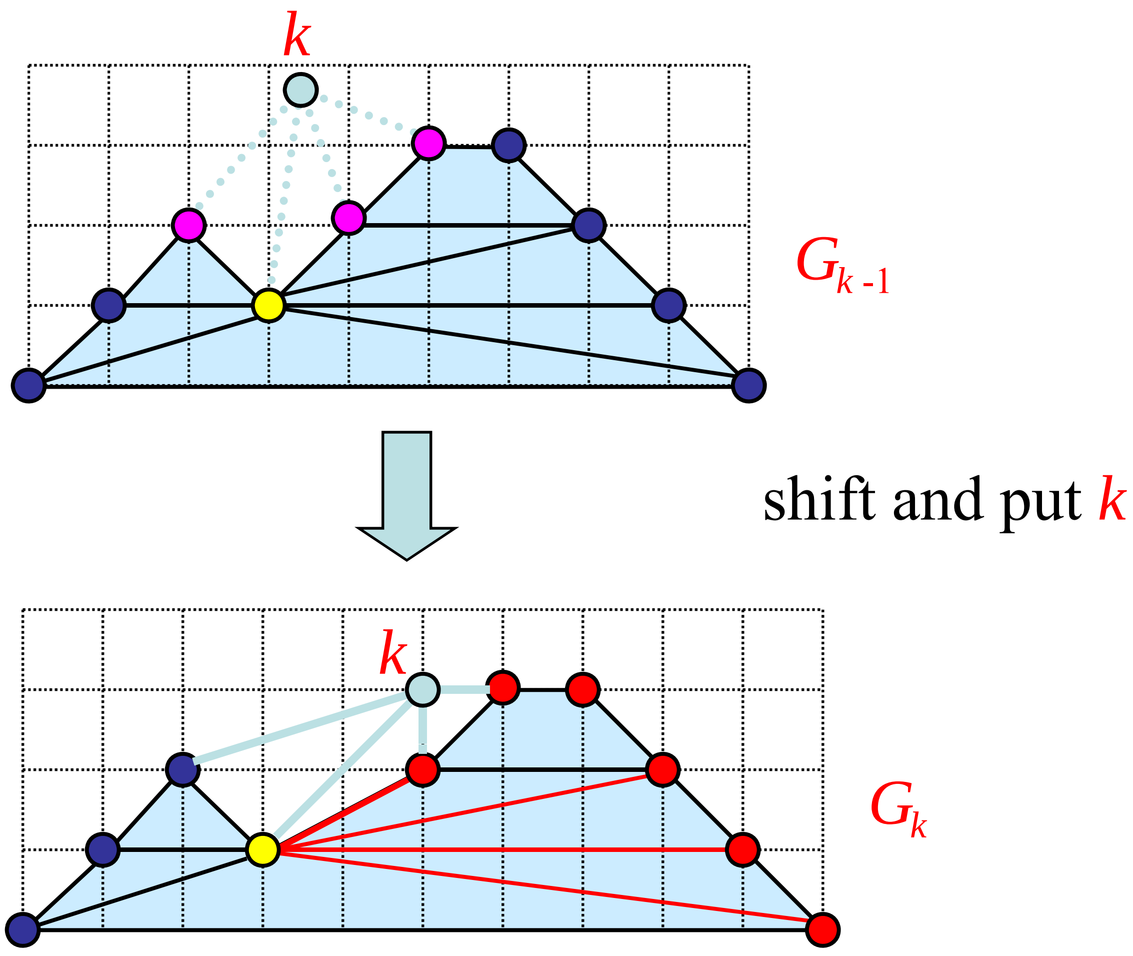
"weakly convex" ○



not "weakly convex" ×

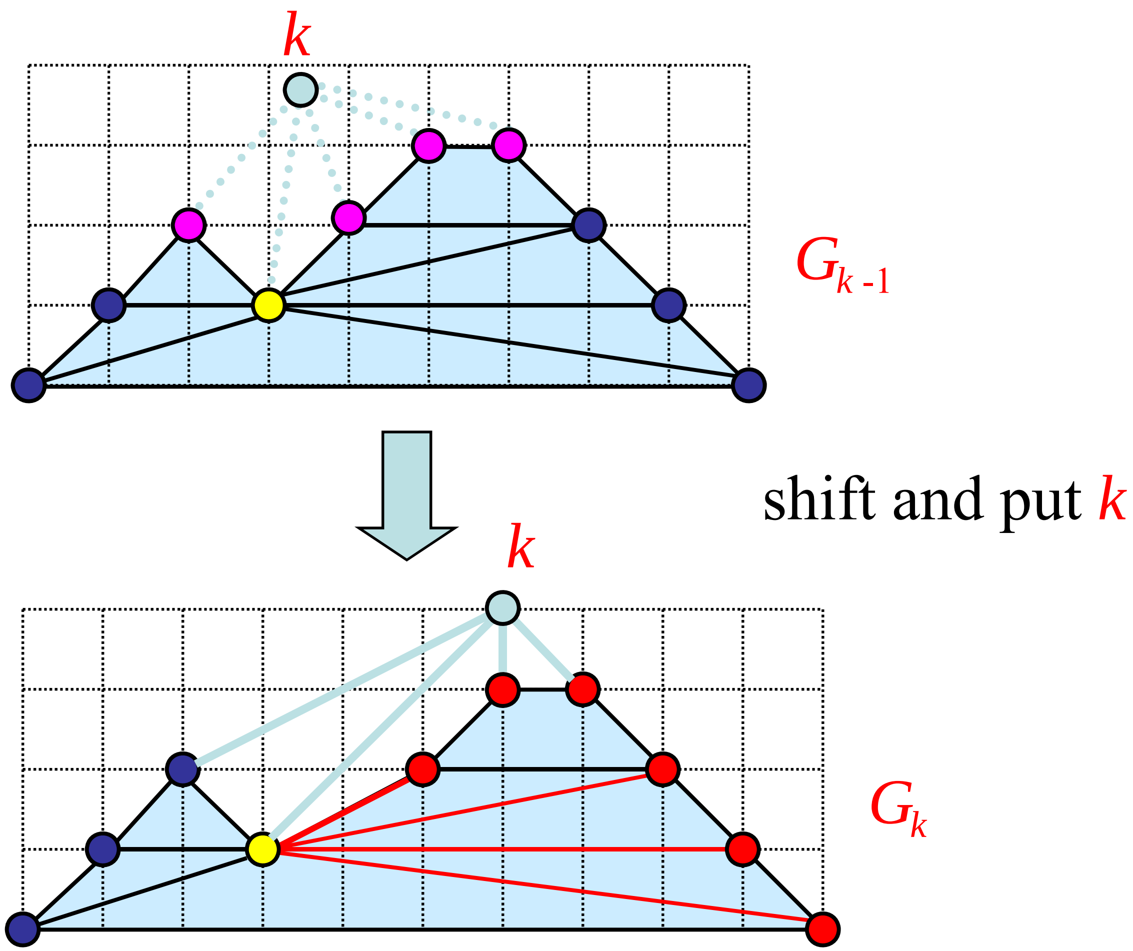
The rightmost neighbor of k is higher than the leftmost neighbor

Case 1 : k has exactly one highest neighbor

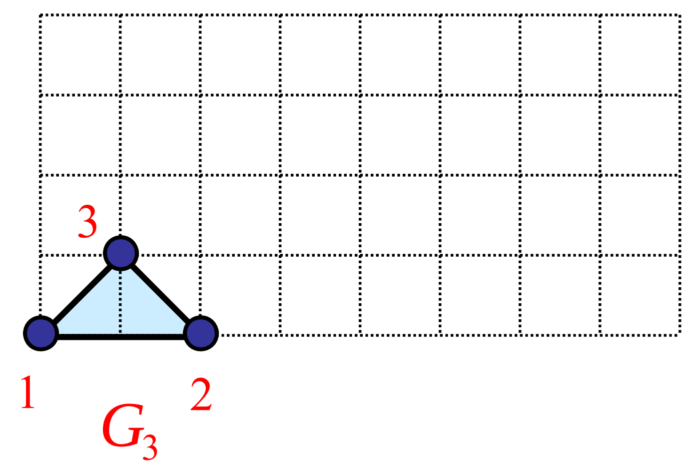


The rightmost neighbor of *k* is higher than the leftmost neighbor

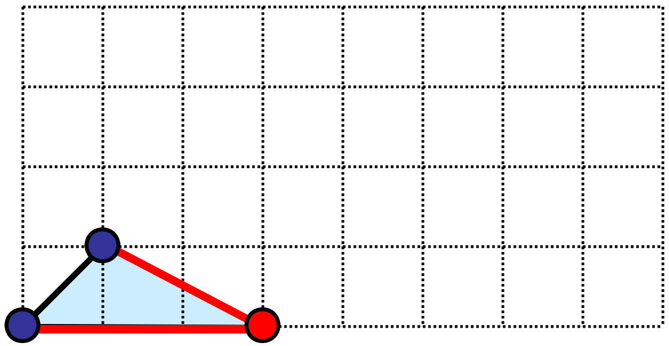
Case 2 : *k* has exactly two or more highest neighbor



Drawing of *G'*

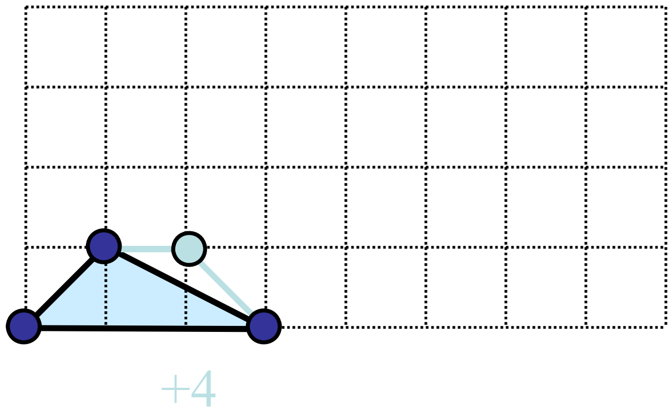


Drawing of *G'*

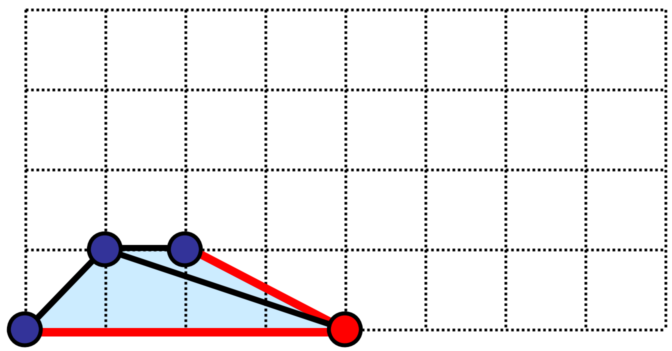


shift

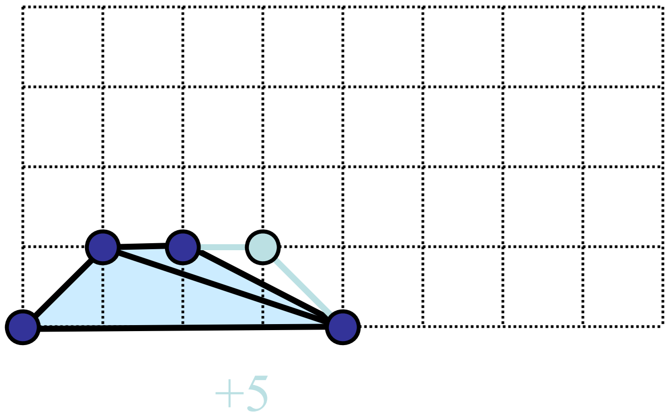
Drawing of *G'*



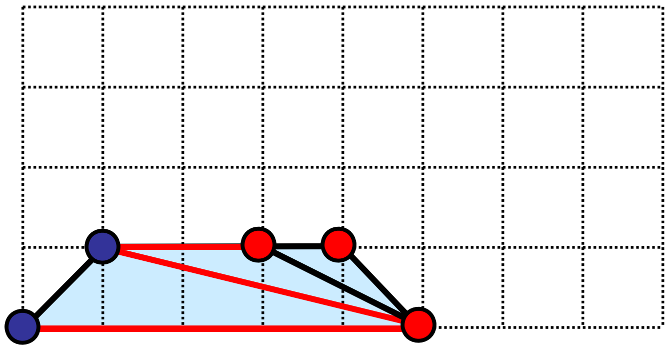
Drawing of *G'*



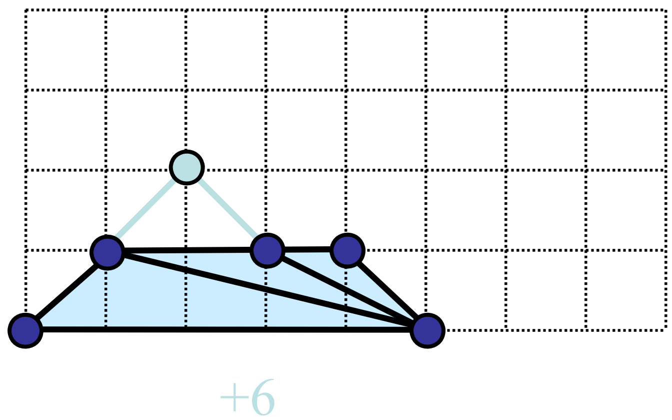
Drawing of *G'*



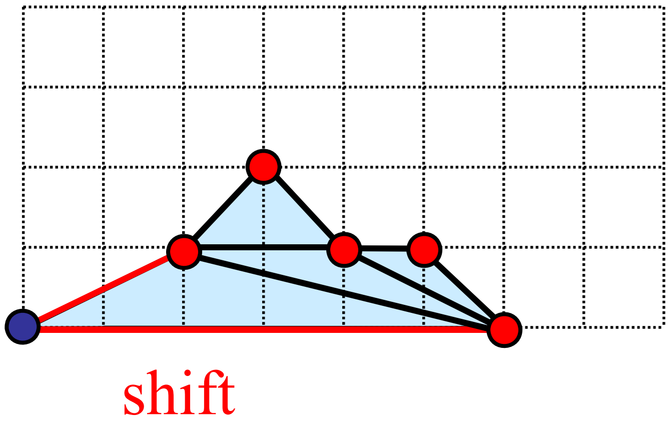
Drawing of *G'*



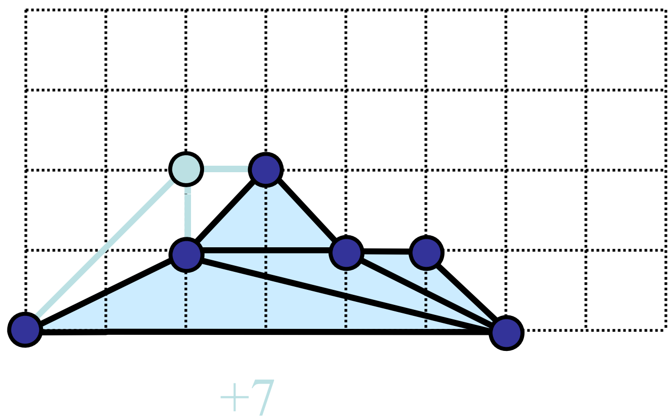
Drawing of *G’*



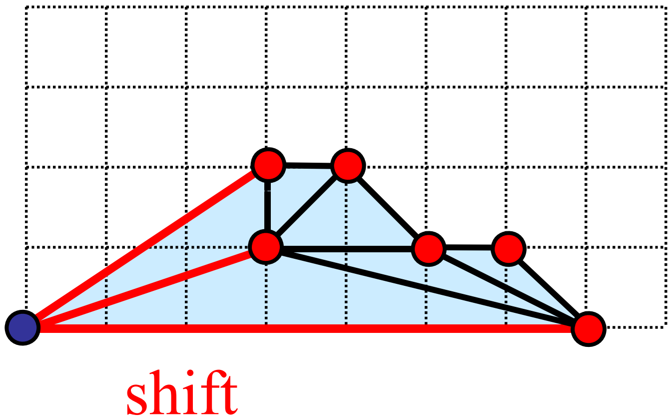
Drawing of *G'*



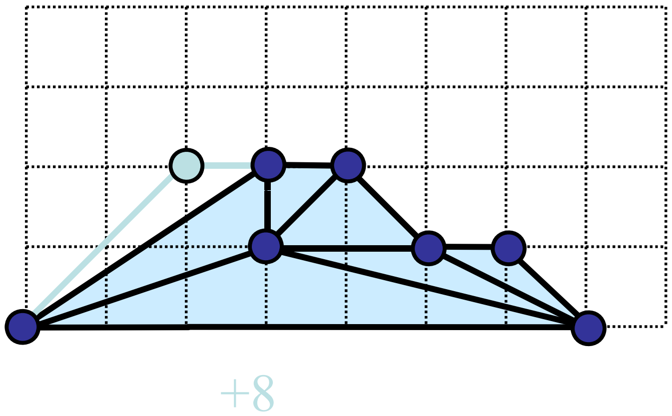
Drawing of *G'*



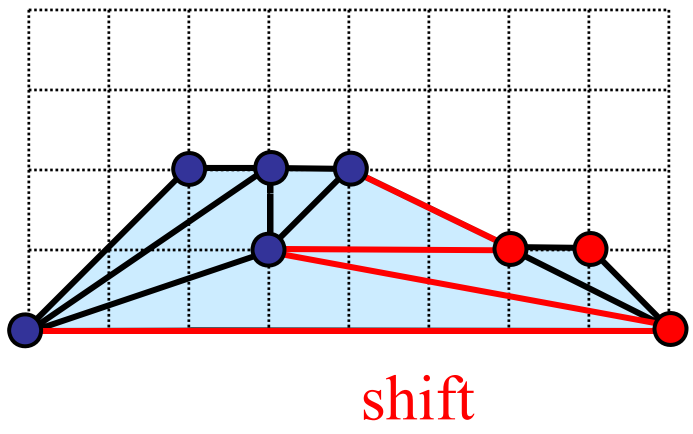
Drawing of *G'*



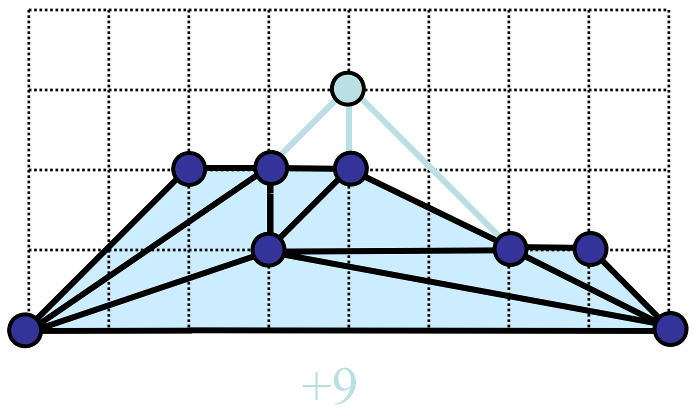
Drawing of *G'*



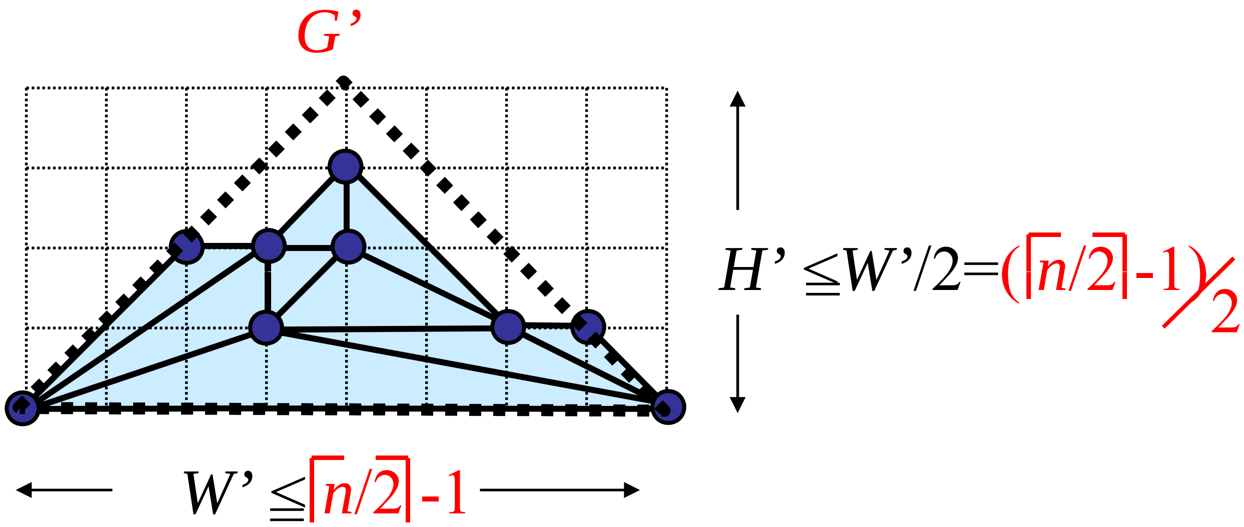
Drawing of *G'*



Drawing of *G'*

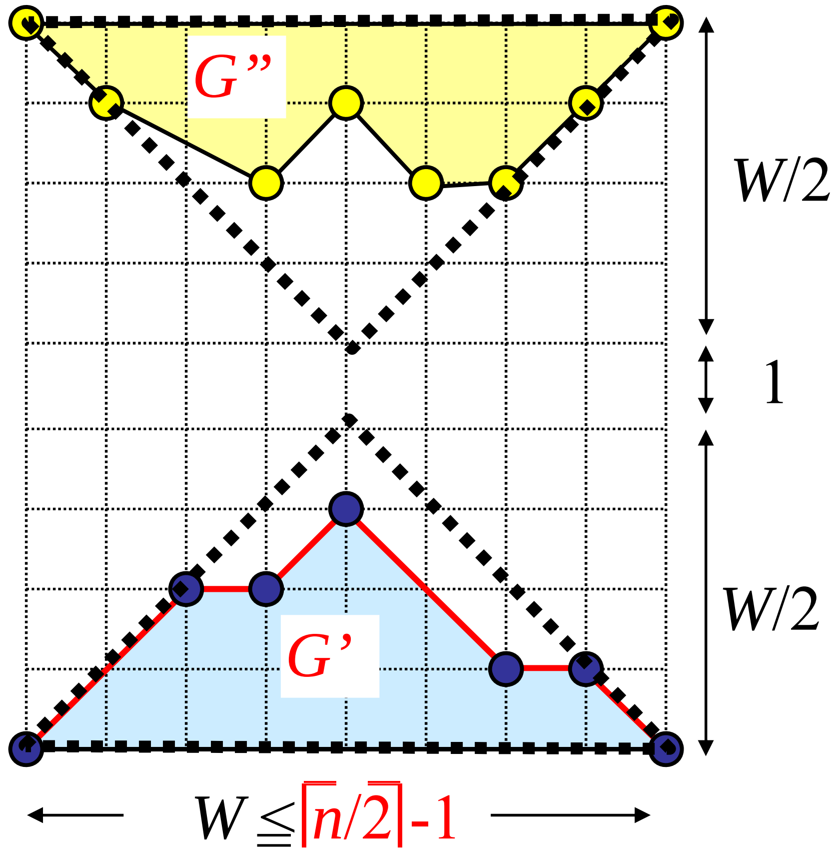


Drawing of *G'*

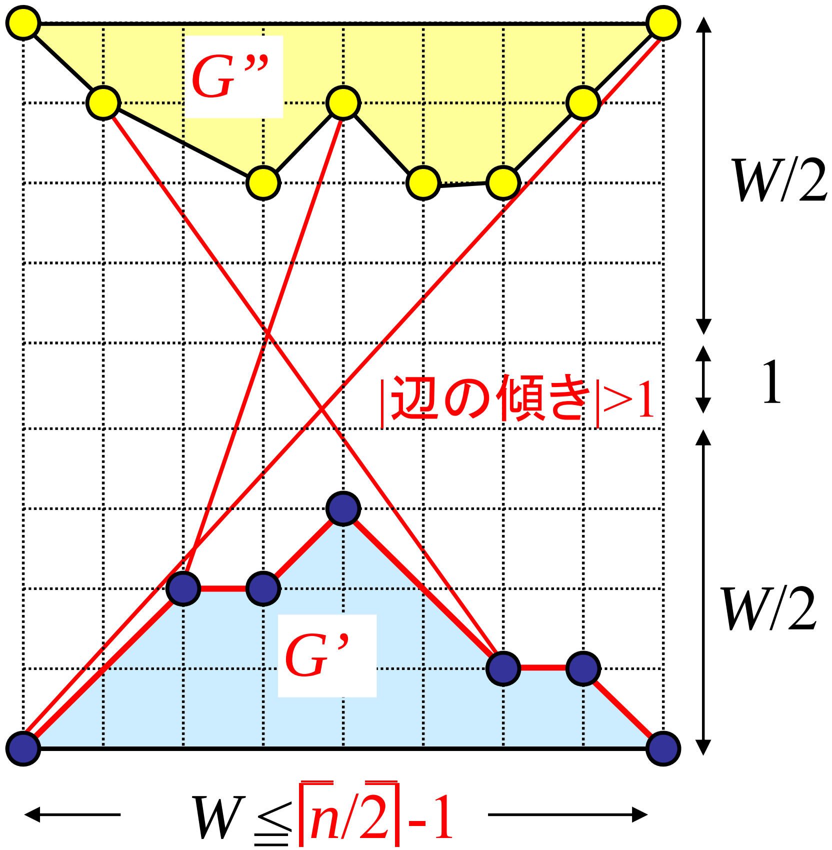


Shift one time for each vertex

Step 5: Combine the drawing of *G'* and *G"*

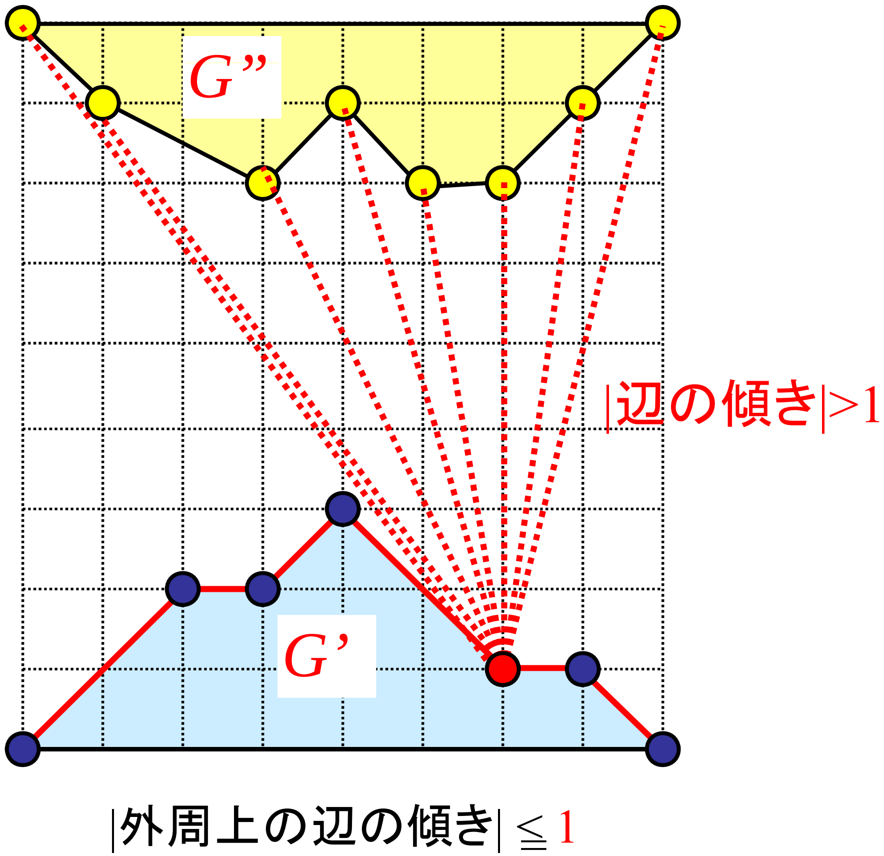


Step 5: Combine the drawing of *G'* and *G"*

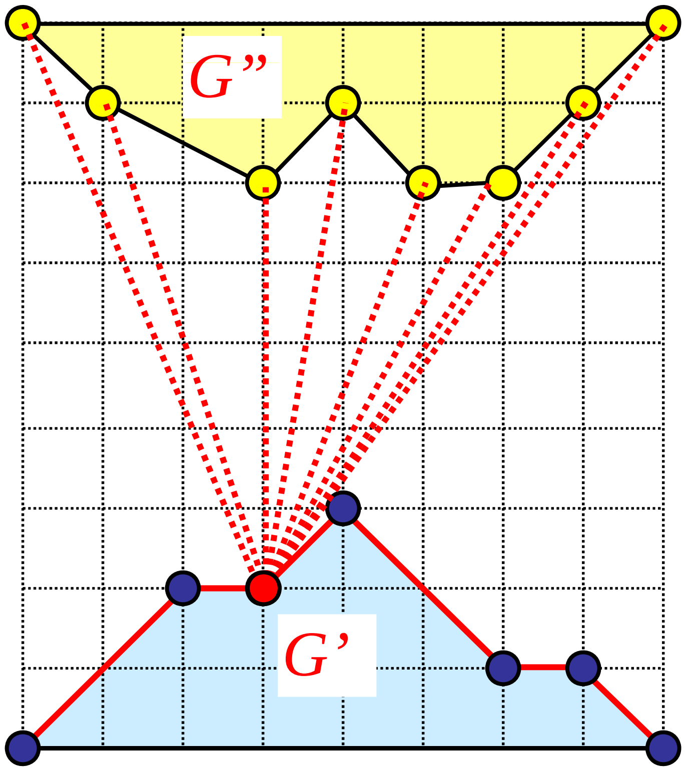


|外周上の辺の傾き| ≦ 1

Step 5: Combine the drawing of *G'* and *G"*

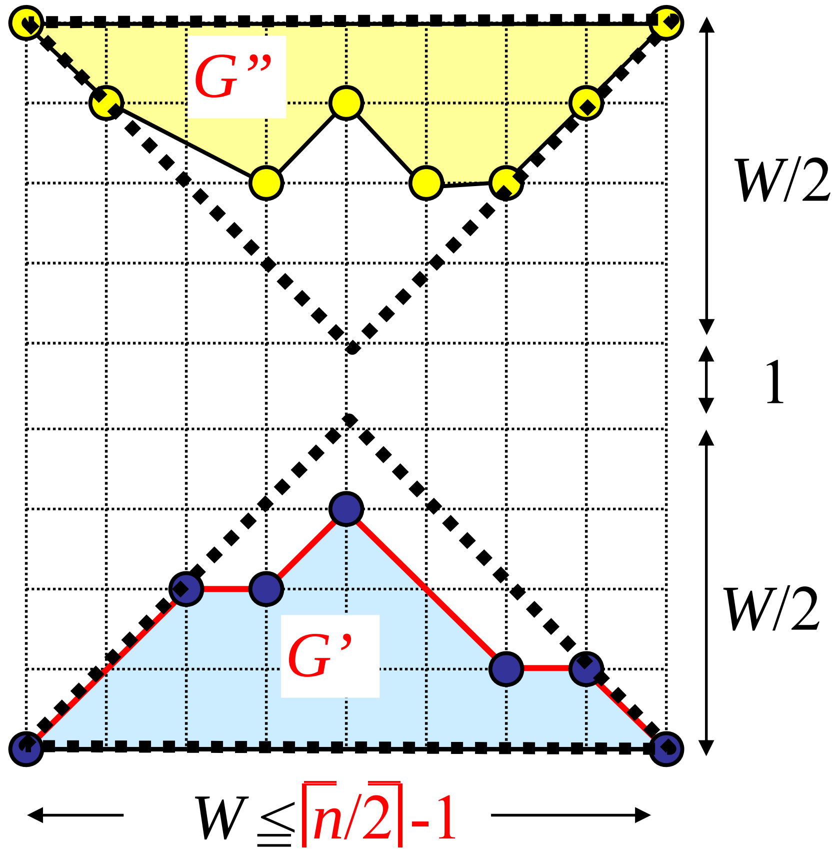


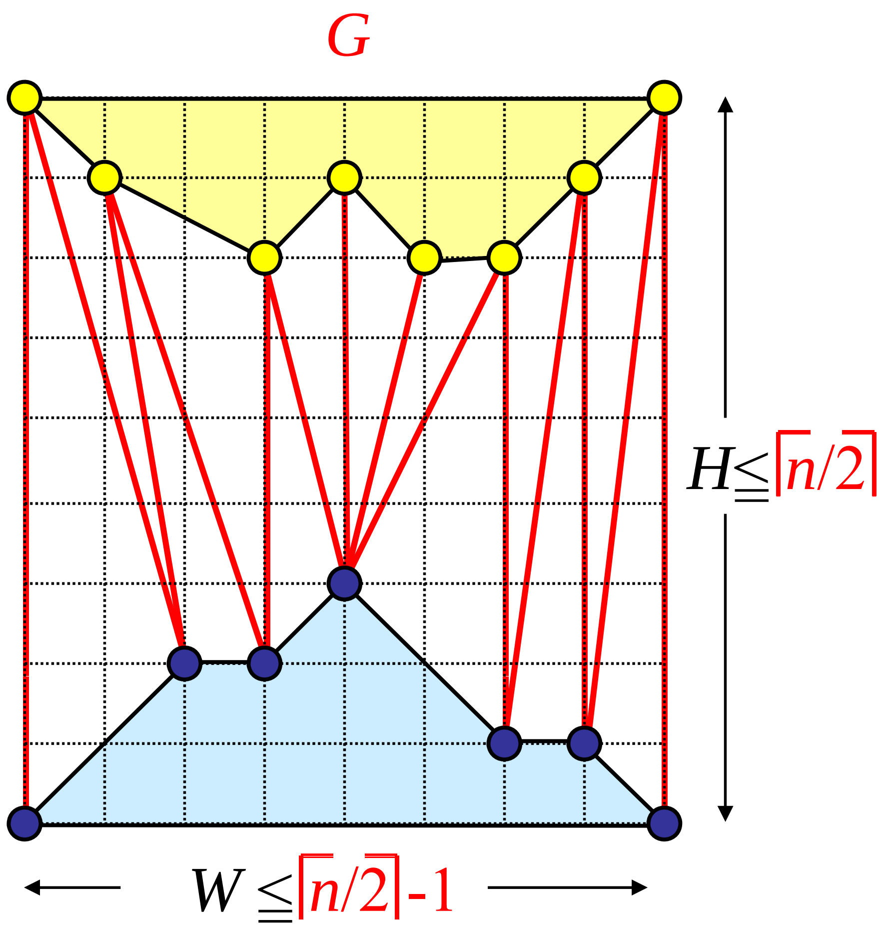
Step 5: Combine the drawing of *G'* and *G"*



x-単調

Step 5: Combine the drawing of *G’* and *G”*





Miura *et al.* '01

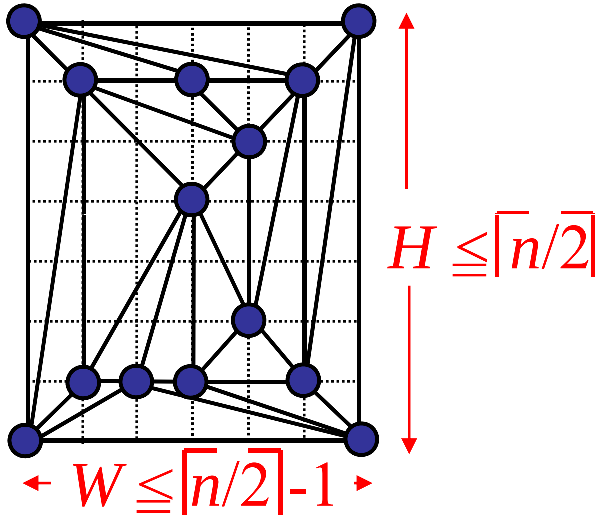
Input: 4-connected plane graph G

Output: a straight line grid drawing

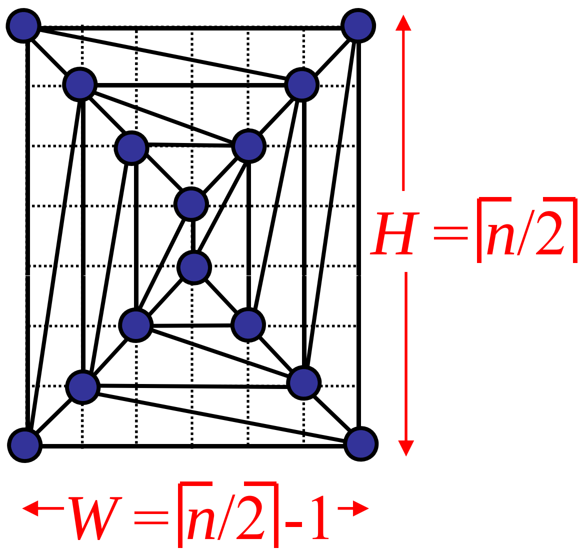
Grid Size : $(W + H \leqq n\text{-}1)$

$W \leqq \lceil n/2 \rceil\text{-}1,\ H \leqq \lceil n/2 \rceil$

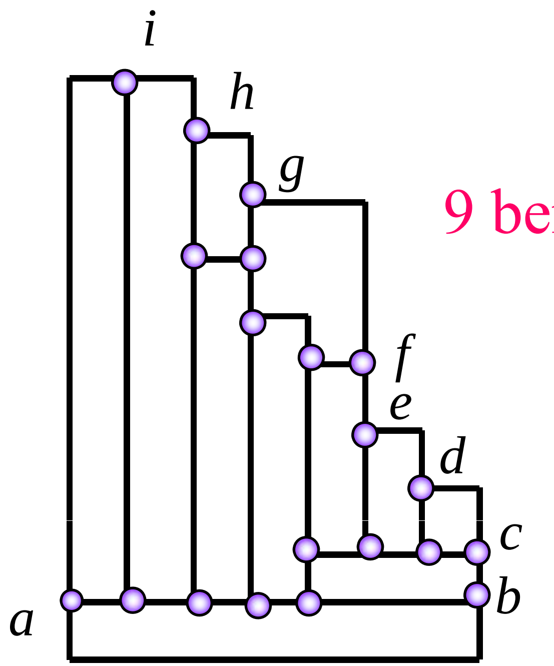
Area: $W \times H \leqq \left(\frac{n-1}{2}\right)^2$



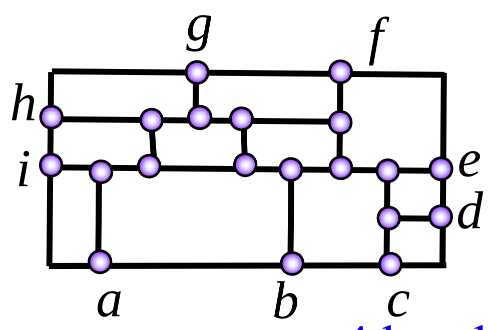
The algorithm is **best possible**



Objective



9 bends



4 bends

an orthogonal drawing with the minimum number of bends.

To minimize the number of bends in an orthogonal drawing.