Planarity

Given two positive integers n and m, construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw it in such a way that there are no crossing edges. If it is not, determine the thickness of the graph.

Construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw a plane graph. If it is not, determine the thickness of the graph.

- simple graph
- planarity
- plane graph
- thickness

Construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw a plane graph. If it is not, determine the thickness of the graph.

- simple graph – an undirected graph in which both multiple edges and loops are disallowed
- planarity
- plane graph
- thickness

Construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw a plane graph. If it is not, determine the thickness of the graph.

- simple graph – an undirected graph in which both multiple edges and loops are disallowed
- planarity – a graph that is planar can be drawn such that no edges cross each other
- plane graph
- thickness

-

Construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw a plane graph. If it is not, determine the thickness of the graph.

- simple graph – an undirected graph in which both multiple edges and loops are disallowed
- planarity – a graph that is planar can be drawn such that no edges cross each other
- plane graph – the actual drawing
- thickness

Construct a random simple graph with n vertices and m edges and determine whether the graph is planar. If it is, draw a plane graph. If it is not, determine the thickness of the graph.

- simple graph – an undirected graph in which both multiple edges and loops are disallowed
- planarity – a graph that is planar can be drawn such that no edges cross each other
- plane graph – the actual drawing
- thickness – the smallest number of planar graphs into which the edges of G can be partitioned

Al Gore Rhythms

- Planarity check
- Thickness algorithm
- Drawing algorithm
-

Al Gore Rhythms

- Planarity check – we made a not-so-efficient algorithm
- Thickness algorithm
- Drawing algorithm
-

Al Gore Rhythms

- Planarity check – we made a not-so-efficient algorithm
- Thickness algorithm – works on some graphs
- Drawing algorithm
-

Al Gore Rhythms

- Planarity check – we made a not-so-efficient algorithm
- Thickness algorithm – works on some graphs
- Drawing algorithm – we didn't write a drawing algorithm
-

Al Gore Rhythms

- Planarity check – we made a not-so-efficient algorithm
- Thickness algorithm – works on some graphs
- Drawing algorithm – we didn't write a drawing algorithm
- But we create did the script

Al Gore Rhythms

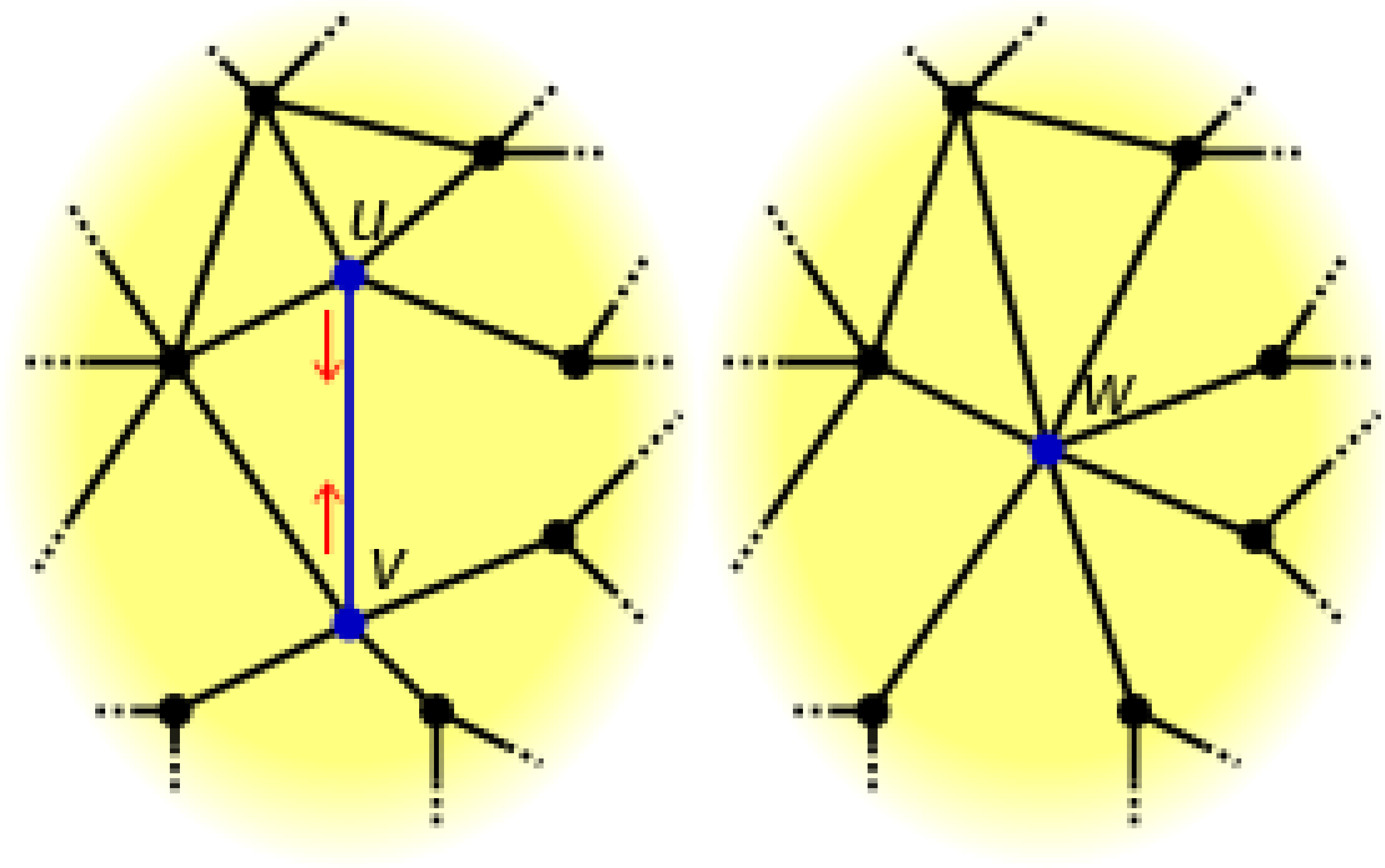
- Planarity check – we made a not-so-efficient algorithm
- Thickness algorithm – works on some graphs
- Drawing algorithm – we didn't write a drawing algorithm
- But we create did the script using libraries

Demo

Stellingen

- Een graaf is planair dan en slechts dan als hij niet ofwel K_5 of $K_{3,3}$ als *minor* heeft
- De minor *H* uit *G* is een graaf die uit *G* gemaakt kan worden door de volgende operaties:
  - Een edge verwijderen
  - Een vertex verwijderen
  - *Edge contraction*

Edge Contraction



Stellingen

- Een graaf is planair dan en slechts dan als al zijn *biconnected* componenten planair zijn
  - *Biconnected* betekend dat er geen enkele vertex kan worden weggehaald zodat de graaf niet meer *connected* is.

Stellingen

- Voor een planaire graaf met v >= 3 geldt:
  e < 3v - 6
- Voor een planaire graaf met v >= 3 zonder cycels van lengte 3 geldt: e < 2v - 4
- Voor *v* het aantal vertices en *e* het aantal edges

Algoritme

- Allereerst wordt de graaf gesplits in zijn *biconnected* componenten
- Voor ieder component wordt bepaald of het K_5 of $K_{3,3}$ als *minor* bevat
- Dit doen we door alle mogelijke *minors* af te gaan en te kijken of deze *isomorph* zijn aan K_5 of $K_{3,3}$
- O(2^n)

Algoritme

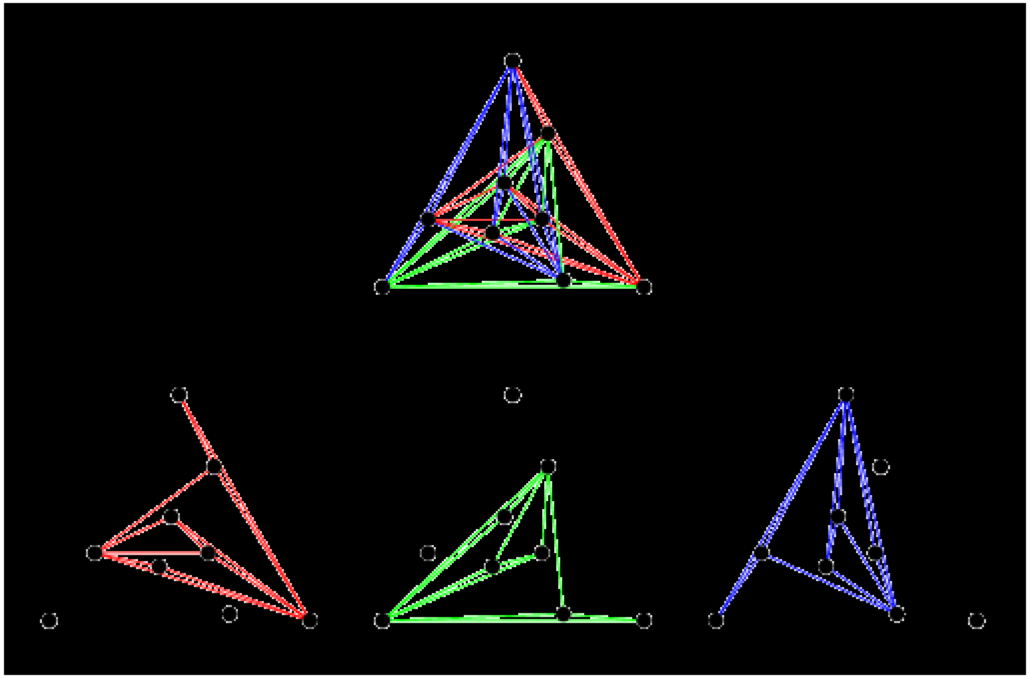
- We kunnen het algorithme nog optimalizeren:
- Wanneer de vergelijkingen $e < 3v - 6$ of $e < 2v - 4$ niet gelden voor een graaf, is de graaf niet planair
- Dus heeft de graaf K_5 of $K_{3,3}$ als *minor*
- Wanneer we dus een *minor* van G vinden waarvoor de vergelijkingen niet gelden, is G ook niet planair

Algoritme

- Dus gaan we eerst de *minors* proberen te maken uit G die zoveel mogelijk edges hebben met zo min mogelijk vertices
- Dus we proberen *e* te maximaliseren, en *v* te minimaliseren
- Is alleen sneller voor niet-planaire grafen, voor planaire grafen moeten nog steeds alle *minors* doorlopen worden

Thickness

- Kleinste aantal planaire deelgrafen waarin je de de edges van een graaf kan verdelen.



Thickness

- Eerste algoritme: complexiteit te groot

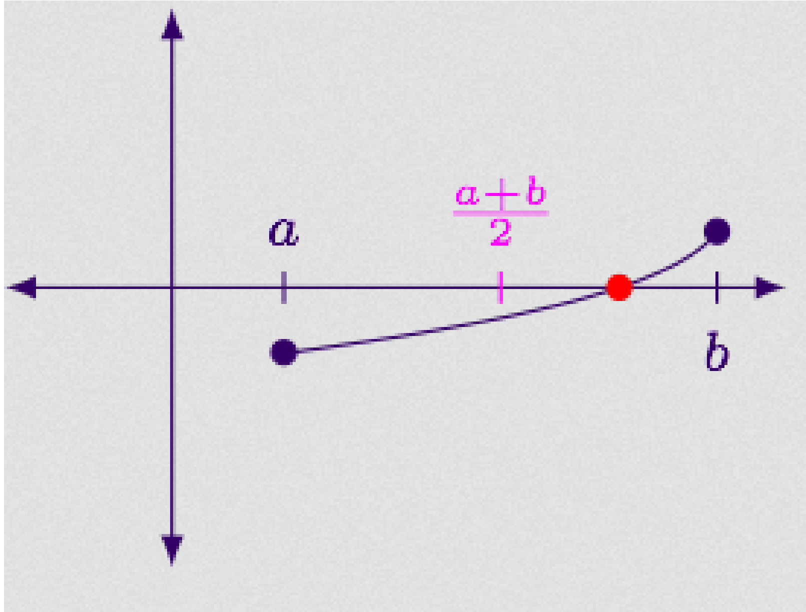
Thickness

- Eerste algoritme: complexiteit te groot

$$e(e!)^2(e^4 + n^2)$$

Thickness

Thickness-Bisection algoritme:



Thickness

Thickness-Bisection algoritme:

```
e               thickness is smaller than this
e-1             thickness is smaller than this
e-2 max,        thickness could be smaller than this
.     |
.     V
.           <- avg = (min + max)/2
.     ^
.     |
3   min         thickness is larger than this
2               thickness is larger than this
1               thickness is larger than this
```

Thickness

Thickness-Bisection algoritme:

- Bepaal of de thickness groter of kleiner dan avg is:
  - Ga alle verdelingen van de graaf in avg delen af
  - Als je een verdeling hebt gevonden met allemaal panaire grafen, is de thickness kleiner dan of gelijk aan avg
  - Anders is de thickness groter dan avg

Thickness

- Testset: K_n, $K_{n,n}$, één vertex, wheel grahp

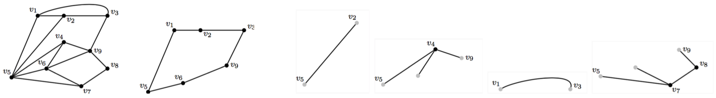
Demo bisection algoritme

Planarity decision algorithm

Demoucron, Malgrange and Pertuiset (1964)

Start with from a graph G

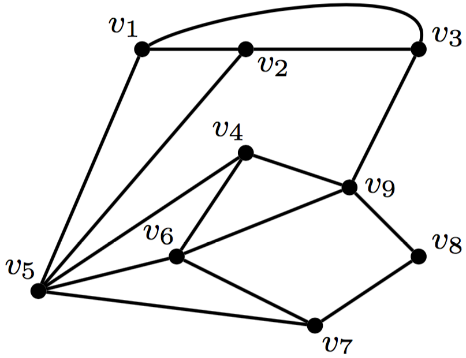
- Take from a graph G a subgraph H
- H is any cycle from G (so H is planar)
- We iteratively extend H to G
- We determine S, a set of
  - edge not in H but endpoints are in H
  - connected component in G, not in H, with the vertices of attachment
- We select a fragment S_i and a face which can accept S_i



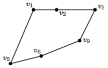
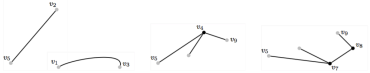
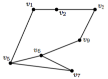
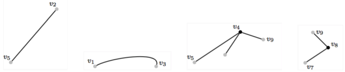
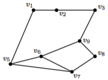
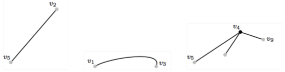
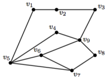
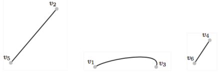
Graph G Subgraph H of G H-fragments

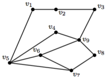
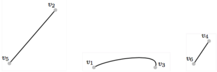
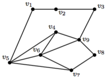
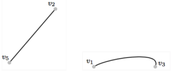
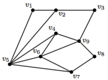
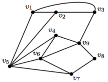
-

1. Choose a H
2. Compute all faces of H
3. Compute the fragments
4. If there are no fragments, the graph is planar
5. Compute admissible faces for fragments
6. If there is a fragment without an admissible face, the graph is not planar
7. If there is a fragment only one admissible face, embed it, go to 2
8. Chose a fragment and embed it



Graph G

	$F_1 = 1234965$ $F_2 = 1234965$	 S_1 S_2 S_3 S_4	$F(S_1) = 1,2$ $F(S_2) = 1,2$ $F(S_3) = 1,2$ $F(S_4) = 1,2$
	$F_1 = 123965$ $F_2 = 567$ $F_3 = 1239675$	 S_1 S_2 S_3 S_4	$F(S_1) = 1,3$ $F(S_2) = 1,3$ $F(S_3) = 1,3$ $F(S_4) = 3$ c
	$F_1 = 123965$ $F_2 = 567$ $F_3 = 6987$ $F_4 = 1239875$	 S_1 S_2 S_3	$F(S_1) = 1,4$ $F(S_2) = 1,4$ $F(S_3) = 1$
	$F_1 = 123945$ $F_2 = 567$ $F_3 = 6987$ $F_4 = 5496$ $F_5 = 1239875$	 S_1 S_2 S_3	$F(S_1) = 1,5$ $F(S_2) = 1,5$ $F(S_3) = 2$

	$F_1 = 123945$ $F_2 = 567$ $F_3 = 6987$ $F_4 = 5496$ $F_5 = 1239875$	 S_1 S_2 S_3	$F(S_1) = 1,5$ $F(S_2) = 1,5$ $F(S_3) = 2$
	$F_1 = 123945$ $F_2 = 567$ $F_3 = 6987$ $F_4 = 465$ $F_5 = 496$ $F_6 = 1239875$	 S_1 S_2	$F(S_1) = 1,6$ $F(S_2) = 1,6$
	$F_1 = 125$ $F_2 = 23945$ $F_3 = 567$ $F_4 = 6987$ $F_5 = 465$ $F_6 = 496$ $F_7 = 1239875$	 S_1	$F(S_1) = 7$
	$F_1 = 125$ $F_2 = 23945$ $F_3 = 567$ $F_4 = 6987$ $F_5 = 465$ $F_6 = 496$ $F_7 = 123$ $F_8 = 139875$		