Planar Graphs

Planarity Testing
Planar Separator Theorem + Applications
Planar Duality
Planar Max-cut

Basic Properties

Planar drawing: No two edges cross.

Face: Given a drawing of G, a face is a “region”
(i.e. equivalence set of points in a plane reachable from each other)

Euler’s formula: If G is connected + planar,
then $v – e + f = 2$

Proof: Induction

Basics

Use Euler's formula to show $m \leq 3n - 6$ for simple planar graphs

Show: For bipartite planar graphs m $\leq$ 2n -4

Show: $K_{3,3}$ and K_5 are non-planar.

Kuratowski's theorem: G is planar if and only if it does not contain a subgraph that is a subdivision of K_5 or $K_{3,3}$.

Subgraph: Obtained by deleting some edges and vertices.

Subdivision: Repeatedly insert a vertex in middle of an edge.

Basics

6-colorable

5-colorable: Induction on v
+ look at conn. components of colors 1 and 3.
+ look at components of 2 and 4.

Thm: Can color in 4 colors.
Big result.

Basics

Fary’s theorem (1936): Any planar graph can be drawn such that edges are straight lines

There is a whole research area called graph drawing.

Want to draw (almost) planar graphs in various ways with various desirable properties.

Planarity Testing

Linear time algorithms known.

We focus on an insightful $O(n^3)$ time algorithm

Assume connected

Assume 2 edge-connected   (each edge lies in a cycle)

[If removing (u,v) disconnects, can contract (u,v)]

Key insight:

Consider a cycle C. If (u,v) and edge with u and v not on C, then either both lie inside C or both lie outside C.

Planarity Testing

Let $G_1, \dots, G_k$ denote components obtained on removing C.

Each G_i is connected to C with at least two edges (by 2-connectivity assumption). Call these "connectors"

Call G_i+ endpoints on C = <u>Segment</u> S_i.

Conflicting segments: If they "obstruct" each other.
[When do segments conflict?]

Construct compatibility graph H.
(one vertex for each segment,   edge between i and j if S_i and S_j conflict
What can you say about H if G is planar?

Algorithm outline

- Assume 2-connected
- Find cycle C, and let $S_1, \ldots, S_k$ be segments.

- If conflict graph not 2-colorable, return NO.
- Else, recursively test planarity of graph $C + S_i$, for each i=1,…,k.

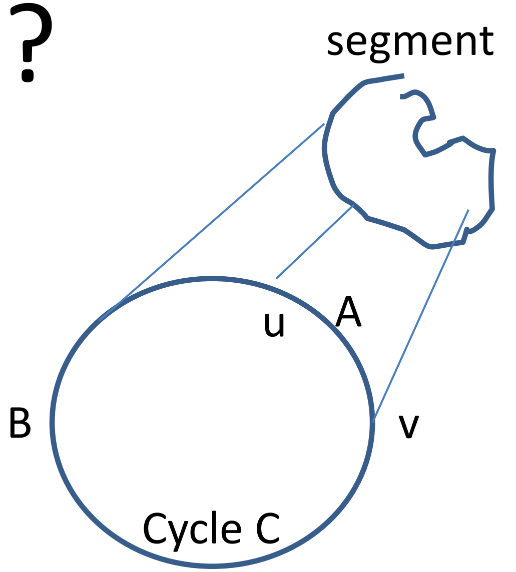
Almost there, except one silly catch

What if C+$S_1 = G$?

Defn: Call a cycle <u>separating</u>, if has >= 2 segments.

Theorem: G has a separating cycle C unless,
(i) G is itself cycle or,
(ii) G is a cycle + segment which is a path

Moreover can find such a C in linear time.



A and B are paths from C from u to v

Proof: Find some cycle C. If it has >= 2 segments, done already.

Else it has one segment S. Consider two consecutive attachments u and v (S may have more than 2 attachments).

Take some path P from u to v in the segment S (not involving C).

Consider a new cycle C' = P + B (see picture)

where B is the "longer" path on C from u to v that contains the other attachments of S (if any).

Planar Separator Theorem

Can break a graph into components by removing few edges such that "no piece too large".

Balanced planar separator theorem: Can remove $4\sqrt{n}$ vertices such that each component of size $\leq \frac{9n}{10}$

[Cannot hope for o($\sqrt{n}$) size balanced separator]

Very useful:

1) Find max independent set in $2^{O(\sqrt{n})}$ time
2) Find $(1+\epsilon)$ approximation in $2^{O(1/\epsilon^2)}$ poly(n) time.
3) ….

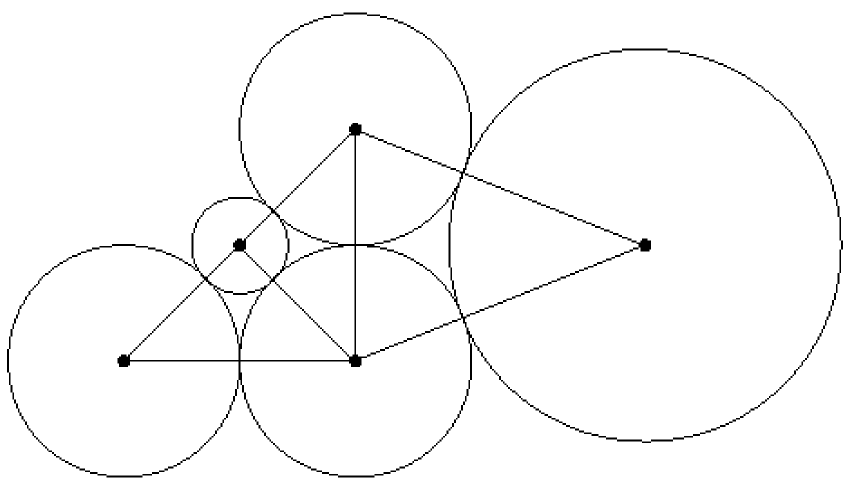
(Solutions for 1 and 2 at the end)

Koebe's disk embedding theorem:

Can draw a planar graph G such that for each vertex u, there is a disc D_u, with the following properties:

(i) Interiors of disks are disjoint

(ii) two discs D_u and D_v touch iff (u,v) is an edge in G.



Proof for Planar separator

Consider disk embedding of G.

Let D be the smallest circle that contains >= n/10 vertices.
Assume radius(D) = 1, center(D) = origin (else scale and shift)

Pick r uniformly at random in interval [1,2]

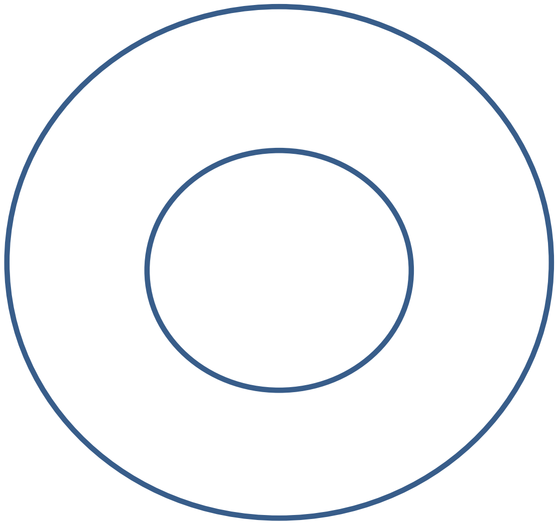
Consider circle C centered at origin of radius r.
Let S = {All disks that intersect C}

(note S is a random set)

Lemma 1: For each r, always get a valid balanced separator (i.e. no component bigger than 9n/10)

Proof: Can cover ring using 8 disks of radius 1

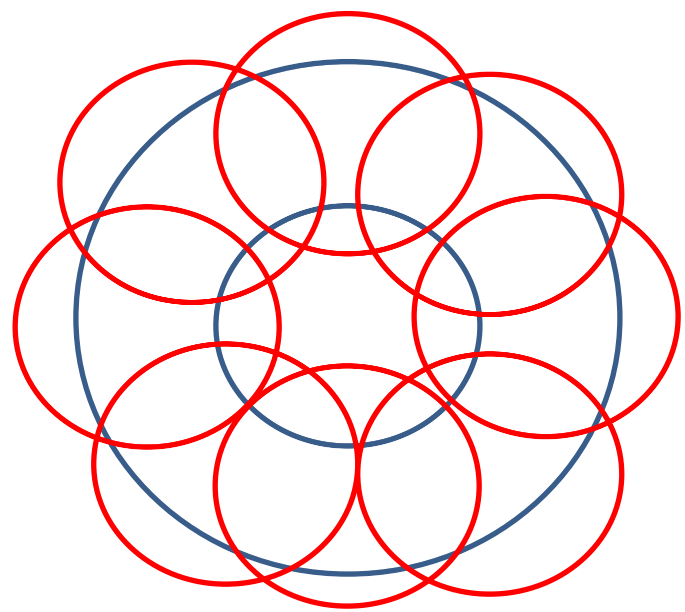
Give upper and lower bounds on number of vertices in inner piece



Lemma 1: For each r, always get a valid balanced separator (i.e. no component bigger than 9n/10)

Proof: Can cover ring using 8 disks of radius 1

Give upper and lower bounds on number of vertices in inner piece



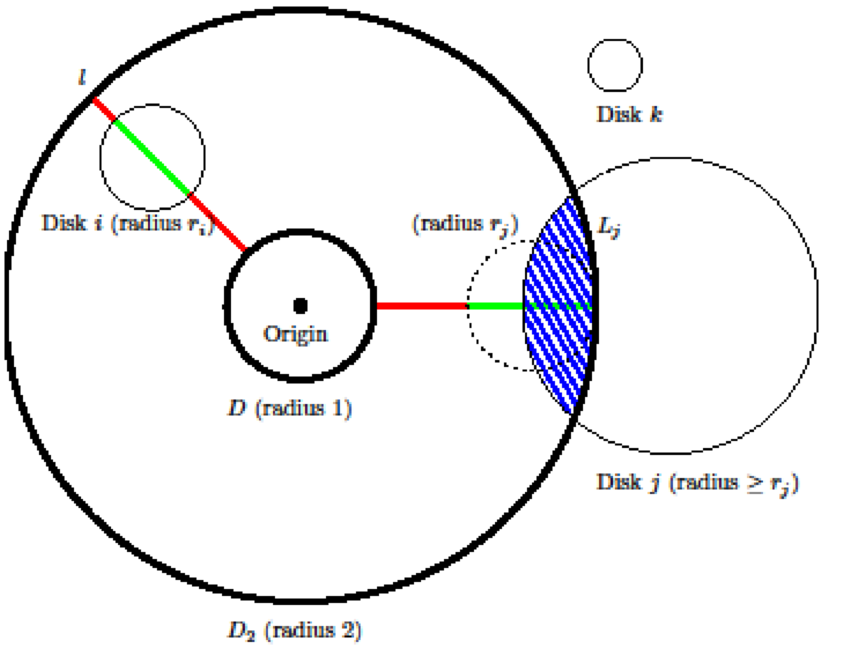
Lemma 2: E[S] $\leq 4\sqrt{n}$.

Analysis: Lemma 2

Pr[Cut Disk i] $\leq 2\, r_i$

E[S] $\leq 2\ \sum_i r_i$

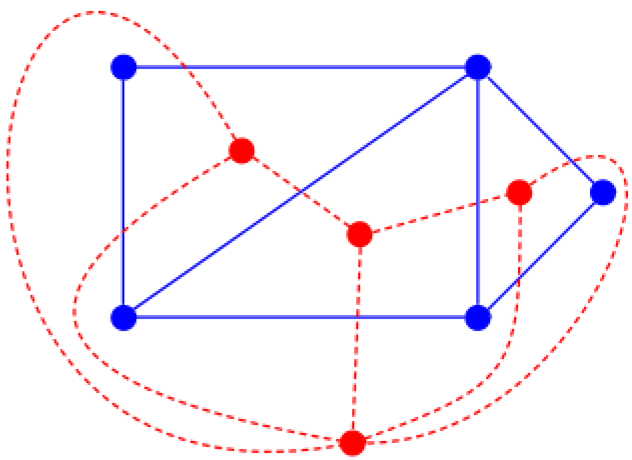
But $\sum_i \pi r_i^2 \leq 4\,\pi$

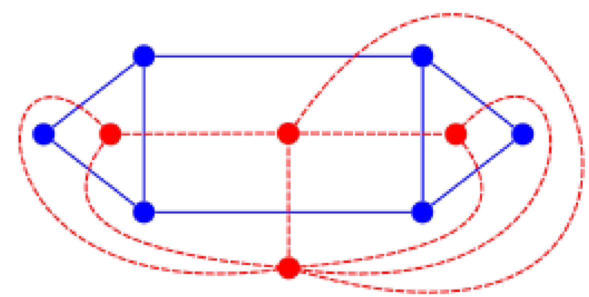


Planar Duality

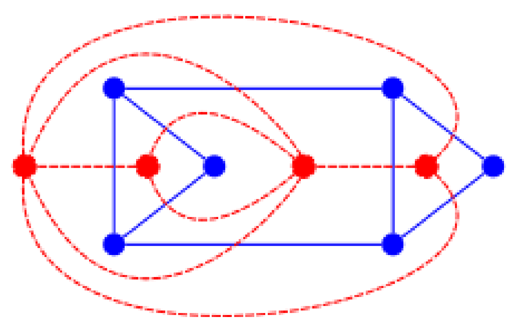
With every connected planar multi-graph can associate a dual graph G^*

Faces -> Vertices   Vertices -> Faces

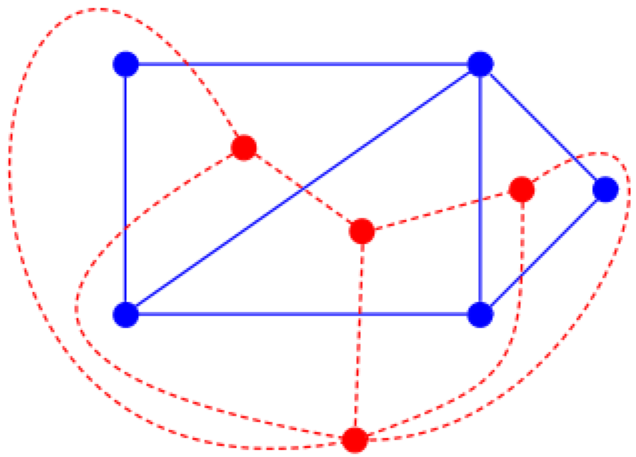




G^* might depend on the particular drawing of G.



Dual(Dual(G)) = G.



No self-loops iff graph is 2 edge-connected.

One can also show that no parallel edges iff no cuts of size 2 (i.e. graph is 3-edge connected).

(formally later when see cycle–cut duality)

Max-cut in Planar Graphs

In general graphs Max-cut is NP-hard (as opposed to min-cut)

Very basic problem

(delete fewest edges to make graph bipartite)

(minimum energy state of charges)

Can solve in planar graphs exactly

Cut Space and Cycle Space

Consider the 0-1 incidence vector of a cut.
(view as vector in n(n-1)/2 dimensions over field F_2)

Claim: Form a vector space.
Proof: Closed under addition and scalar multiplication

Claim: Generated by singleton cuts.
(i.e. span(singleton cuts) = Cut Space)

Cycle Space

Consider the 0-1 incidence vector of collection of edge disjoint cycles

(view as 0-1 vector in n(n-1)/2 dimensions over field F_2)

Claim: Form a vector space.

Proof: Closed under addition and scalar multiplication

Claim: This space is same is Eulerian subgraphs of G

Proof: Why?

Connection

Cut Space(G) = Cycle Space(G^*)

Claim: Every cut in G corresponds to a collection of Cycles in G^*.

Claim: Every collection of cycles in G^* corresponds to a cut in G.
(Hint: Cycle = sum of faces)

So, max-cut (G) = max Eulerian subgraph in G^*.

How do we find it?
(Hint: Think about how we solved Chinese Postman)

Uniqueness of Planar Drawings

Thm: We will show that theorem that if a graph G is 3-vertex connected then it is a unique drawing.

Claim: In any drawing C forms a face

iff it is an induced cycle and G\V(C) has 1 component.

For => (consider where G\(V(C) can lie)

For <= Take a face f, and consider corresponding cycle C

Claim: C is induced cycle. If not, pick Consider C\{u}\{v}

(3 connected implies path from x to y. Add a vertex in face and connect to u,v,x,y. Get K_5 subdivision.)

Also, G\V(C) cannot have >=2 components. Pick two vertices, at least 3 vertex disjoint paths touching C. Let a,b,c, be some 3 points of contact.

Add extra vertex in face and connect to a,b,c. Get K_3,3.

Exercises

1. Using cut-cycle duality, show that G* has

a) No self loops if graph G is 2-edge connected.

b) No parallel edges if G is 3-edge connected

Solutions

1. Let S be separator. And $G_1, \ldots, G_k$ be pieces of G\S.

   For each possible independent set X of S,

   Recursively find optimum independent sets of (G_i + S) for i=1,…,k, conditioned on the fact that X is picked in S.

Recurrence: $T(|G|) \leq 2^{|S|} \; (\sum_i \; T(|G_i|)$

Can show that $2^{c\sqrt{n}}$ fits for n large enough.

Even a sloppy calculation where we assume that we get n components all of size 9n/10 would work

T(n) $\leq$ $2^{4\sqrt{n}}$ n T(9n/10)

Problem 2

Algorithm: Keep applying separator theorem to each piece, as long as it has more than than $\frac{1}{\epsilon^2}$ vertices.

This is more tricky. Let us see two simplest ideas that do not work.

Let S(n) denote the number of edges we need to remove from a n vertex planar graph of get pieces of size 1/\epsilon^2.
Then get recurrence
$S(n) \leq 4\sqrt{n} + S(n_1) + \ldots + S(n_k)$
Where each $n_i \leq \frac{9n}{10}$.

Even if k=2, Assuming n_1, n_2 = 9n/10 will not work.
Guess: $S(n_i) \leq c\,\epsilon\sqrt{n_i}$

Show that S(n) $= c\,\epsilon n - d\sqrt{n}$ works for some constants c and d.

Another solution based on an intuitive charging argument

We need to count how many vertices are removed over all.

So let us try distributing the “mass” of vertices as we remove them, and then collect it later.

This is a very useful way of counting.

Every time you split a piece G by removing the separator X, put X/G mass on every vertex in G.

Let us count how much mass each vertex can get at the end of the algorithm.