# PTASes for Planar Graphs

Glencora Borradaile

# Two decades of PTASes

mid ’90s:
PTASes for “local” planar graph problems [Baker]
(vertex cover, independent set, etc)

’97-’03:
PTASes for low-d geometric problems [Arora + others]
(TSP, Steiner tree, etc)

late ’00s:
PTASes for planar connectivity problems [BKM + others]
(TSP, Steiner tree, etc)

# Baker's Technique

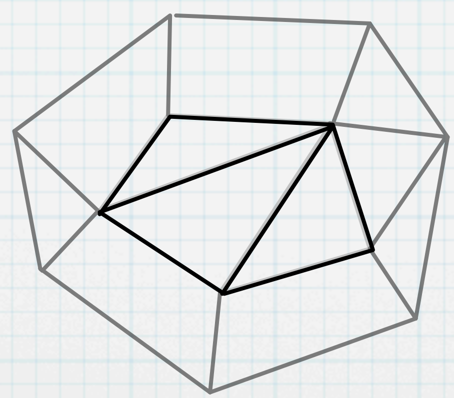
PTAS for max independent set, min vertex cover, etc. in planar graphs

Partition the planar graph into k-outerplanar graphs.

Solve the problem optimally in each k-outerplanar graph.

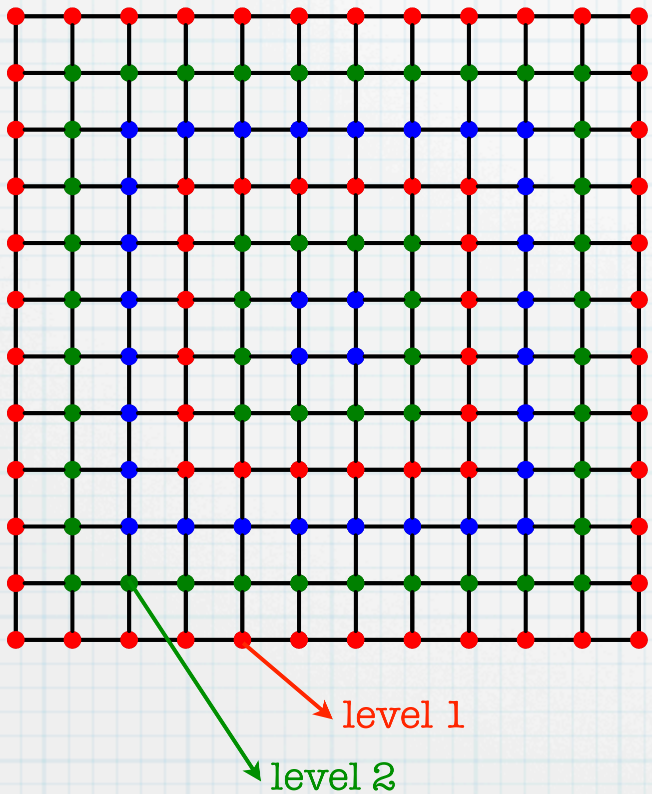
(k-outerplanar graphs have treewidth < 3k;
many NP-hard problems are easy in bounded-treewidth graphs.)

Argue that the union of these solutions is a near-optimal solution for the original graph.



# Baker's Technique

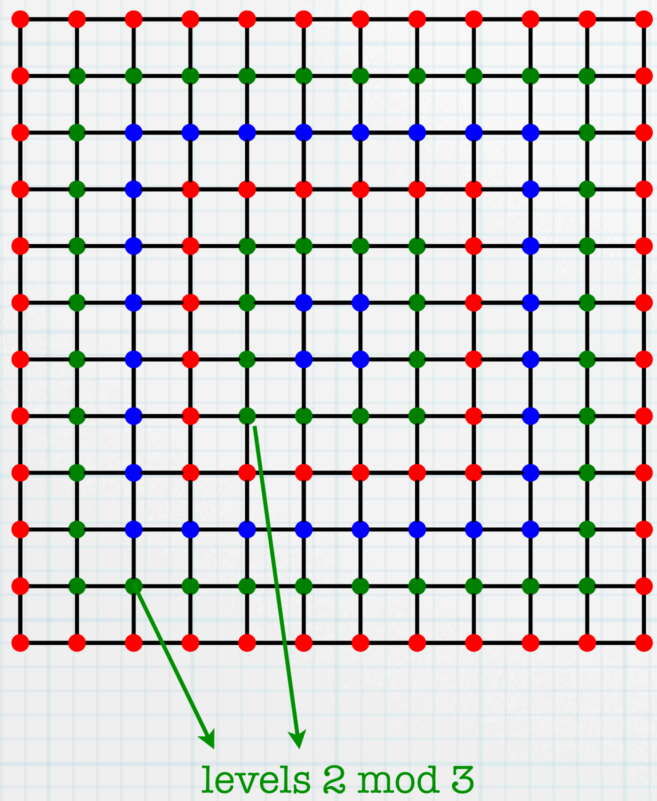
## PTAS for max independent set, min vertex cover, etc. in planar graphs




Partition the vertices into breadth-first search levels from the boundary.

# Baker's Technique

PTAS for max independent set, min vertex cover, etc. in planar graphs




Partition the vertices into breadth-first search levels from the boundary.

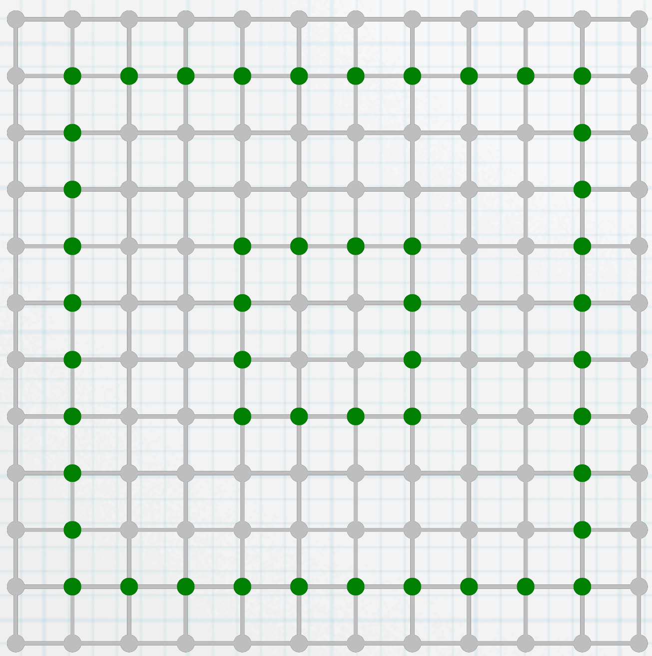
Delete levels congruent to i mod k.

Resulting components are (k-1)-outerplanar.

Find the optimal solution in each component.

# Baker's Technique

PTAS for max independent set, min vertex cover, etc. in planar graphs



Partition the vertices into breadth-first search levels from the boundary.

Delete levels congruent to i mod k.

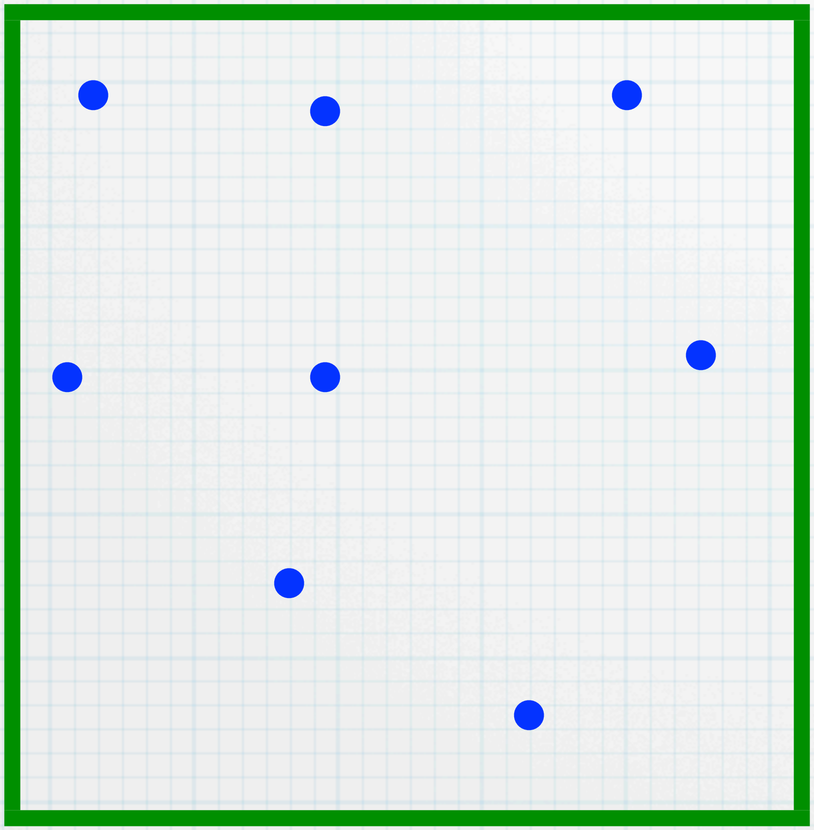
Resulting components are (k-1)-outerplanar.

Find the optimal solution in each component.

OPT ∩ {congruence class} < OPT/k for some i.
The union of solutions in each component is within (1±1/k) OPT.

# Arora's Technique

PTAS for connectivity problems in low-d geometric space



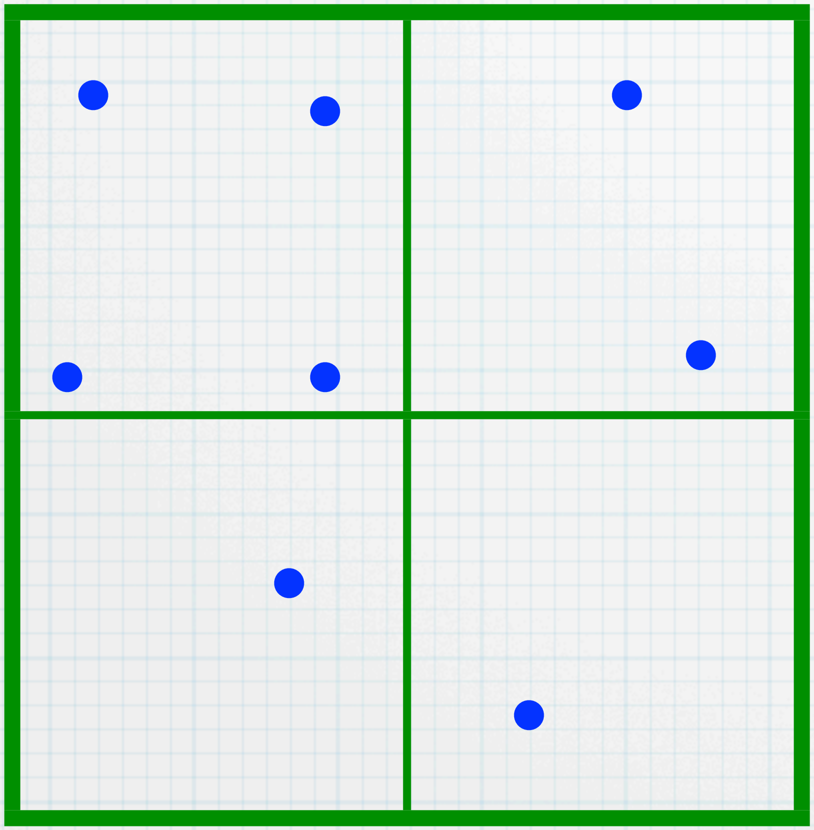
Example: Steiner tree.

Bound terminals with a box.

Decompose space with a quad tree.

# Arora's Technique

PTAS for connectivity problems in low-d geometric space



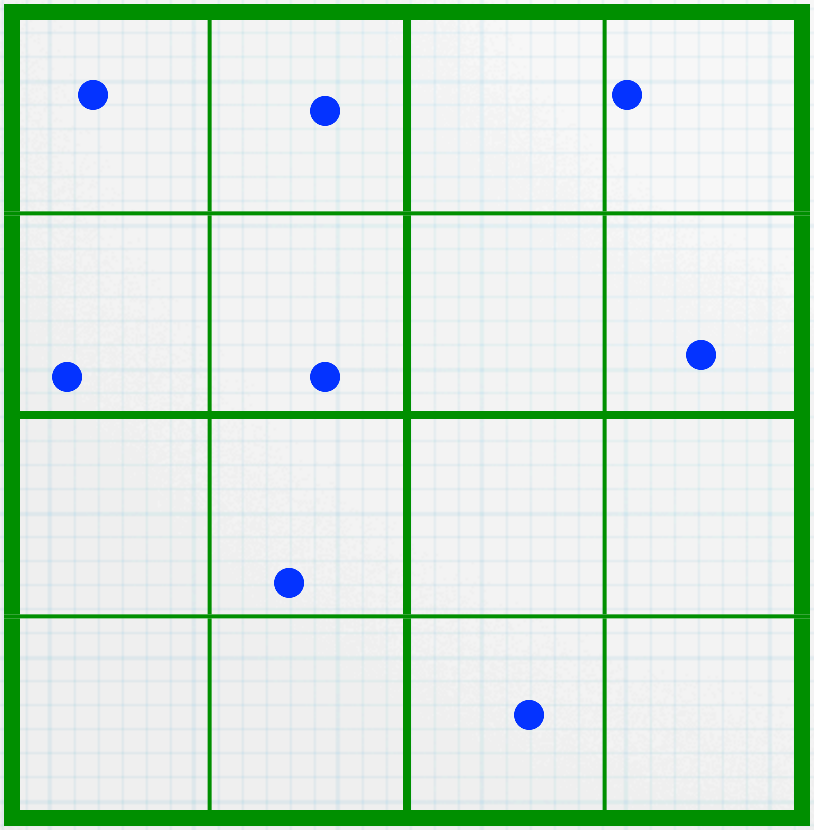
Example: Steiner tree.

Bound terminals with a box.

Decompose space with a quad tree.

# Arora's Technique

PTAS for connectivity problems in low-d geometric space



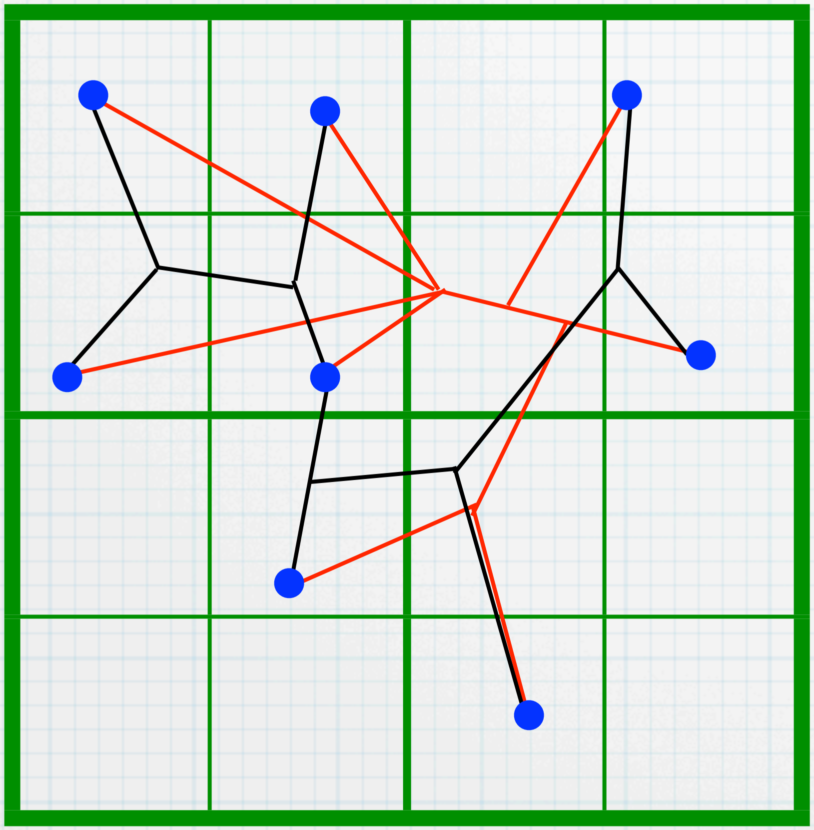
Example: Steiner tree.

Bound terminals with a box.

Decompose space with a quad tree.

# Arora's Technique

PTAS for connectivity problems in low-d geometric space



Example: Steiner tree.

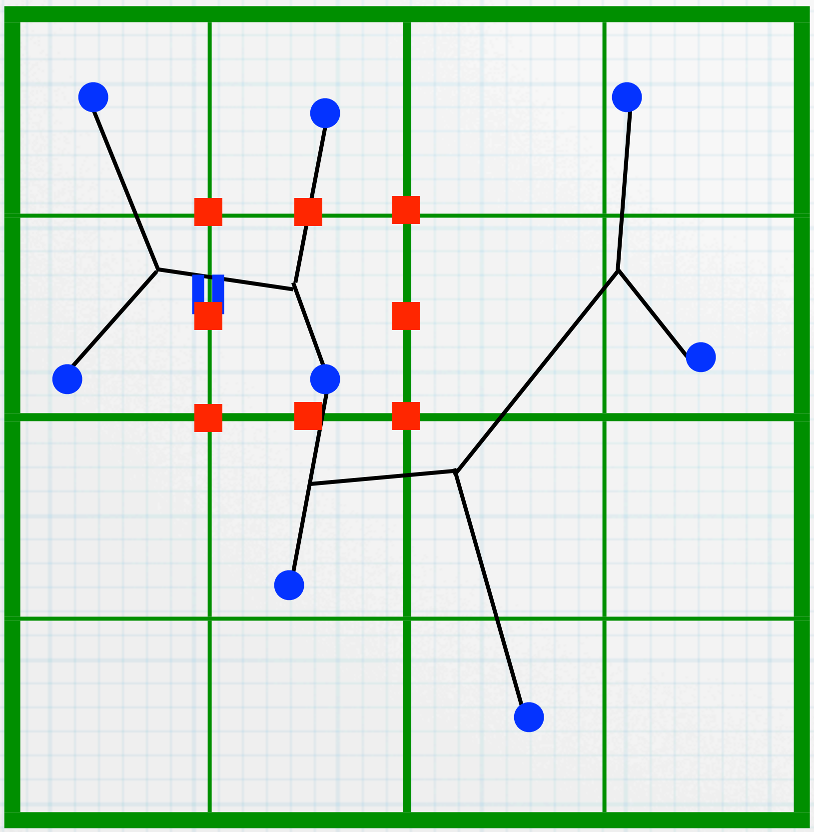
Bound terminals with a box.

Decompose space with a quad tree.

Structure Theorem: There is a near-OPT solution that crosses each grid cell O(1) times.

# Arora's Technique

PTAS for connectivity problems in low-d geometric space



Example: Steiner tree.

Bound terminals with a box.

Decompose space with a quad tree.

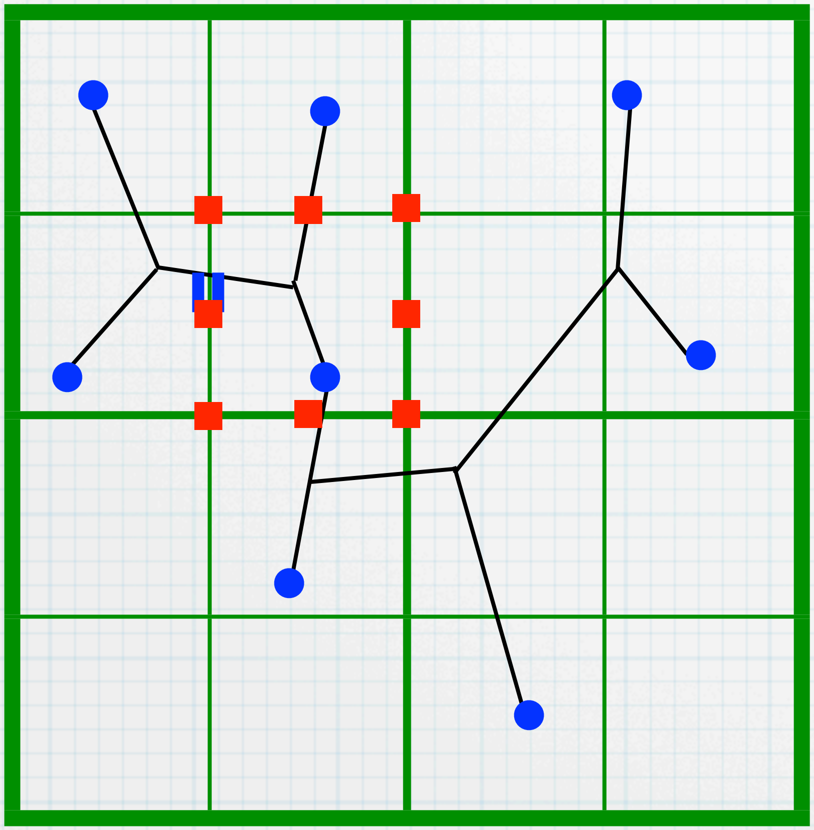
Structure Theorem: There is a near-OPT solution that crosses each grid cell O(1) times.

Limit solution to cross between cells at portals.

(Bound sum of detours by $\epsilon$OPT).

# Arora's Technique

## PTAS for connectivity problems in low-d geometric space



Example: Steiner tree.

Bound terminals with a box.

Decompose space with a quad tree.

Structure Theorem: There is a near-OPT solution that crosses each grid cell O(1) times.

Limit solution to cross between cells at portals.

(Bound sum of detours by $\epsilon$OPT).

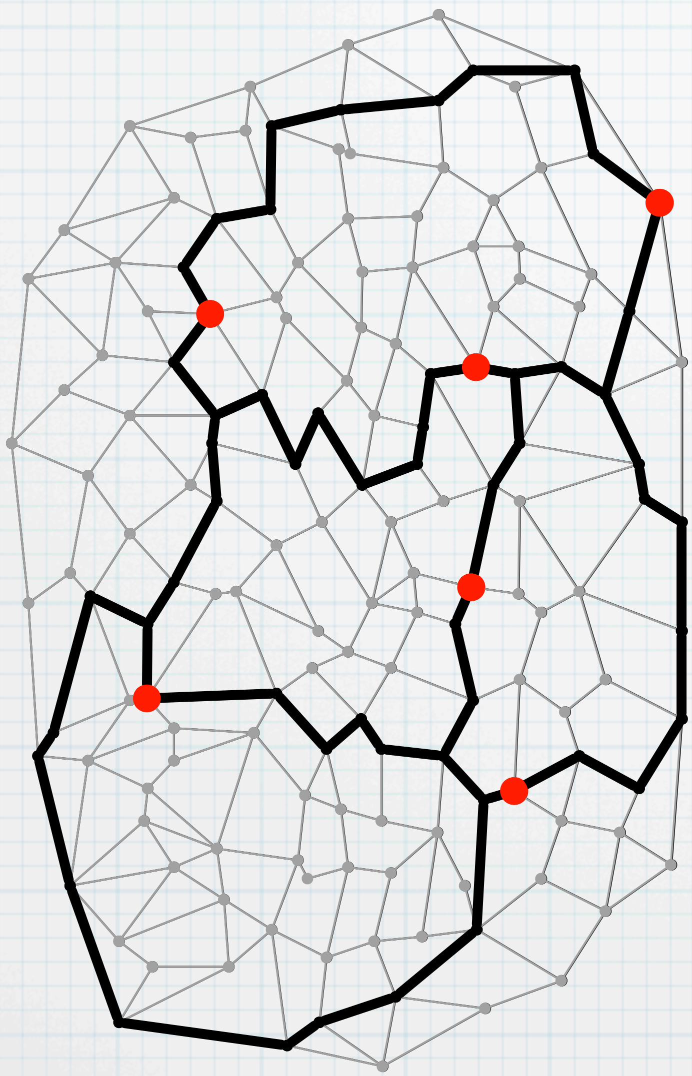
Find the best portal-respecting solution using dynamic programming.

# Brick Technique

PTAS for TSP, Steiner tree, etc. in planar graphs

1. Find a special grid-like subgraph.
   Structure Theorem: There is a $(1+\epsilon)$ OPT solution that crosses the boundary of each face of the grid $O(1)$ times.
2. Restrict solution to use few portals (as Arora).
3. Break grid into k-outerplanar pieces (as Baker).
4. Find the optimal portal-respecting solution in each piece.

# Brick Decomposition

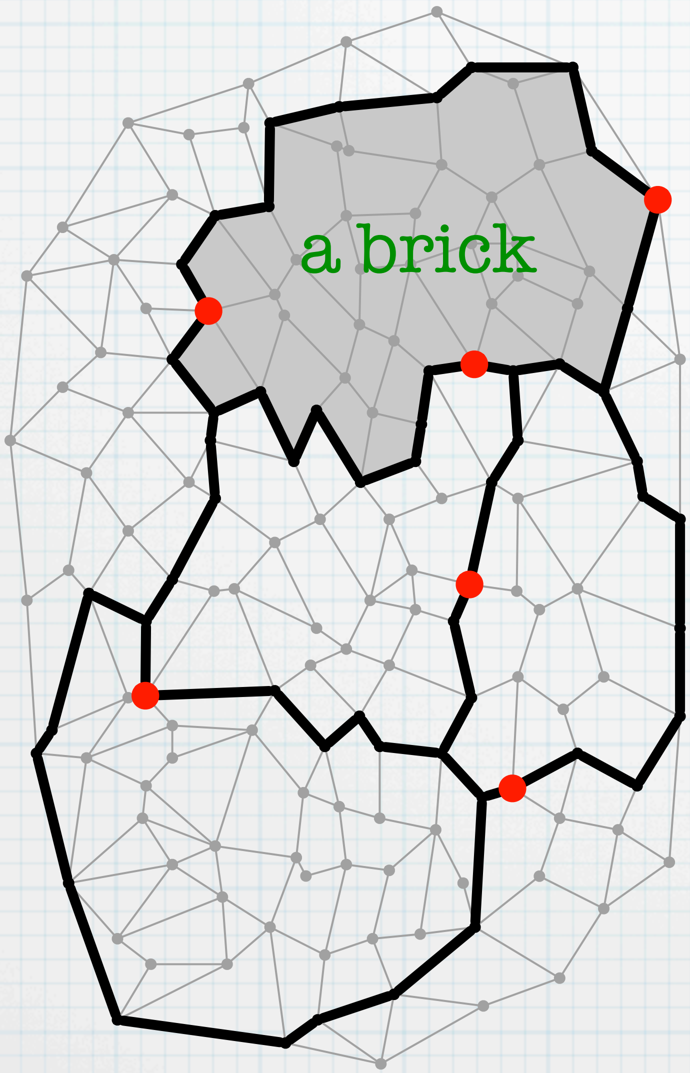


[made of shortest paths]

Grid subgraph:

- spans terminals
- w(grid) is O(OPT)
- each face is a brick

# Steiner-Tree Structure Theorem



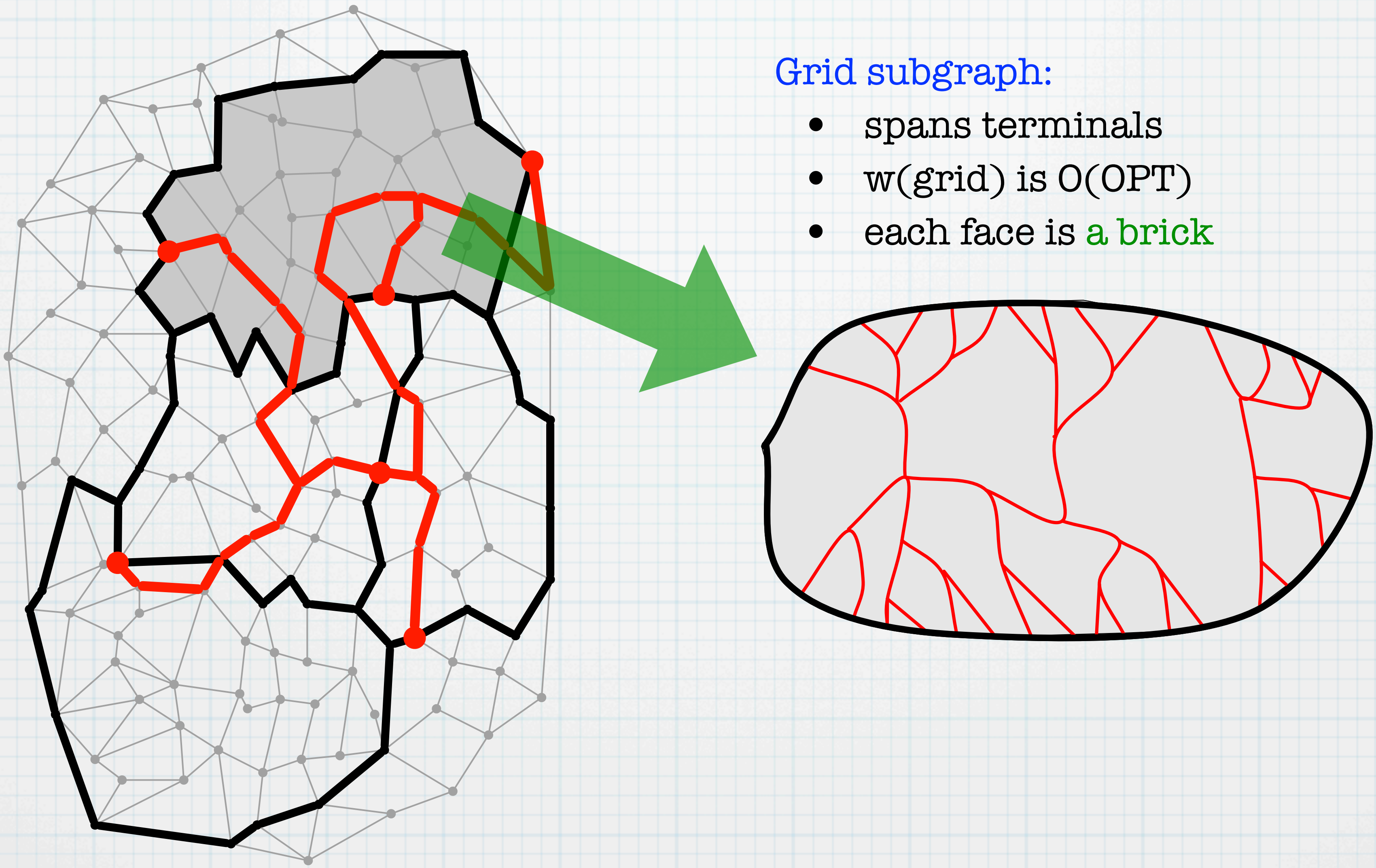

Grid subgraph:

- spans terminals
- w(grid) is O(OPT)
- each face is a brick

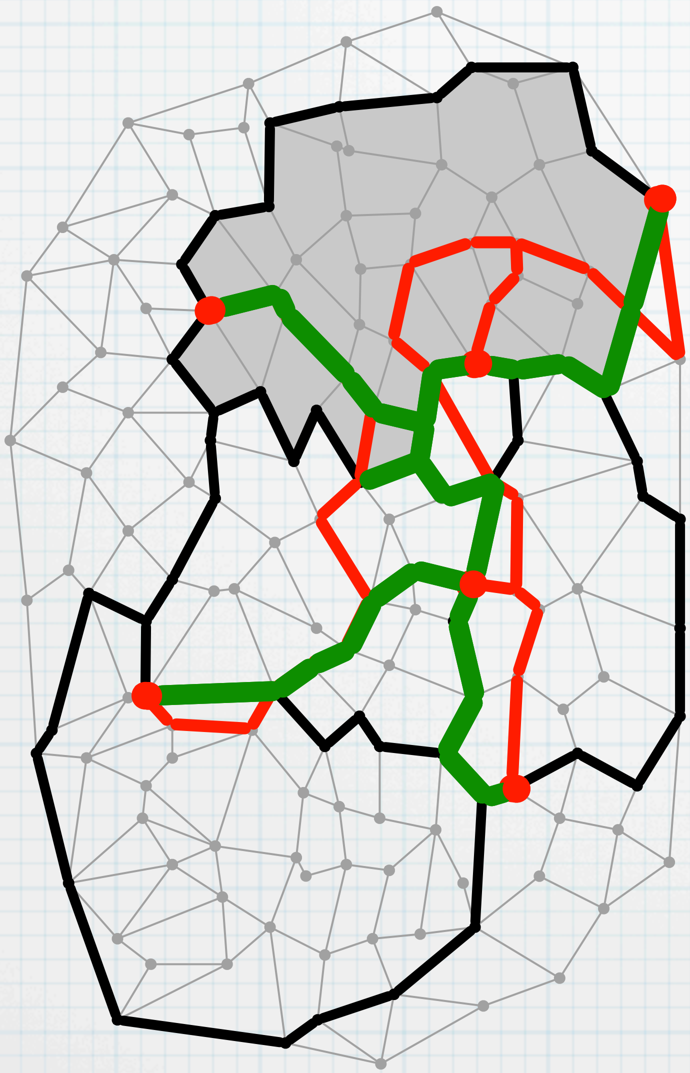
# Steiner-Tree Structure Theorem

Grid subgraph:

- spans terminals
- w(grid) is O(OPT)
- each face is a brick

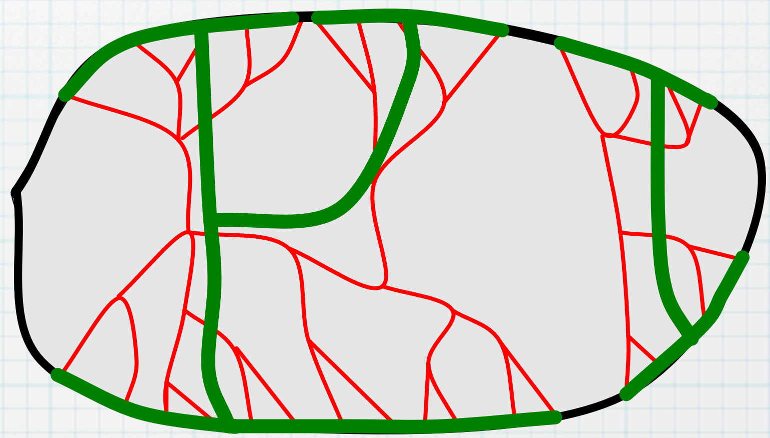


# Steiner-Tree Structure Theorem



Grid subgraph:

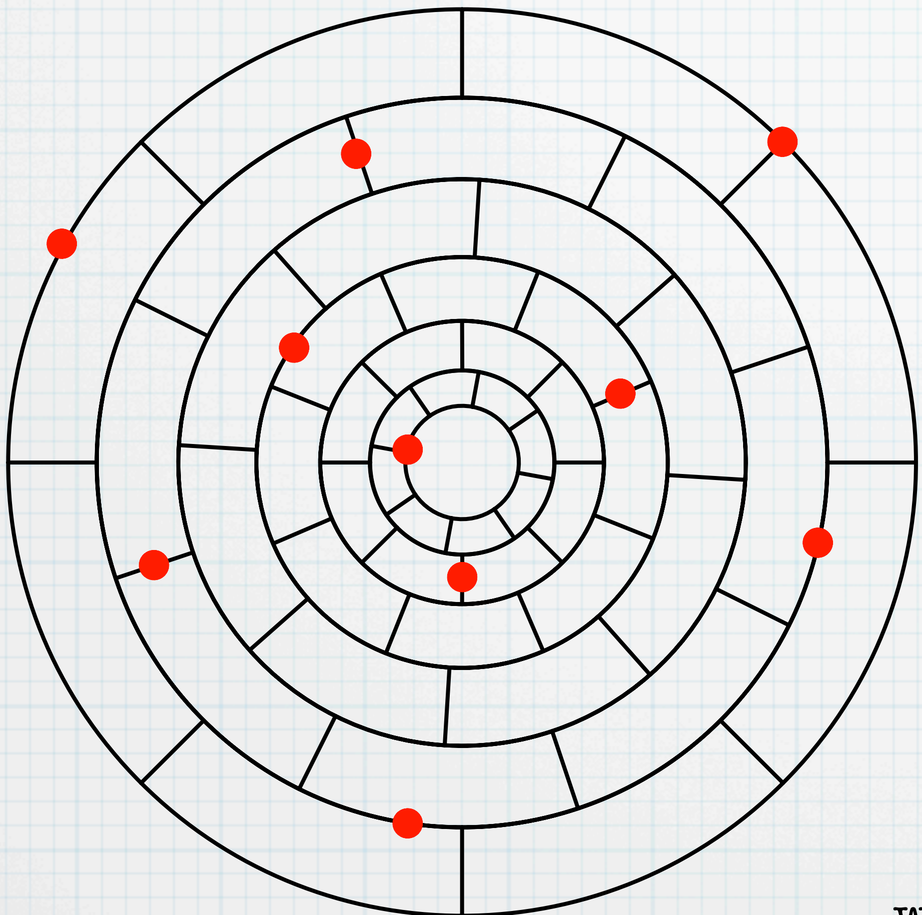
- spans terminals
- w(grid) is O(OPT)
- each face is a brick



Steiner-Tree Structure Theorem:

- w(green) < $(1+\epsilon)$ w(red)
- O(1) green leaves
- green achieves red's connectivity

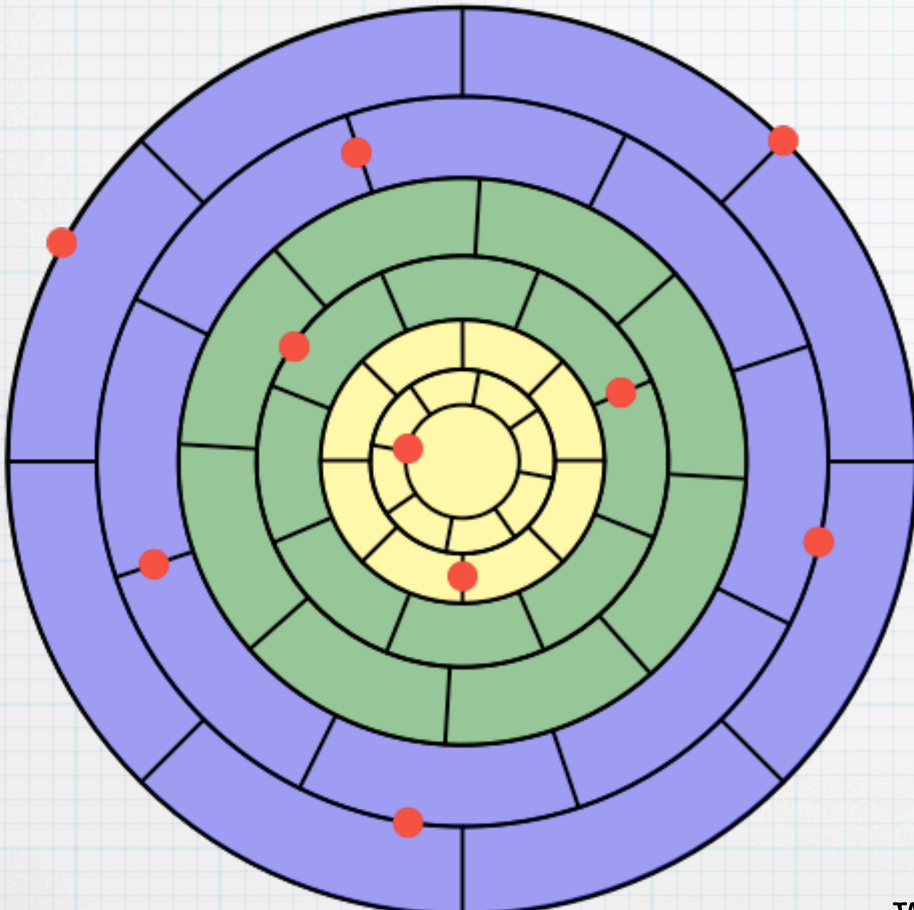
# The PTAS



1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annuli.
6. Union these solutions together.

weight(grid graph) is O(OPT)

# The PTAS

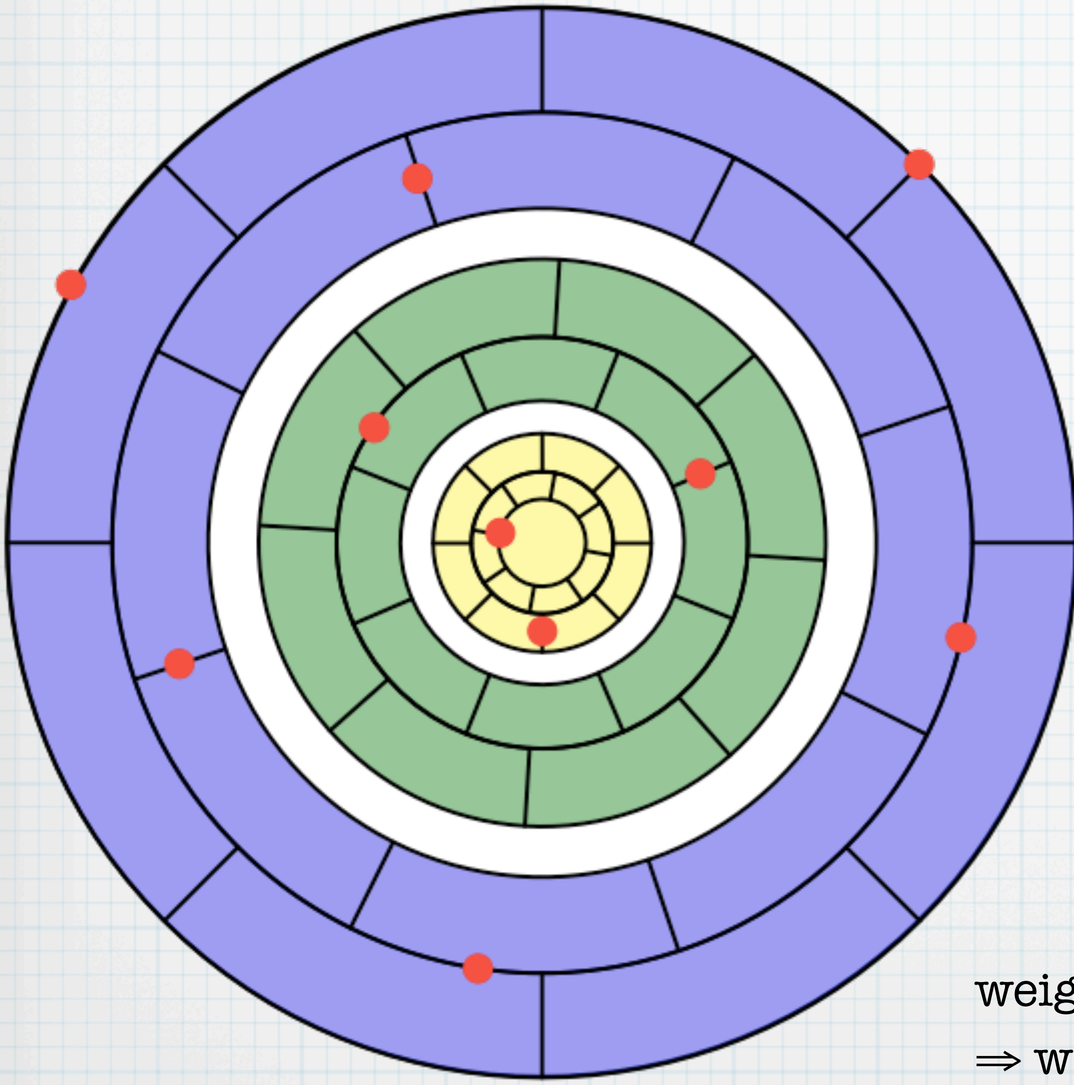


1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annulus.
6. Union these solutions together.

weight(grid graph) is O(OPT)

⇒ weight(annuli boundaries) = $\epsilon$OPT

# The PTAS

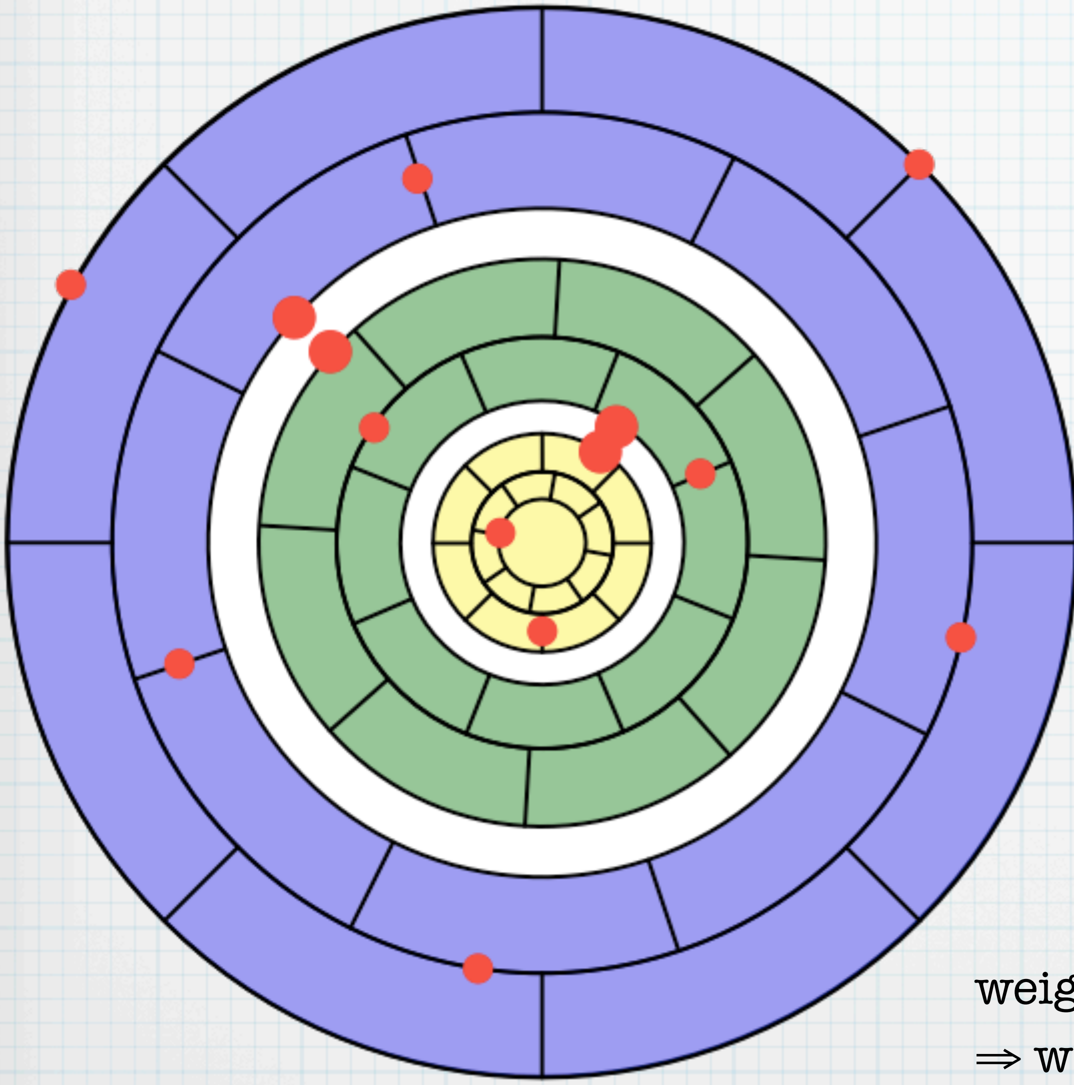


1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annuli.
6. Union these solutions together.

weight(grid graph) is O(OPT)

$\Rightarrow$ weight(annuli boundaries) = $\epsilon$OPT

# The PTAS

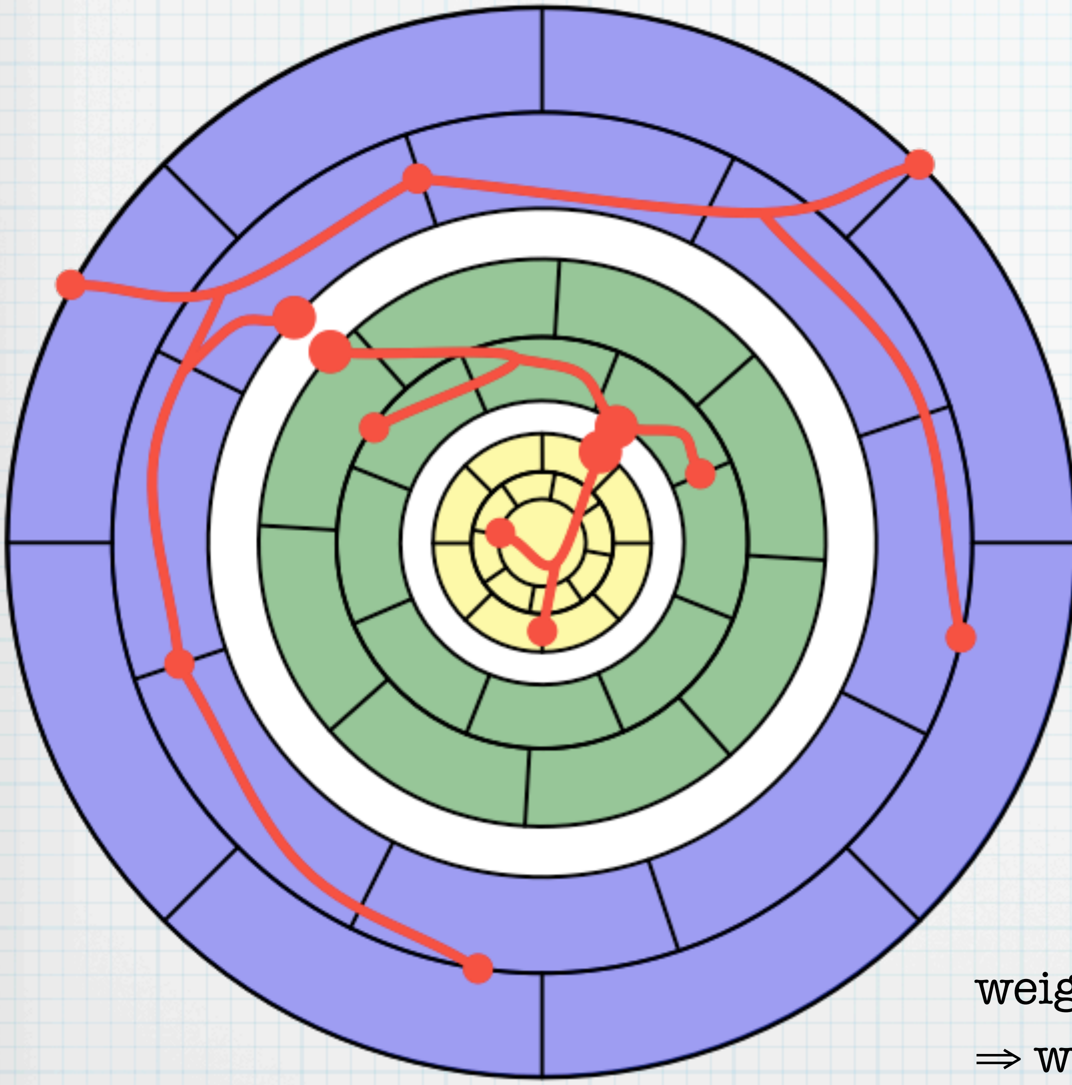


1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annuli.
6. Union these solutions together.

weight(grid graph) is O(OPT)

$\Rightarrow$ weight(annuli boundaries) = $\epsilon$OPT
$\Rightarrow$ connecting to new terminals costs < $\epsilon$OPT

# The PTAS



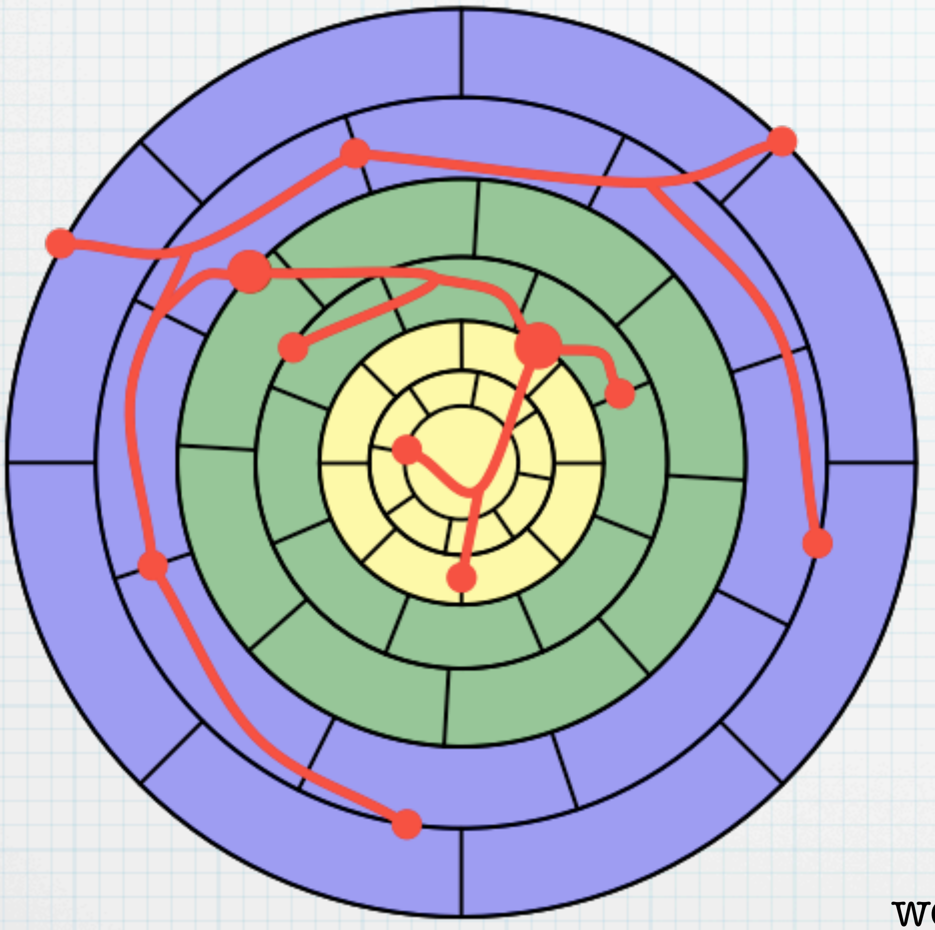
1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annuli.
6. Union these solutions together.

weight(grid graph) is O(OPT)

⇒ weight(annuli boundaries) = $\epsilon$OPT

⇒ connecting to new terminals costs < $\epsilon$OPT

# The PTAS



1. Find the grid subgraph.
2. Group the faces into narrow annuli.
3. Break the annuli apart.
4. Introduce new terminals.
5. Solve the problem in each annuli.
6. Union these solutions together.

weight(grid graph) is O(OPT)

$\Rightarrow$ weight(annuli boundaries) = $\epsilon$OPT

$\Rightarrow$ connecting to new terminals costs < $\epsilon$OPT

# Since then ...

Applied to [Bateni et al.s]:

Steiner forest ['10]
prize collecting problems ['11]
multi-way cut ['12]

and extended to bounded-genus graphs [BDT '09]

using a preprocessing step and then following much as brick technique.

# What makes it all possible?