PTAS for Euclidean Traveling Salesman and Other Geometric Problems

Sanjeev Arora

Journal of the ACM, 45(5):753–782, 1998

PTAS

$\longrightarrow$ same as LTAS, with "Linear" replaced by "Polynomial"

Def Given a problem P and a cost function $|.|$, a PTAS of P is a one-parameter family of PT algorithms, $\{A_\varepsilon\}_{\varepsilon>0}$, such that, for all $\varepsilon > 0$ and all instance I of P, $|A_\varepsilon(I)| \leq (1+\varepsilon)\,|\mathrm{OPT}(I)|$.

PTAS

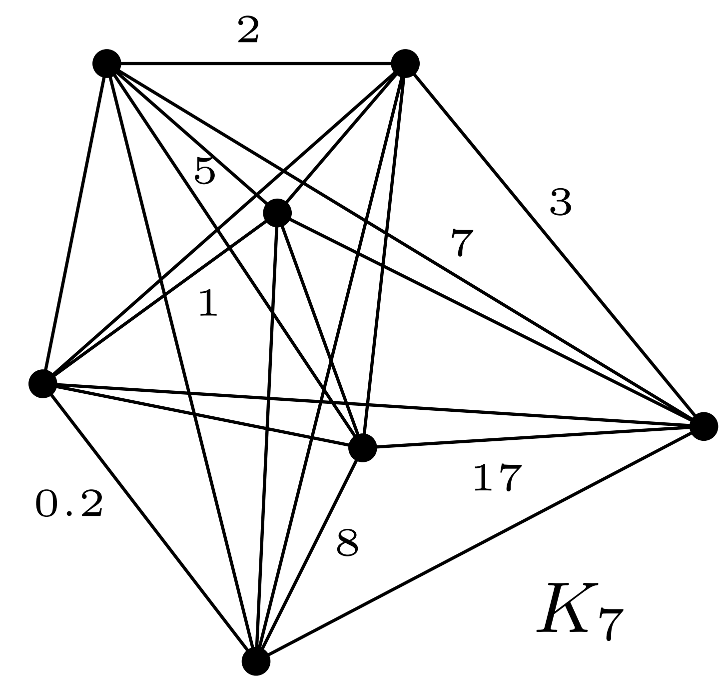
$\rightarrow$ same as LTAS, with "Linear" replaced by "Polynomial"

Def Given a problem P and a cost function $|.|$, a PTAS of P is a one-parameter family of PT algorithms, $\{A_\varepsilon\}_{\varepsilon>0}$, such that, for all $\varepsilon > 0$ and all instance I of P, $|A_\varepsilon(I)| \leq (1 + O(\varepsilon))\ |\mathrm{OPT}(I)|$.

- PT means time complexity $n^{O(1)}$, where the constant may depend on $1/\varepsilon$ and on the dimension d (when pb in $\mathbb{R}^d$)
- As far as we get $n^{O(1)}$, we do not care about the constant
- the constant in $(1 + O(\varepsilon))$ must not depend on I nor on ε

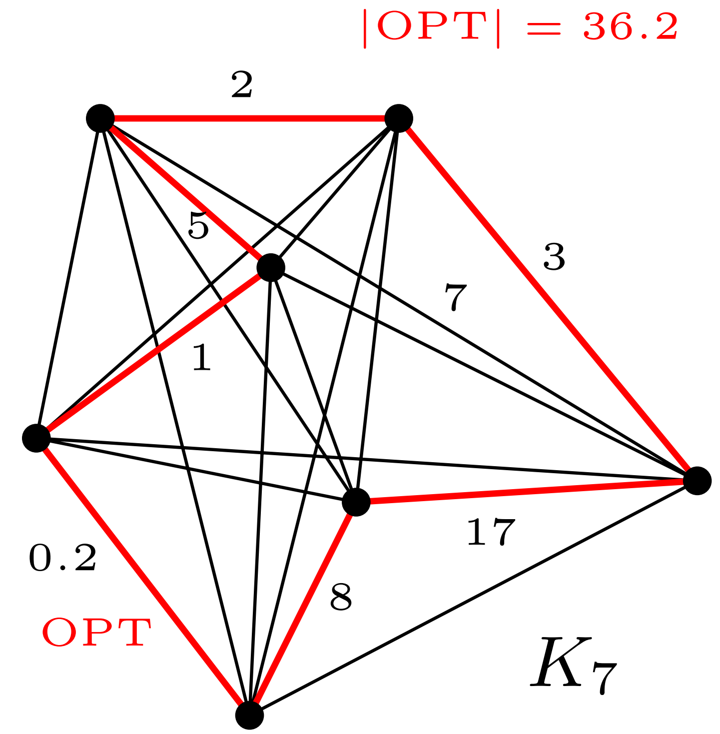
TSP

Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.



TSP

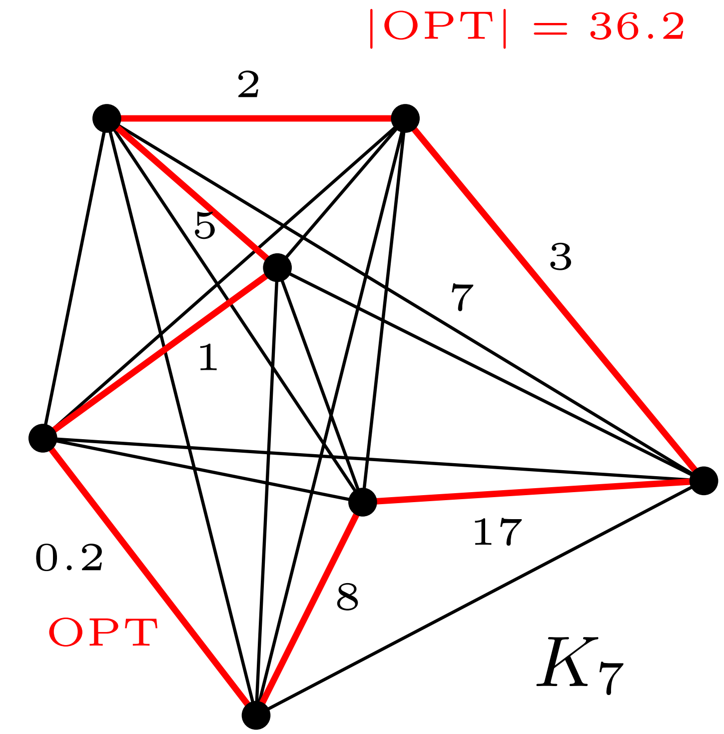
Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.



TSP

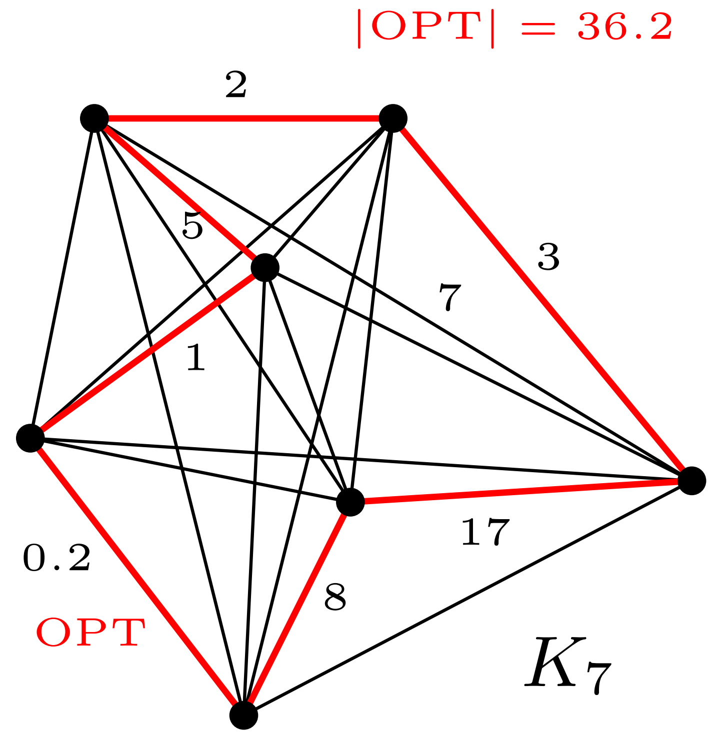
Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.

TSP is NP-hard $\Rightarrow$ no PT algorithm, unless $P = NP$.



TSP

Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.



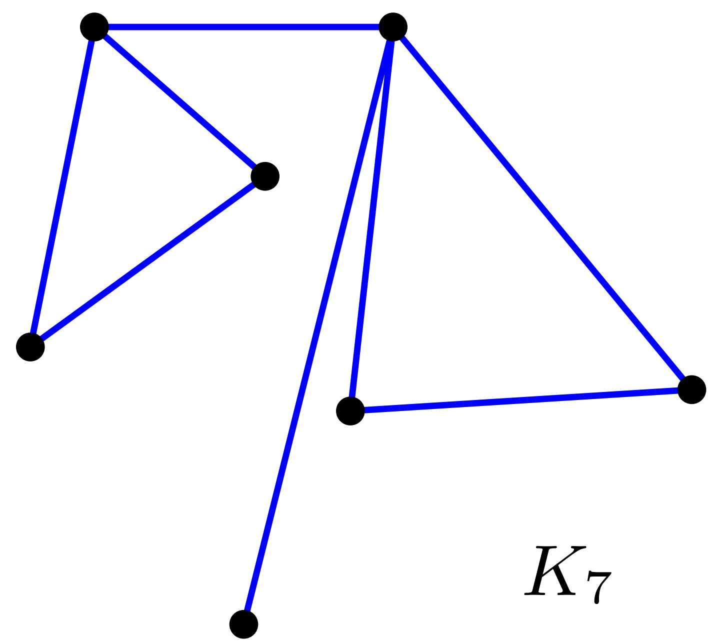
TSP is NP-hard $\Rightarrow$ no PT algorithm, unless $P = NP$.

$\Uparrow$

Thm For all PT computable function $\alpha(n)$, TSP cannot be approximated in PT within a factor of $(1 + \alpha(n))$, unless $P = NP$.

TSP

Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.



TSP is NP-hard $\Rightarrow$ no PT algorithm, unless $P = NP$.

$\Uparrow$

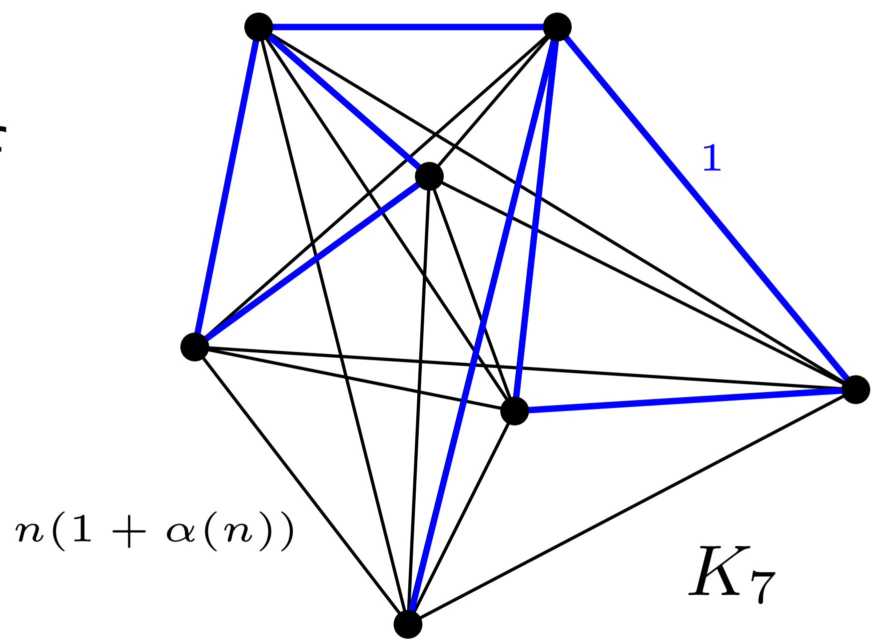
Thm For all PT computable function $\alpha(n)$, TSP cannot be approximated in PT within a factor of $(1 + \alpha(n))$, unless $P = NP$.

Proof Reduction of Hamiltonian Cycle:
Let $G = (V, E)$ unweighted, incomplete $\to G' = (V', E')$ where:

- $V' = V$
- $\forall e \in E$, add $(e, 1)$ to E'
- $\forall e \notin E$, add $(e, (1 + \alpha(n))n)$ to E'

TSP

Given a complete graph $G = (V, E)$ with non-negative weights, find the Hamiltonian tour of minimum total cost.



TSP is NP-hard $\Rightarrow$ no PT algorithm, unless $P = NP$.

$\Uparrow$

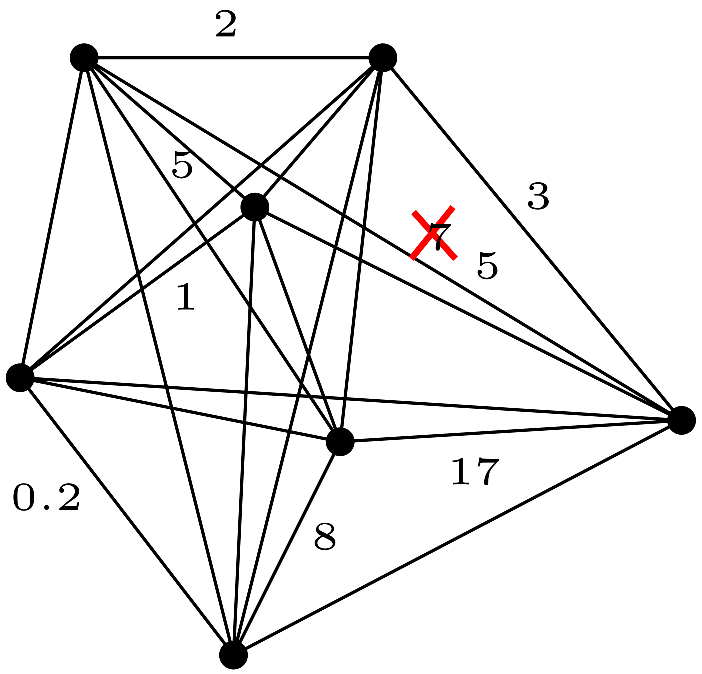
Thm For all PT computable function $\alpha(n)$, TSP cannot be approximated in PT within a factor of $(1 + \alpha(n))$, unless $P = NP$.

Proof Reduction of Hamiltonian Cycle:
Let $G = (V, E)$ unweighted, incomplete $\rightarrow G' = (V', E')$ where:

- $V' = V$
- $\forall e \in E$, add $(e, 1)$ to E'
- $\forall e \notin E$, add $(e, (1 + \alpha(n))n)$ to E'

Metric TSP

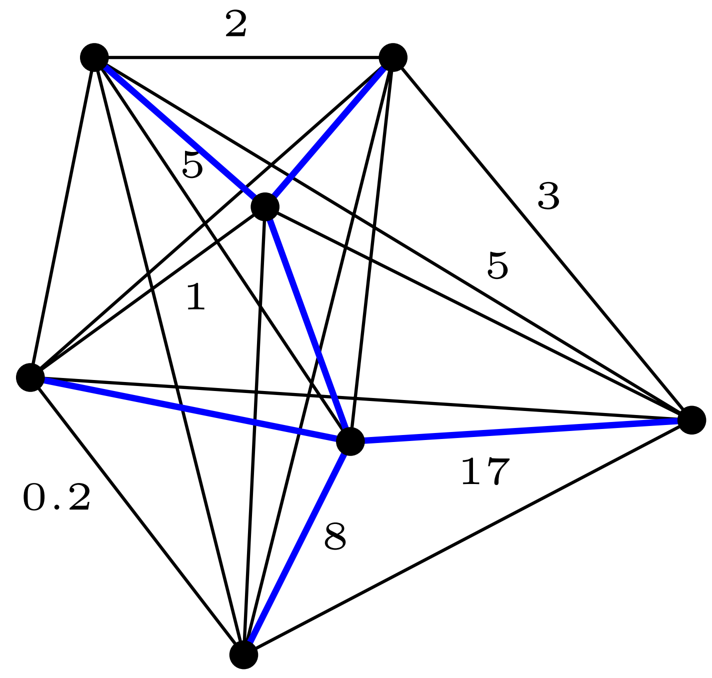
The weights of $G(V, E)$ now satisfy the triangle inequality



Metric TSP

2-approximation algorithm:

(1) build MST M of G (Kruskal)

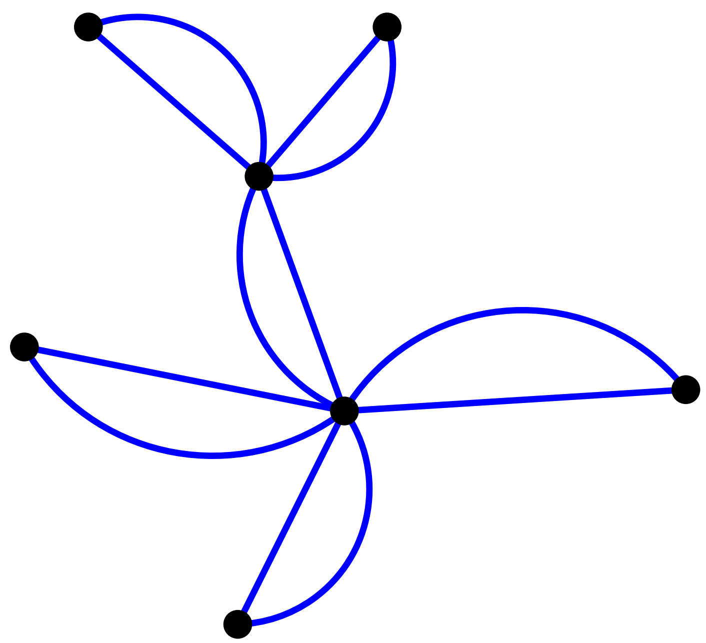


Metric TSP

2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian



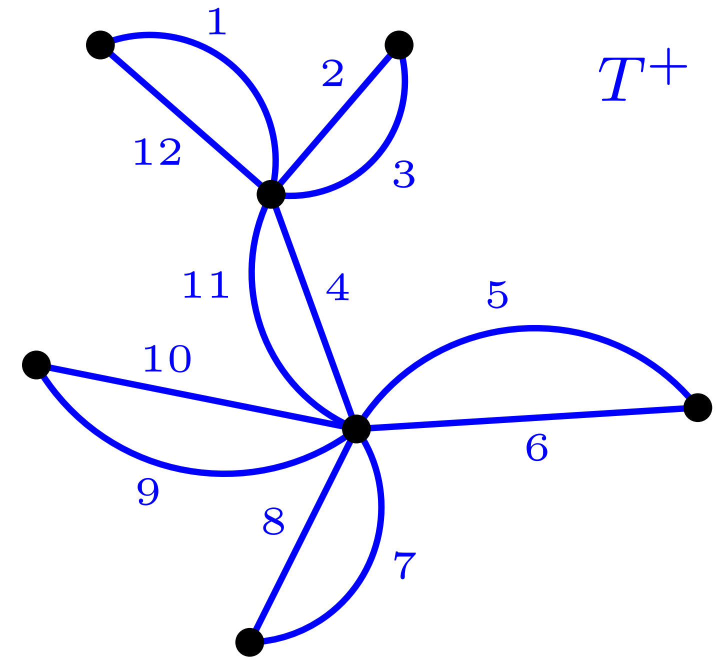
Metric TSP

2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+



Metric TSP

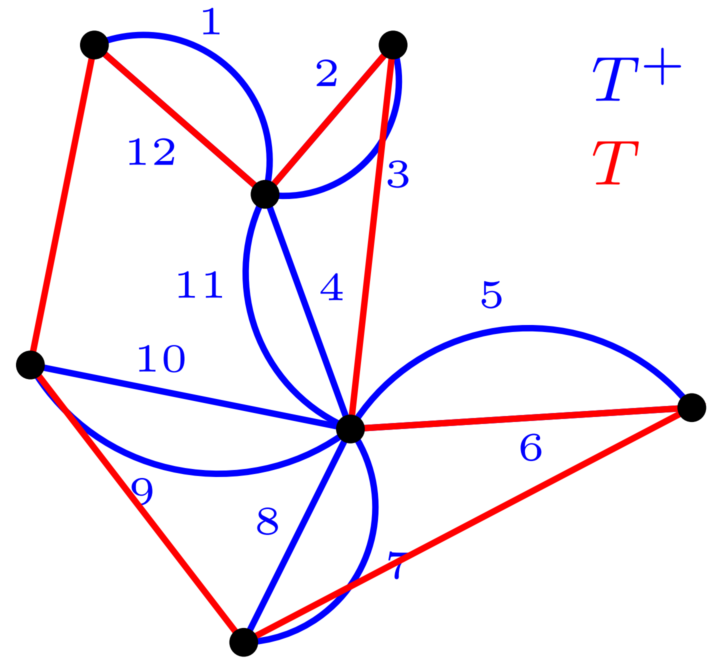
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Metric TSP

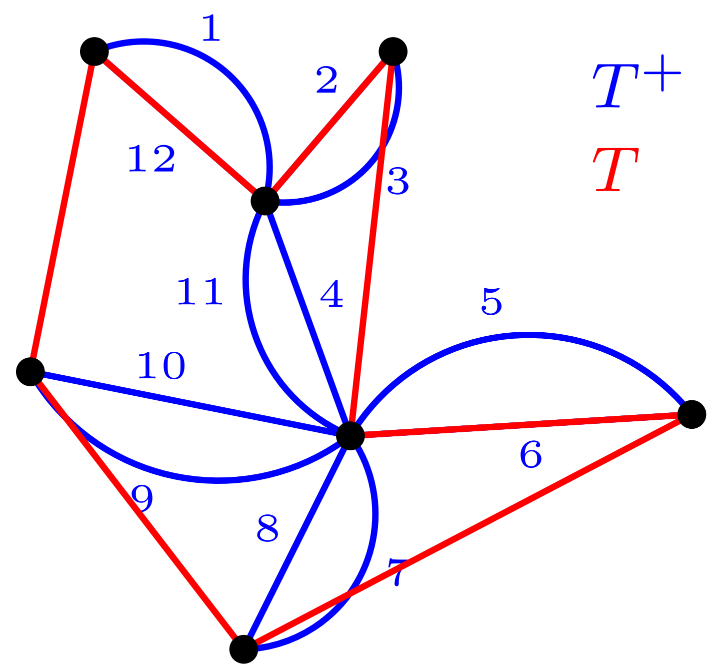
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Thm $|T| \leq 2|\mathrm{OPT}|$

proof $|T| \leq |T^+|$

tri. ineq.

Metric TSP

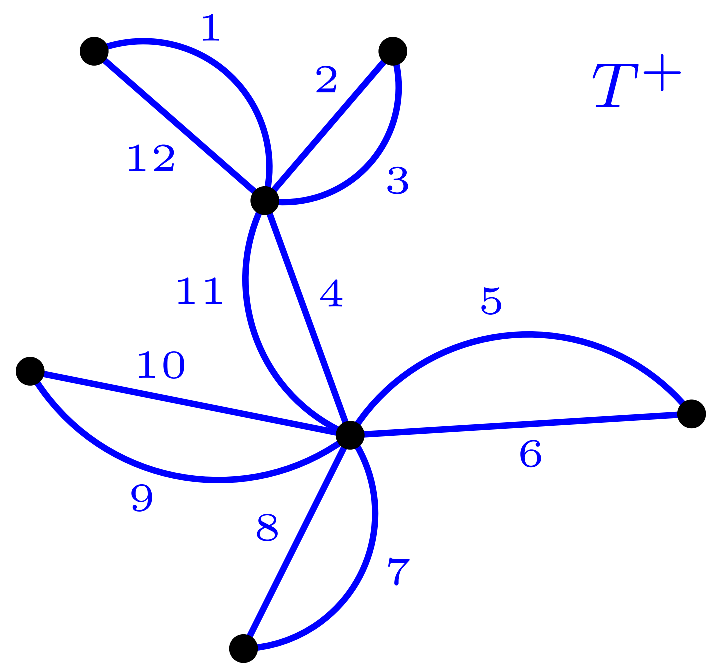
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Thm $|T| \leq 2|\mathrm{OPT}|$

proof $|T| \leq |T^+| = |M^+|$

tri. ineq.

Metric TSP

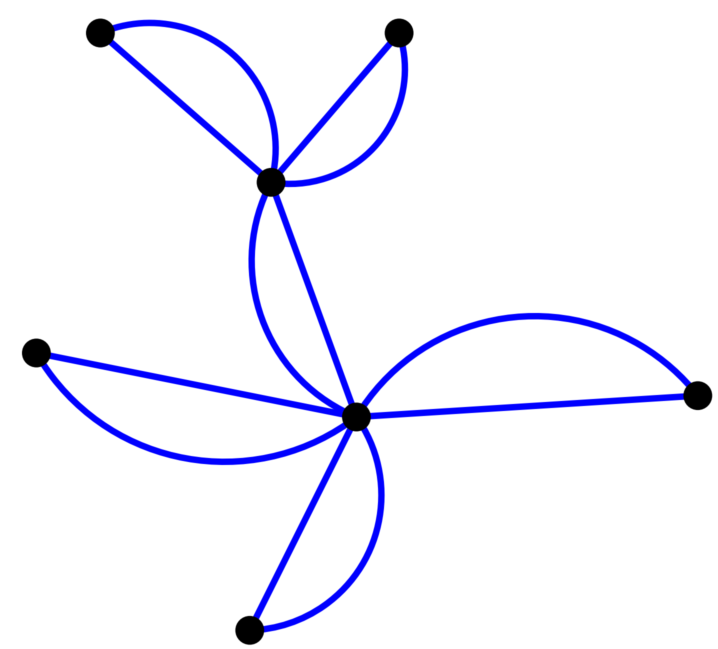
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Thm $|T| \leq 2|\mathrm{OPT}|$

proof $|T| \leq |T^+| = |M^+| = 2|M|$

tri. ineq.

Metric TSP

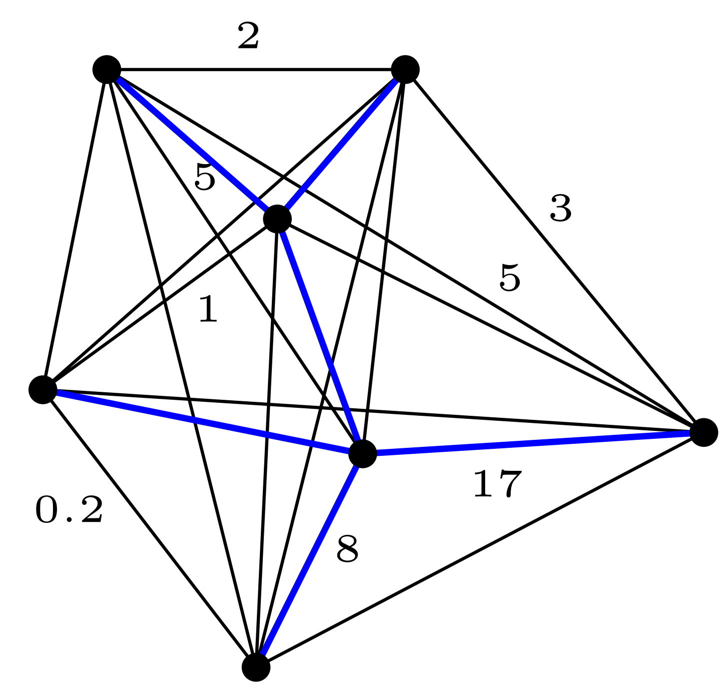
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^{+}$ Eulerian

(3) build greedily a Eulerian tour T^{+} on M^{+}

(4) Trim edges of $T^{+} \rightarrow T$



Thm $|T| \leq 2|\mathrm{OPT}|$

proof $|T| \leq |T^{+}| = |M^{+}| = 2|M| \leq 2|\mathrm{OPT}|$

tri. ineq. OPT="tree+edge"

Metric TSP

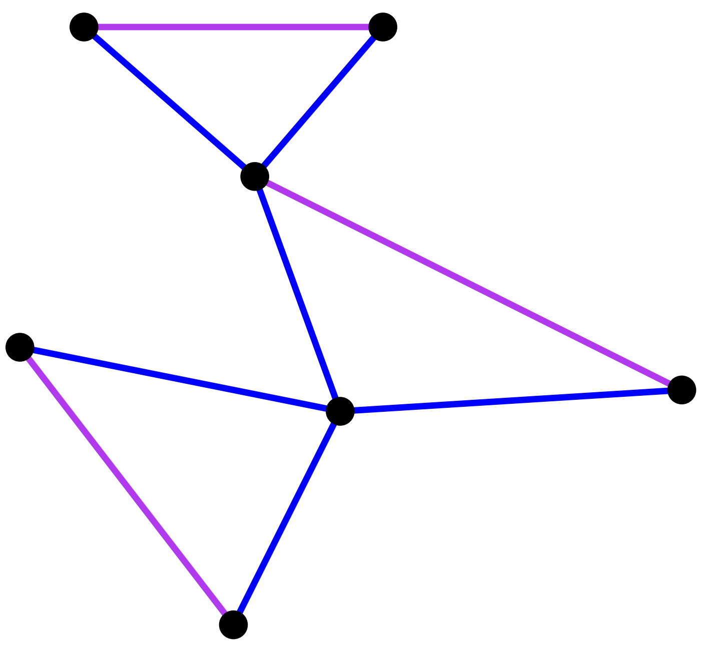
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Replace (2) by adding to M a min cost perfect matching of its odd-valenced vertices $\rightarrow \frac{3}{2}$-approximation [Christofides76]

Q Can we do better?

Metric TSP

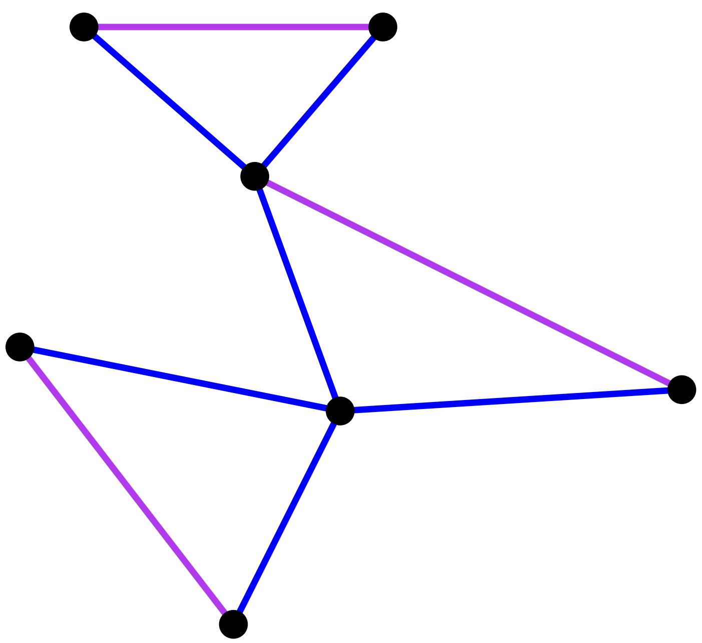
2-approximation algorithm:

(1) build MST M of G (Kruskal)

(2) double edges $\rightarrow M^+$ Eulerian

(3) build greedily a Eulerian tour T^+ on M^+

(4) Trim edges of $T^+ \rightarrow T$



Replace (2) by adding to M a min cost perfect matching of its odd-valenced vertices $\rightarrow \frac{3}{2}$-approximation [Christofides76]

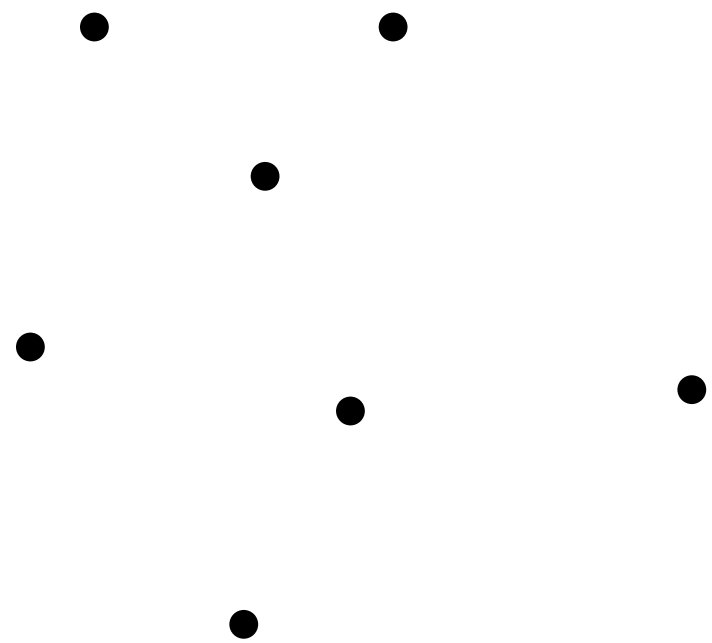
Q Can we do better?

Thm [ALMSS92] There is no PTAS for Metric TSP, unless $P = NP$

Conjecture best approximation factor: $4/3$

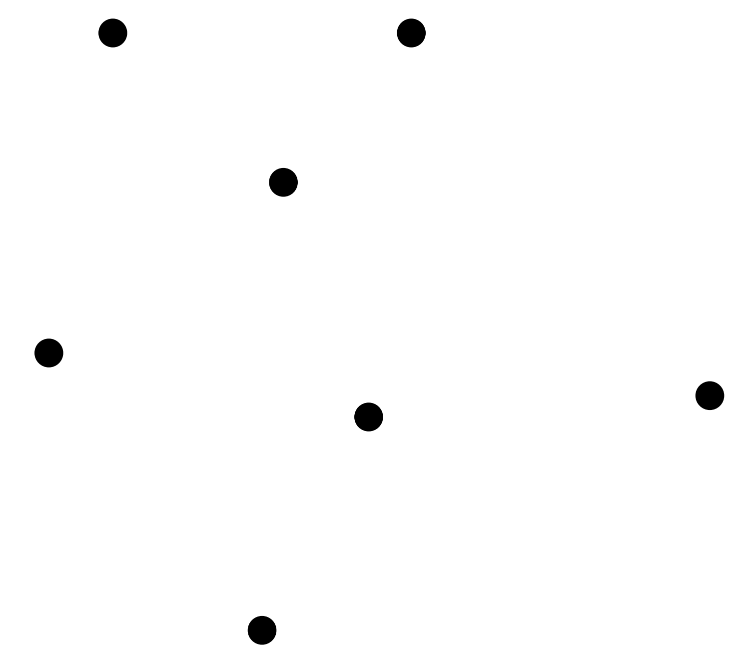
Euclidean TSP

$V \subset \mathbb{R}^d$, E is the set of all pairs weighted by Euclidean distances



Euclidean TSP

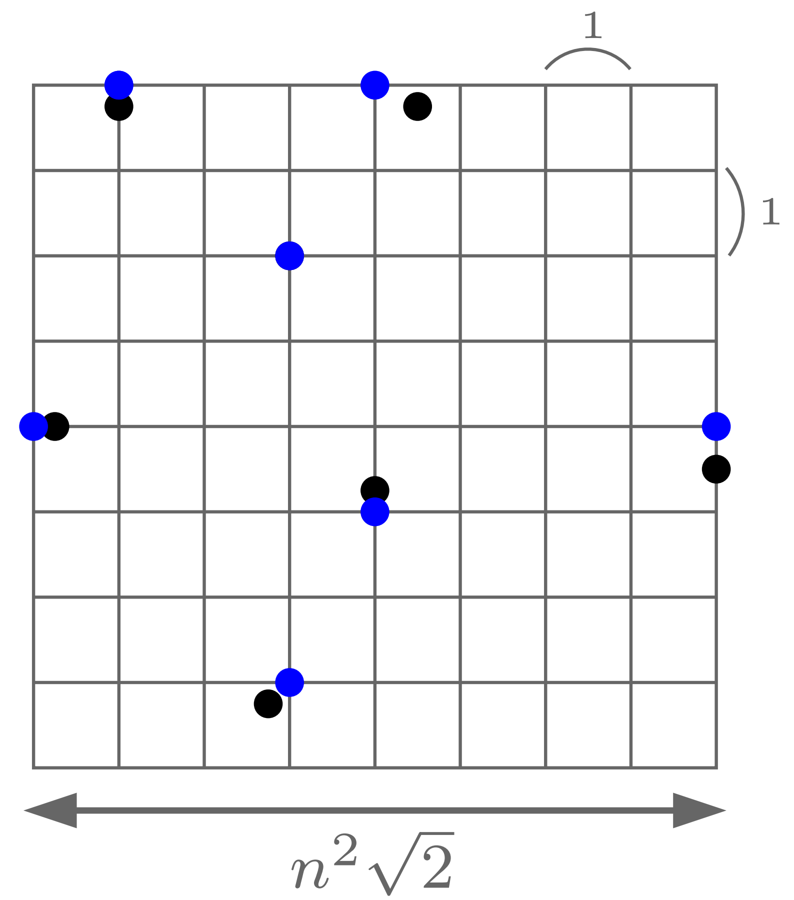
Thm [Arora96] Euclidean TSP admits a PTAS



Overview Let $n = |V|$

Euclidean TSP

Thm [Arora96] Euclidean TSP admits a PTAS



Overview Let $n = |V|$

(1) rescale/snap V

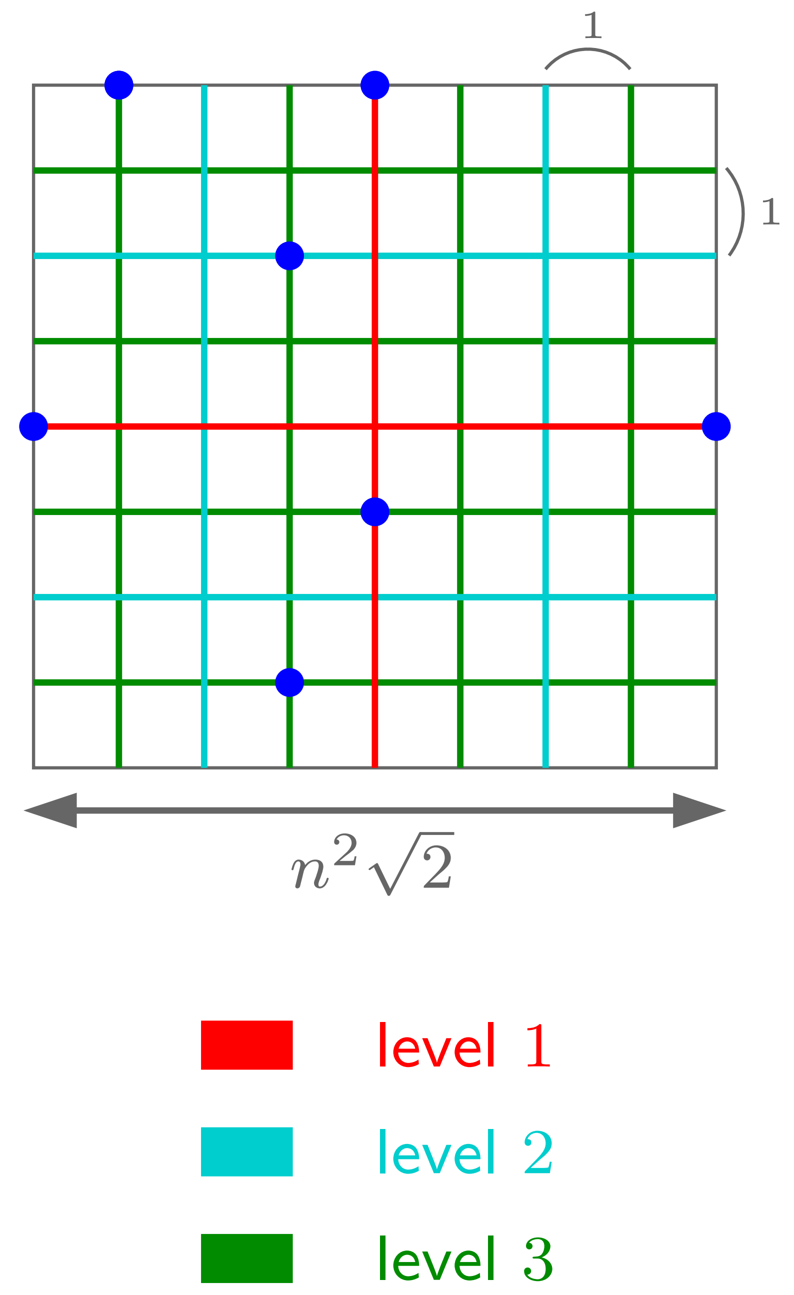
Euclidean TSP

Thm [Arora96] Euclidean TSP admits a PTAS

Overview Let $n = |V|$

(1) rescale/snap V

(2) subdivide the grid with a quadtree



Euclidean TSP

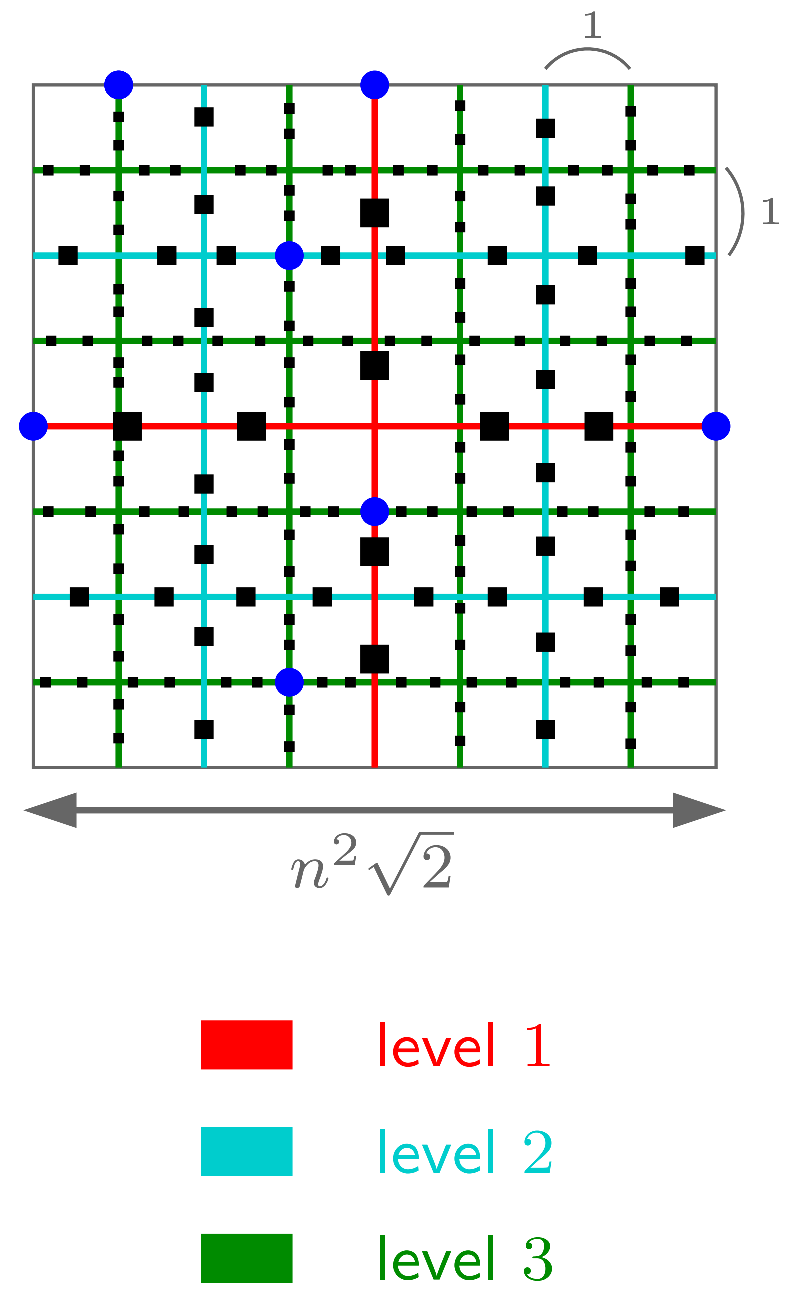
Thm [Arora96] Euclidean TSP admits a PTAS

Overview Let $n = |V|$

(1) rescale/snap V

(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines



Euclidean TSP

Thm [Arora96] Euclidean TSP admits a PTAS

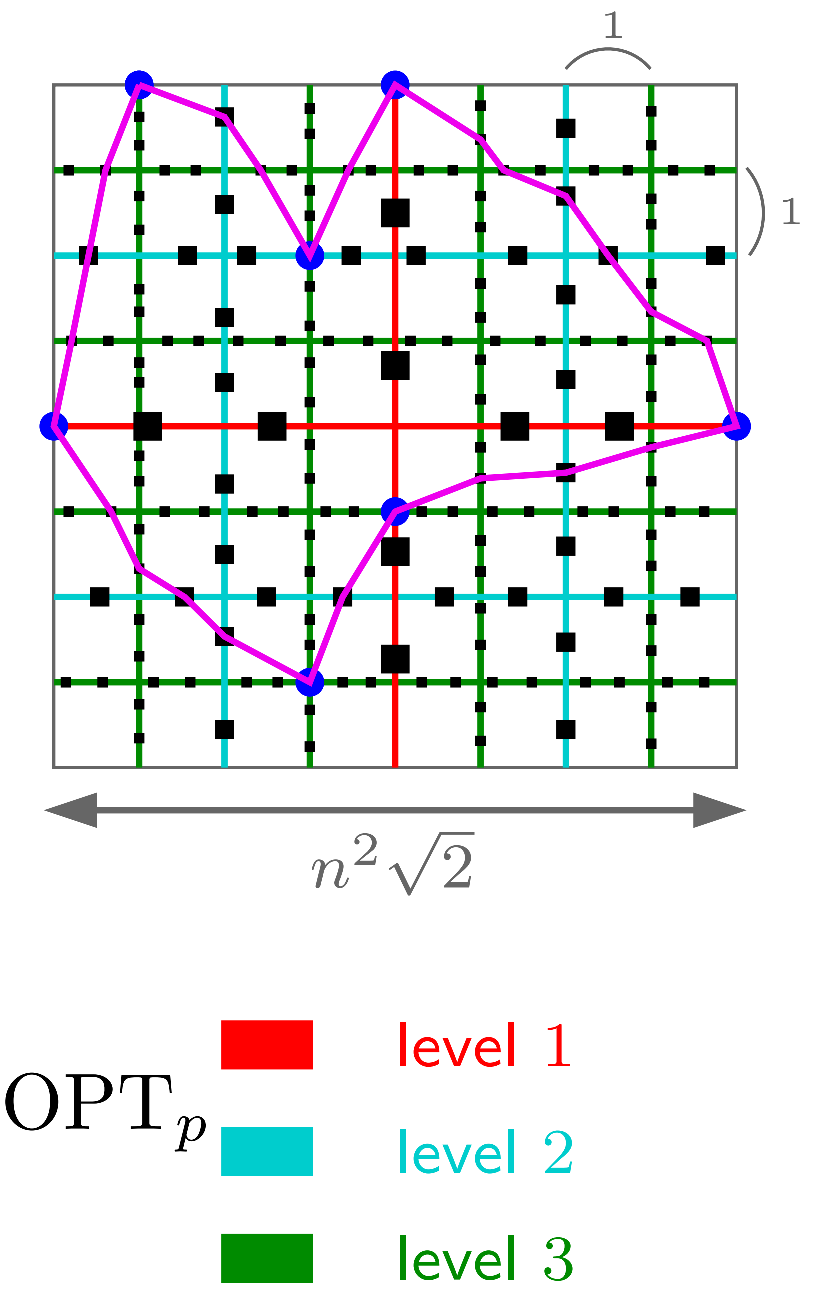
Overview Let $n = |V|$

(1) rescale/snap V

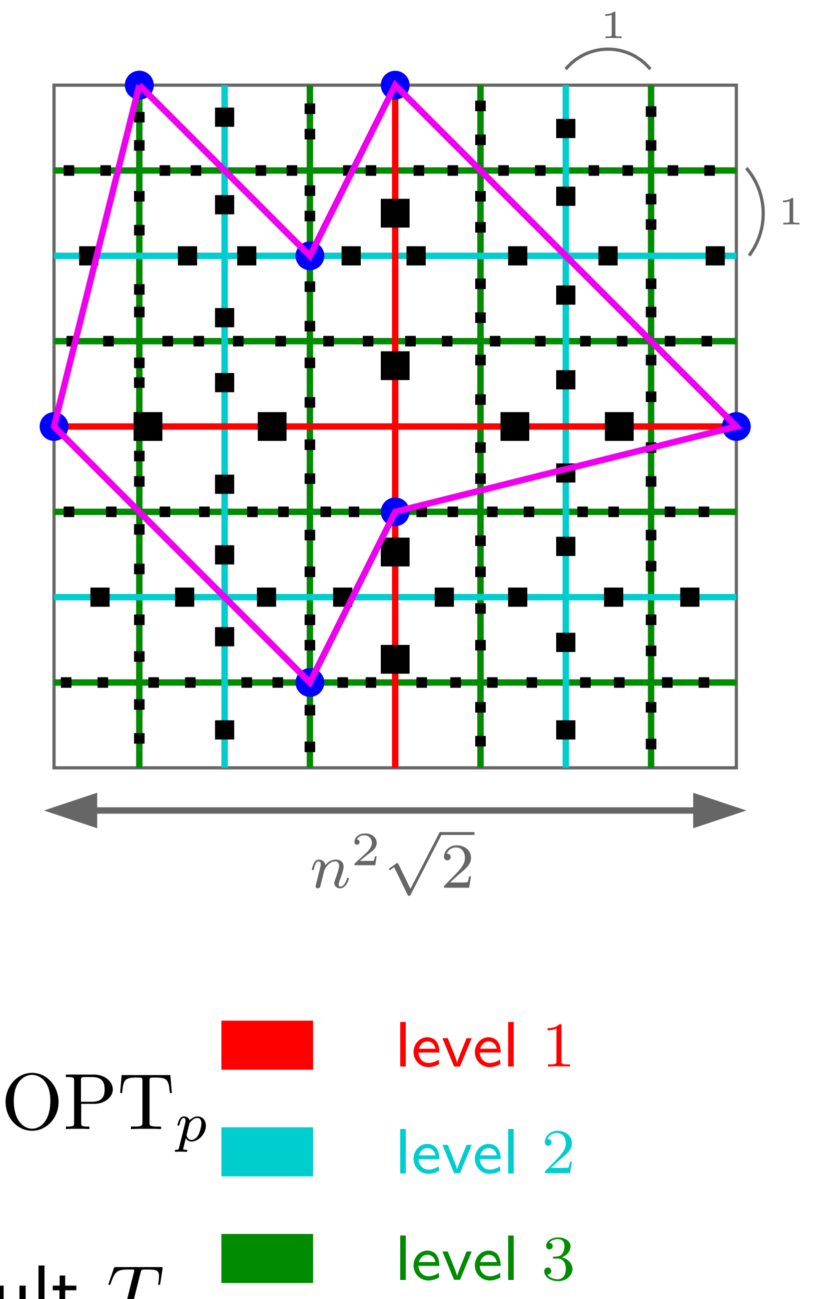
(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p



Euclidean TSP



Thm [Arora96]   Euclidean TSP admits a PTAS

Overview   Let $n = |V|$

(1) rescale/snap V

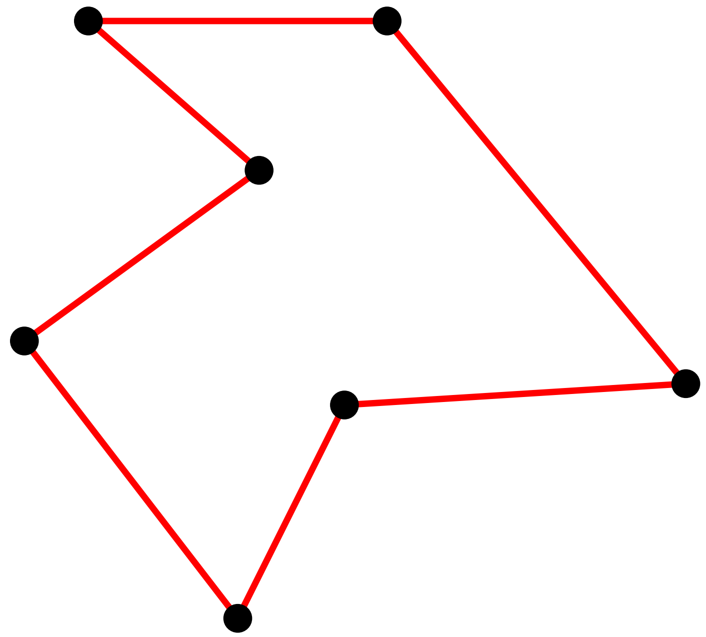
(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p

(5) Trim the edges of OPT_p and output the result T

(1) rescale V



Let V_s be V scaled by a factor of s.

$$\forall T,\ |T|_s = s\ |T|$$

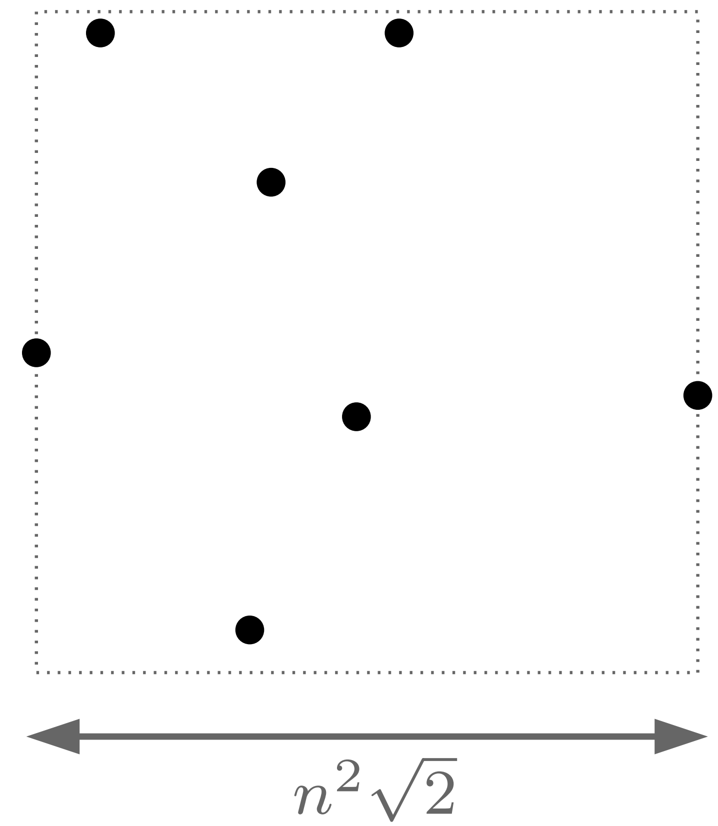
$\Rightarrow$ OPT for V_s is the same as OPT for V

$\Rightarrow$ solving the pb for V_s is the same as solving the pb for V

(1) rescale V

Let V_s be V scaled by a factor of s.

$$\forall T,\ |T|_s = s\ |T|$$



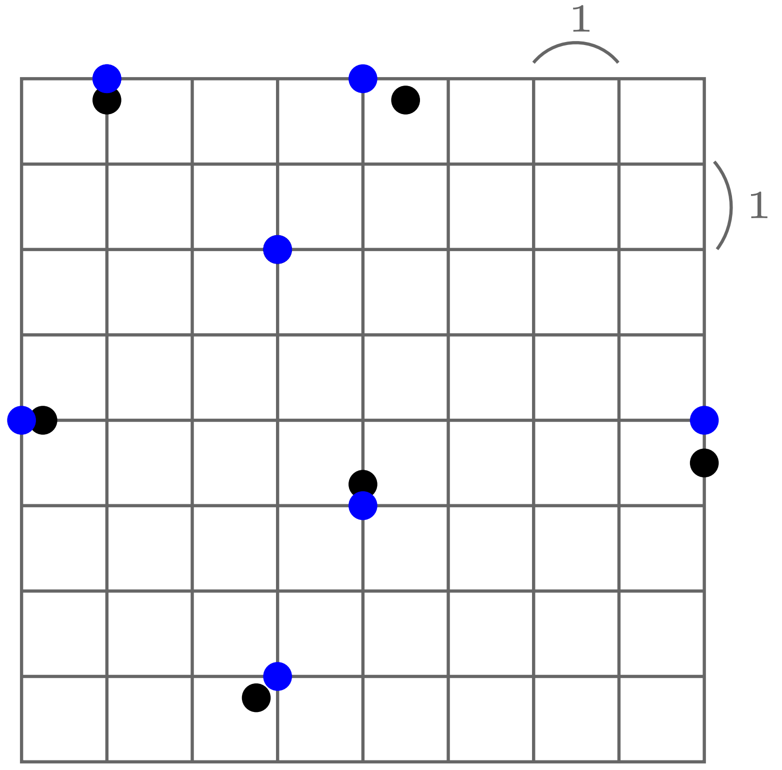
$\Rightarrow$ OPT for V_s is the same as OPT for V

$\Rightarrow$ solving the pb for V_s is the same as solving the pb for V

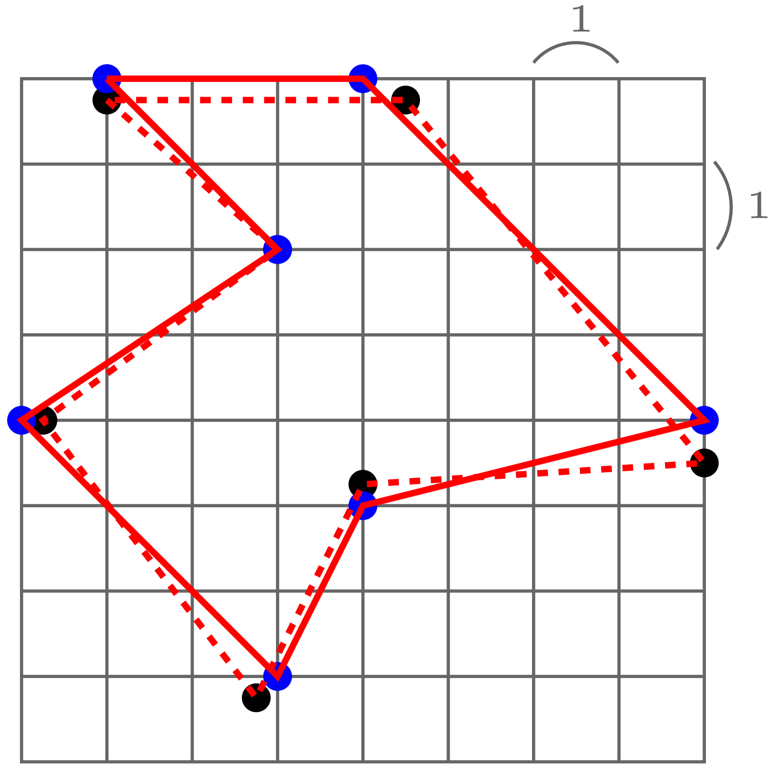
$\rightarrow$ wlog, we assume that the smallest square containing V has sidelength $n^2\sqrt{2}$

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v



(1) snap V



$g : v \in V \mapsto v_g \in$ grid closest to v

$\forall T = (v_1, v_2, \cdots, v_n)$, $g(T) := (g(v_1), g(v_2), \cdots, g(v_n))$

Through g, a vertex is moved by at most $\sqrt{2}/2$

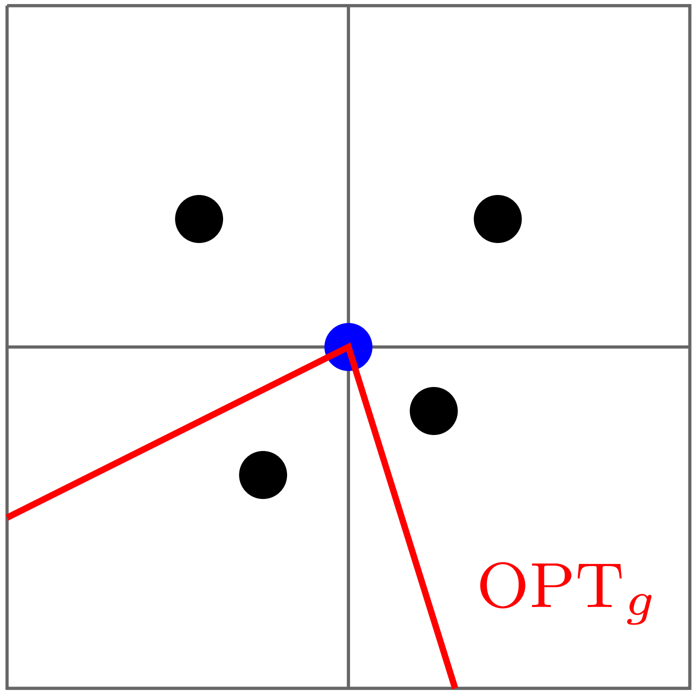
$\Rightarrow$ an edge is elongated/shortened by at most $\sqrt{2}$

$\Rightarrow \forall T$, $||g(T)| - |T|| \leq n\sqrt{2}$

$\Rightarrow |\mathrm{OPT}_g| \leq |g(\mathrm{OPT})| \leq |\mathrm{OPT}| + n\sqrt{2}$

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v

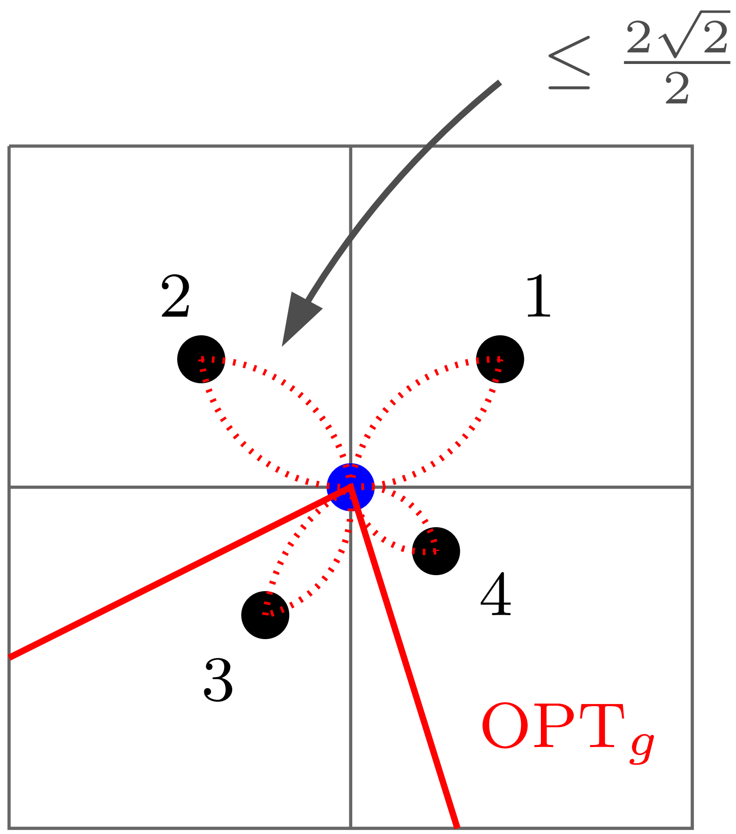


Q How to construct a path for V from OPT_g?

$g^{-1}(\mathrm{OPT}_g)$ is not defined uniquely
(several nodes of V may be mapped to a same grid point)

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v



Q How to construct a path for V from OPT_g?

$g^{-1}(\mathrm{OPT}_g)$ is not defined uniquely
(several nodes of V may be mapped to a same grid point)

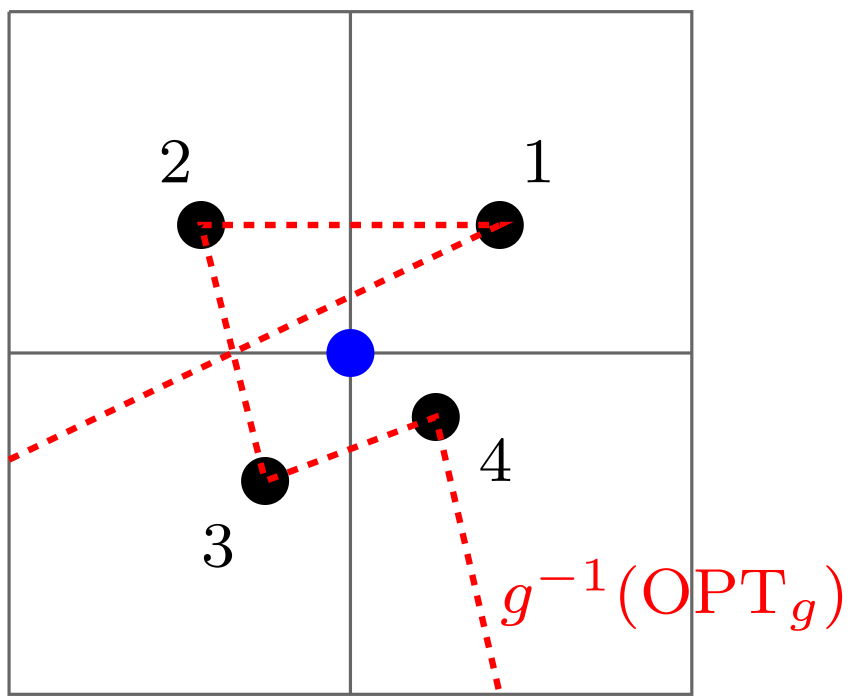
$\longrightarrow$ Define $g^{-1}(\mathrm{OPT}_g)$ as follows: for each vertex v_g of OPT_g,

- order the vertices of V mapped to v_g and connect them to v_g twice

$(+n\sqrt{2})$

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v



Q How to construct a path for V from OPT_g?

$g^{-1}(\mathrm{OPT}_g)$ is not defined uniquely

(several nodes of V may be mapped to a same grid point)

$\rightarrow$ Define $g^{-1}(\mathrm{OPT}_g)$ as follows: for each vertex v_g of OPT_g,

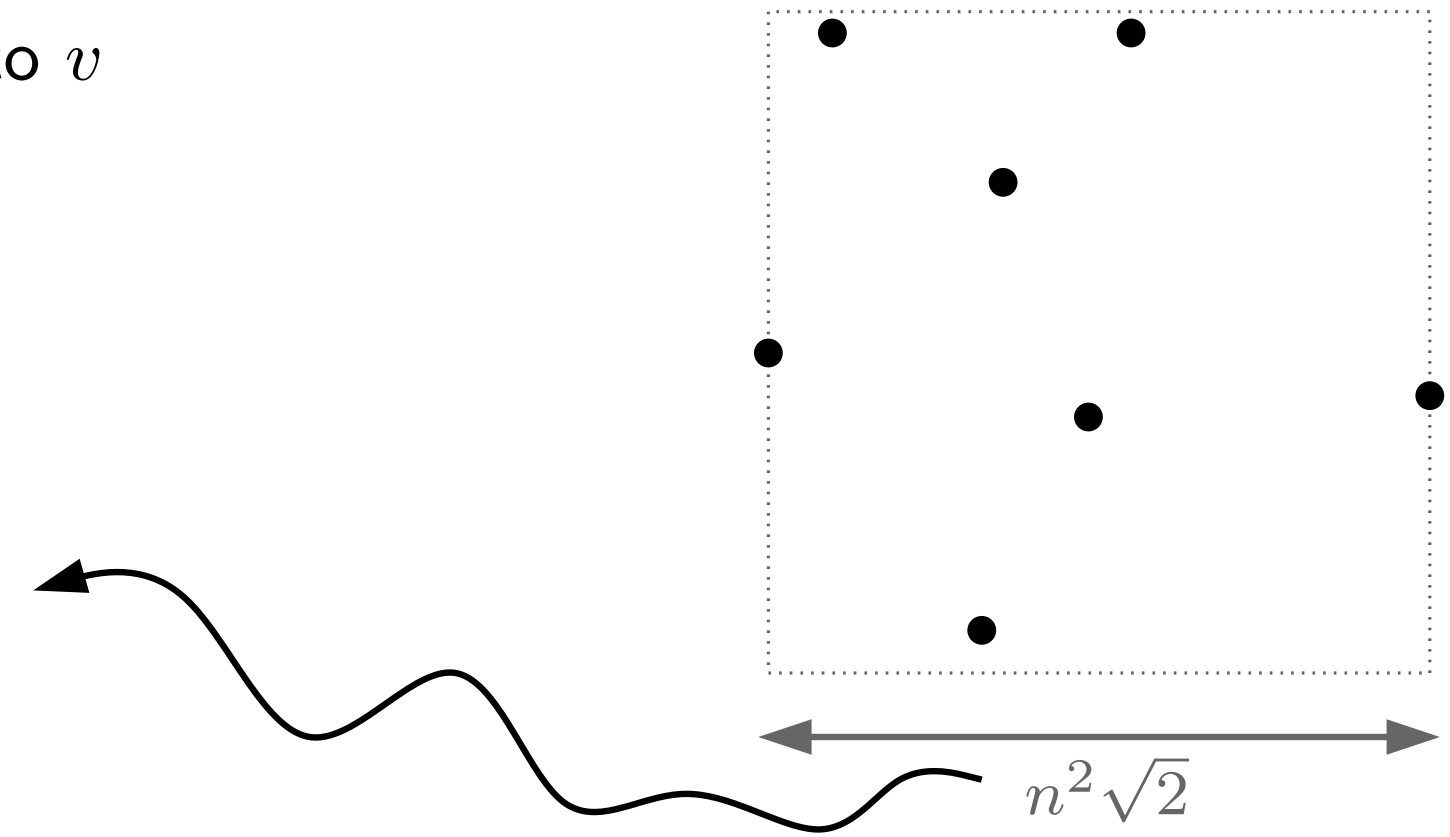
- order the vertices of V mapped to v_g and connect them to v_g twice

$(+n\sqrt{2})$

- trim the resulting path

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v



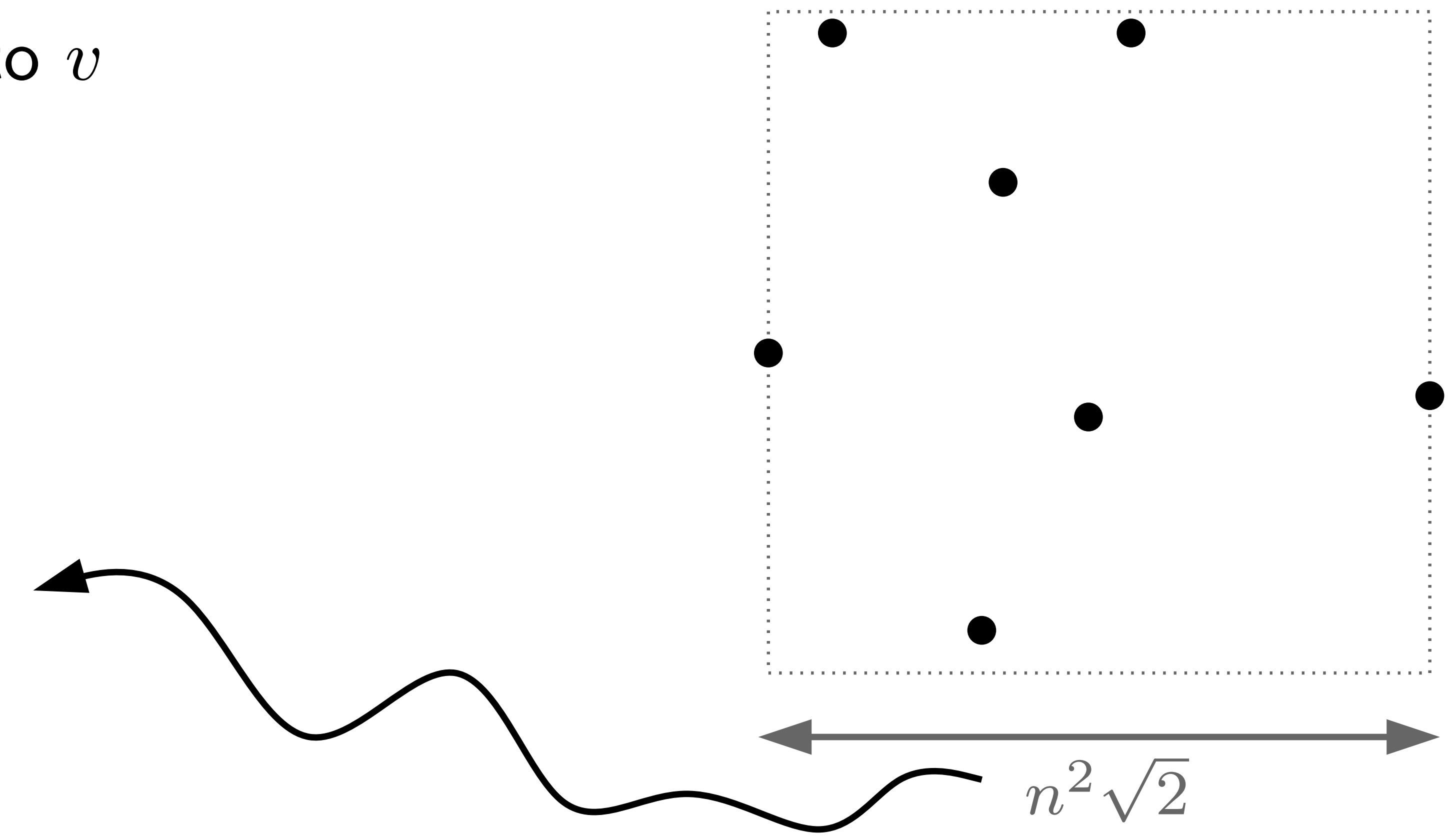
$|\mathrm{OPT}| \geq 2n^2\sqrt{2}$

$$|g^{-1}(\mathrm{OPT}_g)| \leq |\mathrm{OPT}_g| + n\sqrt{2} \leq |g(\mathrm{OPT})| + n\sqrt{2} \leq |\mathrm{OPT}| + 2n\sqrt{2}$$
$$\leq |\mathrm{OPT}| \left(1 + \tfrac{1}{n}\right)$$

$\rightarrow g^{-1}(\mathrm{OPT}_g)$ $(1+\varepsilon)$-approximates OPT for $n \geq \frac{1}{\varepsilon}$

(1) snap V

$g : v \in V \mapsto v_g \in$ grid closest to v



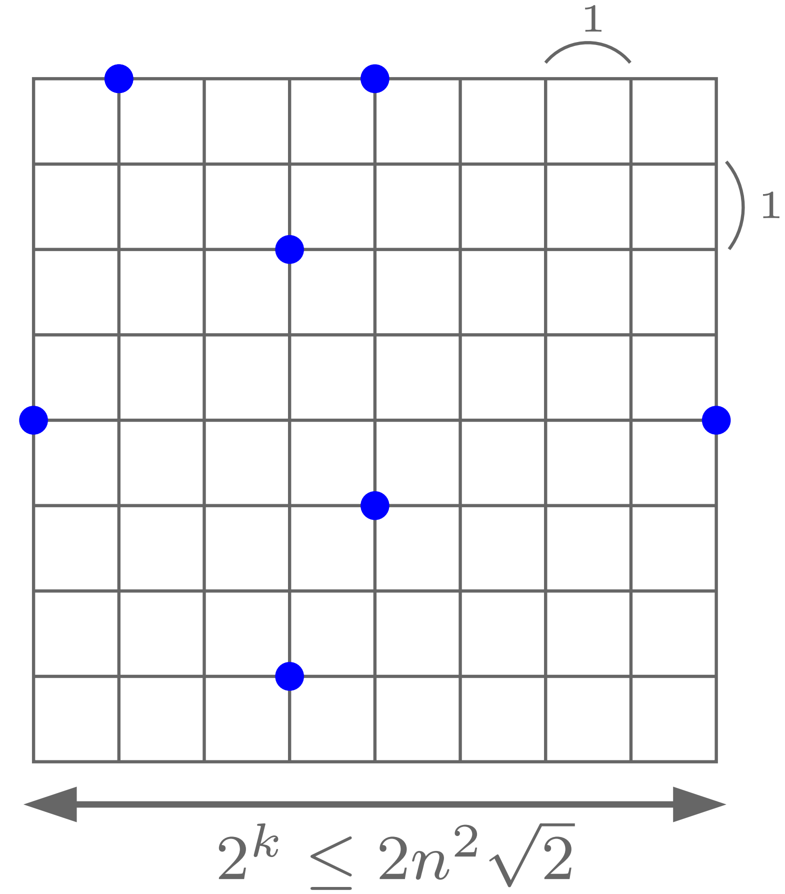
$|\mathrm{OPT}| \geq 2n^2\sqrt{2}$

$$|g^{-1}(\mathrm{OPT}_g)| \leq |\mathrm{OPT}_g| + n\sqrt{2} \leq |g(\mathrm{OPT})| + n\sqrt{2} \leq |\mathrm{OPT}| + 2n\sqrt{2}$$
$$\leq |\mathrm{OPT}| \left(1 + \frac{1}{n}\right)$$

$\rightarrow$ wlog, we assume that the points of V have integer coordinates

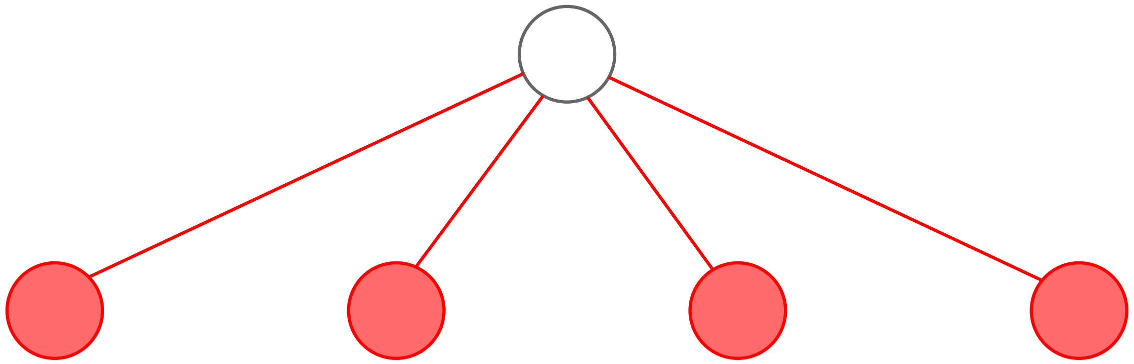
(2) Grid subdivision

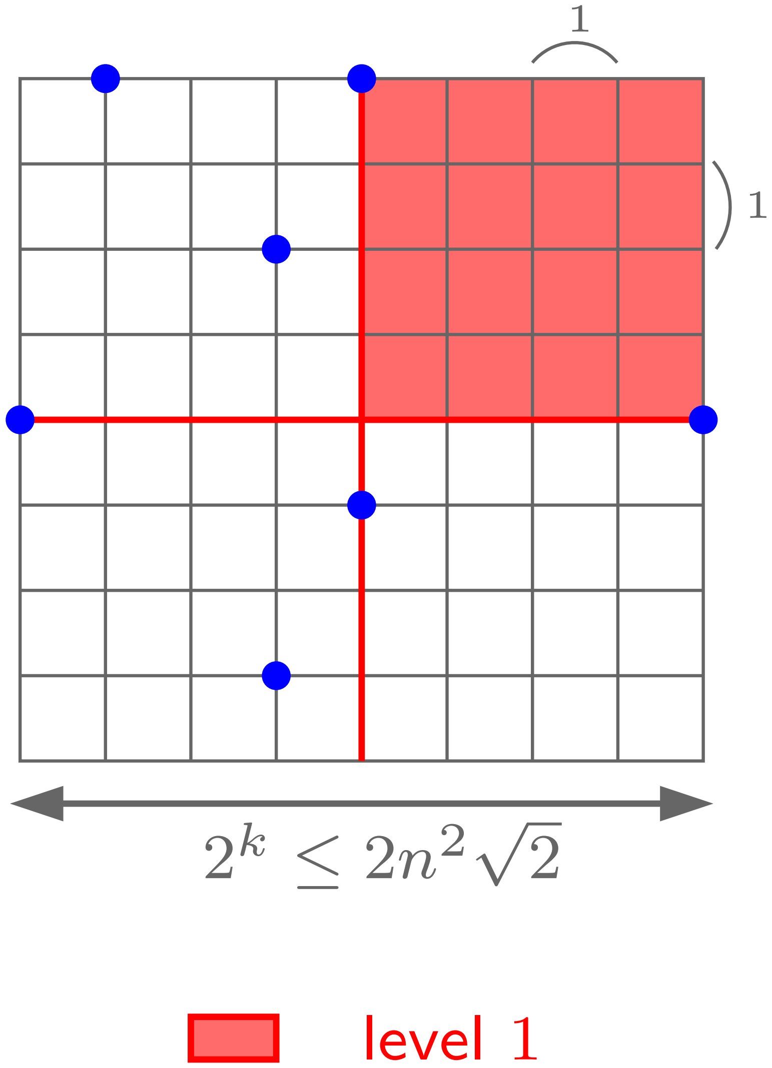
Let k s.t. $2^{k-1} \leq n^2\sqrt{2} \leq 2^k \leq 2n^2\sqrt{2}$



(2) Grid subdivision

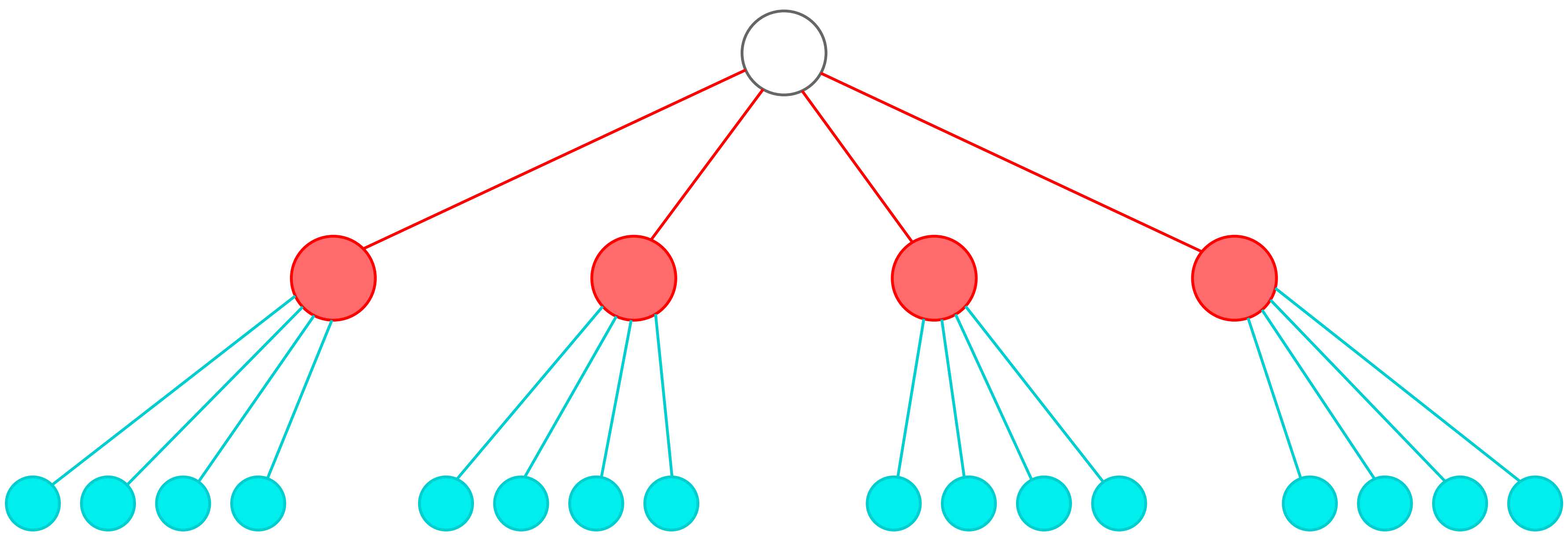
Let k s.t. $2^{k-1} \leq n^2\sqrt{2} \leq 2^k \leq 2n^2\sqrt{2}$

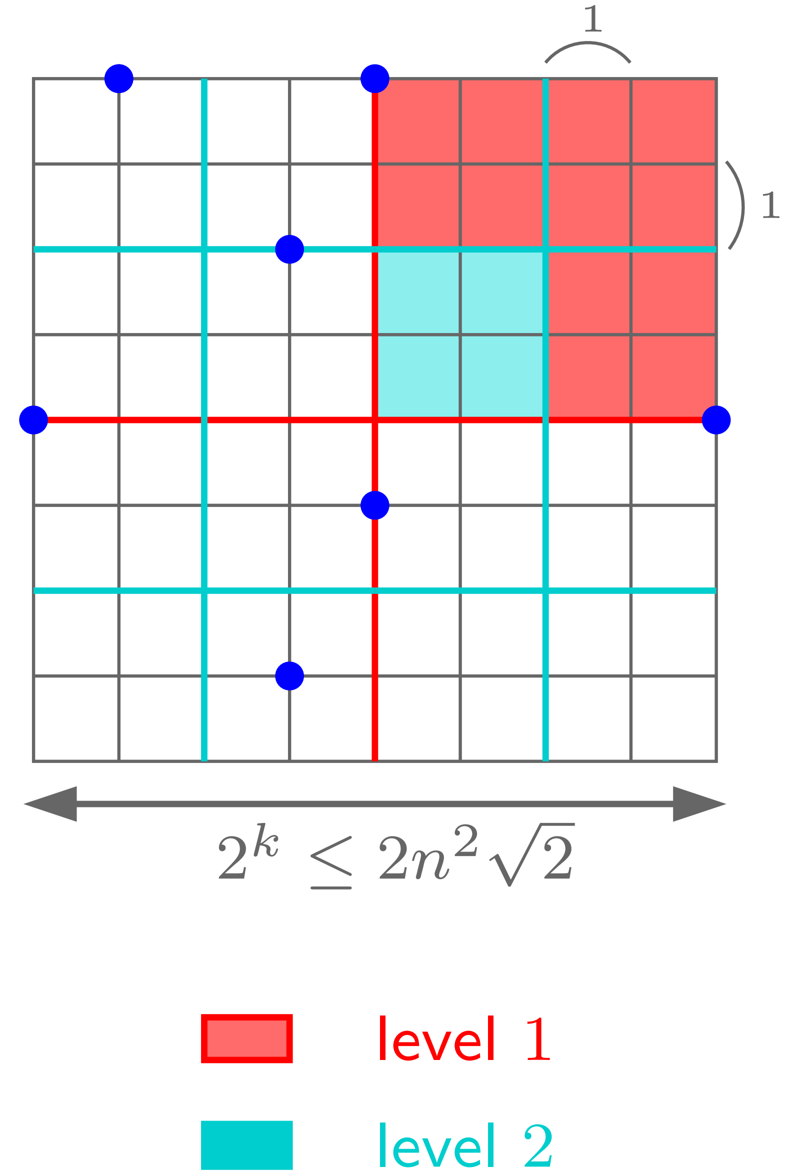




(2) Grid subdivision

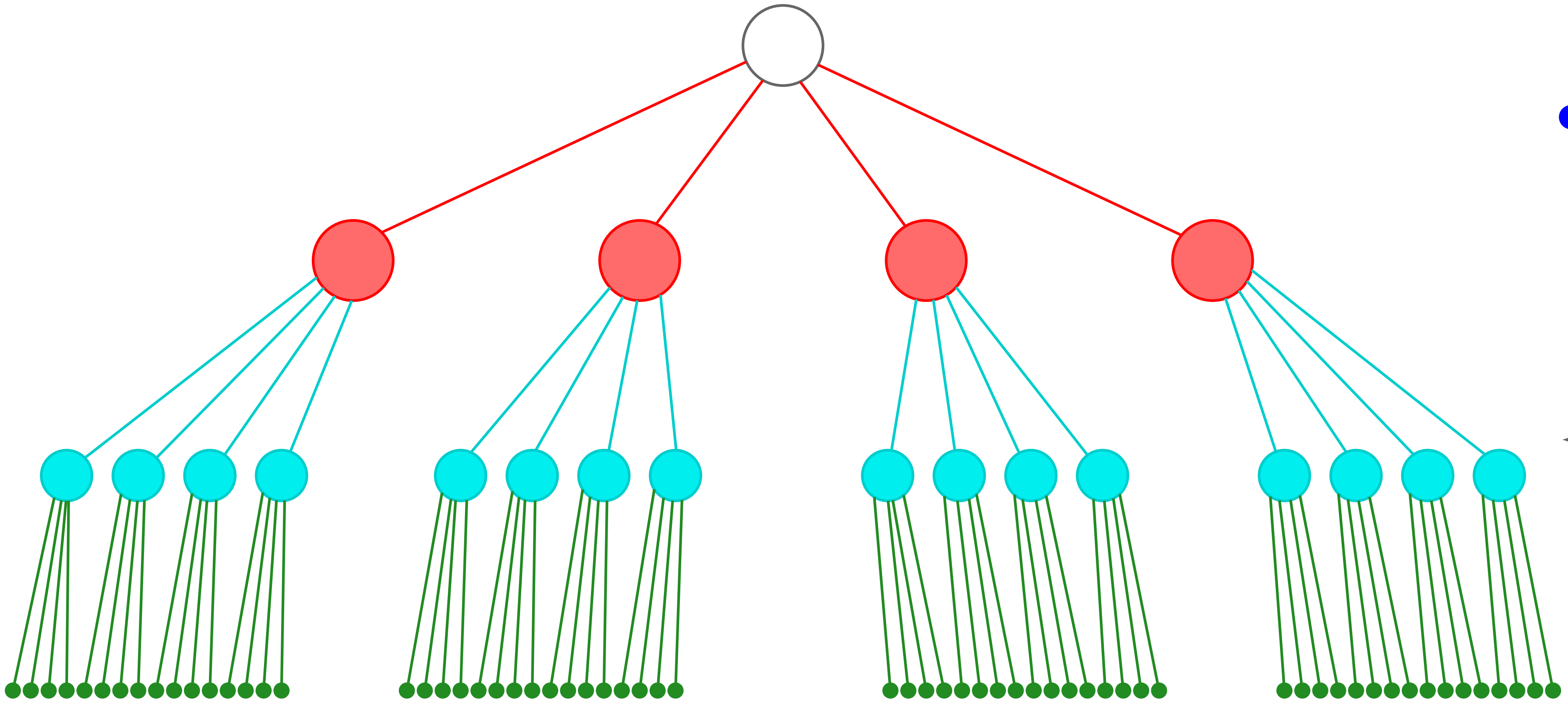
Let k s.t. $2^{k-1} \leq n^2\sqrt{2} \leq 2^k \leq 2n^2\sqrt{2}$

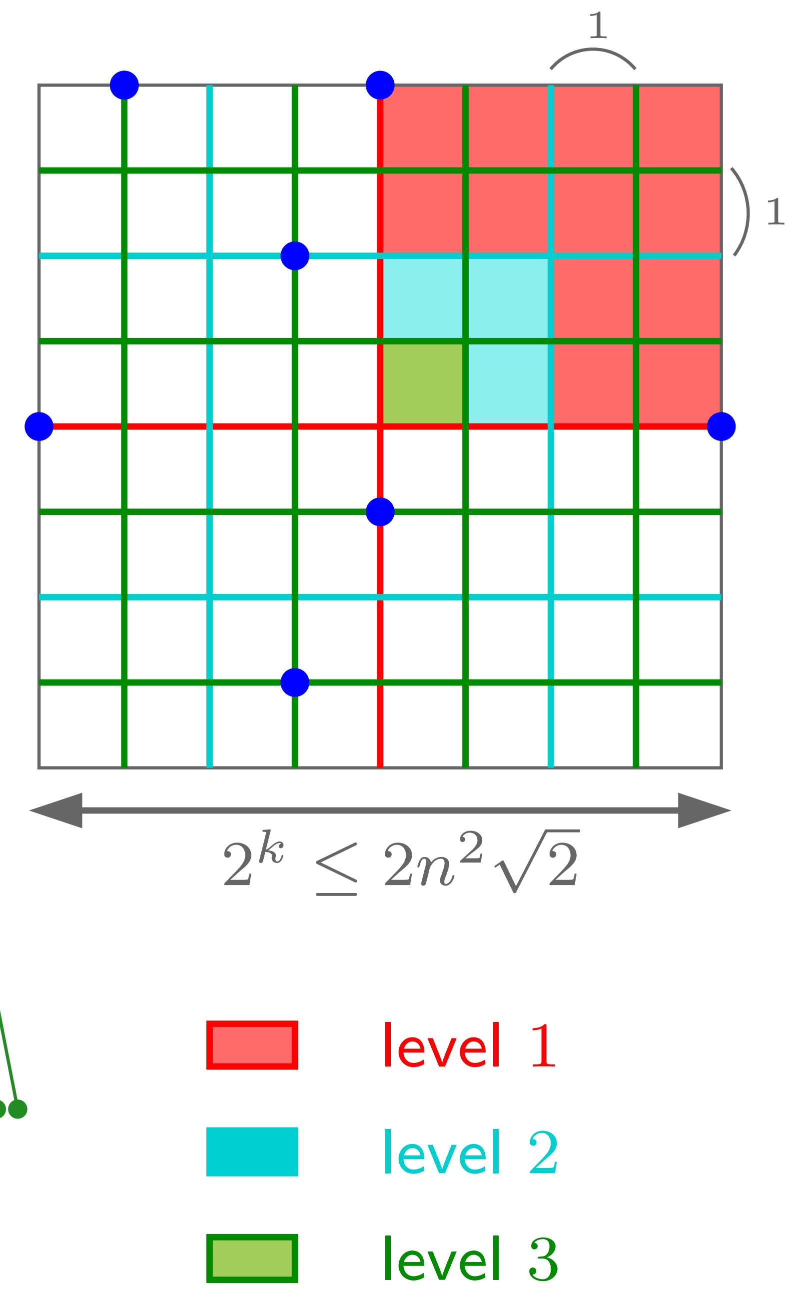




(2) Grid subdivision

Let k s.t. $2^{k-1} \leq n^2\sqrt{2} \leq 2^k \leq 2n^2\sqrt{2}$



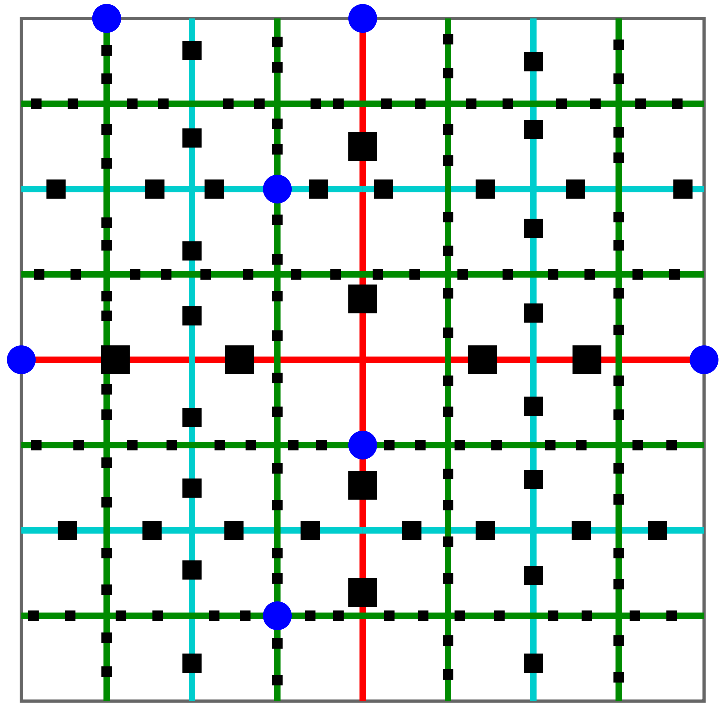


$O(n^4)$ leaves $\Rightarrow$ size $= O(n^4)$

(3) Portals

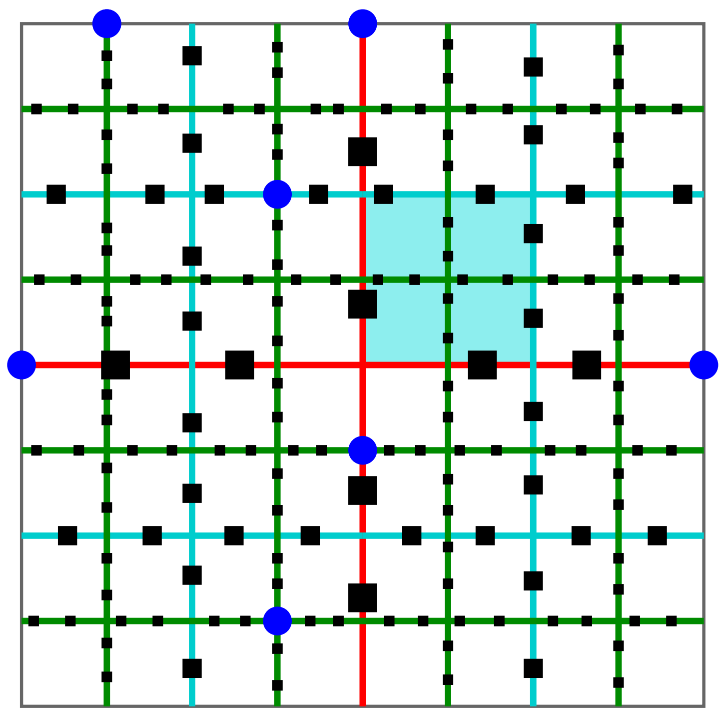
Let $m = \left\lfloor \frac{\log n}{\varepsilon} \right\rfloor$

On each level i line, place $2^i m$ equally-spaced portals, plus one at each grid point



(3) Portals

Let $m = \left\lfloor \frac{\log n}{\varepsilon} \right\rfloor$



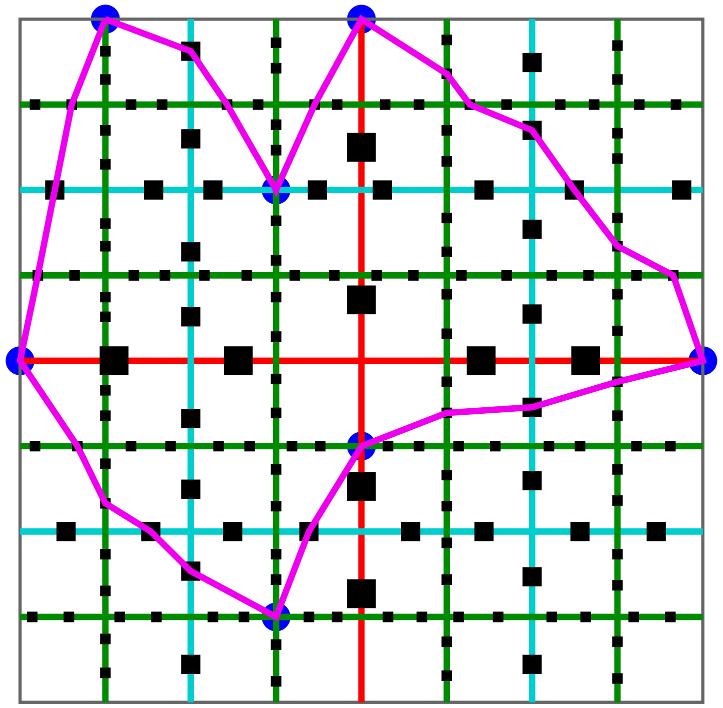
On each level i line, place $2^i m$ equally-spaced portals, plus one at each grid point

Each level i line is incident to 2^i pairs of level i squares $\Rightarrow$ m portals per pair (w/o corners)

Each level i square has a boundary made of level $j \leq i$ lines
$\Rightarrow$ at most $4m + 4$ portals per square

(4) Portal-respecting tours

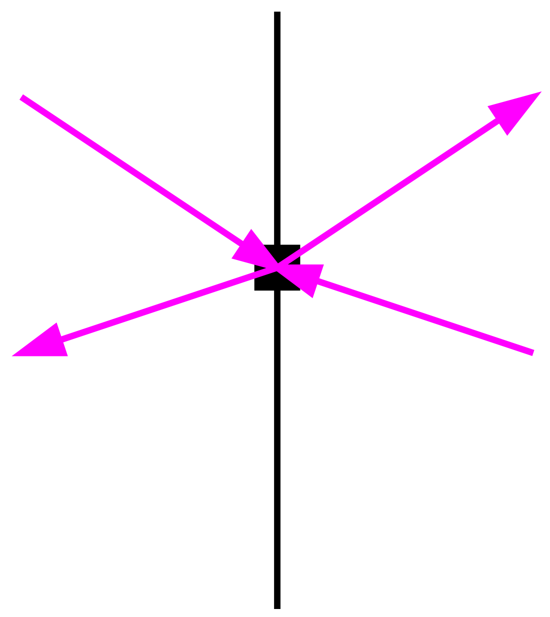
Def A tour is *portal-respecting* if it crosses the grid only at portals



Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals

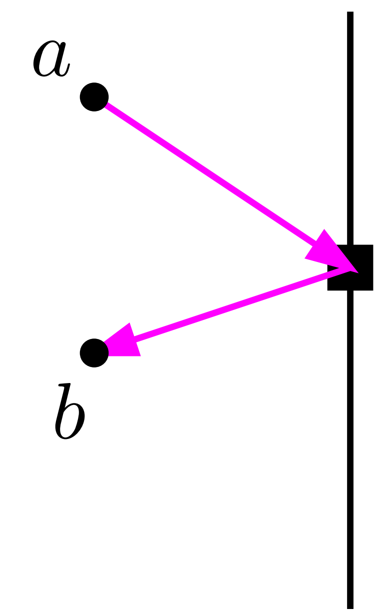


Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k-*light* if each portal is visited at most k times

(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals



Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

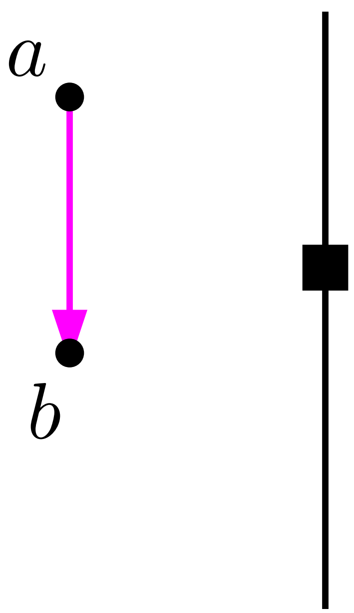
Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light

(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals

Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded



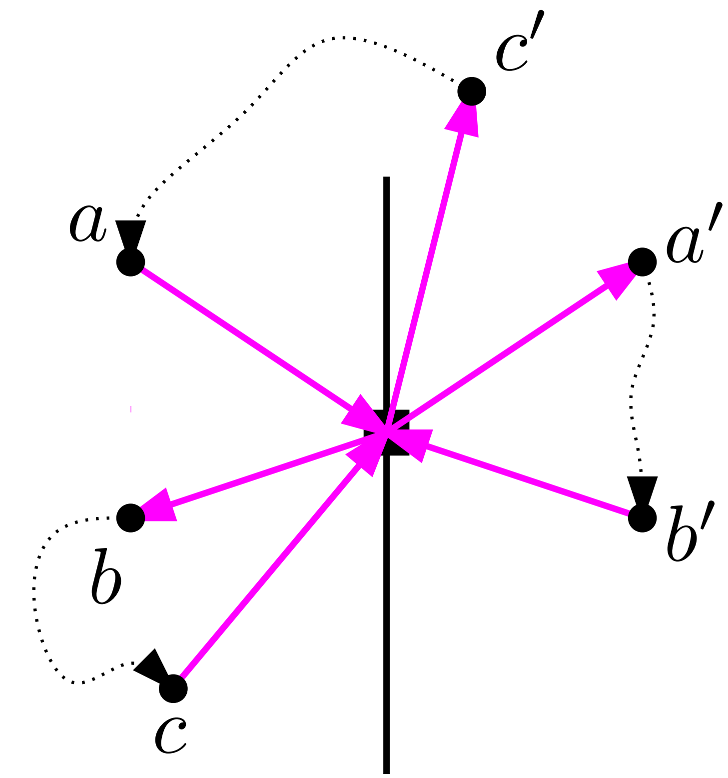
Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light

(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals

Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

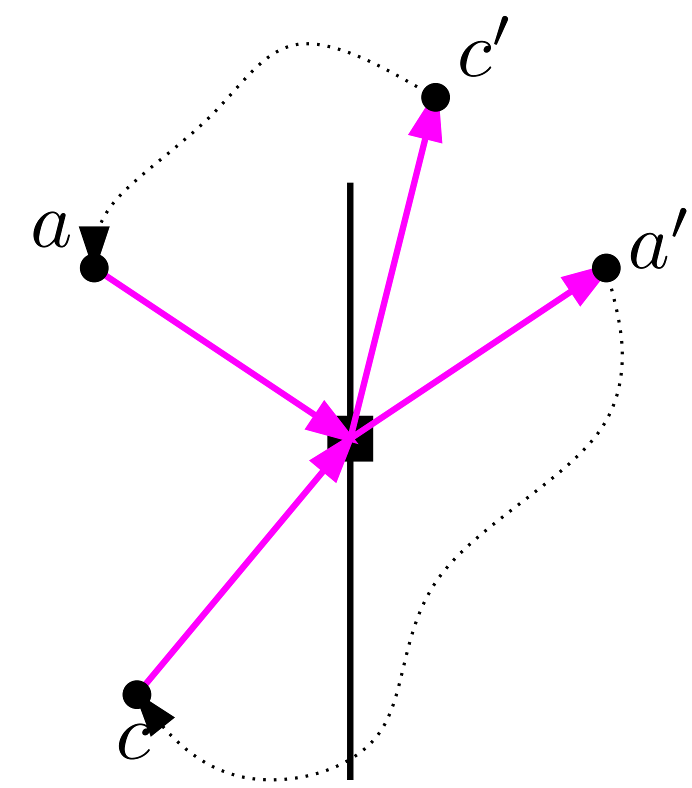


(a, a', b', b, c, c')

Def a tour is *k-light* if each portal is visited at most k times

Prop OPT_p is 2-light

(4) Portal-respecting tours



(a, a', c, c')

Def A tour is *portal-respecting* if it crosses the grid only at portals

Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k*-light* if each portal is visited at most k times

Prop OPT_p is 2-light

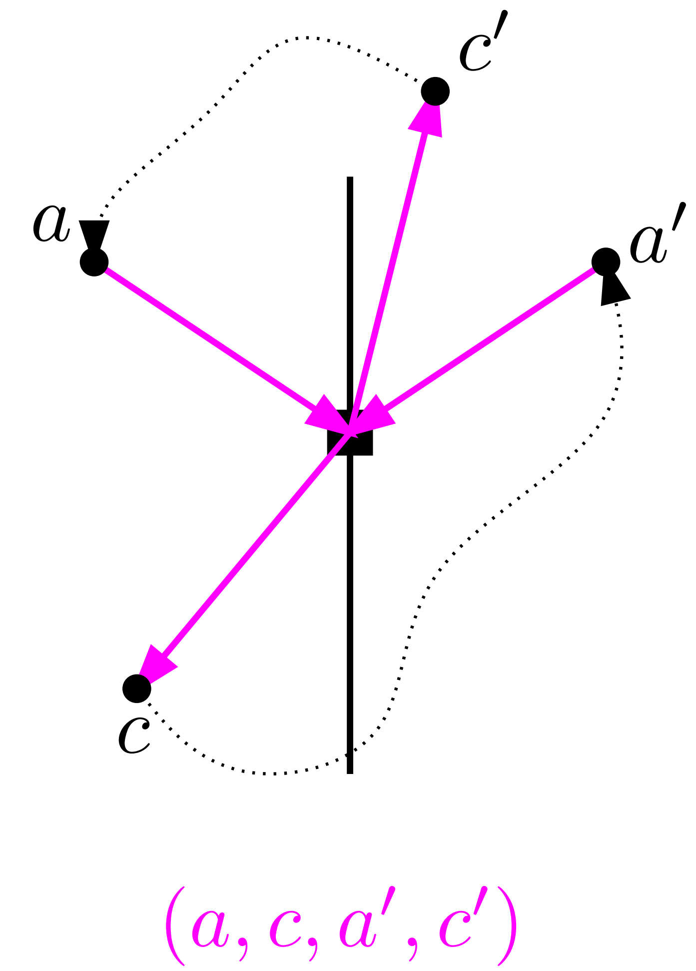
(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals

Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light



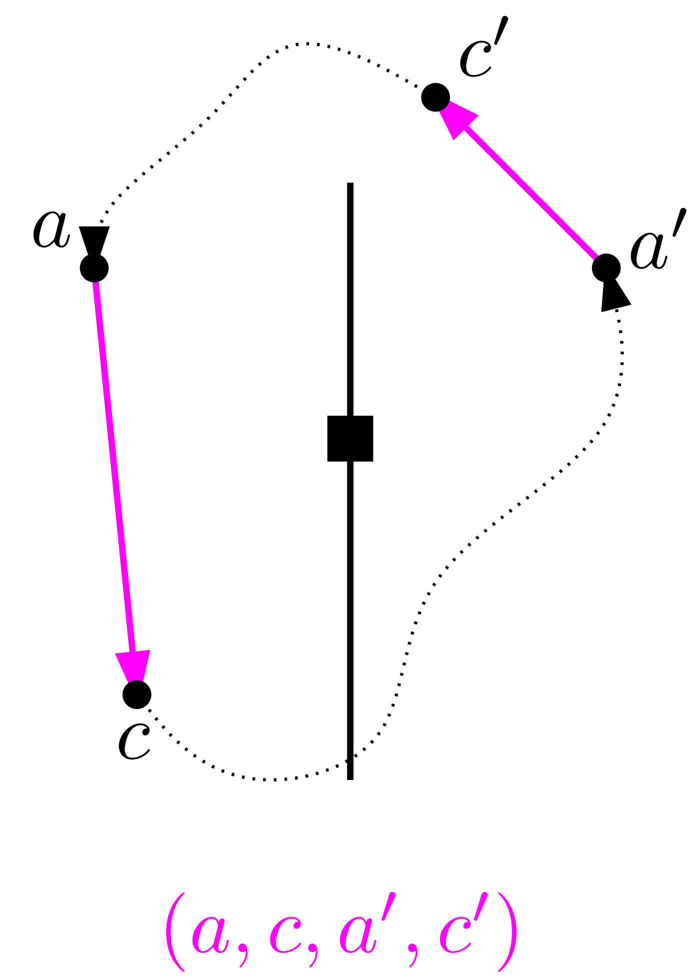
(4) Portal-respecting tours

Def A tour is *portal-respecting* if it crosses the grid only at portals

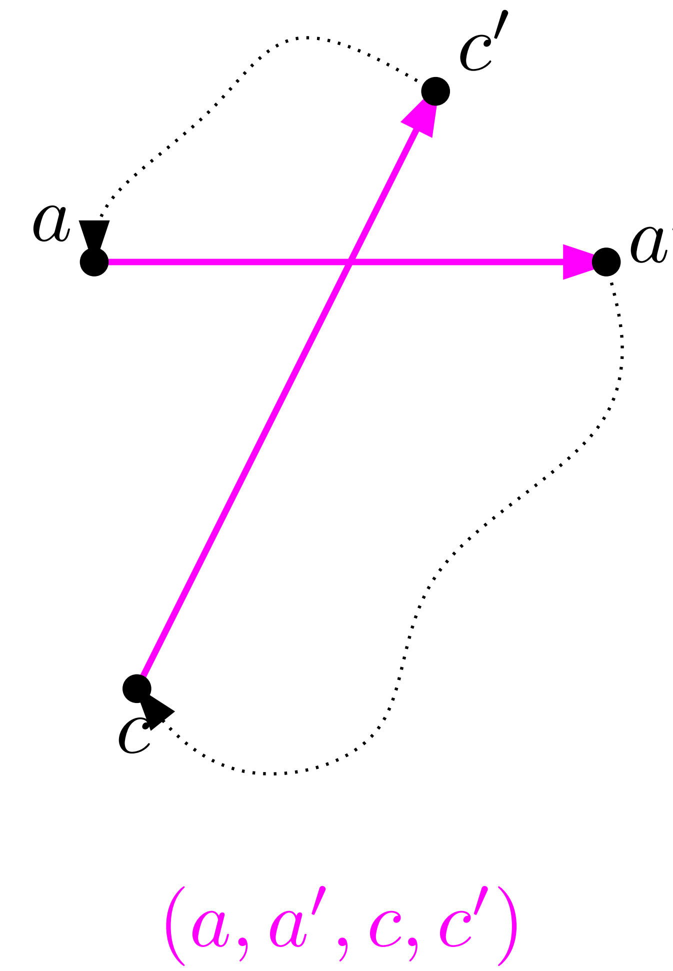
Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light



(4) Portal-respecting tours



Def A tour is *portal-respecting* if it crosses the grid only at portals

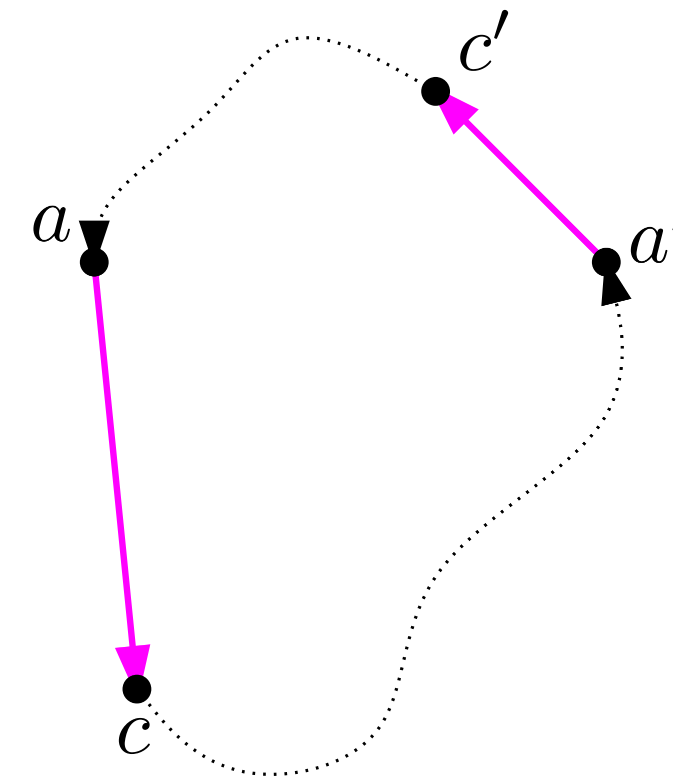
Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light

Prop OPT_p does not self-intersect, except at portals

(4) Portal-respecting tours



(a, c, a', c')

Def A tour is *portal-respecting* if it crosses the grid only at portals

Pb: an exhaustive search has considers infinitely many instances, since the number of passes through a portal is unbounded

Def a tour is k-*light* if each portal is visited at most k times

Prop OPT_p is 2-light

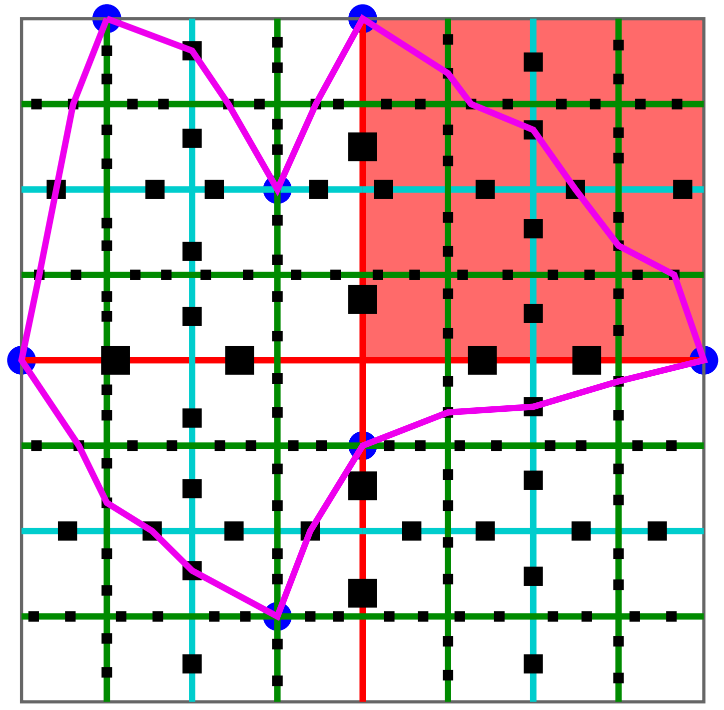
Prop OPT_p does not self-intersect, except at portals

(4) Portal-respecting tours

Goal: find shortest tour that is:

- portal-respecting
- 2-light
- non self-intersecting (except at portals)

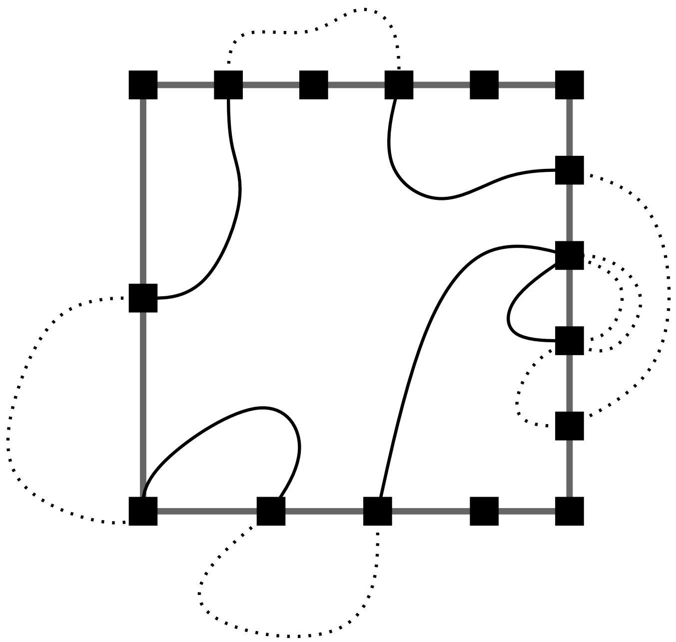
$\longrightarrow$ divide-and-conquer approach, using the quadtree



(4) Portal-respecting tours

Goal: find shortest tour that is:

- portal-respecting
- 2-light
- non self-intersecting (except at portals)



$\rightarrow$ divide-and-conquer approach, using the quadtree

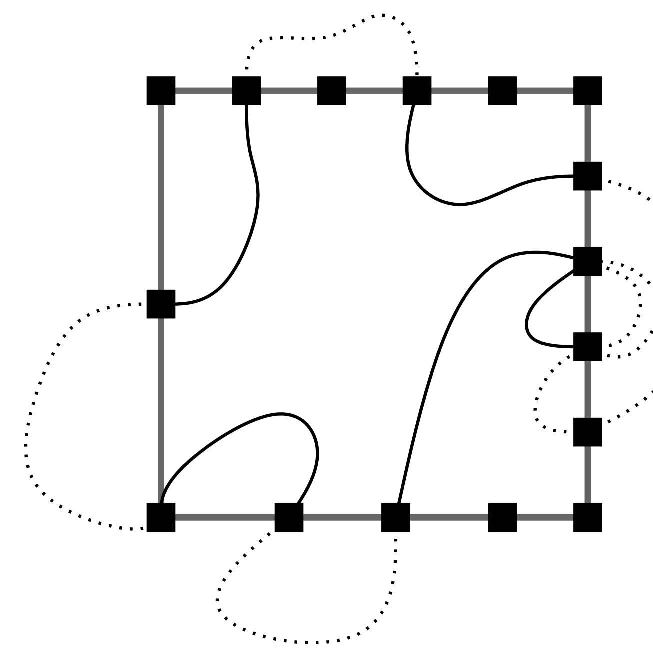
For any square s, interface is defined by:

- a number of passes through each portal of s — $3^{O(m)} = n^{O(1/\varepsilon)}$
- a paring between selected portals — $\Omega(m!) = \Omega(n^{\log n})$

(4) Portal-respecting tours

Goal: find shortest tour that is:

- portal-respecting
- 2-light
- non self-intersecting (except at portals)



$\rightarrow$ divide-and-conquer approach, using the quadtree

For any square s, interface is defined by:

- a number of passes through each portal of s
- a paring between selected portals

$3^{O(m)} = n^{O(1/\varepsilon)}$

$O(C_m) = O\left(2^{2m}\right) = n^{O(1/\varepsilon)}$

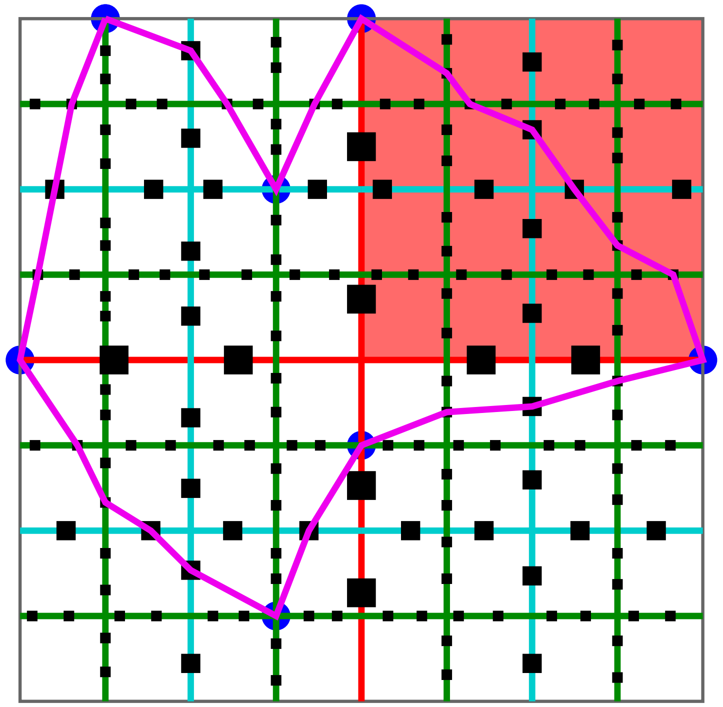
With the ordering of portals along the boundary, valid pairings are mapped injectively to balanced arrangements of parentheses

(4) Portal-respecting tours

Goal: find shortest tour that is:

- portal-respecting
- 2-light
- non self-intersecting (except at portals)

$\longrightarrow$ divide-and-conquer approach, using the quadtree

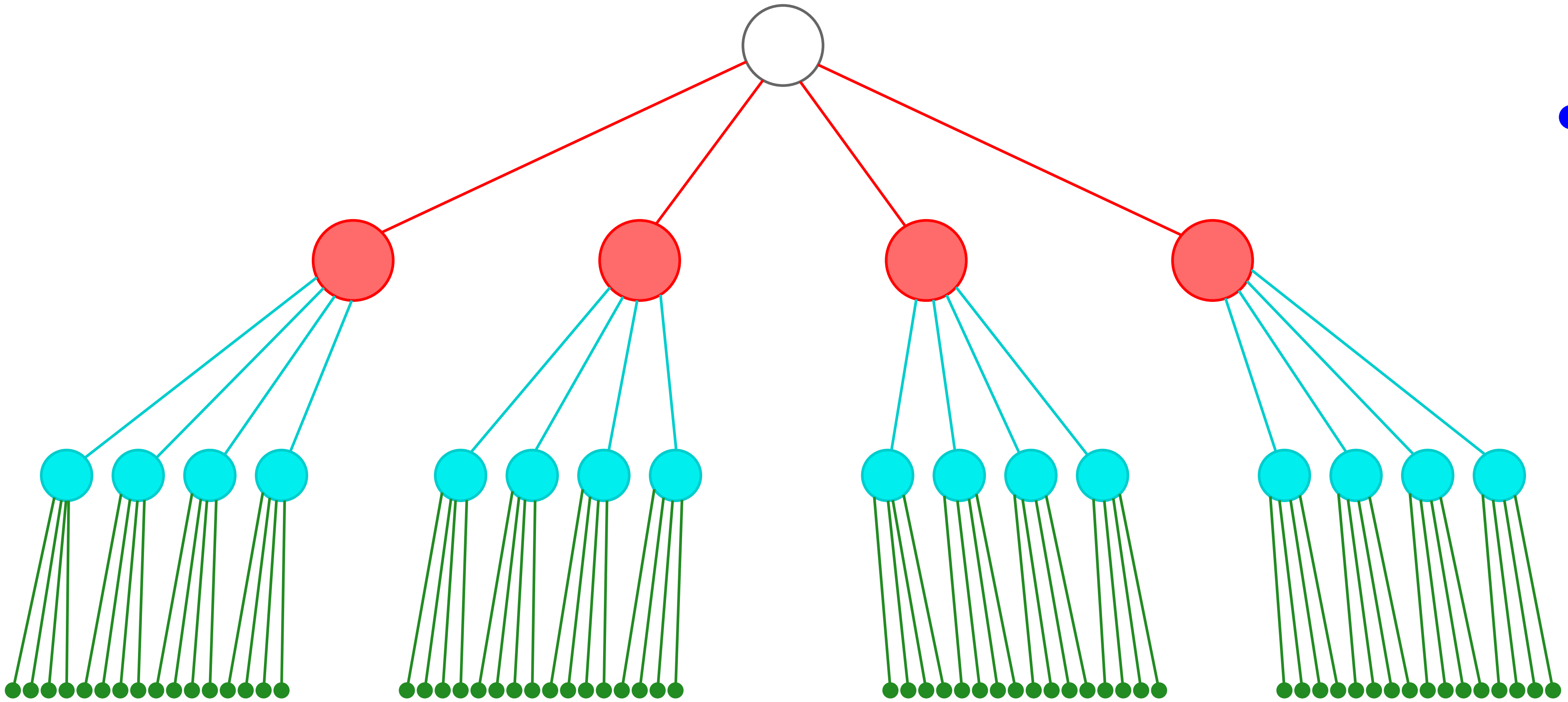


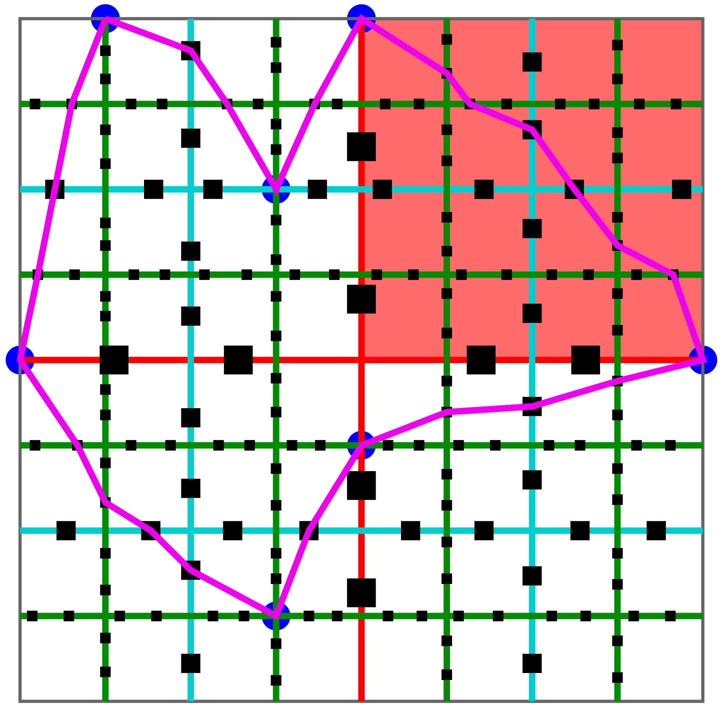
Pb: a simple recursion is not sufficient (optimum for square s is not concatenation of optima of sons of s)

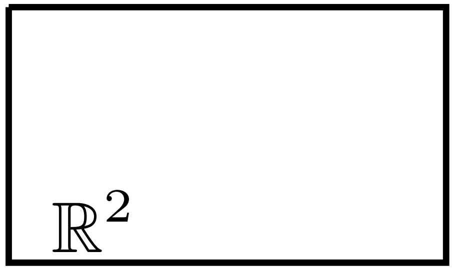
$\longrightarrow$ dynamic programming

(4) Portal-respecting tours

Lookup table:

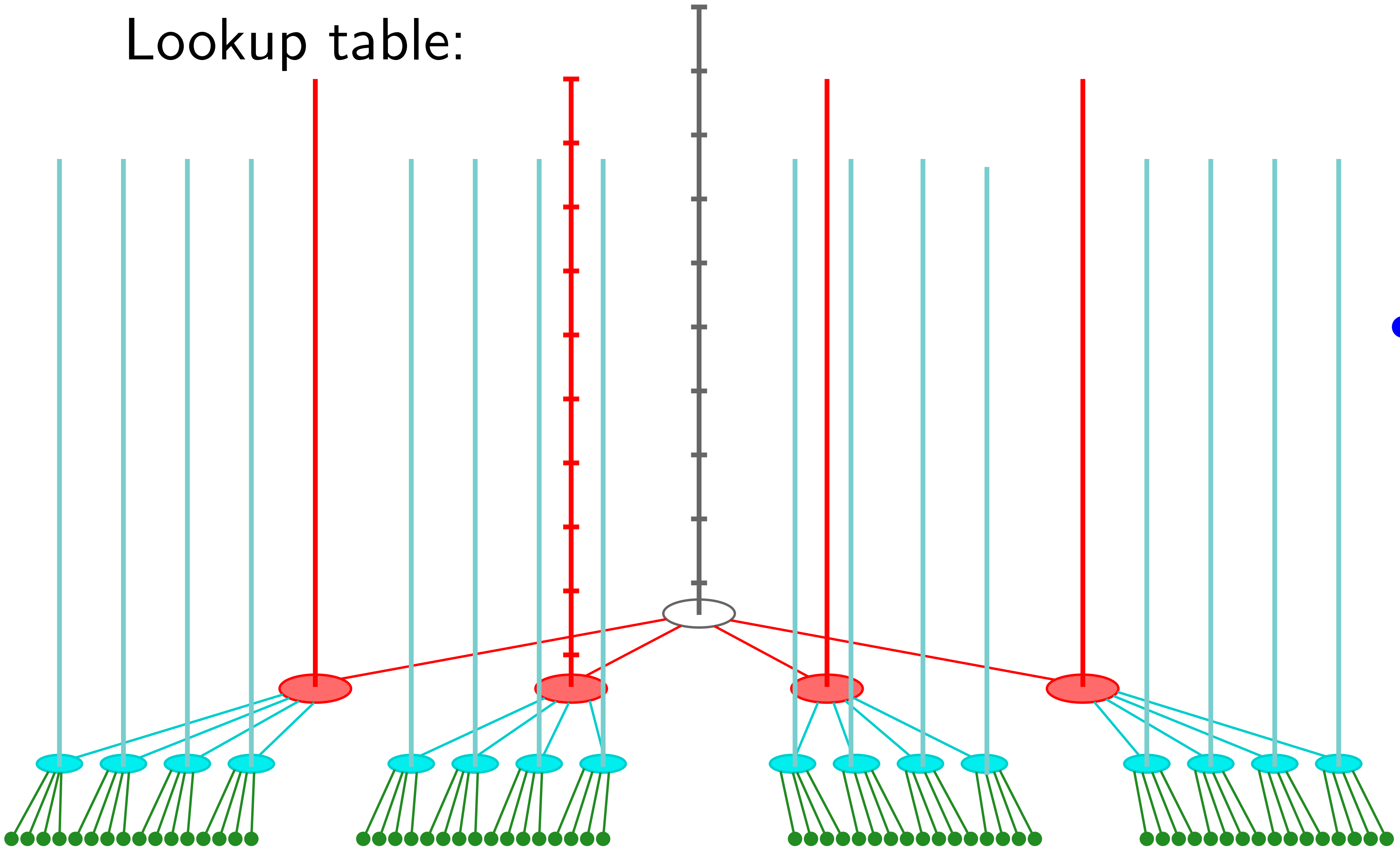


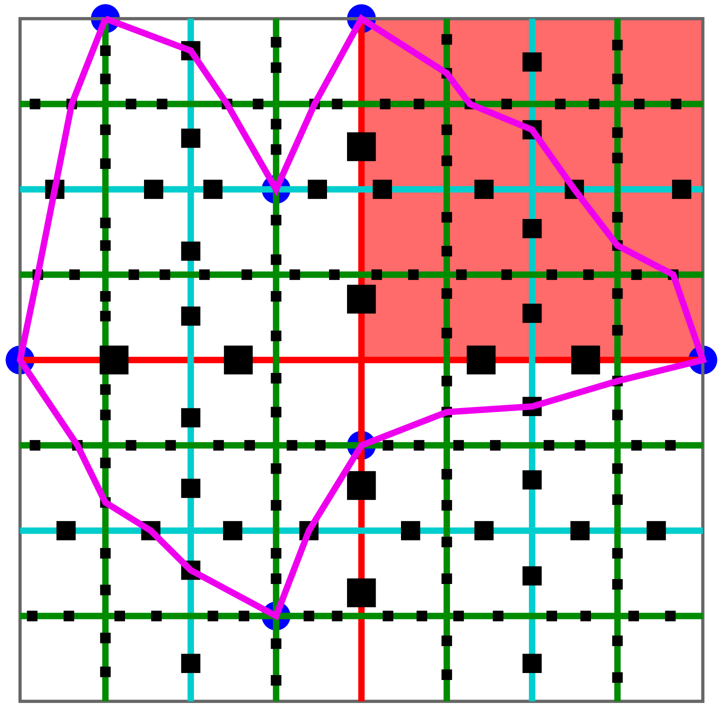




(4) Portal-respecting tours

Lookup table:

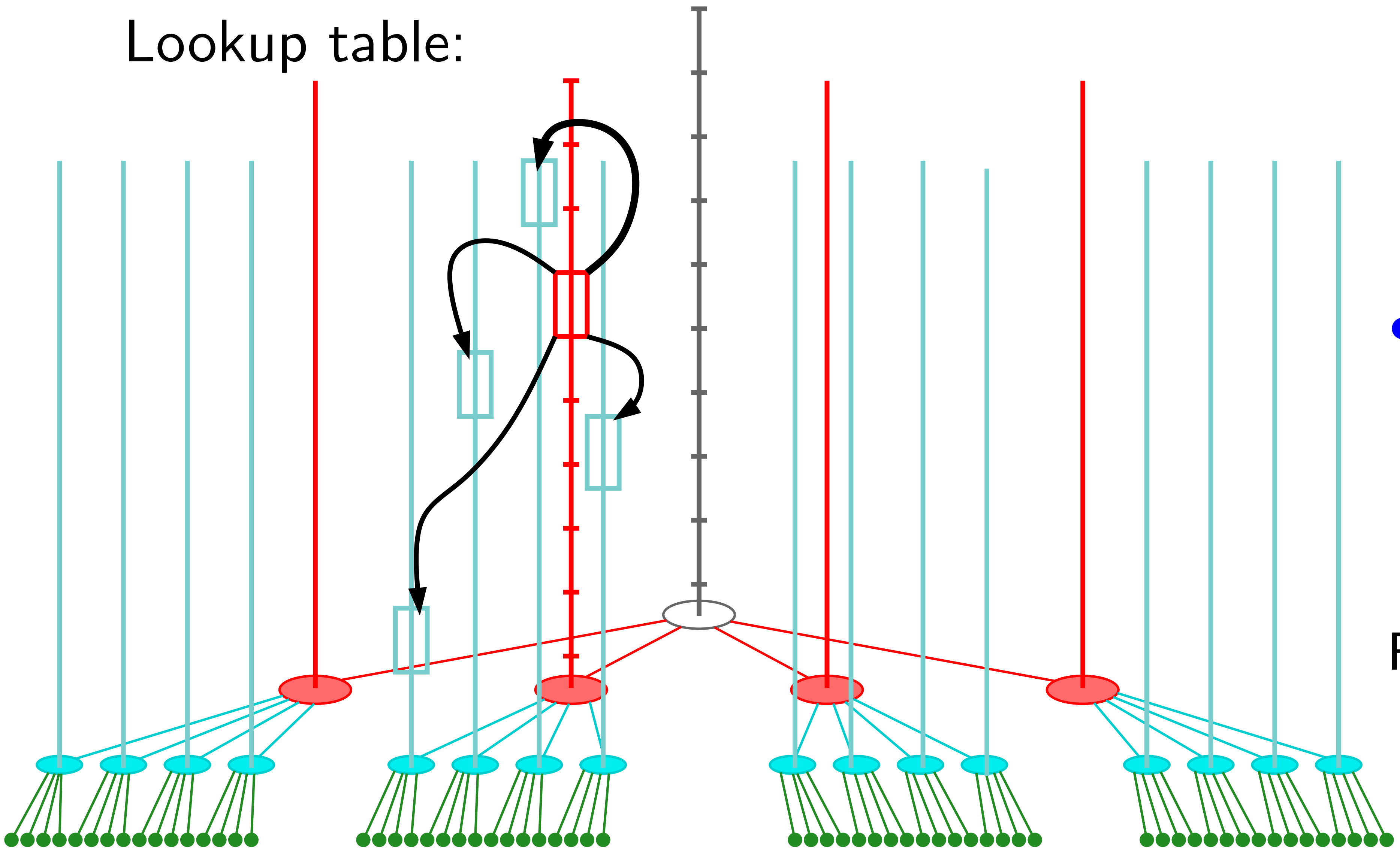


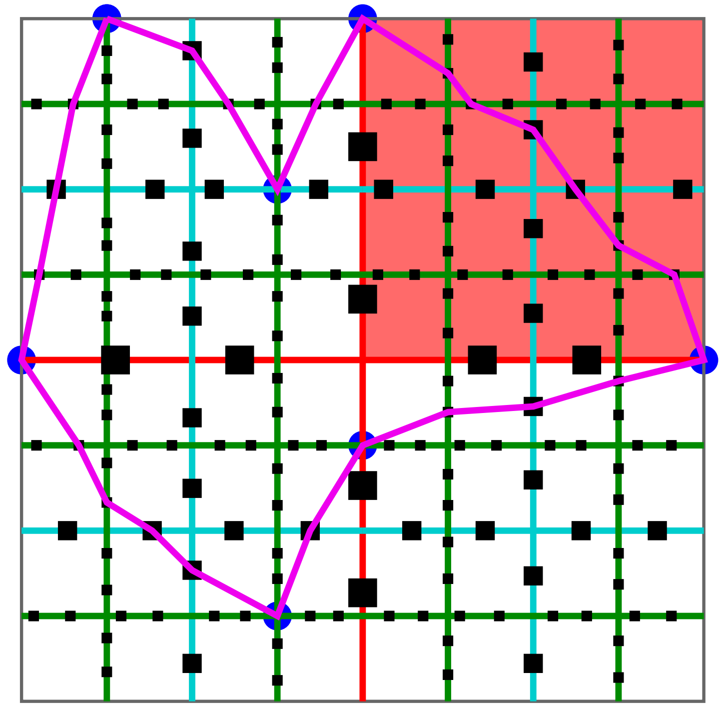


size: $O\left(n^4\, n^{O(1/\varepsilon)}\right)$

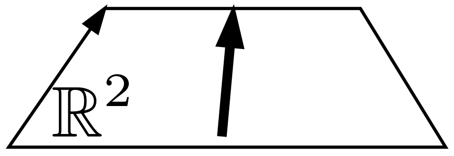
(4) Portal-respecting tours

Lookup table:



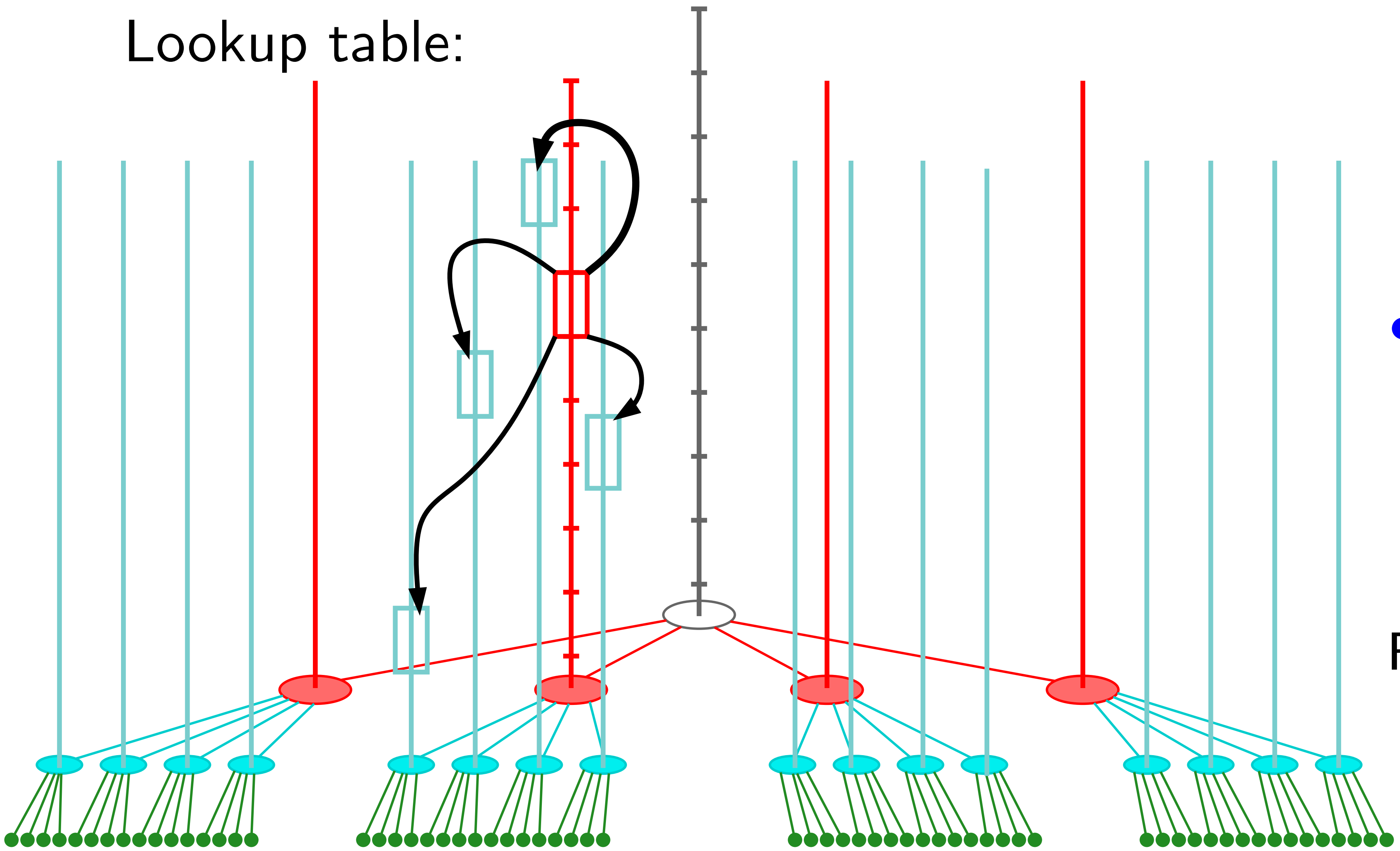


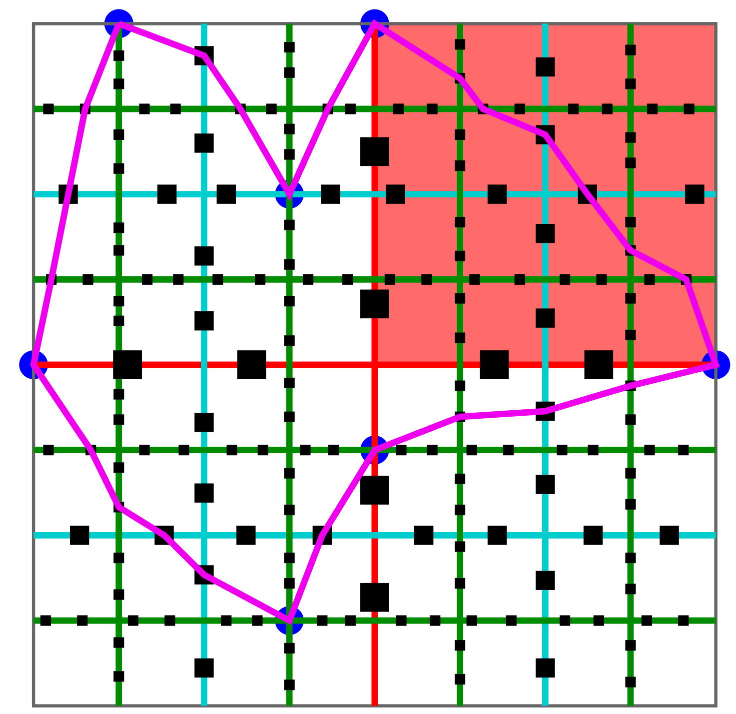
Fill the table "in depth"



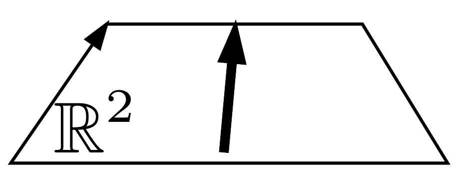
(4) Portal-respecting tours

Lookup table:





Fill the table "in depth"



$\forall$ (leaf,interface),

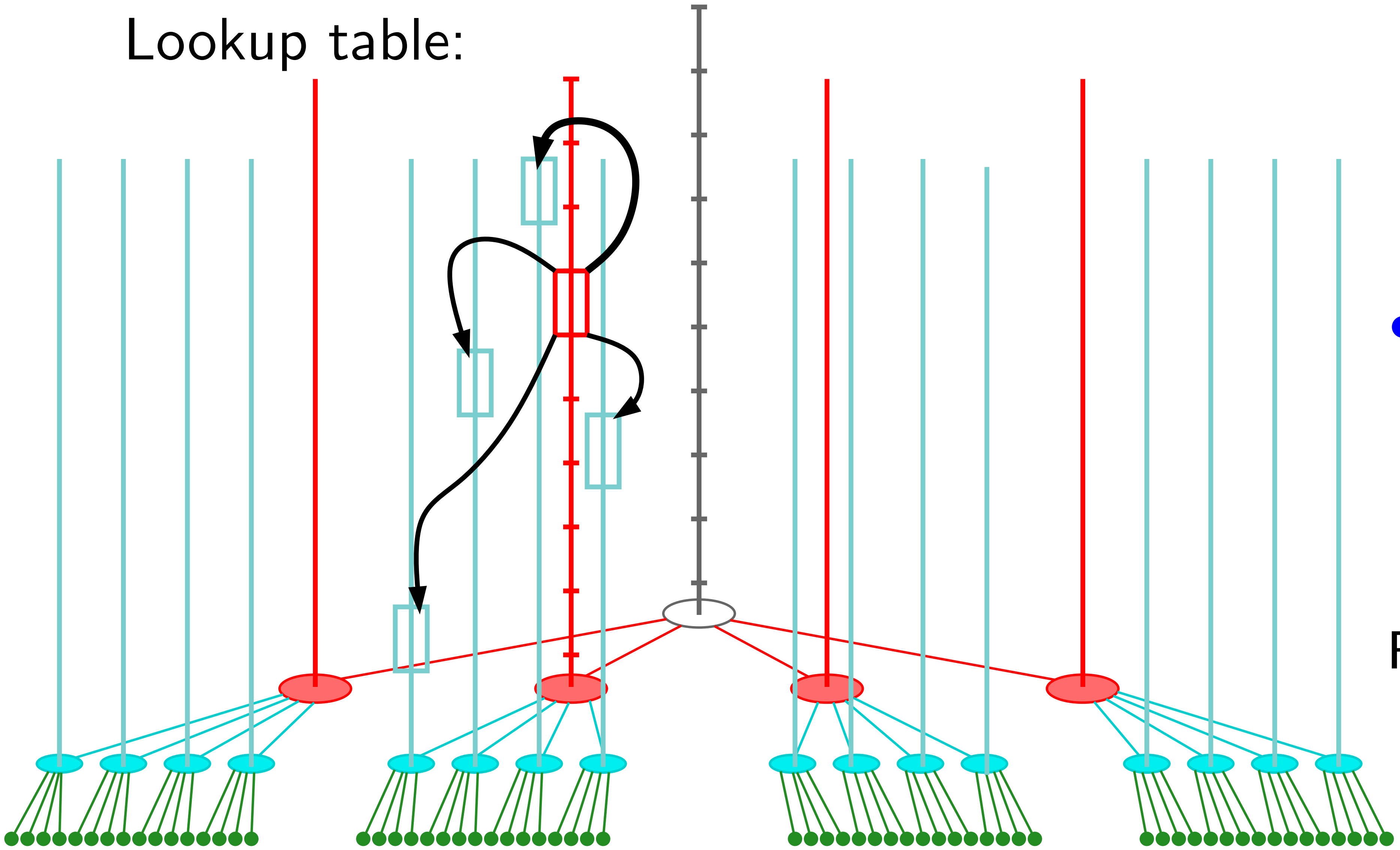
report total length of pairing w/ straight-line segments (nodes are portals) ← $O(1)$

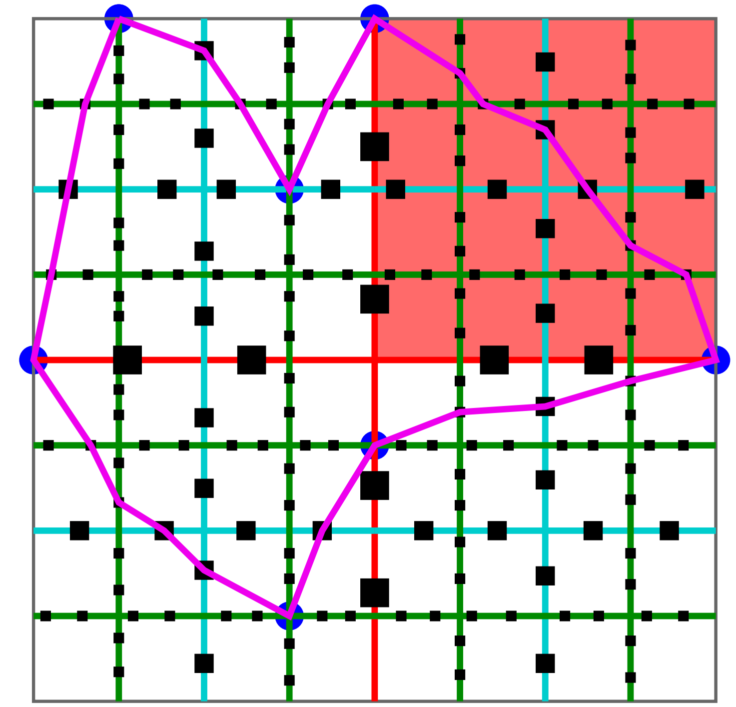
$\forall$ (node, interface),

- select interface for every son ← $n^{O(1/\varepsilon)}$
- retrieve best tour for each selected (son, interface) ← $O(1)$

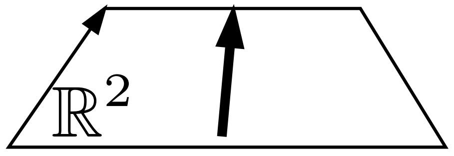
(4) Portal-respecting tours

Lookup table:





Fill the table "in depth"

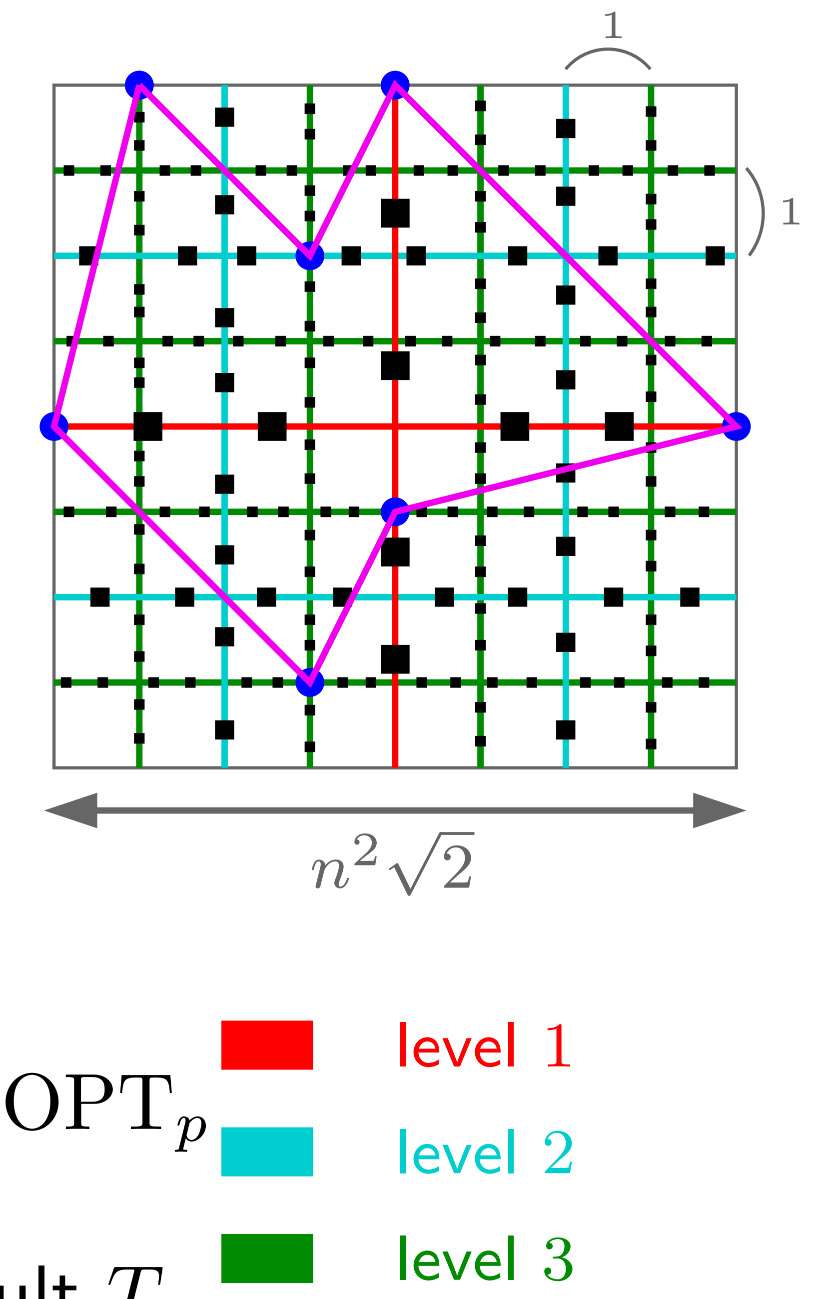


total running time: $O\left(n^4\ n^{O(1/\varepsilon)}\right)$

Output is the shortest tour that is portal-respecting
(and 2-light and non self-intersecting)

Euclidean TSP

Thm [Arora96]   Euclidean TSP admits a PTAS



Overview   Let $n = |V|$

(1) rescale/snap V

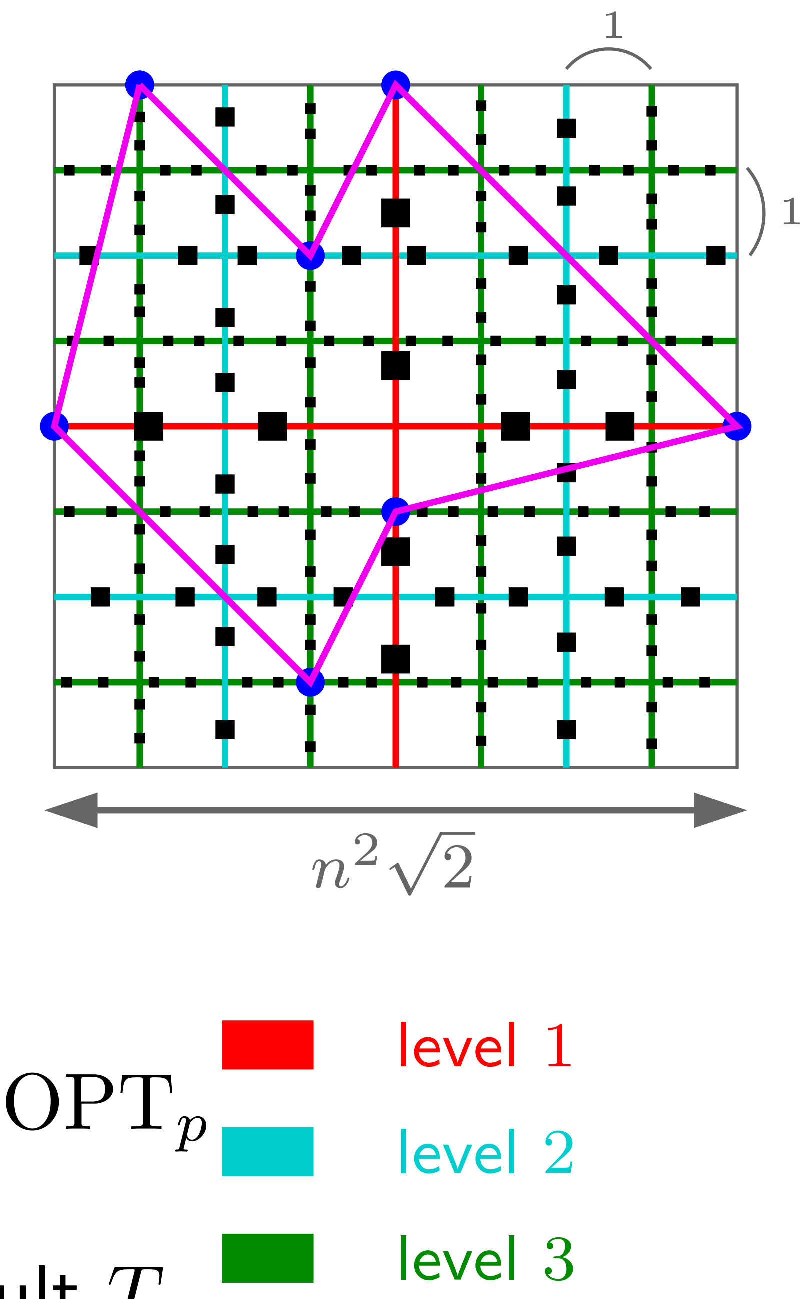
(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p

(5) Trim the edges of OPT_p and output the result T

Euclidean TSP



Thm [Arora96] Euclidean TSP admits a PTAS

Overview Let $n = |V|$

(1) rescale/snap V

(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p

(5) Trim the edges of OPT_p and output the result T

Q Do we have $|T| - |\mathrm{OPT}| \leq O(\varepsilon)\ |\mathrm{OPT}|$?

Euclidean TSP

Thm [Arora96] Euclidean TSP admits a PTAS

Overview Let $n = |V|$

(1) rescale/snap V

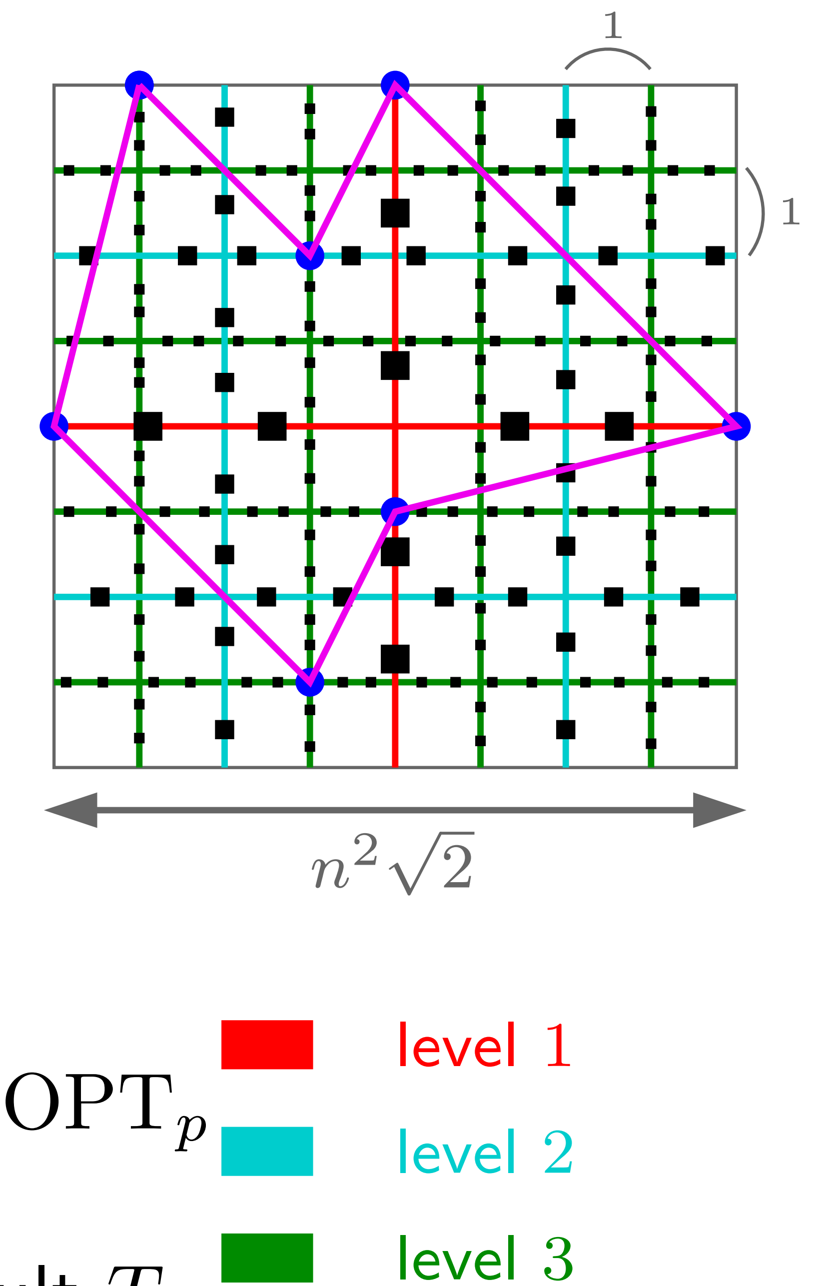
(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p

(5) Trim the edges of OPT_p and output the result T

Q Do we have $|\mathrm{OPT}_p| - |\mathrm{OPT}| \leq O(\varepsilon)\ |\mathrm{OPT}|$?



Euclidean TSP

Thm [Arora96]   Euclidean TSP admits a PTAS

Overview   Let $n = |V|$

(1) rescale/snap V

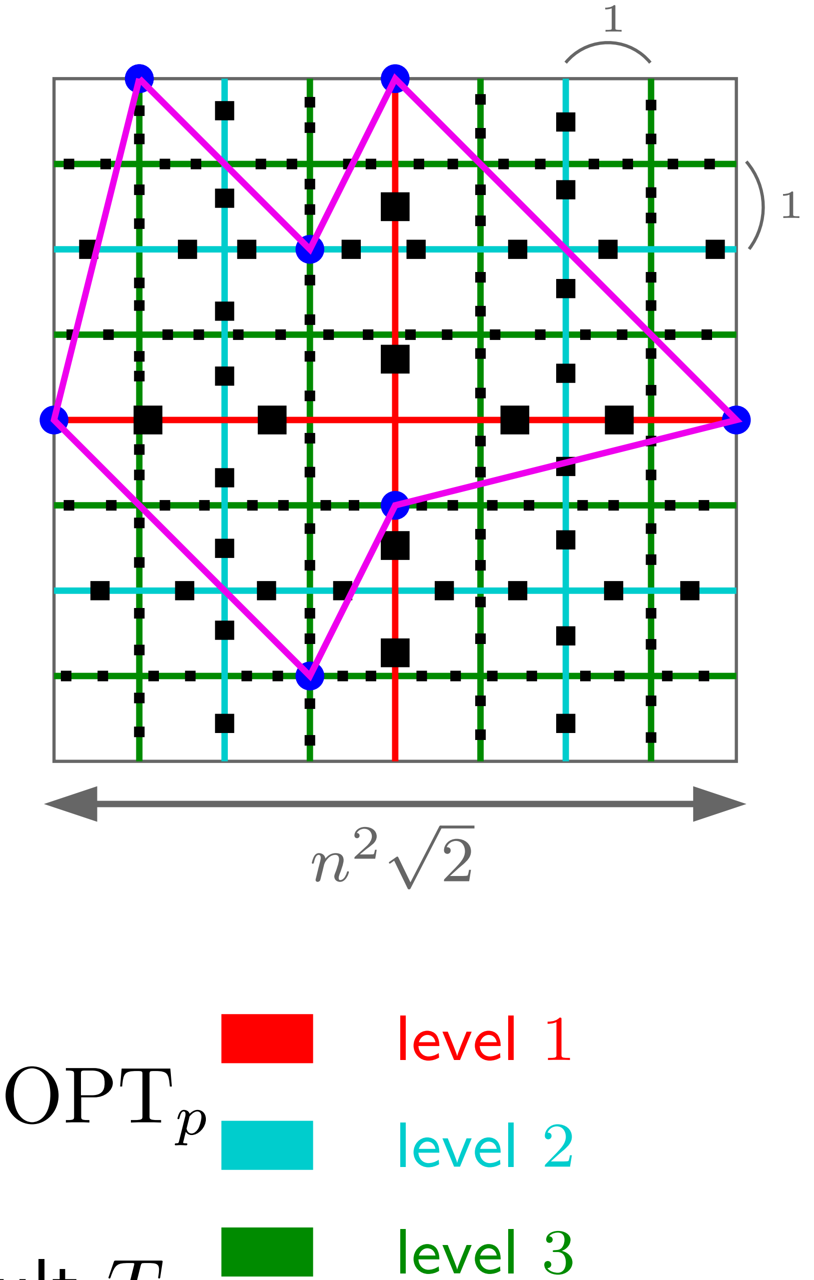
(2) subdivide the grid with a quadtree

(3) place *portals* on grid lines

(4) compute the smallest *portal-respecting* tour OPT_p

(5) Trim the edges of OPT_p and output the result T

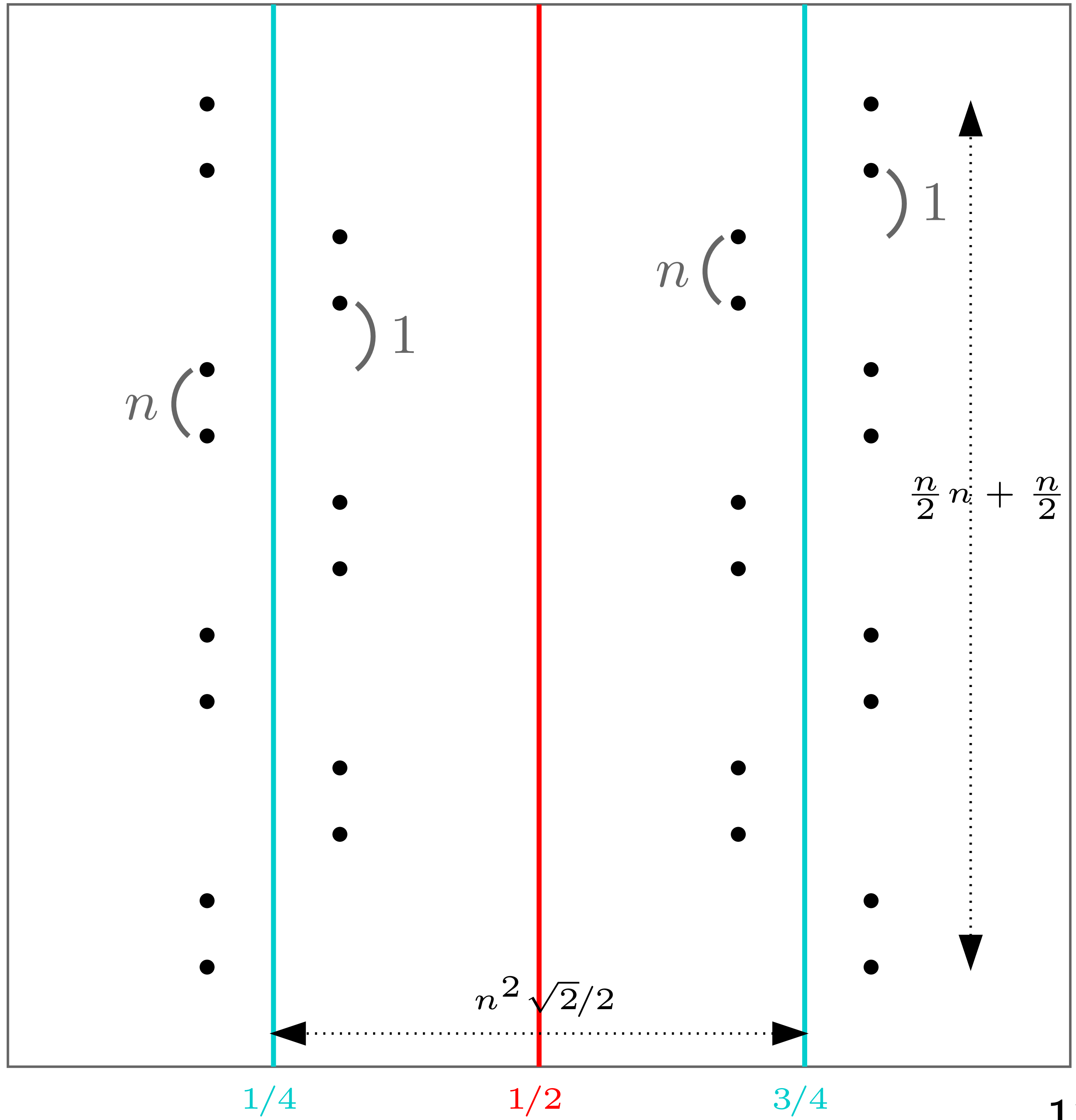
Q   Do we have $|p(\mathrm{OPT})| - |\mathrm{OPT}| \leq O(\varepsilon)\ |\mathrm{OPT}|$?



Structure theorem

Pb: $|\mathrm{OPT}_p|$ can be made arbitrarily large compared to $|\mathrm{OPT}|$

$|V| = 2n$

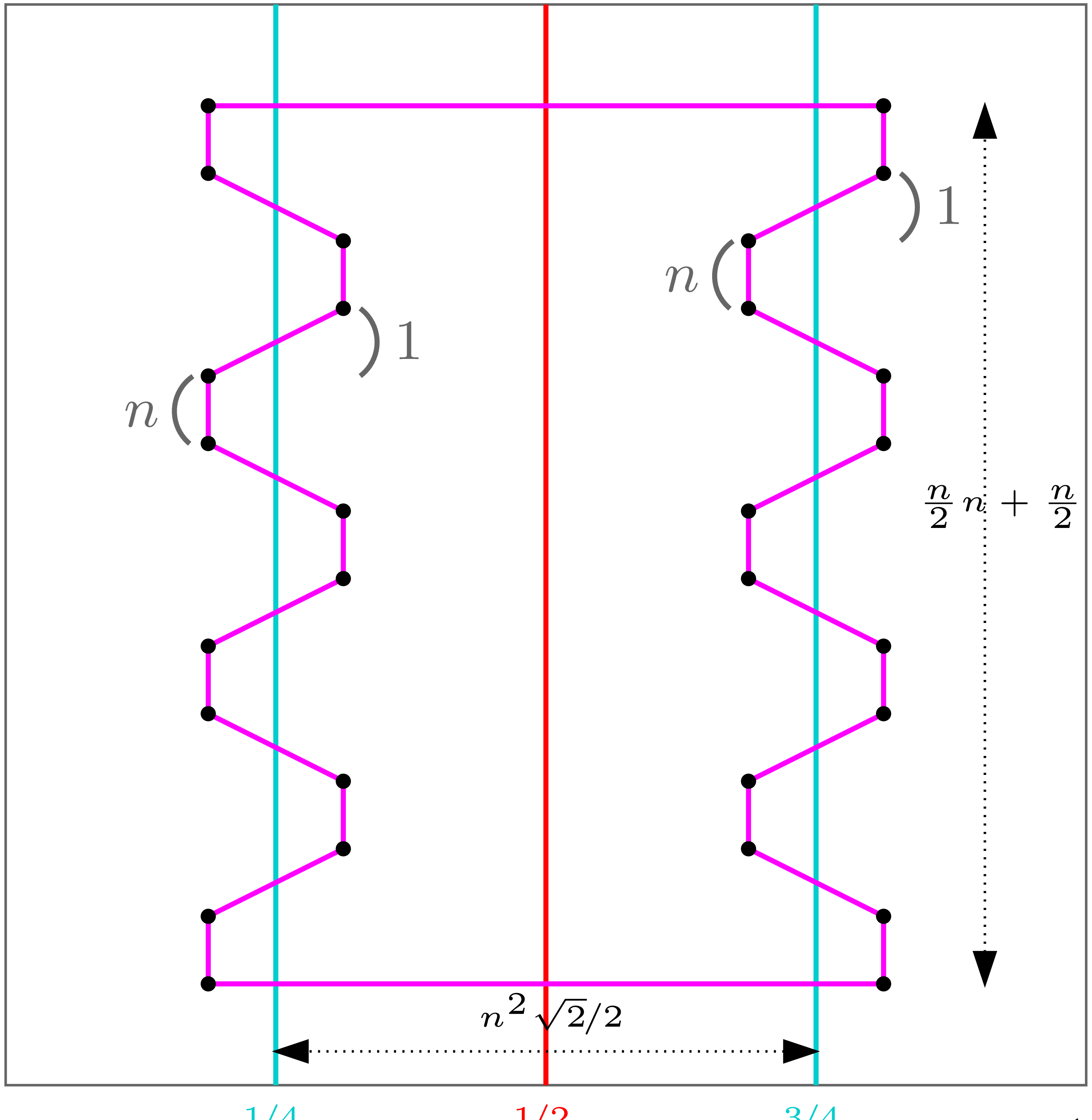


Structure theorem

Pb: $|\mathrm{OPT}_p|$ can be made arbitrarily large compared to $|\mathrm{OPT}|$

$|V| = 2n$

$|\mathrm{OPT}| \leq 2\frac{n}{2}n + 2\frac{n}{2}2\sqrt{2} + 2n^2\frac{\sqrt{2}}{2} = n^2(1+\sqrt{2}) + 2n\sqrt{2}$



Structure theorem

Pb: $|\mathrm{OPT}_p|$ can be made arbitrarily large compared to $|\mathrm{OPT}|$

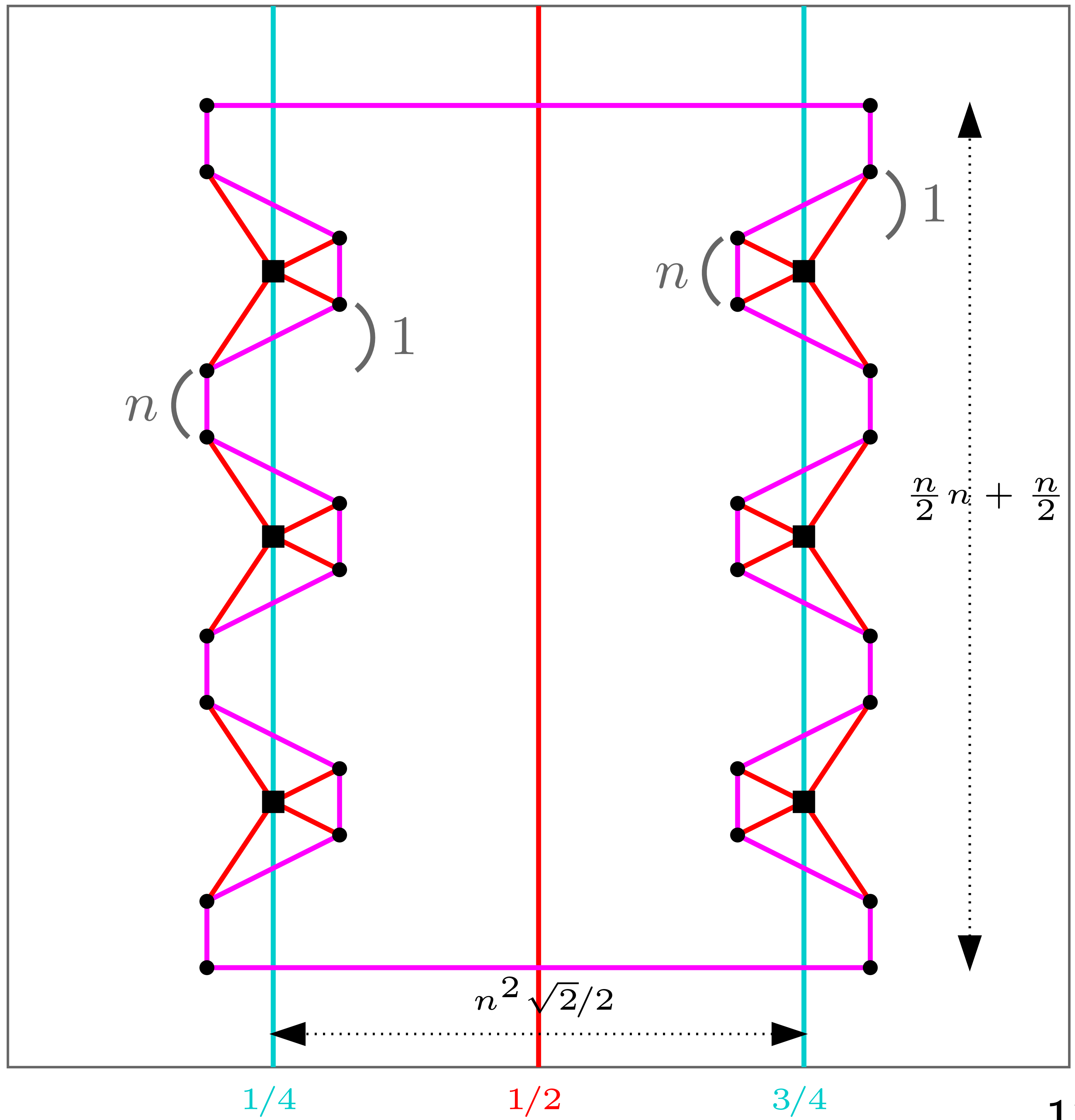
$|V| = 2n$

$|\mathrm{OPT}| \leq 2\frac{n}{2}n + 2\frac{n}{2}2\sqrt{2} + 2n^2\frac{\sqrt{2}}{2} = n^2(1+\sqrt{2}) + 2n\sqrt{2}$

At level 2, $4m$ portals $\Rightarrow$ inter-portal distance $\delta = \frac{n^2+2n}{8m} \gg n$

One crossing every n $\Rightarrow$ overhead per consecutive portals $\geq 2\frac{\delta}{4} = \frac{\delta}{2}$ $\Rightarrow$ total overhead $\geq 4m\frac{\delta}{2} = \frac{(n^2+2n)^2}{4} = \Omega(|\mathrm{OPT}|)$ (indep. of ε)

(same for tours close to OPT)

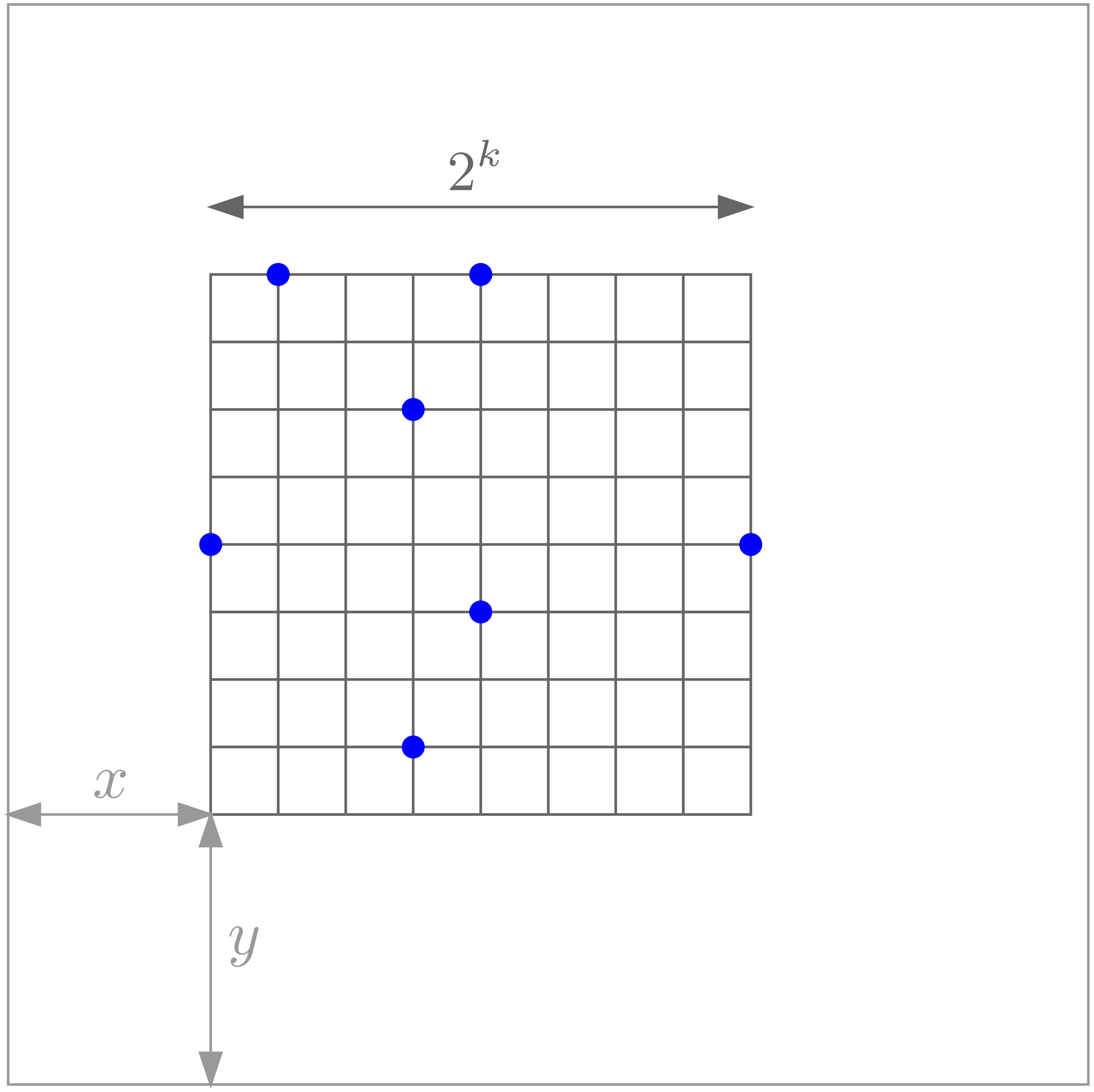


Structure theorem

Pb: $|\mathrm{OPT}_p|$ can be made arbitrarily large compared to $|\mathrm{OPT}|$

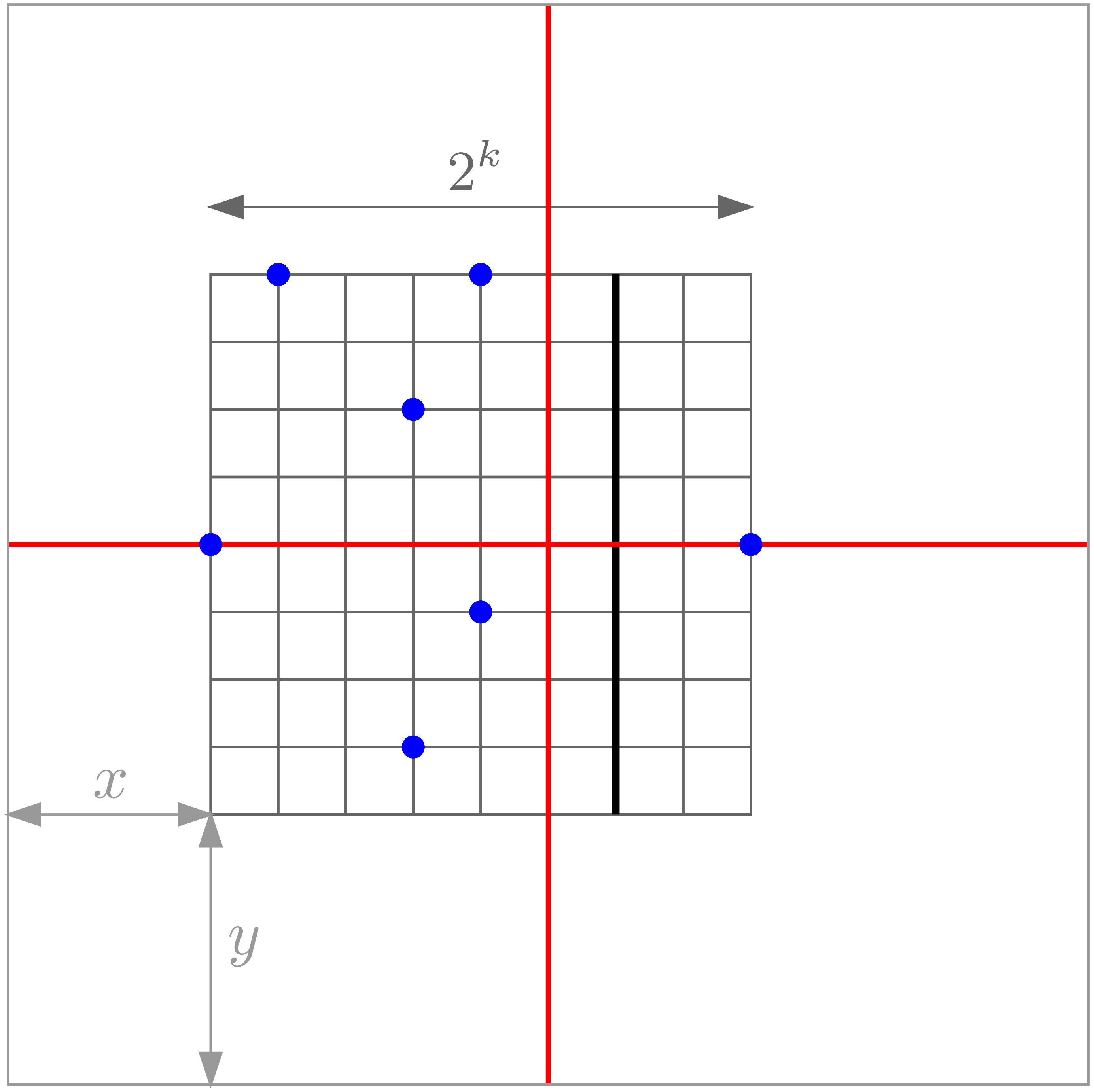
Patch: randomize the algorithm:

Choose random integers $0 \leq x, y \leq 2^k$, then apply (2)-(5) to square of sidelength 2^{k+1} shifted by $(-x, -y)$.



Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}\,|\mathrm{OPT}|$



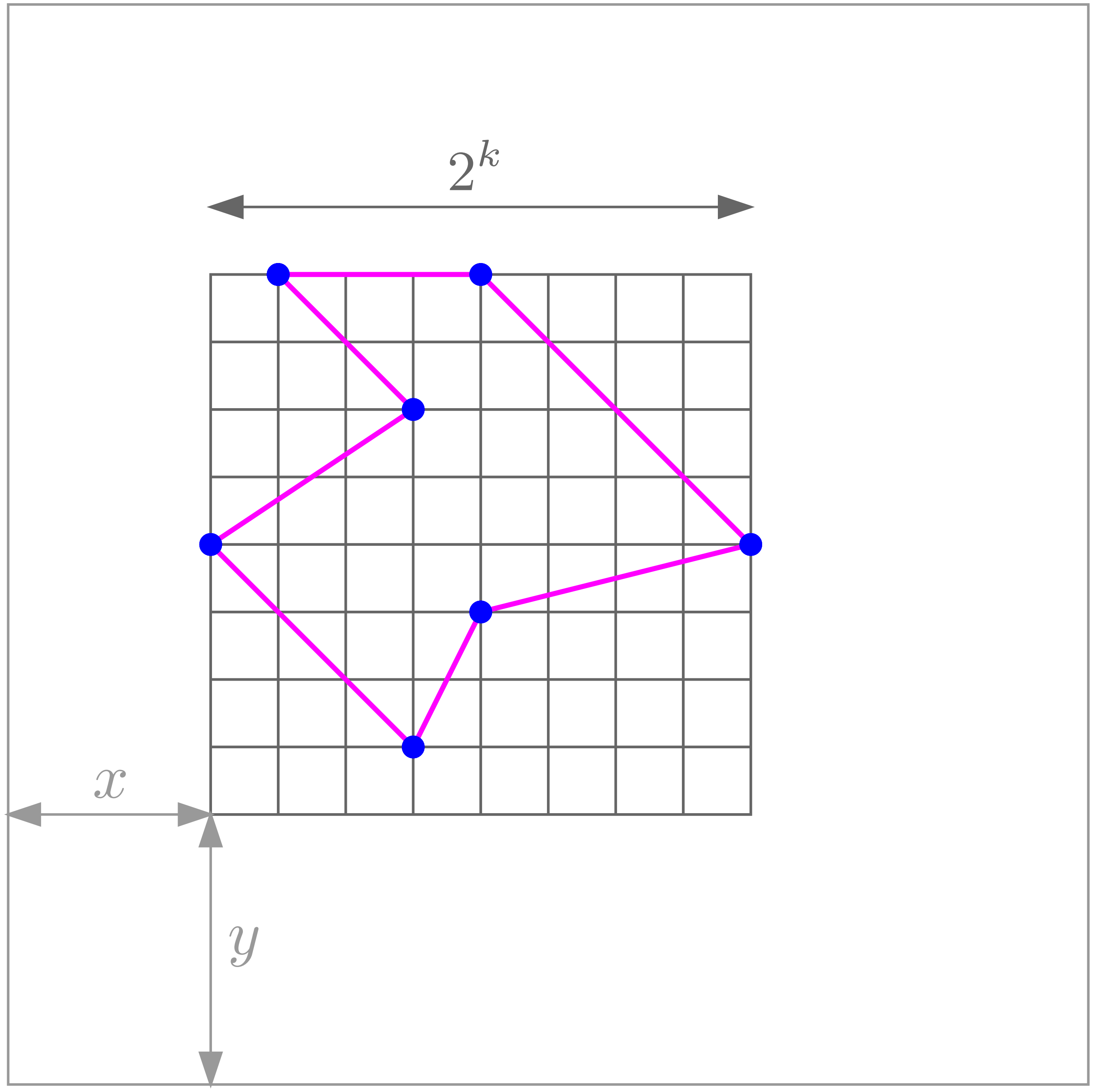
For any vertical line l in domain,
$P_x(\text{l is at level } i) = \frac{2^{i-2}}{1+2^k}$

$$\left(\begin{array}{l} 2^{i-1} \text{ level } i \text{ lines, half of which reach } l \\ 1 + 2^k \text{ possible values for } x \end{array}\right.$$

Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}\,|\mathrm{OPT}|$

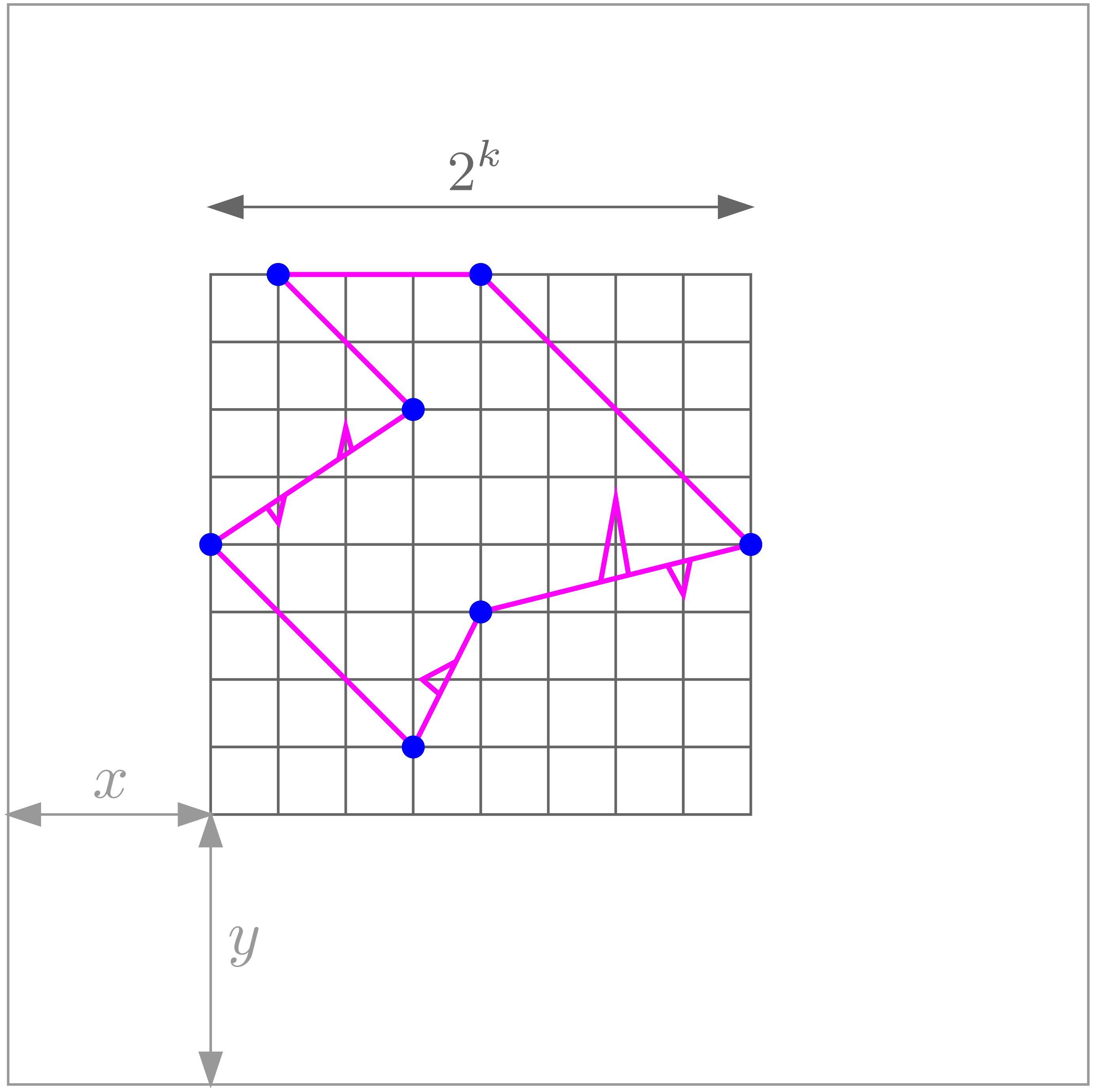
$\longrightarrow$ transform OPT into a portal-respecting tour:



Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m} |\mathrm{OPT}|$

$\longrightarrow$ transform OPT into a portal-respecting tour:



Structure theorem

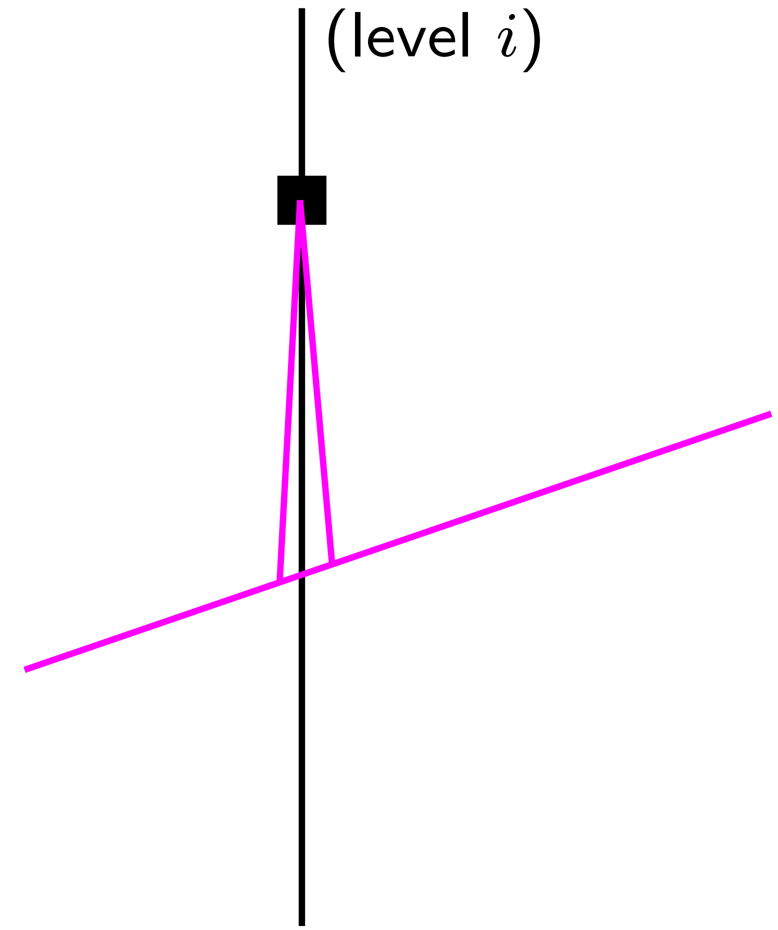
Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}\,|\mathrm{OPT}|$

$\rightarrow$ transform OPT into a portal-respecting tour:

For every crossing, overhead ≤ 2 times half the interportal distance $= \frac{2^{k+1}}{m\ 2^i}$

$P_x(\text{level i}) = \frac{2^{i-2}}{1+2^k}$ (same for y)

Expected overhead: $\sum_{i=1}^{k+1} \frac{2^{i-2}}{1+2^k}\ \frac{2^{k+1}}{m\ 2^i}$
$\leq \sum_{i=1}^{k+1} \frac{2^{i-2}}{2^k}\ \frac{2^{k+1}}{m\ 2^i} = \frac{k+1}{2m}$



Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}\,|\mathrm{OPT}|$

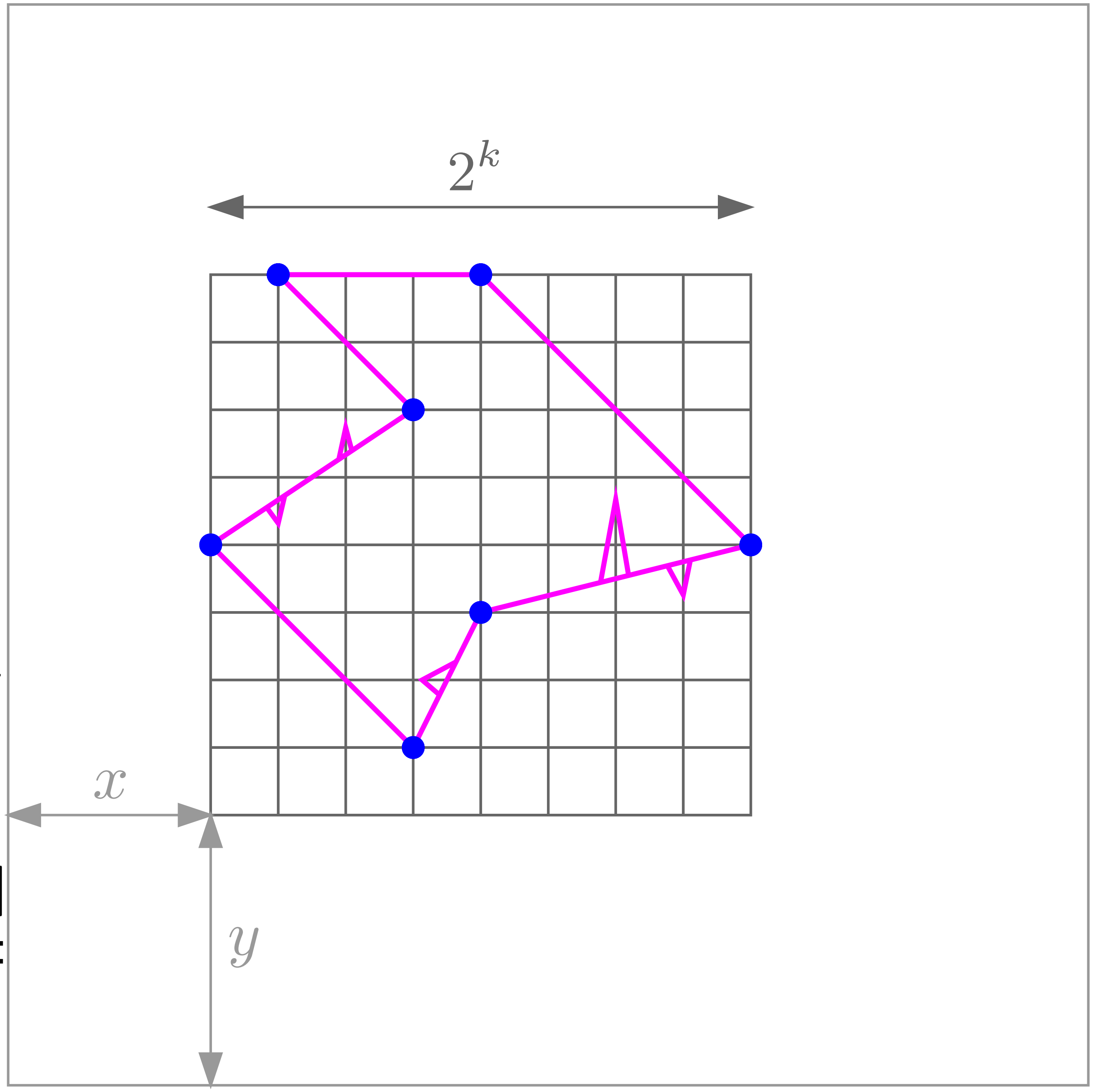
$\rightarrow$ transform OPT into a portal-respecting tour:

For every crossing, overhead ≤ 2 times half the interportal distance $= \frac{2^{k+1}}{m\ 2^i}$

P_x(level i) $= \frac{2^{i-2}}{1+2^k}$ (same for y)

Expected overhead: $\sum_{i=1}^{k+1} \frac{2^{i-2}}{1+2^k}\ \frac{2^{k+1}}{m\ 2^i}$
$\leq \sum_{i=1}^{k+1} \frac{2^{i-2}}{2^k}\ \frac{2^{k+1}}{m\ 2^i} = \frac{k+1}{2m}$

OPT crosses the grid at most $2|\mathrm{OPT}|$ times $\Rightarrow$ total expected overhead: $\frac{k+1}{m}\,|\mathrm{OPT}|$



Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}\,|\mathrm{OPT}| \le \frac{2\log n + 3/2 + 1}{\log n / 2\varepsilon}\,|\mathrm{OPT}| \le (4 + 5/\log n)\,\varepsilon\,|\mathrm{OPT}| \le 9\varepsilon\,|\mathrm{OPT}|$.

$(n \ge 2)$

$$2^k \le 2n^2\sqrt{2}$$
$$m = \left\lfloor \frac{\log n}{\varepsilon} \right\rfloor \ge \frac{\log n}{2\varepsilon}$$

Structure theorem

Thm The expectation (over x, y) of $|\mathrm{OPT}_g| - |\mathrm{OPT}|$ is at most $\frac{k+1}{m}|\mathrm{OPT}| \le \frac{2\log n + 3/2 + 1}{\log n / 2\varepsilon}|\mathrm{OPT}| \le (4 + 5/\log n)\,\varepsilon\,|\mathrm{OPT}| \le 9\varepsilon\,|\mathrm{OPT}|$.

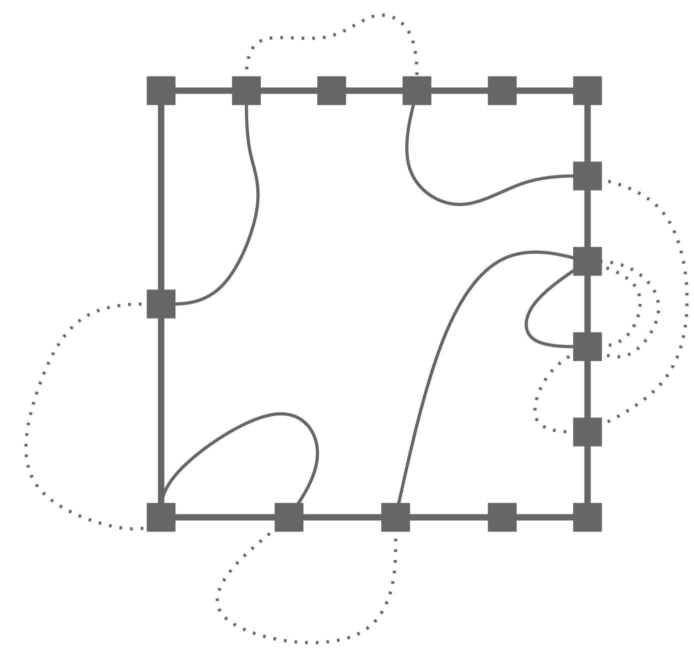
Corollary $P_{x,y}\left(|\mathrm{OPT}_g| - |\mathrm{OPT}| \le 18\varepsilon\,|\mathrm{OPT}|\right) \ge 1/2$

$\rightarrow$ **Monte-Carlo procedure** given a constant $0 < c < 1$, repeat $\lceil \log(1/c) \rceil$ times the process "randomization + (2)-(5)" and keep the best computed tour T. Then, $P\left(|\mathrm{OPT}_g| - |\mathrm{OPT}| \le 18\varepsilon\,|\mathrm{OPT}|\right) \ge 1 - c$

$\rightarrow$ **Derandomization** try all possible choices of (x, y) (there are $O(n^4)$ of those), and keep best tour.

Higher dimensions

The analysis extends to higher dimensions, except for the *valid pairing* argument.

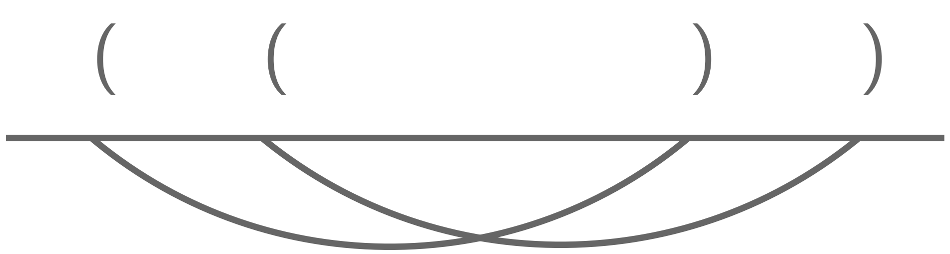


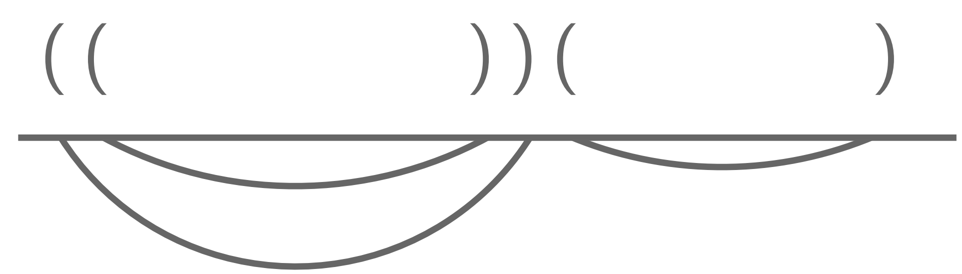
For any square s, interface is defined by:

- a number of passes through each portal of s
- a paring between selected portals

$3^{O(m)} = n^{O(1/\varepsilon)}$

$O(C_m) = O\left(2^{2m}\right) = n^{O(1/\varepsilon)}$





Higher dimensions

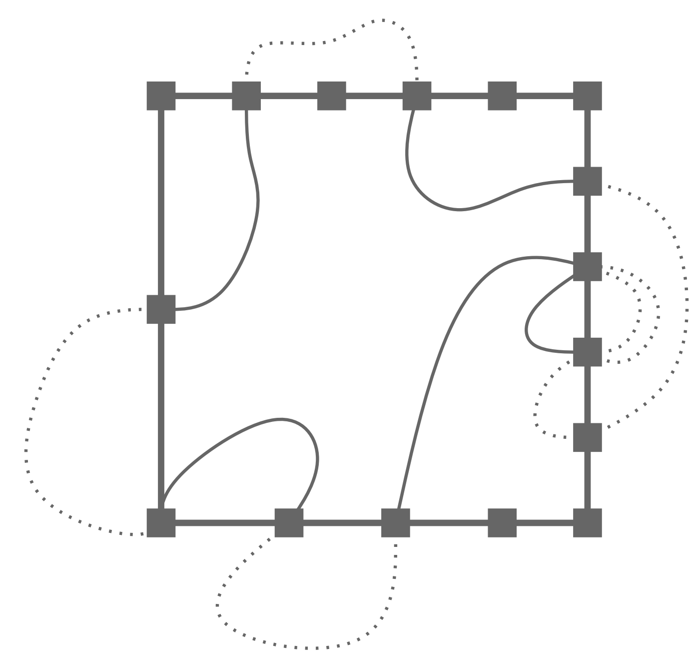
The analysis extends to higher dimensions, except for the *valid pairing* argument.

Patch: instead of considering all 2-light tours, consider only those that intersect each side of the boundary of a given square at most l times.

Goal: find shortest tour that is:

- portal-respecting
- ~~2-light~~
- non self-intersecting (except at portals)

$\longrightarrow$ divide-and-conquer approach, using the quadtree



Higher dimensions

The analysis extends to higher dimensions, except for the *valid pairing* argument.

Patch: instead of considering all 2-light tours, consider only those that intersect each side of the boundary of a given square at most l times.

$\mathbb{R}^2$

Thm $\quad \mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq \left(\frac{\log(n)+1}{m} + \frac{12}{l-5}\right) |\mathrm{OPT}|$

$\rightarrow$ for $l = \Theta\left(\frac{1}{\varepsilon}\right)$ and $m = \left\lfloor \frac{\log n}{\varepsilon} \right\rfloor$:

- $\mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq O(\varepsilon)\, |\mathrm{OPT}|$
- $\forall$ square, $\#\{\text{interfaces}\} \leq m^{O(l)}\, l! \leq (\log n)^{O(1/\varepsilon)}$

  $\Rightarrow$ space complexity $\leq O\left(n^4 (\log n)^{O(1/\varepsilon)}\right)$

  $\Rightarrow$ time complexity $\leq O\left(n^4 (\log n)^{O(1/\varepsilon)}\right)$

Higher dimensions

The analysis extends to higher dimensions, except for the *valid pairing* argument.

Patch: instead of considering all 2-light tours, consider only those that intersect each side of the boundary of a given square at most l times.

$\mathbb{R}^d$

Thm $\quad \mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq O\left(\frac{\log(n)\sqrt{d}}{m^{\frac{1}{d-1}}} + \frac{(l+1)^{1-\frac{1}{d-1}}}{l+1-2^{d+1}}\right)|\mathrm{OPT}|$

$\rightarrow$ for $l = \Theta\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)$ and $m = \Theta\left(\left(\log(n)\sqrt{d}/\varepsilon\right)^{d-1}\right)$:

- $\mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq O(\varepsilon)\,|\mathrm{OPT}|$
- $\forall$ square, $\#\{\text{interfaces}\} \leq m^{O(2dl)}\; l! \leq O\left((\log n)^{O\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)}\right)$

  $\Rightarrow$ space complexity $\leq O\left(n^{2d}(\log n)^{O\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)}\right)$

  $\Rightarrow$ time complexity $\leq O\left(n^{2d}(\log n)^{O\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)}\right)$

Higher dimensions

The analysis extends to higher dimensions, except for the *valid pairing* argument.

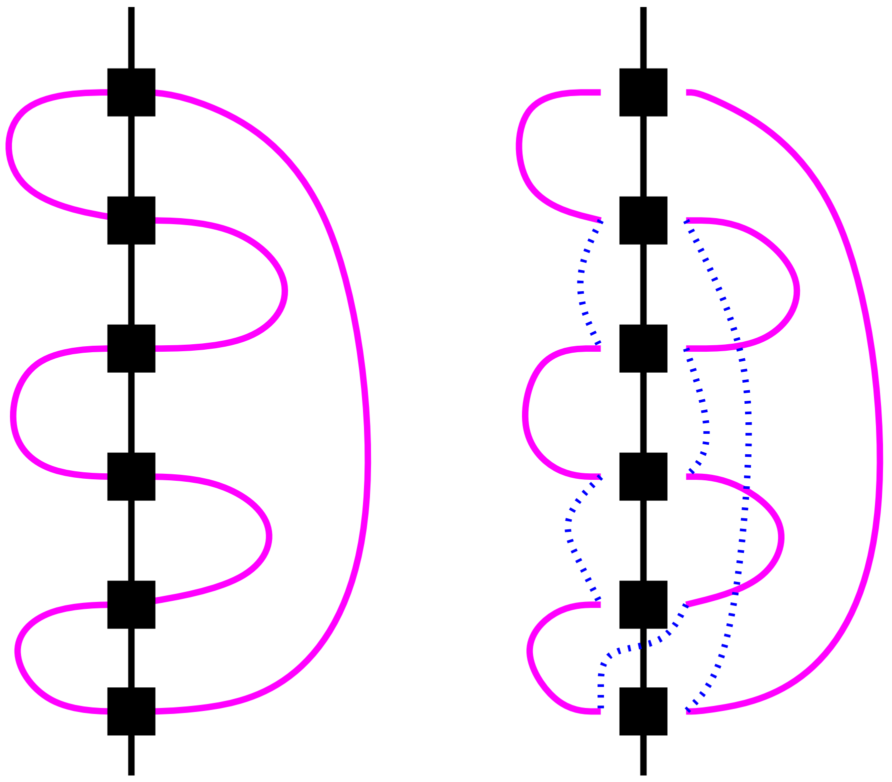
Patch: instead of considering all 2-light tours, consider only those that intersect each side of the boundary of a given square at most l times.

Thm $$\mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq \left(\frac{\log(n)+1}{m} + \frac{12}{l-5}\right) |\mathrm{OPT}|$$

Proof $\rightarrow$ key ingredient: patching lemma.

- reduce the # of crossings by dealing w/ several portals at once

- if line of crossings has length s, then path length increased by at most $3s$



Higher dimensions

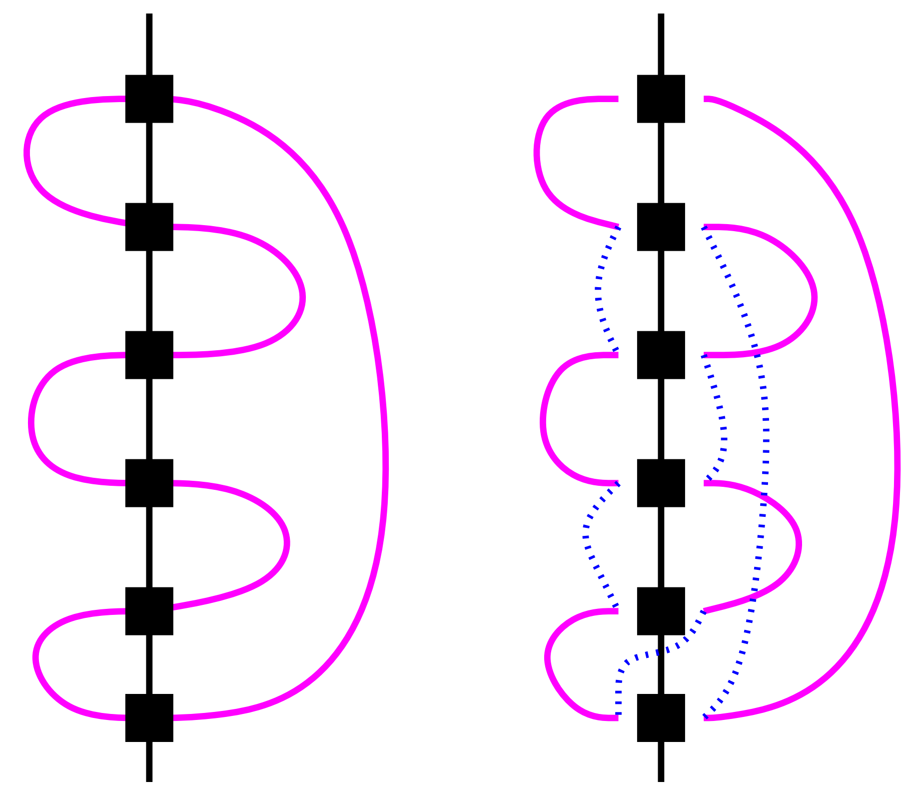
The analysis extends to higher dimensions, except for the *valid pairing* argument.

Patch: instead of considering all 2-light tours, consider only those that intersect each side of the boundary of a given square at most l times.

Thm $\mathrm{E}_{x,y}\left[|\mathrm{OPT}_p(l)| - |\mathrm{OPT}|\right] \leq \left(\frac{\log(n)+1}{m} + \frac{12}{l-5}\right)|\mathrm{OPT}|$

Proof $\rightarrow$ key ingredient: patching lemma.

$\rightarrow$ use patching lemma repeatedly, to reduce the total # of crossings of OPT when made portal-respecting, while amortizing the cost overhead due to patching.



Other norms

- Cannot reduce pb to Euclidean TSP:

$$\left(\begin{array}{l} C_1\ |.|_E \leq |.| \leq C_2\ |.|_E \\ \\ \rightarrow \text{get } T \text{ s.t. } |T|_E \leq (1+\varepsilon)|\mathrm{OPT}|_E \\ |T| \leq C_2|T|_E \leq C_2(1+\varepsilon)|\mathrm{OPT}|_E \leq \frac{C_2}{C_1}(1+\varepsilon)|\underbrace{\mathrm{OPT}}_{\text{Euclidean}}| \end{array}\right.$$

Other norms

- Cannot reduce pb to Euclidean TSP:

$$\left(\begin{array}{l} C_1\ |.|_E \leq |.| \leq C_2\ |.|_E \\ \\ \rightarrow \text{get } T \text{ s.t. } |T|_E \leq (1+\varepsilon)|\mathrm{OPT}|_E \\ |T| \leq C_2|T|_E \leq C_2(1+\varepsilon)|\mathrm{OPT}|_E \leq \frac{C_2}{C_1}(1+\varepsilon)|\underbrace{\mathrm{OPT}}_{\text{Euclidean}}| \end{array} \right.$$

- Algorithm and its analysis hold for any other geometric norm (modulo some constants factors in the optimal values of m and l).

$$\left(\begin{array}{l} \text{norm } (\neq \mathit{metric}) \text{ is important for scaling phase} \\ \\ \text{embedding in } \mathbb{R}^d \text{ is also important} \end{array} \right.$$

Recap

- Euclidean TSP admits a PTAS. *Idem* for TSP in $(\mathbb{R}^d, |.|)$.

- In $\mathbb{R}^d$, the PTAS given has space and time complexities of $O\left(n^{2d}(\log n)^{O\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)}\right)$

- Complexity is reduced to $O\left(n(\log n)^{O\left(\left(\sqrt{d}/\varepsilon\right)^{d-1}\right)}\right)$ if a reduced quadtree is used

- By using a $(1+\varepsilon)$-spanner of the input nodes to give better "hints" of what portals to use, one reduces the complexity to $O\left(n\left(\log(n) + 2^{\text{poly}(1/\varepsilon)}\right)\right)$ in $\mathbb{R}^2$ [RaoSmith]